# Semantically-enriched Pervasive Sensor-driven Systems [1]

**Editor(s):** Name Surname, University, Country
**Solicited review(s):** Name Surname, University, Country
**Open review(s):** Name Surname, University, Country

Juan Ye [a,*], Stamatia Dasiopoulou [b,**], Graeme Stevenson [a], Georgios Meditskos [b], Vasiliki Efstathiou [b], Ioannis Kompatsiaris [b], Simon Dobson [a],
[a] *School of Computer Science, University of St Andrews, UK*
*E-mail:{juan.ye, graeme.stevenson, simon.dobson}.st-andrews.ac.uk*
[b] *Information Technologies Institute, Centre of Research and Technology Hellas, Greece*
*E-mail:{dasiop, gmeditsk, vefstathiou, ikom}@iti.gr*

**Abstract.** Pervasive and sensor-driven systems are by their nature open and extensible, both in terms of their inputs and the tasks they are required to perform. The data streams coming from sensors are inherently noisy, imprecise, and inaccurate, with differing sampling rates and complex correlations with each other: characteristics that challenge traditional approaches to storing, representing, exchanging, manipulating and programming with rich sensor data. Semantic Web technologies allow designers to capture these properties within a uniform framework. The powerful reasoning techniques with such a representation facility have proven to be attractive in addressing issues such as data and knowledge modelling, querying, reasoning, service discovery, privacy and provenance. In this paper we review the application of the Semantic Web to pervasive and sensor-driven systems. We analyse the strengths and weaknesses of current and projected approaches, and derive a roadmap for using the Semantic Web as a platform on which open, standard-based pervasive, adaptive, and sensor-driven systems can be constructed.

Keywords: Semantic Web, Ontologies, Pervasive Computing, Sensor-Driven Systems, Context Modelling, Context Reasoning, Situation Recognition, Uncertainty, Provenance, Event Modelling, Semantic Service Discovery, Privacy and Trust, Programming

## 1. Introduction

The vision of pervasive computing was first articulated by Mark Weiser in 1991: ‘the most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it’ [224]. Two essential characteristics of this vision are that it is both *sensor-driven* and *adaptive*. Pervasive computing assumes an environment saturated with new classes of intelligent and portable devices with sensing, computation and communication capabilities. With the help of these devices, a pervasive system can gather information about users and their surrounding environments and – without explicit instructions from users – provide customised services that adapt to the users’ current contexts and needs. To be effective, pervasive computing must be supported by an open and standard-based representation so as to facilitate integrating information of heterogeneous types and modalities, as well as communicating and exchanging information between devices and components.

[1]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.
[*]Corresponding author.
[**]Corresponding author.

The key features of the Semantic Web, namely its ability to formally capture the intended semantics and to support automated reasoning, allow to elegantly share, integrate and manage knowledge coming from heterogeneous sources, perfectly satisfy a common requirement in pervasive, sensor-driven, and adaptive computing applications. Rendering meaning explicitly, the Semantic Web facilitates the exchange of data between systems and components in an open, extensible manner, maintaining semantic clarity across applications. Semantic Web technologies have successfully addressed some pervasive computing application concerns, such as representing complex sensor data [143], recognising human activities [52], and modelling and querying location data across heterogeneous coordinate systems [188]. Ontological reasoning has proven to be useful in manipulating structured conceptual spaces [23,194,227]. The use of ontologies, especially of upper and lower ontologies with appropriate connections [186], powerfully supports co-operation of data sources within an open system.

However, the potential of Semantic Web technologies to address other key pervasive computing application requirements is yet to be fully explored. Open issues include capturing the temporal semantics of data; reasoning in the presence of extensive uncertainty; querying and applying different reasoning schemes to highly dynamic data; capturing provenance – and integrating all these elements within a single, coherent framework. In this paper we explore these open issues and seek to answer three questions for each: to what extent do existing Semantic Web technologies address the requirements?, what additional techniques might be needed?, and how might the research community address these deficiencies?.

We structure our discussion as follows. Section 2 provides a background to pervasive sensing systems, including representative examples of applications, describes and analyses the characteristics and relationships of 'typical' information that needs to be modelled, including sensor data, context, events, domain knowledge and situations. It further derives requirements on modelling, processing, and reasoning on such information to satisfy various application requirements. Section 3 briefly introduces the Semantic Web and highlights the key enabling features for knowledge capture and reasoning in the open environment of pervasive systems. Section 4 analyses the use of Semantic Web technologies in terms of key pervasive application requirements, including modelling of sensor data, context (information about *when*, *where*, *who*, and *what*, events), reasoning towards higher-level context abstractions, uncertainty handling, semantic service discovery, and privacy and provenance. Section 5 identifies open research issues in pervasive computing which may potentially be addressed through the use of Semantic Web technologies. Section 6 concludes the paper.

## 2. Applications, Information and Research Challenges in Pervasive Computing

Pervasive computing aims to enable and assist people in accomplishing an increasing number of personal and professional transactions using intelligent and portable appliances, devices, and artefacts that allow them to employ intelligent networks and gain direct, straightforward, and secure access to both relevant information and services. It gives people access to information stored across potentially large-scale static and ad-hoc networks, allowing them to easily take actions anywhere and at any time. In the following, we introduce representative application areas to provide a general view of pervasive computing. Then we describe the typical information needs of a pervasive system and discuss the requirements and challenges to modelling and reasoning on it.

### *2.1. Application Areas*

Pervasive sensing, computing, and communication technologies are being applied within many domains everyday.

#### *2.1.1. Healthcare and Smart Homes*

Pervasive healthcare is an emerging research discipline, focusing on the development and application of pervasive computing technologies for healthcare and well-being. This research is driven by the pressing needs of an expanding ageing population, a predicted shortage in caregivers for the elderly, and the maturation of sensing and communication technologies [152].

Pervasive computing technologies introduce new diagnostic and monitoring methods that directly contribute to improvements in care, therapy and medical treatment. These examples involve sensors and monitoring devices that collect and disseminate information, such as heart rate, blood pressure and glucose levels, to healthcare providers [29]. They support a better understanding of facets of patients' daily lives, supporting appropriate adaptation of therapies to the individual. One scenario is a hospital where a patient is constantly monitored, with findings linked to a diagnostic process. Thus, it is possible to advise the hospital canteen to prepare special food for this particular patient and to adapt the patient's specific medication according to his current health condition. Pervasive computing technologies can also improve medical treatment procedures. In emergency care, they can accelerate access to medical records at the emergency site or seek urgent help from multiple experts virtually. In the surgical field, they can collect and process an ever-increasing range of telemetric data from instruments used in an operating room and augment human ability to detect patterns that could require immediate actions [33].

To maintain the health and safety of the home and residents and maximise the comfort of the residents, researchers deploy sensors in a home setting to perceive the state of the environment and its residences, and reason about the state using artificial intelligence techniques [59]. The DOMUS lab, at the University of Sherbrooke, achieves an implementation of smart home based pervasive assistants which act as mobile orthosis [60]. Its research goal is to collect ecological data on symptoms and medication side-effects on cognitively impaired people, establish their behaviour models, allow caregivers to monitor their daily activities, and provide assistance adapting to different types of cognitive deficits such as memory, initiation, planning, and attention [34].

#### *2.1.2. Transportation*

Pervasive computing technologies are entering our everyday life as embedded systems in transportation [63,82]. A number of applications have emerged. In tourist guides, a pervasive computing system can provide personalised services (like locating a particular type of restaurants or planning a day trip) for visitors based on their location and preferences [233]. In traffic control, a system can be immediately informed of incidences of congestion or the occurrence of accidents and notify all approaching drivers. In route planning, a system can suggest the most convenient routes for users based on the current traffic conditions and the transportation modes being used [231].

GPS-enabled devices like smartphones are changing the ways in which people interact with the Web by using their locations as context. Multiple users record their outdoor movements as GPS trajectories for travel experience sharing, life logging, sports activity analysis, and multimedia content management. This results in a large accumulation of GPS trajectories that are accumulating unobtrusively and continuously in Web communities [233]. One use of these data is to discover interesting locations or location sequences as tourist recommendations. Here 'interesting' means that a location is not only visited frequently by tourists but also is a meaningful place such as somewhere with cultural significance, a shopping mall, a restaurant, or a bar [233].

#### *2.1.3. Environment Monitoring*

Advances in sensing and wireless network technologies provide realistic solutions for continuous monitoring of natural environments. They potentially provide new data for environmental science as well as vital hazard warnings [97]. They allow the collection of fine-grained environmental data, which is essential to foster scientific research and to increase human understanding of the environment. For example, detailed air pollution data collection helps to understand pollutants' spreading mechanisms and their influence on human health [176]. In agriculture, sensors and actuators operating at a much finer level of granularity can be used to precisely control, for example, the concentration of fertiliser in soil based on information gathered from the soil itself, the ambient temperature, and other environmental factors [90].

Pervasive sensing technologies are also particularly important in remote, inhospitable, or dangerous environments where fundamental processes have rarely been studied due to their inaccessibility [132]. Some examples are: monitoring a chemical industrial environment to minimise the hazards of chemical wastage [123], monitoring gas emissions at landfill sites [117], and studying active volcanoes with collected seismic and infrasonic (low-frequency acoustic) signals.

#### *2.1.4. Other Applications*

Beyond the above three types of applications, pervasive computing has also been applied to other areas, including workplaces, education, and office environments [60]. In *education*, intelligent classrooms are prototyped to enhance the classroom experience by automating processes such as controlling light settings, playing videos, and displaying slides [229]. *Smart meeting* applications facilitate archiving, analysing, and summarising meetings via a variety of technologies from physical capture from camera, microphone, or other interactive tools and structural analysis to semantic processing [86].

### 2.2. Information in Pervasive Computing

Typical types of information in a pervasive sensor-driven system include raw sensor data, context, domain knowledge, and events. In the following, we will define these terminologies and analyse their features.

*Raw sensor data* are readings reported from sensors; e.g., a complete reading from Ubisense – a coordinate-based positioning sensor is `([4.54,1.92,4.70], tag_bob, 09/05/2008 09:49:39)`, including *when* the reading is generated (e.g., `09/05/2008 09:49:39`), *who* the reading is about (e.g., a tag belonging to the user named `bob`), and *what* the reading value is (e.g., a 3D coordinate-based location datum `[4.54,1.92,4.70]`).

Sensor data can be abstracted into a relation of well-structured concepts called *context*. Context describes a property of an environment or a user, such as *Location*, *Time*, *Person*, and *Resource*. Context provides a uniform way of representing sensor data, which makes sensor data sharable and reusable between different systems without concerning about heterogeneity and complexity of underlying sensing technologies. For example, the above sensor data can be interpreted in a relation between a *Person* and *Location* context: `(bob, isLocatedIn, studyRoom)`.

*Knowledge* here indicates a corpus of knowledge specific to available context, including spatial knowledge for location contexts, user preference and social network for person contexts, or domain knowledge to translate numeric sensor readings to human-understandable descriptions.

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 event of 'a user having heart attack' might trigger an application of calling an ambulance. A low-level event can be conceptually defined as an action or occurrence that happens at a certain time within an environment, which is described through basic elements such as *when*, *where*, *what*, and *who*. A high-level event includes other parameters including *role* and *factor*.

### 2.3. Research Challenges

A pervasive computing system involves a huge amount of information that exhibits variety in types, heterogeneity, dynamicity, uncertainty, and richness in relationships. It is challenging to modelling and processing them to satisfy the various application requirements. We summarise the following requirements that are particularly relevant to the Semantic Web technology community.

**Conceptual modelling** We require a commonly-agreed vocabulary to represent entities and data so as to make them understandable across different systems and platforms; e.g., populating GPS or social data in Web communities, or transferring health records across hospitals and medical organisations. Such a vocabulary should be rich enough to represent properties, structures, and relationships among information, thus facilitating querying, personalising and further processing of data, as well as their sharing and reuse. It will be used to semantically annotate data so not only are they represented in machine-processable formats but they also are 'meaningfully' related to other data; e.g., in the GPS example in Section 2.1.2, a place needs to be annotated not only as a location but also related to cultural or social aspects. More concretely,

- we need a sensor model that *provides machine-readable specifications of sensors, their output types, and the domains in which they operate* so that it can support classifying sensors according to their functionalities, finding a sensor that can perform a certain measurement, or even composing existing sensors to create a virtual sensor for particular functionality [58].
- we need a conceptual model with rich semantics to represent information at different abstraction levels: sensor data, context, and event. The sensor model is required to capture information aspects about observation data such as *when*, *what* and quality measures (e.g., sampling frequency, coverage area, precision or accuracy), as well as to enable the derivation of high-level knowledge from low-level sensor data. The context and event models are required to express uncertainty, temporal features, generalisation, composition, and causality relations. Take an example of intelligent classrooms in Section 2.1.4, representing the sequential relations between events like presentations *followed by* group discussions.

**Querying** Sensor data are produced in real time and frequently updated. It is expense to either store all the produced data in disk or transfer data to a storage server. Also sensor data can capture important states of an environment that needs to be immediately responded to. Thus processing real-time querying over such sensor data is critically necessary in a pervasive sensor-driven system; e.g., detecting an abnormal event in an environmental sensing scenario in Section 2.1.3. It is a challenge to online evaluate such queries due to the tremendously amount of sensor data and the resource constraints on performed devices.

**Reasoning** Reasoning is a key enabler for pervasive applications in that there is a large amount of information in broad types that needs to be filtered and abstracted to higher-level information that is understandable or interesting to applications or humans; e.g., inferring a patient's condition based on symptoms acquired from biochemical sensors, or analysing air pollution data to predict the pollution spreading direction so as to take immediate actions to mitigate its effects.

**Uncertainty** Data are inherently imperfect. Inaccuracies may easily arise, due to erroneous and/or missing sensor readings. Furthermore, when data come from multiple sources and modalities, ambiguities and conflicts may also 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, and derive plausible conclusions, assessing the validity of available sensor data.

**Service discovery** Pervasive and sensor-driven systems are usually assumed to operate in large-scale, open environments where components are loosely coupled and have volatile presence; i.e., they can enter or leave a system freely at any point. Thus, it is desirable to associate them with semantic annotations so as to support matching, discovery, composition, and orchestration in a more intelligent manner.

**Privacy and Provenance** Data in pervasive systems are typically associated with privacy concerns; especially personal data such as health records. We need a specification language to specify rules or policies about how to access, appropriately (perhaps automatically) abstract, and use data.

**Scalability and Performance** In a pervasive sensor-driven system, the scalability and performance issue is critical since data can be stored and processed on resource- and energy-constrained devices (e.g., sensors or mobile devices). Existing tools in semantic web technologies are assumed to be deployed on high performance servers. The challenge here is to develop infrastructures dedicated for small-scale applications running on much less powerful devices [64].

## 3. Semantic Web Technologies

The Semantic Web [22] is a resource-oriented extension of the current Web that aims to afford a common framework for sharing and reusing data across heterogeneous agents, applications and systems. The underlying rationale is to make the semantics of web resources explicit by attaching to them *metadata* that describe meaning in a formal, machine-understandable way. Within this vision, ontologies play a key role, providing consensual and precisely defined terms for the description of resources in an unambiguous manner.

In 2004, the Web Ontology Language (OWL) [214] became a W3C recommendation, paving the way for the development of adequate tool support (ontology editors, reasoners, etc.) and the proliferation of ontology-based applications in a plethora of domains. Formally founded in Description Logics (DLs) [14], OWL is endowed with expressive representational constructs that allow to capture complex knowledge. At the same time, OWL avails of the well-defined DL reasoning services for affording automated reasoning support.

The aforementioned furnish OWL with a multiplicity of appealing, within the context of pervasive application features. For example, in OWL one can effectively model and reason over taxonomic knowledge. This is a desirable feature in pervasive applications where the need to model information at different levels of granularity and abstraction, so as to drive the derivation of successively further detailed contexts is particularly evident. Similarly, OWL supports consistency checking, another useful feature when dealing with imperfect context information coming from multiple sources.

Last but not least, OWL adheres to the open-world assumption, providing an elegant way of modelling incomplete information. Intuitively, open-world semantics assumes that we do not have complete information about the world, but that some information may be missing. Such assumption is well-suited for sensor-driven systems, where information may be incomplete due to a number of reasons, such as sensor inaccuracies or imperfect observations as described in Section 2.3.

### 3.1. Description Logics

The design of OWL have been strongly influenced by Description Logics (DLs) [107]. DLs are a family of knowledge representation formalisms characterised by logically grounded semantics and well-defined reasoning services [14]. 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.

A DL knowledge base K typically consists of a *TBox T* (terminological knowledge) and an *ABox A* (assertional knowledge). The TBox contains axioms that capture the possible ways in which objects of a domain can be associated. For example, the TBox axiom `Dog` $\sqsubseteq$ `Animal` asserts that all objects that belong to the concept `Dog`, are members of the concept `Animal` too. The ABox contains axioms that describe the real-world entities through concept and role assertions. For example, `Dog(Jack)` and `isLocated(Jack,kitchen)` express that Jack is a dog and he is located in the kitchen. The reader is referred to [14] for more details on the syntax and semantics of DLs.

### 3.2. Reasoning Services

Besides formal semantics, DLs come with a set of powerful reasoning services, for which efficient, sound and complete, reasoning algorithms, with well-understood computational properties, are available (e.g., tableaux-based algorithms [15]). Example state-of-the-art implementations include Racer [96], Hermit [140], Pellet [183], and Fact++ [210].

Typical DL reasoning services include *subsumption*, *satisfiability*, *consistency*, *instance checking* and *realisation*. Through *subsumption* one can derive the implicit taxonomic relations among the concepts of a terminology, for example that `Room` subsumes `OccupiedRoom`. *Satisfiability checking* enables the identification of concepts for which it is impossible to have members under any interpretation (for example, an unsatisfiable concept, though trivial, is `OccupiedRoom` $\sqcap$ $\neg$`OccupiedRoom`). *Consistency checking* enables the identification 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 finding 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 [154], reasoning in DLs (and hence in OWL) adopts the *open-world assumption*. 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 amounts to concluding that `Alice` is the sole resident of this house.

### 3.3. OWL and OWL 2

OWL comes in three dialects of increasing expressive power: OWL Lite, OWL DL and OWL Full [10,105]. 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 language was designed to provide full compatibility with RDF(S), which is the most expressive of the three. It neither imposes any constraints on the use of OWL constructs, nor lifts the distinction between instances (individuals), properties (roles) and classes (concepts). This high degree of expressiveness comes however at a price, namely the loss of decidability that makes the language difficult to implement. As a result, focus has placed on the two decidable dialects, and particularly on OWL DL, which is the most expressive of the two.

Despite the rich primitives provided for expressing concepts, OWL DL has often proven insufficient to address the needs of practical applications (see section 2 in [91]). Furthermore, OWL can model only domains where objects are connected in a tree-like manner [215]. This constraint can be restrictive for real-world applications, including the pervasive domain, which requires modelling general relational structures (see Section 4.6).

Responding to this limitation and to other drawbacks that have been identified concerning the use of OWL in different application contexts throughout the years, the W3C working group produced OWL 2 [91]. OWL 2 (equivalent to the $\mathcal{SROIQ}(\mathcal{D})$ DL) is a revised extension of OWL, now commonly refereed to as OWL 1. It extends OWL 1 with qualified cardinality restrictions; hence 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. Three profiles, namely OWL 2 EL, OWL 2 QL and OWL 2 RL, trade portions of expressive power for efficiency of reasoning targeting different application scenarios.

Another prominent OWL 2 feature is the extended relational expressivity that is provided through the introduction of complex property inclusion axioms (property chains). To maintain decidability, a regularity restriction is imposed on such axioms that disallow 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.

### 3.4. Combining Ontologies and Rules

To achieve decidability DLs, and hence OWL, trade some expressiveness for efficiency of reasoning. The tree-model property, mentioned above, is one such example and as a result, it is not possible to describe classes whose instances are related to an anonymous individual through different property paths [92]. To leverage OWL's limited relational expressivity and overcome modelling shortcomings that OWL alone would be insufficient to address, a significant body of research has been devoted to the integration of OWL with rules.

A proposal towards this direction is the Semantic Web Rule Language (SWRL) [106], in which rules are interpreted under the classical first order logic semantics. Allowing concept and role predicates to occur in the head and the body of a rule without any restrictions, SWRL maximises the interaction between the OWL and rule components, but at the same time renders the combination undecidable. To regain decidability, proposals have explored syntactic restrictions on rules [171,139] as well as their expressive intersection [92]. The DL-safe rules introduced for example in [139] impose that rule semantics apply 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 research area in the Semantic Web addresses the seamless integration of open and closed world semantics. Representative initiatives in this quest include among others the hybrid formalism of MKNF knowledge bases proposed by Motik and Rosati [138], the extension of ontologies through the use of integrity constraints proposed by Tao et al. [200], and the so-called grounded circumscription approach [122].

Taking a different perspective, a number of approaches have investigated the combination of ontologies and rules based on mappings of a subset of the ontology semantics on rule engines. For instance, Horst [203] defines the *pD** semantics as a weakened variant of OWL Full, e.g., classes can be also instances, and in [204] they are extended to apply to a larger subset of the OWL vocabulary, using 23 entailments and 2 inconsistency rules. Inspired by the *pD** entailments and DLP [92], the semantics of the OWL 2 RL profile is realised as a partial axiomatisation of the OWL 2 semantics in the form of first-order implications, known as OWL 2 RL/RDF rules. User-defined rules on top of the ontology allow to express richer semantic relations that lie beyond OWL's expressive capabilities, and couple ontological and rule knowledge.

Expressing rich semantic relations is essential for sensor driven systems in the pervasive domain where 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.

### 3.5. Summary

This section 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 to elegantly capture 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 to represent richer semantic relationships. Given the inherently open nature of pervasive, sensor-driven systems, where the need to integrate domain knowledge and meaningfully aggregate and interpret low-level context information is a crucial requirement, it comes as no surprise that Semantic Web technologies have been acknowledged as affording a number of highly desirable features.

## 4. Integrating Semantic Web with Pervasive Systems

The aforementioned outlined key Semantic Web features and research challenges. In the following we explore in detail how these features meet the requirements of typical pervasive systems. Since the conceptual modelling is the foundation for a pervasive system, we will break this requirement into three subsections: modelling sensor and sensor data in Section 4.1, modelling context in Section 4.2, and modelling events in Section 4.3. The rest of the section is organised as semantic complex event processing in Section 4.4, querying in Section 4.5, reasoning in Section 4.6, temporal and stream reasoning in Section 4.7, uncertainty in Section 4.8, semantic service discovery in Section 4.9, and privacy and provenance in Section 4.10, closing with a summary in Section 4.11.

### *4.1. Modelling Sensor and Sensor Data*

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 [42]. The heterogeneity leads to huge difficulty in integrating and query over multiple sensor networks. The main contributions of Semantic Web technologies to modelling sensor and sensor data are providing uniform syntactic representations for sensors and their observations, enhancing the semantic meaning of sensor observations, reasoning over sensor data to derive new knowledge, enabling query over live sensor streams, and discovering and composing relevant sensors. This section will focus on the first two topics, and leave the reasoning and querying topics to Section 4.3, 4.4, 4.5, 4.6 and 4.7.

#### *4.1.1. Uniform Syntactic Representation*

There exist small-scale sensor ontologies as uniform syntactic representations for a particular domain, for example Sensory Data Set Description Language (SDDL) [108] and ontologies in $\mathcal{PI}$ [228] for representing data sets collected from sensors deployed in smart home environments. These ontologies mainly aim towards facilitating sharing and reusing data between researchers.

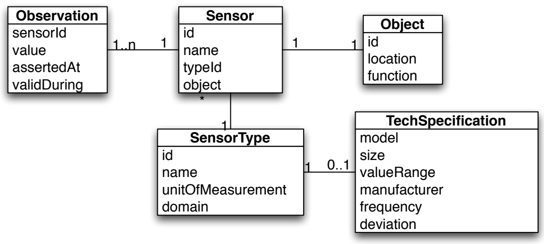


Fig. 1. The sensor and quality specification model in $\mathcal{PI}$ (source: [228])

$\mathcal{PI}$ is a generic conceptual model to model sensors and their quality measure, and link them with domain ontologies on *Location*, *Domain* and *Object*, etc. For example in Figure 1, each sensor type is associated with a *domain*, describing what the sensor type measures (e.g., acceleration or usage of gas), and a technical specification that includes its *manufacturer*, *model*, *size*, *deviation* of readings produced by its sensors (e.g., 'a maximum precision of $\pm$ 1.5% full scale' for a gas flow sensor), its sampling *frequency*, and a number of *valueRange* parameters – boundary values that characterise different states. $\mathcal{PI}$ 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")]
```

The Sensor Web Enablement group (SWE) [199] of the Open Geospatial Consortium (OGC) proposes a more generic framework of data models and encodings for describing sensors and sensor observations, as well as a suite of web service interfaces leveraging these models and encodings [167].

The SWE specifications include three core languages – Sensor Model Language (SensorML), Observations & Measurements (O&M), and Transducer Model Language (TransducerML) – to provide standard models and XML schemas for encoding sensors, sensor systems, process, and sensor observations, etc. For example, SensorML provides classes and associations to create specific sensor profiles that facilitate the processing, geolocation and integration of observations collected from various sensors [174]. SensorML is aimed towards generality, which specifies as few classes and associations as possible. To support a rich semantic data and knowledge model, some classes in SensorML need to be re-conceptualised, re-defined, or extended [174].

The SWE also provides a suite of service interface specifications to publish data, locate data, and receive alerts about data, including Sensor Observations Service (SOS), Sensor Planning Service (SPS), Sensor Alert Service (SAS), and Web Notification Services (WNS) [182]. For example, the SOS provides a web service interface for the pull-based access to archived and near-realtime sensor observations and metadata [37].

The SWE specifications enable a standardised communication and interaction with arbitrary types of sensors and sensor systems. They make sensors available over the Web through well-defined formats and Web service interfaces by hiding the sensor communication details and the heterogeneous sensor protocols from applications working on top of these services [38]. They aim at standardising the discovery, exchange, and processing of sensor observations [37]. 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 [223].

#### *4.1.2. Semantics Enhancement*

To extract new knowledge from raw sensor data, we need to enhance semantic meanings on top of them. This research has been developed through several stages, from relating sensor ontologies to domain ontologies [173,78], to focusing on analysing spatial, temporal and thematic semantics of sensor data [182,223], and to applying the 'linked data' principle [223,111] when more and more public ontological sources are available.

OntoSensor [173] is one of the early works, built on SensorML, the IEEE Suggested Upper Merged Ontology (SUMO) [198] and ISO 19115 [110]. It is a comprehensive deep sensor ontology, which includes a domain theory expressed in a language that is 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. For example, a thermal sensor is associated with an attribute `sample_interval` on which a processing service `detectAlarm(threshold)` is defined; that is, if the thermal expansion or contraction at regular sample intervals are beyond a threshold, then an alarm is reported. OntoSensor also includes more advanced inference mechanisms that can be used for synergistic fusion of heterogeneous data.

Semantic Sensor Web (SSW) [182] provides an enhanced meaning for sensor observations by adding semantic annotations to the SWE standards, 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 time instant or interval when the sensor data is captured, while the thematic metadata describes a real-world state extracted or abstracted from sensor data; for example, objects or events mentioned in Section 2.2.

Following the semantic annotation idea in SSW, the 'linked data' principle presents a more general way by creating RDF links between sensor data and concepts on the Semantic Web and Social Web [111,223]; that is, the concepts published by authoritative sources (e.g., DBpedia), or user-generated contents (e.g., tags) on the Social Web. Such annotation enables reasoning over the sensor data and the linked data to provide advanced sensor data query and retrieval functions. Janowicz et al. [111] present a Linked Data model and a RESTful (Representational State Transfer) proxy for the SOS to improve integration and inter-linkage 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 Semantic Web reasoners and repositories within the OGC services and thus to enable a transparent and seamless integration of Semantic Web technologies with the Spatial Data Infrastructure [141].

More sensor ontologies can be found in a review done by the W3C [220] and Compton et al [58]. Based on the review, the W3C Semantic Sensor Network Incubator group [219] has developed the Semantic Sensor Network (SSN) ontology [57]. It targets at the formal and machine-processable representation of sensor capabilities, properties, observations and measurement processes, with which it aids in searching, querying, and managing the sensor network and its data. Central to the ontology is the Stimulus-Sensor-Observation (SSO) ontology design pattern [112] that provides a lightweight model for representing sensors, their inputs (called stimulus) and observations. SSO is reusable 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 [76] so as to enable the integration into more complex ontologies as a common ground for alignment, matching, translation, and interoperability.

#### *4.1.3. Analysis*

The above sensor-centric and observation-centric ontologies have developed from a syntactic model to a semantic and knowledge model. 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, while more standard solutions like SensorML and SSN aim to standardise interfaces for services and description languages for sensors and their processes to enable syntactic interoperation [58].

As introduced in Section 4.1.2, various sensor ontologies have been developed to derive knowledge from raw sensor data. As one of the initiatives, OntoSensor supports ontology-driven inference by combining the domain conceptual model with the syntactical standards. SSW applies a more general analysis on semantic dimensions of sensor data; that is, the spatial, temporal, and thematic aspects. The principle of linked data presents as a promising approach to enhance semantics of raw sensor data by linking them to domain concepts on other standard resources. SSP aims towards a light-weight solution to support on-the-fly integration of sensor data to the Sensor Web.

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 [58]. However, the full range of features envisaged for semantic sensor web has not realised.

### *4.2. Modelling Context*

The nature of pervasive computing is such that the applications encompassed by this broad term are vastly heterogeneous in nature and broad in their data requirements and scope. Hence, in an open environment it is impossible to define, without *a priori* knowledge of the applications that will use it, a 'complete' model of content. However we may straightforwardly observe that a large number of applications exhibit overlapping data requirements, the most common of which are the need to represent time, location, actors, and resources. A vocabulary for time supports the temporal localisation of entities; in supporting spatial localisation, location has multiple representations, each suited to particular classes of application; people and agents are central actors in pervasive applications; and, the notion of a resource encompasses myriad things, both physical and virtual with which a user or application may interact. These neatly correspond to the notions of *when*, *where*, *who*, and *what* introduced in section 2.2.

#### *4.2.1. 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, car journey duration) correspondence to physical time.

The most common mechanical representation for physical time is the ISO 8601 standard [109], 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 by this standard are adopted by XML schema, which RDF adopts as a means of typing data 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 [6] 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 [150], 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') to make mechanical extraction of and reasoning on this information easier. However, we 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) [48], one of the well-known ontologies in pervasive computing, supports a formal, well-structured way to model context and thus provides rich semantics for programming. It borrows terms from other standard domain ontologies such as FOAF [77], DAML-Time [102], OpenCyc [126], RCC [32,56], and the Rei Policy [115] Ontology. SOUPA's temporal predicates are based on DAML-Time (OWL-Time's predecessor). Ontonym [186] on the other hand, a set of upper ontologies [20] that represent core concepts in pervasive computing, adopts OWL-Time directly in its modelling of temporal concepts. Temporal concepts may be straightforwardly 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.

#### 4.2.2. *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*, to *symbolic*, all of which can be related, and with specific forms more appropriate than others for any given application [72]. Location is often regarded as the most important type of context, and location models and query frameworks had been extensively researched, before the emergence of the Semantic Web (e.g., [103,101,114,165]).

The CONtext ONtology (CONON) [93] is an ontology-based context model with ontologies including a common upper ontology for the general concepts in pervasive computing (such as person, location, computing entity and activity) as well as domain-specific ontologies that apply to different subdomains like smart homes. Aspect-Scale-Context (ASC) is a contextually based model [195]. 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 [196]. The CONON location ontologies exist primarily to differentiate between indoor and outdoor spaces, while ASC's location ontologies have limited scope, supporting the representation of spatial positions in either the WGS84 and Gauss-Kreuger coordinate systems, and representing the distances between positions.

More comprehensive is SOUPA's location ontology, which is formed through the conjunction of two existing location vocabularies, OpenCyc [126] and Region Connection Calculus (RCC) [32,56]. The OpenCyc vocabulary supports the symbolic representation of spaces, while RCC provides a set of spatial relations (e.g., overlap, disconnection, tangental part) that may hold between two regions—essentially the location equivalent of Allen's temporal relations discussed above. Ontonym [186] 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. [114]) and incorporates a notion of relative positioning between entities based on compass directions and distance. The LOC8 framework uses this model to support conversion of positioning data between coordinates, symbolic locations and relative positions. It also provides an API for querying for the position of entities, returning a list of entities within a spatial region, and generating paths between two points in the model [188].

#### 4.2.3. *Who—Person/Agent*

This context type broadly relates to the actors in a pervasive system, which are typically people or agents whose reputations are manifest by software. The data representation requirements associated with these entities are usually application (or a class of applications) 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. SOUPA builds upon the FOAF [77] to support the expression of human profile information (name, age, contact details), and the MoGATU BDI ontology [157] to support the description of agent state – their beliefs, desires, and intensions – the detail of which is an application specific concern. Ontonym's person vocabulary is based on a subset of terms from vCard [216], W3C PIM [218] 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 [85] to define relations between people covering genetic, working, romantic, residential, and friendship connections.

#### *4.2.4. 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 without computational power, or a virtual artefact. While, for the most part, the properties of a resource one might wish to model are 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 vocabulary [66] provides terms for describing aspects of resources such as their creators, representation format, and licensing, and the GoodRelations ontology [99] provides an e-commerce vocabulary for describing products and services.

#### *4.2.5. Analysis*

This section has evaluated the way in which the most common elements of primary context in pervasive computing are represented. While one can feasibly conceive of the widespread adoption of a single, shared temporal model and location model upon which event descriptions are based, the specification of the actors and resources in a pervasive environment will remain 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 [27] plays a critical role in supporting the integration of entities with diverse descriptions across applications and environments.

### *4.3. Modelling Events*

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 is of real interest. Such low-level events are produced directly by the sensors or after low-level data processing. However, high-level, real-world complex events, such as situations and human activities cannot be recognised or processed, due to the absence of an expressive domain knowledge that would enable the further correlation of events and multi-modal information beyond predefined patterns and attributes. Moreover, in heterogeneous environments with a large volume of sensors where data is gathered from heterogeneous systems with potentially different terminologies, the representation, integration and correlation of event knowledge raise new challenges.

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 relationships, solving interoperability problems and providing the means for high-level event interpretations based on ontology reasoners (see section 4.4). This section reviews existing domain-independent ontologies for event modelling, presenting the different design patterns and expressive capabilities they provide.

#### *4.3.1. SOUPA*

Events in SOUPA [48] are modelled as instances of the `soupa:Event` class that represents a set of all events in the domain. However, this class is time and space agnostic in the sense that it does not provide properties for directly associating events with temporal and spatial information.

Spatiotemporal information in SOUPA can be modelled in terms of the `soupa:SpatialTemporalThing` class that is defined as the intersection of the `soupa:TemporalThing` and `soupa:SpatialThing` classes. Particularly for events, the `soupa:-`

Table 1
Comparison of the event ontologies presented in section 4.3

| Ontologies | Time | Location | Participation | Mereology | Causality | Correlation | Interpretation |
|---|---|---|---|---|---|---|---|
| SOUPA | √ | √ | √ | – | – | – | – |
| Ontonym | √ | √ | √ | – | – | – | – |
| Event Ontology | √ | √ | √ | *limited* | *limited* | – | – |
| SEM | √ | √ | √ | *limited* | – | – | √ |
| Event-Model-F | √ | √ | √ | √ | √ | √ | √ |
| LODE | √ | √ | √ | – | – | – | – |

SpatialTemporalEvent class is defined as the intersection of the soupa:Event and soupa:SpatialTemporalThing.

#### 4.3.2. *Ontonym*

The event ontology of Ontonym [186] provides a means of describing activities of interest within a system. The ontonym:Event class 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. Spatial information on events can be expressed through the ontonym:SpatioInstantEvent and ontonym:SpatioIntervalEvent classes. The temporal and spatial information in Ontonym is expressed using its Time and Location ontology constructs introduced in above section.

A role played by a person, device, resource, or object in an event is represented as an instance of the ontonym:Role class. The ontonym:containsRole property identifies the types of roles that can be taken in an event and the ontonym:playsRole property is used to associated an entity with an event.

#### 4.3.3. *The Event Ontology*

The Event Ontology (EO)[1] is a simple upper-level ontology for modelling events, defining three classes (eo:Event, eo:Product and eo:Factor) and eighteen properties. An event may have sub-events (eo:sub_event), actively participating agents (eo:agent), passive factors (eo:factor), products (eo:product, i.e. everything produced by an event), and a location in space (eo:place) and time (eo:time) by reusing the WGS84 Geo Positioning[2] and OWL-Time ontologies, respectively.

The Event Ontology has been mainly used for defining and interlinking events in the LOD cloud [181] and for annotating and extracting multimedia events, such as recordings and compositions, being part of the Music Ontology[3]. For example, an extension of the Music Ontology has been used in [184] in order to develop an ontology-based music recommendation system based on user's emotions/moods that combines user profiles and event (situation) reasoning.

#### 4.3.4. *Simple Event Model*

The Simple Event Model (SEM) [213] is an effort to define an ontology model for events without strong semantic constraints, affording in one hand model reusablity, compromising on the other hand automated reasoning and validation capabilities. Figure 2 presents the relationships among the classes of the SEM ontology.

SEM defines four core classes: sem:Event (for modelling events), sem:Actor (who or what participated in an event), sem:Place and sem:Time (capturing where and when the event takes place). Each core class is associated with the sem:Type class that is used to aggregate implementations of type systems from other ontologies promotig for reusability of existing type vocabularies.

There are two main property types in SEM: sem:EventProperty and sem:type. The former is used to correlate instances of the sem:Event class with other instances of arbitrary classes and the latter correlates instances of the sem:Core class with instances of the sem:Type class. There are also two aggregation relations. The sem:hasSubEvent can be used to define specialisation of events. Similarly, the sem:hasSubType relates instances of the sem:Type class. Finally, there are seven time-related properties, for single time values and time intervals,

[1] http://motools.sourceforge.net/event/event.html
[2] http://www.w3.org/2003/01/geo/wgs84_pos
[3] http://www.musicontology.com

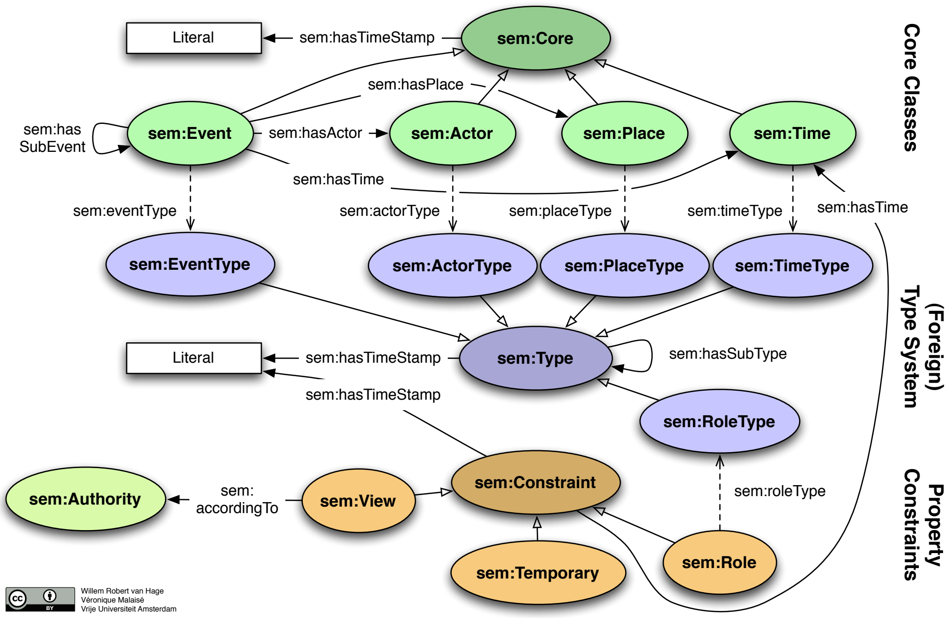


Fig. 2. The classes of the Simple Event Model ontology [213]. Arrows with open arrow heads symbolise `rdfs:subClassOf` properties and regular arrows visualise `rdfs:domain` and `rdfs:range` restrictions

also allowing representation of information for uncertain time intervals.

Property constraints in SEM can be defined either as a reification of the property or by turning the property into n-ary relation. The classes `sem:Role`, `sem:Temporary` and `sem:View` are three types of SEM constraints for defining the role of a class instance (e.g., `sem:Actor`) in an event, the temporal boundary within which a property holds and the different points of view, respectively. More expressive relationships among events, such as causality [177], or active/passive actor participation semantics cannot be modelled in SEM.

SEM defines also mappings of the model on other ontologies, e.g., LODE [181], based on the SKOS vocabulary. Furthermore, the developers provide a Prolog API for SEM in order to ease the procedure of populating the ontology with instances.

#### 4.3.5. Event-Model-F

Event-Model-F [177] defines an expressive model for capturing and representing occurrences in the real world. It is based on DUL, following the descriptions and situations ontology design pattern (DnS) [87] for modelling aspects of events, such as object participation, mereological, causal, and correlative relationships, and different interpretations of the same event (by reifying events in order to describe n-ary relations).

The Event-Model-F ontology introduces six ontology design patterns. The *Participation Pattern* is used to model the participation of objects (`dul:Object`) in events (`dul:Event`); the *Mereology Pattern* is used to model the composition of a composite event (`F:Composite`) out of its component events (`F:Component`); the *Causality Pattern* is used to express relationships between events that play the roles of causes (`F:Cause`) and effects `F:Effect`; the *Correlation Pattern* defines the role `F:Correlate` that `classifies` correlated events (i.e. events that have a common cause and there is no cause-effect relationship among them (causality)); the *Documentation Pattern* is used to provide documentary evidences for events (where evidence may be a specialisation of the `Object` class or another event); and finally, the *Interpretation Pattern* is used for modelling different viewpoints on which the perception of an event may depend; this can be expressed by instantiating accordingly the aforementioned designed patterns and binding them with the interpretation pattern.

#### 4.3.6. Linking Open Descriptions of Events

The Linking Open Descriptions of Events (LODE) ontology [181] provides a model for representing and publishing events that have happened in the past as Linked Data [27]. The model has been designed in such a way, so as to allow the modelling of four basic aspects of events: what, where, when, and who. The LODE vocabulary is aligned with other event-related vocabularies and ontologies, such as DUL, Event Ontology (EO) and CIDOC [73], and consists of one class for representing events (`lode:Event`) and seven properties. In contrast to Event-Model-F, LODE focuses on representing stable characteristics of events, excluding parthood and causal relations that belong to the interpretative dimension of event modelling.

The `lode:Event` class is defined as subclass of the `E2_Temporal_Entity` class of CIDOC and equivalent to the `eo:Event` and `dul:Event` classes. In this way, an event in LODE represents something that happened over a limited extent in time without specifying a change of state or attempting to distinguish events from processes or states.

In LODE, an event can be associated with at most one interval of time through the `lode:atTime` property that is defined as a subproperty of the CIDOC's `P4.has_time-span` and `dul:isObservableAt` properties. It is also equivalent to `eo:time` and the time intervals are defined using the `TemporalEntity` class of the OWL-Time ontology.

The `lode:inSpace` property relates an event to some spatial boundary (at most one), i.e. a region of space, and it is defined as a subproperty of `dul:hasRegion`, equivalent to `eo:place` and super-property of CIDOC's `P7.took_place-at` property. Events can be also related to abstract places through the `lode:atPlace` property that is subproperty of `dul:hasLocation`.

The association of events with objects and agents is performed thought the `lode:involved` property (equivalent to DUL's `hasParticipant`) and its subproperty `lode:involvedAgent` (equivalent to DUL's `involvesAgent` and super-property of `eo:agent`), respectively.

#### *4.3.7. 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 able to semantically represent and correlate common aspects of events (e.g., places, people, objects) amenable to reasoning (see section 4.6), and thus to high-level interpretation.

This section briefly presented existing event ontologies, focusing on the ontology constructs and design patterns they provide. 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 (section 4.3.5). SOUPA and Ontonym follow a modular design with a moderate axiomatisation compared to the Event Ontology, LODE and SEM ontologies. Among them, only SEM is capable of capturing different interpretations of the same event. Table 1 summarises the strong and weak points of each ontology.

### *4.4. Semantic Complex Event Processing*

State-of-the-art *Complex Event Processing (CEP)* frameworks [172] 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 that originate either directly by sensors or after low-level feature processing. The data of the event streams are checked for specific values or correlations with data from other streams in order to identify and extract useful patterns that are described using rules. The key feature is the support of temporal relationships (e.g., Allen's temporal interval relationships) and aggregation operators that enable the identification of complex correlations among the generated events.

*Semantic Complex Event Processing (SCEP)* is an effort to improve the results of CEP by incorporating ontologies and rules into the process of complex event detection [137]. An abstract SCEP architecture is presented in Figure 3. Ontologies are used in this context as common vocabularies for representing knowledge relevant to events, such as event patterns, participants and their relationships, solving interoperability problems. 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, such as the ones mentioned in section 4.5, can be used to process streams of events and uncover temporal correlations. In SCEP frameworks, these semantic correlations of events with background knowledge are further enhanced with rules so as to overcome expressive limitations of the underlying ontology languages, as described in section 3.4.

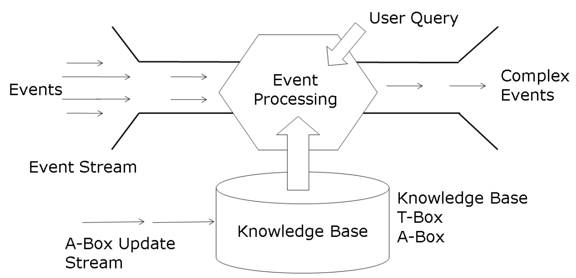


Fig. 3. An abstract SCEP architecture (source: [153])

In [202], a commercial CEP engine (*Coral8*[4]) is combined with ontology-based reasoning. The platform aims at the semantic configuration of the CEP engine using domain ontologies about events, sensors, environments, etc. The event ontology extends SSN 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 [205] present a framework for semantic rule-based complex events processing. The ontology based representation of event data is performed by a map of the plain 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[5] is used in order to implement temporal reasoning operators and to perform reasoning on the ontology-based knowledge.

ETALIS [9] 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 [7]. EP-SPARQL (see section 4.5) is used in ETALIS as the underlying event processing and stream reasoning language to detect events within a stream of RDF triples (low-level events). Based on the EP-SPARQL extensions of SPARQL relevant to binary operators (SEQ, EQUALS, OPTIONALSEQ, and EQUALSOPTIONAL) graph patterns are combined with a set of functions (e.g. `getDURATION`, `getSTARTTIME`) for time-based filtering expressions. The queries are compiled into event-driven backward-chaining rules that can be mixed with other background knowledge.

### 4.5. Querying

The standard query language for working with RDF data is SPARQL [178]. However, 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.

Semantic Streams [225] is an approach to allow users to query streaming event data, using a Prolog engine to hold semantically marked-up data, and evaluate queries against it. Bouillet et al. define a semantic extension to System-S, a stream processing framework for sensor networks [35]. The approach connects sensor data to applications via a number of processing elements that perform filtering, composition, and complex event processing. Data are represented using OWL, and queried by means of well known ontological relations.

Tappolet et al. introduce a syntactic sugar extension to SPARQL to facilitate the querying of graphs valid at a certain time point [201]. McBride et al. [133] also attach temporal extent to objects, upon which they provide a syntactic sugar in query language that maps to SPARQL. Rodriguez et al. introduce a scheme for storing, indexing and querying temporal data designed for sensor networks [3]. Perry et al. [158] introduce the syntax and semantics of an extension to SPARQL, called *SPARQL-ST*, designed for the execution of spatiotemporal queries. O'Connor et al. develop 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 [145,146]. 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.

A number of SPARQL extensions based on the notion of continuous querying have been proposed. This represents a shift away from one-time-only processing to a framework within which queries are evaluated

[4] `http://www.aleri.com/products/aleri-cep/coral8-engine`
[5] `http://http://prova.ws/`

continuously against new data being produced, with query results updated as new matches are discovered.

Streaming SPARQL [30] extends the SPARQL grammar and algebra to support streaming data sources and query processing based on sliding time windows on such sources. Continuous-SPARQL (C-SPARQL) [17,19] also extends the SPARQL grammar adding temporal windows and aggregation operators. The evaluation of C-SPARQL queries involves decomposing the query into static and dynamic components: the former is used to query the reasoner using regular SPARQL, while the latter is translated to CQL [11] and used to query dynamic data which is held in a relational model.

SPARQL Stream [41] provides ontology-based access to stream data via continuous queries. The process involves mapping both the ontological data and queries to SNEEql [36] an existing SQL-based extension for processing streaming data. The SPARQL grammar is extended to support the expression of time windows and an optional step-time gap within the window for successive evaluations.

Continuous Query Evaluation over Linked Streams (CQELS) [125] is a query processor for linked stream data. It extends the SPARQL grammar to support continuous queries evaluated via a sliding time window set on the current time or triggered on the occurrence of a triple, and processes the queries natively rather than mapping to existing relational or logic-based query frameworks.

Event Processing SPARQL (EP-SPARQL) [8] provides a logic-based approach to reasoning on streamed data that focuses on detecting RDF triples occurring with a specific temporal ordering. Both data, and the extended SPARQL language are mapped to a Prolog engine where backwards-chaining inference rules are used to evaluate queries against the data.

These works are among the most relevant to the querying of data in sensor driven systems, however other extensions to SPARQL, unrelated to data's temporal semantics, have been proposed, including extensions to: support path queries [4,120], support navigation of RDF graphs via nested regular expressions using RDFS properties [12]; support Fuzzy semantics [151,53]; integrate RDF data seamlessly with XML data [129]; and integrate data-mining using statistical relational learning [116].

#### *4.5.1. Analysis*

Given the general purpose nature of the RDF model, it comes as no surprise that the SPARQL language 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, namely, 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 by which query languages may be adapted or extended to target specific needs of the pervasive domain. However for this approach to succeed beyond the prototype stage requires the widespread adoption of modelling standards for said domains (for example, 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.

### *4.6. Reasoning*

The higher-level integration of raw context data and 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 directly acquired from sensors is translated to respective ontological class and property assertions. Hence, typical ontology reasoning tasks can be used to check 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 her current location and objects used) that would otherwise remain implicit. An abstract ontology-based framework for modelling and reasoning about high-level context is depicted in Figure 4.

In addition to adopting plain ontology-based solutions though, a number of approaches have explored reasoning frameworks that combine ontologies and rules [23,25] in order to cope with OWL's restricted relational expressivity (see Section 3.4). 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.

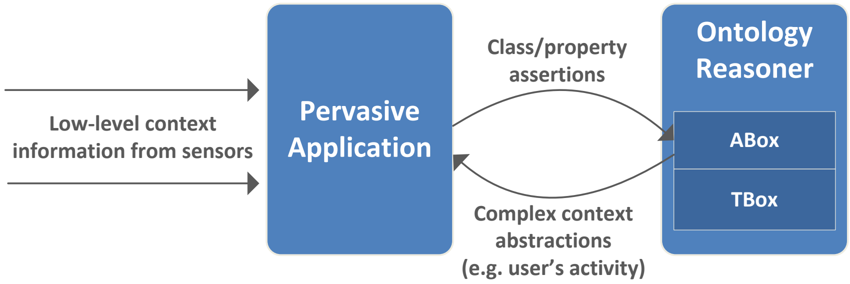


Fig. 4. An abstract ontology-based framework for high-level context reasoning from low-level sensory data

#### 4.6.1. *Ontology-based frameworks*

Characteristic examples of plain ontology-based approaches to reasoning about context are the Context Broker Architecture (CoBrA) [49] and the Semantic Smart Home (SSH) [50] frameworks.

CoBrA is a broker-centric, agent-based architecture supporting context-aware computing in intelligent smart spaces. In CoBrA, through a set of ontologies, useful inferences can be drawn regarding a person's location and activity context. For instance, the detection of Alice's mobile phone in a specific location entails Alice's presence [46]. Jena is used to manage and reason over the knowledge base, while reasoning is invoked through RDQL that periodically queries the knowledge base for the presence of certain context data; e.g., whether a device has been detected in a specific room, or who is the owner of the device. Once queries return matched results, new assertions about the local context are added to the knowledge base. In an alternative implementation, an object-oriented rule-based reasoning engine has been developed using Flora-2 [44]. CoBrA also includes a Prolog-based meta-rule reasoning component to enforce privacy policies tailored to user needs and preferences based on respective sets of user-defined rules.

Semantic Smart Home (SSH) is considered to be an extension of the current smart home solutions with a semantic dimension to enable semantic-based knowledge discovery and intelligent processing. The essence of a SSH is that data, devices and services are given well-defined meaning so that they can be linked within and across smart homes. The ontologies defined within the SSH framework capture knowledge related to physical equipment (such as sensors and electrical appliances), actions and activities of daily living (such as watching television and preparing a meal), location spaces (such as kitchen and living room), people and their roles, medical information, software components as well as temporal information [51]. Modelling activities of daily living (ADLs) through property restrictions in relation to equipment, location and other types of constraints, allows the inference of ADLs 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)`. Further inferences, of higher-level of detail can be drawn as additional sensor descriptions are made available. Continuing the example, the axioms `MakeDrink` $\sqsubseteq$ `KitchenADL` $\sqcap$ $\exists$`hasContainer.Container` and `Cup` $\sqsubseteq$ `Container`, enable to further classify $a$ as an instance of the `MakeDrink` class, once the contact sensor attached to a cup is activated.

In [211], an ontology-based framework for context-aware applications in the smart home domain is presented using as case study a door lock scenario. The concrete situations (i.e. the contextual information directly acquired by sensors) correspond to OWL individuals and realisation is used to determine into which context concepts a specific situation individual falls (e.g., *AuthorisedPersonRinging*). Evaluations with three DL reasoners (Racer, RacerPro and Pellet) demonstrate the feasibility of the proposed framework; yet, the examined scenario is rather simple to safely extend conclusions regarding its evaluation in much broader context-aware applications. Towards a more effective engineering the domain knowledge, a generic methodology for situation modelling has been proposed in a subsequent work [185]. The methodology is based on the systematic decomposition of situations according to so called of *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 more at coarser levels of abstraction) when not all pieces of the relevant context information are available.

A more recent example is the OWL 2 based context modelling and reasoning architecture presented in [169]. As in the aforementioned initiatives, ontologies are used not only to represent activities but also relevant knowledge that can drive their recognition, including locations, objects, and so forth. Unlike the previous approaches though, sensors are directly mapped to ontological classes and properties and the sensor data are directly added to the assertional part of the knowledge base. Sensor data are first fed to COSAR [168], where primitive activities through a combination of statistical and ontological reasoning are derived. These simple activities, in combination with further contextual data, once aggregated and processed so as to resolve possible conflicts [2], formulate subsequently the assertional knowledge, over which OWL 2 DL reasoning can be applied in order to recognise more complex activities. A worth noting feature of the proposed framework is that by exploiting 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 1 DL. Despite the restrictions imposed on the use of the property composition constructor that conditions decidability, the authors argue that the use of OWL 2 DL is a satisfying compromise for effectively reasoning, while avoiding the technical and semantic complexities confronted when combining ontologies and rules.

#### 4.6.2. *Tightly-coupled frameworks*

An early example of a context reasoning framework that combines ontologies and rules can be found in the Gaia infrastructure [170], an infrastructure for smart spaces, which are 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 [164]. For reasoning in first-order logic, the XSB [175] reasoning engine is used. Quantification of variables is performed over the specific domain of values, which is finite, ensuring in this way that evaluations of expressions will always terminate.

More contemporary examples include the context-aware systems [230,232] that investigate the use of SWRL rules. In particular, in [230], 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. 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 Semantic Web-enabled framework for self-management in pervasive computing [232]. 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, and quality of service parameters and dynamic context aspects among others; 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 context abstractions of higher-level. The inferences drawn are subsequently fed to software agents that implement BDI reasoning, i.e. reasoning based on the Beliefs, Desires and Intentions (BDIs) agent model [166]. Provided with enriched contextual knowledge that would otherwise remain implicit, BDI agents can make improved decisions and provide better services such as negotiation and consultation.

Despite the use of SWRL rules, reasoning in both frameworks remains decidable as the existing reasoner implementations employ inherently the DL-safety notions, supporting in practice a form of restricted SWRL.

#### 4.6.3. *Loosely-coupled frameworks*

Unlike the previously described frameworks, loosely-coupled frameworks keep minimal the interaction between ontology- and rule-based reasoning. Rules are applied to the consequences derived by means of ontology reasoning, and affect little the knowledge base. Such a framework is the Service-Oriented Context-Aware Middleware(SOCAM) [94], an architecture which enables building and rapid prototyping of context-aware services in pervasive computing environments. 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 bedroom contained in it. Once implicit knowledge is made explicit, first order logic rules are invoked to derive higher-level contexts such as sleeping, cooking and watching television [222]. Both reasoning modules are implemented using Jena. A similar rationale has been adopted in the Semantic Space framework [221]. RDQL 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. The system uses Jena to implement the forward chaining rules and as the inferred contexts are not stored into the knowledge base, it avoids the need to handle possible conflicting conclusions.

Ontologies and rules have been also coupled in the home-based health monitoring and alarm management system proposed by Paganelli and Giuli [149]. 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 based occurs. Ontology-based reasoning is employed to determine the complex context class to which a specific instance belongs (realisation) and to check the consistency of the knowledge base once new conclusions are added. User-defined rules are employed to trigger alarms and select notification policies based on the available biomedical and environmental context data. For example, a rule may assert an alarm activation when the heart rate frequency is less than 40 beats/minute and systolic blood pressure is higher than 160mm/Hg. Similarly, policy rules are used to assert policy-related facts upon detection of alarm activation facts. Jena has been used to manage and reason over the application ontologies and rules.

The context-aware access control framework proposed in [207] 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 property path relationships. For instance, in a meeting situation there might be the need to model that if the owner of a requested resource is located in a certain place and the resource requestor is located in the same place, then the two persons are co-located. Such relationships are expressed in terms of production (if-then) rules whose head and body match classes and properties of the ontology. The output of the rules is fed into the DL knowledge base to determine the value of each attribute given the current context. In this way, the ontology predicates can be used both in the body and in the head of rules and thus, the ontology knowledge may be modified in iterative ontology and rule-based reasoning steps.

Though seemingly effective in practice, it is inappropriate to compute the consequences of the ontology component first and then apply the rules to the consequences [139]. For example, let us assume the following rules:

```
InsideKitchen(x) ← Cooking(x)
InsideKitchen(x) ← HavingDinner(x)
```

Given the assertion $(Cooking \sqcup HavingDinner)$ : $Alice$, we know that Alice is either cooking or having dinner, but we do not know which one is true. However, though either way one of the rules derives that Alice is in the kitchen, we cannot deduce it by applying the rules to the consequences of the ontology reasoning, as neither $\models Cooking(Alice)$ nor $\models HavingDinner(Alice)$ holds. Detecting and preventing such unwanted behaviour may be straightforward for single cases as in the aforementioned example, but becomes impractical when dealing with large and complex knowledge bases confronted in real-world applications.

Susceptibility to incorrect inferences is further aggravated by the close-world semantics usually employed for rules. 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 [169], where 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, is considered. The knowledge base includes the following definitions:

```
(1) Room ⊓ ¬∃hasOccupant ⊑ EmptyRoom
(2) EmptyRoom ≡ Room ⊓ ¬OccupiedRoom
(3) Room(x) ∧ EmptyHome(y) ∧
      isInside(x,y) → EmptyRoom(x)
(4) ¬EmptyRoom(x) → OccupiedRoom(x)
```

In the example, the front door sensor asserts the entrance of one person, yet none of room sensors succeeds to communicate subsequently that a person is present. Due to the open world semantics of OWL, 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.

#### 4.6.4. Analysis

The aforementioned induce two critical observations:

1. The combination of ontologies and rules is a key prerequisite for effectively meeting the expressivity requirements when modelling and reasoning about context in the pervasive domain; however hybrid reasoning schemes that either only allow for a 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.
2. 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 with the capability 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 the need to reason, keeping certain parts of the world open while closing others, emerges. Similar challenges are confronted in the semantic understanding of visual content [65], 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 3.4. Their practical impact within the pervasive domain remains subject to future investigation.

Parallel to this line of research, hybrid architectures that orchestrate formalisms for distinct types of reasoning have emerged. Such investigations begin from the premise that different aspects of knowledge modelling (temporal, spatial, etc.) impose different requirements that a single, unified formalism is unlikely to meet. An example is the Situation Awareness by Inference and Logic (SAIL) architecture proposed in [13], where a theorem prover and a DL reasoner have been used to reason over streams of time stamped radar data and witness reports in order to derive an understanding of the overall situation (e.g. a potentially hostile aircraft approach), and use linear temporal logic to generate appropriate alerts.

Finally, an issue for further investigation with respect to the reasoning frameworks introduced in this section is extensibility. The reasoning frameworks employed within the presented systems mostly provide domain specific solutions that address the requirements of particular domains. More interoperable approaches that rely on ontological reasoning in upper level ontologies provide a level of abstraction that offers the grounds for adaptation within various domains. However, as semantic commitment in this case is restricted, ontological reasoning needs to be further enhanced by extensions with custom rules leading this way to ad-hoc solutions.

### 4.7. Temporal and Stream Reasoning

As discussed so far, Semantic Web technologies provide rich ontology languages and powerful reasoning and querying mechanisms that meet foundational requirements of typical pervasive systems. However, they normally deal with static data. The need to also deal with with temporal and real-time streaming data has been identified within several pervasive computing applications; characteristic examples include among others mobile telecommunications [130], public health risk monitoring applications(discussed in [69]) and traffic monitoring [98].

Before we perform any temporal or streaming reasoning, we will first explore solutions on how to represent temporal data. 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. [128] introduce Multidimensional RDF, an extension to RDF 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. [95] 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. This foundational work in modelling dynamic data using RDF, influences the design of the temporal component of our sensor data model.

Based on the notion of temporally-extended RDF, Pugliese et al. [163] introduce the tGRIN index structure that builds a specialised index for storing temporal RDF in a relational database based on temporal as well as the structural closeness of triples, while Tappolet et al. [201] present a method for building a meta-index for the validity of named graphs based on Elmasari et al.'s Time Index [80].

Moving beyond the temporal extension, the Semantic Web community have also investigated hybrid frameworks that combine ontologies with temporally-aware formalisms. In the ambient computing framework proposed by Patkos et al. [155] for example, the inherent temporal reasoning capabilities of the Event Calculus [121] are utilised. 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 capture and reason over preconditions and effects of actions and events, based on the Event Calculus theory. A main advantage, in comparison to plain rule-based approaches, is the inherent notion of time in the 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. The situation awareness architecture presented in [13], is another example where different formalisms and available reasoners have been combined to enable efficient inference over 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 4.4, coupling the inherent real-time reasoning capabilities of CEP engines with rich ontological semantics, are yet another example of investigations towards the efficient handling of temporal semantics. Further investigations towards real-time reasoning on data that are changing over time are the stream reasoning frameworks presented in section 4.7.

A recent approach that captures reasoning with non-static data is that of Stream Reasoning [67], an attempt to combine data stream and reasoning technologies towards a solution to real time reasoning about rapidly changing information. Stream Reasoning is defined in [197] 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'*. It is a relatively recent research area; in [69] 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 parallelization and distribution of various tasks to different units.

Theoretical investigations have led to a number of proposals towards stream reasoning languages. In [134], *Construction Descriprtion Logic* ($c\mathcal{ALC}$) is introduced to serve as a semantic type system and knowledge representation formalism for data streams. $c\mathcal{ALC}$ is based on DLs but its semantics are refined to a constructive notion of truth which captures the uncertainty aspects inherent with data streams. In [98], DyKnow, a stream-based knowledge processing middleware is introduced which supports incremental reasoning with streams using Metrical Temporal Logic [148] as the underlying logical language. In [83], the plan of LarKC is introduced, a platform for Web-scale reasoning; further implementation attempts towards LarKC have led to the definition of its underlying language *L2* [1], a lightweight language based on the RDFS vocabulary and a limited subset of OWL. Significant progress in stream reasoning research has been made in querying languages, as discussed in Section 4.5.

#### 4.7.1. Analysis

Despite the aforementioned efforts there is still a gap between the research on advanced reasoning techniques and reasoning on streaming data [69]. Some first steps have been made towards the needs identified in [68] for innovation in foundational theories specific to Stream Reasoning, as well as Stream Reasoning Engineering, but further developments are yet to be addressed [18].

### 4.8. Uncertainty

Because components of a pervasive computing environment deal with the real world, they come with certain caveats: sensors in the field are inherently inaccurate, since they could break down; or they could report inaccurately because they come up against an unusual phenomenon; i.e. one for which they have not been designed.

Since these issues must be taken into account 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 its 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.

Gaia, as a representative of the early works, tries to capture and make sense of the imprecise and conflicting data uncertainty inherent in dealing with real-world data [164]. An uncertainty model is developed based on a predicate representation of contexts and associated confidence values. The predicates' structure and semantics are specified in ontologies that benefit in checking the predicates' validity; simplifying the definition of context predicates in rules; facilitating interoperation between different systems; and further reducing the possibility of uncertainty when interpreting context information. To reason about uncertainty, Gaia employs mechanisms such as probabilistic logic, fuzzy logic, and Bayesian networks, each of which is advantageous under different circumstances. For instance, Gaia uses Bayesian networks to identify causal dependencies (represented as edges) between different events (represented as nodes). 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 to exchange knowledge base specified in a probabilistic model. Responding to the need of modelling imperfect knowledge in the Semantic Web, much research has been devoted to extending formalism and reasoning services so as to handle uncertain and/or vague information. Representative examples including among others fuzzy extensions of DLs [191,190], OWL [28,189] and SWRL [226], and probabilistic extensions such as PR-OWL [62,43] and BayesOWL [70]; for an extensive overview the reader is referred to [192]. Further relevant proposals include the pattern-based approach for representing and reasoning with fuzzy knowledge [212], and the generic, formalised approach for managing uncertainty proposed by Dividino et al. [71]. Few works, however, have explored the applicability of such initiatives in the domain of pervasive applications; an example is the approach presented in [55], where fuzzy reasoning is used to provide personalised mobile services based on situation awareness.

Missing data is another source of uncertainty when reasoning about context: a miss (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 [180,156]. Romero et al. [88,89] 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 whatsoever about the computational framework used to implement the abductive reasoning and the preference criteria used for selecting explanations. Acknowledged as a mode of reasoning that is inherent in a plethora of tasks, much research has been devoted to understanding abduction. For a detailed account on the potential of abductive reasoning in DLs, the interested reader may refer to [75,118].

A formal model based on defeasible logic is proposed by Bikakis et al. to support reasoning with imperfect context in ambient computing environments [24]. 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 an argumentation framework that exploits context and 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.

#### *4.8.1. Analysis*

At present most ontology-based models in 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 also worth of further investigation in that it can help to detect errors (e.g., missing or inaccurate data) which can be used as a feedback to re-tune the system.

### *4.9. Semantic Service Discovery*

One of the core problems in pervasive computing is the question of how to support the introduction of new devices into the environment. As these devices may be unaware of the configuration of the environment or the services available they must undergo a discovery or matchmaking process to best integrate themselves. According to [124], discovery is a process of retrieving services that are able to fulfil a task, and matchmaking is a process of automatically matching semantically annotated services with a semantically annotated request.

Gaia uses DAML+OIL to achieve semantic discovery as it supports some of the operations required for semantic discovery like classification and subsumption. It also allows the definition of relations between concepts. The use of ontologies and semantic discovery replaces scripts and ad hoc configuration files that were used in Gaia previously. Each entity is associated with a DAML+OIL description that describes its properties. The ontology server poses logical queries involving subsumption and classification of concepts to find appropriate matches. Other entities in the environment may query the ontology server to discover classes of components that meet their requirements. Matchmaking uses ontologies to determine a set of concepts that fulfil the intersection of the requirements of two or more parties, such as a supplier and a consumer using the matching algorithms described in [209]. The result is a set of classes that are semantically compatible to the query class.

Christopoulou et al. [54] uses their GAS ontology for discovery in the event of a component failure. All components have ontologies describing their interfaces and available services. If a connection (or synapse) between two or more components is broken, the ontology manager attempts to find an alternative component that offers the same service (i.e., that has the same ontological description) to repair the connection.

The DAML-based Web Service ontology (DAML-S) supplies web service providers with a core set of markup language constructs for describing the properties and capabilities of Web Services [39,40]. It is used in combination with additional ontologies, which organise the concepts appearing in the DAML-S descriptions. The use of DAML-S renders the semantics of the descriptions machine comprehensible; therefore it enables intelligent agents to discover, invoke and compose web services automatically. The Context Ontology Language (CoOL) [196] extends DAML-S with the service context to offer a more formal description of a service's contextual interoperability. The service context consists of two parts: the context obligation, which specifies the obligations of a service in terms of the context of its execution; and the context binding, which is used to establish a virtual link from an atomic process of a service to a specific aspect of the context [196]. This formal semantic service description offers a common understanding of the relations between services and their associated contexts [193]. This facilitates context-awareness and contextual interoperability during service discovery and execution.

[21] introduces a dedicated Service Discovery Protocol (SDP) that enables advertising and discovering services in pervasive environments according to the semantics of networked services and of sought functionalities. The approach is based on the Amigo-S language that extends OWL-S [131] (the successor of DAML-S based on OWL) with new classes and properties. Amigo-S enriches the profile-based annotation paradigm of OWL-S for describing inputs, outputs, preconditions and effects (IOPEs) by incorporating also a number of capabilities for each service. The capabilities are considered either as *provided*, i.e. capabilities that are supported by the service, or *required*, i.e. capabilities that are needed by the service. [127] defines five possible 'degrees of match': *exact*, *plug-in*, *subsume*, *intersection* and *disjoint*. These groups indicate 'how good' a match is.

The OWL-S based approaches cannot support finer-grained service matching where priorities or weights on individual requirements of a service request need to be taken into account. Also they lack an appropriate criterion to approximate and rank the available service with respect to a given request [16]. To address these issues, [16] extends the request description with a priority value to indicate the relevant importance of the individual requirements in a request; e.g., $Req \sqsubseteq (= 1 hasDescription.RD) \sqcap (= 1 hasPriority.PriorityValue)$. The matchmaking process starts with checking if all mandatory requirements are satisfied by a service. If all of them are met, then the service will be evaluated through approximate matching, where the similarity between this service and the request will be determined depending on the semantic deviation of the expected value in request and the available values in the service description for the same requirement. A similarity score will be assigned, which will be used to rank the candidate services later.

*4.9.1. Analysis*

Semantic service discovery is a critical function in pervasive computing that enables ad-hoc associations between service providers and consumers to be established [135]. Gaia, CoOL, and SDP have demonstrated

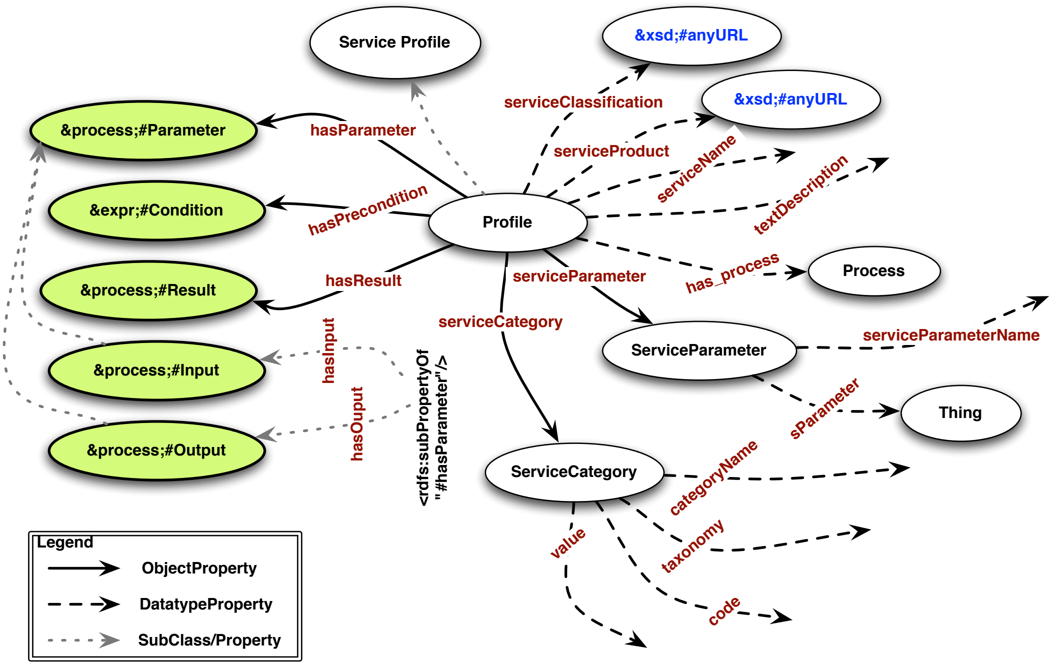


Fig. 5. Selected classes and properties of the profile in OWL-S [131]

that semantic service discovery is better than traditional, rigid, syntactic service matching. OWL-S, an excerpt of whose classes is shown in Figure 5, is currently the most popular of the available technologies for semantic service description. As aforementioned, though it suffers certain limitations when it comes to fine-grained service matching, where priorities on individual service attributes need to be addressed. Other representative semantic service description languages include *WSML*, *WSDL-S/SAWSDL*, *Monolithic DL-based*, and *DSD* for non-logic-based service IOPE profile [119]. Each of these languages has its own strengths and particular application areas, and the current challenge is how to enhance the expressiveness of service annotations so as to increase the capabilities for searching for candidate services.

Approximate matching is another interesting topic, as dealing with vague or incomplete descriptions about the functionality of services or user preferences is not uncommon. Such considerations are not exclusive to the pervasive domain, and have attracted significant body of research in the semantic matchmaking community [119,84].

### *4.10. Privacy and Provenance*

Privacy and provenance are of paramount concern, if pervasive computing applications are to become popular outside the laboratory; yet, only a limited number of the reviewed pervasive frameworks have explicitly considered these aspects.

One such example is the policy-based approach implemented in CoBrA, where the SOUPA policy ontology [47] is used to enforce user privacy. Using this ontology, users can define customised policy rules to permit or forbid different computing entities to access their private information. The policy reasoning algorithm uses a DL inference engine and the DL constructs of OWL to decide whether an action for accessing some user private information is permitted. Since it is often infeasible to define explicit policy rules for every individual action in a domain application, CoBrA uses meta-policies that determine behaviour when policy rules are not defined. These meta-policies can be either conservative (i.e., the system assumes that all actions are forbidden), or liberal (i.e., it assumes all actions are permitted). A CoBra prototype, implemented by Chen et al. [45] for an intelligent meeting room application, was used to demonstrate that the SOUPA policy ontology, and its associated algorithms enable to develop intelligent agents that can provide user privacy protection in a pervasive context-aware environment.

A policy ontology is also proposed by Paganelli and Giuli within their context-aware framework for at home monitoring of patients with chronic diseases [149]. This ontology however addresses the management of alarm notifications, specifying among others priority information about which care person should be contacted first, and does not cover privacy aspects.

#### *4.10.1. Analysis*

As we wish to communicate increasing amounts of (personal) information between services, the topics of security, privacy, and access control become more and more critical. To harvest the advantages that Semantic Web technologies bring in communicating and sharing knowledge across systems, a consensual policy ontology is highly desirable in order to capture in a standardised manner aspects pertinent to privacy and accessibility, such as *who* is granted access to *which parts* of data and for *what purpose* (e.g., reading or editing). The need towards such standardised methods for expressing and exchanging privacy policies, has been widely acknowledge within the Semantic Web community itself and is closely related with the quest and recent effort for formalising the semantics of provenance information (see Section 5.3).

### *4.11. Summary*

This section explicated the use of Semantic Web technologies in the domain of pervasive, sensor-driven applications. The discussion has been structured along the six key requirements, as identified in section 2.3, namely: i) conceptual modelling (ranging from raw sensor data to higher-level context and event abstractions), ii) querying, iii) reasoning, iv) uncertainty management, v) service discovery and vi) privacy and provenance. Besides outlining strengths and weaknesses of the proposed approaches, relevant results within the Semantic Web community that have not been yet explored have been discussed, sketching possible directions for further investigations.

Table 2 summarises the main observations. The use of Semantic Web 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 Semantic Web research too.

Scalability and performance (mentioned as a challenge in Section 2.3) is crucial, as the need to draw complex inferences from millions or billions of pieces of information in real-time, is not restricted in the domain of pervasive systems alone. With the concern in mind, d'Aquin et al. [64] have conducted 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 not only contain the size of data and the response time, but also the factors of devices 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 have shown that these tools are able to cope reasonably well with small-scale ontologies on such a small device; for example, Jena would need up to 50 seconds to perform reasoning on an ontology of less than 4000 triples. The result is promising for a small-scale ontology, however, when faced with millions or billions of triples as mentioned above, we consider the future work should lie with designing a fully distributed approach of coordinating and integrating the reasoning capabilities on small devices especially sensors and mobile devices to enhance the performance.

The seamless integration of open- and closed-world reasoning is another such 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 Semantic Web based pervasive frameworks it was evident that relevant insights often permeated only partially, and in a fragmented manner, the borders between the two research communities.

## 5. Challenging Issues

In the above sections, we have discussed key requirements in pervasive computing and have analysed the benefits and potentials that are induced by using Semantic Web technologies, along with future research inquiries and directions. This section analyses open research issues relevant to *temporal features*, *dynamism*, *provenance*, and *programming* that need to be also investigated towards the seamless integration of Semantic Web technologies and pervasive computing.

Table 2
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 expressivity | facilitate knowledge sharing, reuse and exchange; rich expressiveness for capturing complex semantic relations | upper ontologies; comparative assessment of existing ontologies |
| Querying | SPARQL and its various 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 |
| Reasoning | DLs reasoning services; rule-based reasoning; reasoning with ontologies and rules | inferring implicit knowledge, consistency checking, reasoning about incomplete knowledge | scalability; combining open- and closed-world reasoning; reasoning under inconsistency; temporal/real-time reasoning; hybrid reasoning frameworks |
| Uncertainty handling | extensions for fuzzy/probabilistic semantics (e.g., Fuzzy DLs, PR-OWL); non-standard inferencing (e.g. abduction in DLs ABoxes) | reasoning over imprecise/vague knowledge; reasoning to hypotheses | scalability; seamless integration of uncertainty; unified handling of different types of uncertainty |
| Service discovery | standardised (DAML-S, OWL-S) and custom service-oriented ontologies | semantic discovery and matchmaking of services | scalability; adaptive semantic service composition and execution (e.g. due to service unavailability) |
| Privacy and provenance | customised policy ontologies | facilitate sharing and exchanging of privacy policies | standardised policy/provenance ontologies |

### 5.1. *Temporal Features*

The role of temporal information in the modelling of sensed data is often simplified in the design of data models, systems software, and application APIs. Typically, sensor values are recorded in conjunction with a *timestamp*. 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 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 is often unimportant, however the ability to separately annotate data with *temporal extent*, i.e., a time interval, caters explicitly for the latter case.

Further to this, the use of both an update timestamp 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 offline, or representing a person's predicted future locations based on calendar data. 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 thusly.

Despite strong vocabulary support for representing temporal features, as discussed in section 4.2, 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 use a non-standard extension to the RDF model.

To utilise RDF's reification vocabulary, each triple is expanded as a 'reification quad'[6], with the resultant 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 re-

[6]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.

quiring non-standard tooling to generate and resolve such identifiers.

Section 4.7 has discussed existing temporal extensions to Semantic Web technologies and as well as reasoning mechanisms on such temporal and streaming data. However there is a significant gap between the currently available Semantic Web technologies and the need for native temporal data modelling and reasoning. Relevant suggestions exist in research prototype form, but considerable effort is required before realising standardised solutions that will in turn ensure necessary tool support for efficient reasoning over temporal data and management of temporal queries. Also how to correlate sensor data collected from various sources in terms of their temporal relations to support queries is worthy further investigation. Within this quest, hybrid frameworks present appealing features, as through the combination of different formalisms and reasoners, extended representational and reasoning capabilities can be achieved; the seamless and scalable integration of the heterogeneous modules inevitably poses its own challenges.

### *5.2. Dynamism*

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.

Whether a centralised or decentralised setup is adopted, as the volume 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 approached. However, it does not necessarily follow that the oldest data stored is the least useful. One possible solution might be to require that the data model captures properties describing data dynamics such that policies may be defined to prioritise the discard 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) [187].

Data dynamics also play a role in determining the types of reasoning strategies that should be adopted. There is a need to track derivations such that it may be determined when an inference no longer holds due to modifications to or invalidation of the data upon which it is predicated. Structuring the data model so that data is temporally qualified (and hence inferences are also temporally qualified) forms part of a solution to this issue, but requires the incorporation of the appropriate temporal semantics within 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, choosing to reason on static data as it is generated, adding the inferred knowledge directly to the model, while performing reasoning on highly dynamic data only at the point where an application query is executed, restricting reasoning to the smallest amount of volatile data that will produce a correct answer. Based solely on data's temporal properties, it may be possible to devise a general scheme to partition, reason on, and integrate data over several stages so as to optimise reasoning.

### *5.3. Provenance*

Many of the issues above can be subsumed under the notion of *provenance*, i.e. the capture, representation and manipulation of knowledge relevant to data creation, ownership, transformation and other 'lifecycle' issues. Provenance appears in many guises. The Dublin Core vocabulary [66] is perhaps the best-known, 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 [136] 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, with the degree of each source of error perhaps varying significantly over time. An immediate consequence of this is that sensor-driven systems cannot be directly connected to their input data streams, or respond directly to them: to do so is to allow the system to respond to transient noise. Another way of putting this is that 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.

While sensor fusion is commonplace in much of engineering, many of the approaches make strong assumptions about the nature of the data streams: for example they are from homogeneous sensors looking at the same phenomenon and with well-known sampling frequencies. Such techniques do not generalise well to semantically-enriched systems with highly heterogeneous data sources processed using symbolic reasoning. To note that this issue applies to many other aspects such as querying and reasoning, not only provenance. Conversely, homogeneous data can be handled easily within reasoners, allowing semantic technology to encompass many sensor fusion tasks.

Reasoning over sensor data is affected by a number of provenance factors of data streams and individual data. A trivial example is that the reliability of a temperature estimated from data that is several hours old may be assumed, all other things being equal, 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 and the like, which can be captured generically or using special-purpose markup languages like SensorML (see section 4.1). The provenance here is *associated* with the sensor but *attaches* to the data (stream or items) 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 again we are receiving a stream of temperature values which we intend to use in some decision process. It may matter whether those values are 'raw' (and subject to raw sensor noise), and have been processed to remove obvious outliers, or have been smoothed using an *a priori* or learned smoothing function. It is therefore vital for systems where data comes with provenance that this provenance is manipulated and maintained along the data pathway. For many scientific 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 – otherwise any further statistical calculations are rendered suspect.

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, however, of a standard model for capturing, interchanging and reasoning provenance metadata is a significant impediment to realising applications where the trustworthiness and the quality of the statements is at issue. The Provenance Working Group [161] is an ongoing effort towards the standardisation of an interchange core language (PROV Ontology [159]) for publishing and accessing provenance metadata, drawing on existing vocabularies and ontologies [136,160,162]. In parallel, recent research studies have proposed approaches that enable the handling of provenance while using and reasoning about information and resources in open and collaborative environments [31,147,206].

### 5.4. *Programming*

Lastly we come to the practicalities of programming.

Semantic structures like RDF allow rich encodings of data. Coupled with OWL, SPARQL, ontological and other reasoners, it is possible to answer complex queries by traversing the knowledge graph. The emergence of standard, re-usable reasoners significantly simplifies the use of semantic technologies within programs.

The integration of these tools remains superficial, however. From a programming perspective, a knowledge graph is simply a collection of edges labelled with strings and URIs, possibly with some additional typing at the endpoints supplied by XML Schemata, and with some structure on the relationships provided by the accompanying ontologies (if any). The generality of these structures forces programmers to provide any additional machinery by hand, without the assistance of a compiler, type system of other tooling with which we are familiar in the building of complex software systems. In an effort to simplify the integration of Semantic Web technologies into existing software architectures and languages, a plethora of Application Programming Interfaces (APIs) for working with RDF/OWL ontologies have been developed, such as Jena API [113], OWLAPI [104], dotNetRDF [74], RDFReactor [217] and Sesame [179]. As discussed in Section 4.1.2, Janowicz et al. [111] design and develop a Linked Data model and a RESTful proxy to enable publishing sensor data on the web, which presents as a promising approach.

Use of semantic technology implies considerable familiarity with the tools of XML, notably namespaces, that obscure rather than illuminate the underlying information being encoded. Data so encoded is not held in this form in memory, and so must be translated for storage and exchange: a non-trivial process in the presence of shared sub-structures and update. State-of-the-art ontology editors, such as Protégé [144], TopBraid Composer [208], and NeOn Toolkit [142] offer comprehensive support for developing and validating RDF/OWL ontologies. Moreover, scalable and query efficient RDF repositories, such as OWLIM [26], AllegroGraph [5], and OpenLink Virtuoso [81], and as well as, database-to-RDF mapping tools such as D2RQ [100,79], enable data to be exposed and shared on the Web according to the principles of Linked Data [27].

Entities represented within a knowledge graph may be classified by the ontology according to the information present about them – classifying a person as an employee given the presence of an employee number and/or a line manager within the same organisation, for example. This is superficially the same as a programming language type system, with the difference that an entity's class may change unpredictably and at unpredictable times through the actions of other agents on the knowledge graph, and it can be hard to predict exactly *which* such changes will have such an effect. This destabilises a program's view of the knowledge in the graph.

Use of a reasoner sits outside the programming language, in the same way as does SQL 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 from a lack of supporting tooling. Some of the aforementioned ontology editors provide only a limited support for advanced query formulation, e.g., type-checking.

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

As with any technology, the decision to use Semantic Web 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 subdomain, to describe the richness of this structure, and to have it compose cleanly with other such descriptions of complementary sub-domains defined independently – and to be able to exchange all this knowledge with anyone on the web. But this flexibility comes with a cost and (often) no obvious 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 exactly opposite issues, focusing on close specification of algorithms and data structures to share common data and functionality.

Sensor-driven systems add few, if any, specific points to this general discussion: the issues apply to *all* programming with semantic structures. Sensor-driven systems add the necessity to embrace the noise and errors inherent in data streams, and to capture and make decisions that propagate this uncertainty throughout the system. These are not issues that mainstream programming languages provide structures for, although it is possible to build programs that (for example) maintain the error bars on results derived from processing inputs that themselves have error bars. Combining programming with uncertainty and programming with data that adhere to an ontological structure, seems to be two new frontiers for programming systems design opened-up 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 amount 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 Semantic Web 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 interoperate easily with the wider universe of web-enabled components.

## 6. Discussion and Concluding Remarks

In this paper, we reviewed the landscape of the applications of Semantic Web technologies to the domain of pervasive, adaptive, sensor-driven systems. Many of the features underpinning the Semantic Web, and in particular, the ability to formally capture and reason over rich, semantic interconnections between data, in an open and extensible way, are well-suited to pervasive computing. At the same time however, a number of other important aspects, such as lack of native support for representing and reasoning over temporal or imperfect knowledge, remain under-articulated.

The key benefits of Semantic Web technologies are rather straightforward in terms of modelling and reasoning about context. As afore-described, ontologies have been proposed for describing the four *Ws (when, where, what, who)* that characterise contextual knowledge, complex events and sensor data, facilitating 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 [61] have been resolved to different degrees. What still remains unaddressed, acknowledging the different world-views that different ontologies inevitably serve, 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, underline further 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, possibilities for their combined use and/or (partial) alignment, as well as aspects that require a more elaborate coverage.

In parallel, the deployment of Semantic Web 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. Prominent ones include the need to provide native support for representing and reasoning over temporal knowledge, incorporating provenance into reasoning, and managing uncertainty. An indispensable requirement, underlying all of aforementioned research directions, is computational efficiency. Ensuring reasoning efficiency is crucial, as the need to draw inferences in real-time, from millions or billions of pieces of information, in an open pervasive environment, already challenges existing reasoning engines, even though considering only deductive inference over crisp, consistent knowledge bases.

Subject to further enquiry is also the overall integration of Semantic Web 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 memory, computation and communications capabilities. It is important to remember, however, that the Semantic Web is essentially a tool of modelling and exchange, and only secondarily a tool of implementation: compact representations of information and reasoning that comply to the underlying metamodel of the Semantic Web 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 encourage the view that basing future pervasive and sensor-driven systems around these technologies would lead to significant improvements in interoperability and semantic clarity – *if* 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 Weiser's vision of seamless, integrated pervasive and sensor systems significantly closer.

## References

[1] *Towards a scalable, pragmatic Knowledge Representation Language for the Web*, 04 2009.

[2] A. Agostini, C. Bettini, and D. Riboni. Hybrid reasoning in the care middleware for context awareness. *Int. J. Web Eng. Technol.*, 5(1):3–23, May 2009.

[3] Y. L. Alejandro Rodríguez, Robert McGrath and J. Myers. Semantic management of streaming data. In *Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09)*, volume 522 of *CEUR Workshop Proceedings*, pages 80–95, 2009.

[4] F. Alkhateeb, J.-F. Baget, and J. Euzenat. Extending sparql with regular expression patterns (for querying rdf). *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(2), 2011.

[5] AllegroGraph RDFStore Web 3.0's Database. `http://www.franz.com/agraph/allegrograph/`.

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

[7] 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.
[8] 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, New York, NY, USA, 2011. ACM.
[9] 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.
[10] G. Antoniou and F. v. Harmelen. *A Semantic Web Primer, 2nd Edition (Cooperative Information Systems)*. The MIT Press, 2 edition, 2008.
[11] A. Arasu, S. Babu, and J. Widom. The CQL continuous query language: semantic foundations and query execution. *The VLDB Journal*, 15(2):121–142, June 2006.
[12] M. Arenas, C. Gutierrez, and J. Pérez. An extension of SPARQL for RDFS. In *Proceedings of the joint ODBIS & SWDB workshop on Semantic Web, Ontologies, Databases*, pages 1–20, 2007.
[13] 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.
[14] 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.
[15] F. Baader and U. Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69(1):5–40, 2001.
[16] A. Bandara, T. Payne, D. D. Roure, N. Gibbins, and T. Lewis. Semantic resource matching for pervasive environments: The approach and its evaluation. Technical report, Faculty of Physical and Applied Science, University of Southampton, 2008.
[17] 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.
[18] 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.
[19] D. F. Barbieri, D. Braga, S. Ceri, E. D. Valle, and M. Grossniklaus. Querying rdf streams with c-sparql. *SIGMOD Rec.*, 39(1):20–26, Sept. 2010.
[20] N. Baumgartner and W. Retschitzegger. A survey of upper ontologies for situation awareness. In *In Proceedings of the 4th IASTED International Conference on Knowledge Sharing and Collaborative Engineering*, pages 1–9, St. Thomas, US Virgin Islands, November 2006.
[21] S. Ben Mokhtar, A. Kaul, N. Georgantas, and V. Issarny. Efficient semantic service discovery in pervasive computing environments. In *Proceedings of the ACM/IFIP/USENIX 2006 International Conference on Middleware*, Middleware '06, pages 240–259, Melbourne, Australia, 2006. Springer-Verlag New York, Inc.
[22] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.
[23] 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.
[24] 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.
[25] A. Bikakis, T. Patkos, G. Antoniou, and D. Plexousakis. A Survey of Semantics-based Approaches for Context Reasoning in Ambient Intelligence. In *Proceedings of the Workshop Artificial Intelligence Methods for Ambient Intelligence*, pages 15–24, 2007.
[26] 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.
[27] 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.
[28] F. Bobillo and U. Straccia. Fuzzy ontology representation using OWL 2. *Int. J. Approx. Reasoning*, 52(7):1073–1094, 2011.
[29] J. Bohn, F. C. Gartner, and H. Vogt. Dependability Issues of Pervasive Computing in a Healthcare Environment. In *first International Conference on Security in Pervasive Computing*, Boppard, Germany, 2003.
[30] 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.
[31] P. A. Bonatti, A. Hogan, A. Polleres, and L. Sauro. Robust and scalable linked data reasoning incorporating provenance and trust annotations. *J. Web Sem.*, 9(2):165–201, 2011.
[32] 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.
[33] G. Borriello, V. Stanford, C. Narayanaswami, and W. Menning. Guest editors' introduction: Pervasive computing in healthcare. *IEEE Pervasive Computing*, 6(1):17–19, Jan. 2007.
[34] B. Bouchard and A. Bouzouane. A key hole plan recognition model for alzheimer's patients: First results. *Applied Artificial Intelligence*, 22:1–34, 2007.
[35] E. Bouillet, M. Feblowitz, Z. Liu, A. Ranganathan, A. Riabov, and F. Ye. A semantics-based middleware for utilizing heterogeneous sensor networks. In *Proceedings of the 3rd IEEE international conference on Distributed computing in sensor systems*, DCOSS'07, pages 174–188, Berlin, Heidelberg, 2007. Springer-Verlag.
[36] C. Y. Brenninkmeijer, I. Galpin, A. A. Fernandes, and N. W. Paton. A semantics for a query language over sensors, streams and relations. In *Proceedings of the 25th British national conference on Databases: Sharing Data, Information and Knowledge*, BNCOD '08, pages 87–99, Berlin, Heidelberg, 2008. Springer-Verlag.

[37] 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.

[38] 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.

[39] M. Burstein, J. Hobbs, O. Lassila, D. Martin, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, and H. Zeng. DAML-S Draft Release (May 2001). http://www.daml.org/services/daml-s/2001/05/, 2001.

[40] M. H. Burstein, J. R. Hobbs, O. Lassila, D. Martin, D. V. McDermott, S. A. McIlraith, S. Narayanan, M. Paolucci, T. R. Payne, and K. P. Sycara. DAML-S: Web Service Description for the Semantic Web. In *Proceedings of the First International Semantic Web Conference on The Semantic Web*, ISWC '02, pages 348–363, London, UK, 2002. Springer-Verlag.

[41] J.-P. Calbimonte, O. Corcho, and A. J. G. Gray. Enabling ontology-based access to streaming data sources. In *Proceedings of the 9th international semantic web conference on The semantic web - Volume Part I*, ISWC'10, pages 96–111, Berlin, Heidelberg, 2010. Springer-Verlag.

[42] 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.

[43] 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.

[44] H. Chen, T. Finin, and A. Joshi. An Ontology for Context-Aware Pervasive Computing Environments. *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, 18(3):197–207, May 2003.

[45] H. Chen, T. Finin, and A. Joshi. A Pervasive Computing Ontology for User Privacy Protection in the Context Broker Architecture. Technical Report TR-CS-04-08, University of Maryland, Baltimore County, July 2004.

[46] H. Chen, T. Finin, and A. Joshi. Semantic web in the context broker architecture. In *PERCOM '04: Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04)*, pages 277 – 286, Washington, DC, USA, 2004. IEEE Computer Society.

[47] H. Chen, F. Perich, D. Chakraborty, T. Finin, and A. Joshi. Intelligent agents meet semantic web in a smart meeting room. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '04, pages 854–861, New York, New York, 2004. IEEE Computer Society.

[48] 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.

[49] H. L. Chen. *An Intelligent Broker Architecture for pervasive context-aware systems*. PhD thesis, University of Maryland, 2004.

[50] 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.

[51] L. Chen and C. D. Nugent. Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems*, 5(4):410–430, 2009.

[52] L. Chen, C. D. Nugent, and H. Wang. A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering*, 2011. To appear.

[53] J. Cheng, Z. Ma, and L. Yan. f-sparql: A flexible extension of sparql. In P. Bringas, A. Hameurlain, and G. Quirchmayr, editors, *Database and Expert Systems Applications*, volume 6261 of *Lecture Notes in Computer Science*, pages 487–494. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-15364-8_41.

[54] E. Christopoulou, C. Goumopoulos, I. Zaharakis, and A. Kameas. An ontology-based conceptual model for composing context-aware applications. In *Proceedings of the First International Workshop on Advanced Context Modelling, Reasoning And Management (in UbiComp 2004)*, September 2004.

[55] 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.

[56] A. G. Cohn, B. Bennett, J. Gooday, and N. M. Gotts. Qualitative spatial representation and reasoning with the region connection calculus. *Geoinformatica*, 1(3):275–316, Oct. 1997.

[57] 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.

[58] 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.

[59] D. J. Cook. How smart is your home? *Science*, 335:1579–1581, Mar. 2012.

[60] 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.

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

[62] 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.

[63] R. Cunningham and V. Cahill. System support for smart cars: requirements and research directions. In *Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system*, EW 9, pages 159–164, Kolding, Denmark, 2000. ACM.

[64] M. D'Aquin, A. Nikolov, and E. Motta. How much semantic data on small devices? In *Proceedings of the 17th inter-*

*national conference on Knowledge engineering and management by the masses*, EKAW'10, pages 565–575, Berlin, Heidelberg, 2010. Springer-Verlag.

[65] 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.

[66] Dublin Core Metadata Initiative vocabulary. `http://dublincore.org/documents/dcmi-terms/`.

[67] 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.

[68] 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.

[69] 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.

[70] 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.

[71] 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.

[72] 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.

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

[74] dotNetRDF - Semantic Web/RDF Library for C#/.Net. `http://www.dotnetrdf.org/`.

[75] J. Du, G. Qi, Y.-D. Shen, and J. Z. Pan. Towards practical ABox abduction in large OWL DL ontologies. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011*, pages 1160–1165, San Francisco, California, USA, 2011. AAAI Press.

[76] DOLCE+DnS Ultralite (DUL) ontology. `http://www.loa.istc.cnr.it/ontologies/DUL.owl`.

[77] E. Dumbill. Finding friends with XML and RDF: FOAF, 2002. http://www-106.ibm.com/developerworks/xml/library/x-foaf.html.

[78] M. Eid, R. Liscano, and A. EI Saddik. A universal ontology for sensor networks data. In *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA 2007)*, pages 59–62, 2007.

[79] 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.

[80] 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.

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

[82] K. Farkas, H.-P. Labs, J. Heidemann, and L. Iftode. Intelligent transportation and pervasive computing. *Pervasive Computing*, 5:18–19, 2006.

[83] 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.

[84] G. Fenza, V. Loia, and S. Senatore. A hybrid approach to semantic web services matchmaking. *Int. J. Approx. Reasoning*, 48(3):808–828, 2008.

[85] Davies and Vitiello's relationship vocabulary for FOAF. `http://vocab.org/relationship`.

[86] D. Franklin. Cooperating with people: the intelligent classroom. In *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, AAAI '98/IAAI '98, pages 555–560, Madison, Wisconsin, United States, 1998. American Association for Artificial Intelligence.

[87] 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.

[88] J. Gómez-Romero, M. A. Patricio, J. García, and J. M. Molina. Context-based reasoning using ontologies to adapt visual tracking in surveillance. In *Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, AVSS '09, pages 226–231, Washington, DC, USA, 2009. IEEE Computer Society.

[89] 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.

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

[91] 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.

[92] 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.

[93] 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.

[94] 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.

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

[96] 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.

[97] J. K. Hart and K. Martinez. Environmental sensor networks:a revolution in the earth system science? *Earth-Science Reviews*, 78:177–191, 2006.

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

[99] 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.

[100] M. Hert, G. Reif, and H. C. Gall. A comparison of RDB-to-RDF mapping languages. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 25–32, Graz, Austria, 2011. ACM.

[101] J. Hightower, B. Brumitt, and G. Borriello. The location stack: A layered model for location in ubiquitous computing. *Proceedings of WMCSA 2002*, 00:22–28, 2002.

[102] 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.

[103] F. Hohl, U. Kubach, A. Leonhardi, K. Rothermel, and M. Schwehm. Next century challenges: Nexus—an open global infrastructure for spatial-aware applications. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 249–255, New York, NY, USA, 1999. ACM.

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

[105] I. Horrocks and P. F. Patel-Schneider. KR and reasoning on the semantic web: OWL. In J. Domingue, D. Fensel, and J. A. Hendler, editors, *Handbook of Semantic Web Technologies*, chapter 9, pages 365–398. Springer, 2011.

[106] 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.

[107] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: the making of a Web Ontology Language. *J. Web Sem.*, 1(1):7–26, 2003.

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

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

[110] International organization for standardization, for FDIS 19115 geographic information – metadata. `http://www.ncits.org/ref-docs/FDIS_19115.pdf`.

[111] 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.

[112] K. Janowicz and M. Compton. The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology. In *Proceedings of the 3rd International workshop on Semantic Sensor Networks 2010 (SSN10) in conjunction with the 9th International Semantic Web Conference (ISWC 2010)*, Shanghai, China, Sept. 2010.

[113] Apache Jena. `http://jena.apache.org/`.

[114] 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.

[115] 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.

[116] C. Kiefer, A. Bernstein, and A. Locher. Adding data mining support to sparql via statistical relational learning methods. In *Proceedings of the 5th European semantic web conference on The semantic web: research and applications*, ESWC'08, pages 478–492, Berlin, Heidelberg, 2008. Springer-Verlag.

[117] B. M. Kiernan, S. Beirne, C. Fay, and D. Diamond. Monitoring of gas emissions at landfill sites using autonomous gas sensors. Technical report, Clarity Centre, Ireland, 2010. Project Report. STRIVE, Environmental Protection Agency.

[118] S. Klarman, U. Endriss, and S. Schlobach. ABox Abduction in the Description Logic *ALC*. *J. Autom. Reasoning*, 46(1):43–80, 2011.

[119] M. Klusch. Chapter 4: Semantic web service coordination. In *CASCOM - Intelligent Service Coordination in the Semantic Web*. Birkhaeuser Verlag, Springer, 2008.

[120] K. Kochut and M. Janik. Sparqler: Extended sparql for semantic association discovery. In E. Franconi, M. Kifer, and W. May, editors, *ESWC*, volume 4519 of *Lecture Notes in Computer Science*, pages 145–159. Springer, 2007.

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

[122] A. A. Krisnadhi, K. Sengupta, and P. Hitzler. Local closed world semantics: Keep it simple, stupid! In *Description Logics*, 2011.

[123] K. N. Kumar, R.Prabakaran, V. S. Dhulipala, and P.Ranjith. Future sensors and utilization of sensors in chemical industries with control of environmental hazards. In *Proceedings of the 2nd International Conference on Environmental Science and Development*, pages 224–228. IACSIT Press, 2011.

[124] U. Küster, H. Lausen, and B. König-Ries. Evaluation of semantic service discovery – a survey and directions for future research. In *Post-Proceedings of the 2nd Workshop on Emerging Web Services Technology (WEWST07) in conjunction with the 5th IEEE European Conference on Web Services (ECOWS07)*, 2007.

[125] 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.

[126] 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.

[127] L. Li and I. Horrocks. A software framework for matchmaking based on semantic web technology. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 331–339, Budapest, Hungary, 2003. ACM.

[128] 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.

[129] N. Lopes. Extensions of sparql towards heterogeneous sources and domain annotations. In *9th International Semantic Web Conference (ISWC2010)*, November 2010.

[130] 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.

[131] D. Martin, M. Burstein, and J. Hobbs. OWL-S: Semantic markup for web services. http://www.w3.org/Submission/OWL-S, 2004.

[132] K. Martinez, J. K. Hart, and R. Ong. Environmental sensor networks. *Computer*, 37(8):50–56, Aug. 2004.

[133] 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.

[134] M. Mendler and S. Scheele. Towards a type system for semantic streams. In *Proc. 1st Intï£¡l Workshop Stream Reasoning*, 2009.

[135] S. B. Mokhtar, D. Preuveneers, N. Georgantas, V. Issarny, and Y. Berbers. Easy: Efficient semantic service discovery in pervasive computing environments with qos and context support. *Journal of Systems and Software*, 81(5):785 – 808, 2008.

[136] 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.

[137] 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.

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

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

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

[141] D. Nebert. SDI Cookbook, 2004. Version 2.0.

[142] The NeOn Toolkit. `http://neon-toolkit.org`.

[143] 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.

[144] 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.

[145] 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.

[146] 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.

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

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

[149] 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.

[150] F. Pan and J. R. Hobbs. Time ontology in OWL. W3C working draft, W3C, Sept. 2006. http://www.w3.org/TR/2006/WD-owl-time-20060927/.

[151] J. Z. Pan, G. Stamou, G. Stoilos, S. Taylor, and E. Thomas. Scalable querying services over fuzzy ontologies. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 575–584, New York, NY, USA, 2008. ACM.

[152] J. Paradiso, P. Dutta, H. Gellersen, and E. Schooler. Guest editors' introduction: Smart energy systems. *Pervasive Computing, IEEE*, 10(1):11 –12, Jan-Mar 2011.

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

[154] P. F. Patel-Schneider and I. Horrocks. A comparison of two modelling paradigms in the semantic web. *J. Web Sem.*, 5(4):240–250, 2007.

[155] 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.

[156] 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.

[157] 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.

[158] 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.

[159] PROV-O: The PROV Ontology, W3C Working Draft 03 May 2012. `http://www.w3.org/TR/prov-o/`.

[160] Provenance Vocabulary Core Ontology Specification, 14 March 2012. `http://trdf.sourceforge.net/provenance/ns.html`.

[161] Provenance Working Group. `http://www.w3.org/2011/prov/wiki/Main_Page`.

[162] Provenir Ontology. `http://wiki.knoesis.org/index.php/Provenir_Ontology`.

[163] A. Pugliese, O. Udrea, and V. S. Subrahmanian. Scaling RDF with Time. In *Proceeding of the 17th international confer-*

*ence on World Wide Web (WWW '08)*, pages 605–614, Beijing, China, 2008. ACM.

[164] 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.

[165] 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.

[166] A. S. Rao and M. P. Georgeff. BDI agents: From theory to practice. In *ICMAS*, pages 312–319. The MIT Press, 1995.

[167] C. Reed, F. Collins, M. Botts, J. Davidson, and G. Percivall. OGC sensor web enablement:overview and high level achhitecture. In *Autotestcon '2007*, 2007.

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

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

[170] 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.

[171] 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.

[172] 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.

[173] 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.

[174] D. J. Russomanno, C. R. Kothari, and O. A. Thomas. Building a sensor ontology: A practical approach leveraging ISO and OGC models. In *Proceedings of the 2005 International Conference on Artificial Intelligence*, pages 637–643, Las Vegas, USA, 2005.

[175] 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.

[176] 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.

[177] A. Scherp, T. Franz, C. Saathoff, and S. Staab. A core ontology on events for representing occurrences in the real world. *Multimedia Tools and Applications*, 58(2):293–331, 2012.

[178] A. Seaborne and S. Harris. SPARQL 1.1 query. W3C working draft, W3C, Oct. 2009. http://www.w3.org/TR/2009/WD-sparql11-query-20091022/.

[179] openRDF.org. `http://www.openrdf.org/`.

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

[181] R. Shaw, R. Troncy, and L. Hardman. Lode: Linking open descriptions of events. In *Proceedings of the 4th Asian Conference on The Semantic Web*, ASWC '09, pages 153–167, Shanghai, China, 2009. Springer-Verlag.

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

[183] 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.

[184] S. Song, M. Kim, S. Rho, and E. Hwang. Music ontology for mood and situation reasoning to support music retrieval and recommendation. In *Proceedings of the 2009 Third International Conference on Digital Society*, ICDS '09, pages 304–309, Washington, DC, USA, 2009. IEEE Computer Society.

[185] 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.

[186] 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.

[187] 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.

[188] 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.

[189] G. Stoilos, G. B. Stamou, and J. Z. Pan. Fuzzy extensions of owl: Logical properties and reduction to fuzzy description logics. *Int. J. Approx. Reasoning*, 51(6):656–679, 2010.

[190] 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.

[191] U. Straccia. Towards a fuzzy description logic for the semantic web (preliminary report). In *Proceedings of the Second European conference on The Semantic Web: research and Applications*, ESWC'05, pages 167–181, Heraklion, Greece, 2005. Springer-Verlag.

[192] 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.

[193] T. Strang and C. Linnhoff-Popien. Service interoperability on context level in ubiquitous computing environments. In *Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet (SSGRR2003w)*, L'Aquila/Italy, January 2003.

[194] T. Strang and C. Linnhoff-Popien. A context modeling survey. In *Proceedings of the Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp 2004 - The Sixth International Conference on Ubiqui-*

*tous Computing*, Nottingham/England, September 2004.

[195] 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.

[196] T. Strang, C. Linnhoff-Popien, and K. Frank. CoOL: A Context Ontology Language to enable Contextual Interoperability. In J.-B. Stefani, I. Dameure, and D. Hagimont, editors, *LNCS 2893: Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS2003)*, Lecture Notes in Computer Science (LNCS), pages 236–247, Paris/France, November 2003. Springer Verlag.

[197] 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.

[198] IEEE Suggested Upper Merged Ontology. `http://suo.ieee.org/SUO/SUMO/SUMO_173.kif`.

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

[200] J. Tao, E. Sirin, J. Bao, and D. L. McGuinness. Integrity Constraints in OWL. In *Proceedings of Twenty-Fourth AAAI Conference on Artificial*, 2010.

[201] 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.

[202] 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.

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

[204] H. J. ter Horst. Completeness, decidability and complexity of entailment for rdf schema and a semantic extension involving the owl vocabulary. *Web Semant.*, 3(2-3):79–115, Oct. 2005.

[205] 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.

[206] 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.

[207] 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.

[208] TopBraid Composer. `http://www.topquadrant.com/products/TB_Composer.html`.

[209] D. Trastour, C. Bartolini, and J. Gonzalez-Castillo. A Semantic Web Approach to Service Description for Matchmaking of Services. Technical Report HPL-2001-183, HP Laboratories Bristol, 2001.

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

[211] A.-Y. Turhan, T. Springer, and M. Berger. Pushing doors for modeling contexts with OWL DL a case study. In *PerCom Workshops*, pages 13–17, 2006.

[212] 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.

[213] 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.

[214] F. van Harmelen and D. L. McGuinness. OWL Web Ontology Language Overview. http://www.w3.org/TR/2004/REC-owl-features-20040210/, 2004.

[215] M. Y. Vardi. Why is modal logic so robustly decidable? In *Proceedings of a DIMACS Workshop on Descriptive Complexity and Finite Models*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pages 149–184. American Mathematical Society, Jan. 1996.

[216] vCard Ontoloty. `http://www.w3.org/2006/vcard`.

[217] 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.

[218] W3C PIM. `http://www.w3.org/2000/10/swap/pim/`.

[219] W3C Semantic Sensor Network Incubator group. `http://www.w3.org/2005/Incubator/ssn/`.

[220] Review of sensor and observation ontologies. `http://www.w3.org/2005/Incubator/ssn/wiki/Review_of_Sensor_and_Observations_Ontologies`.

[221] 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.

[222] X. Wang, D. Zhang, T. Gu, and H. K. Pung. Ontology based context modeling and reasoning using owl. In *PerCom Workshops*, pages 18–22, 2004.

[223] 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.

[224] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, Sept. 1991.

[225] K. Whitehouse, F. Zhao, and J. Liu. Semantic streams: a framework for composable semantic interpretation of sensor data. In *Proceedings of the Third European conference on Wireless Sensor Networks*, EWSN'06, pages 5–20, Berlin, Heidelberg, 2006. Springer-Verlag.

[226] 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.

[227] 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.

[228] 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 (PervasiveHealth 2011)*, pages 131–138, Dublin, Ireland, 2011.

[229] Z. Yu and Y. Nakamura. Smart meeting systems: A survey of state-of-the-art and open issues. *ACM Comput. Surv.*, 42(2):8:1–8:20, Mar. 2010.

[230] 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.

[231] J. Yuan, Y. Zheng, X. Xie, and G. Sun. Driving with knowledge from the physical world. In *Proceedings of KDD' 11*, pages 316–325, San Diego, California, USA, 2011.

[232] 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.

[233] 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.