Pervasive and Mobile Computing 25 (2016) 104–124

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

MetaQ: A knowledge-driven framework for context-aware activity recognition combining SPARQL and OWL 2 activity patterns

Georgios Meditskos [a,*], Stamatia Dasiopoulou [b], Ioannis Kompatsiaris [a]

[a] Information Technologies Institute, Centre for Research & Technology - Hellas, P.O. Box 60361, 6th Km Charilaou - Thermi, 57001, Thessaloniki, Greece

[b] Department of Information and Communication Technologies, Pompeu Fabra University, Barcelona

A R T I C L E I N F O

Article history:
Received 19 May 2014
Received in revised form 30 November 2014
Accepted 23 January 2015
Available online 2 February 2015

Keywords:
Ontologies
SPARQL
Patterns
Activity recognition

A B S T R A C T

In this paper we describe MetaQ, an ontology-based hybrid framework for activity recognition in Ambient Assisted Living (AAL) environments that combines SPARQL queries and OWL 2 activity patterns. SPARQL is used as a standardised declarative language for aggregating, interpreting and enriching low-level contextual RDF knowledge bases with higher level derivations. The proposed SPARQL-based reasoning framework supports key inferencing tasks that are important in activity interpretation domains, but not supported by the standard semantics of OWL 2, such as temporal reasoning and dynamic assertion of structured individuals. In order to promote the extensibility and reuse of the underlying interpretation semantics, the reasoning framework is further enhanced with a conceptual layer that allows the formal representation of activity meta-knowledge by means of DOLCE+DnS Ultralite (DUL) ontology patterns. We illustrate the capabilities of the proposed framework through its deployment in a hospital for monitoring activities of Alzheimer's disease patients.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

The demand for *context-aware* user task support has proliferated in the recent years across a multitude of application domains, ranging from healthcare and smart spaces to transportation and energy control [1]. A key challenge in such applications is to abstract and fuse the captured *context* in order to elicit an adequate understanding of the user situation and afford services that are effectively tailored to the user needs [2].

Congruous with the open nature of context-awareness, where information at various levels of abstraction and completeness has to be integrated, ontologies have attracted growing interest as means for modelling and reasoning over contextual information and human activities in particular [3]. The OWL DL [4] and, more recently, OWL 2 [5] ontology languages have been proposed to capture the context elements of interest (e.g. persons, events, activities, locations) and their pertinent relations, and to formalise activity recognition as logical reasoning by mapping the information that is obtained directly through context detectors (e.g. video cameras, contact sensors) to respective class and property assertions.

To harvest the several benefits brought by ontologies (e.g. modelling of complex logical relations, sharing information coming from heterogeneous sources, availability of sound and complete reasoning engines [6,7]), while coping with OWL's

* Corresponding author.
E-mail addresses: gmeditsk@iti.gr (G. Meditskos), stamatia.dasiopoulou@upf.edu (S. Dasiopoulou), ikom@iti.gr (I. Kompatsiaris).

http://dx.doi.org/10.1016/j.pmcj.2015.01.007
1574-1192/© 2015 Elsevier B.V. All rights reserved.

inherent inability to support temporal reasoning, ontology-based frameworks adopt either an a-temporal approach to activity modelling [8–10] or combine ontologies with rules [11–14].

In the first case, the data is segmented into chunks of activities, e.g. based on predefined time windows, allowing complex activities to be derived using standard OWL reasoning for context classification. For example, the tea preparation activity in the kitchen that is inferred on the basis of heating water and using a tea bag and a cup could be modelled in OWL as[1]:

```
MakeHotTea ≡ Activity and (actor only (Person
                 and (uses some TeaBag)
                 and (uses some Cup)
                 and (uses some Kettle)
                 and (near some KitchenBench)))
```

Though relevant to applications where suitable time windows can be reliably defined, a-temporal approaches fall short when intricate activity patterns are involved, requiring, for instance, the discrimination of sequential and interleaved activities, or the modelling of dependencies among the sub-activities that go beyond the property chains expressivity provided by OWL 2. Moreover, such class descriptions can classify only existing instances, meaning that the unknown activity context must be identified and asserted beforehand, e.g. through snapshot-like definitions. For example, the `MakeHotTea` axiom above can classify only existing `Activity` instances that have been asserted as part of the "current" context.

In the second case, the combination of ontologies with rules that manage the temporal information and knowledge updates, affords far more expressive and flexible alternatives than their a-temporal counterparts. The ontology is used to represent activity-related information, whereas rules are used to further aggregate activities, describing the contextual conditions that drive the derivation of complex activities. For example, the following (pseudo) rule defines the assertion of a `MakeHotTea` activity when the person is near the kitchen bench and uses (`UseX` activities) tea-related objects.

```
UseTeaBag(?a1), UseCup(?a2), UseKettle(?a3), NearKitchenBench(?a4),
actor(?a1, ?p), actor(?a2, ?p), actor(?a3, ?p), actor(?a4, ?p),
time(?a1, ?t1), time(?a2, ?t2), time(?a3, ?t3), time(?a4, ?t4),
contains(?t4, ?t1), contains(?t4, ?t2), contains(?t4, ?t3)
→ MakeHotTea(?new), time(?new, ?t4), actor(?new, ?p)
```

However, ontologies in such approaches are essentially used solely as vocabularies for representing and sharing activity-related information. The actual interpretation of activity semantics are defined outside the ontology models, e.g. they are encapsulated in rules that are tightly-coupled with implementation frameworks, hindering interoperability and reuse. Thus, applications that share similar purposes and scope cannot directly avail of existing frameworks. Standardisation efforts for the exchange of rules on the Semantic Web, such as RIF [16], are useful to develop an interchange format between existing rule languages in a syntactic level. These standards, however, fail to capture the semantics in a conceptual level, e.g. through formal and semantically enriched models.

To promote sharing and reuse of activity descriptions, while availing of the temporal reasoning capabilities afforded by the use of rules, we propose MetaQ. MetaQ is an OWL 2 ontology-based framework for context-aware activity recognition that brings together the rigorous semantics of the DnS design pattern [17] of DOLCE + DnS Ultralite (DUL) [18] and of SPARQL [19], the standard query language for the Semantic Web. Extending the DnS design pattern and making use of the OWL 2 meta-modelling capabilities (*punning* [20]), MetaQ defines an activity pattern ontology that serves as a meta-model for the formal description of activities. The activity descriptions are in turn translated into SPARQL queries. SPARQL is an expressive standardised language, whose semantics and complexity have been studied fairly extensively, showing that SPARQL algebra has the same expressive power as relational algebra [21,22]. Moreover, SPARQL is supported by the majority of RDF-compliant frameworks, as well as triple stores and reasoners, thus further advancing the usage of standardised Semantic Web tools. As such, the contributions of the paper can be summarised in the following:

- An activity ontology pattern for the formal modelling of activity meta-knowledge, that is, contextualised relations among domain activity classes. The patterns treat domain activity classes as instances, allowing the modelling of complex class relations that go beyond tree-like dependencies and the restricted form of role chains afforded by OWL 2, such as the temporal relations among sub-activities.
- A context-aware activity recognition framework that promotes a high degree of interoperability and transparency of interpretation semantics.
- A practical framework for the combination of the standard OWL reasoning paradigm and custom activity interpretation rules in the form of SPARQL for (offline) context-based activity recognition, with well-defined semantics and high degree of interoperability.
- A proof of concept implementation and evaluation of MetaQ in a hospital for monitoring activities of people with Alzheimer's disease (AD). The system supports the dynamic translation of the activity patterns into SPARQL queries,

[1] The example follows the OWL Manchester syntax [15].

bridging the gap between conceptualisation and implementation. The prototype uses the OWL 2 RL profile and the OWLIM triple store [23] as the underlying reasoner.

This work has been motivated by two previous works of ours. In [24], the implementation of a SPARQL-based framework for activity recognition is presented, whereas the activity patterns proposed in [25] support the modelling of complex activity semantics. The present work capitalises on their combination, extending the SPARQL-based interpretation framework with activity meta-knowledge patterns. Additionally, in this work: (a) we provide a formal definition of the interpretation queries, (b) we describe the enhanced hybrid reasoning capabilities of the interpretation framework, (c) we revise and extend the activity patterns, allowing the representation of additional domain knowledge, e.g. problems, and (d) we evaluate the proposed framework in a real-world deployment setting.

The rest of the paper is structured as follows. Section 2 reviews relevant ontology-based context reasoning frameworks in the domain of activity modelling and recognition. Section 3 describes and explains the overall concept underpinning the MetaQ framework. The next two sections describe in detail the proposed context-based recognition framework and the ontology patterns. More specifically, the SPARQL-based hybrid reasoning algorithm is described in Section 4, whereas the DnS pattern-based ontology model for capturing activity meta-knowledge is described in Section 5. Section 6 elaborates on the deployment of MetaQ in a hospital for monitoring activities of Alzheimer's disease patients, presenting example use cases and results. Conclusions and future directions are presented in Section 7.

2. Related work

Several ontology-based context reasoning architectures and prototypes have been proposed in the recent past, with a substantial body of work focused on activity recognition. As aforementioned, the relevant literature comprises two research paradigms, namely approaches based solely on ontology reasoning, and approaches that use ontologies in combination with rules.

Under the first paradigm, OWL DL has been widely used within the community for building ontologies, using Description Logics to model the semantics of the domain and reducing complex activity definitions to the atemporal intersection of their constituent parts. For instance, in a smart home [9], concrete situations correspond to OWL individuals and realisation is used to determine into which context concepts a specific situation individual falls. A similar approach is followed in [8,26], where complex activities are recognised based on subsumption reasoning. In [27], a context ontology is used in order to calculate the set of activities that can be potentially performed in a given context. In OWL 2 [5,28], a number of expressivity limitations have been overcome (e.g. through property chain inclusions), thus offering the grounds for richer modelling and reasoning with complex human activities [29]. But still, due to the lack of temporal semantics and the limited support for modelling the semantics of instances composed of other [30,31], possibly interrelated instances, the purely ontology-based approaches fall short to adequately capture the temporal extension of activities and thus intricate activity patterns. In [26], the lack of temporal semantics is partially addressed by using a sliding time window to aggregate contextual data and generate an activity instance that serves as input to the subsumption reasoner for activity recognition. In [32], the ontological reasoning is extended with notions such as *recently used* and *second last activity* to simulate basic temporal reasoning.

Unlike such approaches, MetaQ allows to model and reason over intricate temporal dependencies between activities. As such, MetaQ can support through SPARQL rules the recognition of complex activities whose structure requires higher expressivity than that afforded by OWL 2 or whose structure cannot be effectively captured in a-temporal manner. Moreover, MetaQ can cope with the need to assert and refer to named individuals that are not present in the knowledge base, a situation that is very common during the inference of composite activities.

Under the second paradigm, the combination of ontologies and rules [11,33] has been embraced as means to compensate for the limitations of OWL and thus to ameliorate activity recognition in contexts where the a-temporal counterparts of the first paradigm fall short. The SWRL [34] rule language has been used in a substantial number of such frameworks. In [12], SWRL rules and the 4D fluent approach [35,36] are combined for activity modelling and representation of temporal interval relations. However, SWRL does not allow for assertion of new composite activity instances (DL safety [37]), therefore parts of the activity composition semantics are defined outside the activity model, similar to [32,26]. The SWRL language is also used in [38,39], where rules are combined with SQWRL [40] queries to infer inconsistencies. In [41], SWRL rules are combined with an activity recognition system based on Bayesian Networks and case-based reasoning. Other prominent examples of ontology- and rule-based activity recognition frameworks include the use of nRQL [42], Jena [43], Jess[2] and Prolog-like rules. In [13], the nRQL language of RacerPro [44] is used to detect and assert new complex activities taking into account temporal information. However, the framework strongly depends on the RacerPro reasoner, using the non-standardised nRQL query language. Jena rules in [45] combine ontological information relevant to location, sensors, objects and postures for activity recognition but without taking into account temporal information. The framework in [46] combines rule-based reasoning with Jess and a SAT-based event calculus reasoner.[3] An event ontology is used to capture the notions of atomic and compound events using Event Calculus axioms. In [47], Prolog-like rules are used to detect complex situations over RDF streams.

2 http://herzberg.ca.sandia.gov/.

3 http://decreasoner.sourceforge.net/.

Approaches such as the aforementioned ones afford undoubtedly more expressive and flexible solutions that the purely ontology-based ones, yet suffer from two significant shortcomings. First, the semantics of the underlying the activity models lack the rigorousness and formality that would promote sharing and reuse. Second, the interpretation semantics per se is captured by the implementation, rather than the axiomatisation [48]. A prominent example is the SWRL rule language that, even if it has been an accepted part of the W3C Semantic Web technology stack for many years, it is not supported by many state-of-the-art ontology frameworks, such as OWLIM [23], AllegroGraph,[4] Virtuoso[5] and Jena. As a result, the rule-based activity models cannot be easily shared and reused across frameworks, applications and communities with similar purpose and scope.

MetaQ, on the contrary, uses the SPARQL query language as a rule language to disengage the rule-based activity recognition from implementation aspects. To this end, the proposed framework can be realised on top of any ontology framework that support the execution of SPARQL queries. Furthermore, many extensions to the SPARQL language have been proposed for working with temporal streaming RDF data [47,49–51]. Although the processing of RDF streams is out of the scope of this work, since our focus is currently on the offline recognition of complex activities from RDF datasets, it is worth noting that the common underlying query language (SPARQL) allows for the seamless integration of stream reasoning capabilities in MetaQ. Furthermore, MetaQ allows the formal definition of activity semantics through an *ontology metamodel* [52,53]. The use of an ontology metamodel allows to overcome the limitations of other activity ontologies that have been proposed, such as [54–57], which can model only known (asserted) activity correlations and dependencies, without capturing the structure and context of complex activity semantics. For example, the ontology in [54] supports the representation of temporal relations among detected activities but falls short to provide reusable descriptions of the actual activity semantics, e.g. the activity types that are involved, or to associate activity classes with descriptive contexts, e.g. to define the frequency of occurrence of certain activities. By espousing meta-modelling, we allow the representation of contextualised views on complex activity situations, affording reusable pieces of knowledge that cannot otherwise be expressed by the standard ontology semantics.

Several ontologies have been explored as means to capture in a declarative way meta-knowledge. In [58] a top-level ontology is proposed to model the semantics common to all dimensions of an information space, i.e. levels of granularity, conflicting, and overlapping relationships that can be used to evaluate and compare concepts and terms of the ontologies built upon them. In [59] an ontology-based framework, based on the Event-Condition-Action (ECA) pattern, is presented in order to integrate heterogeneous semantic web services via rule definition. In [60] an ontology is used to model different types of event rules in order to enable automatic service discovery, while in [61], a Rule Management Ontology is presented to support the representation of event-based rules that trigger specific actions in a context-aware recommender system. Similar to [61], we aim to promote reusable and interoperable contextual activity models. However, unlike [61] that focuses on the definition of a vocabulary for representation of event-based rules, we use the DnS ontology pattern [17] of the DOLCE + DnS Ultralite (DUL) [18] to formalise abstract activity descriptions.

Finally, much research has been devoted to extending formalisms and reasoning services, so as to handle uncertain and/or vague information. Representative examples include – among others – fuzzy extensions of DLs [62,63], OWL [64,65] and SWRL [66], and probabilistic extensions such as PR-OWL [67,68] and BayesOWL [69]; for an extensive overview the reader is referred to [70]. Further relevant proposals include a pattern-based approach for representing and reasoning with fuzzy knowledge [71], and a generic, formalised approach for managing uncertainty [72]. The applicability of such initiatives in the domain of pervasive applications has been explored by a number of works, such as the approaches presented in [73], where fuzzy reasoning is used to provide personalised mobile services based on situation awareness, in [74], where probabilistic ontological reasoning is employed to progressively infer higher level activities from their simpler components, and in [75], where a fuzzy OWL 2 ontology is proposed for human activity modelling. The current implementation of MetaQ translates the meta-knowledge OWL patterns into SPARQL queries that are not able to handle uncertainty and imperfect information. Towards this direction, we are currently investigating two extensions to our framework. First, to extend the meta-knowledge OWL patterns allowing the definition of fuzzy thresholds and weights, so as to generate fuzzy SPARQL (f-SPARQL [76]) rules that can be translated into the crisp SPARQL syntax in MetaQ. In addition, we plan to enrich the interpretation layer with Defeasible reasoning [77] that constitutes an extremely suitable tool for handling uncertainty (and conflicts), e.g. through rule superiority relationships. More details regarding possible extensions are provided in Section 6.5.

3. The MetaQ framework in a nutshell

The architecture of MetaQ is depicted in Fig. 1 and consists of the *representation*, *interpretation* and *activity meta-knowledge* layers. The representation layer provides the ontology vocabulary for modelling basic activity-related information, such as domain activity classes, actors, locations, etc. Moreover, it supports the modelling of two higher level activity correlations, namely *classifications* and *compositions*. These correlations derive by the interpretation layer that infers complex situations through an iterative combination of OWL reasoning and SPARQL queries. Finally, the activity meta-knowledge layer encapsulates a formal description of the interpretation semantics of the framework by means of OWL activity meta-patterns. In the rest of this section, we briefly describe the aim and functionality of each layer.

[4] http://www.franz.com/agraph/allegrograph/.

[5] http://virtuoso.openlinksw.com/.

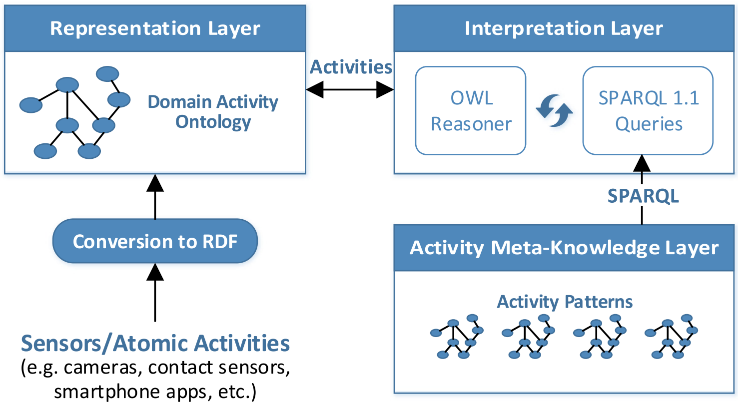


Fig. 1. The MetaQ architecture.

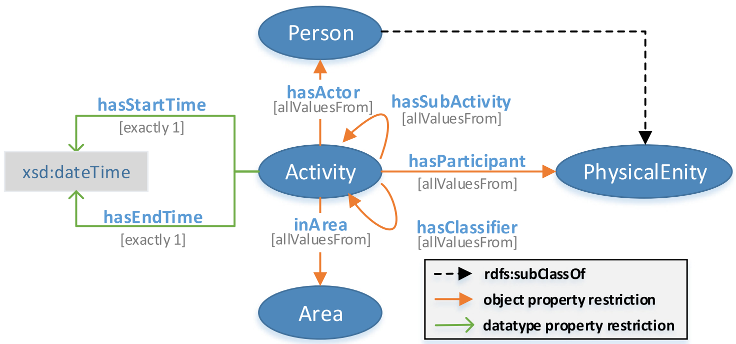


Fig. 2. The core vocabulary of the domain activity ontology.

3.1. Representation layer

The representation layer encapsulates the lightweight *Domain Activity Ontology* in Fig. 2 for capturing basic information relevant to activity monitoring. More specifically, both the atomic and complex activities are represented as instances of the `Activity` class and they are linked to ranges of time through the use of the `hasStartTime` and `hasEndTime` datatype properties. Actors are defined using the `hasActor` property, whereas relevant participating entities in an activity, such as objects or other persons, are represented using the `hasParticipant` property. Spatial information can be associated with the activities using the `inArea` property, e.g. the room where an activity takes place. Finally, the ontology supports the correlation of activities through the `hasSubActivity` and `hasClassifier` properties that derive by the interpretation procedure described in Section 4. The aforementioned modelling capabilities have been designed with a minimum of semantic commitment to guarantee maximal interoperability. As such, the Domain Activity Ontology can be aligned with relevant foundational ontologies, such as SEM [78] and Ontonym [79], reusing existing vocabularies for modelling different aspects of activities, e.g. entities, places, etc.

3.2. Interpretation layer

The interpretation layer derives complex activities by meaningfully aggregating and interpreting primitive activities. The rationale is that, although the primitive activities are informative on specific aspects of interest, the individual pieces of information themselves are not capable of delineating the situations in which the person may be involved. Combined pieces of information on the other hand can plausibly render the behaviour of the person. To this end, MetaQ implements a hybrid reasoning architecture that combines the OWL 2 reasoning paradigm and the execution of SPARQL `CONSTRUCT` queries. More specifically, the semantics of the Domain Activity Ontology, e.g. class subsumption, property restrictions, sub-properties, inverse properties and so forth, is handled by the OWL 2 reasoner, whereas the complex activity recognition is realised by domain-specific SPARQL `CONSTRUCT` queries.

Essentially, the scope of the hybrid reasoning framework that is described in Section 4, is to support the key inferencing tasks that have been described in Sections 1 and 2. To this end, the SPARQL query language is used to enhance the OWL reasoning paradigm with (a) temporal reasoning, by checking the temporal correlations among activities (Allen's temporal operators [80]), (b) complex contextual semantics that characterise higher level activities and (c) knowledge updates in terms of assertions of new activity instances.

3.3. Activity meta-knowledge layer

MetaQ provides an activity pattern ontology based on the DnS pattern of DUL that serves as a metamodel to capture the structural notions of atomic and compound activities. The metamodel is specified in two levels of granularity: (a) the *situation*, that provides an abstract description of the complex activity in terms of the domain activity types that are involved, and (b) the *description*, that can be thought of as a descriptive context that further classifies the activity classes of the situation, creating a view.

The adopted DnS pattern-oriented implementation and the alignment with the DUL foundational ontology that are described in Section 5 provide native support for modularisation and extension by domain specific ontologies, as well as, provide a further basis for future extensions. Moreover, the metamodel enables the reuse of the encapsulated semantics across applications with similar scope but different implementation frameworks, by translating the pattern-based models into the respective framework language. In MetaQ, the activity patterns are used to dynamically generate the SPARQL-based interpretation rules that implement the underlying structural and temporal activity relations.

4. Hybrid activity recognition

Given the activities of interest for an application domain and the contextual information that can be elicited directly through respective context detectors (e.g. primitive activities such as walking, a person's location, posture and objects that they interact with), the aim of the hybrid reasoning framework is to meaningfully aggregate, correlate and interpret the information, so as to elicit an understanding of the situation. To this end, the domain knowledge about the involved activities needs to be captured making use of the Domain Activity Ontology (DAO) vocabulary (Fig. 2). As in any ontology engineering task, the definition of the ontology is driven by the requirements and specifications that apply in the given application context, including those related to the capabilities and types of the available context detectors.

In this section we elaborate on the type of the interpretation queries that are supported by the framework and we describe the hybrid reasoning algorithm that incorporates the results of OWL 2 reasoning. To better illustrate the capabilities of the algorithm, we use a running example from the healthcare domain. The scenario involves the monitoring of elderly people to recognise nocturia incidences during the night sleep. The particular setting involves the following atomic activities:

- `NightSleep`: The overall night sleep duration of a person.
- `OutOfBed`: It is detected when the person is out of the bed.
- `InBathroom`: It is detected when the person is inside the bathroom. In terms of the DAO vocabulary, this concept is defined as $\texttt{InBathroom} \equiv \texttt{Activity} \sqcap \exists \texttt{inArea.Bathroom}$.

In the rest of this section, we illustrate the way MetaQ can be applied to semantically interpret and combine the aforementioned atomic activities, so as to derive:

- `BedExits`: Refer to out of bed activities occurred during the night sleep.
- `Nocturia` incidences: Inferred when bed exit activities involve bathroom visits.
- `Nocturia` problems: Night sleep activities with more than three nocturia incidences.

4.1. SPARQL interpretation queries

SPARQL is a well-established W3C standard implemented by many industrial-strength RDF APIs and triple-stores. Although it is mostly known as a query language for RDF, it is possible to use SPARQL queries as rules that can create new RDF data by combining and updating existing RDF graphs [81]. The interpretation layer of MetaQ is based on the definition of such rules using `CONSTRUCT` graph patterns.

Given an infinite set $\mathcal{U}$ of URIs, an infinite set $\mathcal{B}$ of blank nodes (i.e. anonymous RDF resources) and an infinite set $\mathcal{L}$ of literals, a triple $\langle s, p, o\rangle \in (\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L})$ is called *RDF triple*. A set of such triples is called *RDF Graph*. Most forms of SPARQL query contain a set of triple patterns, called a *basic graph pattern*, that are like RDF triples except that each of the subject (s), predicate (p) and object (o) of the triple may be a variable. A basic graph pattern matches a subgraph of the RDF graph when RDF terms from that subgraph may be substituted for the variables and the result is an RDF graph equivalent to the subgraph.

The interpretation queries in MetaQ are defined in terms of a `CONSTRUCT` and a `WHERE` clause: the `CONSTRUCT` clause defines the graph patterns, i.e. the set of triple patterns that should be added to the underlying RDF graph upon the successful pattern matching of the graph in the `WHERE` clause. Fig. 3 presents two example `CONSTRUCT` queries; the query in Fig. 3(a) links activity instances to the locations of observed objects, whereas the query in Fig. 3(b) associates patients with the location of the performed activities. Triple variables are marked by the use of "?" and "a" is used as an alternative for the predicate `rdf:type` that denotes the type of an instance. Finally, triple patterns with a common subject can be written so that the subject is only written once by employing the "`;`" notation.

The interpretation queries in MetaQ can be conceptually classified in two categories, namely *classification* and *composition* that are described in the rest of this section.

a
```
CONSTRUCT {
   ?activity :inArea ?area.
}
WHERE {
   ?activity a :Activity ;
      :observedEntity ?object .
   ?object :inArea ?area .
}
```

b
```
CONSTRUCT {
   ?person :inArea ?area .
}
WHERE {
   ?person a :Person ;
      :isActorOf ?activity .
   ?activity a :Activity ;
      :inArea ?area .
}
```

Fig. 3. (a) SPARQL query for deriving the location of an activity based on the detected objects, (b) SPARQL query for deriving the location of the person based on the location of the detected activity.

a
```
1: CONSTRUCT {
2:    ?x a D;
3:       :hasClassifier ?y_i ;
4:       ...
5:       :hasClassifier ?y_m .
6: }
7: WHERE {
8:    ?x a C .
9:    ?y_i a J_i .
10:    ...
11:    ?y_m a J_m .
12:    FILTER (func(?x, ?y_i, ...) &&
          ... && func(?x, ?y_m, ...)) .
13: }
```

b
```
CONSTRUCT {
  ?x a :BedExit;
    :hasClassifier ?y.
}
WHERE{
  ?x a OutOfBed;
    :hasStartTime ?st1;
    :hasEndTime ?et1.
  ?y a NightSleep;
    :hasStartTime ?st2;
    :hasEndTime ?et2.
  FILTER(contains(?st2, ?et2, ?st1, ?et1)) .
}
```

Fig. 4. (a) The abstract form of the classification queries, (b) the classification query for the `BedExit` activity.

4.1.1. Classification queries

The classification queries propagate activity instances in the activity class hierarchy based on contextual dependencies with other activities, called *classifiers*. The `WHERE` clause of such queries contains graph patterns that match classifiers in terms of their types, property values and temporal constraints, whereas the `CONSTRUCT` clause defines the classification of an activity in a new class.

More specifically, let C and D be two activity classes of the Domain Activity Ontology ($\mathcal{DAO}$), such that $C \in \mathcal{DAO}$, $D \in \mathcal{DAO}$ and $C \neq D$. Moreover, let $\mathcal{J}_{C\rightarrow D}$ be the set of classifiers ($\mathcal{J}_{C\rightarrow D} \subset \mathcal{DAO}$) that comprises the *classification context* of activities from class C to class D. The abstract query $Q_{\mathcal{J}_{C\rightarrow D}}$ in Fig. 4(a) defines the graph pattern for the realisation of the $C \rightarrow D$ classification. More specifically, the `WHERE` clause consists of a triple pattern for matching instances and properties of class C (line 8), one or more triple patterns for matching instances and properties of classifiers $J_i \in \mathcal{J}_{C\rightarrow D}$ (lines 9–11) and one or more constraints (e.g. temporal) to be applied to the graph pattern matching in the form of `FILTER` expressions. The `CONSTRUCT` clause defines the classification of x to class D (line 2) that is further associated with its classifiers through the `hasClassifier` property (lines 3–5).

Example. The bed exit activities in our running example can be derived using the classification query in Fig. 4(b), where C = `OutOfBed` and D = `BedExit`, with `NightSleep` comprising the classification context (classifier), i.e. $\mathcal{J}_{\texttt{OutOfBed}\rightarrow\texttt{BedExit}} \equiv \{\texttt{NightSleep}\}$. The query matches in the `WHERE` clause `NightSleep` activities that contain[6] `OutOfBed` activities, classifying the latter as `BedExits` in the `CONSTRUCT` clause. Each `BedExit` instance is semantically associated with its classifier through the `hasClassifier` property, i.e. the `NightSleep` instance, allowing the modelling of the classification dependency between the two activity types that can be used to derive further inferences, as we describe in Section 4.2.

4.1.2. Composition queries

The composition queries derive composite activities, i.e. structured activities [30] that are composed of other activities, called *sub-activities*. In contrast to the classification queries that consider only existing activities, the composition queries assert new activity instances to the activity graph, along with information about their temporal extension, i.e. the start and end times.

More specifically, let C be a composite activity of the Domain Activity Ontology, such that $C \in \mathcal{DAO}$. Moreover, let $\mathcal{S}_C$ be the set of the sub-activities of C ($\mathcal{S}_C \subset \mathcal{DAO}$). The abstract query $Q_{\mathcal{S}_C}$ in Fig. 5(a) defines the graph pattern for the derivation of composite activities of type C. The `WHERE` clause defines triple patterns for matching instances and properties

[6] The SPARQL function `contains` checks whether the first time interval (`[?st2, ?et2]`) contains the second (`[?st1, ?et1]`).

a
```
1: CONSTRUCT {
2:   ?new a C;
3:        :hasStartTime ?st_k;
4:        :hasEndTime ?et_k;
5:        :hasSubActivity ?y_i ;
6:        ...
7:        :hasSubActivity ?y_m .
8: }
9: WHERE {
10:   ?y_i a S_i .
11:   ...
12:   ?y_k :hasStartTime ?st_k ;
13:        :hasEndTime ?et_k .
14:       ...
15:   ?y_m a S_m .
16:   FILTER (func(?x, ?y_i, ...) &&
          ... && func(?x, ?y_m, ...))) .
17:   BIND (URI(?y_i, ..., ?y_m) as ?new) .
18:   NOT EXISTS {?new a [ ] .}
19: }
```

b
```
CONSTRUCT {
  ?new a :Nocturia;
    :hasStartTime ?st1;
    :hasEndTime ?et1;
    :hasSubActivity ?x;
    :hasSubActivity ?y.
}
WHERE {
  ?x a BedExit;
    :hasStartTime ?st1;
    :hasEndTime ?et1.
  ?y a :InBathroom;
    :hasStartTime ?st2;
    :hasEndTime ?et2.
  FILTER(contains(?st1, ?et1, ?st2, ?et2)) .
  BIND(URI(?x, ?y) as ?new) .
  NOT EXISTS {?new a [ ] .}
}
```

Fig. 5. (a) The abstract form of the composition queries, (b) the composition query for `Nocturia`.

of sub-activities (lines 10–15) and, similar to the classification queries, one or more `FILTER` expressions can be defined as constraints on the activity graph (line 16). In addition, each time a new assertion is derived, the query assigns a URI (Universal Resource Identifier) that encapsulates the assertions used to derive it (lines 17 and 18). The *URI* function actually generates a unique URI for the new activity by concatenating its sub-activity URIs. Thus, each query is executed only once for a certain set of sub-activities (safety condition in line 18), ensuring termination. Finally, the `CONSTRUCT` clause assigns the composite activity to class C (line 2). The instance is further linked to ranges of time that have been matched in the `WHERE` clause (lines 3 and 4) and is associated with its sub-activities through the `hasSubActivity` property.

Example. The query in Fig. 5(b) handles the composition semantics of nocturia, using the classes `BedExit` and `InBathroom` as its sub-activities, i.e. $\mathcal{S}_{\texttt{Nocturia}} \equiv \{\texttt{BedExit}, \texttt{InBathroom}\}$. The query matches in the `WHERE` clause `BedExits` that contain `InBathroom` activities, generating a new URI for each matching. If there is no other instance with the same URI in the activity graph, a new nocturia instance is derived in the `CONSTRUCT` clause that inherits the temporal extension of the `BedExit` activity and is further associated with its sub-activity instances (`hasSubActivity`). It is worth noting the dependency between the two rules of our example, since the `WHERE` clause of the composition rule matches `BedExit` activities that are inferred by the classification rule.

4.2. Reasoning algorithm

Despite the fact that the standard semantics of OWL are insufficient to support the classification and composition semantics in MetaQ, they still play an important role in the framework. The semantics of the Activity Domain Ontology, such as class subsumption, property domain/range restrictions, instance class memberships, property relationships, e.g. transitive, inverse, etc., can be efficiently handled by OWL 2 ontology reasoners. For example, an activity that has been detected in the bathroom is automatically classified by the OWL reasoner in the `InBathroom` class of the domain ontology, based on the axiom $\texttt{InBathroom} \equiv \texttt{Activity} \sqcap \exists\texttt{inArea.Bathroom}$ we described in Section 4.

In order to combine the standard reasoning services of OWL and the extended SPARQL-based reasoning services of the interpretation layer, MetaQ defines the hybrid reasoning algorithm in Fig. 6. Assuming that G_a is the RDF graph with the atomic activity assertions, Q_J is the set with the classification queries $Q_{\mathcal{J}_{C\to D}}$, Q_S is the set with the composition queries, R_{OWL} is the OWL reasoning module and R_{SPARQL} is the SPARQL reasoning module, the algorithm in Fig. 6 updates G_a with additional interpretation results. More specifically, the architecture follows an iterative combination of OWL reasoning and SPARQL query execution. Initially, the OWL reasoning module is used over G_a to derive inferences based on the standard OWL semantics ($R_{OWL}(G_a)$). These inferences are added back to the G_a set (line 3) that is subsequently used as the underlying activity graph for the SPARQL-based activity recognition process (lines 4–9), where each classification and composition query is used by the SPARQL query module to derive further inferences. The results of each query execution are added to the G' set (lines 5 and 8). When all SPARQL queries have been executed, the derived assertions in the G' set are added to the G_a graph (line 10), completing a reasoning iteration. If R_{SPARQL} does not produce any inferences, i.e. $G' = \emptyset$ (line 11), then G_a contains the complete set of the atomic and inferred activities that can be derived and the algorithm terminates. Otherwise, a new reasoning iteration begins to update the G_a graph with additional inferences.

In effect, the reasoning algorithm consists of successive steps of OWL reasoning, materialisation, and SPARQL queries execution. MetaQ can be characterised as a loosely-coupled bidirectional hybrid framework in the sense that the two

Require: $G_a \neq \emptyset, Q_J, Q_S$
repeat
  $G' \leftarrow \emptyset$
  $G_a \leftarrow G_a \cup R_{OWL}(G_a)$
  for all $Q_{\mathcal{J}_{C\rightarrow D}} \in Q_J$ **do**
    $G' \leftarrow G' \cup R_{SPARQL}(Q_{\mathcal{J}_{C\rightarrow D}}, G_a)$
  end for
  for all $Q_{S_C} \in Q_S$ **do**
    $G' \leftarrow G' \cup R_{SPARQL}(Q_{S_C}, G_a)$
  end for
  $G_a \leftarrow G_a \cup G'$
until $G' = \emptyset$

Fig. 6. The hybrid complex activity recognition algorithm.

reasoning modules are separate: the materialisation results of OWL reasoning are sent to the SPARQL module for executing the rules on top of the information. Then the results of SPARQL are sent back to the OWL reasoner. In principle, decidability in frameworks that combine ontologies and rules is ensured by allowing rule variables to bind only to explicitly named individuals. SPARQL follows the Closed-World Assumption (i.e. the knowledge available is thought to be a complete encoding of the domain of interest) and hence inherently satisfies this condition. Therefore, even if OWL reasoning follows the OWA, the SPARQL-based interpretation layer queries the underlying knowledge in a CWA-like manner. Also note that, in the generation of new individuals during the execution of the SPARQL queries, unique URIs are enforced that ensure that an individual that has already been generated in a previous reasoning cycle, cannot be reintroduced in a subsequent one (`NOT EXISTS` triple pattern operator).

As far as the complexity of the algorithm is concerned, it is delineated by the computational complexity of the two underlying reasoning modules that are successively invoked. More specifically, the computational complexity of the OWL 2 reasoning module depends on the OWL 2 profile used for modelling domain knowledge.[7] In our prototype implementation, we have used the OWL 2 RL profile and the OWL 2 RL reasoner of OWLIM (NP-COMPLETE). By using, for example, the OWL 2 DL fragment (under direct semantics), the reasoning complexity increases to N2EXPTIME-complete, supporting though higher expressivity. Regarding SPARQL, the computational complexity strongly depends on the language constructs used for defining queries/rules. As it has been reported in [21,82], the full SPARQL (e.g. using FILTER, UNION, OPTIONAL operators) is PSPACE-complete, whereas all OPTIONAL-free graph patterns are either NP-COMPLETE (whenever operator AND cooccurs with UNION or SELECT/CONSTRUCT) or in PTime. Specifically in MetaQ, the graph patterns are defined using AND and FILTER operators in SELECT/CONSTRUCT queries with complexity NP-COMPLETE, similar to OWL 2 RL.

Example. The semantics of the nocturia problem in our running example can be handled by a classification query, where $C = \mathtt{NightSleep}, D = \mathtt{NocturiaProblem}$ and $\mathcal{J}_{\mathtt{NightSleep}\rightarrow\mathtt{NocturiaProblem}} \equiv \{\mathtt{Nocturia}\}$. In this case, the `COUNT` aggregation operator of SPARQL can be used to match `NightSleep` instances that contain more than three `Nocturia` events, so as to be classified in the `NocturiaProblem` class. However, in order to better illustrate the hybrid interpretation capabilities of MetaQ, we describe the way the OWL 2 and SPARQL reasoning modules can be combined to derive nocturia problems.

Assuming that `classifies` and `isSubActivityOf` are the inverse properties of the `hasClassifier` and `hasSubActivity` properties of DAO, respectively, the following complex class description can be used by an OWL 2 reasoner to further classify `NightSleep` instances in the `NocturiaProblem` class:

$$\mathtt{NocturiaProblem} \equiv \mathtt{NightSleep} \sqcap \geq 3\,\mathtt{classifies}(\mathtt{BedExit} \sqcap \exists \mathtt{isSubActivityOf}\ \mathtt{Nocturia}).$$

The axiom defines the classification semantics using the activity types and high-level property assertions that are derived by the SPARQL reasoning module (hybrid approach). More specifically, a `NightSleep` instance is classified as a `NocturialProblem` if it is the classifier of more than three `BedExits` that are, in turn, sub-activities of some `Nocturia` activity. In that way, the axiom considers only bed exists that have been linked to some bathroom visit by the SPARQL module, ignoring bed exits that involve, for example, only a visit in the kitchen. It is worth mentioning that the two high-level property assertions `hasClassifier` and `hasSubActivity`, apart from the classification and composition dependencies they impose on the linked activities, they also encapsulate temporal context. For example, the fact that a night sleep activity classifies a bed exit activity also implies that the bed exit activity took place during the night sleep, due to the temporal constraints of the query in Fig. 4(b). Through the use of the two high-level property assertions, the hybrid reasoning architecture of MetaQ allows the OWL 2 reasoning module to incorporate contextual information from the SPARQL-based reasoning module that cannot be otherwise derived by the semantics of OWL, such as instances of composite activities, e.g. `Nocturia` instances (Fig. 5(b)).

7 http://www.w3.org/TR/owl2-profiles/#Computational_Properties.

4.3. Advanced hybrid reasoning scheme

So far, we have presented the basic syntax of the SPARQL rules and the hybrid reasoning algorithm of MetaQ. Through a running example, we demonstrated the detection of an abnormal situation where the SPARQL assertions regarding the temporal correlation between a `BedExit` and a `Nocturia` event are added to the OWL 2 reasoning module for the classification of the `NightSleep` instance as a nocturia problem. These reasoning capabilities have been tested in practice through the real-world deployment of MetaQ in a hospital for monitoring AD patients (a detailed description of the deployment is presented in Section 6).

In order to better illustrate the hybrid reasoning capabilities of MetaQ, we present in this section an advanced hybrid reasoning scheme that involves a bidirectional interaction between the two MetaQ reasoning modules. More specifically, OWL 2 is used in this setting for defining the semantics of *situations*, i.e. abstract contexts associated with primitive observations that indicate the presence of complex activities, such as tea preparation. The inferred situation instances are then sent to the SPARQL module for deriving additional contextualisations, taking into account the temporal aspects of situations. The inferred SPARQL contexts are finally sent back to the OWL 2 reasoning module for the recognition of the complex activities.

The aforementioned reasoning scheme is part of an ongoing investigation regarding the deployment of MetaQ in home settings. One of the key challenge is such environments is to support the continuous, unrestricted monitoring of activities, as opposed to the deployment in the hospital (see Section 6) that involves the execution of a protocol with certain constraints. The key idea that underpins our approach is that contextual information is described in two levels of abstraction: *situation context* and *temporal context*. The situation context involves the use of OWL 2 semantics to define (in an a-temporal manner) the logical dependencies among the primitive observations that characterise certain situations of the domain through concept equivalence axioms. On the other hand, the temporal context encapsulates the temporal dependencies that need to be satisfied by the situation, so as to be classified as a complex activity.

As an example, suppose the following OWL axiom that models the situation `HotTeaSituation`:

$$\texttt{HotTeaSituation} \equiv \texttt{Situation} \sqcap \exists\texttt{involves.TeaBag} \sqcap \exists\texttt{involves.KettleOn} \sqcap \exists\texttt{involves.FillCup} \sqcap \exists\texttt{involves.InKitchen}$$

Intuitively, a situation context describes abstract dependencies among domain observations that signify the presence of complex activities. For example, the above axiom would classify a `Situation` instance in the `HotTeaSituation` class, if the situation is associated with a tea bag object, a kettle on and fill cup event and a kitchen-related localisation through the `involves` property. In practice, such situation instances can be generated by periodically collecting[8] and associating generated observations through `involves` property assertions.

The majority of the ontology-based approaches in the literature adopt such an a-temporal approach to conceptualisation. For example, the semantics of the activity `PrepareHotTea` would be equivalent to the semantics of the `HotTeaSituation` axiom, i.e. `PrepareHotTea` ≡ `HotTeaSituation`. However, the lack of inherent temporal semantics considerably restricts the practicality and application scope of OWL in domains that require the recognition of complex context elements, such as human activities that are generally characterised by intricate temporal associations. For instance, in our example, although the fact that a situation containing tea-related objects and events is an indication of an ongoing `PrepareHotTea` activity, the temporal aspects of the context provide equally important information that should be taken into account for the final derivation of a successful tea preparation context, e.g. to ensure that all the tea-related objects and events have been recognised when the person was inside the kitchen and that the cup was filled after having boiled the water. As such, the `PrepareHotTea` activity is defined in MetaQ as:

$$\texttt{PrepareHotTea} \equiv \texttt{HotTeaSituation} \sqcap \texttt{HotTeaPreparationContext}$$

The `HotTeaPreparationContext` encapsulates the temporal aspects of the tea preparation activity and it is defined in terms of the following SPARQL classification query:

```
CONSTRUCT {
 ?x a :HotTeaPreparationContext .
}
WHERE{
 ?x a HotTeaSituation; :hasStartTime ?st; :hasEndTime ?et.
 ?ki a InKitchen; :hasStartTime ?st1; :hasEndTime ?et1.
 ?ke a KettleOn; :hasStartTime ?st2; :hasEndTime ?et2.
 ?t a TeaBag; :hasStartTime ?st3; :hasEndTime ?et3.
 ?f a FillCup; :hasStartTime ?st4; :hasEndTime ?et4.
FILTER (contains(?st, ?et, ?st1, ?et1, ..., ?st4, ?et4)).
FILTER (contains(?st1, ?et1, ?st2, ?et2, ..., ?st4, ?et4)).
FILTER (before(?st2, ?et2, ?st4, ?et4)).
}
```

[8] The time window length and/or the triggering criteria of the observation collection process are still under investigation.

The rule matches the `HotTeaSituation` instance that has been inferred by the OWL 2 reasoning module (line 5), retrieves the observations[9] that are temporally associated with this context (line 10) and checks their temporal relations (lines 11, 12). If the graph pattern in the `WHERE` clause is satisfied, the `HotTeaSituation` instance is further classified in the `HotTeaPreparationContext`. This additional assertion will be available in the next reasoning iteration of the algorithm in Fig. 6, where the OWL reasoner with finally classify the situation instance in the `PrepareHotTea` class, provided that both members of the intersection are satisfied.

5. Activity meta-knowledge patterns

The hybrid activity interpretation framework in MetaQ adopts a standard-based approach to address the shortcomings of the native OWL reasoning, using the SPARQL query language to enrich RDF activity graphs with higher level inferences. However, the fact that part of the activity semantics are defined in queries and not in formal, semantically enriched models, hampers the reuse and extensibility of the underlying activity models. For example, the SPARQL query in Fig. 5(b) ensures the syntactic interoperability of the nocturia interpretation logic across frameworks that support the execution of SPARQL queries. It fails though to formally capture the semantics that underpin the nocturia context, e.g. to provide a machine manipulable model of the situation that `Nocturia` is composed of a `BedExit` and `InBathroom` activities.

To promote a well-defined description of complex activity contexts and achieve a better degree of knowledge sharing and reuse, an activity pattern ontology has been developed as extension of the Descriptions and Situations (DnS) [17] pattern of the DOLCE + DnS Ultralite (DUL) ontology [18]. The aim of the proposed ontology is to formally capture *activity meta-knowledge*, that is, the structural notions of atomic and composite activities and to define operators among activity sets. Towards this end, the ontology makes use of the meta-modelling capabilities of OWL 2, namely *punning* [20] that allows treating domain activity concepts as instances, allowing property assertions to be made among domain activity concepts. In this way, the proposed activity patterns enable to formally represent contextualised views on complex activities, and afford reusable pieces of knowledge that cannot otherwise be directly expressed by the standard ontology semantics, e.g. temporal correlations among activities that are not connected in a tree-like manner [13].

It is important to note that by affording a conceptualisation (i.e. the representation and activity meta-knowledge layer), the interpretation is not only decoupled from the implementation but has also rigorous, formal semantics. MetaQ makes it is rather straightforward to share and reuse activity definitions among activity recognition frameworks that implement different interpretation strategies.

In the following, we briefly describe the DnS pattern and its implementation in DUL. We then introduce the core activity pattern and two instantiations to capture the semantics of the classification and composition queries in MetaQ.

5.1. Core activity pattern

The DnS design pattern provides a principled approach to context reification through a clear separation of states-of-affairs, i.e. a set of assertions, and their interpretation based on a non-physical context, called a *description* [17]. Intuitively, DnS axioms try to capture the notion of "*situation*" as a unitarian entity out of a state of affairs, with the unity criterion being provided by a "*description*". In that way, when a description is applied to a state of affairs, a situation emerges. In MetaQ, we use DnS to formally provide precise representations of contextualised situations and descriptions on activity concepts of the Domain Activity Ontology, describing the different activity types and temporal relations that can be associated with complex domain activities.

The basic implementation of the DnS pattern in DUL allows the relation of situations (`dul:Situation`) and descriptions (`dul:Description`) with domain events (`dul:Event`) and concepts (`dul:EventType`). More specifically, a situation describes the entities of a context, e.g. the events/activities that are involved, and satisfies (`dul:satisfies`) a description. The description in turn defines (`dul:defines`) concepts that classify (`dul:classifies`) the entities of the situation, describing the way they should be interpreted. A known implementation of the DUL DnS pattern is the Event-Model-F [83] ontology that implements a number of instantiations on top of the DnS pattern to describe relations among events, such as causality and correlation. For example, Event-Model-F (`emf`) allows the association of composite events with their sub-events through descriptions that use the concepts $\mathtt{emf:Composite} \sqsubseteq \mathtt{dul:EventType}$ and $\mathtt{emf:Component} \sqsubseteq \mathtt{dul:EventType}$. In that way, an instance of the `Composite` class can be used to classify a detected "make hot tea" activity, whereas its sub-activities "use kettle" and "use tea bug" can be classified by instances of the `Component` class.

The conceptual model of Event-Model-F, however, allows the representation of asserted relations only, e.g. relations among already recognised activities and their known sub-activities, without capturing the structure and semantics of the respective derivations. In contrast, the scope of the core activity pattern in MetaQ is to conceptually describe the activity context and semantics that define complex activities *at the class level*, and not to capture relations directly among asserted activity instances. Towards this end, we have reused and extended the DnS pattern of DUL for the classification of activity classes instead of activity instances, allowing the definition of contextualised views at the schema level. Fig. 7 illustrates the classes and properties of the core activity pattern, together with their alignment with the DUL vocabulary. The pattern

[9] The rule follows the template presented in Fig. 4(a), without exploiting the `involves` property of situations.

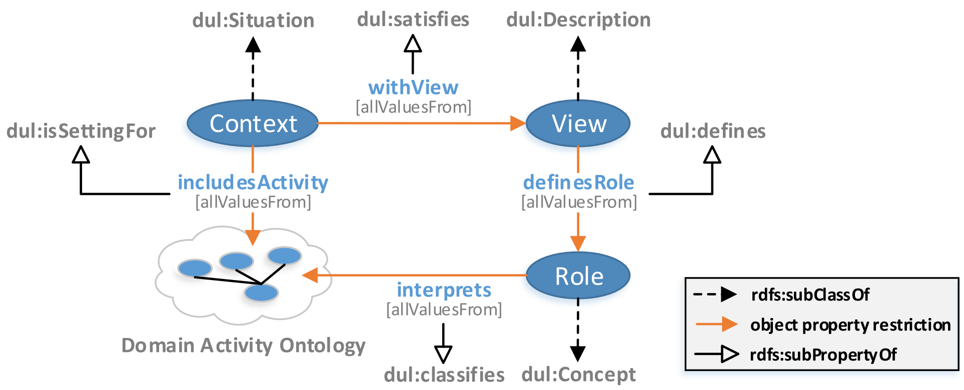


Fig. 7. The core activity pattern.

allows the representation of the following activity-related conceptualisations:

- **Context**: A context is a `dul:Situation` that defines the set of activity classes that are involved (`includesActivity`) in a specific pattern instantiation and it is described on the basis of a view (`withView`).
- **View**: A view is a `dul:Description` that serves as a viewpoint over a context, defining the roles (`definesRole`) that the activity classes of the context play.
- **Role**: A role is a `dul:Concept` that interprets (`interprets`) an activity class in the context of a specific instantiation.

Roughly speaking, the definition of a pattern is specified in two levels of granularity: (a) the context/situation, that provides an abstract description of the activity in terms of the domain activity classes that are involved, and (b) the view/description, that can be thought as a descriptive context that interprets the activity classes of the context through roles, creating a view.

In the following, we describe the two implementations of the core activity pattern to capture the classification and composition semantics in MetaQ. We also present example instantiations to support the modelling of the `BedExit` and `Nocturia` activities of the running example in Section 4. Finally, we describe the use of the patterns to dynamically generate the interpretation queries in MetaQ, as depicted in Fig. 1.

5.1.1. Classification pattern

The classification pattern enables to formally capture complex activities that are defined as further classifications of a given atomic or other complex activity, as it has been described in Section 4.1.1. As shown in Fig. 8(a), a definition of this type is expressed by a `Classification` context that is linked to a `Cl_View`. The pattern defines three roles for the interpretation of the activities that are included in a classification: the `Asserted` and `Derived` roles express the asserted and derived activity type of the classification, respectively, while the `Classifier` role is used to model the classifiers of the context. In order to enable the modelling of temporal correlations among the interpreted activities, all the roles of the pattern subsume the `time:TemporalEntity` class of the OWL Time vocabulary [84].

Fig. 8(b) presents the instantiation of the classification pattern to capture the semantics of the `BedExit` activity. The instance `c` is linked to the activity classes `OutOfBed`, `NightSleep` and `BedExit` that comprise the classification context. In addition, `c` is linked to `v` that captures the view over the classification context by defining the role instances (`rx`) that interpret the activity classes. More specifically, the `OutOfBed` activity in interpreted as the `Asserted` activity by `r1`, `NightSleep` is interpreted as the classifier by `r2` and `BedExit` is the `Derived` activity type of the classification, as denoted by `r3`. In addition, the instantiation of the pattern defines the temporal constraint that should be met between the role instances `r2` and `r3`, i.e. the night sleep should contain an out of bed activity.

5.1.2. Composition pattern

The composition pattern enables to formally capture complex activities that are defined as the composition of atomic or other complex activities. As shown in Fig. 9(a), a composite activity is expressed by a `Composition` context that defines the classes of the composition (`includesActivity`), and a `Cm_View` that defines the roles for the interpretation of the contextual activities. More specifically, the `SubActivity` role interprets the sub-activities, while the `Composite` role interprets the composite activity of the composition, respectively. Similar to the classification pattern, these roles subsume the `time:TemporalEntity` class of OWL Time. In addition, the pattern defines the roles `Start` and `End` that are used to denote the temporal boundary of the composite activity.

Fig. 9(b) presents the instantiation of the composition pattern for the `Nocturia` activity. The instance `c` is linked to the activity classes `BedExit`, `InBathroom` and `Nocturia` that admit the composition context, while `v` captures the interpretation of the activities. More specifically, the `BedExit` and `InBathroom` activities are designated as the `SubActivities` (`r1` and `r2`), while the `Nocturia` activity is the composite activity of the pattern (`r3`). In addition, the

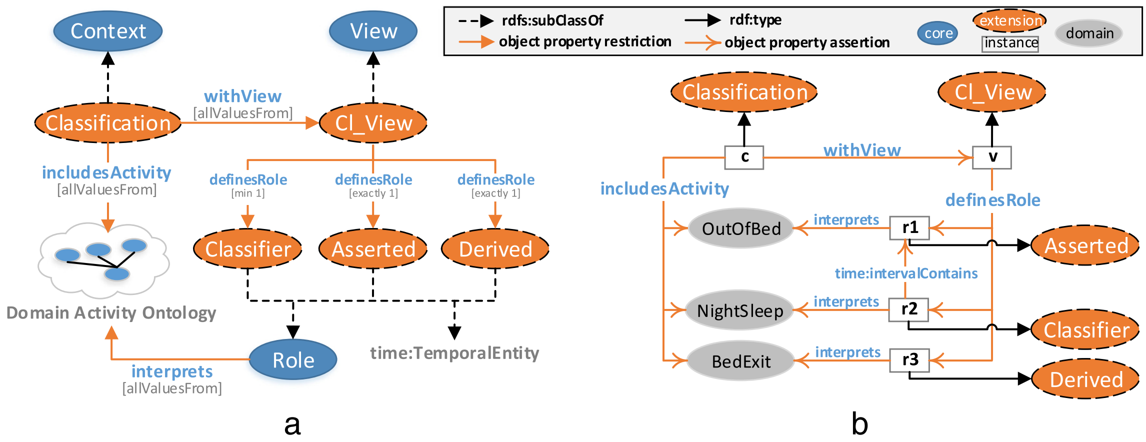


Fig. 8. (a) The classification pattern, (b) the `BedExit` instantiation.

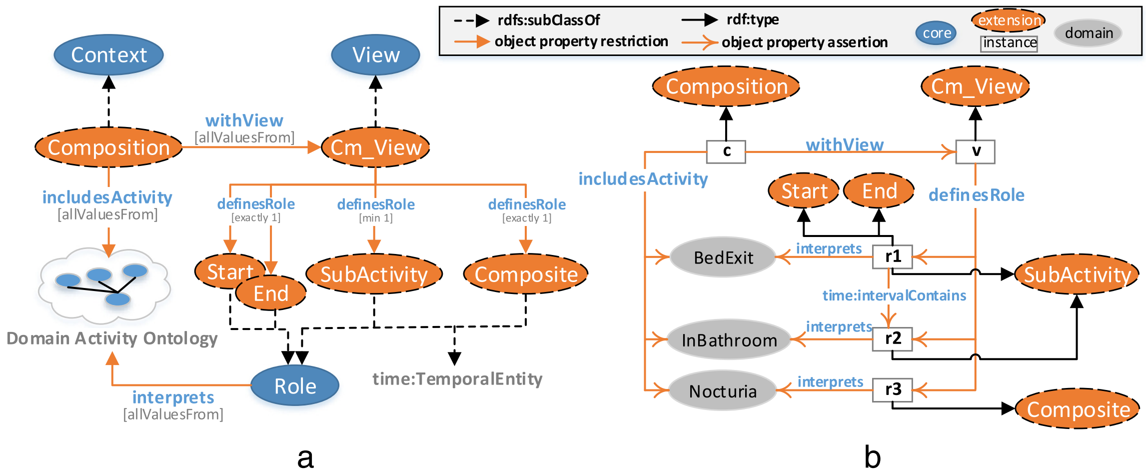


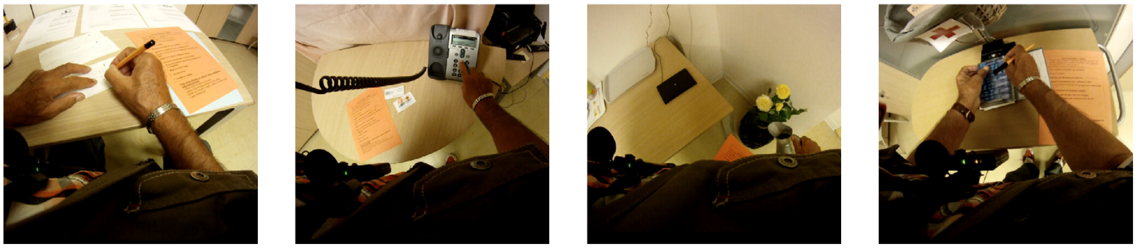
Fig. 9. (a) The composition pattern, (b) the `Nocturia` instantiation.

example defines that the `BedExit` should contain the `InBathroom` activity, and that `BedExit` is both the initiating and terminating activity (`r1` belongs to both `Start` and `End` roles). In that way, the composite activity inherits the temporal extension of the `BedExit`, as the rule in Fig. 5(b) implements. It is worth mentioning the fact that the class `BedExit` is given two different interpretations in the same domain. For example, the `BedExit` class has the `Derived` role in the pattern of Fig. 8(b), while the same class is also a sub-activity in the example of Fig. 9(b).

5.2. Pattern transformation into SPARQL

In addition to serving as formal, reusable models for capturing activity meta-knowledge, the defined patterns are used in MetaQ for implementing the activity interpretation semantics. Based on the DnS and DUL compliant design principles, we have defined an automated translation of the encapsulated semantics into SPARQL queries. As such, the interpretation semantics of the framework can be conceptually defined by means of formal ontology models, as well as, they can be dynamically extended and adapted to reflect changes in the interpretation logic, e.g. after learning new behavioural profile patterns.

In the case of the classification pattern, the activity classes that are interpreted by the `Asserted` and `Classifier` roles, together with their temporal assertions, are used to define the triple patterns that match the corresponding activity instances in the `WHERE` clause of the query. For example, the classes `OutOfBed` and `NightSleep` in Fig. 8(b) are used to generate the triple patterns in the `WHERE` clause of the query in Fig. 4(b), whereas the temporal pattern constraint between the two roles that classify these two classes is used to generate the `FILTER` expression. Moreover, the `Derived` role that interprets the activity type of the classification is used to define the triple pattern in the `CONSTRUCT` clause that asserts the additional class type of the classified instance. In the example of Fig. 8(b), this is the `BedExit` class that is transformed into the `CONSTRUCT` triple pattern of Fig. 4(b). The `hasClassifier` property assertions in the query derive directly from the classifiers of the pattern.



(a) Writing a check. (b) Making a phone call. (c) Watering the plant. (d) Preparing the drug box.

Fig. 10. Example IADL activities of the lab protocol (wearable camera screenshots).

In a similar manner, the activity classes that are interpreted by the `SubActivity` role, together with their temporal assertions, are used to define the triple patterns that match the corresponding activity instances in the `WHERE` clause of a composition query. For example, the classes `BedExit` and `InBathroom` in Fig. 9(b) are used to generate the triple patterns in the `WHERE` clause of the query in Fig. 5(b), whereas the temporal constraint between the two activity types is used to generate the `FILTER` expression. Moreover, the `Composite`, `Start` and `End` roles are used to define the class of the composition and the temporal extension of the new activities in the `CONSTRUCT` clause of the query. In the example of Fig. 9(b), the class `Nocturia` has the composite role and inherits the temporal extension of the `BedExit` activity. The `hasSubActivity` property assertions in the query derive directly from the sub-activities of the pattern.

6. MetaQ deployment

We have deployed MetaQ in a hospital for monitoring Alzheimer's disease patients.[10] The aim of this deployment is to help clinicians assess the condition of the patients, based on a goal-directed protocol where participants perform predefined activities in an experimentation room. The participants have to perform a list of 10 Instrumental Activities of Daily Living (IADL), such as preparing the drug box, talking on phone and watering the plant (Fig. 10), within a timeframe of 20 min. MetaQ is used in this context for detecting the IADLs performed by the participants and to inform the clinicians, who are not in the room during the execution of the protocol, about problems, such as activities that have been missed or repeated, or problems regarding the duration of activities.

The rest of this section is structured as follows: Section 6.1 briefly describes the aim of the protocol and the ADLs the participants have to perform. Section 6.2 provides detail regarding the inclusion criteria of the participants and the sensors/components used in the installation. The next section presents details about the definition of the OWL patterns and key clinical requirements, while Section 6.4 contains example pattern instantiations and SPARQL queries. Finally, Section 6.5 presents the evaluation results.

6.1. *Overview of the goal-directed protocol*

The protocol focuses on the improvement of assessment tools for the diagnosis of Alzheimer's disease at early stages, through short-term tests proposed during a standard memory consultation. The data collection is conducted in an experimentation room equipped with home appliances, where participants have to perform predefined activities of daily living. The setting includes video sensors (ambient and wearable), audio sensors (ambient and wearable), accelerometers, energy consumption and physiological sensors, as well as software modules for the recognition of low-level events (referred to as *observations*). More details about the set of sensors and components used in the evaluation are provided in the Section 6.2.

All in all, the task-based protocol aims to collect patient data in a controlled environment towards the implementation of an objective assessment of autonomy and goal-oriented cognitive functions at different stages of the AD. Through this protocol, it is envisioned that the clinical practitioners will acquire complementary objective information that would help them to better assess the difficulties of participants in activities of daily living compared to the current rating scales based on interviews used in clinical practice. Table 1 summarises the list of activities the participants have to perform.

It should be noted that the goal-directed session is part of an overall clinical protocol that also involves medical and clinical consultation. The results of all phases are used by the clinical practitioners for providing an objective assessment about the condition of the participant. In the following subsections we provide details regarding the experimental setup and the definition of the interpretation semantics of MetaQ for the recognition of the activities during the goal-directed test. The description or validation of the overall clinical protocol is out of the scope of the paper.

[10] The system has been installed in the Memory Resource and Research Centre (CMRR) of the University Hospital in Nice (CHUN), under the Dem@Care FP7 EU Project (http://www.demcare.eu/).

Table 1
Protocol activities.

ADL	Description
Establish account balance	The participant has to establish the amount balance of two bills (e.g. electricity and phone)
Prepare drug box	The participant has to check that the drug box is not empty and contains the suggested posology
Prepare hot tea	The participant has to switch on the electric kettle and pour the water on the glass containing the tea bag
Search for bus line	The participant has to write on a sheet of paper the bus lines to take for a provided itinerary
Make a phone call	The participant has to make a phone call
Watch TV	The participant has to take the remote control and turn on the TV
Water the plant	The participant has to take the water can and water the plant
Write a check	The participant has to pay the phone bill by check
Read an article	The participant needs to read an article and write down the answers to three questions
Leave the room	The participant has to leave the room when they feel that all activities have been performed

6.2. Experimental setup

For the evaluation of the framework, we used a dataset of 9 patients with Mild Cognitive Impairment (MCI) and 10 patients with Alzheimer's Disease (AD). The inclusion criteria for both groups were:

- Male or female ≥65 years (mean age of participants: 78 years).
- Patients with a score of 0 to items of "tremors" and "muscle stiffness" of the UPDRS III [85].
- Patients with no criteria for major depressive episode according to DSM IV-R [86].

In addition, for the MCI group, patients with a diagnosis of MCI were included according to the criteria of the National Institute on Ageing and Alzheimer's Association group [87], or with predemential AD stage [88]. For the AD group, the inclusion criteria involved diagnosis of AD according to NINCDS-ADRDA [89] or Alzheimer's typical or atypical [89] and Mini-Mental State Exam (MMSE) [90] ≥16.

As described in Sections 4 and 5, MetaQ provides an ontology-based framework for aggregating and interpreting primitive observations generated by multiple sources. In practice, these events may be relevant to direct sensor outputs, e.g. activations of motion sensors, or they may refer to results of intermediate analysis of sensor data, e.g. by applying posture recognition algorithms (e.g. pattern recognition) on the collected videos. The scope of this section is to present the way MetaQ is used to aggregate and interpret the various multi-modal observations generated during the protocol. Note that fusion is performed after having collected all the available data/results from all the sensors/components of our installation in an offline manner, after the participant has completed the protocol.

That said, MetaQ is used in our experiment to aggregate and interpret four types of information:

- Location: Information about the location of the participant during the protocol (e.g. in the reading zone) is provided by two sources. First, a 2D video camera is used (AXIS®, Model 215PTZ, 30 fps) to detect the location with respect to predefined zones.[11] In addition, a wearable camera is used (GoPro®, 30 fps) for detecting the location from recognised scene objects [92].
- Objects: In addition to object recognition from egocentric videos (wearable camera), information about the objects used during the protocol is provided by motion sensors (Wireless Sensor Tags®) attached to objects, e.g. to cups, that monitor and record motion events.
- Postures: The 2D video camera is also used for recognising the posture of the participant (e.g. sitting), using the same configuration as for location/zone detection [91].
- State of appliances: Energy consumption sensors (Plugwise®) are used for identifying the power state (on or off) of electronic appliances, e.g. the state of kettle during tea preparation. Both motion sensors and energy consumption sensors have a 10-s tolerance/timeout interval, before sending data to the system.

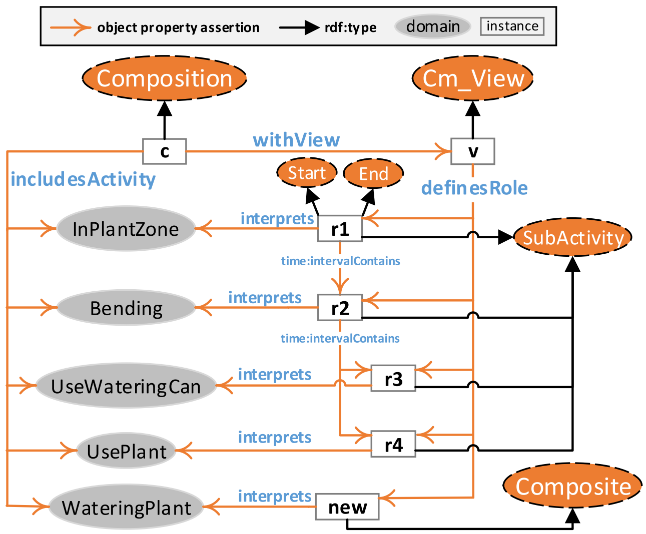
6.3. Deployment-specific patterns and rules

The interpretation semantics of the deployment involved instantiations of activity meta-knowledge patterns for the 10 IADLs of the protocol (see Table 1 for a complete list of the activities), as well as, extensions to the core pattern in Fig. 7 to define assessment queries for problem detection.

More specifically, the first step towards configuring MetaQ for the deployment was to define the OWL activity patterns that encode in a structured way the information relevant to the domain activities of the protocol. This translates to recognising the behaviour of the participant with respect to the protocol specifications, i.e. recognising the activities performed, the time required for each activity, the number of repetitions of a certain activity, etc. Consequently, the overall requirement is to augment the available set of observations of the installation by extracting the pieces of aggregated information that cannot be provided by means of the installation sensors and components alone.

[11] A variation of the framework presented in [91] is used.

a



b

```
CONSTRUCT {
  ?new a :WateringPlant;
    :hasStartTime ?st1;
    :hasEndTime ?et1;
    :hasSubActivity ?r1,?r2,?r3,?r4.
}
WHERE {
  ?r1 a InPlantZone;
    :hasStartTime ?st1;
    :hasEndTime ?et1.
  ?r2 a Bending;
    :hasStartTime ?st2;
    :hasEndTime ?et2.
  ?r3 a :UseWateringCan;
    :hasStartTime ?st3;
    :hasEndTime ?et3.
  ?r4 a :UsePlant;
    :hasStartTime ?st4;
    :hasEndTime ?et4.
  FILTER(contains(?st1,?et1,?st2,?et2)
    && contains(?st2,?et2,?st3,?et3)
    && contains(?st2, ?et2, ?st4, ?et4)) .
  BIND(URI(?r1, ?r2, ?r3, ?r4) as ?new) .
  NOT EXISTS {?new a [ ] .}
}
```

Fig. 11. (a) `WateringPlant` pattern instantiation, (b) the generated SPARQL query.

The activity patterns were derived after several iterations with the clinicians in order to clearly define when an activity should be considered successful or not in the experimentation context. For example, in order to recognise that the participant is making a phone call, clinicians suggested that the constituent activities "in phone zone", "use phone" and "phone object", need to be recognised by the available sensors and components. Based on the knowledge elicited from the clinicians, we defined 10 meta-knowledge activity patterns for aggregating and deriving the activities of the protocol. An example pattern is presented in Section 6.4 (Fig. 11(a)) that defines the constituent observations for the `WateringPlant` activity.

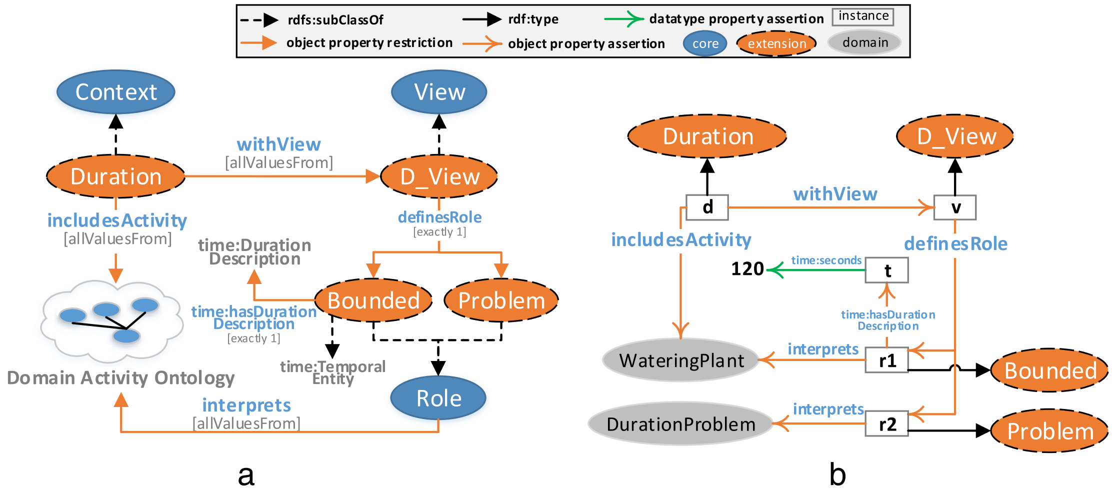
In addition, clinicians highlighted the fact that, apart from recognising protocol activities, the derivation of problematic situations would further support them in the diagnosis/assessment. Towards supporting this requirement, we further extended the core meta-knowledge pattern to model and detect protocol activities with long duration, which is one of the most important monitoring aspects suggested by the clinicians that indicates problematic behaviour. Fig. 12 depicts the extended pattern along with an instantiation for modelling domain knowledge regarding the average duration of the `WateringPlant` activity and the SPARQL rule that is generated to derive duration-related problems.

6.4. *Example interpretation semantics*

In the following, we present an example instantiation of the composition pattern to recognise the `WateringPlant` IADL and the extension of the core pattern to model and detect protocol activities with long duration.

WateringPlant composition semantics. The `WateringPlant` IADL involves the fusion of three primitive activities relevant to location (`InPlantZone`), posture (`Bending`) and objects used (`UsePlant`, `UseWateringCan`). Fig. 11(a) presents the instantiation of the composition pattern to model the `WateringPlant` activity. Intuitively, the pattern defines the watering plant activity as the situation where the person is inside the plant zone and interacts with relevant objects, i.e. the watering can and the plant, while they are bending. It is worth mentioning the fact that there is no direct temporal constraint among the instance that classifies the `InPlantZone` activity (`r1`) and the instances that classify the `UseWateringCan` (`r3`) and `UsePlant` (`r4`) activities. However, the `time:intervalContains` property is transitive and therefore, these implicit relations derive from the OWL 2 semantics, following the property paths `r1` $\rightarrow$ `r2` $\rightarrow$ `r3` and `r1` $\rightarrow$ `r2` $\rightarrow$ `r4`. This is an example of the additional inferences that can be drawn over the patterns, exploiting the formal representation and semantics of the contextual information. The generated query is depicted in Fig. 11(b).

Activity duration problems. Fig. 12(a) depicts the new pattern that has been introduced to formalise an upper limit on the duration of the protocol activities. Two additional roles have been introduced, namely `Bounded` and `Problem`, to interpret the activity of interest and the problem, respectively, reusing the OWL Time vocabulary for defining the duration context. The instantiation of the pattern for the `WateringPlant` activity is depicted in Fig. 12(b), where the duration limit of the activity was set by the clinicians to 2 min. The corresponding SPARQL query that is generated is depicted in Fig. 12(c). A new problem instance is asserted to the class that is interpreted by the `Problem` role, capturing the fact that the duration of the `WateringPlant` activity was longer than the expected. The problem instance is linked to the activity instance through the `abnormalActivity` property.




```
CONSTRUCT {
   ?new a :DurationProblem;
      :abnormalActivity ?r1.
}
WHERE{
   ?r1 a WateringPlant;
      :startTime ?st1;
      :endTime ?et1.
   FILTER(:durationSec(?st1, ?st2) > 120)
   BIND(URI(?r1) as ?new) .
   NOT EXISTS {?new a [ ] .}
}
```

c

Fig. 12. (a) The `Duration` pattern, (b) the `WateringPlant` instantiation, (c) the SPARQL query.

6.5. Results and discussion

MetaQ does not impose any restriction on the OWL 2 profile that can be used for modelling domain knowledge and the decision is mainly driven by the application requirements regarding expressivity and implementation. In our prototype implementation/deployment we have used the OWL 2 RL profile that adequately captures the semantics of the domain and the OWLIM [23] semantic repository that provides a highly scalable OWL 2 RL reasoner and a SPARQL query execution engine. The SPARQL-based interpretation procedure has been realised using the SPARQL Inferencing Notation (SPIN) [81].

Table 2 summarises the performance of MetaQ on a dataset of 19 participants, where True Positives (TP) is the number of IADLs correctly recognised, False Positives (FP) is the number of IADLs incorrectly recognised as performed, False Negatives (FN) is the number of IADLs that have not been recognised and True Negative (TN) is the number of IADLs that have not been performed by the participants. The $\mathcal{Q}$ value denotes the number of the sub-activities or classifiers that comprise the context of the IADL pattern, e.g. the `WateringPlant` activity in Fig. 11(a) is defined in terms of four sub-activities and therefore, $\mathcal{Q} = 4$. The True Positive Rate (TPR) and Positive Predicted Value (PPV) measures denote the recall and precision, respectively, and they are defined as:

$$TPR = \frac{TP}{TP + FN}, \qquad PPV = \frac{TP}{TP + FP}.$$

The activity recognition performance of MetaQ strongly depends on the $\mathcal{Q}$ value of the interpretation patterns, and consequently on the strictness of the SPARQL queries that are generated. More specifically, IADL patterns with low $\mathcal{Q}$ value, such as the "Leave the room", "Talk on phone" and "Write a check" activities, are characterised by high recall but relatively low precision. On the other hand, activity patterns with higher $\mathcal{Q}$ values, such as the "Prepare hot tea" and "Water the plant" activities, depict relatively low recall but demonstrate high precision. In the first case (low $\mathcal{Q}$ values), the SPARQL queries that are generated integrate a relatively small number of sub-activities. As a result, the probability to correctly capture the performed activity is high (high recall) but these queries are prone to false positives (low precision), e.g. due to noisy input. As the $\mathcal{Q}$ value increases, the SPARQL queries become more strict since they check and fuse a larger number of activities and therefore, they are more susceptible to false negatives (low recall), e.g. due to missing information. The precision, however, is high since the successful pattern matching of the strict `WHERE` clauses, e.g. the `WateringPlant` query in Fig. 11(b), is a

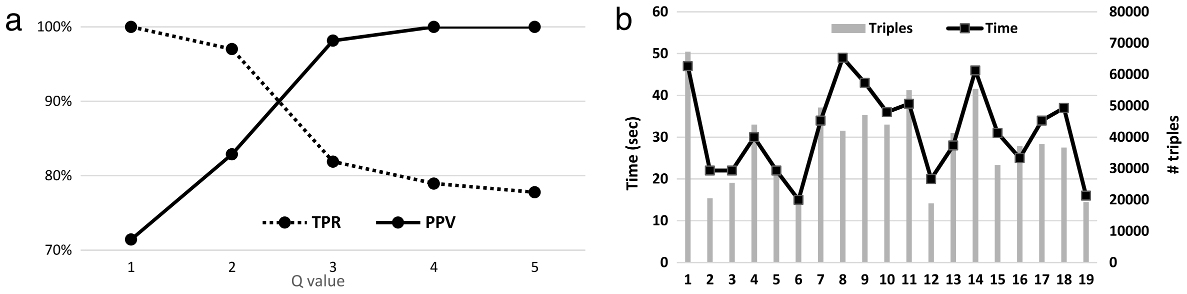


Fig. 13. (a) Average TPR and PPV of queries in relation to their $\mathcal{Q}$ value, (b) total query execution time in relation to the number of triples that is generated for each participant of the protocol.

Table 2
Activity interpretation results.

IADL	$\mathcal{Q}$	TP	FP	FN	TN	TPR%	PPV%
Establish account balance	2	14	3	1	5	93.33	82.35
Prepare drug box	3	17	0	2	4	89.47	100.00
Prepare hot tea	5	14	0	4	6	77.78	100.00
Search for bus line	2	14	4	0	5	100.00	77.78
Make a phone call	2	18	3	1	2	94.74	85.71
Watch TV	3	15	0	4	2	78.95	100.00
Water the plant	4	15	0	4	4	78.95	100.00
Write a check	2	12	2	0	7	100.00	85.71
Read an article	3	18	1	2	3	77.27	94.44
Leave the room	1	19	4	0	1	100.00	71.43

strong indication that the activity has been actually performed. Fig. 13(a) presents the average TPR and PPV of the queries used in our experiment according to their Q value.

MetaQ achieves an average TPR and PPV close to 89% on the dataset of the 19 participants, demonstrating the feasibility of the approach on real (small scale at the moment) settings. MetaQ, however, still has certain limitations that we consider them very important research directions in our future work. First, currently MetaQ cannot handle incomplete or missing information, since the activity meta-knowledge patterns are translated into strict SPARQL graph pattern conditions, e.g. sub-activities, that should be all satisfied by the underlying RDF graph. Second, the interpretation algorithm does not handle uncertainty and conflicts, assuming that all activities (primitive and high-level) have the same confidence (100%). To this end, we are currently investigating the use of Defeasible reasoning [77] and query rewriting techniques [93] to enhance the interpretation capabilities of MetaQ. However, despite the current limitations of the interpretation framework regarding the handling of imperfect information, MetaQ demonstrated very promising results, capitalising upon a hybrid combination of ontologies and SPARQL queries towards information fusion and providing advanced knowledge representation capabilities.

Fig. 13(b) depicts the time MetaQ needs to process the RDF datasets for the 19 participants of the protocol, along with the actual number of triples. As already mentioned, MetaQ is currently used in an offline mode, where the data of each participant is collected and processed after the execution of the protocol. Therefore, the time MetaQ needs to process the activities of a participant is not a critical requirement in the current deployment setting. In fact, MetaQ processed each dataset in less than a minute, which is in line with the runtime application requirements of the deployment. More specifically, the total duration of each patient visit in the hospital is about 2.5 h: 30 min for the medical consultation, 20 min for the protocol and a maximum of 1 h and 30 min for the clinical consultation with the neuropsychologist. At the end of the medical consultation, after having collected the clinical information, the analysis of the protocol data is initiated.

It is worth discussing at this point that the time MetaQ needs to apply the interpretation procedure on a dataset is not always proportional to the size of the dataset (number of triples). For example, MetaQ needs more time to process the dataset #8 that contains ∼42,000 triples than the dataset #7 that contains ∼50,000 triples. This is because participant #8 performed all the activities whose detection relies on queries with large $\mathcal{Q}$ value, such as "Prepare hot tea" and "Water the plant" and, therefore, the performance of the algorithm in Fig. 6 strongly depends on the execution of such complex queries. On the contrary, the participant #7 did not perform activities with high $\mathcal{Q}$ value and, therefore, the corresponding complex queries are not "active", resulting in faster query execution time.

7. Conclusion

The combination of ontologies and rules is a key prerequisite for effectively meeting the expressivity requirements when modelling and reasoning about context. In this paper, we presented MetaQ, a pragmatic approach towards the definition of a hybrid framework for complex activity recognition, combining the standard reasoning semantics of OWL 2 and the standard query language of the Semantic Web (SPARQL). Two activity interpretation query types have been presented, namely classification and composition that satisfy common interpretation requirements in activity recognition domains. Through the use of two high-level property assertions, the hybrid reasoning architecture of MetaQ allows the OWL 2 reasoning module to incorporate contextual information from the SPARQL-based reasoning module that cannot be otherwise derived by the semantics of OWL, such as temporal correlations of composite activities.

MetaQ also supports an activity pattern ontology that serves as a metamodel over domain activity classes, capturing the structural notions of atomic and compound activities through well-defined and interoperable activity models, based on the DnS pattern implementation in DUL. The aim is to allow the formal representation of activity interpretation models over activity classes that are generally characterised by intricate temporal associations, and where it is often the case that the aggregation of individual activities entails the existence of a new (composite) activity, notions that cannot be directly expressed in OWL. We also elaborated on the implementation of MetaQ that maps the semantics of the domain-dependent instantiations of the activity patterns on SPARQL `CONSTRUCT` graph patterns.

MetaQ has been deployed in a hospital for monitoring Alzheimer's disease patients, assisting clinicians to assess the condition of the patients through a goal-oriented protocol with instrumental activities of daily living (IADL). We evaluated the performance of the framework using a dataset of 19 participants, achieving very promising results. Our ongoing work focuses on enhancing MetaQ reasoning and fusion capabilities to handle uncertainty and imperfect information. Future work involves the further extension of the activity patterns to capture complex activities whose definition requires cardinality and negation information. In the near future, we also plan to provide an API for enabling users to define pattern instantiations without going into the details of the implementation of the activity patterns.

The real-time continuous human activity recognition is also a very important research direction of our work. The ability to fuse and recognise activities on a (near) real-time manner would definitely increase the potentials of our framework and would also provide additional, more practical deployment opportunities, e.g. in home settings. For example, in the healthcare domain, the ability to detect ongoing ADLs (e.g. when the person is cooking or showering) or even abnormal and critical situations, such as deviations from daily schedules or activities with longer duration than normal, would offer the possibility to provide timely assistance for caregivers, clinicians or even inhabitants themselves. Sliding windows and/or observation lifecycle management policies (e.g. event expiration policies, semantic complex event processing [94], RDF streams [95]) could be used to support real-time processing. In addition, many extensions to the SPARQL language have been proposed for working with temporal streaming data, such as in [96] that combines SPARQL and Prova rules,[12] C-SPARQL [49] and Streaming SPARQL [97]; the common underlying core rule language (SPARQL) allows for the seamless integration of such frameworks in MetaQ.

Acknowledgement

This work has been supported by the EU FP7 project Dem@Care: Dementia Ambient Care—Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support under contract No. 288199.

References

[1] D.J. Cook, J.C. Augusto, V.R. Jakkula, Ambient intelligence: technologies, applications, and opportunities, Pervasive Mob. Comput. 5 (4) (2009) 277–298.
[2] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, A survey of context modelling and reasoning techniques, Pervasive Mob. Comput. 6 (2) (2010) 161–180.
[3] T. Tiberghien, M. Mokhtari, H. Aloulou, J. Biswas, Semantic reasoning in context-aware assistive environments to support ageing with dementia, in: Proceedings of the 11th International Conference on The Semantic Web—Volume Part II, ISWC'12, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 212–227.
[4] D.L. McGuinness, F. van Harmelen, OWL Web ontology language overview, W3C Recommendation, World Wide Web Consortium, February 2004.
[5] B.C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, U. Sattler, OWL 2: the next step for OWL, Web Semant.: Sci. Serv. Agents World Wide Web 6 (4) (2008) 309–322.
[6] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, Y. Katz, Pellet: a practical OWL-DL reasoner, Web Semant.: Sci. Serv. Agents World Wide Web 5 (2) (2011) 51–53.
[7] V. Haarslev, 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, 2003, pp. 27–36.
[8] D. Riboni, C. Bettini, COSAR: hybrid reasoning for context-aware activity recognition, Pers. Ubiquitous Comput. 15 (3) (2011) 271–289.
[9] L. Chen, C.D. Nugent, Ontology-based activity recognition in intelligent pervasive environments, Int. J. Web Inf. Syst. 5 (4) (2009) 410–430.
[10] G. Okeyo, L. Chen, H. Wang, R. Sterritt, Dynamic sensor data segmentation for real-time knowledge-driven activity recognition, Pervasive Mob. Comput. 10, Part B (2014) 155–172.
[11] T. Eiter, G. Ianni, T. Krennwallner, A. Polleres, Rules and ontologies for the semantic Web, in: C. Baroglio, P. Bonatti, J. Małuszyński, M. Marchiori, A. Polleres, S. Schaffert (Eds.), Reasoning Web, in: Lecture Notes in Computer Science, vol. 5224, Springer, Berlin, Heidelberg, 2008, pp. 1–53.
[12] G. Okeyo, L. Chen, W. Hui, R. Sterritt, A hybrid ontological and temporal approach for composite activity modelling, in: G. Min, Y. Wu, L.C. Liu, X. Jin, S.A. Jarvis, A.Y. Al-Dubai (Eds.), TrustCom, IEEE Computer Society, 2012, pp. 1763–1770.
[13] M. Wessel, M. Luther, M. Wagner, The difference a day makes—recognizing important events in daily context logs, in: C&O:RR, 2007.
[14] G. Meditskos, N. Bassiliades, Rule-based OWL reasoning systems: implementations, strengths and weaknesses, in: Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches, IGI Global, ISBN: 978-1-60566-402-6, 2009.
[15] M. Horridge, P.F. Patel-Schneider, OWL 2 Web ontology language manchester syntax (second edition), W3C working group note 11 december 2012. URL: http://www.w3.org/TR/owl2-manchester-syntax/.
[16] H. Boley, M. Kifer, RIF basic logic dialect (second edition), W3C recommendation 5 february 2013. URL: http://www.w3.org/TR/rif-bld/.

12 https://prova.ws/.

[17] A. Gangemi, P. Mika, Understanding the semantic Web through descriptions and situations, in: Proceedings of the International Conference on Ontologies, Databases and Applications of Semantics, 2003, pp. 689–706.
[18] DOLCE+DnS Ultralite (DUL) ontology, URL: http://www.loa.istc.cnr.it/ontologies/DUL.owl.
[19] S. Harris, A. Seaborne, SPARQL 1.1 query language, W3C recommendation 21 march 2013. URL: http://www.w3.org/TR/sparql11-query/.
[20] N. Jekjantuk, G. Gröner, J. Pan, Modelling and reasoning in metamodelling enabled ontologies, in: Y. Bi, M.-A. Williams (Eds.), Knowledge Science, Engineering and Management, in: Lecture Notes in Computer Science, vol. 6291, Springer, Berlin, Heidelberg, 2010, pp. 51–62.
[21] J. Pérez, M. Arenas, C. Gutierrez, Semantics and complexity of SPARQL, ACM Trans. Database Syst. 34 (3) (2009) 16:1–16:45.
[22] M. Arias, J.D. Fernández, M.A. Martínez-Prieto, P. de la Fuente, An empirical study of real-world SPARQL queries, in: CoRR.
[23] B. Bishop, A. Kiryakov, D. Ognyanoff, I. Peikov, Z. Tashev, R. Velkov, OWLIM: a family of scalable semantic repositories, Semant. Web 2 (1) (2011) 33–42.
[24] G. Meditskos, S. Dasiopoulou, V. Efstathiou, I. Kompatsiaris, SP-ACT: a hybrid framework for complex activity recognition combining OWL and SPARQL rules, in: PerCom Workshops, 2013, pp. 25–30.
[25] G. Meditskos, S. Dasiopoulou, V. Efstathiou, I. Kompatsiaris, Ontology patterns for complex activity modelling, in: RuleML, 2013, pp. 144–157.
[26] L. Chen, C.D. Nugent, H. Wang, A knowledge-driven approach to activity recognition in smart homes, IEEE Trans. Knowl. Data Eng. 24 (6) (2012) 961–974.
[27] D. Riboni, C. Bettini, Context-aware activity recognition through a combination of ontological and statistical reasoning, in: Proceedings of the 6th International Conference on Ubiquitous Intelligence and Computing, UIC'09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 39–53.
[28] C. Golbreich, E. K. Wallace (Eds.), OWL 2 Web ontology language, new features and rationale (second edition), W3C recommendation 11 december 2012. URL: http://www.w3.org/TR/owl2-new-features/.
[29] D. Riboni, C. Bettini, OWL 2 modeling and reasoning with complex human activities, Pervasive Mob. Comput. 7 (3) (2011) 379–395.
[30] B. Motik, B. Cuenca Grau, U. Sattler, Structured objects in OWL: representation and reasoning, in: Proceedings of the 17th International Conference on World Wide Web, WWW'08, ACM, New York, NY, USA, 2008, pp. 555–564.
[31] M.Y. Vardi, Why is modal logic so robustly decidable? in: Proceedings of a DIMACS Workshop on Descriptive Complexity and Finite Models, in: DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1996, pp. 149–184.
[32] D. Riboni, L. Pareschi, L. Radaelli, C. Bettini, Is ontology-based activity recognition really effective?, in: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops, 2011, pp. 427–431.
[33] B. Motik, U. Sattler, R. Studer, Query answering for OWL-DL with rules, Web Semant. 3 (1) (2005) 41–60.
[34] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, SWRL: A semantic Web rule language combining OWL and RuleML, Tech. Rep, National Research Council of Canada, Network Inference, and Stanford University, 2004, May.
[35] C. Welty, R. Fikes, A reusable ontology for fluents in OWL, in: Proceedings of the 2006 Conference on Formal Ontology in Information Systems: Proceedings of the Fourth International Conference, (FOIS 2006), IOS Press, Amsterdam, The Netherlands, 2006, pp. 226–236. URL: http://dl.acm.org/citation.cfm?id=1566079.1566106.
[36] S. Batsakis, E.G.M. Petrakis, SOWL: a framework for handling spatio-temporal information in OWL 2.0, in: Proceedings of the 5th International Conference on Rule-based Reasoning, Programming, and Applications, RuleML'2011, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 242–249. URL: http://dl.acm.org/citation.cfm?id=2032787.2032812.
[37] B. Motik, U. Sattler, R. Studer, Query answering for OWL-DL with rules, in: S. McIlraith, D. Plexousakis, F. Harmelen (Eds.), The Semantic Web—ISWC 2004, in: Lecture Notes in Computer Science, vol. 3298, Springer, Berlin, Heidelberg, 2004, pp. 549–563.
[38] S. Zhang, P. McCullagh, C. Nugent, H. Zheng, An ontology-based context-aware approach for behaviour analysis, in: L. Chen, C.D. Nugent, J. Biswas, J. Hoey (Eds.), Activity Recognition in Pervasive Intelligent Environments, in: Atlantis Ambient and Pervasive Intelligence, vol. 4, Atlantis Press, 2011, pp. 127–148.
[39] S. Zhang, P. McCullagh, C. Nugent, H. Zheng, N. Black, An ontological approach for context-aware reminders in assisted living' behavior simulation, in: Proceedings of the 11th International Conference on Artificial Neural Networks Conference on Advances in Computational Intelligence—Volume Part II, IWANN'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 677–684.
[40] M.J. O'Connor, A.K. Das, SQWRL: a query language for OWL, in: OWLED, 2009.
[41] K. Maria, E. Vasilis, A. Grigoris, S-CRETA: smart classroom real-time assistance, in: P. Novais, K. Hallenborg, D.I. Tapia, J.M.C. Rodríguez (Eds.), Ambient Intelligence—Software and Applications, in: Advances in Intelligent and Soft Computing, vol. 153, Springer, Berlin, Heidelberg, 2012, pp. 67–74.
[42] V. Haarslev, R. Möller, M. Wessel, Querying the semantic Web with racer + nRQL, in: Proceedings of the KI-2004 International Workshop on Applications of Description Logics, ADL'04, 2004.
[43] B. McBride, Jena: implementing the RDF model and syntax specification, in: SemWeb, 2001.
[44] V. Haarslev, K. Hidde, R. Möller, M. Wessel, The RacerPro knowledge representation and reasoning system, Semant. Web 3 (3) (2012) 267–277.
[45] K. Wongpatikaseree, M. Ikeda, M. Buranarach, T. Supnithi, A. Lim, Y. Tan, Activity recognition using context-aware infrastructure ontology in smart home domain, in: 2012 Seventh International Conference on Knowledge, Information and Creativity Support Systems, KICSS, 2012, pp. 50–57.
[46] T. Patkos, I. Chrysakis, A. Bikakis, D. Plexousakis, G. Antoniou, A reasoning framework for ambient intelligence, in: Proceedings of the 6th Hellenic Conference on Artificial Intelligence: Theories, Models and Applications, SETN'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 213–222.
[47] D. Anicic, S. Rudolph, P. Fodor, N. Stojanovic, Stream reasoning and complex event processing in etalis, Semant. Web 3 (4) (2012) 397–407.
[48] M. Grüninger, Ontology repositories make a world of difference, in: Proceedings of the 7th International Conference on Theory, Practice, and Applications of Rules on the Web, RuleML'13, Springer-Verlag, Berlin, Heidelberg, 2013, pp. 1–12.
[49] D.F. Barbieri, D. Braga, S. Ceri, E.D. Valle, M. Grossniklaus, Querying RDF streams with C-SPARQL, SIGMOD Rec. 39 (1) (2010) 20–26.
[50] A. Bolles, M. Grawunder, 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, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 448–462.
[51] J. Tappolet, 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, Springer-Verlag, Heraklion, Crete, Greece, 2009, pp. 308–322.
[52] B. Glimm, S. Rudolph, J. Völker, Integrated metamodeling and diagnosis in OWL 2, in: Proceedings of the 9th International Semantic Web Conference on The Semantic Web—Volume Part I, ISWC'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 257–272.
[53] N. Jekjantuk, G. Gröner, J.Z. Pan, Modelling and reasoning in metamodelling enabled ontologies, in: Proceedings of the 4th International Conference on Knowledge Science, Engineering and Management, KSEM'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 51–62.
[54] G. Okeyo, L. Chen, H. Wang, R. Sterritt, Ontology-based learning framework for activity assistance in an adaptive smart home, in: L. Chen, C.D. Nugent, J. Biswas, J. Hoey (Eds.), Activity Recognition in Pervasive Intelligent Environments, in: Atlantis Ambient and Pervasive Intelligence, vol. 4, Atlantis Press, 2011, pp. 237–263.
[55] K.A.P. Ngoc, Y.-K. Lee, S.-Y. Lee, OWL-based user preference and behavior routine ontology for ubiquitous system, in: Proceedings of the 2005 OTM Confederated International Conference on On the Move to Meaningful Internet Systems: CoopIS, COA, and ODBASE—Volume Part II, OTM'05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 1615–1622.
[56] I. Mocanu, A.M. Florea, A model for activity recognition and emergency detection in smart environments, in: 1st International Conference on Ambient Computing, Applications, Services and Technologies, AMBIENT, 2011, pp. 13–19.
[57] A. Scherp, T. Franz, C. Saathoff, S. Staab, F–a model of events based on the foundational ontology Dolce+DnS Ultralight, in: Proceedings of the Fifth International Conference on Knowledge Capture, K-CAP'09, ACM, New York, NY, USA, 2009, pp. 137–144.
[58] J. Ye, G. Stevenson, S. Dobson, A top-level ontology for smart environments, Pervasive Mob. Comput. 7 (3) (2011) 359–378.
[59] W. May, J.J. Alferes, R. Amador, An ontology- and resources-based approach to evolution and reactivity in the semantic Web, in: OTM Conferences (2), 2005, pp. 1553–1570.
[60] V. Beltran, K. Arabshian, H. Schulzrinne, Ontology-based user-defined rules and context-aware service composition system, in: ESWC Workshops, 2011, pp. 139–155.

[61] J. Debattista, S. Scerri, I. Rivera, S. Handschuh, Ontology-based rules for recommender systems, in: SeRSy, 2012, pp. 49–60.
[62] G. Stoilos, G. Stamou, J.Z. Pan, V. Tzouvaras, I. Horrocks, Reasoning with very expressive fuzzy description logics, J. Artificial Intelligence Res. 30 (1) (2007) 273–320.
[63] U. Straccia, Towards a fuzzy description logic for the semantic Web (preliminary report), in: The Semantic Web: Research and Applications, Springer, 2005, pp. 167–181.
[64] F. Bobillo, U. Straccia, Fuzzy ontology representation using OWL 2, Int. J. Approx. Reason. 52 (7) (2011) 1073–1094.
[65] G. Stoilos, G. Stamou, J.Z. Pan, Fuzzy extensions of OWL: Logical properties and reduction to fuzzy description logics, Int. J. Approx. Reason. 51 (6) (2010) 656–679.
[66] T.W. Wlodarczyk, C. Rong, M. O'Connor, 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, ACM, New York, NY, USA, 2011, pp. 39:1–39:9.
[67] R.N. Carvalho, K.B. Laskey, P.C.G. Costa, PR-OWL 2.0—bridging the gap to OWL semantics, in: Uncertainty Reasoning for the Semantic Web II, in: Lecture Notes in Computer Science, 2013, pp. 1–18.
[68] P.C. Costa, K.B. Laskey, K.J. Laskey, PR-OWL: a Bayesian ontology language for the semantic Web, in: Uncertainty Reasoning for the Semantic Web I, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 88–107 (Chapter).
[69] Z. Ding, Y. Peng, R. Pan, BayesOwl: uncertainty modeling in semantic Web ontologies, in: Z. Ma (Ed.), Soft Computing in Ontologies and Semantic Web, in: Studies in Fuzziness and Soft Computing, vol. 204, Springer, Berlin, Heidelberg, 2006, pp. 3–29.
[70] U. Straccia, Managing uncertainty and vagueness in description logics, logic programs and description logic programs, in: Reasoning Web, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 54–103 (Chapter).
[71] M. Vacura, V. Svátek, 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, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 227–234.
[72] R. Dividino, S. Sizov, S. Staab, B. Schueler, Querying for provenance, trust, uncertainty and other meta knowledge in RDF, Web Semant. 7 (3) (2009) 204–219.
[73] A. Ciaramella, M.G.C.A. Cimino, F. Marcelloni, 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, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 31–45.
[74] R. Helaoui, D. Riboni, H. Stuckenschmidt, A probabilistic ontological framework for the recognition of multilevel human activities, in: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp'13, ACM, New York, NY, USA, 2013, pp. 345–354.
[75] N.D. Rodríguez, M.P. Cuéllar, J. Lilius, M.D. Calvo-Flores, A fuzzy ontology for semantic modelling and recognition of human behaviour, Knowl.-Based Syst. 66 (2014) 46–60.
[76] J. Cheng, Z.M. Ma, L. Yan, f-SPARQL: a flexible extension of SPARQL, in: Proceedings of the 21st International Conference on Database and Expert Systems Applications: Part I, DEXA'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 487–494.
[77] F. Maier, D. Nute, Well-founded semantics for defeasible logic, Synthese 176 (2) (2010) 243–274.
[78] W.R. van Hage, V. Malaisé, R. Segers, L. Hollink, G. Schreiber, Design and use of the Simple Event Model (SEM), J. Web Semant. 9 (2) (2011) 128–136.
[79] G. Stevenson, S. Knox, S. Dobson, 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, ACM, New York, NY, USA, 2009, pp. 9:1–9:8.
[80] J.F. Allen, Towards a general theory of action and time, Artif. Intell. 23 (2) (1984) 123–154.
[81] H. Knublauch, J.A. Hendler, K. Idehen, SPIN—overview and motivation, W3C member submission, World Wide Web Consortium, February 2011.
[82] M. Schmidt, M. Meier, G. Lausen, Foundations of sparql query optimization, in: Proceedings of the 13th International Conference on Database Theory, ICDT'10, ACM, New York, NY, USA, 2010, pp. 4–33.
[83] A. Scherp, T. Franz, C. Saathoff, S. Staab, A core ontology on events for representing occurrences in the real world, Multimedia Tools Appl. 58 (2) (2012) 293–331.
[84] F. Pan, J.R. Hobbs, Time ontology in OWL, W3C working draft, W3C, http://www.w3.org/TR/2006/WD-owl-time-20060927/ (September 2006).
[85] The unified Parkinson's disease rating scale (UPDRS): Status and recommendations, Mov. Disorders 18 (7) (2003) 738–750.
[86] M.B. First, Diagnostic and Statistical Manual of Mental Disorders, DSM IV, fourth ed., APA, 1994, pp. 97–327.
[87] M.S. Albert, S.T. DeKosky, D. Dickson, B. Dubois, H.H. Feldman, N.C. Fox, A. Gamst, D.M. Holtzman, W.J. Jagust, R.C. Petersen, P.J. Snyder, M.C. Carrillo, B. Thies, C.H. Phelps, The diagnosis of mild cognitive impairment due to alzheimer's disease: recommendations from the national institute on ageing-alzheimer's association workgroups on diagnostic guidelines for alzheimer's disease, Alzheimer's Dement. 7 (3) (2011) 270–279.
[88] B. Dubois, H.H. Feldman, C. Jacova, J.L. Cummings, S.T. DeKosky, P. Barberger-Gateau, A. Delacourte, G. Frisoni, N.C. Fox, D. Galasko, S. Gauthier, H. Hampel, G.A. Jicha, K. Meguro, J. O'Brien, F. Pasquier, P. Robert, M. Rossor, S. Salloway, M. Sarazin, L.C. de Souza, Y. Stern, P.J. Visser, P. Scheltens, Revising the definition of alzheimer's disease: a new lexicon, Lancet Neurol. 9 (11) (2010) 1118–1127.
[89] B. Dubois, H.H. Feldman, C. Jacova, S.T. DeKosky, P. Barberger-Gateau, J. Cummings, A. Delacourte, D. Galasko, S. Gauthier, G. Jicha, K. Meguro, J. O'Brien, F. Pasquier, P. Robert, M. Rossor, S. Salloway, Y. Stern, P.J. Visser, P. Scheltens, Research criteria for the diagnosis of Alzheimer's disease: revising the NINCDS-ADRDA criteria, Lancet Neurol. 6 (8) (2007) 734–746.
[90] M.F. Folstein, L.N. Robins, J.E. Helzer, The mini-mental state examination, Arch. Gen. Psychiatry 40 (7) (1983) 812.
[91] B. Fosty, C.F. Crispim-Junior, J. Badie, F. Bremond, M. Thonnat, Event recognition system for older people monitoring using an RGB-D camera, in: ASROB—Workshop on Assistance and Service Robotics in a Human Environment, Tokyo, Japan, 2013.
[92] D. Szolgay, J. Benois-Pineau, R. Mégret, Y. Gaëstel, J.-F. Dartigues, Detection of moving foreground objects in videos with strong camera motion, Pattern Anal. Appl. 14 (3) (2011) 311–328.
[93] T. Venetis, G. Stoilos, G.B. Stamou, Query extensions and incremental query rewriting for OWL 2 QL ontologies, J. Data Semant. 3 (1) (2014) 1–23.
[94] K. Teymourian, A. Paschke, Enabling knowledge-based complex event processing, in: Proceedings of the 2010 EDBT/ICDT Workshops, EDBT'10, ACM, New York, NY, USA, 2010, pp. 37:1–37:7. http://dx.doi.org/10.1145/1754239.1754281. URL: http://doi.acm.org/10.1145/1754239.1754281.
[95] D. Anicic, P. Fodor, S. Rudolph, 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, ACM, New York, NY, USA, 2011, pp. 635–644.
[96] K. Teymourian, M. Rohde, A. Paschke, Fusion of background knowledge and streams of events, in: Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems, DEBS'12, ACM, New York, NY, USA, 2012, pp. 302–313.
[97] A. Bolles, M. Grawunder, 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, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 448–462.