Knowledge-driven Activity Recognition and Segmentation Using Context Connections

3 authors:

Georgios Meditskos
The Centre for Research and Technology, Hellas

Efstratios Kontopoulos
The Centre for Research and Technology, Hellas

Ioannis (Yiannis) Kompatsiaris
The Centre for Research and Technology, Hellas

Knowledge-Driven Activity Recognition and Segmentation Using Context Connections

Georgios Meditskos, Efstratios Kontopoulos, and Ioannis Kompatsiaris

Information Technologies Institute
Centre for Research & Technology - Hellas
6^{th} km Xarilaou - Thermi, 57001, Thessaloniki, Greece
{gmeditsk,skontopo,ikom}@iti.gr

Abstract. We propose a knowledge-driven activity recognition and segmentation framework introducing the notion of context connections. Given an RDF dataset of primitive observations, our aim is to identify, link and classify meaningful contexts that signify the presence of complex activities, coupling background knowledge pertinent to generic contextual dependencies among activities. To this end, we use the Situation concept of the DOLCE+DnS Ultralite (DUL) ontology to formally capture the context of high-level activities. Moreover, we use context similarity measures to handle the intrinsic characteristics of pervasive environments in real-world conditions, such as missing information, temporal inaccuracies or activities that can be performed in several ways. We illustrate the performance of the proposed framework through its deployment in a hospital for monitoring activities of Alzheimer's disease patients.

Keywords: ontologies, activity recognition, segmentation, context.

1 Introduction

In recent years, the demand for intelligent, customized user task support has proliferated across a multitude of application domains, ranging from smart spaces and healthcare [30] to transportation and energy control [10]. The key challenge in such applications is to abstract and fuse the captured *context* in order to elicit a higher level understanding of the situation. Towards this direction, a growing body of research has been investigating ontology-based (knowledge-driven) frameworks for modelling and reasoning about context [4], [5], [7]. The idea is to map low-level information (e.g. objects used, postures, location) and activity models onto ontologies, enabling the inference of high-level activities using domain knowledge and ontology reasoning. In many cases, activity recognition is further augmented with rules [12] for representing richer relationships not supported by the standard ontology semantics, like e.g. structured (composite) activities [19].

A significant challenge in activity recognition is the ability to identify and recognise the context signifying the presence of complex activities. Time windows [21], [9] and slices [24], [20], background knowledge about the order [27], [23]

P. Mika et al. (Eds.) ISWC 2014, Part II, LNCS 8797, pp. 260–275, 2014.

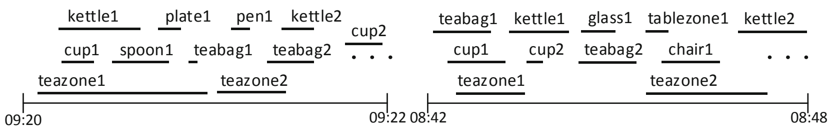


Fig. 1. Example observations detected during tea preparation

or duration [32] of activities constitute commonly used approaches in knowledge-driven activity recognition. Such approaches, however, define strict contextual dependencies, e.g. start/end activities or maximum activity duration, assuming that all the information is available. Thus, they fail to capture many intrinsic characteristics of pervasive environments in real-world conditions, such as imperfect information, noise or inaccurate temporal correlations.

An important factor to take into consideration is that contextual information is typically collected by multiple sensors and complementary sensing modalities. Each modality generates information from a different perspective, but by combining them together we are in a position to infer far more about a person's activities than by any sensor alone. Thus, a further challenge is to effectively fuse multiple sources of heterogeneous, noisy and potentially inconsistent information in a way that provides accurate and useful outputs.

In order to better highlight the challenges, consider the two example time-lines in Fig. 1 that contain information regarding the location and the objects that a person interacts with when preparing hot tea. These are subsets of real-world data obtained by monitoring Alzheimer's disease patients in the FP7 project Dem@Care[1]. As illustrated in the figure:

– The duration of activities usually varies, even when they are performed by the same person. The use of time windows, slices or background knowledge regarding activity duration fails to capture this characteristic, or, at least, the segmentation task becomes too complex.
– Many activities are carried out differently even by the same person. Thus, the use of strictly structured background knowledge relevant to the order of activities or their temporal boundaries is not always practical and flexible.
– The information integrated from heterogeneous sources is intrinsically noisy, incomplete, with inaccurate temporal correlations. For example, the cup2 and teabag2 observations in the second time-line in Fig. 1 do not coincide with information regarding the location of the person. Such information cannot be processed by patterns that, e.g., explicitly enumerate the sub-activities and temporal relations involved in a complex behaviour.

Towards addressing these intrinsic challenges of pervasive applications, this paper presents a practical ontology-based activity recognition framework. The framework detects complex activities in multi-sensor fusion environments based

[1] Dementia Ambient Care: Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support, `http://http://www.demcare.eu/`

on loosely coupled domain activity dependencies rather than on strict contextual constraints. More specifically, given an RDF dataset of primitive activities (observations), we define a procedure for assigning *context connections*, i.e. links among relevant groups of observations that signify the presence of complex activities. The connections are determined by semantically comparing *local contexts*, i.e. the type and number of neighbouring observations, against *context descriptors*, i.e. background knowledge about domain activity dependencies. We formalise these descriptors by capitalising on the Situation concept of the DnS pattern [13] of the DOLCE+DnS Ultralite (DUL) [11] ontology, exploiting the OWL 2 meta-modelling capabilities (punning [15]) for defining generic relations among classes. As a result, this paper features the following contributions:

- We deliver a flexible and reusable framework for recognising complex activities that is applicable to a wide range of activity recognition scenarios.
- We propose a simple and reusable ontology pattern for capturing common sense background knowledge regarding domain activity dependencies.
- We present a context-aware activity recognition and segmentation algorithm that incorporates the level of relaxation needed during context classification.

In this work we consider non-interleaving human activities, i.e. only one activity can be performed each time. Moreover, we focus on offline activity recognition. We consider the support of interleaved and concurrent activities, as well as, the real-time continuous activity recognition as very important research directions of our ongoing work. On the other hand, we believe our approach is quite suitable for further research purposes, e.g. towards extracting behaviour patterns and detecting behaviour changes, including the manner in which activities are performed, idiosyncratic and habitual knowledge, as well as recurrent routines. We further elaborate on this direction in Section 5.

The rest of the paper is structured as follows: Section 2 reviews relevant ontology-based activity recognition frameworks. Section 3 describes the ontology pattern we use to associate high-level activities with generic context descriptors, whereas the algorithms for segmentation and activity recognition are described in Section 4. Section 5 presents the results of the evaluation of our approach on real-world data. Conclusions and future directions are presented in Section 6.

2 Related Work

Ontologies have been gaining increasing attention as a means for modelling and reasoning over contextual information and, particularly, human activities. Under this paradigm, OWL is used for describing the elements of interest (e.g. events, activities, location), their pertinent logical associations, as well as the background knowledge required to infer additional information. For example, a 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 (TBox) as:

$$MakeTea \equiv \mathit{Activity\ and}\ (\mathit{actor\ only}\ (Person\ \mathit{and}\ (\mathit{uses\ some}\ TeaBag)\\ \mathit{and}\ (\mathit{uses\ some}\ Cup)\ \mathit{and}\ (\mathit{uses\ some}\ Kettle)\ \mathit{and}\ (\mathit{in\ some}\ Kitchen)))$$

In such cases, the data (ABox) needs to be segmented into chunks of activities to allow complex activities to be derived using standard OWL reasoning. However, the issue of data segmentation in ontology-based activity recognition has received little attention. For instance, in [8] and [26], situations correspond to OWL individuals, while DL reasoning [2] is used for determining which contextual concepts a specific situation falls into. However, no details are provided with respect to the method used for segmenting the data. In [25] statistical inferencing is used for segmenting the data and for predicting the most probable activities, whereas symbolic DL reasoning is applied for further refining the results.

The most common approach for ontology-based activity segmentation involves the use of time windows and slices. In [9] and [21] dynamically sliding time windows are used for activity recognition. The activities are specified as sequences of user-object interactions, whereas subsumption reasoning is used for inferring the ongoing activity. In [20], time slices are used for grouping activities and inferring complex situations. In [24], one-minute fixed time slices are used for activity recognition, using notions such as *recently used* and *second last activity*. In most cases, though, such approaches require prior domain knowledge, such as maximum duration of activities or window length, which results in inflexible and difficult to implement approaches, even if they can be dynamically adjusted.

In [32], an approach is presented for the recognition of multi-user concurrent activities where each activity is constrained by necessary conditions, e.g. activity "prepare breakfast" should occur in the morning and activity "bath" should last more than 5 minutes. In [23], an ontology is used for capturing atomic and compound events and for defining operators among event sets (e.g. sequence, concurrency). In addition, many approaches combine ontologies with rules [20], [31], [33], CEP patterns [18], [28] and SPARQL queries [29], [1], [3], [17]. The main limitation of these approaches is that they encapsulate strict activity patterns that cannot provide enough flexibility for handling the imprecise and ambiguous nature of real-world events in multi-sensor fusion environments.

The authors in [27] conceive activity recognition as a plan recognition problem. An activity plan consists of a partially ordered sequence of temporal constraints and actions that must be carried out in order to achieve the activity's goals. Though relevant to environments where the order of observations is accurate, plans fall short when more intricate activity patterns are involved, e.g. when fusing multi-sensor data with inherent temporal incoherences.

Our work has been mainly inspired by [22] and [14]. In [22], a data-driven approach is described for human activity recognition based on the order of object usage. The authors use web mining for extracting the objects most relevant to specific activities and each activity is then associated with a key object. The limitation is that each activity in the list is assumed to have a unique key object. Moreover, the proposed algorithms handle only sequential traces without overlaps. In our work, we follow a more formal and flexible approach, defining the activities relevant to high-level situations in terms of an ontology, without needing to specify key objects. Moreover, the recognition algorithms take into account the type and number of overlapped activities.

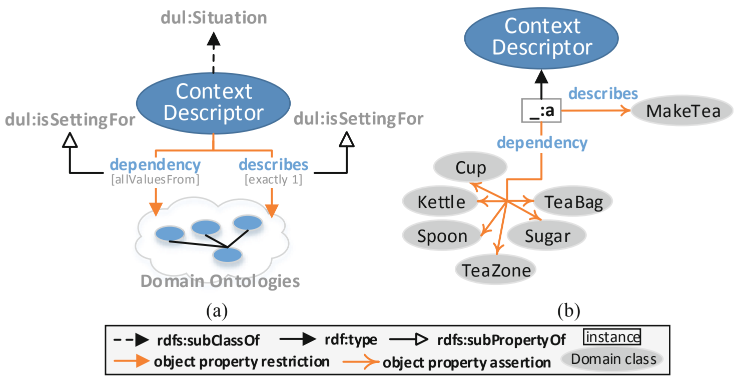


Fig. 2. (a) The `ContextDescriptor` class, (b) Example annotation of the `MakeTea` domain activity

In [14] a similarity-based segmentation approach is proposed. Generic activity models are built that are specific to the deployment environment and contain sensor activation sets for each complex activity. The segmentation and recognition are based on similarity measures for comparing timely ordered sensor sequences and sensor models of activities. The key difference of our work is that we use an ontology for modelling common sense high-level knowledge about activity dependencies, which also allows us to incorporate hierarchical relationships in context classification. Moreover, similarly to [22], instead of using window thresholds for analysing input data, we examine neighbouring events.

3 Domain Context Descriptors

In order to describe the context pertinent to each high-level activity in an abstract yet formal way, we reuse the *Situation* concept of the Descriptions and Situations (DnS) [13] pattern of DOLCE+DnS Ultralite (DUL) [11]. The aim is to provide the conceptual model for annotating domain activity classes with lower-level observation types. Fig. 2 (a) shows the specialisation of the *Situation* class, along with two sub-properties of the $isSettingFor$ upper-level property.

Our aim is to define relations among classes, therefore, the proposed ontology treats classes as instances, allowing property assertions to be made among domain concepts. Intuitively, the ontology can be thought of as a conceptual (meta) layer that can be placed on top of any domain activity ontology. This way, instances of the $ContextDescriptor$ are used to link domain activities (*describes* property) with one or more lower-level conceptualisations through *dependency* property assertions. Fig. 2 (b) presents an example of annotating class $MakeTea$ with class types relevant to objects (e.g *Cup*) and location (e.g. $TeaZone$).

The model also allows annotated classes to inherit the context dependencies of the superclasses through the following property chain axiom:

$$describes \circ subClassOf \circ isDescribedBy \circ dependency \sqsubseteq dependency$$

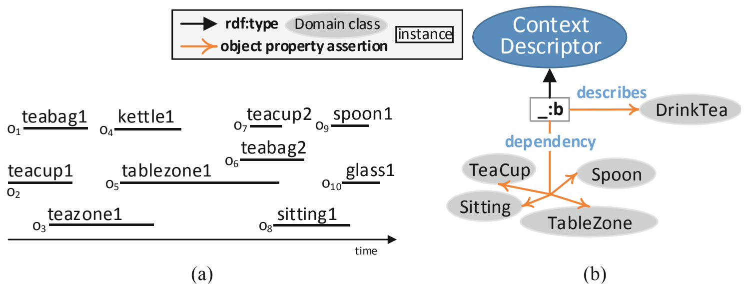


Fig. 3. (a) Example observations relevant to making and drinking tea, (b) The context descriptor of $DrinkTea$

In the rest of the paper, we use the term "context descriptor" to refer to the set of classes, denoted as d_C, that a domain activity C has been annotated with. For example, the context descriptor of $MakeTea$ is denoted as $d_{MakeTea}$ and is equal to the set $\{Cup$, $Kettle$, $Spoon$, $TeaZone$, $Sugar$, $TeaBag\}$.

4 Segmentation and Activity Recognition

Given a set $O = \{o_1, o_2, ..., o_n\}$ with RDF instances representing low-level observations, e.g. objects, locations, postures, etc., and a set of domain context descriptors $D = \{d_{C_1}, d_{C_2}, ..., d_{C_k}\}$, we describe in this section the steps involved in identifying meaningful contexts in O for recognizing higher level activities. We use Fig. 3 (a) as a running example that involves observations relevant to making and drinking tea, while the corresponding context descriptors are depicted in Fig. 2 (b) and Fig. 3 (b), respectively.

4.1 Local Contexts

The first step of the segmentation algorithm is to define the *local contexts* for each observation $o_i \in O$ that capture information relevant to the neighbouring observations of o_i and the most plausible activities that o_i can be part of. More specifically, let N_i^r be the set of observations o_j in the neighbourhood of o_i that either overlap with o_i $(o_i \circ o_j)$ or are the r-nearest to o_i $(n(o_i, o_j) \leq r)$, based on their temporal ordering. Moreover, let $T(o_i)$ be the most specific class of o_i, D the set of domain context descriptors and φ a local context similarity function.

Definition 1. *A local context l_i of an observation $o_i \in O$ is defined as the tuple $\langle o_i, N_i^r, C\rangle$, where $N_i^r = \{o_j \mid \forall o_j \in O, o_i \circ o_j \vee n(o_i, o_j) \leq r\}$ and C is the high-level class of the most plausible classification of l_i, such that $\nexists d_A \in D$: $\varphi(N_{i,T}^r, d_A) > \varphi(N_{i,T}^r, d_C)$, where $d_C \in D, d_C \neq d_A$ and $N_{i,T}^r = \{t \mid \forall o_j \in N_i^r, t = T(o_j)\}$.*

The class C denotes the most plausible domain activity classification of l_i, derived by computing the φ similarity between the set with the most specific observation classes $N^r_{i,T}$ and the domain context descriptors $d_k \in D$. The sets $N^r_{i,T}$ are represented as multisets (duplicates are allowed), since the number of observations with similar class types in the neighbourhood of o_i is important.

Local Context Similarity φ. The φ measure captures the similarity between the multiset N^r_T of a local context against the context descriptor set d_C of a class C. It is defined as

$$\varphi(N^r_T, d_C) = \frac{\displaystyle\sum_{\forall n \in N^r_T} \max_{\forall c \in d_C} \left[\delta(n, c)\right]}{|N^r_T|} \tag{1}$$

where N^r_T is the multiset with neighbouring observation class types and d_C is the context descriptor of C. φ is computed as the mean value of the maximum δ similarities for each concept $n \in N^r_T$, since each n may have more than one relevant concepts in d_C. Intuitively, φ captures the local plausibility of an observation o_i to be part of a complex activity C. If $\varphi = 1$, then all the classes in N^r_T appear in d_C and, therefore, it is very likely that the corresponding local context is part of the complex activity C.

Equation (1) uses the δ function that computes the similarity of a neighbouring observation class $n \in N^r_T$ against a context descriptor class $c \in d_C$ as

$$\delta(n, c) = \begin{cases} 1, & \text{if } n \sqsubseteq c \text{ (includes } n \equiv c) \\ \frac{|U(n) \cap U(c)|}{|U(n)|}, & \text{if } c \sqsubseteq n \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $U(C)$ is the set of the superclasses of C, excluding the $Thing$ concept, such that $U(C) = \{A \mid C \sqsubseteq A, A \neq \top\}$. Intuitively, an observation class n in the neighbourhood of o_i exactly matches a class c in the context descriptor set d_C, if it is equivalent to or a subclass of c. In this case, n is subsumed by c and, thus, fully satisfies the contextual dependency imposed by d_C that there should be at least one observation of type c. On the other hand, if c is subsumed by n ($c \sqsubseteq n$), then n is a more general concept than the one required by the context descriptor and the similarity is computed based on the rate of the superclasses of n that are also superclasses of c. For example, if *Spoon* is a direct subclass of *Cutlery* ($Spoon \sqsubseteq Cutlery$), $n = Spoon$ and $c = Cutlery$, then $\delta(Spoon, Cutlery) = 1$, since *Spoon* is subsumed by *Cutlery*. If $n = Cutlery$ and $c = Spoon$, then $\delta(Cutlery, Spoon) < 1$, depending on their superclasses.

Creating Local Contexts. Algorithm 1 describes the procedure for creating set L with the most plausible local contexts for each $o_i \in O$. The algorithm begins by defining set N^r_i with the neighbour observations of o_i (line 3). Then, the partial context set P_i is created as the multiset of the most specific class types of the observations in N^r_i (line 4). The algorithm then computes the φ similarity S_k of P_i against each context descriptor d_{C_k}, creating the set G_i with tuples

Algorithm 1. Creating local contexts

Data: Observations: $O = \{o_1, o_2, ..., o_i\}$, Domain context descriptors: $D = \{d_{C_1}, d_{C_2}, ..., d_{C_k}\}$, Nearest observations threshold: r.
Result: The set L with the most plausible local contexts.

1 $L \leftarrow \emptyset$;
2 **foreach** $o_i \in O$ **do**
3     $N_i^r = \{o_j \mid \forall o_j \in O, o_i \circ o_j \vee n(o_i, o_j) \leq r\}$;
4     $P_i \leftarrow \{t \mid \forall o_j \in N_i^r, t = T(o_j)\}$;
5     $G_i \leftarrow \emptyset$;
6   **foreach** $d_{C_k} \in D$ **do**
7   **if** $\exists A \in d_{C_k}, T(o_i) \sqsubseteq A$ **then** $G_i \leftarrow G_i \cup \{\langle C_k, \varphi(P_i, d_{C_k})\rangle\}, \varphi \neq 0$
8   **forall the** $\langle C_k, S_k\rangle \in G_i$ *with the max* S_k **do**
9         $L \leftarrow L \cup \{\langle o_i, N_i^r, C_k\rangle\}$;

of the form $\langle C_k, S_k\rangle$ (lines 5 to 7). If the class type of o_i does not semantically belong to class descriptor d_{C_k}, then the corresponding similarity tuple is omitted (line 7), ignoring noisy observations. Finally, a tuple $\langle o_i, N_i^r, C_k\rangle$ is created for all $\langle C_k, S_k\rangle$ with the maximum similarity in G_i and inserted into L. Note that G_i may contain more than one $\langle C_k, S_k\rangle$ tuples with the maximum similarity, and, therefore, more than one local contexts can be generated for o_i.

Example. We describe the definition of the local context for *teacup*2 (o_7) in Fig. 3 (a), using $r = 1$. Observations o_5, o_6 and o_8 overlap with o_7, whereas o_9 and o_4 are the 1-nearest to o_7. Thus, $N_7^1 = \{o_7, o_5, o_6, o_8, o_9, o_4\}$ (line 3) and $P_7 = \{TeaCup, TableZone, TeaBag, Sitting, Spoon, Kettle\}$ (line 4). According to Figs. 2 (b) and 3 (b), the context descriptor set is $D = \{d_{MakeTea}, d_{DrinkTea}\}$, where $d_{MakeTea} = \{TeaCup, Kettle, Spoon, TeaZone, Sugar, TeaBag\}$ and $d_{DrinkTea} = \{TeaCup, Sitting, TableZone, Spoon\}$. The class type for o_7 is $TeaCup$ that exists in both context descriptors, therefore φ will be computed for both of them. According to (1), $\varphi(P_7, d_{MakeTea}) = \frac{1+0+1+0+1+1}{6} = 0.66$ and $\varphi(P_7, d_{DrinkTea}) = \frac{1+1+0+1+1+0}{6} = 0.66$ (assuming that there are no hierarchical relationships among the domain class types). Thus, there are two local contexts for o_7 with maximum plausibility 0.66: $l_7 = \langle o_7, N_7^1, MakeTea\rangle$ and $l_7' = \langle o_7, N_7^1, DrinkTea\rangle$. Similarly, we have the following local contexts (φ similarity is also depicted for completeness): $l_1 = \langle o_1, N_1^1, MakeTea\rangle^{1.0}$, $l_2 = \langle o_2, N_2^1, MakeTea\rangle^{1.0}$, $l_3 = \langle o_3, N_3^1, MakeTea\rangle^{0.83}$, $l_4 = \langle o_4, N_4^1, MakeTea\rangle^{0.75}$, $l_5 = \langle o_5, N_5^1, DrinkTea\rangle^{0.5}$, $l_6 = \langle o_6, N_6^1, MakeTea\rangle^{0.66}$, $l_8 = \langle o_8, N_8^1, DrinkTea\rangle^{0.57}$, $l_9 = \langle o_9, N_9^1, MakeTea\rangle^{0.5}$, $l_9' = \langle o_9, N_9^1, DrinkTea\rangle^{0.5}$, l_{10} = -.

4.2 Context Connections

Based on the local contexts obtained in the previous section, the next step is to define *context connections*, that is, links among relevant local contexts that will be used to create the final segments for activity recognition.

Algorithm 2. Creating context connections

Data: Local contexts: $L = \{l_1, l_2, ..., l_j\}$, where $l_j = \langle o_j, N_j^r, C_k \rangle$.
Result: Set C_{set} with context connections.

1 $C_{set} \leftarrow \emptyset$;
2 **foreach** $l_i = \langle o_i, N_i^r, C_m \rangle \in L$ **do**
3   **foreach** $l_j = \langle o_j, N_j^r, C_n \rangle \in L$, *where* $o_i \neq o_2$, $C_m \equiv C_n$ *and* $o_i \in N_j^r$ **do**
4         $C_{set} \leftarrow C_{set} \cup \{l_i \xmapsto{C_m} l_j\}$

Definition 2. *Two local contexts* $l_i = \langle o_i, N_i^r, C_m \rangle$ *and* $l_j = \langle o_j, N_j^r, C_n \rangle$ *are linked with a context connection, denoted as* $l_i \xmapsto{C_m} l_j$, *if* $o_i \in N_j^r$ *and* $C_m \equiv C_n$.

Intuitively, a context connection captures the contextual dependency between two neighbouring observations o_i and o_j with respect to a common high-level classification activity C_m ($C_m \equiv C_n$). Note that symmetry and transitivity do not hold. For example, in Fig. 3 (a), *spoon*1 (o_9) belongs to the neighbourhood set N_5^1 of *table.zone*1 (o_5) ($N_5^1 = \{o_1, o_3, o_4, o_5, o_6, o_7, o_8, o_9\}$), but *table.zone*1 does not belong to the neighbourhood set N_9^1 of *spoon*1 ($N_9^1 = \{o_6, o_8, o_9, o_{10}\}$).

Algorithm 2 describes the process for creating the set of context connections C_{set}. Two local contexts $l_i = \langle o_i, N_i^r, C_m \rangle$ and $l_j = \langle o_j, N_j^r, C_n \rangle$ are retrieved from L, such that o_i belongs to the neighbourhood of o_j ($o_i \in N_j^r$) and $C_m \equiv C_n$ (lines 2 and 3), and the context connection $l_i \xmapsto{C_m} l_j$ is added to C_{set} (line 4).

Example. Algorithm 2 creates 29 context connections among the local contexts described in Section 4.1 for the running example in Fig. 3 (a). For example, o_7 belongs to the neighbourhood of the local contexts l_5, l_6 and l_8, i.e. $o_7 \in N_5^1, o_7 \in N_6^1$ and $o_7 \in N_8^1$. As described in Section 4.1, the classification class of l_5 and l_8 is $DrinkTea$ (DT), whereas the classification class of l_6 is $MakeTea$ (MT). Moreover, o_7 has two local contexts: l_7 with classification class $MakeTea$ and l'_7 with classification class $DrinkTea$. Therefore, $l_7 \xmapsto{MT} l_6$ and $l'_7 \xmapsto{DT} l_5$, $l'_7 \xmapsto{DT} l_8$. The other context connections that are generated are: $l_1 \xmapsto{MT} l_2$, $l_1 \xmapsto{MT} l_3$, $l_1 \xmapsto{MT} l_4$, $l_2 \xmapsto{MT} l_1$, $l_2 \xmapsto{MT} l_3$, $l_3 \xmapsto{MT} l_1$, $l_3 \xmapsto{MT} l_2$, $l_3 \xmapsto{MT} l_4$, $l_4 \xmapsto{MT} l_1$, $l_4 \xmapsto{MT} l_2$, $l_4 \xmapsto{MT} l_3$, $l_4 \xmapsto{MT} l_6$, $l_4 \xmapsto{MT} l_7$, $l_6 \xmapsto{MT} l_3$, $l_6 \xmapsto{MT} l_7$, $l_6 \xmapsto{MT} l_9$, $l_9 \xmapsto{MT} l_6$, $l_9 \xmapsto{MT} l_7$, $l_5 \xmapsto{DT} l_7$, $l_5 \xmapsto{DT} l_8$, $l_8 \xmapsto{DT} l_5$, $l_8 \xmapsto{DT} l_7$, $l_8 \xmapsto{DT} l_9$, $l_9 \xmapsto{DT} l_5$, $l_9 \xmapsto{DT} l_7$, $l_9 \xmapsto{DT} l_8$.

4.3 Activity Situations and Recognition

The last step is to create *activity situations*, i.e. subsets of the initial set of observations O, and to compute the similarity σ to the context descriptor d_C.

Definition 3. *An activity situation* S *is defined as the tuple* $\langle Obs, C, V \rangle$, *where* $Obs \subseteq O$ *is the set of the observations that belong to the activity situation and* V *denotes the similarity of* S *to the context descriptor* d_C, *such that* $V = \sigma(d_C, Obs_T)$, *where* $d_C \in D$ *and* $Obs_T = \{t \mid \forall o_i \in Obs, t = T(o_i)\}$.

Situation Similarity σ. The σ measure captures the similarity between the domain context descriptor of class C, namely d_C, and set Obs_T with the most specific classes of the observations in a situation.

$$\sigma(d_C, Obs_T) = \frac{\sum\limits_{\forall n \in d_C} \max\limits_{\forall c \in Obs_T} \left[\delta(c, n)\right]}{|d_C|} \tag{3}$$

Similarly to φ in (1), σ denotes the similarity of two sets of concepts. However, φ aims to capture the local (partial) similarity of neighbourhood class types (N_T^r) against the context descriptor d_C. In contrast, σ captures the similarity of the context descriptor d_C against the set of situation observation class types (Obs_T), in order to derive the final plausibility for the corresponding situation. If $\sigma = 1$, then all the classes in d_C appear in Obs_T, meaning that the situation can be considered identical to the context descriptor d_C, and, therefore, to class C.

Creating Activity Situations. An activity situation is derived by simply traversing the path defined by context connections $l_a \xmapsto{C_m} l_b \xmapsto{C_m} \ldots \xmapsto{C_m} l_e$, collecting the observations o_i of the local contexts l_i found in the path. The collected observations constitute set Obs of a situation $S = \langle Obs, C_m, V \rangle$.

Algorithm 3 describes the aforementioned procedure. It begins by selecting a context connection $l_i \xmapsto{C_m} l_j$, which has not been visited yet (line 2), as the root of the current path, adding it to the $Expand$ set (line 4). In each iteration, a context connection $l_k \xmapsto{C_m} l_l$ is selected from the $Expand$ set and: (a) the observations of the pertinent local contexts are added to Obs (line 7), (b) the current context connection is added to the $Visited$ set (line 8), and (c) the context connections $l_p \xmapsto{C_m} l_q$ are retrieved from C_{set} and added to the $Expand$ set, such that $l_p = l_l$ (lines 9, 10). An empty $Expand$ set denotes that there are no other context connections in the current path. In this case, the context descriptor of C_m (d_{C_m}) is compared against the set Obs_T with the most specific types of observations in Obs to compute the σ similarity of S (line 11).

Example. By applying Algorithm 3 over the context connections presented in Section 4.2, two situations are generated: $S_1 = \langle Obs_1, MakeTea, 0.833 \rangle$ and $S_2 = \langle Obs_2, DrinkTea, 1.0 \rangle$, where $Obs_1 = \{o_1, o_2, o_3, o_4, o_6, o_7, o_9\}$ and $Obs_2 = \{o_5, o_7, o_8, o_9\}$. It is worth noting that despite the overlapping and noisy nature of the observations in the example (e.g. the location-related observations o_3 and o_5 overlap), the algorithm is able to discriminate the two situations of making and drinking tea by also connecting the relevant observations.

Moreover, the nearest observations threshold r in the running example was set to 1, meaning that, apart from overlapping observations, the 1-nearest observations were also taken into account to define neighbourhood relations. If we instead use $r = 0$, then we get the following situations: $S_1' = \langle Obs_1', MakeTea, 0.666 \rangle$ and $S_2' = \langle Obs_2, DrinkTea, 1.0 \rangle$, where $Obs_1' = \{o_1, o_2, o_3, o_4\}$ and $Obs_2' = \{o_5, o_7, o_8, o_9\}$. In this case, o_6 ($teabag2$) is not connected with observations

Algorithm 3. Creating activity situations

Data: Context connections: $C_{set} = \{l_a \xmapsto{C_k} l_b, l_e \xmapsto{C_l} l_f, ..., l_i \xmapsto{C_m} l_j\}$.
Result: The set S_{set} with activity situations S.

$S_{set}, Visited \leftarrow \emptyset$;
foreach $l_i \xmapsto{C_m} l_j \in C_{set} \wedge l_i \xmapsto{C_m} l_j \notin Visited$ **do**
  $Obs \leftarrow \emptyset$;
  $Expand \leftarrow \{l_i \xmapsto{C_m} l_j\}$;
  while $Expand \neq \emptyset$ **do**
    $l_k \xmapsto{C_m} l_l \leftarrow Expand.pop$;
    $Obs \leftarrow Obs \cup \{o_k, o_l\}$;
    $Visited \leftarrow Visited \cup \{l_k \xmapsto{C_m} l_l\}$;
    $Cons \leftarrow \{l_p \xmapsto{C_m} l_q \mid l_p = l_l, \forall l_p \xmapsto{C_m} l_q \in C_{set}, l_p \xmapsto{C_m} l_q \notin Visited\}$;
    $Expand \leftarrow Expand \cup Cons$;
  $S_{set} \leftarrow S_{set} \cup \{S\}$, where $S \leftarrow \langle Obs, C_m, \sigma(d_{C_m}, Obs_T)\rangle$ and $Obs_T = \{t \mid \forall o \in Obs, t = T(o)\}$;

relevant to the $MakeTea$ activity, and is considered as noise, breaking also the connection of o_7 and o_9 with the $MakeTea$ activity. Despite the fact that the $MakeTea$ activity is detected with a lower plausibility when $r = 0$, it could be argued that the resulted situations are more meaningful for further analysis. Intuitively, r allows to control the amount of contextual information taken into account during the definition of the neighbourhood sets and local contexts of observations. Currently, r is defined manually based on domain knowledge regarding the quality and temporal characteristics of the data used.

5 Deployment, Results and Discussion

We have implemented our approach on top of OWLIM [6], following an ontology-based representation of local contexts, context connections and situations and using SPARQL queries (rules) for implementing the algorithms (SPARQL Inferencing Notation - SPIN [16]). Fig. 4 presents a sample SPARQL query that implements Algorithm 2 for creating context connections.

Our framework is part of a real-world deployment for monitoring Alzheimer's disease patients in a hospital[2]. The aim is to help clinicians assess the patients' condition through a goal-directed protocol of 10 Instrumental Activities of Daily Living (IADL), e.g. preparing the drug box (Fig. 5). Based on primitive observations, high-level activities are recognised that inform the clinicians, who are not present during the protocol, about activities with too long duration or activities that have been missed or repeated. The setting involves wearable and ambient video and audio sensors, accelerometers and physiological sensors. Table 1 presents the context descriptors used for the detection of the 10 activities.

[2] The system has been installed in the Memory Resource and Research Centre (CMRR) of the University Hospital in Nice (CHUN), under Dem@Care FP7 Project.

```
CONSTRUCT {
  [] a :ContextConnection; :li ?li; :lj ?lj; :classification ?Cm.
}
WHERE {
  ?li a :LocalContext; :obs ?oi; :classification ?Cm.
  ?lj a :LocalContext; :obs ?oj; :classification ?Cm; :neighbour ?oi.
  FILTER (?oi != ?oj).
  NOT EXISTS {[] a :ContextConnection; :li ?li; :lj ?lj; :classification ?Cm.}.
}
```

Fig. 4. SPARQL-based implementation of Algorithm 2

Table 1. The context descriptor dependencies of the high-level activities

IADL	**Context Descriptor Classes**
Establish account balance	Sitting, Accounts, Table, TableZone, Pen
Prepare drug box	Pillbox, Basket, MedicationZone
Prepare hot tea	Kettle, TeaZone, TeaBag, Cup, Sugar, TeaBox
Search for bus line	Map, MapZone, RouteInstructions
Make a phone call	Phone, PhoneZone, PickUpPhone, Talk
Watch TV	Remote, TV, TVZone, Sitting
Water the plant	WateringCan, PlantZone, Bending, Plant
Write a check	Sitting, Pen, Check, TableZone, Table
Read an article	Sitting, TableZone, Newspaper, Table
Enter/Leave the room	DoorOpen, EmptyRoom

Table 2 summarises the performance on a dataset of 25 participants, where True Positives (TP) is the number of IADLs correctly recognised, False Positives (FP) is the number of IADLs incorrectly recognised as performed and False Negatives (FN) is the number of IADLs that have not been recognised. The True Positive Rate (TPR) and Positive Predicted Value (PPV) measures denote the recall and precision, respectively, and are defined as:

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

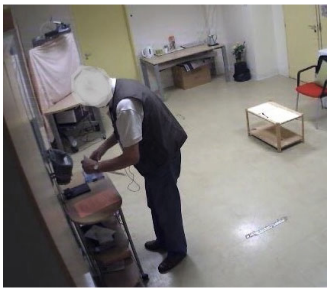
We used $r = 0$, since the dataset contains highly overlapping observations, and we set a minimum threshold on σ (≥ 0.65), so as to ignore activities with low plausibility. To demonstrate the effect of a higher r value in our experiments, we also present the performance using $r = 5$. By increasing the r value, the number of neighbours for each observation also increases. This way, more local contexts are generated, affecting precision and recall. As explained, the optimal r value depends on the data quality and temporal characteristics and, in principle, datasets with highly overlapping and incoherent observations need small r values.

Our approach achieves the best accuracy for activities “Prepare hot tea”, “Make a phone call”, “Watch TV” and “Water the plant”, whose context descriptors encapsulate richer domain contextual information, compared to “Prepare drug box” and “Search for bus line”. On the other hand, the recall of these

Table 2. Activity recognition performance

	$r=0$					$r=5$	
IADL	**TP**	**FP**	**FN**	**TPR%**	**PPV%**	**TPR%**	**PPV%**
Establish account balance	30	10	4	88.24	75.00	85.71	73.17
Prepare drug box	23	3	2	92.00	88.46	85.19	82.14
Prepare hot tea	23	1	6	79.31	95.83	76.67	88.46
Search for bus line	24	4	1	96.30	86.67	92.86	83.87
Make a phone call	24	1	3	89.29	96.15	86.21	92.59
Watch TV	21	1	4	84.00	95.45	80.77	91.30
Water the plant	20	1	5	80.00	95.24	80.00	86.96
Write a check	28	8	4	87.50	77.78	87.50	75.68
Read an article	23	4	1	95.83	85.19	92.00	85.19
Enter/Leave the room	49	0	1	98.00	100	98.00	98.00

(a) Writing a check



(b) Preparing the drug box

Fig. 5. Example IADL activities of the protocol (wearable and ambient camera)

activities is relatively low, since they are more susceptible to false negatives, requiring richer contextual dependencies to be present.

Another interesting finding involves activities "Establish account balance", "Write a check" and "Read an article". The context descriptors of these activities have many members in common; e.g. "Accounts" and "Checks" are the only discriminating contextual objects between "Establish account balance" and "Write a check". This way, our approach detects both activities when the corresponding observations are missing, resulting in low accuracy. Finally, the "Enter/Leave the room" activities share exactly the same context descriptors; however, we distinguish them (in an ad hoc manner) by the order in which they take place.

As already mentioned, our approach is currently used in an offline mode, where each participant's data is collected and processed after the execution of the protocol; therefore, the processing time is not a critical requirement in the current setting. However, Fig. 6 gives a gist about the required time for processing different sets of observations using a Dell Optiplex 7010 PC configuration (i7-3770 Processor, Quad Core, 3.40GHz). In each protocol execution, approx. 500 to 800 observations are generated, depending on the participant's performance.

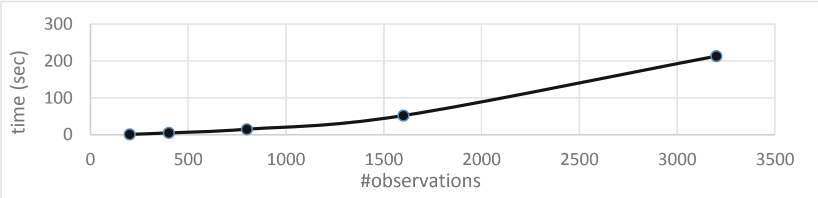


Fig. 6. Activity recognition time in relation to the number of observations

Discussion. Our framework achieves an average TPR and PPV close to 90%, demonstrating the feasibility of our approach in real-world settings. However, there are still certain limitations, which we consider as very important research directions for future work. First, our approach cannot handle interleaved activities, nor can it resolve conflicts after the recognition process, as argued for the "Establishing account balance" and "Write check" activities. We are investigating the use of defeasible reasoning on top of the framework for further enhancing the activity recognition capabilities. Second, our next step is to deploy the framework in homes for providing context-aware real-time assistance to Alzheimer's patients. To this end, we are currently investigating adaptations of our algorithms to allow the dynamic and incremental generation of local contexts and context connections for real-time activity segmentation and recognition.

In addition, one of the most challenging tasks in pervasive healthcare is patient profiling, which can provide the behaviour interpretation and feedback services with knowledge-driven personalisation capabilities and adaptive support services. To this end, we are exploring methods for extracting behaviour patterns and detecting behaviour changes from activity situations that are generated based on the abstract context descriptors presented here. Regarding the representation of these patterns, our objective is to take full advantage of the DnS pattern, associating each Situation to one or more behavioural Description instantiations pertinent to patients' idiosyncratic and habitual information.

6 Conclusions

We propose a knowledge-driven framework towards activity recognition and segmentation, coupling ontology models of abstract domain activity dependencies with a context-aware approach for multi-sensor fusion and monitoring. We formalise activity dependencies, capitalising upon the *Situation* conceptualisation of the DnS ontology pattern in DUL for defining generic context descriptors, whereas activity segmentation and recognition is reduced in linking and classifying meaningful contextual segments. We elaborated on the obtained results from the evaluation of our approach in a real-world deployment, monitoring activities of elderly people during the execution of a clinical protocol for assessing Alzheimer's disease. The use of generic context descriptors in representing activity models achieves very promising results, leading to handling the intrinsically noisy and imperfect information in multi sensory environments, beyond strict activity patterns and background knowledge (e.g. max activity duration).

The key directions that underpin our ongoing research involve (a) introducing an additional layer for detecting interleaved activities and resolving conflicts, (b) adapting our algorithms for supporting real-time context-aware monitoring, and, (c) patient profiling through the extraction and learning of behavioural patterns from the detected activity situations. In addition, we are investigating extensions to the *Situation* model for capturing richer contextual dependencies, such as compositions of context descriptors.

Acknowledgments. 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. Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: Ep-sparql: a unified language for event processing and stream reasoning. In: Proceedings of the 20th International Conference on World Wide Web, pp. 635–644. ACM (2011)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
3. Barbieri, D., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: C-SPARQL: A Continuous Query Language for RDF Data Streams. International Journal of Semantic Computing (IJSC) 4(1) (2010)
4. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of context modelling and reasoning techniques. Pervasive Mob. Comput. 6(2), 161–180 (2010)
5. Bikakis, A., Antoniou, G., Hasapis, P.: Strategies for contextual reasoning with conflicts in ambient intelligence. Knowl. and Infor. Systems 27(1), 45–84 (2011)
6. Bishop, B., Kiryakov, A., Ognyanoff, D., Peikov, I., Tashev, Z., Velkov, R.: OWLIM: A family of scalable semantic repositories. Sem. Web 2(1), 33–42 (2011)
7. Chen, L., Khalil, I.: Activity recognition: Approaches, practices and trends. In: Activity Recognition in Pervasive Intelligent Environments, vol. 4, pp. 1–31 (2011)
8. Chen, L., Nugent, C.D.: Ontology-based activity recognition in intelligent pervasive environments. Int. Journal of Web Information Systems 5(4), 410–430 (2009)
9. Chen, L., Nugent, C.D., Wang, H.: A knowledge-driven approach to activity recognition in smart homes. IEEE Trans. Knowl. and Data Engin. 24(6), 961–974 (2012)
10. Cook, D.J., Augusto, J.C., Jakkula, V.R.: Ambient intelligence: Technologies, applications, and opportunities. Perv. and Mobile Computing 5(4), 277–298 (2009)
11. DOLCE Ultralite ontology, `http://www.loa.istc.cnr.it/ontologies/DUL.owl`
12. Eiter, T., Ianni, G., Krennwallner, T., Polleres, A.: Rules and ontologies for the semantic web. In: Baroglio, C., Bonatti, P.A., Małuszyński, J., Marchiori, M., Polleres, A., Schaffert, S. (eds.) Reasoning Web. LNCS, vol. 5224, pp. 1–53. Springer, Heidelberg (2008)
13. Gangemi, A., Mika, P.: Understanding the semantic web through descriptions and situations. In: Meersman, R., Schmidt, D.C. (eds.) CoopIS/DOA/ODBASE 2003. LNCS, vol. 2888, pp. 689–706. Springer, Heidelberg (2003)
14. Hong, X., Nugent, C.D., Mulvenna, M.D., Martin, S., Devlin, S., Wallace, J.G.: Dynamic similarity-based activity detection and recognition within smart homes. Int. J. Pervasive Computing and Communications 8(3), 264–278 (2012)
15. Jekjantuk, N., Gröner, G., Pan, J.Z.: Modelling and reasoning in metamodelling enabled ontologies. In: Bi, Y., Williams, M.-A. (eds.) KSEM 2010. LNCS (LNAI), vol. 6291, pp. 51–62. Springer, Heidelberg (2010)

16. Knublauch, H., Hendler, J.A., Idehen, K.: SPIN - overview and motivation. W3C member submission, World Wide Web Consortium (February 2011)
17. Meditskos, G., Dasiopoulou, S., Efstathiou, V., Kompatsiaris, I.: Sp-act: A hybrid framework for complex activity recognition combining owl and sparql rules. In: PerCom Workshops, pp. 25–30 (2013)
18. Moser, T., Roth, H., Rozsnyai, S., Mordinyi, R., Biffl, S.: Semantic event correlation using ontologies. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2009, Part II. LNCS, vol. 5871, pp. 1087–1094. Springer, Heidelberg (2009)
19. Motik, B., Cuenca Grau, B., Sattler, U.: Structured objects in OWL: representation and reasoning. In: World Wide Web, pp. 555–564 (2008)
20. Okeyo, G., Chen, L., Hui, W., Sterritt, R.: A hybrid ontological and temporal approach for composite activity modelling. In: TrustCom, pp. 1763–1770 (2012)
21. Okeyo, G., Chen, L., Wang, H., Sterritt, R.: Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. Pervasive and Mobile Computing 10(Pt. B), 155–172 (2014)
22. Palmes, P., Pung, H.K., Gu, T., Xue, W., Chen, S.: Object relevance weight pattern mining for activity recognition and segmentation. Perv. Mob. Comput. 6(1), 43–57 (2010)
23. Patkos, T., Chrysakis, I., Bikakis, A., Plexousakis, D., Antoniou, G.: A reasoning framework for ambient intelligence. In: Konstantopoulos, S., Perantonis, S., Karkaletsis, V., Spyropoulos, C.D., Vouros, G. (eds.) SETN 2010. LNCS (LNAI), vol. 6040, pp. 213–222. Springer, Heidelberg (2010)
24. Riboni, D., Pareschi, L., Radaelli, L., Bettini, C.: Is ontology-based activity recognition really effective? In: Perv. Comp. and Commun., pp. 427–431 (2011)
25. Riboni, D., Bettini, C.: COSAR: hybrid reasoning for context-aware activity recognition. Personal Ubiquitous Comput. 15(3), 271–289 (2011)
26. Riboni, D., Bettini, C.: OWL 2 modeling and reasoning with complex human activities. Pervasive and Mobile Computing 7(3), 379–395 (2011)
27. Roy, P., Giroux, S., Bouchard, B., Bouzouane, A., Phua, C., Tolstikov, A., Biswas, J.: A possibilistic approach for activity recognition in smart homes for cognitive assistance to alzheimers patients. In: Activity Recognition in Pervasive Intelligent Environments, Atlantis Ambient and Pervasive Intelligence, vol. 4, pp. 33–58 (2011)
28. Teymourian, K., Paschke, A.: Semantic rule-based complex event processing. In: Governatori, G., Hall, J., Paschke, A. (eds.) RuleML 2009. LNCS, vol. 5858, pp. 82–92. Springer, Heidelberg (2009)
29. Teymourian, K., Rohde, M., Paschke, A.: Fusion of background knowledge and streams of events. In: Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems, New York, NY, USA, pp. 302–313 (2012)
30. Tiberghien, T., Mokhtari, M., Aloulou, H., Biswas, J.: Semantic reasoning in context-aware assistive environments to support ageing with dementia. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part II. LNCS, vol. 7650, pp. 212–227. Springer, Heidelberg (2012)
31. Wessel, M., Luther, M., Wagner, M.: The difference a day makes - recognizing important events in daily context logs. In: C&O:RR (2007)
32. Ye, J., Stevenson, G.: Semantics-driven multi-user concurrent activity recognition. In: Augusto, J.C., Wichert, R., Collier, R., Keyson, D., Salah, A.A., Tan, A.-H. (eds.) AmI 2013. LNCS, vol. 8309, pp. 204–219. Springer, Heidelberg (2013)
33. Zhang, S., McCullagh, P., Nugent, C., Zheng, H.: An ontology-based context-aware approach for behaviour analysis. In: Activity Recognition in Pervasive Intelligent Environments, vol. 4, pp. 127–148. Atlantis Press (2011)