Combining Ontological and Temporal Formalisms for Composite Activity Modelling and Recognition in Smart Homes

George Okeyo[1] Liming Chen[2] Hui Wang[3]
[1]School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology, Kenya
(gokeyo@icsit.jkuat.ac.ke)

[2]School of Computer Science and Informatics, De Montfort University, United Kingdom
(liming.chen@dmu.ac.uk)

[3]School of Computing and Mathematics, University of Ulster, United Kingdom

(h.wang@ulster.ac.uk)

Abstract. Activity recognition is essential in providing activity assistance for users in smart homes. While significant progress has been made for single-user single-activity recognition, it still remains a challenge to carry out real-time progressive composite activity recognition. This paper introduces a hybrid ontological and temporal approach to composite activity modelling and recognition by extending existing ontology-based knowledge-driven approach. The compelling feature of the approach is that it combines ontological and temporal knowledge representation formalisms to provide powerful representation capabilities for activity modelling. The paper describes in details ontological activity modelling which establishes relationships between activities and their involved entities, and temporal activity modeling which defines relationships between constituent activities of a composite activity. As an essential part of the model, the paper also presents methods for developing temporal entailment rules to support the interpretation and inference of composite activities. In addition, this paper outlines an integrated architecture for composite activity recognition and elaborated a unified activity recognition algorithm which can support the recognition of simple and composite activities. The approach has been implemented in a feature-rich prototype system upon which testing and evaluation have been conducted. Initial experimental results have shown average recognition accuracy of 100**%** and 88.26**%** for simple and composite activities, respectively.

Keywords. Composite activities; interleaved activities; concurrent activities; activity recognition; activity modelling; ontologies; smart homes.

1 INTRODUCTION

Smart Homes (SH) have been widely viewed as a promising paradigm for technology-driven assistive living for aging population [1]. A SH can be described as a home setting augmented with a diversity of multi-modal sensors, actuators and devices along with Information and Communication Technology (ICT) based services and systems [2]. By monitoring environmental changes and inhabitant's activities, an assistive system within a SH can process sensor data, infer an inhabitant's needs and take appropriate actions to help the inhabitant perform daily living activities. As such, a SH can help older people prolong their independent living and enhance quality of life within their own homes. Generally, two types of daily living activities exist, namely, activities of daily living (ADL), and instrumental activities of daily living (IADL). ADL refers to activities concerned with taking care of one's own body. Essentially, it relates to activities that involve functional mobility (called basic ADL), and personal care (called personal ADL) [3]. IADL refers to activities concerned with interacting with the environment and as such they can be delegated and performed by other people in the environment [3]. In the rest of this paper, we will use ADL or activity to refer to both ADL and IADL for ease of reference.

SH inhabitants typically perform ADLs in complex patterns. For instance, an inhabitant may perform two (or more) activities in sequence or in parallel. Whenever activities are performed sequentially or in parallel, there will be underlying inter-activity dependencies among the activities involved. These inter-activity dependencies should be encoded during activity modelling so as to support activity recognition in the presence of complex activity patterns, e.g. composite activities. Applications that provide SH inhabitants with services, e.g. assistive services, should be able to correctly identify both simple and composite activities. Activity recognition is the process of tracking users and identifying the activities they are performing. It involves activity sensing, activity

modelling, and activity inference. Activity sensing is responsible for monitoring users and their situated environment to obtain sensor data streams. Activity modelling creates computational activity models that are used to analyze and classify collections of sensor data into activities. Activity inference uses relevant algorithms to process sensor data against computational activity models to identify the ongoing activity.

In this paper we categorise activities as *actions*, *simple activities*, and *composite activities*. An action is an atomic (or indivisible) activity, e.g. grasping the fridge door. A simple activity is an ordered sequence of actions, e.g. preparing coffee. Finally, a composite activity is a collection of two or more simple activities occurring within a given time interval, e.g. preparing dinner and washing dishes. Composite activities can be further categorised as sequential or multi-task activities. A sequential activity is a sequence of activities that occur in consecutive time intervals, i.e., there is temporal dependency between constituent activities. A multi-task activity occurs when a single user performs two or more activities simultaneously or when multiple residents occupy a smart environment and perform activities concurrently.

Activity recognition has been widely investigated using three categories of approaches, namely, data-driven (DD) [4-8], knowledge-driven (KD) [9-13], and hybrid [14-16] activity recognition approaches. In data-driven activity recognition, activity models are learnt from pre-existing datasets using existing well-developed machine learning techniques. Activity inference is then performed against the learnt activity models whenever sensor data is obtained. In knowledge-driven activity recognition, knowledge engineers and domain experts specify activity models using a knowledge engineering process. The activity models capture commonsense and domain knowledge about activities. Artificial intelligence-based reasoning techniques are then used to infer activities from the models whenever sensor data is obtained. Hybrid activity recognition approaches combine data-driven and knowledge-driven techniques.

Simple activity recognition has been widely explored in DD [6, 17-20], KD [9-11, 21-23] , and hybrid [13, 24, 25] activity recognition. However, composite activity recognition is only investigated to a limited extent in DD [4, 6, 8, 26-28] and hybrid [14-16] activity recognition communities. In the KD activity recognition research community, the recognition of composite activities still remains largely unexplored. This challenge can be attributed to the two tasks of activity modelling and activity inference. Composite activity modelling is a challenge because activity models must capture and reason with inter-activity dependencies that are typically encoded as temporal knowledge [29]. Moreover, mechanisms are needed to process sensor data against the resulting composite activity models to infer the ongoing activities [30].

The use of ontologies in activity modelling and activity recognition has spurred interest but the focus has largely been on simple activities [9, 10, 31]. Ontological activity modelling can be used to define activity ontologies that describe activities and their characteristics [9, 10]. The resulting activity ontologies represent activity models for mostly simple activities and support semantic reasoning for activity recognition. To support composite activity modelling and recognition, we have developed a novel activity modelling approach that combines ontologies and temporal knowledge to create activity models that represent inter-activity dependencies using temporal relationships. The approach enhances ontological activity models by adding qualitative temporal knowledge based on Allen's temporal logic relations [32]. It is worth pointing out that the study presented in this paper is contextualised in a single-resident SH environment within which the user performs both simple and composite activities.

In this paper we make a number of knowledge contributions. Firstly, we introduce a novel hybrid approach to composite activity modelling and recognition. The combination of ontological and temporal knowledge representation formalisms provides a more expressive representation formalism required for representing and modelling the complex ontological and temporal relationships of composite activities. Secondly, we develop generic activity models for composite activities based on the presented approach. This includes three core elements, namely ontological activity models, temporal activity models and entailment rules; each element models a specific aspect of composite activities. The generic models can be applied to modelling composite activities in different application scenarios. In this paper we create reusable activity models for ADLs in the context of smart homes for the purpose of illustration, testing and evaluation. Thirdly, we develop an integrated system architecture for composite activity recognition and a unified activity recognition algorithm. The algorithm can reason over sensor data streams against composite activity models to perform real-time progressive activity recognition for both simple and composite activities. In addition, we have developed a system prototype and well-designed experiments for testing and evaluation. The presented approach and associated models and methods have not been seen in related research communities.

The remainder of the paper is organised as follows. Section 2 discusses related works. Section 3 presents the hybrid approach for activity recognition. In Section 4, activity models, inference rules, and recognition algorithms are described. Section 5 presents the system prototype. The experiments and evaluation results are provided in Section 6. Finally, Section 7 concludes the paper and outlines future work.

2 RELATED WORK

In the DD activity recognition community, existing approaches capable of both simple and composite activity modelling and recognition include hidden Markov models (HMM) [6], interleaved HMM [4], factorial conditional random fields (FCRF) [26], skip-chain conditional random fields (SCCRF) [8, 28] [27] and mining of emerging patterns [5]. DD approaches have the ability to handle uncertain knowledge and are based on well-explored machine learning based techniques. They also have the advantage to handle temporal information that can capture short- and long-term temporal dependencies, e.g. inter-activity relationships and activity history, thereby making them suited to composite activity recognition. The main drawback is that large amounts of initial training data are needed to learn the activity models. As users perform activities in a variety of ways, all these activity variants must be present in the data set if they are to be successfully learnt, modeled and subsequently recognized. In most cases it is difficult to obtain representative and sufficient data sets to be used for learning activity models, thus leading to the "cold start" problem. In addition, users perform activities in different manners; as a result models learnt from one user's datasets would not be reused by another user, which results in reusability problem.

KD activity recognition approaches use knowledge representation formalisms to provide explicit activity models which can be processed by artificial intelligence-based inference for activity recognition. The KD has a number of strengths. For instance, it is grounded in logic theory making it possible to capture the semantics of a domain and support automated reasoning. It also allows common sense domain knowledge and heuristics associated with activities to be incorporated into activity models. Domain knowledge is especially important in modelling complex real-world activity scenarios, e.g. interleaved and concurrent activities. Moreover, it can support reuse and knowledge sharing between applications. Domain knowledge and common sense knowledge is essentially common across applications, hence the ability to encode and share it would make application development easier through reuse. The KD approach has been used for simple activity modeling and recognition, but little work has been done in composite activity recognition. Saguna et al. [33] addressed both simple and composite activity recognition by combining ontological and spatio-temporal modelling and reasoning. It uses the notion of context-driven activity theory (CDAT) to encode context information in order to model both primitive actions and simple activities. The resulting models are combined with ontological situation models and used to infer interleaved and concurrent activities. The authors derive models of situations, based on spatio-temporal information, from the context spaces theory [34], and use the resulting situations in activity inference. Essentially, it uses a layered approach to activity recognition consisting of atomic activity recognition (using machine learning techniques), simple activity recognition (using ontological inference), and finally interleaved and concurrent activity recognition (using rule-based inference).

Hybrid approaches to activity recognition combine techniques from data-driven and knowledge-driven activity recognition. So far, only Markov logic networks (MLN) [15] and HMMs with Allen logic [16] have been used to support activity recognition of simple activities, and interleaved and concurrent activities. Both approaches encode and use temporal knowledge but rely on automatically extracting the relevant temporal patterns from data sets. The main strength of the hybrid approaches is that they can model and recognize a range of simple and composite activities due to their ability to encode rich domain knowledge, e.g. temporal knowledge, and still utilize well-developed learning and probabilistic models. However, they suffer the "cold start" problem just like the DD approaches. Since our approach requires activity models to be specified based on domain knowledge, it overcomes the "cold start" problem.

Our work follows the KD approach but differs from Saguna's work, in two main aspects. First, the latter requires the training of atomic activity recognition models from data sets. Secondly, it uses an ad-hoc method to encode both temporal and spatial knowledge into activity ontologies. Basically, it simply captures temporal and spatial knowledge as properties of ontology concepts, a strategy that ignores reasoning and querying challenges arising from processing temporal or spatial knowledge in ontologies. Our work adopts a systematic and clear method for encoding and reasoning with temporal knowledge based on 4D-fluents [35] and therefore provides a clear mechanism for seamlessly integrating and exploiting qualitative temporal knowledge in activity recognition for both simple and composite activities.

3 THE HYBRID APPROACH TO COMPOSITE ACTIVITY MODELLING

ADLs possess several unique characteristics that make activity modelling a difficult task. Firstly, there are over 20 categories of ADLs [36] [37] [38]. These include dental care, hygiene, bathing, dressing, using the toilet, drinking, transferring, mobility, orientation to time, driving/ using public transport, managing finances, drink preparation, use of the telephone, food preparation, housework, communication, shopping, eating, orientation to space, and games and hobbies [38-40]. Furthermore, each category can be classified to activities at multiple levels of granularity. For example, the activity *Food Preparation* can be broken down to child activities such as '*Prepare Coffee', 'Prepare Toast, etc.* Also*, 'Prepare Coffee* can further be classified into its child activities '*Prepare Espresso*, '*Prepare Latte'*, etc. Thus *Food Preparation* is coarse grained, whereas '*Prepare*

Coffee is fine grained. Therefore, activity modelling approaches should support the different categories and granularities of activities. Secondly, most ADLs involve performing a number of actions, with the ordering of actions dependent on an individual's preferences or abilities leading to a large number of ADL variants. Activity models should encode this activity diversity. Thirdly, the performance of activities may continuously change, e.g. activity duration or the sequence of objects can change, based on the users' abilities or preferences. Activity models, therefore, should be flexible to accommodate these variations. Fourthly, users also perform activities using complex patterns, such as interleaved and concurrent activities. Therefore, activity models should encode such complex relationships. Fifthly, activities are performed under different contexts, e.g. specific locations, objects, time, space, and goals. This is even more evident in composite activities. The resulting contextual information should be used to characterize activities. For instance, composite activities can be described by specifying inter-activity relationships using temporal or spatial information. In general, activities are characterized by rich temporal information, e.g. repetitive time patterns, temporal sequences, temporal duration, and time instants or intervals. For example, the occurrence of two activities *A* and *B* within the same time interval can represent a temporal inter-activity dependency that signals the occurrence of a composite activity.

The range of characteristics discussed above constitutes domain knowledge and heuristics upon which we have built a high-level conceptual activity model, as shown in Figure 1, for composite activities based on the conceptual activity model that was proposed by Chen and Nugent [10]. The conceptual activity model in [7] describes an activity based on contextual elements (i.e., identity, time, space, actor, related activities, resources, environment elements, and goals) and properties that support inference (i.e., conditions and effects). However, its main limitation is that it does not explicitly provide a means to encode temporal inter-activity dependencies that typically characterize composite activities. This is because it ignored the important role that the model of change plays in composite activity modelling. Therefore, to support both simple and composite activity modelling, we have added two properties to the Time concept, i.e., *temporal reference* and *model of change*, to produce the revised model. The *temporal reference* is needed for both simple and composite activities, whereas the *model of change* is only mandatory for composite activity modelling. The temporal reference indicates the time interval or time instant that a given simple or composite activity occurs. The model of change represents the property that within a given temporal inference, a composite activity consists of two or more simple activities, whereby each simple activity can be identified by its respective temporal reference.

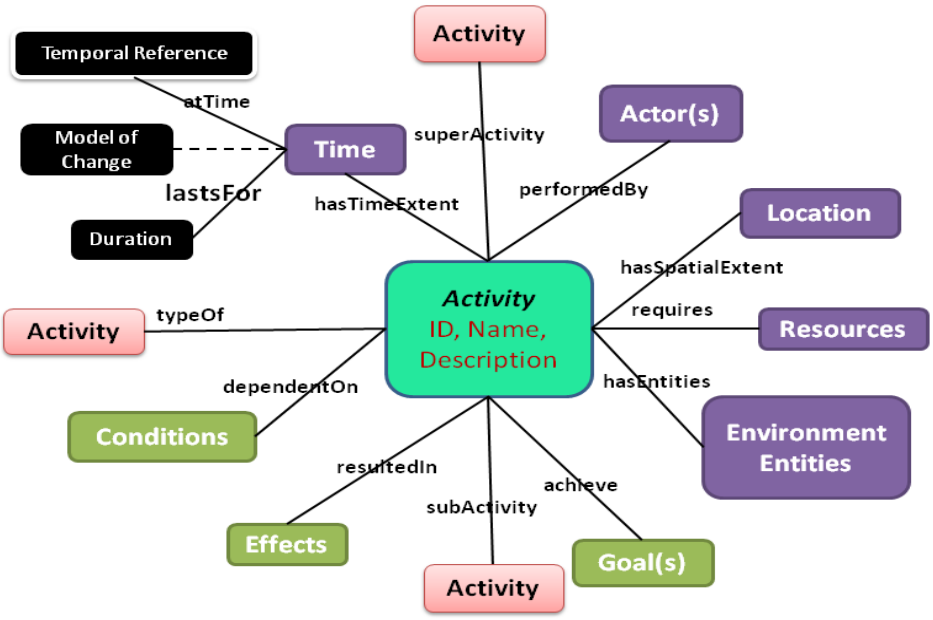

Figure 1: Revised conceptual activity model

Ontological activity modelling has been used to create simple activity models as ADL ontologies [9, 41]. In this case, ADL activities are structured in a hierarchical tree with the most specific ADL descriptions represented as leaf concepts - all leaf concepts have no child classes. Each concept is associated with a number of role (property) restrictions. All child concepts inherit all the roles of their parent concepts but may specify further constraints. A generic activity refers to an ADL class that has associated descendant classes; whereas, a specific activity (the so-called *leaf activity*) is an activity with no descendant classes in the ontology. For instance, '*Bathroom ADL'* is a generic activity, while its descendants '*Have Bath'* and '*Brush Teeth'* can be specific

activities. Nevertheless, the aforementioned approach does not work for modelling composite activities. Despite capturing temporal information, e.g. time and duration, pure ontological modeling does not support temporal inference. For instance OWL DL [42] only allows ontologies to capture temporal knowledge but does not support temporal reasoning and querying. To model composite activities, the approach to activity modelling has to be able to capture and model temporal inter-activity dependencies, and further support temporal reasoning. For this purpose we have proposed the ontological and temporal approach to activity modelling, which are described in details below.

3.1 Representing temporal knowledge in ontologies

Description logics (DL) [43] provide a mechanism that uses concepts, relations, and axioms for representing and reasoning with domain knowledge. Web ontology language (OWL) [42], a semantic web ontology language based on DL, provides a set of constructors and axioms for creating ontologies. In addition, it allows axioms for specifying subsumption, equivalence, disjointness, as well as property characteristics to be defined. The

TABLE 1: OWL CONSTRUCTORS, AXIOMS AND DL SYNTAX

OWL Constructor	DL Syntax	OWL Axiom	DL Syntax
intersectionOf	$C \sqcap D$	subClassOf	$C \sqsubseteq D$
unionOf	$C \sqcup D$	equivalentClass	$C \equiv D$
complementOf	$\neg C$	subPropertyOf	$r1 \sqsubseteq r2$
one of	$\{x1 \ldots xn\}$	equivalentProperty	$r1 \equiv r2$
allValuesFrom	$\forall r.C$	disjointWith	$C \sqsubseteq \neg D$
someValuesFrom	$\exists r.C$	sameAs	$\{x1\} \equiv \{x2\}$
hasValue	$\exists r.\{x1\}$	differentFrom	$\{x1\} \sqsubseteq \neg\{x2\}$
minCardinality	$(\geq n\ r)$		
maxCardinality	$(\leq n\ r)$		
inverseOf	r^-		

constructors, axioms, and DL equivalents are shown in Table 1 [43] . The symbols used in DL formulas are *C* and *D* for concepts; r_i for role or property names; x_j for an instances; and *n* a non negative integer.

On the other hand, temporal logic allows representation of and reasoning with temporal knowledge. Qualitative temporal knowledge naturally occurs in humans' activities, e.g. the user performs two activities, one after the other. Such qualitative temporal knowledge can be used to model complex relationships between activities that represent composite activities, e.g., sequential, and interleaved and concurrent relationships. In essence, each composite activity can be viewed as an activity that has changing relationships with other simple or composite activities. For instance, a composite sequential activity relates to two activities that occur in consecutive time intervals. In general, temporal knowledge allows knowledge at a particular moment of time

TABLE 2: THIRTEEN INTERVAL RELATIONS

Relation	Symbol	Inverse Symbol	Pictorial Illustration
X before Y	<	>	XXX YYY
X equal Y	=	=	XXX YYY
X meets Y	m	mi	XXXYYY
X overlaps Y	o	oi	XXX YYYYY
X during Y	d	di	XXX YYYYYYY
X starts Y	s	si	XXX YYYYYYY
X finishes Y	f	fi	XXX YYYYY

and the notion of change in knowledge to be encoded and reasoned with. Therefore, a temporal representation specifies a *temporal reference* and *model of change*. The temporal reference captures order in the sequence of events using either point-based or interval-based time representation. The model of change captures the changing relationships between individuals relative to the temporal reference. The two aspects (i.e. change and temporal reference) can be used to capture complex relationships between activities, e.g., sequential, and interleaved and concurrent relationships. This can be achieved by using an appropriate temporal knowledge representation mechanism to encode qualitative temporal knowledge that naturally occurs in humans' activities, e.g. the user performs two activities, one after the other.

3.2 A hybrid ontological and temporal approach

Representing temporal knowledge in OWL is a challenge because OWL only supports unary and binary relations, while adding a temporal dimension requires at least a ternary relation. Therefore, we adopt the 4D-fluents approach [35, 44] to add a temporal model as a layer on top of the underlying DL. The 4D-fluents approach uses two fundamental building blocks, namely, *time slices* and *fluents*, to provide a vocabulary to represent dynamic temporal parts of individuals. It represents concepts that have a temporal extent as 4-dimensional objects, with the fourth dimension being the time, captured as *time slices*. The time slices represent the temporal parts of a specific entity at specific moments of time and the concept itself is then defined as an aggregate of all of its time slices. Time instances and time intervals are represented as instances of a time interval class. The instances are then associated with time slices to relate them with concepts varying in time. On the other hand, *fluents* are properties that hold at specific moments in time, whether interval or instant. In essence, the fluent property holds among two time slices. Changes occur on the properties of the temporal part of the ontology while keeping the entities of the other parts of the ontology unchanged. The 4D-fluents approach is chosen because it preserves OWL semantics when incorporating temporal knowledge into OWL ontologies and can therefore exploit existing OWL reasoning support. By combining ontological and temporal representation we can obtain a hybrid representation that not only encodes temporal knowledge but also supports inference with such knowledge.

The main idea that the hybrid approach uses for composite activity modelling is that within a time interval (*a temporal reference*), a composite activity can be characterised by one or more simple activities, and the simple activities involved can vary within sub-intervals of the main interval (*model of change*). We refer to models in which it is not mandatory to represent the model of change as static activity models, whereas those that encode both the temporal reference and change are denoted as dynamic activity models.

When the 4D-fluents approach is extended with qualitative relations [44], e.g. Allen temporal logic relations [32], it can model relations that are necessary for encoding composite activities. Allen's temporal logic [32] refers to a constraint-based representation that uses a temporal interval as a primitive to support qualitative temporal knowledge representation and reasoning. It is based on the idea that much of the temporal knowledge is relative and so can be mapped into relations between intervals. The approach uses thirteen interval relations (shown in Table 2) that are considered adequate to express any relationship that can hold between two time intervals [32].

In this work we combine ontologies and temporal knowledge representation to create activity models for both simple and composite activities. To enable the resulting models to be exploited in composite activity recognition, interval relations and inference rules are used to provide procedural inference. In the next section, we apply this approach to generate activity models of simple and composite activities.

4 COMPOSITE ADL ACTIVITY MODELLING

4.1 Definitions

This section provides a set of definitions for concepts that are used to specify activity models in subsequent sections.

4.1.1 Characterization of the contextual information of smart environments

To model smart environments, we identify and define the following sets and transformations between sets: environmental entities (O), sensors (S), sensor observations (SO), and associated context information (C).

Definition 1- The set of all sensors, **S**, lists all physical sensors installed in the environment. It is defined in (1).

$$S: \{s_1, s_2,...,s_q\} \quad (1)$$

Definition 2- The set of all possible sensor observations, **SO**, lists all sensor observations that are made in the environment. Each physical sensor can generate one or more sensor observations over time. It is defined in (2).

$$SO :\{so_1, so_2, ..., so_z\} \quad (2)$$

Definition 3- The set of all objects, **O**, lists all objects that the user can interact with in the smart environment. It is defined in (3).

$$O :\{o_1, o_2,...,o_m\} \quad (3)$$

Definition 4- The set of all context elements, **C**, lists all context elements that are monitored during activity recognition. For example, it can include temporal or spatial context. It is defined in (4).

$$C: \{c_1,c_2,...,c_p\} \quad (4)$$

Definition 5- The function, ***f***, maps a sensor observation to the corresponding object that the user just interacted with. By iteratively applying the function, *f,* to the set of sensor observations, the list of objects that the user has interacted with in a given time interval can be derived and used to describe a user activity. It is defined in (5).

$f: so_i \rightarrow o_j, so_i \in SO, o_j \in O$ (5)

4.1.2 Characterization of activities of daily living

To help understand and characterize the human activities, we introduce various terminologies, namely, action, activity description, simple activity, and static and dynamic composite activities. These terms are used to derive composite activity models in the next section.

Definition 6- *Primitive action* (*a):* A single indivisible activity preformed by the user. A primitive action is specified as a 2-tuple consisting of a collections of sensor observations and context information as provided in (6).

$a:<SO_a,C_a>, SO_a \subseteq SO, C_a \subseteq C$ (6)

Definition 7- *Activity description* (AD): A collection of primitive actions, a_i, over a specific time interval. An activity description may fully or partially describe an activity and is specified using a set as shown in (7).

$AD: \{a_1, a_2,...,a_m\}$ (7)

Definition 8- *ADL*: This is the set that lists all activities of daily living (ADL) concepts, A_i , for defining simple activities in the activity model and is specified in (8).

$ADL: \{A_1, A_2,... A_n\}$ (8)

Definition 9- *lADL:* This set provides a list of all leaf ADL concepts, i.e., ADL concepts with no child concepts. It is defined in (9).

$lADL: \{lA_1, lA_2, ...,lA_k\}, k \leq n, lADL \subseteq ADL$ (9)

Definition 10- *Simple Activity (*lA_i*)*: An ordered sequence of primitive actions. It is specified in (10).

$lA_i: <AD_L,L>$ (10)

Where L is a text string to act as the label for the ongoing activity and AD_L is an activity description for activity L.

Definition 11- *Composite activity*: A collection of two or more simple activities occurring within a given time interval.

Definition 12- *Dynamic composite activities* (*dCA) set*: lists a collection of all sequential, or interleaved and concurrent activities. It is specified in 11.

$dCA: \{dcA_1, dcA_2, ..., dcA_d\}$ (11)

Definition 13- S*ingle dynamic composite activity (*dcA_i*)*: A composite activity that has properties whose values vary in time, implying the notion of change. It is defined in (12).

$dcA_i: < \Phi , \tau, L >, dcA_i \in dCA$ (12)

Where L is a text string to act as a label for the pattern and Φ is a collection of leaf ADLs or a collection of dynamic composite activities such that $\phi \subseteq lADL \cup dCA$. In addition, τ . a subset of C, is the union of temporal contexts for all activities in dcA_i.

To illustrate a dynamic composite activity, consider the activity labelled '*make dinner and watch television'*. We can specify ϕ as ϕ= *{make dinner, watch television}*. In addition, τ can be specified by τ = *{time-interval-of-make-dinner, time-interval-of-watch-television}*.

Definition 14- *Static composite activity (sCA) set:* defines a set of all sequential, or interleaved and concurrent activities as shown in (13).

$sCA: \{scA_1,scA_2,...,scA_g\}$ (13)

Definition 15: A s*ingle static composite activity* (scA_i): This is a composite activity whose properties take values that do not change in time. It is specified as a 3-tuple in (14).

$scA_i: < \Phi , \theta, L> , scA_i \in sCA$ (14)

Where, ϕ is a collection of leaf ADLs or a collection of static composite activities such that $\phi \subseteq lADL \cup sCA$. θ is an aggregate of task contexts associated with contained activities and it is a subset of C.

To illustrate a static composite activity, given the activity '*make dinner and watch television'*, we have ϕ = *{make dinner, watch television}*. Also, θ is specified by task context given by descriptions, i.e., *'make dinner begins'; 'as make dinner continues watch television begins'; 'make dinner continues and ends' and the relationship is parallelism.*

4.2 Ontological activity modelling

Based on the definitions in the previous sub-section and the revised conceptual activity model, we have developed ontological concepts for specifying simple and composite activity models. The key concepts are discussed below and their properties are provided in Table 3. In addition, the DL formulas for selected concepts are provided in Table 4.

4.2.1 General concepts for representing activities, context, and the environment

Activity: This concept is the overall concept for all types of activities.

MonitoredEntity: This is a general concept to represent the set of all entities in the environment occupied by an actor. Each *MonitoredEntity* relates to sub-concepts of *Activity* using the *hasMonitoredEntity* property.

Location: This is a context concept that is used to indicate the location of interest to the actor. For instance, in

a SH environment, it can be used to represent locations like the kitchen, bedroom, bathroom, living room, lounge, etc. Each *Location* concept can relate to the *MonitoredEntity* and *Activity* concepts using the *hasLocation* property.

Sensor: This concept is used to denote the class of all sensors that are deployed in a smart environment. To link sensors to environment entities and objects, instances of *Sensor* are associated with instances of *MonitoredEntity* using the *hasSensor* property. Instances of the Sensor concept are used to encode the runtime state of the SH environment when the user is performing activities. Such runtime information can be used to derive contextual information.

TimeInterval: This concept defines a time interval and indicates the moment of time that a time slice refers to.

TimeSlice: This encodes the temporal extent of activities as a collection of time slices. The temporal extent is specified by associating Activity concept with *TimeSlice* using *timeSliceOf* property. In addition, instances of *TimeSlice* relate to instances of *TimeInterval* through the *hasTimeInterval* property.

4.2.2 Concepts for simple activities

ADLActivity: This concept is the parent concept to all simple activity concepts. All simple activities are defined as subclasses of the *ADLActivity* concept. In general, given that the hypothetical ADL, *SimpleADL*, is a subclass of *ADLActivity*, it can be declared in DL as:

$$\text{SimpleADL} \sqsubseteq \textit{ADLActivity} \sqcap \exists\, \textit{property1.Range1} \sqcap \exists\, \textit{property2.Range2} \ldots \sqcap \exists\, \textit{propertyN.RangeN}$$

TABLE 3: PROPERTIES FOR THE CONCEPTS IN THE ACTIVITY MODELS

Name	Domain	Range	Other properties	Description
hasMonitoredEntity	ADLActivity	MonitoredEntity		Indicates the entities used by an ADLActivity
hasLocation	Activity, MonitoredEntity	Location		Associates Activity and MonitoredEntity with their spatial context.
hasSensor	MonitoredEntity	Sensor		Attaches environment entities to be monitored to the respective sensors.
timeSliceOf	TimeSlice, BasicActivityTS, CompositeActivityTS	StaticCompositeActivity , ADLActivity, DynamicCompositeActivity	Functional	Indicates temporal extent of activity concepts
hasTimeInterval	TimeSlice	TimeInterval	Functional	Associates TimeSlice to TimeInterval
hasOngoingActivity	CompositeActivityTS	ADLActivityTS, CompositeActivityTS	Irreflexive	A dynamic (fluent) property that captures the notion of change.
has Activity	StaticCompositeActivity	ADLActivity, StaticCompositeActivity	Irreflexive	Provides components of a composite activity
startedBy	StaticCompositeActivity	ADLActivity, StaticCompositeActivity	Irreflexive	Indicates starting activity of a composite activity
endedBy	StaticCompositeActivity	ADLActivity, StaticCompositeActivity	Irreflexive	Indicates activity that marks the end of composite activity.
entailsCompositeActivity	StaticCompositeActivity	DynamicCompositeActivity		Used to indicate if a given DynamicCompositeActivity has a corresponding StaticCompositeActivity in the model
relationshipType	StaticCompositeActivity	String	Functional	Indicates whether a SEQUENCE or PARALLEL relation exists

TABLE 4: DL FORMULAS FOR A SELECT SET OF CONCEPTS USED IN ACTIVITY MODELS

- $\textit{MonitoredEntity} \sqsubseteq \exists\, \textit{hasSensor.Sensor} \sqcap \exists\, \textit{hasLocation.Location}$
- $\textit{Activity} \sqsubseteq \exists\, \textit{hasLocation.Location} \sqcap (\textit{ADLActivity} \sqcup \textit{CompositeActivity})$
- $\textit{ADLActivity} \sqsubseteq \textit{Activity} \sqcap \exists\, \textit{hasMonitoredEntity.MonitoredEntity}$
- $\textit{TimeSlice} \sqsubseteq \exists\, \textit{timeSliceOf.Activity} \sqcap =1\ \textit{timeSliceOf} \sqcap \exists\, \textit{hasTimeInterval.Interval} \sqcap =1\ \textit{hasTimeInterval}$
- $\textit{BasicActivityTS} \sqsubseteq \textit{TimeSlice} \sqcap \exists\, \textit{timeSliceOf.ADLActivity} \sqcap =1\ \textit{timeSliceOf}$
- $\textit{CompositeActivityTS} \sqsubseteq \textit{TimeSlice} \sqcap \exists\, \textit{timeSliceOf.DynamicCompositeActivity} \sqcap =1\ \textit{timeSliceOf} \sqcap \exists\, \textit{hasOngoingActivity.TimeSlice} \sqcap \geq 2\ \textit{hasOngoingActivity}$
- $\textit{StaticCompositeActivity} \equiv \exists \textit{hasActivity.}(\textit{ADLActivity} \sqcup \textit{StaticCompositeActivity}) \sqcap \geq 2\ \textit{hasActivity} \sqcap \exists \textit{startedBy.}(\textit{ADLActivity} \sqcup \textit{StaticCompositeActivity}) \sqcap \exists \textit{endedBy.}(\textit{ADLActivity} \sqcup \textit{StaticCompositeActivity}) \sqcap \exists \textit{entailsCompositeActivity.}\ \textit{DynamicCompositeActivity} \sqcap \exists \textit{relationshipType.string}$

In the above example, property1…propertyN are sub-properties of *hasMonitoredEntity*, and Range1…RangeN are sub-concepts of *MonitoredEntity* associated with *SimpleADL*. The sensor observations and context specified in Definitions 2 and 4 are realized by property restrictions that are defined on *ADLActivity*. For example, a typical observation, e.g. '*using the kettle'* can be represented by a *hasKettle* property restriction that is defined on the relevant subclasses of *ADLActivity*. In this example, *hasKettle* is a sub-property of *hasMonitoredEntity*; whereas the concept for *kettle* is a sub-concept of *MonitoredEntity*. Further details on simple activity concepts will be provided in Section 5.1 since the concepts are dependent on the application domain.

BasicActivityTS: This is a sub-class of *TimeSlice* and is used to add a temporal dimension to instances of *ADLActivity*.

4.2.3 Concepts for composite activities

CompositeActivity: This concept represents a dynamic or static composite activity. It is used to denote sequential and interleaved or concurrent activities.

DynamicCompositeActivity: This is a sub-concept of *CompositeActivity* and represents a dynamic composite activity corresponding to Definition 13. Property restrictions to encode change are defined on this concept. For instance the notion of change is represented by implications derived from the fluent property *hasOngoingActivity*. The instances of this concept are intended to be derived at runtime.

CompositeActivityTS: This is a subclass of *TimeSlice* that relates to DynamicCompositeActivity. It explicitly captures the notion of change by defining a restriction on the fluent property *hasOngoingActivity*. Essentially, objects of *CompositeActivityTS* associate with objects of *BasicActivityTS* or another *CompositeActivityTS* through the fluent property *hasOngoingActivity* to denote change. Each composite activity can be derived from the activities whose *TimeSlice* objects have been associated with the *hasOngoingActivity* over a given time interval. It associates with *DynamicCompositeActivity* using the *timeSliceOf* property.

DynamicConcurrentActivity: Sub-concept of *DynamicCompositeActivity* whose instances are dynamic concurrent or interleaved activities.

DynamicSequentialActivity: Sub-concept of *DynamicCompositeActivity* whose instances are dynamic sequential activities.

StaticCompositeActivity: This is a sub-concept of *CompositeActivity* and defines a static composite activity as per Definition 15. It simply captures the activities (whether simple or composite) that constitute a composite activity. It typically captures inter-activity relations using the *hasActivity* property to specify the activities that constitute the composite activity. It associates with *DynamicCompositeActivity* through the *entailsCompositeActivity* property. This concept can specify a time slice but does not encode the notion of change, hence, it relates to *TimeSlice* concept using *timeSliceOf* property.

StaticConcurrentActivity: Sub-concept of *StaticCompositeActivity* whose instances are static concurrent or interleaved activities.

StaticSequentialActivity: Sub-concept of *StaticCompositeActivity* whose instances are static sequential activities.

Figure 2 illustrates how the various concepts relate to obtain activity models for simple and composite

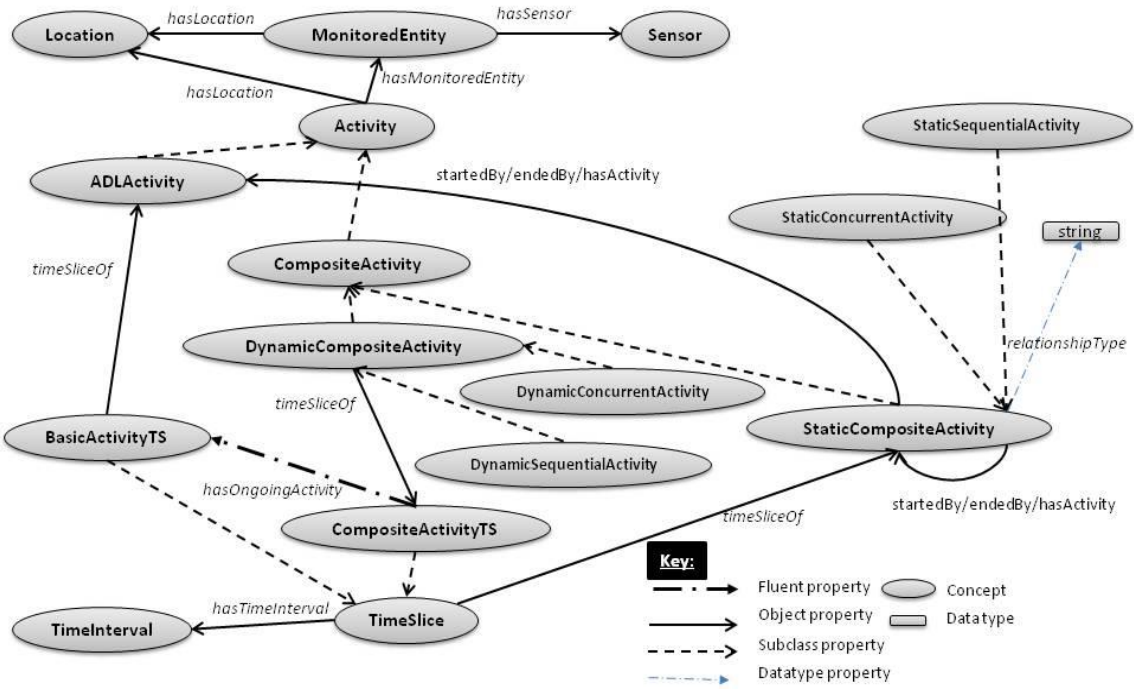

Figure 2: Part of the activity models showing concepts and their inter-relationships

activities.

4.3 Interval logic-based temporal activity modelling

In addition to ontological modelling of relationships between activities and entities described above, we use Allen interval relations to model temporal relationships between simple activities of composite activities. The models show how to relate temporal intervals of composite activities to the temporal intervals of their composing activities. The resulting models can be used to infer composite activities from temporal intervals of simple activities or other composite activities.

4.3.1 Models of sequential composite activities

Sequential activities are modelled by associating their respective intervals using the Allen relations *before/after* and *meets/met-by*. The relation before/after signifies that there is a gap between the two intervals, while meets/met-by indicates that the two intervals follow each other with no gap between them. These two relations (marked by solid arrows) and their implications (marked by dotted arrows) are represented in Figure 3 (a).

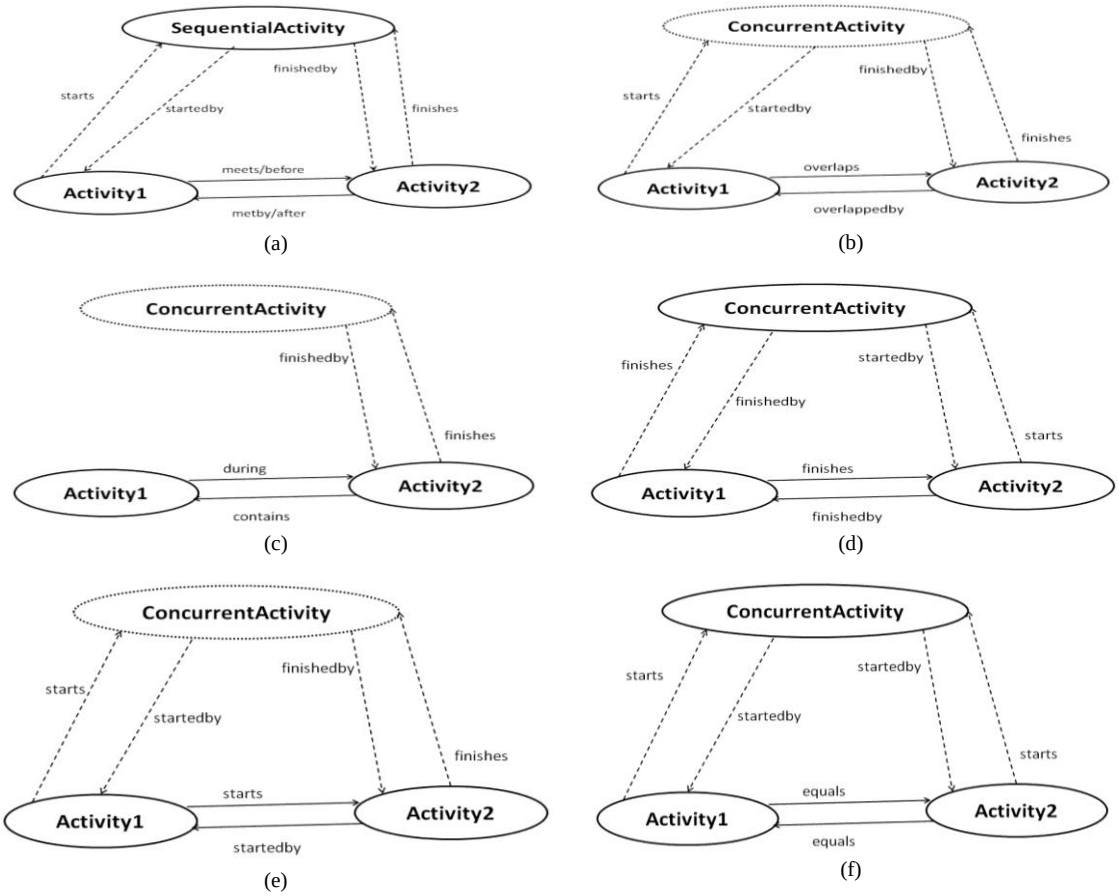

Figure 3: Temporal relationship models of composite activities (a) before/after and meets/met-by; (b) overlaps/overlapped-by; (c) during/contains; (d) finishes/finished-by; (e) starts/started-by;(f) equals models

4.3.2 Models of interleaved and concurrent composite activities

The models of interleaved and concurrent activities encode the notion that activities can occur simultaneously only if their time intervals overlap fully or partially. The models of interleaved and concurrent activities are created using nine temporal relations, i.e., overlaps*/overlapped-by*, *during/contains, starts/started-by, finishes /finished-by and equals* as described below. The resulting temporal models are shown in Figure 3 (b)-(f):

i) *Overlaps/overlapped-by*- this shows that two activities have components of their intervals that are shared, but with one interval starting or ending before the other interval.
ii) *Contains /during*- this models a composite activity made up of simple activities, e.g. 'prepare meal' that contains 'prepare soup' and 'prepare vegetable'. The longer interval encloses the shorter one.
iii) *Starts/started-by*-shows the simple/composite activity that starts another simple/composite activity.
iv) *Finishes /finished-by*- shows the simple/composite activity that finishes another simple/composite activity.

v) *Equals*- Theoretically, this scenario only applies to concurrency by parallelism. The two intervals start and end at the same time.

4.4 Entailment rules for activity modelling

The previous section describes interval-based temporal interrelationships between simple activities of a composite activity using ontological concepts. Nevertheless, the mechanism for interpreting the temporal relationships is still missing, which is required in order to infer composite activities. To this end, we have defined a set of entailment rules as an essential part of the composite model, based on Semantic Web Rule Language (SWRL) [45], which can infer complex dependencies among activities and therefore the ongoing composite activities. These rules can be used to derive the ongoing composite activities by identifying the existing relationships between temporal intervals of ongoing activities. Three categories of entailment rules have been designed, namely, *rules to derive interval relations and assert dynamic composite activities*; *rules to assert instances of fluent property*; and *rules to derive and assert static composite activities*. Due to limitation of space, in the following we use three entailment rules based on the *overlaps/overlapped-by* relationship to illustrate the development of rules and their use for activity inference.

```
ProperInterval(?x), ProperInterval(?y), before(?a, ?c), before(?b, ?d),
before(?c, ?b), hasBeginning(?x, ?a), hasBeginning(?y, ?c), hasEnd(?x, ?b),
hasEnd(?y, ?d) -> intervalOverlaps(?x, ?y)
```

(a)

```
ADLActivity(?ax), ADLActivity(?ay), ComplexActivityTS(cplxConcurrentTS),
DynamicConcurrentActivity(dynConcurrentActivity), TimeSlice(?tsx),
TimeSlice(?tsy), Interval(dynConcurrentInterval), ProperInterval(?x),
ProperInterval(?y), hasTimeInterval(?tsx, ?x), hasTimeInterval(?tsy, ?y),
hasTimeInterval(cplxConcurrentTS, ?w), timeSliceOf(?tsx, ?ax),
timeSliceOf(?tsy, ?ay), timeSliceOf(cplxConcurrentTS, dynConcurrentActivity),
hasBeginning(?x, ?a), hasEnd(?y, ?d), intervalOverlaps(?x, ?y) ->
hasOngoingActivity(cplxConcurrentTS, ?tsx),
hasOngoingActivity(cplxConcurrentTS, ?tsy), hasBeginning(?w, ?a), hasEnd(?w, ?d),
intervalFinishes(?y, ?w), intervalStarts(?x, ?w)
```

(b)

```
ComplexActivityTS(?tsw), ConcurrentActivity(?sa), DynamicConcurrentActivity(?aw),
TimeSlice(?tsx), TimeSlice(?tsy), endedBy(?sa, ?ay), hasActivity(?sa, ?ax),
hasActivity(?sa, ?ay), hasOngoingActivity(?tsw, ?tsx),
hasOngoingActivity(?tsw, ?tsy), hasTimeInterval(?tsw, ?w),
hasTimeInterval(?tsx, ?x), hasTimeInterval(?tsy, ?y), startedBy(?sa, ?ax),
timeSliceOf(?tsw, ?aw), timeSliceOf(?tsx, ?ax), timeSliceOf(?tsy, ?ay),
intervalFinishes(?y, ?w), intervalOverlaps(?x, ?y), intervalStarts(?x, ?w),
relationshipType(?sa, "PARALLEL") -> entailsCompositeActivity(?aw, ?sa).
```

(c)

Figure 4: Entailment rules for inferring composite activities

4.4.1 Derive interval relations and assert dynamic composite activity intervals

Given two existing intervals for a pair of primary activities that have a qualitative temporal relationship, the rules in Fig 4(a) and (b) are used to assert interval end-points of a dynamic composite activity. The rule in Fig. 4(a) derives the interval relation intervalOverlaps by using the interval end-points of two primary activities. Fig. 4(b) provides the rule for obtaining the inferred values of intervalOverlaps property. The left hand side (LHS) of the rule in Fig. 4(b) obtains three TimeSlice objects, and determines the beginning and ending points for each primary activity's interval. Finally, it derives the existing interval relationship for the primary activities. The right hand side (RHS) of the rule uses the facts established on LHS to assert the beginning and ending points of the time interval for the dynamic composite activity.

4.4.2 Assert instances of fluent property

This is based on Fig. 4(b), and the rule allows the TimeSlice objects linked to the ongoing primary activities to be related with the TimeSlice object of the DynamicCompositeActivity through the fluent property, hasOngoingActivity. The LHS obtains three TimeSlice objects, i.e., the two for primary activities and one for the dynamic composite activity, checks for the temporal dependency between the primary activities, and asserts instances of the fluent property. If two TimeSlice objects share a temporal relation, then they are associated with the TimeSlice object of the dynamic composite activity using the fluent property.

4.4.3 Derive and assert static composite activities

This is based on Fig. 4(c) above and the rule's LHS checks that there exists an instance of DynamicCompositeActivity, an instance of StaticCompositeActivity, as well as instances of the fluent property that are defined on the former's instance. The RHS then uses the facts established by the LHS to assert instances of the entailsCompositeactivity that is defined as a property of the concept StaticCompositeActivity (see Table 3). Essentially, this rule is used to infer and validate the ongoing composite activity that is subsequently reported to the user. Validation fails if instances of DynamicCompositeActivity do not have corresponding instances of StaticCompositeActivity. Whenever validation fails the resulting composite activity described by the instance of DynamicCompositeActivity should be suggested for addition into the static model of activities.

Similar rules are defined for other qualitative temporal relations (i.e., equals, during/contains, starts/started-by, finishes/finished-by, before/after, and meets/met-by) but due to space limitations we do not provide them here.

5 Simple and Composite Activity Recognition

5.1 ADL Ontology

To support SH-based activity recognition, we have aggregated the models created above including concepts and properties for simple and composite activities, and the entailment rules to form the ADL Ontology. The ADL ontology is constructed based on common ADL activities related to *food preparation*, *recreation*, and *hygiene*. Table 5 shows a list of simple ADL activity models, namely ADL concepts, their properties, and the range concepts representing the objects used by the users to perform respective ADLs. All the Range concepts are sub-concepts of *MonitoredEntity*. The super concepts are listed in the order from the immediate super-concept to the most general super-concept. These simple activities can form composite activities if a user performs them in sequence or concurrently, as discussed in Section 6.

TABLE 5: SUMMARY OF ADL CONCEPTS IN THE ADL ONTOLOGY

Concept	Super Concept	Property	Range Concepts
HaveBath	BathRoomADL, BasicADLActivity, ADLActivity, Activity	hasHygieneItem	BathBrush, BathSoap, BodyWash, Towel
		hasHygieneAppliance	BathTap
		hasLocation	BathRoom
BrushTeeth	BathRoomADL, BasicADLActivity, ADLActivity, Activity	hasHygieneAppliance	Toothbrush, WashingSinkTap
		hasHygieneItem	Toothpaste, MouthWash
		hasLocation	BathRoom
WashHands	BathRoomADL, BasicADLActivity, ADLActivity, Activity	hasHygieneAppliance	WashingSinkTap
		hasHygieneItem	Towel, HandWash
		hasLocation	BathRoom
MakeTea	MakeHotDrink, MakeDrink, KitchenADL, FunctionalADLActivity, ADLActivity, Activity	hasFlavoring	Milk, Sugar
		hasContainer	Cup
		hasHotDrinkType	Tea
		hasLocation	Kitchen
MakeChocolate	MakeHotDrink, MakeDrink, KitchenADL, FunctionalADLActivity, ADLActivity, Activity	hasFlavoring	Milk, Sugar
		hasContainer	Cup
		hasHotDrinkType	Chocolate
		hasLocation	Kitchen
MakeCoffee	MakeHotDrink, MakeDrink, KitchenADL, FunctionalADLActivity, ADLActivity, Activity	hasFlavoring	Milk, Sugar
		hasContainer	Cup
		hasHotDrinkType	Tea
		hasLocation	Kitchen
MakePasta	MakeHotMeal, MakeMeal, KitchenADL, FunctionalADLActivity,ADLActivity,Activity	hasFlavoring	Salt
		hasContainer	Plate
		hasCookingAppliance	Cooker
		hasHygieneAppliance	KitchenSinkTap
		hasMaterial	Pasta
		hasUtensil	Drainer, Pan
		hasLocation	Kitchen
WatchTelevision	LoungeADL, RecreationalADLActivity, ADLActivity, Activity	hasEntertainmentAppliance	TV
		hasEntertainmentAppliance	TVRemote
		hasFurniture	Sofa
		hasLocation	Lounge

5.2 Activity recognition architecture

Whenever a user performs activities along a timeline, the sensor data stream can be analyzed to recognize the ongoing activities based on the extracted contextual information. A sensor data stream can be partitioned to

obtain segments of sensor data and associated contextual information. To partition a sensor data stream, the time window based segmentation method described in [46] has been used to support dynamic segmentation. The segmentation approach uses information from the activity ontologies, e.g. activity duration, and feedback from activity inference. It generates segments from streaming sensor data that can be further analyzed to infer the ongoing activities.

We propose the architecture shown in Figure 4 to support both simple and composite activity recognition. The architecture supports three main tasks, namely, activity modelling, activity recognition, and sensor data stream segmentation. In activity modelling, two types of models (i.e., static and dynamic models as previously described) are created as described in Section 4. To create the two models, activity modelling involves two processes, i.e., static activity modelling and dynamic activity modelling. Static activity modelling involves creating static models of simple and composite activities. Dynamic activity modelling is used to create the dynamic model of composite activities. Activity recognition is modeled as three interdependent tasks, namely, action recognition, simple activity recognition, and composite activity recognition. The action recognition task is tightly coupled with and subsumed in the simple activity recognition task. By separating activity recognition into interdependent tasks, it is possible to use different techniques for each task. In this work, instance retrieval, subsumption reasoning and equivalence reasoning are used for action and simple activity recognition. For composite activity recognition, rule-based inference techniques are exploited. Finally, segmentation is used to aid real-time processing by supporting online segmentation of streaming sensor data. The resulting segments are processed during activity recognition and mapped to corresponding simple or composite activities.

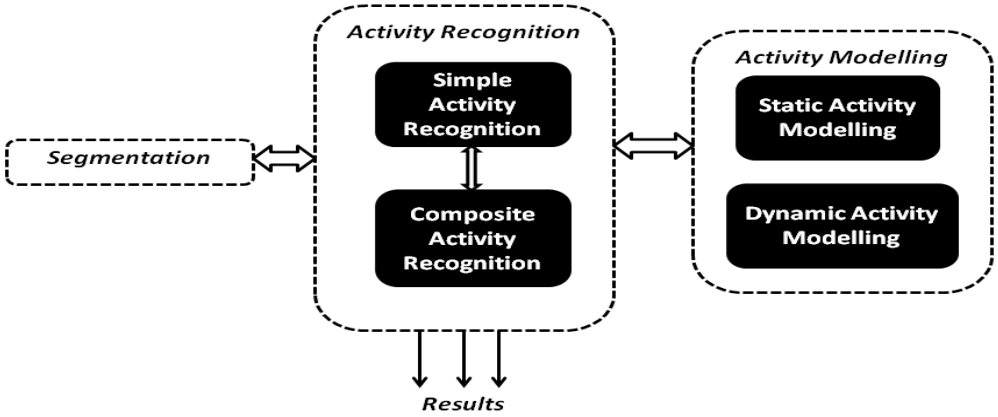

Figure 5: Architecture for activity modelling and recognition

5.3 Activity recognition algorithm

Given a segment of sensor data stream, we describe the algorithm that derives the corresponding simple or composite activities. The steps described here are summarized in the algorithm listing in Figure 5.

In the first step, the enclosed observations are converted into primitive actions by checking property restrictions specified in the *ADL Ontology* through ontological reasoning. The second step groups primitive actions into one or more activity descriptions corresponding to the simple activities that are defined in the *ADL Ontology*. Each activity description is constructed such that it contains only sensor data and context information that can be mapped to a single specific simple activity or to a general class of simple activities. For instance an activity description can match the definition of the general class '*MakeHotDrink'* (a super activity for '*MakeTea*, '*MakeCoffee'*, and '*MakeChocolate'*) or that of the simple class 'MakeTea'. Therefore, an activity description is considered more general if it corresponds to a general class of activities and more specific otherwise. A more general activity description signifies that more sensor data and context information is still needed to identify the correct sub-activity. To group primitive actions, the algorithm checks the ontology to determine the ADL concepts that are in the domain of the corresponding property restriction. For instance, given the knowledge that the user is in the kitchen, the algorithm can obtain the triples <*MakeTea, hasLocation, kitchen*>, <*MakeCoffee, hasLocation, kitchen*>, and <*MakeChocotate, hasLocation, kitchen*>, and so on, for all the activities that can take place in the kitchen. As a result it will form a number of partial activity descriptions with one description per possible activity and each activity will be associated with the primitive action representing '*in the kitchen'*. Since each activity description contains only sensor data and context information corresponding to an activity or class of activities, several activity descriptions can be generated for a given segment of the sensor data stream. As more sensor data is obtained new activity descriptions will be created or the existing ones will be modified and enriched. For example, if the next sensor observation relates to the user using the *tea bag*, only the partial activity description corresponding to *MakeTea* will be updated to accommodate the new primitive action. During this process, a few partial activity descriptions will become complete when all the relevant primitive

actions have been executed. However, all the existing partial or complete activity descriptions will remain valid until sensor observations within a given segment are discarded.

INPUT: sensor data stream (Ω), ADL ontology (ADL-O), inference rule base (RB)

OUTPUT: Composite activity (CA) or simple activity (SA)

```
RECOGNIZE-ACTIVITY (Ω, ADL-O, RB)
BEGIN:
    WHILE data stream is active DO

        Segment data stream (Ω) into a set of segments S={s1, s2, ..., sn}
        /*for each segment map observations to activities*/
        FOR each si∈S DO
            /*Create activity descriptions to form set AD={AD1, AD2,...ADk}*/
            Extract observations from segment si, O={o1,o2,...oh}
            FOR each oi∈O DO //for each observation
                Retrieve all activities described by oi,  A(oi)={A1, A2, ...An}
            ENDFOR
            Create a set of all activities A=A(o1)⊔ …A(oi)⊔ … A(oh)
            FOR each x∈A DO/*for all activities*/
                Collect all observations constituting the definition of x as activity descriptions ADx
                Add ADx to AD
            ENDFOR
            /*classify activity*/
            FOR each ADi∈AD DO
                Map ADi to a simple activity /*use ontological inference*/
                IF a leaf activity is returned THEN
                    Report it (SA)
                ELSE
                    IF goal is still valid THEN
                        Wait for updated activity description (go to start of current loop)
                    ELSE Communicate status report and terminate
                    ENDIF
                ENDIF
                Update classification status into recognition status RS= {st1,st2,...stm}
            ENDFOR
            /*to help aggregate results for composite activities*/
            Define the set of time intervals TI
            FOR each sti∈RS DO
                Obtain temporal interval Ii and add it to TI
            ENDFOR
            IF only one interval is present THEN
                Report it (SA) /*SA-activity in interval I*/
            ELSE
                Infer interval relations using RB
                Derive ongoing composite activity relationships
                Check corresponding instances of static composite activities
                IF instance in static model THEN
                    Report it (CA)
                ELSE
                    Recommend the activity ontology be updated to accommodate it
                ENDIF
            ENDIF
            IF results are conclusive THEN
                Convey results (SA or CA)
            ELSE
                Update segment and window
            ENDIF
        ENDFOR
    ENDFOR
```

In the third step, simple activity recognition is performed to map activity descriptions into activity labels. To perform simple activity recognition, we adopt and modify the ontological reasoning approach described in [9, 10]. The original approach progressively aggregates sensor data within a data stream segment and performs subsumption and equivalence reasoning to infer the entailed activity. In the modified approach, the algorithm processes each activity description obtained as previously described against the *ADL Ontology*. Basically, it compares each activity description with activity models in the *ADL Ontology* using semantic reasoning and the activity label for the model that is closest to the activity description is reported as the ongoing simple activity. The activity model returned by instance retrieval is considered the closest model. In the absence of a model returned by instance retrieval, then the model returned by equivalence reasoning is taken as the closest. Otherwise, the model returned by subsumption is the closest. Given the possibility of multiple activity descriptions per data stream segment, parallel simple activity recognition processes can be initiated with each process dedicated to a single activity description.

The fourth and last step performs composite activity recognition by using the inference rules to aggregate the results of simple activity recognition. The strategy is to progressively aggregate the results of simple activity recognition in order to recognize composite activities. If only one simple activity has been identified for a sensor data segment, this can be reported to the user. Alternatively, if more than one simple activity is identified from corresponding activity descriptions, the results are processed to determine if ongoing simple activities share qualitative temporal relationships. The simple activities that share qualitative temporal relationships are inferred as components of a composite activity. However, before the composite activity is reported to the relevant applications, the ontology is checked for a corresponding instance in the static activity model. If a corresponding instance exists, it is reported to the user; otherwise, it is considered a novel composite activity and recommended for inclusion in the ontology. To perform this analysis, the approach uses temporal inference rules. The rules can infer qualitative temporal relationships and derive corresponding composite activities from the activity models using the temporal inference mechanism.

6 EXPERIMENTS AND EVALUATION

6.1 System prototype

To support experiments and evaluation, we developed a system prototype for simple and composite activity recognition. The prototype consists of the ADL Ontology and a multi-agent system. The multi-agent system consist is built using Java Agents Development Framework (JADE) [47]. Overall, four types of agents were implemented as follows: *a) an agent to receive sensor input and to segment the sensor data stream; b) an agent to manage action inference, generate activity descriptions, and to oversee summarization of activity information; c) an agent to manage inference rule execution to aggregate the results from simple activity recognition to infer composite activities and, finally; d) multiple agents to infer simple activities*. The benefit of choosing agent as an implementation artefact is because agents provide the different components with autonomy needed to perform their respective tasks. In addition, each component can continuously review and react to changes in its goals. Also, there is massive parallelism involved in executing the various tasks involved in the framework and these tasks are implemented as agent behaviours. The ADL Ontology was built using OWL 2[48] constructs in Protégé [49] ontology editor. The prototype implements Java-based code interacting with the Pellet [50] OWL reasoner to provide ontological reasoning support. The inference rules were implemented as Semantic Web Rule Language (SWRL) [45] rules. To facilitate the execution of the inference rules we translated the activity ontology and the SWRL rules to Java Expert System Shell (JESS) [51] fact and rule bases, respectively. We used the OWL2Jess and SWRL2Jess translators based on [52]. In the prototype, the JESS fact and rules bases are accessed and processed by a JESS rule engine. The engine is accessed by behaviours in the JADE agent responsible for aggregating the results of simple activity recognition. A segment of the activity ontology and a runtime snapshot of the agent system are shown in Figure 6.

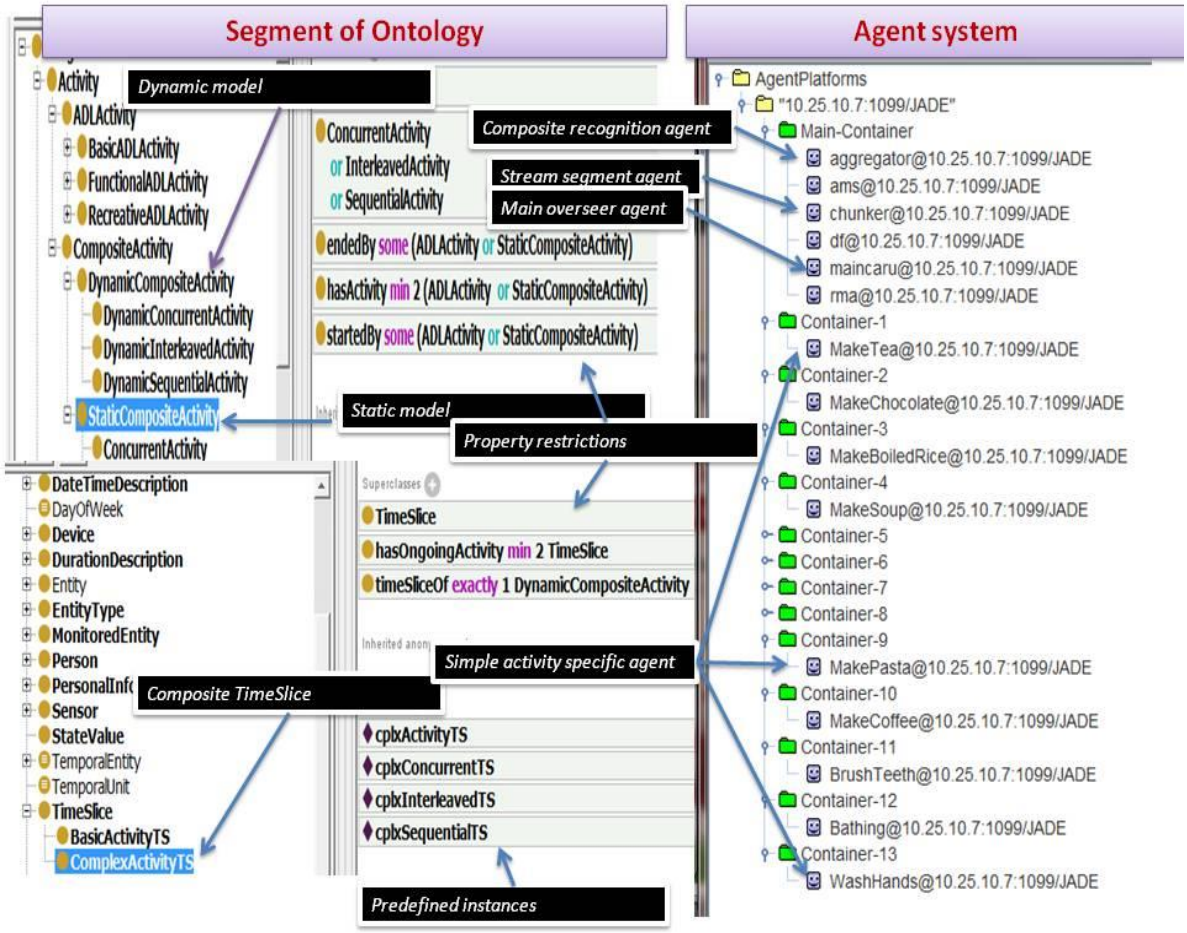

Figure 7: Snapshot of activity ontology and runtime agent system

6.2 Experiment design

To evaluate and demonstrate the feasibility of the proposed approach, we used the synthetic data generator developed and described in [46] to generate synthetic ADL data. To generate the synthetic ADL data, seven typical simple ADLs related to meals (e.g. *MakeTea*, *MakeCoffee*, *MakeChocolate*, and *MakePasta*), hygiene (e.g. *HaveBath*, *WashHands*) and recreation (e.g. *WatchTelevision*) were used. The synthetic ADL data possesses the necessary temporal information and allows us to evaluate the feasibility of the developed approach. To generate synthetic ADL data, we specified 'seed' ADL patterns for both simple activities (e.g. *MakeTea*, *HaveBath*) and composite activities (e.g. *MakePastaAndMakeTea*). The synthetic ADL data generator then derives different permutations of these patterns. To select the permutation to use, it uses a random number generator to guarantee fairness in pattern selection. The transition time (in seconds) between ADLs is specified for each ADL pattern. For example, the pattern *WashHands-0, MakePastaAndMakeTea-600*, implies that *WashHands* is the first ADL in the pattern, while the ADL *MakePastaAndMakeTea* will occur 600 seconds after *WashHands* is completed. As described in [46], one or more patterns of sensor activations is provided for each simple ADL with each sensor in the pattern activated after a specific amount of time. It is therefore possible to derive the approximate activity duration from the temporal information associated with the sensor activations.We generated eight weeks of synthetic ADL data consisting of 56 episodes of simple or composite activities. A total of 104 activities were generated consisting of 23 interleaved and concurrent activities (46 simple activities), 25 sequential activities (50 simple activities), and eight standalone simple activities to provide the ground truth. Table 6 presents an analysis of all simple activities while Table 7 provides a summary of composite activities.

TABLE 6. SUMMARY OF SIMPLE ACTIVITIES IN SYNTHETIC DATA SET

Simple activity	#in parallel	#in sequential	#standalone	Sub-total
MakeTea	8	18	0	26
MakeCoffee	1	0	2	3

MakeChocolate	1	0	6	7
MakePasta	18	7	0	25
HaveBath	8	7	0	15
WashHands	0	10	0	10
WatchTelevision	10	8	0	18
Total				**104**

TABLE 7. SUMMARY OF COMPOSITE ACTIVITIES IN SYTHETIC DATA SET

Concurrent and interleaved	**# of occurrences**	**Sequential**	**# of occurrences**
MakePasta and MakeTea (a)	3	MakePasta then HaveBath(g)	6
MakePasta and WatchTelevision(b)	5	MakeTea then WashHands(h)	4
MakePasta and HaveBath(c)	8	WashHands then MakeTea(i)	6
WatchTelevision and MakeTea (d)	5	MakeTea then WatchTelevision (j)	3
MakePasta and MakeChocolate (e)	1	WatchTelevision then MakeTea(k)	5
MakePasta and MakeCoffee(f)	1	HaveBath then MakePasta(l)	1
Total	**23** ***(46 simple activities)***	**Total**	**25** ***(50 simple activities)***

6.3 Experiments and results

To test the approach and associated algorithms for activity recognition, we use the simulation tool presented in [46] to play back the synthetic ADL data described above. The sensor data is then fed to the activity recognition system as if the sensor activations are occurring in real-time. As the data is played back, the recognition system will attempt to identify the ongoing simple or composite activities. A total of 104 activities were played back in real-time and processed by the system prototype for activity recognition. The overall accuracy value obtained for simple activities is 100% since all 104 simple activities were successfully recognized. Figure 7 shows the precision, recall, and accuracy values for composite activities. An overall accuracy value of 88.26% was obtained.

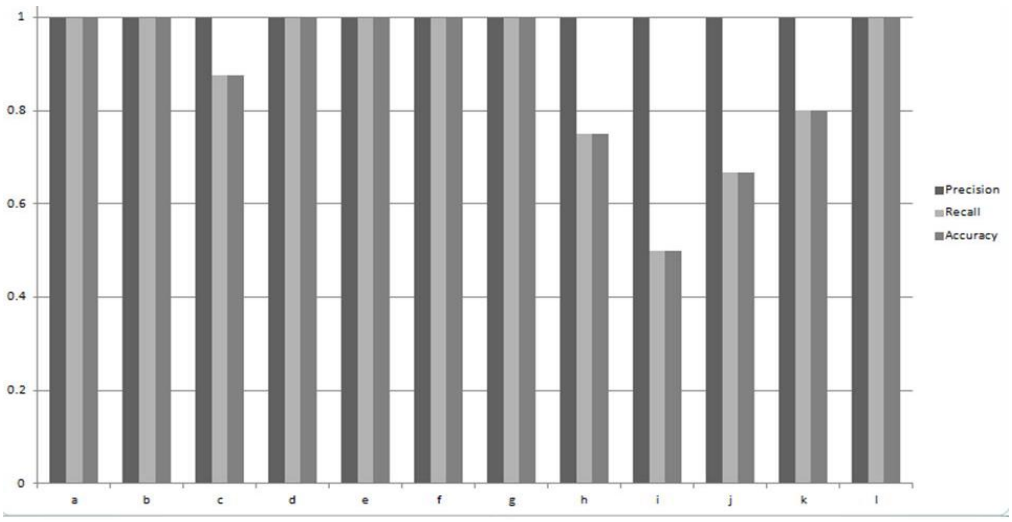

Figure 8: Summary of results for composite activities

6.4 Discussion

We observed with interest that the accuracy for simple activities was 100%. We attribute this to the decision we made to derive all possible activity descriptions for each data stream segment. By deriving activity descriptions as presented by the algorithm in Figure 5, the approach guarantees that only sensor observations that are relevant to a particular type of activity are included in its activity recognition. Essentially, primitive actions are used to derive activity descriptions before mapping the resulting descriptions to activity labels. As a result each simple activity recognition unit can correctly classify its activity based on the sensor observations that it obtains.

The overall recognition accuracy for composite activities is 88.26% which is quite encouraging. It is important to note that recognition accuracy is lower when the composite activity consists of activities that are

executed in sequence. This can be attributed to the transitions between activities and how well the system keeps track of the previously recognized activities. We believe that this can be increased by using feedback from composite activity recognition in segmentation. On the other hand, impressive accuracy results for concurrent and interleaved activities can be attributed to the absence of temporal transitions between the activities involved.

7 CONCLUSIONS AND FUTURE WORK

This paper presented a hybrid approach and associated system architecture, models, methods and algorithms for composite activity modelling and recognition. The approach combines ontological and temporal knowledge representation formalisms to provide required modelling and representation capabilities for composite activity modelling. In this paper, we have developed a generic conceptual activity model that encodes the characteristics of simple and composite activities and from which activity models can be specified. We have developed a unified activity recognition algorithm that processes streaming sensor data against composite activity models to support the identification of both simple and composite activities, e.g. interleaved and concurrent activities. We have described an integrated system architecture for composite activity recognition in smart homes and further developed a system prototype that was used to evaluate the approach. Using the prototype we have conducted well-designed experiments which have observed average accuracy values of 100% and 88.26% for simple and composite activities, respectively. To the best of our knowledge, this is the first effort in related research communities that uses a purely knowledge-driven approach that addresses both temporal representation and reasoning requirements to recognize both simple and composite activities. We believe this research enriches the literature and advances the research frontiers of the knowledge-driven approach to activity recognition.

Future work should involve extending the conceptual activity model with additional spatial features to enhance the quality of context information used in activity inference. The use of spatial features is intended to increase the ability to use spatial context to discern activity relationships. Also, we will perform further experiments to assess the computational performance of the recognition algorithms in terms of complexity and soundness, completeness, and decidability of reasoning.

References

[1] M. Chan, E. Campo, D. Esteve and J. Fourniols, "Smart homes - Current features and future perspectives," *Maturitas,* vol. 64, pp. 90-97, 10/20, 2009.

[2] D. J. Cook, H. Hagras, V. Callaghan and A. Helal, "Making our environments intelligent," *Pervasive and Mobile Computing,* vol. 5, pp. 556-557, 2009.

[3] A. B. James, "Activities of daily living and instrumental activities of daily living," in *Occupational Therapy.*, 10th ed., E. B. Crepeau , E. S. Cohn and B. B. Schell, Eds. Philadelphia: Lippincott, Williams and Wilkins.: Willard and Spackman, 2008, pp. 538-578.

[4] J. Modayil, T. Bai and H. Kautz, "Improving the recognition of interleaved activities," in *10th International Conference on Ubiquitous Computing, UbiComp 2008, September 21, 2008 - September 24,* 2008, pp. 40-43.

[5] T. Gu, L. Wang, Z. Wu, X. Tao and J. Lu, "A Pattern Mining Approach to Sensor-Based Human Activity Recognition," *IEEE Trans. Knowled. Data Eng.,* vol. 23, pp. 1359-72, 2011.

[6] D. J. Patterson, D. Fox, H. Kautz and M. Philipose, "Fine-grained activity recognition by aggregating abstract object usage," in *Ninth IEEE International Symposium on Wearable Computers,* 2005, pp. 44-51.

[7] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz and D. Hahnel, "Inferring activities from interactions with objects," *IEEE Pervasive Computing,* vol. 3, pp. 50-57, 2004.

[8] T. Van Kasteren, A. Noulas, G. Englebienne and B. Krose, "Accurate activity recognition in a home setting," in *10th International Conference on Ubiquitous Computing, UbiComp 2008, September 21, 2008 - September 24,* 2008, pp. 1-9.

[9] L. Chen, C. Nugent and H. Wang, "A Knowledge-Driven Approach to Activity Recognition in Smart Homes," *Knowledge and Data Engineering, IEEE Transactions on,* vol. PP, pp. 1-1, 2011.

[10] L. Chen and C. Nugent, "Ontology-based activity recognition in intelligent pervasive environments," *International Journal of Web Information Systems,* vol. 5, pp. 410-30, 2009.

[11] L. Chen, C. Nugent, M. Mulvenna, D. Finlay, X. Hong and M. Poland, "Using event calculus for behaviour reasoning and assistance in a smart home," in *6th International Conference, ICOST 2008,* 2008, pp. 81-9.

[12] S. McKeever, J. Ye, L. Coyle, C. Bleakley and S. Dobson, "Activity recognition using temporal evidence theory," *Journal of Ambient Intelligence and Smart Environments*, vol. 2, pp. 253-269, 2010.

[13] H. Storf, M. Becker and M. Riedl, "Rule-based activity recognition framework: Challenges, technique and learning," in *2009 3rd International Conference on Pervasive Computing Technologies for Healthcare - Pervasive Health 2009, PCTHealth 2009, April 1, 2009 - April 3*, 2009, .

[14] R. Helaoui, M. Niepert and H. Stuckenschmidt, "Recognizing interleaved and concurrent activities: A statistical-relational approach," in *9th IEEE International Conference on Pervasive Computing and Communications, PerCom 2011, March 21, 2011 - March 25*, 2011, pp. 1-9.

[15] R. Helaoui, M. Niepert and H. Stuckenschmidt, "Recognizing interleaved and concurrent activities using qualitative and quantitative temporal relationships," in 2011, pp. 660-670.

[16] H. J. Steinhauer, S. Chua, H. W. Guesgen and S. Marsland, "Utilising temporal information in behaviour recognition," in *2010 AAAI Spring Symposium, March 22, 2010 - March 24*, 2010, pp. 54-59.

[17] T. Huynh, U. Blanke and B. Schiele, "Scalable recognition of daily activities with wearable sensors," in *Third International Symposium, LoCA 2007*, 2007, pp. 50-67.

[18] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King and R. Friedman, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," in *Eleventh IEEE International Symposium on Wearable Computers*, 2007, pp. 37-40.

[19] E. M. Tapia, S. S. Intille and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *Proceedings*, 2004, pp. 158-75.

[20] Seon-Woo Lee and K. Mase, "Activity and location recognition using wearable sensors," *IEEE Pervasive Computing*, vol. 1, pp. 24-32, 07, 2002.

[21] B. Bouchard, S. Giroux and A. Bouzouane, "A smart home agent for plan recognition of cognitively-impaired patients," *Journal of Computers*, vol. 1, pp. 10, 08, 2006.

[22] S. Chua, S. Marsland and H. W. Guesgen, "Spatio-temporal and context reasoning in smart homes," in *International Conference on Spatial Information Theory*, 2009, pp. 9-20.

[23] J. C. Augusto and C. D. Nugent, "The use of temporal reasoning and management of complex events in smart homes," in *European Conference on Artificial Intelligence*, 2004, pp. 778.

[24] F. Lafti, B. Lefebvre and C. Descheneaux, "Ontology-based management of the telehealth smart home, dedicated to elders in loss of cognitive autonomy," in *OWLED 2007 Workshop on OWL: Experiences and Directions*, 2007, .

[25] V. Osmani, S. Balasubramaniam and D. Botvich, "A bayesian network and rule-base approach towards activity inference," in *VTC '07/Fall*, 2007, pp. 254-8.

[26] T. Wu, C. Lian and J. Y. Hsu, "Joint recognition of multiple concurrent activities using factorial conditional random fields," in *2007 AAAI Workshop, July 23, 2007 - July 23*, 2007, pp. 82-87.

[27] D. Hao Hu, S. J. Pan, V. W. Zheng, N. N. Liu and Q. Yang, "Real world activity recognition with multiple goals," in *10th International Conference on Ubiquitous Computing, UbiComp 2008, September 21, 2008 - September 24*, 2008, pp. 30-39.

[28] D. H. Hu and Q. Yang, "CIGAR: Concurrent and interleaving goal and activity recognition," in *23rd AAAI Conference on Artificial Intelligence and the 20th Innovative Applications of Artificial Intelligence Conference, AAAI-08/IAAI-08, July 13, 2008 - July 17*, 2008, pp. 1363-1368.

[29] G. Okeyo, L. Chen, H. Wang and R. Sterritt, "A hybrid ontological and temporal approach for composite activity modelling," in *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2012, pp. 1763-70.

[30] G. Okeyo, L. Chen, H. Wang and R. Sterritt, "A knowledge-driven approach to composite activity recognition in smart environments," in *6th International Conference on Ubiquitous Computing and Ambient Intelligence, UCAmI 2012, December 3, 2012 - December 5*, 2012, pp. 322-329.

[31] D. Riboni and C. Bettini, "OWL 2 modeling and reasoning with complex human activities," *Pervasive and Mobile Computing*, vol. 7, pp. 379-95, 06, 2011.

[32] J. F. Allen, "Maintaining knowledge about temporal intervals," *Commun ACM*, vol. 26, pp. 832-43, 11, 1983.

[33] Saguna, A. Zaslavsky and D. Chakraborty, "Recognizing concurrent and interleaved activities in social interactions," in *CGC, December 12, 2011 - December 14*, 2011, pp. 230-237.

[34] A. Padovitz, S. W. Loke and A. Zaslavsky, "Towards a theory of context spaces," in *Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004, pp. 38-42.

[35] C. Welty and R. Fikes, "A reusable ontology for fluents in OWL," in *Fourth International Conference on Formal Ontology in Information Systems*, 2006, pp. 226-236.

[36] S. Katz, T. D. Downs, H. R. Cash and R. C. Grotz, "Progress in Development of Index of Adl," *Gerontologist*, vol. 10, 1970, 1970.

[37] M. P. Lawton and E. M. Brody, "Assessment of older people: self-maintaining and instrumental activities of daily living." *Gerontologist*, vol. 9, 1969, 1969.

[38] R. S. Bucks, D. L. Ashworth, G. K. Wilcock and K. Siegfried, "Assessment of activities of daily living in dementia: Development of the Bristol activities of daily living scale," *Age Ageing*, vol. 25, MAR 1996, 1996.

[39] R. S. Bucks and J. Haworth, "Bristol Activities of Daily Living Scale: a critical evaluation." *Expert Review of Neurotherapeutics*, vol. 2, 2002-Sep, 2002.

[40] A. Bartos, P. Martinek and D. Ripova, "The Bristol Activities of Daily Living Scale BADLS-CZ for the Evaluation of Patients with Dementia," *Ceska a Slovenska Neurologie a Neurochirurgie*, vol. 73, 2010, 2010.

[41] D. Riboni, L. Pareschi, L. Radaelli and C. Bettini, "Is ontology-based activity recognition really effective?" in *Proceedings of CoMoRea'11, 8th IEEE Workshop on Context Modeling and Reasoning*, 2011, .

[42] I. Horrocks, "OWL: A description logic based ontology language," in *11th International Conference on Principles and Practice of Constraint Programming - CP 2005, October 1, 2005 - October 5*, 2005, pp. 5-8.

[43] F. Baader, I. Horrocks and U. Sattler, "Description logics," in *Handbook of Knowledge Representation*, F. van Harmelen, V. Lifschitz and B. Porter, Eds. Elseiver, 2008, pp. 135-180.

[44] S. Batsakis and E. G. M. Petrakis, "SOWL: A framework for handling spatio-temporal information in OWL 2.0," in *5th International Symposium, RuleML 2011-Europe*, 2011, pp. 242-9.

[45] I. Horrocks, P. Patel-Schneider, S. Bechhofer and D. Tsarkov, "OWL rules: A proposal and prototype implementation," *Web Semantics*, vol. 3, pp. 23-40, 2005.

[46] G. Okeyo, L. Chen, H. Wang and R. Sterritt, "Dynamic sensor data segmentation for real-time knowledge-driven activity recognition," 2013.

[47] F. Bellifemine, A. Poggi and G. Rimassa, "JADE a FIPA2000 compliant agent development environment," in *Fifth International Conference on Autonomous Agents, may 28, 2001 - June 1*, 2001, pp. 216-217.

[48] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider and U. Sattler, "OWL 2: The next step for OWL," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, pp. 309-322, 11, 2008.

[49] Stanford Center for Biomedical Informatics Research, "The Protege Ontology Editor and Knowledge Acquisition System," vol. 2011, .

[50] Clark and Parsia, "Pellet: OWL 2 Reasoner for Java," vol. 2011, .

[51] E. Friedman-Hill, "Jess, the Rule Engine for the Java™ Platform," vol. 2012, 12-Oct-2012, 2012.

[52] J. Mei and E. P. Bontas, " Technical reports: Reasoning paradigms for owl ontologies ," http://www.ag-nbi.de/research/owltrans/, http://www.ag-nbi.de/research/owltrans/, 2005.