

# A Hybrid Ontological and Temporal Approach for Composite Activity Modelling

George Okeyo   Liming Chen   Hui Wang   Roy Sterritt

*School of Computing and Mathematics*

*Computer Science Research Institute,*

*University of Ulster at Jordanstown,*

*Shore Road, Co. Antrim, Newtownabbey,*

*BT37 0QB, United Kingdom.*

*okeyo-g@email.ulster.ac.uk, {l.chen,h.wang,r.sterritt}@ulster.ac.uk*

**Abstract-** Activity modelling is required to support activity recognition and further to provide activity assistance for users in smart homes. Current research in knowledge-driven activity modelling has mainly focused on single activities with little attention being paid to the modelling of composite activities such as interleaved and concurrent activities. This paper presents a hybrid approach to composite activity modelling by combining ontological and temporal knowledge modelling formalisms. Ontological modelling constructors, i.e. concepts and properties for describing composite activities, have been developed and temporal modelling operators have been introduced. As such, the resulting approach is able to model both static and dynamic characteristics of activities. Several composite activity models have been created based on the proposed approach. In addition, a set of inference rules has been provided for use in composite activity recognition. A concurrent meal preparation scenario is used to illustrate both the proposed approach and associated reasoning mechanisms for composite activity recognition.

*Keywords-Temporal knowledge; activity modelling; ontologies; interleaved activities; concurrent activities; smart homes.*

## I. INTRODUCTION

Activity recognition is a process to infer activities from a series of observations collected from a user situated environment. It has attracted increasing research interest due to the important role it plays in the provision of activity assistance in smart environments such as smart homes. Broadly speaking, activity recognition encompasses three important tasks, namely activity sensing, activity modelling, and activity inference. Activity sensing tracks the user, environment, and user activities using sensory equipments. Activity modelling produces computational models of activities that serve as the basis for analysing activities. Finally, activity inference processes sensory data against computational activity models to determine the identity of the ongoing activity. Activity sensing has made significant progress recently due to the advancement of sensor technologies and the prevalence of sensor devices. Activity modelling, however, has received relatively little attention. The majority of existing work has focussed on activity modelling based on a single-user, single activity scenario. It still remains a challenge to model composite activities. In this paper we present work that aims at addressing the problem of modelling sequential and multi-tasked activities in a single-user environment.

Activity can be defined in different ways. In this paper we categorise activities as actions, simple activities, and composite activities. An action is an atomic (or indivisible) activity, e.g. turning the door knob. A simple activity is an ordered sequence of actions, e.g. preparing toasted bread. Finally, a composite activity is a collection of two or more simple activities occurring within a given time interval, e.g. having a drink while watching television. Composite activities can be further categorised as sequential or multi-tasked 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-tasked activity occurs when a single user performs two or more activities simultaneously or when multiple residents occupy a smart environment and perform activities concurrently.

Currently, there are three main approaches for activity modelling, namely data-driven, knowledge-driven, and hybrid activity modelling. In data-driven modelling, activity models are learnt from existing data sets using machine-learning based techniques [1, 2]. Alternatively, in knowledge-driven modelling, knowledge engineers and domain experts use knowledge engineering techniques to specify the activity models by encoding commonsense and domain knowledge [3, 4]. Finally, hybrid approaches combine knowledge engineering and machine learning to formulate activity models [5, 6]. The use of ontologies in activity modelling [3] provides a specific mechanism for knowledge-driven activity modelling of simple activities but ignores composite activity modelling. Generally, activities are characterized by rich temporal information, for instance, repetitive time patterns, temporal sequences, temporal duration, and time instances or intervals. Such temporal information, especially time instants and intervals, are crucial in modelling of composite activities. Nevertheless, ontological activity modelling per se does not support temporal modelling and inference.

In this work, we developed a novel knowledge-driven approach to composite activity modelling by combining ontologies and temporal information associated with activities. It is then applied to generate composite activity models. Essentially, the approach enhances ontology-based activity models with qualitative temporal information based on Allen's temporal logic [7]. As a result, this paper makes a number of contributions. Firstly, we develop a novel approach to composite activity modelling that combines two knowledge representation formalisms, i.e., description logics and temporal logic. Secondly, we provide concrete

composite activity models that can support recognition of composite activities, e.g., interleaved and concurrent activities. Moreover, we have developed rules for procedural temporal inference that can be used for procedural processing of the composite activity models. We believe that utilizing the composite activity model and inference rules will provide a means for modelling and subsequent recognition of both simple and composite activities.

The remainder of the paper is organized as follows. Section 2 describes related work. Section 3 presents the approach and its theoretical background. The composite activity models are provided in Section 4. Section 5 presents the inference rules. Section 6 provides an example to illustrate the model. Finally, Section 7 concludes the paper.

## II. RELATED WORK

Composite activity modelling is an active yet challenging research area. In the data-driven community, several approaches are explored. The approaches can be grouped into two categories, namely, approaches that model temporal relationships and those that do not. Among the approaches that model temporal relationships are hidden Markov models (HMM) [8], interleaved HMM [1], factorial conditional random fields (FCRF) [9], and skip-chain conditional random fields (SCCRF) [10, 11] [12]. On the other hand, an approach that does not model temporal relationships, the mining of emerging patterns [2], is also used to recognize interleaved and concurrent activities. Data-driven approaches have the ability to handle uncertain knowledge and are based on well-explored machine learning based techniques. Furthermore, temporal-based approaches can capture long-term temporal dependencies, e.g. activity history, thereby making them suited to composite activity modelling. However, the main drawback is that large amounts of initial training data are needed to learn the activity models. Also, since 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 modelled and recognized. Since our approach requires activity models to be specified based on domain knowledge, it overcomes the need to obtain large data sets.

Knowledge-driven approaches can be divided into those that encode temporal information and those that do not. Approaches that model temporal information include rule based and logic-based systems, e.g. spatio-temporal methods [13], spatio-temporal and context reasoning [14], temporal reasoning and active databases [15], and Event Calculus [16]. On the other hand, the ontology based approach [3], which is based on description logics (DL) [17], does not model and utilize temporal information in activity modelling. Although most knowledge driven approaches assume that the user only performs one sequential activity, a spatio-temporal approach for composite activity modelling using the notion of context-driven activity theory is presented in [18]. Essentially, context information is used to model both primitive actions and simple activities and the resulting models are combined with situation models and used to infer interleaved and concurrent activities. Situations refer to the set of activities that users perform in given contexts; whereby context may refer to location, time, or environment

conditions. The authors derive models of situations, based on spatio-temporal information, from the context spaces theory [19], and use the resulting situations in activity inference. While the authors report success in applying their approach to recognize interleaved and concurrent activities, no explicit model of these composite activities is provided. Instead an aggregation operation, e.g. union, is performed on instances of the model of simple activities to describe composite activities. The main drawback of this approach is that the semantics of resulting composite activity model are unclear and cannot be independently examined. Our work based on temporal ontologies differs from [18] because, in addition to models of actions and simple activities, we also specify an explicit model of composite activities with clear semantics to facilitate automated reasoning.

Hybrid approaches to activity modelling, i.e., Markov logic networks (MLN) [5] and HMMs with Allen logic [6], have also been used to model interleaved and concurrent activities. Both approaches encode and use temporal knowledge but rely on automatically extracting these temporal patterns from data sets. Other hybrid approaches, e.g. [20], only support recognition of sequential activities. Hybrid approaches have the advantage that they can model a range of 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, large datasets are still needed to train the activity models. Since our work only uses domain knowledge it overcomes the need to obtain and annotate large data sets.

## III. A NOVEL APPROACH TO COMPOSITE ACTIVITY MODELLING

### A. Web ontology language (OWL) ontologies and temporal knowledge

OWL, a semantic web ontology language based on DL, provides a set of constructors for specifying OWL classes from simpler classes and properties in order to create ontologies. In addition, it allows axioms for specifying subsumption, equivalence, disjointness, as well as property characteristics to be defined. The constructors, axioms, and DL equivalents are shown in Table 1 [17]. 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.

Temporal knowledge allows knowledge at a particular moment of time and the notion of change in knowledge to be encoded and reasoned with. Typically, 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.

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	$\{x1 \dots xn\}$	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	$r^-$		

the user performs two activities, one after the other. We propose using Allen's temporal logic [7], a constraint-based representation that uses a temporal interval as a primitive, to encode qualitative temporal knowledge. 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 [7]) that are considered adequate to express any relationship that can hold between two time intervals. In this work we combine ontologies and temporal knowledge representation to create composite activity models.

### B. Ontological activity modelling

Ontological activity modelling is used to create logical activity models to formally conceptualize the smart home domain [3, 21]. The resulting activity models are based on objects, environmental elements, events, and interrelationships between activities. The logical models are processed through semantic reasoning to infer activities [3]. SH inhabitants perform activities of daily living (ADL) in

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

certain locations, using particular objects, and at specific times. This information, called contextual information, together with other information, e.g. the different ways to perform the same task, is called domain knowledge and is captured in ADL ontologies [3]. The use of ADL ontologies provides a description based representation of activities that is simple to understand and able to support progressive

activity recognition as an ADL unfolds. The main weakness of a purely ontology-based approach is that despite capturing contextual information, e.g. time and location, it does not support inference with this information. For instance, OWL DL [22] only allows ontologies to capture temporal knowledge but does not support temporal reasoning. This limits it to the modelling and recognition of only simple activities. To support composite activity modelling, an approach that facilitates temporal modelling of activities, integrates into OWL DL, and facilitates temporal reasoning should be adopted.

### C. A hybrid ontological and temporal approach

Representing temporal knowledge in OWL is a challenge because OWL supports unary and binary relations, while adding a temporal dimension requires at least a ternary relation. We adopt the 4D-fluents approach [23, 24] 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 temporal parts of individuals that change some of their properties in time. 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. Typically, 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 static part unchanged. The 4D-fluents approach preserves OWL semantics when incorporating temporal knowledge into OWL ontologies and can therefore exploit existing OWL reasoning support.

Our approach to composite activity modelling is founded on the idea 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 (*notion of change*). When the 4D-fluents approach is extended with qualitative relations [24], e.g. Allen temporal logic relations [7], it can model relations that are necessary for encoding composite activities. To enable the resulting models to be exploited in composite activity recognition, interval relations and SWRL rules [25] are used to provide procedural inference. In the next section, we apply this approach to generate composite activity models.

## IV. COMPOSITE ACTIVITY MODELS

### A. Definitions

#### 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 as:

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

**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 as:

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

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

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

**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 as:

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

**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 as shown below:

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

## 2) Characterization of activities

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 performed by the user. A primitive action is specified as a 2-tuple consisting of a collections of sensor observations and context information as provided below.

$$a: \langle SO_a, C_a \rangle, SO_a \subseteq SO, C_a \subseteq C \quad (6)$$

**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 below:

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

**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 as:

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

**Definition 9- IADL:** This set provides a list of all leaf ADL concepts, i.e., ADL concepts with no child concepts.

$$IADL: \{IA_1, IA_2, \dots, IA_k\}, k \leq n, IADL \subseteq ADL \quad (9)$$

**Definition 10- Simple Activity ( $LA_i$ ):** An ordered sequence of primitive actions. It is specified as shown below:

$$LA_i: \langle AD_L, L \rangle \quad (10)$$

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 below:

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

**Definition 13- Single dynamic composite activity ( $dcA_i$ ):** A composite activity that has properties whose values vary in time, implying the notion of change. It is specified as:

$$dcA_i: \langle \Phi, \tau, L \rangle, dcA_i \in dCA \quad (12)$$

Where **L** is a text string to act as a label for the pattern and  $\Phi$  is a collection of leaf ADLs or a collection of dynamic composite activities such that  $\Phi \subseteq IADL \cup dCA$ . In addition,  $\tau$ , 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  $\Phi$  as  $\Phi = \{make\ dinner, watch\ television\}$ . In addition,  $\tau$  can be specified by  $\tau = \{time-interval-of-make-dinner, time-interval-of-watch-television\}$ .

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

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

**Definition 15:** A single 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 as:

$$scA_i: \langle \Phi, \theta, L \rangle, scA_i \in sCA \quad (14)$$

Where,  $\Phi$  is a collection of leaf ADLs or a collection of static composite activities such that  $\Phi \subseteq IADL \cup sCA$ .  $\theta$  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  $\Phi = \{make\ dinner, watch\ television\}$ . Also,  $\theta$  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.

## B. Composite activity model

### 1) Overview of proposed ontology concepts

Using the definitions in the previous section, we have identified several ontology concepts and properties for specifying the composite activity model. The concepts are discussed below and the properties are provided in Table 3. Fig. 1 shows the relationships between the concepts.

**ADLActivity:** Each simple activity that the user can perform is represented as a subclass of this concept in [3]. The sensor observations and context specified in Definitions 2 and 4 are realized by property restrictions that are defined on ADLActivity. A typical observation, e.g. ‘using the kettle’ can be represented by a hasKettle restriction that is defined on the relevant subclasses of ADLActivity. Further details on ADLActivity concepts can be obtained in [3]. We introduce a temporal extent for this concept by associating it with the TimeSlice.

**DynamicCompositeActivity:** This represents a composite activity that has a temporal extent and whose properties take values that vary in time 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.

TABLE 3 : PROPERTIES IN COMPOSITE MODEL ONTOLOGY

Name	Domain	Range	Other properties	Description
timeSliceOf	TimeSlice	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.
hasActivity	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.
relationshipType	StaticCompositeActivity	string	Functional	Indicates whether a SEQUENCE or PARALLEL relation exists

**TimeSlice:** This encodes the temporal extent of instances of ADLActivity and DynamicCompositeActivity. The temporal extent is achieved by associating the two concepts with Timeslice using timeSliceOf property.

$$TimeSlice \sqsubseteq \exists timeSliceOf. (ADLActivity \sqcup DynamicCompositeActivity) \sqcap = 1 timeSliceOf \sqcap \exists hasTimeInterval.Interval \sqcap = 1 hasTimeInterval \quad (15)$$

**BasicActivityTS:** This is a sub-class of TimeSlice whose instances can only relate to ADLActivity instances. Each simple activity (subclass of ADLActivity) is derived from the collection of time slices denoted by its BasicActivityTS objects.

$$BasicActivityTS \sqsubseteq TimeSlice \sqcap \exists timeSliceOf.ADLActivity \sqcap = 1 timeSliceOf \quad (16)$$

**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 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.

$$CompositeActivityTS \sqsubseteq TimeSlice \sqcap \exists timeSliceOf.DynamicCompositeActivity \sqcap = 1 timeSliceOf \sqcap \exists hasOngoingActivity.TimeSlice \sqcap \geq 2 hasOngoingActivity \quad (17)$$

**TimeInterval:** This concept defines a time interval and indicates the moment of time that a time slice refers to. Its instances are accessed through a BasicActivityTS's or a CompositeActivityTS's instance.

**StaticCompositeActivity:** This defines a static composite activity as per Definition 15. It simply captures the activities (whether simple or composite) that constitute a composite activity. It does not specify any temporal extent but instead captures inter-activity relations.

$$StaticCompositeActivity \equiv \exists hasActivity. (ADLActivity \sqcup StaticCompositeActivity) \sqcap \geq 2 hasActivity \sqcap \exists startedBy. (ADLActivity \sqcup StaticCompositeActivity) \sqcap \exists endedBy. (ADLActivity \sqcup StaticCompositeActivity) \sqcap \exists relationshipType.string \quad (18)$$

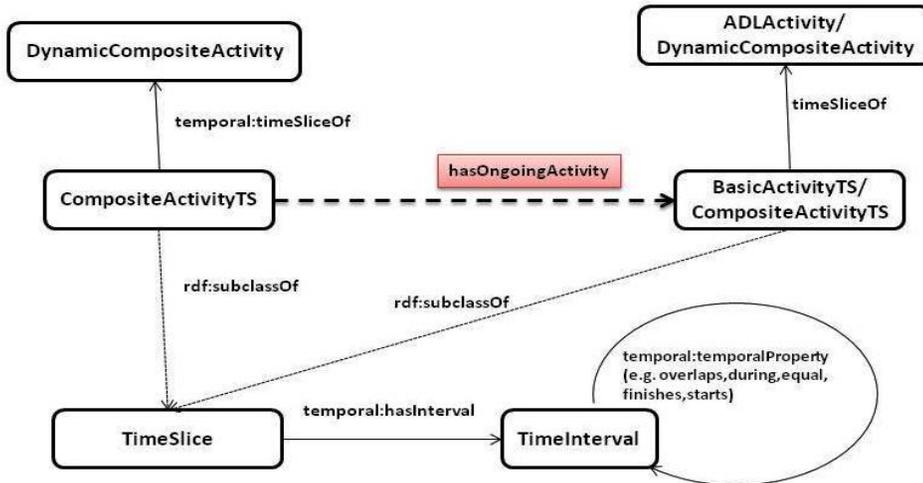


Figure 1: Graphical representation of composite activity model

## 2) Interval logic-based models of composite activities

Further to the ontology concepts described above, we use Allen interval relations to specify dynamic models of composite activities. The models show how temporal intervals of composite activities relate to intervals of their composing activities. The resulting models can be used to infer composite activities from intervals of simple activities or other composite activities.

### a) 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 Fig. 2 (a).

### b) Models of concurrent composite activities

The models represent the notion that activities can occur simultaneously only if their time intervals overlap fully or partially. Such activities are said to be concurrent and concurrency is achieved either by parallelism or interleaving. The presented models are suitable for concurrency through parallelism that is modelled by defining interval relations at the level of activities. However, the models cannot adequately cover interleaved interval relations since interleaving occurs at the level of actions that constitute an activity rather than at the level of the activities themselves. To encode interleaving requires that we use more fine-grained temporal relationships, e.g., between elements of sensor data or actions. This finer level of modelling should be performed in conjunction with relationships that hold between activities. To attain this level of fine-grained modelling, duration information for actions is needed. Given that duration information may vary from one individual to another, we believe that the task is better solved by automatically mining duration information instead of having the model predefined. In this study we perform interval modelling at the level of activities. The following Allen temporal relations are used to model concurrent activities and the models are shown in Fig. 2 (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.

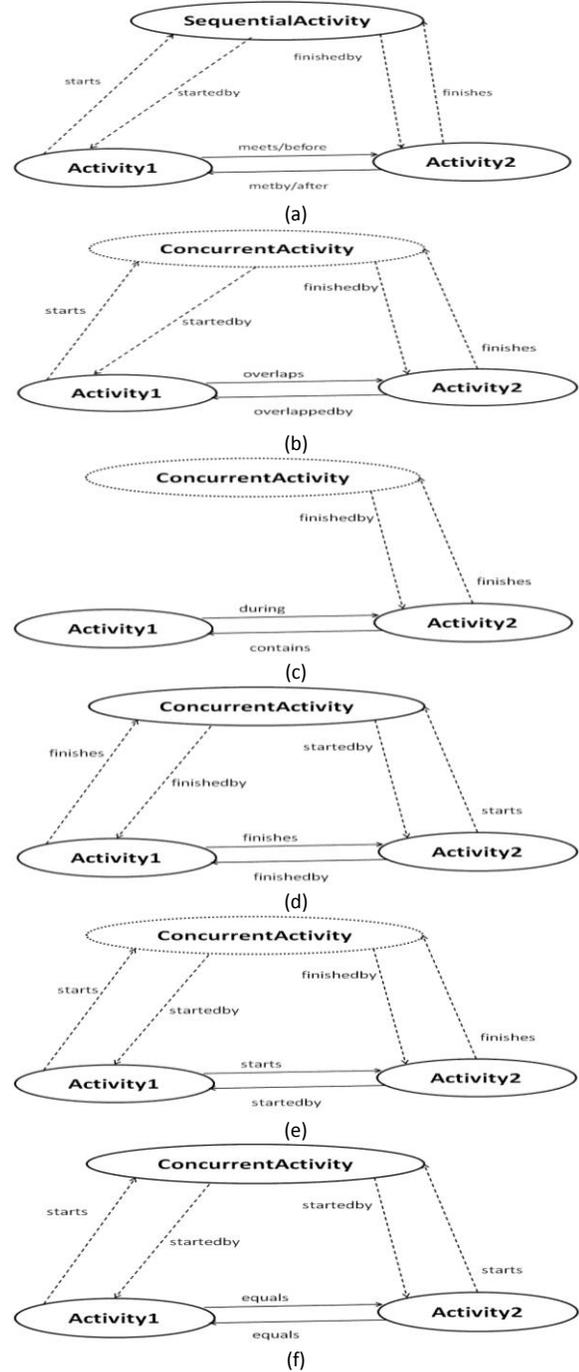


Figure 2: Temporal relationship models of composite activities (a) before/after and meets/met-by; (b) overlaps/overlapped-by; (c) during/contains; (d) finishes/finished-by; (e) starts/started-by; (f) equals models

## V. REASONING AND INFERENCE RULES

To support procedural processing of the composite activity model we have defined a set of rules, based on Semantic Web Rule Language (SWRL) [25], which can infer complex dependencies among activities and therefore the ongoing composite activities. These rules are used to derive the ongoing composite activities by identifying the existing

relationships between temporal intervals of ongoing activities. The rules belong to three categories, 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*. For illustration, we provide rules based on the overlaps/overlapped-by relationship.

#### A. Derive interval relations and assert dynamic composite activities

Given two existing intervals (for primary activities) that have a qualitative temporal relationship, the following rule is used to assert a new interval that associates the intervals with a dynamic composite activity. The left hand side (LHS) obtains two TimeSlice objects, and derives associated TimeInterval objects. It then checks the interval relation between the TimeInterval objects. The right hand side (RHS) of the rule uses the facts established on LHS to assert a new TimeInterval object, sets its beginning and end points, asserts a new TimeSlice object and associates the latter with the asserted TimeInterval object. Finally, an object of DynamicCompositeActivity is created and linked with the TimeSlice object asserted earlier.

```
TimeSlice(?tsx), TimeSlice(?tsy),
hasTimeInterval(?tsx,?x), hasTimeInterval(?tsy,?y),
intervalOverlaps(?x,?y), hasBeginning(?x,?a),
hasEnd(?y,?d) ->Interval(?w), hasBeginning(?w,?a),
hasEnd(?w,?d), intervalStarts(?x,?w),
intervalFinishes(?y,?w), TimeSlice(?tsw),
hasTimeInterval(?tsw,?w), timeSliceOf(?tsw,?aw),
DynamicCompositeActivity(?aw)
```

#### B. Assert instances of fluent property

This is based on the previous rule, and it 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. This is shown by the rule below:

```
TimeSlice(?tsx), TimeSlice(?tsy), TimeSlice(?tsw),
hasTimeInterval(?tsx,?x), hasTimeInterval(?tsy,?y), hasTime
Interval(?tsw,?w), intervalOverlaps(?x,?y),
intervalStarts(?x,?w), intervalFinishes(?y,?w),
timeSliceOf(?tsw,?aw), timeSliceOf(?tsx,?ax),
timeSliceOf(?tsy,?ay) -> hasOngoingActivity(?tsw,?tsx),
hasOngoingActivity(?tsw,?tsy)
```

#### C. Derive and assert static composite activities

This is based on the two rules above and the LHS checks that there exists an instance of DynamicCompositeActivity, 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 a new instance of StaticCompositeActivity. In addition, the RHS creates instances of hasActivity, startedBy, endedBy, and relationshipType properties that are defined for the StaticCompositeActivity concept. Essentially, this rule is

used to infer and identify the ongoing composite activity that is subsequently validated and reported to the user.

```
TimeSlice(?tsx), TimeSlice(?tsy), TimeSlice(?tsw),
hasTimeInterval(?tsx,?x), hasTimeInterval(?tsy,?y),
hasTimeInterval(?tsw,?w), intervalOverlaps(?x,?y),
intervalStarts(?x,?w), intervalFinishes(?y,?w),
timeSliceOf(?tsw,?aw), timeSliceOf(?tsx,?ax),
timeSliceOf(?tsy,?ay), hasOngoingActivity(?tsw,?tsx),
hasOngoingActivity(?tsw,?tsy),
DynamicCompositeActivity(?aw) ->
StaticCompositeActivity(?ca), hasActivity(?ca,?ax),
hasActivity(?ca,?ay), startedBy(?aw,?ax),
endedBy(?aw,?ay), relationshipType(?aw,?z)
```

Similarly, 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.

## VI. EXAMPLE

A smart-home based concurrent meal preparation scenario is modelled to illustrate this work with the ADL ontology and SWRL rules specified using Protégé [26], an OWL ontology editor. The scenario involves a user performing the two tasks 'boil pasta' and 'make tea' such that 'make tea' is performed entirely within the duration of 'boil pasta' as shown in Fig. 3. The start of a time interval is marked by the time the first primitive action is performed, while the end is marked by the activation time of the latest primitive action.

Taking snapshots of activities at various time points, we can observe the following. At time **t2**, the actions *fill kettle*, *boil water*, and *fill pan with water* have been activated. The recognition system classifies this as 'make tea' with time slice *mt-ts#1* {*fill kettle*, *boil water*} and 'boil pasta' with time slice *bp-ts#1* {*fill kettle*, *boil water*, *fill pan with water*}. The intervals associated with time slices *mt-ts#1* and *bp-ts#1* will be denoted as *make-tea* (start=*t0*, end=*t2*), and *boil-pasta* (start=*t0*, end=*t2*), respectively. The temporal inference engine then concludes that the two intervals are equal. From the composite activity model, the relationship equals between temporal intervals denotes concurrent activities. So, from the partial actions, the system reports that the user is most likely to be concurrently performing 'boil pasta' and 'make tea' activities. At **t9**, we have the actions *fill kettle*, *boil water*, *fill pan with water*, *add pasta*, *cook*, *fill cup*, *add tea*, *add milk*, and *add sugar*. These are classified into 'make tea' and 'boil pasta' with time slices *mt-ts#2* {*fill kettle*, *boil water*, *fill cup*, *add tea*, *add milk*, *add sugar*}, and *bp-ts#2* {*fill kettle*, *boil water*, *fill pan with water*, *add pasta*, *cook*}, respectively. The intervals associated with time slices *mt-ts#2* and *bp-ts#2* will be denoted as *make-tea* (start=*t0*, end=*t9*), and *boil-pasta* (start=*t0*, end=*t9*), respectively. As in previous cases, the inference engine concludes that the intervals are equal and reports that the user is concurrently performing 'boil pasta' and 'make tea' activities. At this point, 'make tea' is fully described and should end.

Finally, at **t12**, we have the actions *fill kettle*, *boil water*, *fill pan with water*, *add pasta*, *cook*, *fill cup*, *add tea*, *add milk*, *add sugar*, *add salt*, *cook*, and *drain*. This is classified with time slice *bp-ts#3* {*fill kettle*, *boil water*, *fill pan with water*, *add pasta*, *cook*, *add salt*, *cook*, *drain*}. The intervals associated with time slices *mt-ts#2* and *bp-ts#3* will be

denoted as *make-tea* ( $start=t_0$ ,  $end=t_9$ ), and *boil-pasta* ( $start=t_0$ ,  $end=t_{12}$ ), respectively. The inference engine then infers that *mt-ts#2*'s time interval starts *bp-ts#3*'s time interval. Similarly, from the activity models, the inference engine reports concurrent activities '*make tea*' and '*boil pasta*'.

Each time a temporal relationship is derived, an instance of *CompositeActivityTS* is created based on the *hasOngoingActivity* restriction. In this particular case, the instance will have the range of *hasOngoingActivity* property set to time slices for '*boil pasta*' and '*make tea*', which are instances of *BasicActivityTS* concept. Thereafter, dynamic composite activity, say, '*boil-pasta-and-make-tea*' will be inferred and associated with the *CompositeActivityTS* object. Finally, an instance of static composite activity is derived, validated, and reported. The activities that constitute it will be '*boil pasta*' and '*make tea*'; '*boil pasta*' marked as the starting and ending activity; and the relation indicated as '*PARALLEL*'.

## VII. CONCLUSION

This paper presented a hybrid ontological and temporal approach for creating composite activity models. The resulting approach extends existing ontological activity modelling by employing Allen's temporal logic relations to encode temporal knowledge. To support the use of the generated composite activity models for activity recognition, SWRL-based inference rules are developed to enable procedural inference. A multi-tasked meal preparation activity scenario has been described to illustrate the use of the presented approach. The scenario demonstrates that an ongoing composite activity can be adequately characterized and inferred using the presented activity model. Future work involves developing activity recognition algorithms and architectures to exploit the model in smart home-based composite activity recognition.

### REFERENCES

[1] J. Modayil, T. Bai and H. Kautz, "Improving the recognition of interleaved activities," Proc. 10th Int. Conf. on Ubiquitous Computing (UbiComp 2008), pp. 40-43, Sept. 2008.  
 [2] T. Gu, L. Wang, Z. Wu, X. Tao and J. Lu, "A Pattern Mining Approach to Sensor-Based Human Activity Recognition," IEEE Trans. Knowledge and Data Eng., vol. 23, pp. 1359-72, 2011.  
 [3] L. Chen, C. Nugent and H. Wang, "A Knowledge-Driven Approach to Activity Recognition in Smart Homes," IEEE Trans. Knowledge and Data Eng., vol. 24(6), pp. 961-974, 2011.  
 [4] 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.

[5] R. Helaoui, M. Niepert and H. Stuckenschmidt, "Recognizing interleaved and concurrent activities using qualitative and quantitative temporal relationships," Pervasive and Mobile Computing, vol.7 (6), pp. 660-670, 2011.  
 [6] H.J. Steinhauer, S. Chua, H.W. Guesgen and S. Marsland, "Utilising temporal information in behaviour recognition," in 2010 AAAI Spring Symp., March 22, 2010 - March 24, pp. 54-59, 2010.  
 [7] J.F. Allen, "Maintaining knowledge about temporal intervals," Commun ACM, vol. 26, pp. 832-43, 11. 1983.  
 [8] D.J. Patterson, D. Fox, H. Kautz and M. Philipose, "Fine-grained activity recognition by aggregating abstract object usage," Proc. Ninth IEEE Int. Symp. on Wearable Computers, pp. 44-51, 2005.  
 [9] T. Wu, C. Lian and J.Y. Hsu, "Joint recognition of multiple concurrent activities using factorial conditional random fields," Proc. 2007 AAAI Workshop, pp. 82-87, July 2007.  
 [10] T. Van Kasteren, A. Noulas, G. Englebienne and B. Krose, "Accurate activity recognition in a home setting," Proc. 10th Int. Conf. on Ubiquitous Computing (UbiComp 2008), pp. 1-9, Sept. 2008.  
 [11] D.H. Hu and Q. Yang, "CIGAR: Concurrent and interleaving goal and activity recognition," Proc. 23rd AAAI Conf. on Artificial Intelligence and the 20th Innovative Applications of Artificial Intelligence Conf. (AAAI-08/IAAI-08), pp. 1363-1368, July 2008.  
 [12] D. Hao Hu, S.J. Pan, V.W. Zheng, N.N. Liu and Q. Yang, "Real world activity recognition with multiple goals," Proc. 10th Int. Conf. on Ubiquitous Computing (UbiComp 2008), pp. 30-39, Sept. 2008.  
 [13] D. Lymberopoulos, A. Bamis, T. Teixeira and A. Savvides, "BehaviorScope: real-time remote human monitoring using sensor networks," Proc. 2008 Int. Conf. on Information Processing in Sensor Networks (IPSN 2008), pp. 533-4, 2008.  
 [14] S. Chua, S. Marsland and H.W. Guesgen, "Spatio-temporal and context reasoning in smart homes," Proc. Int. Conf. on Spatial Information Theory, pp. 9-20, 2009.  
 [15] J.C. Augusto and C.D. Nugent, "The Use of Temporal Reasoning and Management of Complex Events in Smart Homes," Proc. European Conf. on Artificial Intelligence, pp. 778, 2004.  
 [16] 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," Proc. 6th International Conference (ICOST 2008), pp. 81-9, 2008.  
 [17] F. Baader, I. Horrocks and U. Sattler, "Description Logics," in Handbook of Knowledge Representation, F. van Harmelen, V. Lifschitz and B. Porter, Elsevier, 2008, pp. 135-180.  
 [18] S. Saguna, A. Zaslavsky and D. Chakraborty, "Complex activity recognition using context driven activity theory in home environments," Proc. 11th Int. Conf. and 4th Int. Conf. on Smart spaces and next generation wired/wireless networking (NEW2AN'11/ruSMART'11), 2011.  
 [19] A. Padovitz, S.W. Loke and A. Zaslavsky, "Towards a theory of context spaces," Proc. Second IEEE Annual Conf. on Pervasive Computing and Communications Workshops, pp. 38-42, 2004.  
 [20] D. Riboni and C. Bettini, "COSAR: hybrid reasoning for context-aware activity recognition," Personal & Ubiq. Computing, vol. 15 (3), pp.271-289, March 2011.  
 [21] D. Riboni, L. Pareschi, L. Radaelli and C. Bettini, "Is Ontology-based Activity Recognition Really Effective?" in Proc. 8th IEEE Workshop on Context Modeling and Reasoning (CoMoRea'11), 2011.  
 [22] I. Horrocks, "OWL: A description logic based ontology language," in 11th Int. Conf. on Principles and Practice of Constraint Programming (CP 2005), pp. 5-8, Oct. 2005.  
 [23] C. Welty and R. Fikes, "A Reusable Ontology for Fluents in OWL," in Fourth Int. Conf. on Formal Ontology in Information Systems, pp. 226-236, 2006.  
 [24] S. Batsakis and E.G.M. Petrakis, "SOWL: A Framework for Handling Spatio-temporal Information in OWL 2.0," in 5th Int. Symp. RuleML 2011-Europe, pp. 242-9, 2011.  
 [25] 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.  
 [26] Stanford Center for Biomedical Informatics Research, "The Protege Ontology Editor and Knowledge Acquisition System", 2012.

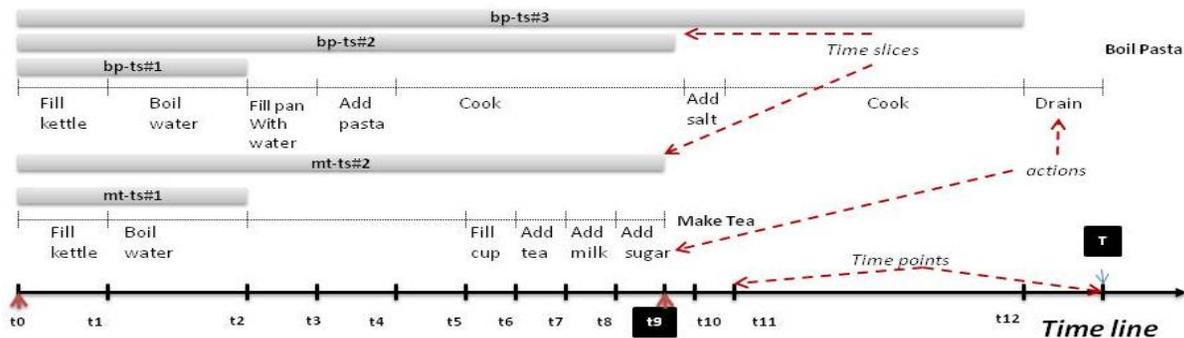


Figure 3: Illustrative composite activity scenario