*Journal of Universal Computer Science, vol. 19, no. 17 (2013), 2577-2597*
*submitted: 15/2/13, accepted: 22/4/13, appeared: 1/11/13 © J.UCS*

# An Agent-mediated Ontology-based Approach for Composite Activity Recognition in Smart Homes

**George Okeyo**
(University of Ulster, Jordanstown, United Kingdom
okeyo-g@email.ulster.ac.uk)

**Liming Chen**
(School of Computer Science and Informatics, De Montfort University
Leicester, United Kingdom
liming.chen@dmu.ac.uk)

**Hui Wang**
(University of Ulster, Jordanstown, United Kingdom
h.wang@ulster.ac.uk)

**Abstract**: Activity recognition enables ambient assisted living applications to provide activity-aware services to users in smart homes. Despite significant progress being made in activity recognition research, the focus has been on simple activity recognition leaving composite activity recognition an open problem. For instance, knowledge-driven activity recognition has recently attracted increasing attention but mainly focused on simple activities. This paper extends previous work by introducing a knowledge-driven approach to recognition of composite activities such as interleaved and concurrent activities. The approach combines the recognition of single and composite activities into a unified framework. To support composite activity modelling, it combines ontological and temporal knowledge modelling formalisms. In addition, it exploits ontological reasoning for simple activity recognition and qualitative temporal inference to support composite activity recognition. The approach is organized as a multi-agent system to enable multiple activities to be simultaneously monitored and tracked. The presented approach has been implemented in a prototype system and evaluated in a number of experiments. The experimental results have shown that average recognition accuracy for composite activities is 88.26**%**.

**Keywords:** Activity recognition, composite activities, interleaved activities, concurrent activities, temporal knowledge, ontology. agents
**Categories:** H.1.2

## 1 Introduction

Ambient assisted living (AAL) utilizes Information and Communication Technologies (ICT) solutions to provide services, e.g. activity assistance, that address problems arising from ageing populations [Hoof, 2011]. Providing services to the elderly enables them to maintain autonomy and functional independence [AAL, 2012]. Therefore, AAL technologies can improve the overall quality of life and to support independent and dignified living for the elderly persons in their preferred environments, e.g. homes. To provide activity assistance, an AAL application uses knowledge of the individual’s tasks, habits and preferences which enables the application to help the individual to correctly execute activities. Smart Homes (SH) have emerged as a viable technology realization of the AAL metaphor [Chan, 2009]. In a SH inhabited by the elderly or disabled, the SH should identify the activities and behaviours of the inhabitants with a view to providing its inhabitants with relevant support services.

In general, individuals perform routine activities that are collectively referred to as daily living activities (ADLs) [James, 2008]. The ability of individuals to perform ADLs is considered a key indicator of autonomy and functional independence [Katz, 1970; Lawton, 1969; Bucks, 1996; Bucks, 2002]. Activity recognition is a process to infer activities from a series of observations collected from a user situated environment [Okeyo, 2012]. Accurate activity recognition is important in the provision of effective assisted living services. We have characterized activities as actions, simple activities and composite activities in a previous study [Okeyo, 2012]. To help understand this piece of work we briefly introduce the concepts here. An action refers to an atomic activity (or indivisible) activity, e.g. flushing a toilet. A simple activity is defined as an ordered sequence of actions, e.g. the actions executed while having a bath. A composite activity refers to a collection of two or more simple activities occurring within a given time interval, e.g. drinking juice while watching television. Further, composite activities can be grouped into sequential or multi-tasked activities. A sequential composite activity occurs when two or more activities occur in consecutive time intervals. Conversely, a multi-tasked composite activity occurs when the user performs two or more activities simultaneously, i.e., interleaved and concurrent activities. In general, users perform both simple and composite activities, e.g. sequential activities, and interleaved and concurrent activities. For instance, an individual may perform several activities at the same time or following each other.

There are three main categories of approaches used for activity recognition, namely, data-driven (DD) [Modayil, 2008; Gu, 2011; Patterson, 2005; Philipose, 2004; Kasteren, 2008], knowledge-driven (KD) [Chen, 2011; Chen 2009; Chen, 2008; McKeever, 2010; Strorf, 2009; Saguna, 2011], and hybrid [Helaoui, 2011a; Helaoui, 2011b; Steinhauer, 2010] activity recognition approaches. DD activity recognition approaches elicit activity models from pre-existing datasets using existing well-developed machine learning techniques. Thereafter, activity inference is performed against the learnt activity models whenever sensor data becomes available. In KD activity recognition approaches, knowledge engineers and domain experts specify activity models using a knowledge engineering process. In essence, the activity models encode commonsense and domain knowledge about activities. The resulting activity models are used to support reasoning to infer activities whenever sensor data is obtained. The hybrid activity recognition approaches combine data-driven and knowledge-driven techniques.

Both simple activity recognition and composite activity recognition have been explored within data-driven and hybrid activity recognition approaches. However, only simple activity recognition is researched in knowledge-driven activity recognition [Chen, 2011; Chen, 2008; Strorf, 2009], with composite activity recognition receiving very little attention. Therefore, addressing composite activity recognition is important in addressing real-world activity recognition. To this end, this paper focuses on the problem of recognition of composite activities in a single-user environment from a knowledge-driven perspective.

The use of ontologies in activity recognition [Chen, 2011; Riboni, 2011] has attracted increasing attention but existing research has mainly focused on simple activity modelling and recognition. In previous work, we have made effort to develop a knowledge-driven approach to composite activity recognition. Nevertheless, our earlier work concentrated on composite activity modelling, namely combining ontological and temporal knowledge modelling formalisms to create composite activity models [Okeyo, 2012]. This paper presents our follow-on work on knowledge-driven composite activity recognition, which is built upon previous research results of composite activity modelling. This paper makes three main knowledge contributions. Firstly, we propose a knowledge-driven approach to composite activity recognition and introduce a process and related methods for activity recognition. Secondly, we conceive and develop a unified agent-mediated framework to support simple and composite activity recognition. Thirdly, we develop a prototype system and associated recognition algorithms that can support the application and deployment of the proposed approach. The approach, the agent-mediated architecture and associated algorithms have been validated by experiments with impressive experimental results.

The remainder of the paper is organized as follows. Section 2 discusses related work. Section 3 describes the unified activity recognition approach. The agent-mediated architecture and recognition algorithms are discussed in Section 4. We present the system prototype and the experimental results in Section 5. Finally, Section 6 concludes the paper.

## 2 Related Work

Much of the research related to composite activity recognition has been based on data-driven [Modayil, 2008; Gu, 2011; Patterson, 2005] and hybrid [Helaoui, 2011b; Steinhauer, 2010] approaches to activity recognition. Patterson et al. [Patterson, 2005 investigated the use of hidden Markov models (HMM) to recognize interleaved and concurrent activities from object use. Modayil et al. [Modayil, 2008] explored the use of interleaved HMMs for recognition of multi-tasked activities in mobile platforms. Both works highlighted the use of intra-and inter-activity dynamics, e.g. temporal relationships among activities, in activity modelling. Gu et al. [Gu, 2011] investigated the use of emerging patterns based approach, a data mining technique, to interleaved and concurrent activity recognition in a sensor-based platform. The approach by Gu et al. does not use temporal inter-activity relationships; instead, it uses feature relevance to segment the boundary of adjacent activities. In addition, it builds activity models for sequential activities then uses the models to support interleaved and concurrent activity recognition. In this way, it does not require data sets with interleaved and concurrent activities to be present in order to initialize the activity models.

Helaoui et al. [Helaoui, 2011b] investigated the use of Markov logic networks (MLN), a statistical relational approach able to encode domain knowledge, to recognize interleaved and concurrent activities. The approach builds composite activity models from data sets and augments the models with domain knowledge to support composite activity recognition. Steinhauer et al. [Steinhauer, 2010] also investigated the use of HMMs enhanced with qualitative temporal relationships based on Allen logic [Allen, 1983]. Basically, it uses contextual maps, e.g. qualitative temporal information, qualitative spatial information, etc., together with HMMs to support activity recognition. It uses Allen's temporal relations to encode inter-activity relationships that denote interleaved and concurrent activities. In particular, it first uses HMM to identify the activity then checks if the identified activity satisfies the temporal constraints. It only reports the activities that satisfy the constraints.

Data-driven and hybrid approaches are well supported since they are based on well-developed learning and probabilistic techniques. Nevertheless, large amounts of initial training data are still needed to learn the activity models, leading to the "cold start" problem. In addition, since users perform activities in a variety of ways, activity models for one user may not be applicable to the other, resulting in model reusability and applicability problems. Since our approach uses only domain knowledge to specify activity models, it easily overcomes the "cold start" problem. Moreover, by modelling common sense activity knowledge as part of domain knowledge, our approach addresses the model re-usability and applicability problem. Common sense activity knowledge includes activity characteristics that are common to most users, e.g. meal preparation activities occur in the kitchen; bathroom activities occur in the bathroom; grooming activities occur in the bathroom or bedroom; a person cannot be sleeping while preparing tea, etc.

Even though the knowledge-driven approach solves the "cold start" problem, little research has been performed in composite activity recognition. So far, only the work by Saguna et al. [Saguna, 2011] addresses composite activity recognition. The work by Saguna et al. combines ontological and spatio-temporal modelling and reasoning to recognize interleaved and concurrent activities. However, it ignores the challenges that arise from spatio-temporal querying and reasoning from within ontologies. Other works that use ontologies to model pervasive computing environments are reported in [Wang, 2004] and [Fook, 2006]. The work by Wang et al., however, does not model temporal context; unlike our approach that uses temporal context as the primary means of modelling composite activities. The work by Fook et al. uses ontologies to model context for handling agitation behaviour in persons with dementia. It uses temporal context, but work it does not support qualitative temporal inference needed to infer inter-activity dependencies that characterize composite activities. In a nutshell, our work adopts a systematic method for modelling and processing temporal knowledge based on 4D-fluents [Welty, 2006] to address composite activity recognition. This ensures that we can integrate and exploit ontological temporal knowledge for activity recognition. In addition, this paper provides a scalable agent-mediated architecture that can be used to monitor and track several activities in real-time.

## 3 Background and the Unified Activity Recognition Approach

### 3.1 The approach

Generally, in a dense sensing based SH each sensor is attached to a specific object and whenever a user interacts with an object in performing an activity, contextual information, e.g. time and location, will be captured as corresponding sensor data.

Suppose a user performs activities along a timeline, then the streaming sensor data can be analyzed to recognize the ongoing simple or composite activities. Recognizing composite activities involves three main tasks, namely, composite activity modelling, sensor data stream segmentation and composite activity recognition.

Composite activity modelling creates two types of activity models, namely static activity and dynamic activity models. Static activity models use concepts that describe that an activity occurs at a specific time, i.e., a temporal reference, *e.g. activity X occurs at time 10 am*. Dynamic activity models encode inter-activity relationships that exist between the activities that constitute a composite activity. Therefore, dynamic activity models are characterized by both a temporal reference and the fact that the activities share a qualitative temporal relationship, i.e., reflects the notion of change), *e.g. given three activities, X, Y and Z, then '*X involves Y at 10am followed by Z at 10.15 am*'* shows a *sequential relationship*. To create dynamic models of composite activities, we have used the 4D-fluents [Welty, 2006] to add qualitative temporal knowledge modelling, representation, and reasoning capabilities to Web Ontology Language (OWL) [Horrocks, 2005a; Grau, 2008] ontologies. The 4D-fluents approach uses two fundamental building blocks, namely, *time slices* and *fluents* to provide a vocabulary to represent temporal knowledge. The time slices represent the temporal parts of a specific entity at a given time instant or time interval. To describe the entity over a given time interval, its time slices are aggregated. On the other hand, *fluents* are properties that hold at specific moments in time, which could be time interval or time instant. In essence, the fluent property holds among two time slices. Therefore, an ontology that associates concepts using the fluent property is called a dynamic ontology. Because the ontology describes activity models, it results in dynamic activity models. A key advantage of the 4D-fluents approach is that it preserves OWL semantics when incorporating temporal knowledge into OWL ontologies and can therefore exploit existing OWL reasoning support. However, to support temporal inference and to perform reasoning with the dynamic activity models, we use entailment rules based on Allen's qualitative temporal relations [Allen, 1983]. Further details on composite activity modelling including the entailment rules can be found in [Okeyo, 2012] whereby composite activity models are formulated using both ontological and temporal modelling formalisms.

To segment streaming sensor data, we use a time window based segmentation method that dynamically segments the stream by using information (e.g. activity duration) from the activity models and the feedback from activity inference as described in [Okeyo, 2013]. The resulting segments are analyzed during activity recognition and mapped to corresponding simple or composite activities.

To perform composite activity recognition, this paper introduces a unified approach that combines simple and composite activity recognition into a single framework. To this end, the approach divides the recognition task into three interdependent sub-tasks, namely, action recognition, simple activity recognition, and composite activity recognition. Action recognition processes the sensor data available in sensor data segments using ontological reasoning to derive primitive actions. The action recognition task is performed as part of simple activity recognition. For simple activity recognition, we adopt and modify the ontological activity recognition approach described in [Chen, 2011; Chen 2009]. Ontological activity recognition uses a logic-based ontology language, e.g. OWL, to structure and describe activities during activity modelling. It encodes activity models as activity ontologies, and then uses semantic reasoning (e.g. subsumption and equivalence reasoning or instance retrieval) to process sensor data against the ontological activity models during activity recognition. In this paper, we modify the ontological approach by including a step to generate activity descriptions. We define an activity description as a collection of primitive actions that together, partially or fully, describe a simple activity. An activity description can be created by grouping the primitive actions into one or more activity descriptions corresponding to the simple activities that are defined in the activity models. As more sensor data is obtained new activity descriptions are created or the existing ones are updated. The modified ontological approach then compares each activity description with activity models using semantic reasoning and reports the activity model that is closest to the activity description 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. Thereafter, the results of simple activity recognition are aggregated using the mechanism described in the next section. By separating activity recognition into interdependent tasks, it is possible to use different techniques for each task. In this work, instance retrieval or subsumption and equivalence reasoning is used for action and simple activity recognition. Instance retrieval determines which objects are instances of a given ontology concept. Subsumption reasoning finds all concepts that are sub-concepts of a given concept. Equivalence reasoning returns all concepts that are semantically equivalent to a given concept. For composite activity recognition, rule-based inference techniques are exploited.

### 3.2 The Modular Architecture

The approach is depicted in a modular architecture as shown in Fig. 1 that consists of a number of core components that interact with each other to provide intended functions. Core to the architecture is three *knowledge bases* (KB), namely, static activity model KB (StatSKB), dynamic model of composite activities KB (DynaCAKB), and context-driven rule-base (ContextRB), which are utilized by the different components during operation. *StatSKB* provides the static model of activities and includes definitions of activities of daily living as well as predefined composite activities. *DynaCAKB* encodes the dynamic model of activities. *ContextRB* encodes the rules for inferring qualitative temporal relations between activities and therefore deriving the ongoing composite activities. At runtime DynaCAKB and ContextRB are used to derive the temporal dependencies that exist among ongoing activities. The presence of temporal dependencies among activities implies the existence of composite activities. The knowledge bases are encoded as an activity of daily living (ADL) ontology. The *data monitoring and segmentation* component monitors and collects contextual and sensory data whenever a user interacts with objects in performing daily activities and then segments the sensor data stream.

The *integrated activity inference* component performs three tasks, namely iterative action inference, activity inference and activity analysis and refinement. These tasks are performed in complex activity recognition unit (CARU) and simple activity recognition unit (SARU). *CARU* performs its task using *iterative action* *inference* and *fusion, and activity analysis and refinement* components based on the StatSKB, DynaCAKB, and ContextRB. The *iterative action inference and fusion* component uses the sensory information contained in the data segment and the StatSKB to derive primitive actions and activity descriptions. The *activity analysis and refinement* component is used to discover complex dependencies among ongoing activities. It uses various elements of context information, e.g. task-related context and temporal context encoded in ContextRB. The analysis and refinement component outputs simple activities or composite activities together with feedback that is used in the segmentation component to modify the parameters used in the segmentation mechanism. On the other hand, *SARU* performs the necessary activity inference autonomously and communicates its status. It uses its activity inference component to derive the activity that corresponds to a given activity description. Activity inference uses StatSKB as well as recognition algorithms described in [Chen, 2011; Chen, 2009].

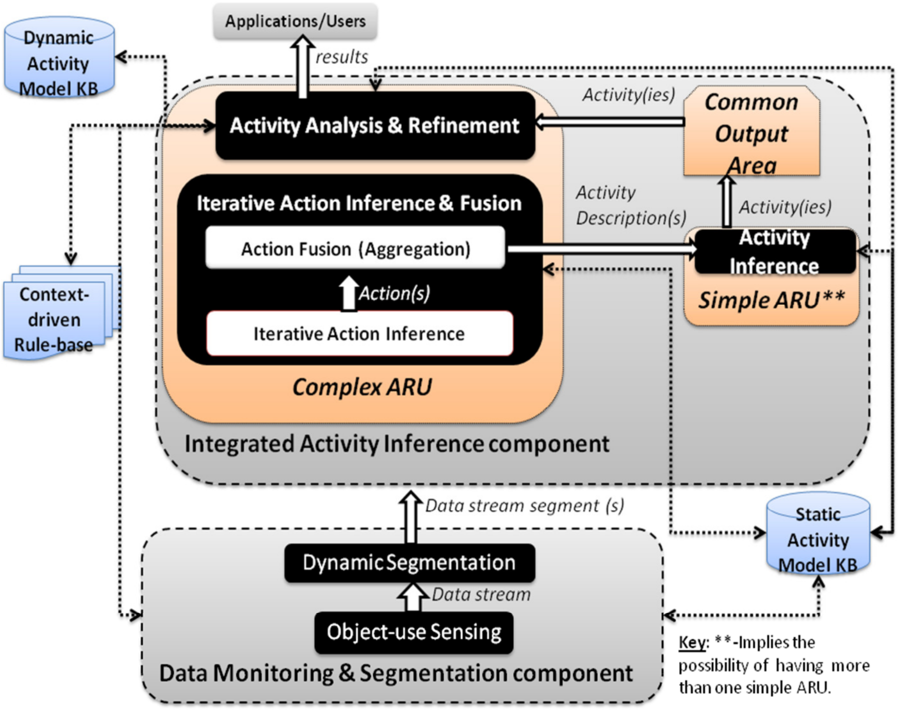

Applications/Users
results
Dynamic Activity Model KB
Activity(ies)
Common Output Area
Activity Analysis & Refinement
Iterative Action Inference & Fusion
Action Fusion (Aggregation)
Action(s)
Iterative Action Inference
Complex ARU
Activity Description(s)
Activity(ies)
Activity Inference
Simple ARU**
Context-driven Rule-base
Integrated Activity Inference component
Data stream segment (s)
Static Activity Model KB
Dynamic Segmentation
Data stream
Object-use Sensing
Data Monitoring & Segmentation component
Key: **-Implies the possibility of having more than one simple ARU.

*Figure 1: The modular architecture of the proposed approach*

## 4 Activity Recognition Mechanisms

### 4.1 The process of composite activity recognition

For ease of reference, we use *ADL Ontology* to refer to the entire collection of knowledge bases introduced in Section 3.2. Given a sensor data segment the recognition of composite activities proceeds using the three-step process described below. In the first step, it derives activity descriptions using the algorithm in Fig 2. In essence, the enclosed observations are converted into primitive actions by checking property restrictions specified in the *ADL Ontology*. Thereafter, the primitive actions are grouped together to form activity descriptions based on activity definitions in the *ADL Ontology*.

```
Input: Segment (s), ADL ontology (ADL-O)
Output: Set of activity descriptions (AD) /*partial or complete*/
DERIVE-ACTIVITY-DESCRIPTIONS (s, ADL-O)
Begin: Extract observations from segment s, $O=\{o_1,o_2,...o_h\}$
For each $o_i \in O$ Do //for each observation
        Determine all activities that are described by the observation $A(o_i)=\{A_1, A_2, ...A_n\}$
End for
Create a set of all activities $A=A(o_1)\sqcup ...A(o_i)\sqcup ... A(o_h)$
For each $x \in A$ Do/*for all activities*/
        Collect all observations describing x as activity descriptions $AD_x$
Add $AD_x$ to AD End for
Return AD
End
```

*Figure 2: Algorithm to transform segment into activity descriptions (inside CARU)*

In the second step, simple activity recognition is performed to map activity descriptions into activity labels. Basically, this involves comparing activity descriptions with activity models in the *ADL Ontology* and returning the label of the closest model. If multiple activity descriptions are obtained for a given segment, the mapping into activity labels is performed simultaneously. The steps for achieving simple activity recognition are summarized by the algorithm listing in Fig. 3.

```
Input: Activity description ($AD_i$),ADL ontology (ADL-O)
Output: Simple activity or failure report
RECOGNIZE-SIMPLE-ACTIVITY ($AD_i$, ADL-O ) /*Classify activity  through ontological
inference*/
Begin: Map $AD_i$ to a simple activity /*use ontological inference*/
If a leaf activity is returned Then
        Report it
Else
        If goal is still valid Then
                Wait for updated activity description (go to beginning)
        Else    Communicate status report and terminate
        End if
End if
End
```

*Figure 3: Algorithm to infer simple activities from activity descriptions (inside SARU)*

In the final step, the results of simple activity recognition are analyzed and the aggregated results communicated to the user. If only one simple activity has been identified for a 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 user, the ontology is checked for a corresponding instance in the static activity models. 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. For instance, if it is found that there is a temporal dependency between two activities, say MakePasta and BrushTeeth, which occur in parallel, yet there is no matching description in the existing static activity model, a new composite activity MakePastaAndBrushTeeth will be recommended for inclusion into the ontology. To perform temporal analysis, the approach uses temporal entailment inference rules. The rules can infer qualitative temporal relationships, derive corresponding composite activities from the dynamic activity models, and then check for corresponding composite activities in the static activity models. Due to space limitations, the interested reader is referred to [Okeyo, 2012] for details of rule specification and usage. The steps for aggregating the results of simple activity recognition are summarized in the listing in Fig. 4.

```
Input: Recognition statuses (RS), ADL ontology (ADL-O)
Output: Composite activity (CA) or simple activity (A)
AGGREGATE-RESULTS (RS, ADL-O)
Begin: Define the set of time intervals TI
For each sti∈RS Do
        Obtain temporal interval I and add it to TI
End for
If only one interval is present Then
        Return (A) /*A is activity associated with interval I*/
Else
        Infer interval relations
        Use inferred interval relations to derive ongoing composite activity relationships
        Check corresponding instances of static composite activities in ADL ontology
        IF obtained composite activity relationships are in ADL ontology Then
                Return (CA)
        Else    Recommend the activity ontology be updated to accommodate it.
        End if
End if
End
```

*Figure 4: Algorithm to infer interval relationships and aggregate the results (inside CARU).*

The algorithm listing in Fig. 5 provides an aggregate for all three steps as described in Fig 2, 3, and 4 and depicts the tasks that occur during composite activity recognition.

```
Input: sensor data stream ($\Omega$), ADL ontology (ADL-O)
Output: recognition result /*composite/simple activity label or failure status*/
RECOGNIZE-ACTIVITY ($\Omega$, ADL-O)
Begin: While data stream is active Do /*segmentation*/
Segment data stream into a set of segments $S=\{s_1, s_2, ..., s_n\}$
For each $s_i \in S$ Do /*for each segment map observations to activities*/
        /*Create partial/complete activity descriptions to form set $AD=\{AD_1, AD_2,...AD_k\}$*/
        DERIVE-ACTIVITY-DESCRIPTIONS ($s_i$, ADL-O) /* returns set AD*/
        For each $AD_j \in AD$ Do /*for each activity description classify activity*/
                RECOGNIZE-SIMPLE-ACTIVITY (ADj, ADL-O ) /*return recognition
                status*/
                Update classification status into recognition status RS= { $st_1, st_2,...st_m$ }
                AGGREGATE-RESULTS (RS, ADL-O) /*Perform analysis and
                refinement*/
                If results are conclusive Then
                        Convey results
                Else  Update segment and window  End if
        End for
End for
End while
End
```

*Figure 5: Algorithm depicting overall recognition strategy (inside CARU and SARU)*

### 4.2 The Multi-agent architecture for activity recognition

To realize the modular architecture and the algorithms described above, we have formulated the unified activity recognition approach as a multi-agent system.

An agent refers to a software system that is situated in a dynamic, complex environment, and is capable of sensing the changes in the environment and interacting with other entities in order to take actions that achieve its design objectives. An agent should exhibit four basic properties, namely, autonomy, social ability, reactivity (responsiveness) and pro-activity [Wooldridge, 1995]. An agent that is autonomous is able to act with no direct intervention from humans or other agents and has control over both its internal state and actions. Social ability refers to the ability of an agent to interact with other agents, including humans. Generally, an agent is situated in an environment, and the ability to perceive the environment and respond in a timely fashion to environmental changes is referred to as reactivity. Pro-activeness indicates an agent's capability to exhibit goal-directed behaviour by taking initiative to achieve its set goals and design objectives. The four features and other features (e.g. adaptability, intelligence, rationality, mobility, flexibility, temporal continuity, etc.), can be exploited to design agents for use in applications in the AAL domain.

Agents can be used to structure solutions in application areas that are characterized by complexity, ubiquity, and distributed data, control, expertise, and resources [Jennings, 1998a]. One such application area is the smart home and by extension the task of activity recognition. The smart home is characterized by various components, e.g. sensors, actuators, people, activities, and interactions, making it complex. In addition, the smart home is by definition ubiquitous. Moreover, activity recognition can be characterised as distributed because it involves various interdependent tasks such as environment and behaviour monitoring, segmentation, and activity inference. Also, each task requires its own data, resources, control and expertise. A multi-agent system (MAS) refers to a system consisting of a group of agents capable of interacting with each other to achieve their design objectives [Jennings, 1998b]. Therefore, the multi-agent approach can be used to model, structure, and implement a complex software system as a collection of interacting, autonomous agents [Jennings, 2001]. From the foregoing, we adopt the multi-agent approach to specify a system for unified activity recognition.

To realize the modular architecture shown in Fig. 1, four agent roles are identified, namely, *segment the sensor data stream*, *generate activity descriptions and convey recognition results*, *infer simple activities*, and *manage inference rule execution and infer composite activities*. The resulting multi-agent system consists of four types of agents that play the roles stated and the agents include monitor and segment agent (MSA), composite activity recognizer agent (CARA), simple activity recognizer agent (SARA), and activity analysis agent (AAA). Using the agents, the CARU component as described in Fig. 1 is implemented using the two agents, i.e., CARA, and AAA; SARU is implemented by the SARA agent; and the data monitoring and segmentation component implemented by the MSA agent.

We have chosen agent as an implementation artefact because agents provide the different components with autonomy needed to perform their respective tasks. In addition, each component can continuously and proactively review and react to changes in its goals. There is also massive parallelism involved in executing the various tasks involved and the MAS can implement the tasks as parallel agent behaviours or tasks. The resulting multi-agent architecture is shown in Fig. 6. Each of the agents in the architecture is described below.

***Monitor and segment agent (MSA).*** The monitor and segment agent plays the role '*segment the sensor data stream*'. Essentially, MSA receives streaming sensor data from the environment and uses time windows to segment the stream in real-time. It then sends the resulting segments to the CARA agent for further processing.

***Composite activity recognizer agent (CARA).*** The composite activity recognizer agent plays the role '*generate activity descriptions and convey recognition results*'. The CARA agent obtains segments from MSA agent and processes them to determine the actions entailed. Consequently, it uses the actions to generate activity descriptions that approximate the activities that are likely to be occurring. The CARA agent then spawns the SARA agents, and provides each with the relevant activity description. It will keep updating the activity descriptions and communicating the descriptions to SARA agents. In addition, it receives feedback related to activity labels from SARA agents, and conveys activity data to AAA agent. Finally, it obtains the results from AAA agent and provides results, i.e., identified simple and composite activity labels, to applications. Moreover, it sends information about the recognition status to MSA agent to facilitate dynamic segmentation.

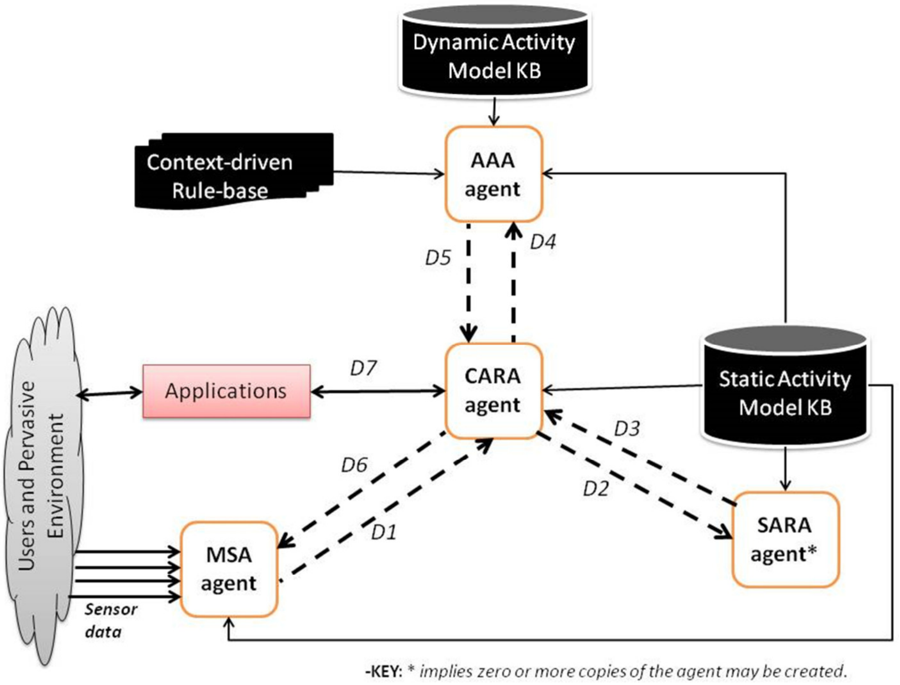


*Figure 6: The multi--agent architecture for unified activity recognition.*

The messages exchanged convey the following: D1-data segments; D2-activity descriptions; D3-simple activity labels; D4-activity data; D5-simple and composite activity labels; D6-recognition status; D7-identified activity labels.

***Simple activity recognizer agent (SARA).*** The simple activity recognizer agent plays the role '*infer simple activities*'. The SARA agent receives activity descriptions and their revisions from CARA agent and performs ontological inference to determine associated activity labels. It then conveys its recognition status - a specific or generic activity label - to CARA agent. In addition, the SARA agent continuously reviews its status and can terminate if a predefined upper temporal duration threshold is exceeded. At runtime, zero or multiple SARA agents can be created and executed, with each agent corresponding to exactly one activity description. Whenever activities are performed in parallel during a particular time interval, multiple activity descriptions will be derived and corresponding SARA agents will be executed thus allowing the entailed activities to be recognized. The results from these multiple SARA agents are used as input to composite activity recognition.

***Activity analysis agent (AAA).***The activity analysis agent plays the role '*manage inference rule execution and infer composite activities*'. It receives activity data from the CARA agent, and executes inference rules to determine the presence of inter-activity dependencies, e.g. sequence or concurrency. It only signals the presence of composite activities if it can determine that inter-activity dependencies exist. Finally, the AAA agent conveys the results - simple or composite activity labels - to the CARA agent.

# 5 Experiments and Evaluation

## 5.1 Experimental environment

We have implemented the presented approach to composite activity recognition as an agent-based application using Java Agents Development Framework (JADE) [Bellifemine, 2001]. JADE is a Foundation of Intelligent Physical Agents (FIPA) [FIPA, 2012]-compliant agent development environment. The basic standards for FIPA include agent communication, agent management, and agent/software integration. The standard for agent management is aimed at allowing agents to register, deregister, be searched, and be modified. The standard for agent communication is concerned with the message transport protocol, message content, and communication language.

Four types of agents have been developed to play the following roles described in the previous section. Each agent advertises its capabilities by registering with JADE's Directory Facilitator so that other agents can search for it. The agents communicate with each other by exchanging messages represented as serialized objects. Each agent decides on the type of agent that should receive a particular message. In addition, the multi-agent system uses communicative act theory to manage conversations between agents.

The *ADL Ontology* is designed using OWL 2 [Grau, 2008] in Protégé [Protégé, 2011] ontology editor. This includes an implementation of the entailment rules as Semantic Web Rule Language (SWRL) [Horrocks, 2005b] rules as part of the ADL Ontology. Fig. 7 shows a snapshot of some concepts and instances in the ontology generated using OntoGraf protégé plug-in. The right hand side of Fig. 7 shows a list of some of the relationships between concepts and individuals.

The prototype system uses Java-based application programming interfaces (APIs) to interact with the Pellet [Pellet, 2011] OWL reasoner for ontological reasoning. To facilitate the execution of the inference rules we translated the *ADL Ontology* and the SWRL rules to Java Expert System Shell (JESS) [Jess, 2012] fact and rule bases. We used the OWL2Jess and SWRL2Jess translators based on the guidelines provided by Mei and Bontas [Mei, 2005]. In the prototype, the JESS fact and rules bases are accessed and processed by a JESS rule engine. The rule engine is accessed and manipulated by the AAA agent that is responsible for aggregating the results of simple activity recognition.

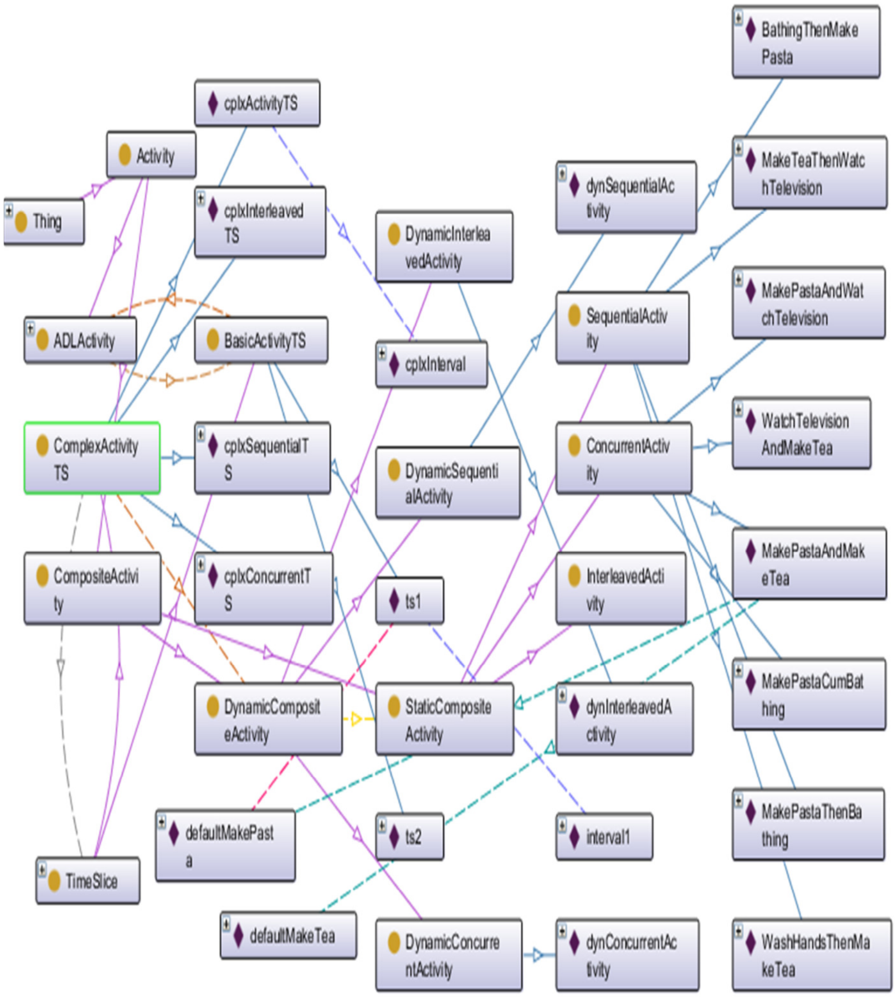


*Figure 7: Snapshot of ADL Ontology captured using OntoGraf protégé plug-in*

### 5.2 Experiment design

To evaluate the presented approach, we developed an ADL simulator to generate synthetic ADL data. Synthetic ADL data has been generated based on seven typical ADLs that are related to meals (i.e., MakeTea, MakeCoffee, MakeChocolate, and MakePasta), hygiene (i.e., HaveBath, WashHands) and recreation (i.e., WatchTelevision). We specified representative ADL patterns for both simple activities and composite activities. The simulator randomizes the ADL patterns to generate the synthetic ADL data. In addition, one or more patterns of sensor activations are provided for each ADL pattern. The ADL patterns, including activity duration and temporal distance between sensor activations, are based on both common sense knowledge and guidelines from how-to websites, e.g. http://www.wikihow.com/. By having varied ADL patterns, the synthetic ADL data for any single ADL activity reflects the reality that the user can perform an activity in different ways. For instance, in making tea the user may use 'milk before sugar' in one instance, 'sugar then milk' in another, or even completely leave out sugar or milk. To incorporate more temporal meaning, each sensor in a pattern is activated after a given amount of time after the immediately preceding activation, e.g. CupObj@20 means the object should be activated 20 seconds after the previous observation. By implication, this ensures that duration information of activities is included when synthetic ADL data is generated.

- ***MakeTea***: KitchenDoorObj@0, KettleObj@20, CupObj@180, TeaObj@20, MilkObj@20, SugarObj@20
- ***MakePastaAndMakeTea:*** KitchenDoorObj@0, CookerObj@20, SaucePanObj@20, KitchenTapObj@20, ItalianPastaObj@20, KitchenTapObj@20, KettleObj@20, SaltObj@600, DrainerObj@300, PlateObj@60, CupObj@150, WholeMilkObj@30, BritishTeaObj@20, SandSugarObj@20

### 5.3 Experiments

*We generated eight weeks of synthetic ADL data consisting of 56 episodes of simple or composite activities to provide the ground truth. Table 1 provides a summary of the composite activities. The episodes contain simple activities that are performed in various combinations and orders. There are 23 episodes of interleaved and concurrent activities (46 simple activities), 25 episodes of sequential activities (50 simple activities, and eight episodes of standalone simple activities. In summary, the episodes are based on 104 simple activities, including MakeTea (26), MakeCoffee(3), MakeChocolate (7), MakePasta (25), HaveBath (15), WachHands (10), and WatchTelevision (18).*

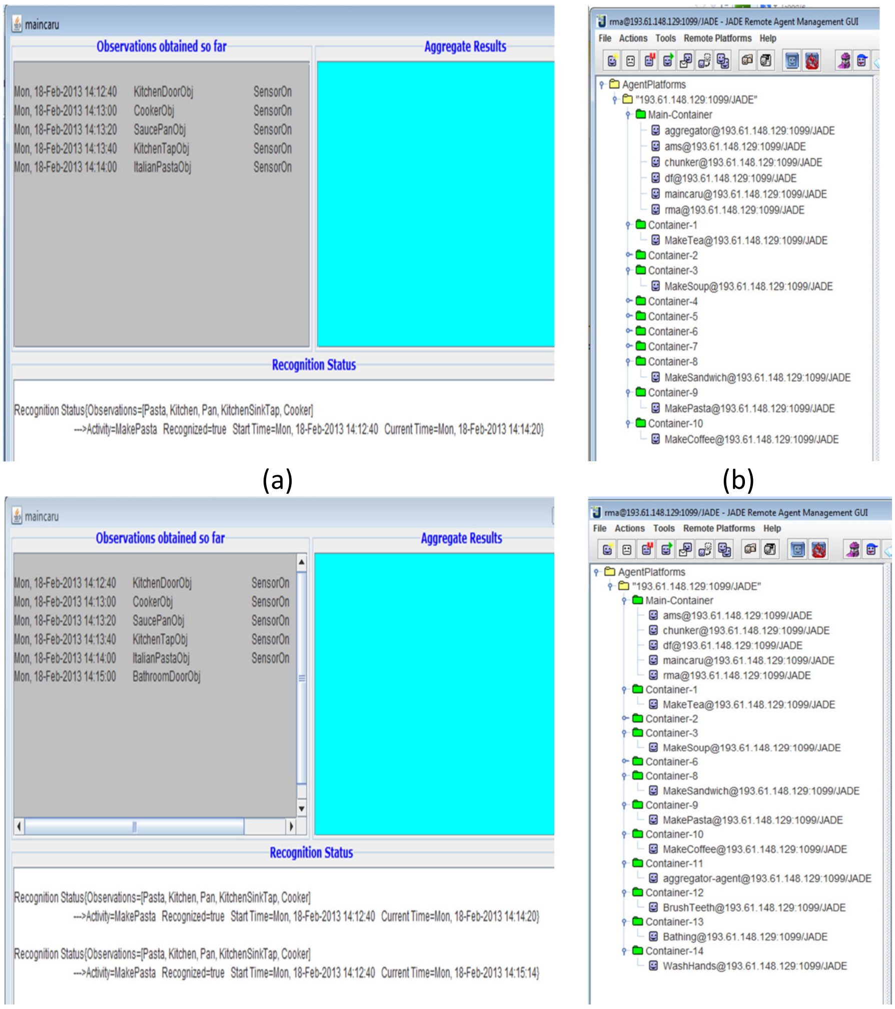
| **Concurrent and interleaved** | **Instances** | **Sequential** | **Instances** |
|---|---|---|---|
| MakePasta and MakeTea **(a)** | 3 | MakePasta then HaveBath**(g)** | 6 |
| MakePasta and WatchTelevision**(b)** | 5 | MakeTea then WashHands**(h)** | 4 |
| MakePasta and HaveBath**(c)** | 8 | WashHands 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** | **Total** | **25** |

*Table 1: Summary of composite activities in synthetic data set*

To facilitate experiments, the initial length of the time window was set tobe equivalent to the duration of the longest simple activity as provided in the ADL

Ontology, i.e., MakePasta. As provided in Okeyo et al. [Okeyo, 2013], the time window length would be modified at runtime either by being increased or reduced based on the prevailing status of activity recognition.

The 104 activities were played back in real-time and processed by the system prototype for activity recognition. A snapshot of the runtime agent system is shown in Fig. 8 during the composite activity *MakePastaAndHaveBath*. Fig. 8 (a) shows five sensor observations for the MakePasta activity. At the same time Fig. 8(b) shows the agent instances as obtained from JADE's remote management agent (RMA) facility. RMA provides a graphical user interface (GUI) facility for visualizing and managing



```
Mon, 18-Feb-2013 14:12:40    KitchenDoorObj     SensorOn
Mon, 18-Feb-2013 14:13:00    CookerObj          SensorOn
Mon, 18-Feb-2013 14:13:20    SaucePanObj        SensorOn
Mon, 18-Feb-2013 14:13:40    KitchenTapObj      SensorOn
Mon, 18-Feb-2013 14:14:00    ItalianPastaObj    SensorOn
```

```
Recognition Status{Observations=[Pasta, Kitchen, Pan, KitchenSinkTap, Cooker]
        --->Activity=MakePasta  Recognized=true  Start Time=Mon, 18-Feb-2013 14:12:40  Current Time=Mon, 18-Feb-2013 14:14:20}
```

(a)

rma@193.61.148.129:1099/JADE - JADE Remote Agent Management GUI
File Actions Tools Remote Platforms Help

```
AgentPlatforms
  "193.61.148.129:1099/JADE"
    Main-Container
      aggregator@193.61.148.129:1099/JADE
      ams@193.61.148.129:1099/JADE
      chunker@193.61.148.129:1099/JADE
      df@193.61.148.129:1099/JADE
      maincaru@193.61.148.129:1099/JADE
      rma@193.61.148.129:1099/JADE
    Container-1
      MakeTea@193.61.148.129:1099/JADE
    Container-2
    Container-3
      MakeSoup@193.61.148.129:1099/JADE
    Container-4
    Container-5
    Container-6
    Container-7
    Container-8
      MakeSandwich@193.61.148.129:1099/JADE
    Container-9
      MakePasta@193.61.148.129:1099/JADE
    Container-10
      MakeCoffee@193.61.148.129:1099/JADE
```

(b)

```
Mon, 18-Feb-2013 14:12:40    KitchenDoorObj     SensorOn
Mon, 18-Feb-2013 14:13:00    CookerObj          SensorOn
Mon, 18-Feb-2013 14:13:20    SaucePanObj        SensorOn
Mon, 18-Feb-2013 14:13:40    KitchenTapObj      SensorOn
Mon, 18-Feb-2013 14:14:00    ItalianPastaObj    SensorOn
Mon, 18-Feb-2013 14:15:00    BathroomDoorObj
```

```
Recognition Status{Observations=[Pasta, Kitchen, Pan, KitchenSinkTap, Cooker]
        --->Activity=MakePasta  Recognized=true  Start Time=Mon, 18-Feb-2013 14:12:40  Current Time=Mon, 18-Feb-2013 14:14:20}

Recognition Status{Observations=[Pasta, Kitchen, Pan, KitchenSinkTap, Cooker]
        --->Activity=MakePasta  Recognized=true  Start Time=Mon, 18-Feb-2013 14:12:40  Current Time=Mon, 18-Feb-2013 14:15:14}
```

(c)

rma@193.61.148.129:1099/JADE - JADE Remote Agent Management GUI
File Actions Tools Remote Platforms Help

```
AgentPlatforms
  "193.61.148.129:1099/JADE"
    Main-Container
      ams@193.61.148.129:1099/JADE
      chunker@193.61.148.129:1099/JADE
      df@193.61.148.129:1099/JADE
      maincaru@193.61.148.129:1099/JADE
      rma@193.61.148.129:1099/JADE
    Container-1
      MakeTea@193.61.148.129:1099/JADE
    Container-2
    Container-3
      MakeSoup@193.61.148.129:1099/JADE
    Container-6
    Container-8
      MakeSandwich@193.61.148.129:1099/JADE
    Container-9
      MakePasta@193.61.148.129:1099/JADE
    Container-10
      MakeCoffee@193.61.148.129:1099/JADE
    Container-11
      aggregator-agent@193.61.148.129:1099/JADE
    Container-12
      BrushTeeth@193.61.148.129:1099/JADE
    Container-13
      Bathing@193.61.148.129:1099/JADE
    Container-14
      WashHands@193.61.148.129:1099/JADE
```

(d)

*Figure 8: A snapshot of the runtime agent system*

JADE agents. In the main container, we can observe CARA agent (maincaru@193.61.148.129:1099/JADE), MSA agent (chunker@193.61.148.129: 1099/JADE), and AAA agent (aggregator@193.61.148.129:1099/JADE). From the five sensor observations that have been obtained, the CARA agent has spawned various agents to monitor the ongoing activity or activities and these are launched in agent containers- i.e. Container-1 to Container-10. For instance due to the fact that the user is in the kitchen (*given by the observation Mon 18-Feb-2013 14:12:40 KitchenDoorObj SensorOn*), it can be observed that CARA launches various SARA agents to monitor kitchen-based activities e.g., MakeTea@193.61.148.129:1099/ JADE on Container-1, MakeSoup@193.61.148.129:1099/JADE on Container-3, MakePasta@193.61.148.129:1099/JADE on Container-9, etc. Fig. 8(c) shows the observation *Mon 18-Feb-2013 14:15:00 BathroomDoorObj SensorOn* has been made. At this stage the CARA agent is shown to launch further SARA agents to monitor bathroom-based activities as shown in Fig. 8(d), e.g. BrushTeeth@ 193.61.148.129:1099/JADE on Container-12, Bathing@193.61.148.129:1099/JADE on Container-13, and WashHands@193.61.148.129:1099/JADE on Container-14. In the meantime, Fig. 8 (c) displays the result that the simple activity *MakePasta* has been identified, showing that it started at 14:12:40, when the first observation was made, and the current time is 14:15:14. This process proceeds as long as sensor data continues to be obtained.

### 5.4 Results and discussion

From the experiments, the results for precision, recall, and recognition accuracy are summarized in Table 2 and Fig. 9. The overall accuracy obtained for simple activities is 100% since all 104 simple activities were successfully recognized. This level of accuracy is attributed to the creation and use of activity descriptions as described in Section 3. An overall accuracy value of 88.26% was obtained for composite activities.

We can observe that the individual recognition accuracy is lowest for sequential composite activities that involve location transitions, e.g. *MakeTea then WashHands* (kitchen then bathroom), *WashHands then MakeTea* (bathroom then kitchen), *MakeTea then WatchTelevision* (kitchen then lounge), and *WatchTelevision then MakeTea* (lounge then kitchen). It is important to note that we introduced the 4D-fluents approach to support temporal modelling, representation, reasoning and querying for temporal knowledge in ontologies [Okeyo, 2012]. As a result we are able to model composite activities based on temporal inter-activity relationships and to infer composite activities from the temporal relationships. However, the modelling approach does not address reasoning and querying from spatial information. Therefore, we believe that the diminished recognition accuracy can be attributed to the inference and querying limitation for ontological spatial information. We believe the problem can be addressed by adopting a method for explicitly modelling spatial or location information in ontologies to facilitate spatial querying and inference.

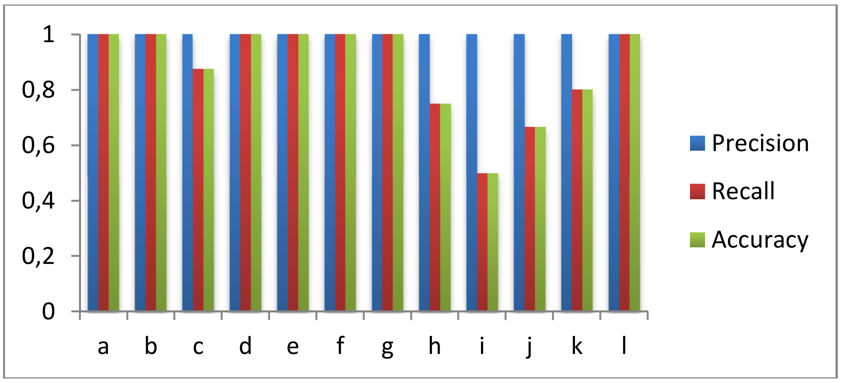


*Figure 9: Summary of precision, recall, and accuracy results for composite activities*

| Composite Activity | Alias | Precision | Recall | Accuracy |
|---|---|---|---|---|
| MakePasta and MakeTea | a | 100.0% | 100.0% | 100.0% |
| MakePasta and WatchTelevision | b | 100.0% | 100.0% | 100.0% |
| MakePasta and HaveBath | c | 100.0% | 87.5% | 87.5% |
| WatchTelevision and MakeTea | d | 100.0% | 100.0% | 100.0% |
| MakePasta and MakeChocolate | e | 100.0% | 100.0% | 100.0% |
| MakePasta and MakeCoffee | f | 100.0% | 100.0% | 100.0% |
| MakePasta then HaveBath | g | 100.0% | 100.0% | 100.0% |
| MakeTea then WashHands | h | 100.0% | 75.0% | 75.0% |
| WashHands then MakeTea | i | 100.0% | 50.0% | 50.0% |
| MakeTea then WatchTelevision | j | 100.0% | 66.7% | 66.7% |
| WatchTelevision then MakeTea | k | 100.0% | 80.0% | 80.0% |
| HaveBath then MakePasta | l | 100.0% | 100.0% | 100.0% |
| ***Average*** | | **100.0%** | **88.3%** | **88.3%** |

*Table 2: Precision, recall, and accuracy results of composite activity recognition*

## 6 Conclusion and Future Work

This paper presented an agent-mediated approach based on hybrid ontological and temporal formalisms for composite activity recognition. It described the architecture and mechanism to support the recognition of composite activities, e.g. interleaved and concurrent activities. In addition, the algorithms and a multi-agent system to support unified activity recognition was described. The implementation of a prototype that consists of activity ontologies, entailment rules, and an agent-mediated activity recognition system was also provided. Experimental results are presented with average recognition accuracy values of 88.26% and 100% for composite activities and simple activities, respectively. To the best of our knowledge, this research provides the first purely knowledge-driven approach that can infer both simple activities and composite activities. Future work involves conducting further experiments to assess the performance requirements, such as the time cost and memory requirements, of the algorithms and the agent-mediated system. We also plan to conduct additional experiments to compare the performance of our approach with other approaches using publicly available datasets. In addition, we will enhance the activity models to support the querying and inference with spatial information to address issues of reduced recognition accuracy for composite activities that involve location transitions.

## References

[AAL, 2012] J. AAL, "Ambient Assisted Living Joint Programme," http://www.aal-europe.eu/, 2012.

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

[Bellifemine, 2001] 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, pp. 216-217, 2001.

[Bucks, 1996] 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..

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

[Chan, 2009] 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.

[Chen, 2008] 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, pp. 81-9, 2008.

[Chen, 2009] 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.

[Chen, 2011] 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.

[FIPA, 2012] FIPA, "Foundation of Intelligent Physical Agents," http://www.fipa.org/, 2012.

[Fook, 2006] Victor Foo Siang Fook, Siew Choo Tay, Maniyeri Jayachandran, Jit Biswas, Daqing Zhang: An Ontology-based Context Model in Monitoring and Handling Agitation Behaviour for Persons with Dementia. PerCom Workshops 2006: 560-564.

[Grau, 2008] 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.

[Gu, 2011] 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.

[Helaoui, 2011a] 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, pp. 1-9, 2011.

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

[Hoof, 2011] J. van Hoof, H.S.M. Kort, P.G.S. Rutten and M.S.H. Duijnstee, "Ageing-in-place with the use of ambient intelligence technology: Perspectives of older users," Int.J.Med.Inf., vol. 80, pp. 310-331, 2011.

[Horrocks, 2005a] 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, pp. 5-8, 2005.

[Horrocks, 2005b] 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.

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

[Jennings, 1998a] N.R. Jennings and M. Wooldridge, " Applications of intelligent agents," in Agent technology, N.R. Jennings and M. Wooldridge, New York, Inc., Secaucus, NJ, USA: Springer-Verlag, 1998, pp. 3-28.

[Jennings, 1998b] N.R. Jennings and M.J. Wooldridge Eds., Agent Technology: Foundations, Applications, and Markets, Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1998.

[Jennings, 2001] N.R. Jennings, "An agent-based approach for building complex software systems," Commun ACM, vol. 44, pp. 35-41, 04. 2001.

[Jess, 2012] E. Friedman-Hill, "Jess, the Rule Engine for the JavaTM Platform," 2012.

[Kasteren, 2008] 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, pp. 1-9, 2008.

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

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

[McKeever, 2010] 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.

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

[Modayil, 2008] 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, pp. 40-43, 2008.

[Okeyo, 2012] 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), pp. 1763-70, 2012.

[Okeyo, 2013] G. Okeyo, L. Chen, H. Wang and R. Sterritt, "Dynamic sensor data segmentation for real-time knowledge-driven activity recognition,", Pervasive and Mobile Computing, Elsevier, DOI: 10.1016/j.pmcj.2012.11.004, 2013.

[Patterson, 2005] 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, pp. 44-51, 2005.

[Pellet, 2011] Clark and Parsia, "Pellet: OWL 2 Reasoner for Java," http://clarkparsia.com/pellet/ ,2011 .

[Philipose, 2004] 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.

[Protege, 2011] Stanford Center for Biomedical Informatics Research, "The Protege Ontology Editor and Knowledge Acquisition System," URL=http:// http://protege.stanford.edu/ , 2011.

[Riboni, 2011] 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.

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

[Steinhauer, 2010] 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, pp. 54-59, 2010.

[Strorf, 2009] 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.

[Wang, 2004] X.H. Wang,Daqing Zhang, T. Gu, H.K. Pung, "Ontology Based Context Modeling and Reasoning using OWL", In Proceedings of CoMoRea, the Second IEEE International Conference on Pervasive Computing and Communications (PerCom 2004), Orlando, Florida USA, March 2004.

[Welty, 2006] C. Welty and R. Fikes, "A Reusable Ontology for Fluents in OWL," in Fourth International Conference on Formal Ontology in Information Systems, pp. 226-236, 2006.

[Wooldridge, 1995] M. Wooldridge and N.R. Jennings, "Intelligent agents: theory and practice," Knowl.Eng.Rev., vol. 10, pp. 115-52, 06. 1995.