Ontology Evolution for Personalized and Adaptive Activity Recognition

Muhammad Safyan[1], Zia Ul Qayyum[2], Sohail Sarwar[2], Muddesar Iqbal[3*], Raul Garcia Castro[4], Anwer Al-Dulaimi[5]
[1] GC University, Lahore, Pakistan
[2] University of Gujrat, Gujrat, Pakistan
[3] London South Bank University, England
[4] Universidad Politecnica de Madrid, Spain
[5] EXFO, Toronto, Canada

[*]m.iqbal@lsbu.ac.uk

Abstract: Ontology-based knowledge driven Activity Recognition (AR) models play a vital role in realm of Internet of Things (IoTs). However, these models suffer the shortcomings of static nature, inability of self-evolution and lack of adaptivity. Also, AR models cannot be made comprehensive enough to cater all the activities and smart home inhabitants may not be restricted to only those activities contained in AR model. So, AR models may not rightly recognize or infer new activities. In this paper, a framework has been proposed for dynamically capturing the new knowledge from activity patterns to evolve behavioural changes in AR model (i.e. ontology based model). This ontology based framework adapts by learning the specialized and extended activities from existing user-performed activity patterns. Moreover, it can identify new activity patterns previously unknown in AR model, adapt the new properties in existing activity models and enrich ontology model by capturing change representation to enrich ontology model. The proposed framework has been evaluated comprehensively over the metrics of accuracy, statistical heuristics and Kappa Coefficient. A well-known dataset named DAMSH has been used for having an empirical insight to the effectiveness of proposed framework that shows a significant level of accuracy for AR models.

1. Introduction

Human Activity Recognition (HAR) determines the activities that have been performed by humans based upon certain knowledge and context. Earlier, Activity Recognition (AR) was performed by observing and analyzing human activities through surveillance cameras. Such manual observation driven AR seemed cost-intensive and demanding around the clock e.g. personnel deployment in home care services was infeasible financially. However, automated HAR systems resolved the issues by providing efficient and cost-effective measures instead of human-centered observations and analysis. Continuous scientific and technical progress has directed the human expectation from HAR toward Personalized HAR [2] for personalized service provision. A rich growth of data-driven and knowledge-driven modeling techniques have been proposed in [3] [4] [5]. Limitations of data-driven problems are cold start problem and non-reusability [1] [3]. Whereas, knowledge-driven techniques are static in nature, incomplete, and non-adaptable [1] [4]. One recent contribution in knowledge-driven techniques is based upon ontologies [5] [6]. Compared with the rest of the approaches, ontology-based models provide a higher degree of automation, better reasoning ability and solid technological foundations but still lacking the self-evolution. In this paper, we extend our work described in [7] for ontology evolution. Proposed ontological model for Activity Recognition (AR) adopts hybrid activity modeling approach (knowledge-driven and data-driven) in which seed knowledge about activities is modeled in an ontology. Seed knowledge comprises of a set of actions necessary to perform an activity called Perceptible Activity Models (PAMs). Model described in [7] transfer the sensor stream into action properties. This sensor stream is used with different ontological contexts such as duration, location, object type, temporal dependencies among actions and feature-based semantic similarities [7] to recognize personalized activity patterns.

Practically, in activity modeling, it is not possible to completely model all the activities at once due to the following reasons:

- Inhabitants of the smart homes are not restricted to perform specific activities modeled in the ontology. Instead, they may perform activities (with existing home-objects) which have not been modeled in knowledge base yet.
- Installation of new home-objects may introduce new activities to be performed at home.
- Underlying knowledge provided by domain experts may change from inhabitant to inhabitant and may trigger the process of ontology evolution e.g. duration to perform an activity or location change etc.

Ontology needs to undergo continual up-gradation to new concepts or modifications of existing concepts due to the reasons stated above. This process is called ontology evolution or ontology enrichment [8].

Various research groups are working in the area of ontology evolution and their evolution lifecycle phases overlap with each other [9], [10], [11]. An agreed-upon evolution process is roughly categorized into six distinct phases performed in sequence (i) Capture change (ii) Change representation (iii) Semantics of change (iv) Change implementation and verification (v) Change propagation and (vi)Change validation

Change capturing phase refers to the need for identifying change; these changes are formally represented during the change representation phase. The third phase is the semantics of change in which the effects of the change(s) to the ontology itself are determined and resolved. The change implementation phase applies changes to ontology physically. The ontology engineer is directed to log these changes. These changes need to be propagated to dependent elements, carried out in change propagation phase. Finally, the change validation phase allows the ontology engineer to review changes and fix them if required.

One of the major challenges in ontology evolution is to model the process of capturing the change. Capturing the change process requires a thorough analysis of the domain and domain-specific policies to enrich the model. This paper proposes a statistical based algorithmic solution to analyze the labeled activities along with action sequence to capture, represent and implement change for ADL ontology.

In order to capture a change, Stojanovic [12] categorized capturing change processes into four types: (i) Structure-driven: refers to changes in an ontology structure e.g. A class having only one child class should be merged with the superclass. (ii) Data-driven: corresponds to changes observed during analysis of instances in an ontology e.g. A class with many instances should be split into subclasses. (iii) Usage-driven: Changes are inferred from the usage of concepts in the ontology. For example, classes that have not been used for a long time should be deleted (iv) Discovery-driven: refer to the changes when an instance is not properly described.

In this paper, a novel discovery-driven AR ontology evolution approach has been proposed to support the continuous learning of the ontology model. We have already developed a seed ontology described in [7] specifying a generic set of actions for each activity called perceptible activity model (PAMs) from which personalized/complete set of actions (i.e. CAMs) for particular activities are derived. These personalized patterns along with their labels are stored in a log file. If a set of actions is not recognized by the model then it is labeled as "unidentified". The log file patterns are discovered for ontology evolution by the proposed framework. For example, PAMs may learn a specialized activity of the existing activity or it may learn a new activity from the unidentified patterns. Figure 1 shows the conceptual views of ADL ontology evolution.

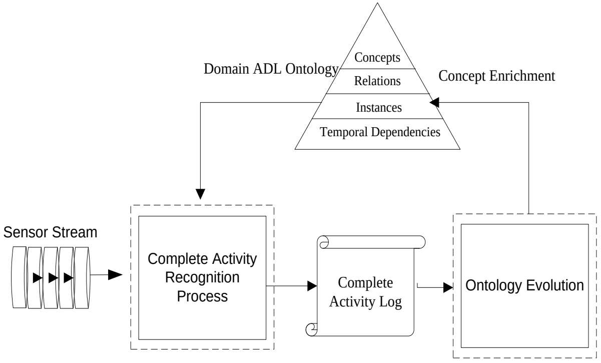

***Fig.1**. Conceptual view of ADL ontology evolution*

In summary, the proposed framework is envisaged to have the following features:

(i) Ability to learn new activities by exploiting the context of existing activities and activity patterns in log file. For instance, it may learn the specialized activity of existing activity or a new activity from unidentified patterns

(ii) Ability to reuse the existing knowledge of ontology to learn an activity

Enable the model to be adaptable and flexible. Adaptability and flexibility enable the model to dynamically update its existing activity model like an action sequence, duration etc.

2. Literature review

Noy, N and et al. [13][33] compare the database schema evolution with ontology evolution and conclude that ontology evolution is similar to object-oriented database schema and semantically richer than database schemas due to it inheritance principles and conclude that ontology evolution approaches are a kind of extension rather than an adaptation of existing approaches. Solutions controlling change consequence are based on rules to satisfy to fulfill to keep up consistency and inference mechanisms based on axioms. Ontology change goals join (depending on ontology language) these two complementary solutions.

[] describes general ontology evolution approaches that can be applied in different domains such as social media, archeology, automotive diagnosis, software bug reports, event management and university management domains. Moreover, a framework is proposed for a comparative study to demonstrate the differences between the approaches

Ontology evolution phases and their issues have been described by the different researchers such as capturing change requirements described by [14] [15], change detection and version logging described by[16], formal change specification described by [12], change implementation and consistency maintenance described by [13].

According to Khattak A., M and et al [17]. The evolution in ontology is mainly of two types i.e., Ontology Population and Ontology Enrichment. Ontology Population adds new instances for already existing concepts. Here only the new instance(s) of the concept is introduced and the ontology is populated. Ontology Enrichment implies changes in the structure of the ontology. For instance when we get a new concept(s), which is absolutely new for our ontology or the idea has a type of changes from its counter concept in the ontology. By then new suited changes are applied to enrich ontology for its instance(s)

Chen, L and et al. [18] proposed the ontological model for smart home in which domain ontology learn from large scale data pertaining to inhabitant activity daily life are caught and mined over an extensive stretch of time. Identify the regular activities pattern along with their descriptive properties as a personalized model. The agent can use the learnt activity to grow ADL ontologies as such an activity model can evolve and subsequently improve the performance of activity recognition.

Gorka and et al. [19] proposed activity recognition and learning model. Recognition module recognizes the inhabitant personalized activity sequence from the generic model encoded in the domain ontology. It only works for sequential activity. Learning module evolute the ontology with specialization of concepts only by observing the frequent pattern of inhabitant and add them in a text file for the future recognition process. The core idea is to fuse the similar patterns through cosine similarity matrix , Find the outlier regular patterns through statistical technique and present to the domain experts to endorse them as a new activity.

Okeyo and et al. [20] presents a framework that mines the label and unlabeled activities pattern stored in the log file. Unlabeled activities are the patterns for which an activity recognition system is unlabeled to identify them. A pattern having occurrence frequency greater than average are considered a candidate as a new activity in the ontology. The author did not consider an action pattern and action-value pattern and sensor noise as three separate metrics to identify the regular pattern.

Our proposed technique is different from the [19][20]. We evolute the model by learning from action properties, object values, sensor noise and unidentified pattern. We learned, not only specialized activities but extended activities and new activities whose coarse grain actions partially exist in two different activities.

Javed and et al. [21] focused on the analysis of ontology change logs formalized as attributed graphs. The objective of this approach is the discovery of reusable domain specific change patterns that can be used in existing knowledge management systems. The approach relies on the use of change logs to reflect all the changes that are explicitly occurred on the ontology entities.

Shufeng ,W and et al. [22] proposed an approach for ontology evolution based on the use of folksonomy. Its objective is the extraction of potential latent semantics provided by folksonomies. Folksonomy is composed of two words; “folk” and “taxonomy”. Folksonomy reflects the vocabulary terms used among users. Vander [23] has defined the folksonomy as “the aftereffect of individual free labeling of data and objects for one’s own retrieval. The labeling is done in a social domain (normally shared and open to other people). Folksonomy is made from the demonstration of labeling by the individual devouring the data". The methodology accepts folksonomy and ontology as data sources and delivers a folksonomies philosophy as a yield. It demonstrates the three steps of the proposed ontology evolution approach are Extraction, Enrichment and Evolution.

The DYNAMO-MAS approach was proposed by Sellami, Z and et al. [24]. Its objective is to help the ontology engineer to determine the relevant data extracted from the text that can be used to evolve ontologies. The DYNAMO approach is based on the use of Adaptive Multi-Agent Systems (MAS or AMAS). These agents are used to provide the ontology being modified by the extracted data from the text.

Haase and et al. [25] have defined the ontology evolution as the process that “adapt and change the ontology in a consistent way”.

Mariem M and et al. [26] [29] discussed the ontology model for formalizing ontology changes which present the inconsistencies that can occur due to the ontology changes. Inconsistencies involve data redundancy, isolated nodes, orphaned individuals and axioms contradiction. The idea is based on the use of Typed Graph Grammar (TGG) to represent the evolved ontology. The inputs of this approach involve the graph and a set of rules. The user codifies manually the rules by the use of Attributed Graph Grammar (AGG) tool to ensure the consistency of the changes that were occurred on the ontology. The outputs are possible inconsistencies.

Stojanovic, L. et al. [27] talked about different ontology editors , their limitations, and complexities, and ease of use issues of these tools for ontology evolution administration. As refinements are required by ontology, so it must be updated by making appropriate changes in it. Therefore, methods to cope with the changes that result from evolution are an essential requirement for ontology editors. paper.

Kondylakis, H. and et al. [30] exploits provenance queries to dynamically explores the evolution of RDF/S ontologies. We construct an ontology of changes for modeling the language of changes and we store all changes as instances of this ontology in a triple store. A protégé plugin and visualization tool have been developing for exploration purpose.

3. Proposed Approach for ontology evolution

In order to provide an agreed upon vocabulary to architectural components and discussion, some important definitions have been devised for implementing the concept as given in the following:

Definition 1: Action Sequence: is a set of actions (pattern) labeled after the recognition process.

Definition 2: Candidate Action Sequence (CAS): is a distinct regular action sequence pattern whose occurrence frequency more than average occurrence frequencies among all the patterns of activity. CAS are potential patterns for capturing the change process.

Definition 3: Specialized sub-Activity: is a sub-activity having the same properties as of its parent class/activity but their properties range/restriction is different from its parent class.

Definition 4: Extended sub-Activity: is a sub-activity having more properties than of its parent class.

Definition 5: Coarse-grained Activity: is a concept placed at the root or intermediary level in ontology holding some properties that inherits into their child level concept. Coarse-grained activities are generalized concepts to determine the context of patterns and are not usually assigned to patterns performed by the inhabitants

Definition 6: Fine-grained Activity: is a concept placed at leaf level in a hierarchy and is directly performed by the inhabitant.

Definition 7: sensor noise: sensor stimulation due to the user’s mistaken interaction with objects that are not part of any ongoing activity.

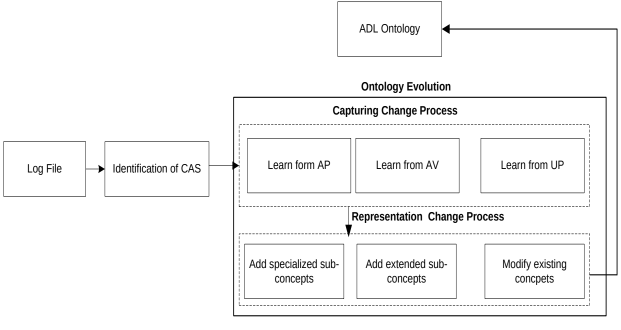

***Fig.2**. ADL Ontology Evolution Framework*

Assumption 1: Installation of sensors attached to objects: When a new object is installed at home, it must be associated with senor and appropriate level of information is provided to the system such as sensor type, home-object type to which like cup, kettle etc. and list of activities in which home-object may be used.

Figure 2 shows the proposed model in which a log file is taken as input. It is processed by the Candidate Action Sequence (CAS) module which exploits a frequency-based statistical algorithm to determine the potential patterns for learning a model. These CAS patterns are passed-on to different components of the ontology evolution process in a sequence which learn the activities in three aspects: (i) Learn from Action Properties(AP) (ii) Learn from Action Value(AV) and (iii) Learn from Unidentified Patterns(UP) and enrich the update the existing ontological model with new activities or modifying the existing activities.

3.1. Identification of Candidate Action Sequence(CAS)

In object-based personalized activity modeling, the usage of the objects in a unique way provides a heuristic to enrich ontology. If a person gets started to perform an existing activity in a specific way on regular basis other than the previous routine. Such patterns need to process either they are a new specialized or extended version of an existing activity? The core idea is to calculate the frequency of such patterns. If the occurrence frequency of such distinct patterns is greater than the average occurrence frequency of all previous patterns of an activity, then they should be considered for ontology evolution. Such patterns are called the Candidate Action Sequence (CAS).

The frequency of each pattern and its significance can be calculated with the following algorithm in listing 1.

Listing 1. Algorithm for Candidate Action Sequence(CAS)

```
Input: Labeled log file
Output: Candidate action sequence(CAS)
Begin
Let's
A Tagged Activity ($S_i$) having an action sequence <$s_1$, $s_2$, $s_3$…….. ….  $s_n$> and
K is the total No. of action sequences against a tag/label in log file is:
    $K = total\ no.\ of\ actions\ sequence = count(Si)$
Let's  $DAS_i$ be a Distinct Actions Sequence and M is total No. of distinct action sequences in S
        $M = \text{total no. of distinct actions} = \text{count(DAS)}$
        $T_i$ be the total No. of action sequences of against each $DAS_i$
        $T$    $i =$ total no. of Distinct action sequence in DASi
Frequency ratio $FR_i$ of each $DAS_i$
    $FRi = \frac{Total\ no\ of\ occurrence\ against\ DASi}{Total\ no\ of occurrence\ against\ Si} = \text{Ti/k}$
Average Frequency Ratio of pattern sequence.
        $\text{A}R = \frac{Sum(FRi)}{M}$
Candidate action sequence ($CAS_i$) =
$if\ FRi > AR$   ⇨ $return(DASi € CASi)$
    $else$     ⇨ $ignore\ sequence$
End
```

CAS patterns are passed to different modules for the ontology evolution process in a sequence for learning the ontology model as shown in Figure 2.

3.2. Learning from Action Values(AV)

Installation of new home-objects may trigger to the learning of new sub-concepts as a specialized activity. For example, if a home-object says “green tea” is newly installed at home under the category “tea-type”. Inhabitant may start making “green tea” instead of using regular tea i.e. “brown tea”. Regular use of both “brown tea” and “green tea” in a generic action sequence of “making tea” provides a heuristic for learning two new specialized sub-concepts of making tea i.e. “making brown tea” and “making green tea” with added property restrictions “*hasteatype”* brown and “*hasteatype*” green respectively.

To learn specialized sub-concepts, only action-values are considered as learning metrics. Identification process extracts the Candidate Action Sequence (CAS) on the basis of distinct action-values of PAM’s actions. In this case, ontology is enriched only when at least two Candidate Action Sequence (CAS) exist. Property restrictions on action-properties are implemented by getting the difference of action-values between the candidate action sequences.

Let, two Candidate Action Sequences S_i and S_j

$$V_1 = S_i - S_j \quad \text{and} \quad V_2 = S_j - S_i$$

Range restriction of the properties is V1 and V2 in Si and Sj specialized sub-concepts.

Figure 3 shows the snapshot of the original ontology at left side while the right side shows the resultant ontology after learning two specialized activities of making tea activity i.e. making green tea and making brown tea.

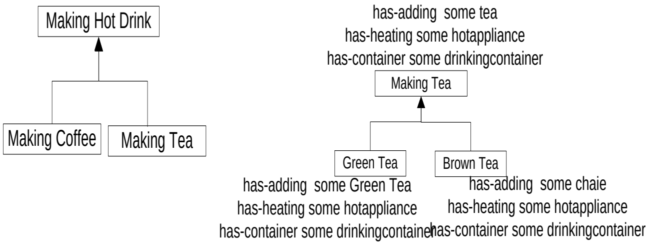
Making Hot Drink
Making Coffee
Making Tea
has-adding some tea
has-heating some hotappliance
has-container some drinkingcontainer
Making Tea
Green Tea
Brown Tea
has-adding some Green Tea
has-heating some hotappliance
has-container some drinkingcontainer
has-adding some chaie
has-heating some hotappliance
has-container some drinkingcontainer

Fig.3. *ADL Ontology learning from Action-values.*

Coarse-grained activity label in log file also provides heuristics to learn newly specialized sub-concept. The coarse-grained label is assigned to an action sequence when one or more action-values of sequence do not satisfy the range of the properties for its fine-grained activities. Coarse-grained label is assigned only when the relationship between the super and sub-concepts exist in a specialized manner. For example, if an inhabitant starts making pink tea then the action sequence of making tea is not satisfied by any of two specialized sub-activities of making green tea and making brown tea. In such cases, the activity recognition module labels the pattern for which its patterns are satisfied i.e. making tea. Such coarse-grained labeled patterns may extend to a specialized sub-activities having their own property restriction. Figure 4 shows another specialized sub-concept named “pink tea” with property restriction “hasteatype pink”. The algorithm for ontology enrichment with action value is presented in listing 2.

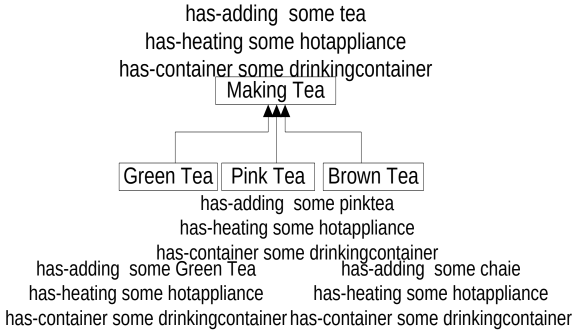
has-adding some tea
has-heating some hotappliance
has-container some drinkingcontainer
Making Tea
Green Tea
Pink Tea
Brown Tea
has-adding some pinktea
has-heating some hotappliance
has-container some drinkingcontainer
has-adding some Green Tea
has-heating some hotappliance
has-container some drinkingcontainer
has-adding some chaie
has-heating some hotappliance
has-container some drinkingcontainer

Fig.4. *Learning sub-activity from the coarse-grained label.*

Listing 2. Enrich Ontology with Action Values

```
Input: candidate action sequence, domain ADL ontology
Output: specialized concepts
begin
If (label == Fine-grained) && CAS >= 2) || (label == Coarse-grained) Then
        {
        →Add two New Sub concept with same property of Fine-
            grained Concept.
        →Add property restriction on actions having variation of
            activity value.
        }
End if
return (specialized concept)
end
```

3.3. *Learning from Actions Properties (AP)*

In personalized activity modeling, while performing an activity, it is necessary to perform generic action sequence whereas user-specific actions are optional and may vary based on inhabitants. For example, activity “making tea” with two personalized action sequences could be as follows:-

1. *hasheating(stove), hasadding(liquidmilk), hastea(brown), hascontainer(cup) , hasutensil(kettle),hassadding(sugar), hasadding(milk)*
2. *hasheating(stove), hasstrainer(strainer), hastea(brown), hascontainer(cup) , hasaddingl(sugar), haswater(water), haadding(milk)*

First sequence completes personalized model with user-specific actions {hasadding(liquidmilk), hasutensil(kettle), hasadding(sugar)} while second completes the model with actions { hasstrainer(strainer), hasutensil(kettle), haswater(water) }. The occurrence frequency of optional actions in the action sequence provides two heuristics to extend or modify the concepts.

1. If the occurrence frequency of user-specific actions is equal to more than average occurrence frequency of generic action sequence of activity, add that optional action into a generic action sequence.
2. If an action in positive sensor noise having occurrence frequency greater than or equal to the average occurrence frequency of generic actions sequences. Modify the used-in property of noisy actions by extending its range with under process activity. Add the actions into a generic action sequence. Otherwise, compare its occurrence frequency with an average occurrence frequency of optional sequences. Also, the frequency of action sequence is greater than previous ones, the activity-type property of noisy action by extending its range under process activity. Algorithm for enriching ontology with action property is illustrated in Listing 3.

3.4. *Learning from Unidentified Pattern (UP)*

AR systems may not assign labels to action sequences that do not match with any generic sequence of domain ontology due to two reasons i.e. positive sensor noise and inhabitant performed an activity that is not encoded in the domain ontology.

Listing 3. Enrich Ontology with Actions Properties

```
Input: candidate action sequence, domain ADL ontology
Output: enrich ADL ontology with specialized concepts
Begin
Let a Tagged Activity ($S_i$) with a personalized action sequence
<$s_1$,$s_2$, $s_3$………… $s_n$ $o_1$,$o_2$, $o_3$…….. ….   $o_n$ >
Where $s_i$ are generic action sequence and $u_i$ are user-specific actions.
K is the total No. of action sequences S is:
        $K = count(S)$
        $M_i$ is occurrence frequency of a single generic action sequence $s_i$ in S
        $Mi = count(Si)/K$
        Average occurrence Frequency of Generic Action Sequence= FGS= $average(Mi)$
        $N_i$ is occurrence frequency of a single user-specific actions $u_i$ in S = $Ni = count(ui)/Mi$

        Average occurrence Frequency of user-specific Action Sequence FOS= $count(ui)/Ni$

        Let’s $Q_i$ is occurrence frequency of a single Positive Sensor Noise in $S_i$
        $Qi = count(ni)/k$
If (Ni>FGS) then {
        ⇨ DASi € CASi
Add a sub concept of under process activity with
        GAS= {GAS of under process activity+ action $N_i$}
Else if ($Q_i$>FGS)
        Extend the range of the Home-object activity-type property by the under process activity.
        Add a sub-concept of under process activity with
        GAS= {GAS of under process activity+ action $Q_i$}
Else if (Qi>FOS)
        Extend the range of the Home-object activity-type property by the under process activity.
}
End if
return (ADL ontology)
End
```

In the case of sensor noise, AR model is unable to assign the label to action sequence and labeled with

"Unidentified". Such patterns can never be a regular action sequence and cannot be selected as CAS. The second case is applicable when inhabitants perform an activity and that activity is not encoded in the domain knowledge. Unidentified labeled action sequence comprises of the actions belonging to multiple activities but none of the generic action sequence is complete. For example, generic action sequence of activities "making Russian salad", "making milk", "making tea" are given below:-

making Russian salad={hasFruit(fruit), hasvigitable(vegitable) , hascontainer(bowl)}
making milk={hasmilk(milk), hascontainer(cup)}
making tea={hasheating(stove), hastea(black), hascontainer(cup), hasweetening(sugar)}

Let's observe an unidentified action sequence:-
hasFruit(fruit), hasmilk(milk), hasblender(blender), hascontainer(cup), hasweetening(sugar), hasice(ice)

If such action sequences exist in some regular fashion, they need to be modeled in the ontology.

For the purpose of change representation, the position of concepts in the activity hierarchy of ontology is determined. In the above example, it is clearly observed that pattern is of "cold drink" being the reason for the "cup", "milk" and "ice" are the properties of "drink". Whereas, cold drink activities are linked with fruit, blender, and juice.

The only way to find the position of a new concept in the hierarchy is to measure the similarity between unidentified patterns and all generic action sequences of domain ontology. The one having the maximum similarity with the generic action sequence is the sibling of unidentified concept.

In order to find the similarity among the action sequences, pattern matching techniques are required not considering the order of the actions. Among well-known techniques for pattern matching are: Principal component analysis [27], Jacquard coefficient [27], Sørensen–Dice index [28], Tversky's similarity [29].

In proposed work, Tversky's similarity algorithm is used to find the similarity between two action sequences. Let S1 and S2 are two action sequences of the same activity, similarity matching can be calculated as:

$$simT(S_1,S_2)=\frac{S_1\cap S_2}{(S_1-S_2)+(S_2-S_1)+(S_1\cap S_2)} \quad (1)$$

Where
$S_1\cap S_2$ Are the common values of S1 and S2.
(S_1-S_2) Value in S1 not in S2
(S_2-S_1) Value in S2 not in S1

Similarity among every possible pair of action sequences can be calculated by the so-called Similarity Matching Matrix (SMM), which is a matrix of all action sequences in ontology and unidentified CAS patterns.

$$TM=\begin{bmatrix} TS\{U_0,S_0\} & - - - - - - - & TS\{U_0,S_n\} \\ TS\{U_1,S_0\} & - - - - - - - & TS\{U_1,S_n\} \\ - & & - \\ - & & - \\ - & & - \\ TS\{U_n,S_0\} & - - - - - - - & TS\{U_n,S_n\} \end{bmatrix}$$

Fig.5*. Similarity Matching Matrix*

Statically, the highest value of each row shows the maximum similarity between an action sequence Si and unidentified label ui. Change representation has the Sibling relationship between Ui, Si and GAS of Ui is a generic action sequence of Si and Suggestive action by a domain expert. Algorithm for ontology enrichment with unidentified patterns is presented in listing 4.

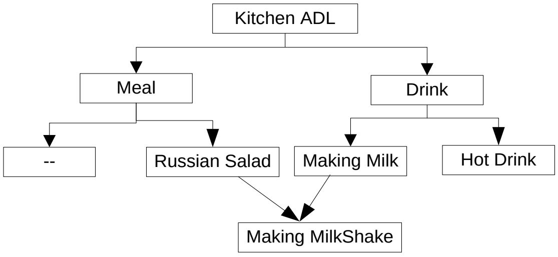

Fig.6*. Unidentified pattern evolution*

Listing 4. Enrich Ontology with Unidentified patterns

```
Input: candidate action sequence, domain ADL ontology
Output: enrich ADL ontology with specialized concepts
Being
Foreach (Ui)
{
Calculate Foreach(Si)
          {
                    HighSim = 0
Sim=Sim (Ui, Si)
                    IF (sim) > HighSim )
HighSim=sim
          }
End loop
Create Sibling Concept of Si with GAS = GAS(Si) + Domain Expert
}
End loop
return (ADL ontology)
End
```

4. Result and Evaluation

In order to evaluate the performance of the proposed framework and have an empirical view of its effectiveness, baseline datasets described in [1] named Data Acquisition Methodology for Smart Homes (DAMSH) has been used. Dataset acquisition statistics such as ground truth facts, unlabelled data acquisition source files and dataset file are available online: *https://www.dropbox.com/sh/28e7ca7gdeap7d9/AAAFGhIcXmdEmJIYh5fTxAVya?dl=0*

DAMSH contains 2.483 sensors stimulations in 111 days. A total of 10 distinct activities have been performed in a parallel and sequential fashion. The parallel activities in the dataset are "watching tv", "making tea", "making coffee", and "making pasta". Among the sequential activities are "taking nap", "chores", "shaving", "bathing", "taking medicine", and "washing cloth". The dataset contains 10% sensor noise of actual data. Sensor noise is unbiased and generated without any human interruption by the random () function. DAMSH code, sensor noise stimulation code and software manual are available online:

https://www.dropbox.com/sh/vis7da2hi0f8fa9/AABSoRDpUuqpCKlDOEXy4hiEa?dl=0. Table 1 shows the summary statistics of the data.

Evaluation is performed on patterns of specific activities extracted from the data set to evaluate the different components of the proposed system. We targeted the specific parts of ontology for evolution. For the first experiment, we targeted the part of ontology shown in figure 6 which considers ADL Kitchen activities as shown in Figure 7.

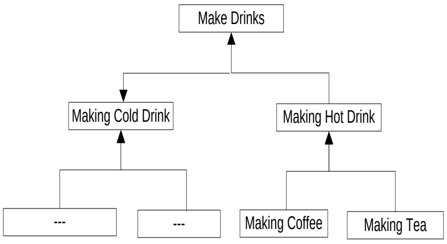

Fig.7. *Snapshot of kitchen ADL ontology*

Action sequence (patterns) in the log file for activity "making a hot drink (30)" and its sub-activities "making tea (54)" and "making coffee (59)" have been taken by the system as input as shown in Figure 7.

Table 1. Dataset statistics

Data set characteristics	- Randomize sequence of actions for activities of each day - Contains 10% sensor noise of actual data - Unbiased data, generated by using contextual knowledge of activity ontology, and sensor noise by random() function - Single variant	No. of days.	111
Parallel activities	- Watching tv with making tea - Watching tv with making pasta - Making tea with making coffee - Making pasta with making tea	No. of activities occurrence	- making pasta(70) - making tea(111) - making coffee(37) - watching Tv(111) - taking nap(134) - chores(100) - shaving(70) - bating(50) - taking medicine(30) - washing cloth(10)
Total No. of sensor occurrence	2.483	Sensor noise	10% of the actual data set

The first experiment shows the evolution of fine-grained activities "making tea" and "making coffee" and then on coarse-grained activity "making the hot drink".

Targeted patterns have been processed by Candidate Action Sequence (CAS) module (as elaborated in listing 1) to determine the patterns having frequency more than the average frequencies of distinct patterns. Four distinct patterns of "making tea" and six distinct patterns of "making coffee were found. Thereafter, the frequency ratio of each pattern has been calculated. Finally, CAS is selected. Table 2 states the CAS for making tea activity i.e. tea-pattern3 and tea-pattern4.

Table 2. Occurrence Frequencies of Fine-grained activities

Action Sequence	Occurrence Frequency	Frequency of pattern	Average Frequency	Candidate Action Sequence
Tea-pattern1	2	.037	.246	Tea-pattern3 Tea-pattern4
Tea-pattern2	2	.037		
Tea-pattern3	20	.37		
Tea-pattern4	30	.54		
Coffee-pattern1	1	.017	.167	Coffee-pattern4 Coffee-pattern5
Coffee-pattern2	3	.051		
Coffee-pattern3	5	.085		
Coffee-pattern4	20	.34		
Coffee-pattern5	25	.42		

Similarly, two candidate action sequences were shortlisted for making coffee activity i.e. coffee-pattern4 and coffee-pattern5 are extracted as CAS. Resultant ontology as shown in Figure 8 (following the process of the algorithm given in listing 2), has two specialized activities "making brown tea" and "making green tea"

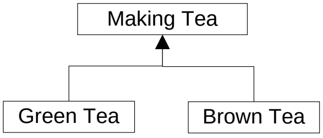

Fig.8. *Fine-grained evolution for making tea activity*

The second experiment was performed for the coarse-grained activity label making tea (30). Twenty four out of thirty patterns have a specific value for a property "has adding" i.e. "pink tea". The reason for the unsatisfied range restriction of fine-grain activities, the coarse-grained label is assigned. For example, one pattern of "making tea" is *has adding(pinktea), hashotappliance(stove) hasadding(milk).* Action property sequence is the same as that of "making brown tea" except one property *range* of *hasadding* i.e. the object *hasadding pinktea* having *range* *hasadding browntea*(in "making brown tea" activity). In such cases, ontology needs to evolute with new activity having a satisfied range of properties. Table 3 depicts the process of CAS for making tea activity. Figure 9 shows the resultant snapshot of ontology after evolution coarse-grained activity.

Table 3. OccurrenceFrequencies of Coarse-grained activities

Action Sequence	Occurrence Frequency	Frequency of pattern	Average Frequency	Candidate Action Sequence
Tea-pattern1	1	.037	.246	Tea-pattern4
Tea-pattern2	1	.037		
Tea-pattern3	2	.37		
Tea-pattern4	20	.54		

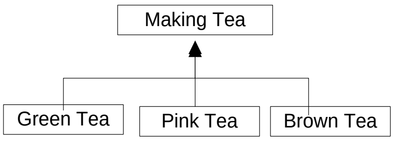

Fig.9. *Coarse-grained evolution for making tea activity*

The third set of experiments was performed for the evolution of extended sub-activities from the regular presence of sensor noise or optional sensors in patterns.

Sensor noise is a mistaken interaction with objects that are not present in pattern persistently. If there is a regular noise in patterns then it is an indication that inhabitants are changing their behavior and started using an object in an activity for which knowledge is not encoded in the ontology.

In order to evaluate this scenario, two activities were selected i.e. "making brown tea" and "making pink tea". Perceptible activity models of both activities are given below:
Making brown tea: has adding(tea), has adding(milk), hashotappliance(stove)
Making pink tea: has adding(milk), hashotappliance(stove), has adding(pink)

In "making brown tea", use of optional objects is taken into consideration while in "making pink tea" sensor noise is considered for sake of experiment. Table 4 shows the object frequencies along with CAS objects for optional objects and sensor noise in both activities. While Table 5 illustrates the frequencies of unidentified patterns

Table 4. Occurrence Frequencies of sensor noise and optional sensors

Action Sequence	Object Occurrence	Frequency of objects	Average Frequency	Object learned along with property
Making brown tea optional sensor (16)				
Water	10	.67	.37	Water
Strainer	4	.27		
Has sugar	1	.07		
Making pink tea sensor noise (18)				
Cardamom	12	.92	.5	Cardamom
Spoon	1	.08		

Resultant perceptible activity model for "making brown tea" and making pink tea" is given as follows:
Making brown tea: hasadding(tea), hasadding(milk), hashotappliance(stove), has adding(water)
Making pink tea: hasadding(milk), hashotappliance(stove), hasadding(pink), hasadding(cardamom).

Table 5. Occurrence Frequencies of Unidentified patterns

Action Sequence	Occurrence Frequency	Frequency of pattern	Average Frequency	Candid-ate Action Sequence
Unlabeled-pattern1	1	.017	.167	Unlabeled - pattern4, Unlabeled – pattern5
Unlabeled - pattern2	3	.051		
Unlabeled - pattern3	5	.085		
Unlabeled - pattern4	20	.34		
Unlabeled - pattern5	2	.42		
Unlabeled - pattern6	5	.085		

5. Conclusion

A dynamic, adaptive and self-evolution framework for activity recognition has been proposed in this paper. The AR models, mechanisms and algorithms for activity evolution and activity learning are discussed in detail. From experiments performed, it is evident that the proposed ontology evolution model is fundamental to support reusable, adaptive and personalized AR model. Moreover, it has the capacity to learn new activities by determining the contexts from existing activities. Another visible contribution is the development of dataset and scenarios to test various scenarios using the proposed framework. Hence, it may be asserted that a promising solution for the cold start problem is implemented.

The work presented is extendible for large-scale experiments and ontology consistency evaluation. One promising future direction is to work on learning the actions for an activity based on temporal dependencies among the actions. The proposed model is presented for simple activities that can be extendible for complex activities like preparing breakfast. The proposed model is unable to handle the patterns having missing sensor values, handling the missing sensors values in the adaptable model is one appealing future directions.

References

[1] Chen, L., Hoey, J., Nugent, C. D., Cook, D. J., & Yu, Z.. 'Sensor-based activity recognition'. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2012, 42(6), pp 790-808.

[2] Yang, P., Stankevicius, D., Marozas, V., Deng, Z., Liu, E., Lukosevicius, A., Min, G. 'Lifelogging data validation model for internet of things enabled personalized healthcare'. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2008, 48(1), pp 50-64.

[3] Ni, Q., García Hernando, A. B., & de la Cruz, I. P. 'The elderly's independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development'. Sensors, 2015, 15(5), pp 11312-11362.

[4] Meditskos, G., Kontopoulos, E., & Kompatsiaris, I. 'Knowledge-driven activity recognition and segmentation using context connections'. In International Semantic Web Conference Springer, Cham., 2014

[5] Riboni, D., & Bettini, C.. 'COSAR: hybrid reasoning for context-aware activity recognition'. Personal and Ubiquitous Computing, 2011, 15(3), pp 271-289.

[6] Chen, L., Nugent, C. D., & Wang, H. 'A knowledge-driven approach to activity recognition in smart homes'. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(6), pp 961-974.

[7] Safyan, M., Qayyum, Z. U., Sarwar, S., García-Castro, R., & Ahmed, M. 'Ontology-driven semantic unified modelling for concurrent activity recognition (OSCAR)'. Multimedia Tools and Applications, 2018, pp 1-32.

[8] Khattak, A. M., Batool, R., Pervez, Z., Khan, A. M., & Lee, S.. 'Ontology Evolution and Challenges.' J. Inf. Sci. Eng., 2013, 29(5),pp 851-871.

[9] G. Flouris and D. Plexousakis, ‘Handling ontology change: Survey and proposal for a future research direction’, Technical Report FORTH-ICS/TR-362, 2005.

[10] Stojanovic, L., A. Madche, B. Motik, and N. Stojanovic, ‘User driven ontology evolution management’, in Proceedings of European Conference on Knowledge Engineering and Management, 2002, pp. 285-300..

[11] Stojanovic, L.. Methods and tools for ontology evolution Msc Thesis, 2004

[12] Stojanovic, L., Maedche, A., Motik, B., & Stojanovic, N. ‘User-driven ontology evolution management, In International Conference on Knowledge Engineering and Knowledge Management Springer, Berlin, Heidelber, 2002, pp. 285-300.

[13] Noy, N. F., & Klein, M.. ‘Ontology evolution: Not the same as schema evolution’, Knowledge and information systems, 2002, 6(4), 428-440.

[14] Stojanovic, L., Stojanovic, N., & Handschuh, S. ‘Evolution of the metadata in the ontology-based knowledge management systems,. In GI Proceedings of the 1st German Workshop on Experience Management GWEM ,2002, LNAI(10), pp. 65-77.

[15] Stojanovic, L, Stojanovic, N., Gonzalez, J. & Studer, R. . ‘ OntoManager— A System for the usage-based ontology management’. In On The Move to Meaningful Internet Systems: CoopIS, DOA, and ODBASE , Berlin, Germany: Springer. 10.1007/b94348. 2003, LNCS(2888), pp 858–875.

[16] Völker, J., Hitzler, P. & Cimiano, P. . ‘Acquisition of OWL DL axioms from lexical resources’. In E. Franconi, M. Kifer, & W. May (Eds.), The Semantic Web: Research and Applications, Proceedings of the 6th International Semantic Web Conference, Berlin, Germany: Springer. doi: 10.1007/978-3-540-72667-8 ISWC 2007 LNCS(4519), pp. 670-685.

[17] Khattak A. M., Latif K., Lee S. Y., and Lee Y. K., ‘Ontology evolution: A survey and future challenges’, in Proceedings of the 2nd International Conference on u-and e-Service, Science and Technology, 2009, pp. 68-75.

[18] Chen L., Nugent C.D. and Wang H., ‘A Knowledge-Driven Approach to Activity Recognition in Smart Homes’, IEEE Transactions on Knowledge and Data Engineering. 2011

[19] Azkune, Gorka, et al. ‘Extending knowledge-driven activity models through data-driven learning techniques’, *Expert Systems with Applications* 2015, 42(6) pp 3115-3128.

[20] Okeyo, G., Chen, L., Wang, H., Sterritt, R. ‘A Knowledge-Driven Approach to Composite Activity Recognition in Smart Environments’. Ubiquitous Computing and Ambient Intelligence. 2012, Vol. 7656 of the series Lecture Notes in Computer Science. Springer Berlin Heidelberg. pp. 322-329.

[21] Javed, M., Abgaz, Y. M., & Pahl, C., ‘Graph-based discovery of ontology change patterns’, in: Proceedings of the Joint Workshop on Knowledge Evolution and Ontology Dynamics (EvoDyn) at ISWC, Bonn, Germany, 2011.

[22] Wang, S., Wang, W., Zhuang, Y., & Fei, X.. ‘An ontology evolution method based on folksonomy’, Journal of Applied Research and Technology, 2015, 13(2), pp 177-187 .

[23] http://vanderwal.net/folksonomy.html, accessed: November 2018.

[24] Sellami, Z., Camps, V., & Aussenac-Gilles, N. ‘DYNAMO-MAS: a multi-agent system for ontology evolution from text’, Journal on Data Semantics, 2013, 2(2-3), 145-161.

[25] Haase, P., & Stojanovic, L. ‘Consistent evolution of OWL ontologies’. In European Semantic Web Conference, 2005m pp. 182-197.

[26] Mahfoudh, M., Forestier, G., Thiry, L., & Hassenforder, M.. ‘Algebraic graph transformations for formalizing ontology changes and evolving ontologies’, Knowledge-Based Systems (Elsevier), 2015, pp 212-226.

[27] Stojanovic L., Madche A., Motik B., and Stojanovic N., “User driven ontology evolution management,” in Proceedings of European Conference on Knowledge Engineering and Management, 2002, pp. 285-300.

[28] Noy, N. F., & Klein, M. ‘Ontology evolution: Not the same as schema evolution’. Knowledge and Information Systems, 2004, 6(4), pp 428-440.

[29] Ermel, C., Rudolf, M., & Taentzer, G, ‘The AGG approach: Language and environment’, in: Handbook of Graph Grammars and Computing by Graph Transformation, World Scientific Publishing Co. Inc. 199, pp 551-603

[30] Kondylakis, H., & Papadakis, N.. EvoRDF: evolving the exploration of ontology evolution. The Knowledge Engineering Review, 33., 2018.