This is the published version of a paper presented at 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015).

Citation for the original published paper:


http://dx.doi.org/10.1016/j.procs.2015.05.007

N.B. When citing this work, cite the original published paper.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:umu:diva-101444
The 6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015)

A Probabilistic Non-Monotonic Activity Qualifier
Juan Carlos Nievesa,∗, Saeed Partoniaa, Esteban Guerreroa, Helena Lindgrena

aDepartment of Computing Science, Umeå University, SE-901 87, Umeå, Sweden

Abstract
The International Classification of Functioning, Disability and Health (ICF) defines Functioning and Disability as the results of the interaction between the health conditions of a person and his/her environment. It considers a set of components and qualifiers to evaluate activity and participation. In this paper, we interpret a performance quantifier under a human activity recognition process. To this end, we introduce a novel definition of an activity which is based on ICF guidelines. This definition gives place to a probabilistic non-monotonic activity qualifier. In order to recognize an activity according to our novel activity’s definition, we explore non-monotonic reasoning techniques to capture domain knowledge in terms of action specification languages. By considering an action specification language, called CTAM, and Answer Set Programming, we propose and develop a system called ActRec system which takes background information into consideration and recognize activities according to our suggested definition. Moreover, we show that by aggregating our probabilistic non-monotonic activity qualifier, we are able of detecting complex activities, e.g., long-term activities. We illustrate our approach in the context of an ambient assisted living environment called As-A-Pal.

© 2015 The Authors. Published by Elsevier B.V. Peer-review under responsibility of the Conference Program Chairs
Keywords: Reasoning technology for ambient sensing; Activity Recognition; Non-monotonic Reasoning; Smart-Environments.

1. Introduction
The International Classification of Functioning, Disability and Health (ICF) belongs to the family of international classifications, categorizing functioning and disability associated with health conditions, developed by the World Health Organization (WHO)33. Functioning is treated in ICF as a generic term for body functions, body structures, activities and participation. By contrast, Disability, as defined in ICF, is an umbrella term for impairments, activity limitations and participation restrictions. These two aspects of the interaction between an individual and that individual’s contextual factors (environmental and personal factors), require different quantification and qualification tools for measuring the notions of “health” and “disability”.

Qualifiers record the presence and severity of a problem in functioning at the body, person and societal levels. The ICF defines two main qualifiers in relation to activity and participation: Performance and Capacity14.

∗ Corresponding author.
E-mail address: jcnieves@cs.umu.se; saeed.partonia@gmail.com; esteban@cs.umu.se; helena@cs.umu.se
The **Performance** qualifier describes what an individual does in his or her current or natural environment. This would be what an individual really does, e.g., what purposeful activities an older adult does at home, or in their community.

The **Capacity** qualifier describes an individual’s ability to execute a task or an action in a controlled environment. This qualifier tells us what an individual is capable of doing in a structured clinical situation, e.g., measuring the ability of an older adult to stand in a feet tandem position when executing the Short Physical Performance Battery test.

Having access to both Performance and Capacity data enables clinicians and practitioners to determine the gap between capacity and performance. If capacity is less than performance, then the person’s current environment has enabled him or her to perform better than what data about capacity would predict: the environment has facilitated performance. On the other hand, if capacity is greater than performance, then some aspect of the environment is a barrier to performance.

In the literature, quantification measures and evaluation of physical activity based on information obtained by sensors are accepted as valid evidence of a physical phenomena. In these technological approaches, the evidence is linked to the confidence of the captured observations.

As-A-Pal is a smart environment where the acronym As-A-Pal refers to **Agent-Supported Assessment for Adaptive and Personalized Ambient Assisted Living**. As-A-Pal also refers to like a friend, an artificial companion that knows the immediate needs of the human actor, her preferences, priorities and abilities, so that adaptive and personalized services tailored to the current context can be provided. Kitchen As-A-Pal builds on and complements our earlier work on smart homes, adaptive systems for older adults and knowledge-based support systems for the medical domain. Kitchen As-A-Pal serves as a living laboratory environment for designing and developing a range of different knowledge-based applications intended to be deployed as part of a holistic approach to ambient assisted living. The objects in Kitchen As-A-Pal are embedded with sensors, actuators, physical interfaces and ambient displays. Kitchen As-A-Pal is augmented with sensors and passively tagged objects. The physical and ambient interfaces provide access to information and services. Some of the mock-up services include recipe provider, medication manager, shopping assistant and self-improvement games.

Understanding and inferring human activities and the context in which they take place is a research challenge. Human activities take place at multiple levels simultaneously: from the level of body and body-part movements, to the interaction with situated objects, to performing goal-directed actions to performing complex activities with clearly defined motives. While there are several approaches to activity recognition in a smart home that are of importance, such approaches are not based on a formal definition of an activity and their associated context thereby answering “what activity was performed?” but are insufficient in answering “how the activity was performed?” and “how can we be sure that the activity was performed?” A smart home worthy of its name in providing activity support requires additional knowledge about an activity and better qualitatively describe the activity performed.

Against this background, we introduce a **novel definition of an activity** which follows the guidelines of ICF. This definition is introduced in terms of actions and sets of fluents which are called states. Considering this definition, we introduce the so called **Δ-performance qualifier**. In order to capture context information, we model an environment in terms an action specification language called **CTAID**. To recognize activities according to our suggested definition, we propose and develop a system called **ActRec** system which has been plugged in Kitchen As-A-Pal. The ActRec system infers the Δ-performance qualifier of a given activity. This degree shows evidence about the performance of a given activity; moreover, according to ICF, this degree can be regarded as a performance quantifier. We will show that by considering long terms evaluations of this performance quantifier, we are able to observe an emerging behavior of an observed Persona.

The rest of the paper is divided as follows: Section 2 introduces the theoretical contributions of this paper, namely new definitions of an activity and the definition of the Δ-performance qualifier. Section 3 presents a short description of ActRec system. Section 4 illustrates the work flow of ActRec system. Section 5 presents an application of the Δ-performance qualifier. In the last section, we outline our conclusions.
2. Theoretical Framework

In this section, we introduce the first two contributions of this paper: new definitions of an activity and a novel performance quantifier of an activity. The last one is based on both the former one and a non-monotonic reasoning process for activity recognition.

2.1. Activity Definitions

We start presenting some basic concept on action languages. Our particular implementation is based on the action specification language $C_{TAID}$\(^6\). Due to lack of space, we omit to present $C_{TAID}$’s both syntax and semantics. We will only present the relevant definitions for presenting our results.

The alphabet of $C_{TAID}$ consists of two nonempty disjoint sets of symbols: a set of action names $A$ and a set of fluent names $F$. As in the original definition of $C_{TAID}$, we deal with propositional fluents that are either true or false. A fluent literal is a fluent symbol $f$ (called positive fluent literal) or the negation of fluent symbol $\neg f$ (called negative fluent literal). A state $s$ is a collection of fluents. We say a fluent $f$ holds in a state $s$ if $f \in s$. We say a fluent literal $\neg f$ holds in $s$ if $f \not\in s$.

In order to define our definition of an activity, the definition of a trajectory is introduced.

**Definition 1.** Let $D(A,F)$ be a domain description according to $C_{TAID}$\(^6\). A trajectory of the form $(s_0,A_1,s_1,\ldots,A_n,s_n)$ of a domain description $D(A,F)$ is a sequence of states $s_i$ and sets of actions $A_i \subseteq A$ for $(0 \leq i < n)$, and expresses the possible evolution of the system with respect to $D(A,F)$.

A trajectory ensures a set of conditions related to the semantics of $C_{TAID}$\(^6\). Considering the concept of a trajectory, we will introduce the first definition of an activity which is called a basic activity.

**Definition 2.** Let $S$ be a set of fluents. A basic activity is of the form $(S_I,S_F)$ such that $S_I \subseteq S$, $S_F \subseteq S$ and $S_I \neq S_F$. $S_I$ and $S_F$ are called initial and final states, respectively.

An activity may involve one or several actions. These actions may follow a certain order or not. For example, in a "Breakfast scenario", the order of performing actions, e.g., eating cereal, drinking, etc., may not be so important. However, for some activities, it is necessary that the actions must be performed in a certain order. For instance, for an activity of "Ordering pizza", one usually picks up the phone, dials the restaurant’s number, orders pizza, pays the order and finally receives pizza. Apparently, it is not possible to receive a pizza before ordering it. Hence, it is necessary to differentiate between two activities by defining A-activity and O-activity for non-ordered and ordered actions, respectively.

**Definition 3.** Let $A$ be a set of actions and $S$ be a set of fluents. An A-activity is of the form $(S_I,A,S_F)$ such that:

- $(S_I,S_F)$ is a basic activity.
- $\exists$ a trajectory $T = (S_0,A_1,S_1,\ldots,A_n,S_n)$, such that $S_0 = S_I$, $S_F = S_n$ and $A_i \subseteq A$ $(1 \leq i \leq n)$.

**Definition 4.** Let $A$ be a set of actions. An O-activity is of the form $<S_I,A,L_A,S_F>$ such that:

- $<S_I,S_F>$ is a basic activity.
- $\exists$ a trajectory $T = (S_0,A_1,S_1,\ldots,A_n,S_n)$ such that $S_0 = S_I$, $S_F = S_n$ and $A_i \subseteq A$ $(1 \leq i \leq n)$.
- $L_A = [A^1,A^2,\ldots,A^n]$ is a list. We assume that a list preserves an order between their elements.
- Let $T_A$ be a function which returns the set of sets of actions which appear in $T$, $L_A$ be a function which returns the set of sets of actions which appear in $L_A$ and $A_i,A_j \in T_A$ $(1 \leq i,j \leq n)$, $A_p,A_q \in L_A$ $(1 \leq p,q \leq m)$. If $A_i = A^p$ and $A_j = A^q$ such that $p < q$ then $i < j$.

From hereon whenever we refer to an activity, it can be either a basic activity, A-activity or O-activity. Let us observe that these three notions of an activity have relevance and applicability in the evaluation of physical activities in terms of the ICF. For instance, O-activity is related with a fine granularity in the evaluation of the essentials of
walking\textsuperscript{1} as is defined by ICF, including the combination of several ordered body functions (and actions associated with their body structures)\textsuperscript{14,13}, such as rising, mobility of structural support bones, center of mass displacement, etc.

2.2. ActivityQualifier

The semantics of $C_{TAID}$ is usually been implemented by using Answer Set Programming\textsuperscript{3}. In this setting, given an action theory according to $C_{TAID}$, there is a set of answer sets which basically are trajectories, see Definition 1. Therefore we will obtain $n$ trajectories from action theory rather than one to recognize an activity from an action theory. Hence we need to treat the set of trajectories in a way that allows us to draw conclusions about activities. In this setting, we introduce an activity qualifier.

Definition 5. Let $AT = (D, O)$ be an action theory according to $C_{TAID}$\textsuperscript{6} and $\mathcal{A}$ be a set of activities. The $\Delta$-performance qualifier of $A \in \mathcal{A}$ with respect to $AT$ is:

$$\Delta(A) = \frac{\sum_{i=1}^{n} V(T_i, A)}{n}$$

$n$ is the number of trajectories which $AT$ has. $V(T, A)$ returns 1 if the activity $A$ is recognized within trajectory $T$, and 0 otherwise.

We would like to mention here that the $\Delta$-performance qualifier of a given activity is probabilistic non-monotonic, i.e., the $\Delta$-performance qualifier has the ability of changing its probability as a result of new information. Moreover, let observe that Definition 5 provides us an activity qualifier that, in terms of the ICF, will estimate what an individual “really does” regarding to both a set of actions and observations of an particular activity. It is important to establish a difference between, what a person is capable to accomplish through a Capacity qualifier, e.g., by the number of hypothetical actions that might perform, and what an individual really does measured by analysis of the set of captured trajectories in an activity.

3. Description of ActRec System

In order to recognize activities according to the definitions introduced in Section 2 and inferring $\Delta$-performance qualifiers, the ActRec system has been implemented\textsuperscript{2}. In this section, we introduce a short description of the ActRec system. Basically, the ActRec system consists of three main parts: 1) a mapper, 2) a solver and 3) an activity recognizer, see Figure 1. The work flow is as follows:

- First, the mapper converts the user input $C_{TAID}$ program into a logic program solvable by a solver.
- Then, the solver interacts with an ASP-solver and passes a logic program to the solver.
- The output of the solver then is passed to the recognizer for activity recognition. The process of activity recognition will be refereed as ActRec algorithm in the later sections. The algorithm is presented in detail in \cite{15}.

4. Experiments

The ActRec system has been evaluated in Kitchen As-A-Pal. Hence, this section presents a discussion about this evaluation. We performed experiments with some scenarios on the Kitchen As-A-Pal. Basically, we present two scenarios; making pasta and making coffee. We attempt to specify in a declarative way the environment with fluents and actions and show the work flow of ActRec system.

\textsuperscript{1} Essentials of walking has a ICF code d450 relating multiple body functions and body structures, as well as environment factors which are coded in the ICF classification: http://apps.who.int/classifications/icfbrowser/

\textsuperscript{2} A full description of the ActRec system can be found in \cite{15}.
4.1. Formulating the environment

Table 1 shows fluents and actions used in the experimented scenarios. Needless to say, these fluents and actions are up to the users and one may add or remove any arbitrary number of those to formulate the environment for both scenarios. We can now show how these two scenarios can be represented as a domain description in CTAID. For sake of simplicity, we represent the steps in details for the making coffee scenario. Similar approach could be applied to the scenario of making pasta.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Fluents</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Making Coffee</td>
<td>coffe_m_is_on, coffe_m_has_powder, coffe_m_has_old_coffe_powder, coffe_m_has_water, coffe_m_is_jug_removed</td>
<td>coffe_m_activate, coffe_m_add_coffe_powder, coffe_m_remove_coffe_powder, coffe_m_add_water, coffe_m_add_jug, coffe_m_remove_jug</td>
</tr>
<tr>
<td>Making Pasta</td>
<td>heat_on, vessel_on_heat, vessel_water_full, vessel_pasta_full, pasta_cooked, vessel_sauce_full, vessel_sauce_mix, pasta_sauce_drained, pasta_water_drained, pasta_water_boiled, pasta_cooked_drain, pasta_sauce_add</td>
<td>vessel_water_boil, pasta_boil, pasta_cooked_drain, pasta_sauce_add</td>
</tr>
</tbody>
</table>

Table 1. Fluents and actions used in each scenario.

Our knowledge about making coffee gives rise to the following dynamic causal rules:

\[
\text{coffe}_m\text{activate} < \text{causes} > \text{coffe}_m\text{is}_{-}\text{on} < \text{if} > \neg(\text{coffe}_m\text{is}_{-}\text{on}),
\]

\[
\text{coffe}_m\text{add_water} < \text{causes} > \text{coffe}_m\text{has}_{-}\text{water} < \text{if} > \neg(\text{coffe}_m\text{has}_{-}\text{water}),
\]

\[
\text{coffe}_m\text{add_coffe}_\text{powder} < \text{causes} > \text{coffe}_m\text{has}_{-}\text{powder} < \text{if} > \neg(\text{coffe}_m\text{has}_{-}\text{powder}).
\]

Additionally, some static causal rules could be added as follows:

\[
\text{coffe}_m\text{has}_{-}\text{old}_\text{coffe}_\text{powder} < \text{if} > \text{coffe}_m\text{has}_{-}\text{water}, \text{coffe}_m\text{has}_{-}\text{has}_{-}\text{water}, \text{coffe}_m\text{is}_{-}\text{on}.
\]

\[
\neg(\text{coffe}_m\text{has}_{-}\text{old}_\text{coffe}_\text{powder}), < \text{if} > \text{coffe}_m\text{has}_{-}\text{powder}.
\]

\[
\neg(\text{coffe}_m\text{has}_{-}\text{powder}) < \text{if} > \text{coffe}_m\text{has}_{-}\text{old}_\text{coffe}_\text{powder}.
\]

The latter two rules ensure that \text{coffe}_m\text{has}_{-}\text{powder} and \text{coffe}_m\text{has}_{-}\text{old}_\text{coffe}_\text{powder} cannot be true at the same time. Now we can define a set of observations for our scenario. The initial state can be defined by the following fluent observations as follows:

\[
\neg(\text{coffe}_m\text{is}_{-}\text{on}) < \text{at} > 0. \neg(\text{coffe}_m\text{has}_{-}\text{water}) < \text{at} > 0. \neg(\text{coffe}_m\text{has}_{-}\text{powder}) < \text{at} > 0.
\]

For a time bound of \(t_{\text{max}} = 5\), we obtain 4 possible answer sets. Again, it worth mentioning that by adding and/or removing some fluents, actions and rules to the scenario we may obtain different number of answer sets.
4.2. Applying ActRec Algorithm

Now that we obtained answer sets from the ASP solver, we can define and infer activities. As mentioned earlier, answer sets are mapped into trajectories before feeding into the ActRec algorithm for recognition. In order to infer about an activity, we first need to define what we expect from an activity. In other words, initial and final states of the activity needs to be determined. For the scenario of making coffee, we defined the initial state as follows: \( \text{initialState} = [\text{coffee}\_m\_has\_powder,\text{coffee}\_m\_has\_water] \) and final state as \( \text{finalState} = [\neg(\text{coffee}\_m\_is\_jug\_removed),\text{coffee}\_m\_has\_old\_coffee\_powder] \).

<table>
<thead>
<tr>
<th>Scenario</th>
<th>#Answer Sets</th>
<th># Recognized</th>
<th>#Ongoing</th>
<th>Failed</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Making Pasta</td>
<td>618</td>
<td>16</td>
<td>24</td>
<td>578</td>
<td>0.025</td>
</tr>
<tr>
<td>Making Coffee</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 2. Results of running ActRec algorithm on two scenarios.

As we defined the activity, we can run our ActRec algorithm to see whether we can recognize the defined activity in any on the obtained trajectories (answer sets). For the scenario of making coffee, we can see that one activity is recognized given initial and final states. In addition, three trajectories include ongoing activities. Table 2 shows the results of running the algorithm for both scenarios in more details. In the next section we discuss how to analyze the results in details.

5. Aggregating Δ-performance qualifiers

In the previous section, we illustrated how we can run the ActRec system to recognize an activity. This approach can be used for instance in occupational therapy scenarios, where a physical activity performance is necessary to be assessed. Different approaches for evaluating and assess occupational performance have been developed. Dynamic Performance Analysis (DPA) is an approach to occupational analysis that focuses on the client’s actual performance in order to identify where performance breaks down and test out solutions. An initial part of DPA decision tree is the establishing of a task hierarchy (DPA uses the concept of “task” for what we consider an “activity”), where a set of actions are defined by the expert therapist as is shown in Figure 2.

![Fig. 2. Occupational performance hierarchy: bathing. Reprinted from](image)

By running ActRec and calculating the Δ-performance qualifier of several activities such as: cooking, walking into kitchen, bathing, etc., we can provide a probabilistic measurement about the performance patterns which refers to habits, routines, roles, and rituals used in the process of engaging in occupations or activities. Figure 3 depicts the idea of recognizing and evaluating different activities over a period of time using simulated data entries in a health-care scenario.

We can see that certain activities could comprise groups of them as they are close together, e.g., cooking, watching TV and bathing in Figure 3. In a preliminary laboratory experiment, we use ActRec in a home-care scenario considering the guidance of an occupational therapists expert defining the set of actions. Moreover, in the same direction we use a data mining approaches for detecting groups of activities such as clustering. Clustering is a form of unsupervised learning that takes unlabeled data and places objects of the data in groups in such a way that objects of the same group
Clustering is used when the data we are dealing with has no label. Clusters of different actions/activities could represent a general classification of actions/activities or a complex activity, such as: indoor, outdoor, preparing and taking breakfast, etc. For instance, the activities cooking, watching TV and bathing in Figure 3, represent a set of activities resembling a category such as indoor, where a breakdown-point is detected by the expert representing the moment when the performance of an activity declines, e.g., when the performance of outdoor activities (walking and gardening) descends or ceases in the simulated scenario in Figure 3.

6. Conclusions

We attempt to build a complex activity recognition technique that not only rely on the sensory data\textsuperscript{1} but also involve domain knowledge as well. We approached the problem of recognizing human activity through non-monotonic reasoning, logic programming and using an action language namely \textit{CTAID} for modeling the environment. \textit{CTAID}, provides us different features for describing the environment in a declarative action specification. The problem instance is described in the action language \textit{CTAID} and then converted into an ASP program by mapping. In this setting, we introduce a novel definition of an activity which follows current approaches used in health-care context, particularly in Occupational Therapy and Physiotherapy. From the technical point of view, our definition of activity is based on the trajectories of an action theory. To this end, we introduce a complex activity recognition technique that not only relies on data but also involve domain knowledge. A recent work\textsuperscript{17} proposes the use of “context attributes” to capture the background information, being a context attribute a piece of data that holds information about the background. While the accuracy of the proposed model is acceptable, it does not provide an explanation to the solution rather than a probability. We propose and develop \textit{ActRec}, a system based on non-monotonic reasoning that not only provides a probability of a particular recognized activity, but also, an explanation of how the activity is performed.

Considering our definition of an activity, we introduce the so called \textit{Δ-performance qualifier}. The \textit{Δ}-performance qualifier, after performing the activity recognition task, determines the confidence level of occurrence of a certain activity. \textit{Δ}-performance qualifier can be used along with other ICF qualifiers to describe frequency, environmental factors
(through fluent observations) as well as confidence of the occurrence of an activity in clinical or ambulant/health-care contexts. Part of our future work is to define other ICF qualifiers in terms of action theories.

Our suggested frame has been implemented through the ActRec system which takes as an input an action theory and tells us whether an activity is performed or not. We attempted to answer the questions of the form: Which activity was performed? How was the activity performed? How can we be sure that the activity was performed? By introducing the specification of the desired activity to the system, we can run the recognition task to see whether a certain activity is performed or not. The trajectory of the activity gives us an explanation of the activity, in case the activity is recognized in a trajectory.

We showed how to interpret the data from ActRec system in order to infer complex activities by aggregating the $\Delta$-performance qualifier of several activities.

Acknowledgements

This research is partly funded by VINNOVA (Sweden’s innovation agency) and the Swedish Brain Power.

References