Event Recognition for Unobtrusive Assisted Living

Nikos Katzouris[1 2], Alexander Artikis[1], and Georgios Paliouras[1]

[1] Institute of Informatics & Telecommunications
National Center for Scientific Research 'Demokritos'
[2] Department of Informatics & Telecommunications
National Kapodistrian University of Athens
{nkatz, a.artikis, paliourg}@iit.demokritos.gr

Abstract. Developing intelligent systems towards automated clinical monitoring and assistance for the elderly is attracting growing attention. USEFIL is an FP7 project aiming to provide health-care assistance in a smart-home setting. We present the data fusion component of USEFIL which is based on a complex event recognition methodology. In particular, we present our knowledge-driven approach to the detection of Activities of Daily Living (ADL) and functional ability, based on a probabilistic version of the Event Calculus. To investigate the feasibility of our approach, we present an empirical evaluation on synthetic data.

1 Introduction

Developing intelligent systems towards automated clinical monitoring and assistance for the elderly is attracting significant attention, due to the increase in the ageing population. Age-related demographic trends in most western countries and increasing health-care costs, indicate a need for robust telehealth solutions which shall prolong seniors' independent living. USEFIL[3] is an FP7 project aiming to provide health-care assistance to seniors who live alone. The USEFIL system relies on a three-layer architecture. The bottom layer consists of an in-house data acquisition platform comprising off-the-self, low cost sensors. A data fusion component in the intermediate layer of the system is responsible for combining data from multiple sources via spatio-temporal reasoning. Fused data are consumed by a Decision Support System in the top-level layer, which provides input to a number of user-friendly, interactive health-care/monitoring apps, designed both for the USEFIL resident and the medical/caregiving stuff. The main objective of the Decision Support System is the identification of early deterioration signs for a plethora of medical cases and behavioural disturbances.

We present our approach to USEFIL's data fusion component. Its role is to interpret the parameters of raw sensor data into a semantic representation of human behaviour and functional ability, while counterbalancing confidence in the sensor measurements by fusing data from multiple sources. Its tasks range from contextualization of sensor measurements to detecting sleep and Activities of Daily Living (ADL), and characterising functional ability.

[3] http://www.usefil.eu/

USEFIL's sensor network is able to provide a wide range of measurements and indications, related both to user's activities and to the environment. The sensors include a depth camera (kinect), a light wrist watch (Wrist Wearable Unit – WWU), a hidden camera installed in a smart-PC, a number of microphones around the house, and so on. A key requirement in most assisted living applications is unobtrusiveness: Monitoring should not intervene with daily activities, so that the user feels comfortable and sensor data are collected naturally and in an unbiased fashion. Thus, rich as may it be, the sensor network does not cover the entire house, nor can it provide indications for all situations of interest. Moreover sensor data may often be corrupted by noise. Thus, in order to increase confidence in the representation of the user status it is necessary to exploit the existing monitoring equipment as much as possible, by combining different sensor resources, while addressing uncertainty. To this end, we employ a Complex Event Recognition methodology, which allows to combine heterogeneous data sources by means of event hierarchies. Our approach is based on a probabilistic version of the Event Calculus [14], which allows for handling noise and modelling uncertainty.

The rest of this paper is structured as follows. Section 2 contains basic background on the Event Calculus and ProbLog, the probabilistic logic programming language on which our implementation relies. Section 3 describes the construction of event patterns related to the detection of ADL and functional ability. In Section 4 we present an empirical evaluation of our approach, and in Sections 5 and 6 we summarise and put our work in context.

2 Event Recognition & Probabilistic Event Calculus

Complex Event Recognition [10] refers to the automatic detection of event occurrences within a system. From a sequence of *low-level events* (LLEs) – such as sensor data, an event recognition system recognizes complex, or *high-level events* (HLEs) of interest, that is, events that satisfy some pattern. Event recognition systems with a logic-based representation of event definitions, are attracting significant attention in the event processing community for a number of reasons, including the expressiveness and understandability of the formalized knowledge and their declarative, formal semantics.

In this paper we follow the standard logic programming notation, so variables start with an upper-case letter, while predicates, function symbols and constants start with a lower-case letter. The Event Calculus is a many-sorted, first-order predicate calculus for representing and reasoning about events and their effects. Its ontology comprises *time points* (integer or real numbers), *fluents* (properties that have values in time) and *events* (occurrences that may affect the values of fluents). In this work, we assume that the time model is linear and integer-valued. For a fluent F, the expression $F = V$ means that F has the value V, and intuitively $F = V$ holds at a particular time point if it has been *initiated* at a previous time point and has not been *terminated* since. The *domain-independent* axioms of the Event Calculus formalize the commonsense law of *inertia*, according to which fluents persist in time unless they are affected by events. The axioms of the dialect employed in this work are as follows:

$$\begin{aligned}&\mathsf{holdsAt}(F = V, T) \leftarrow\\&\quad \mathsf{initially}(F),\\&\quad \mathsf{not\ broken}(F = V, 0, T).\end{aligned} \tag{1}$$

$$\begin{aligned}&\mathsf{holdsAt}(F = V, T) \leftarrow\\&\quad \mathsf{initiatedAt}(F = V, T_s),\\&\quad T_s < T,\\&\quad \mathsf{not\ broken}(F = V, T_s, T).\end{aligned} \tag{2}$$

$$\begin{aligned}&\mathsf{broken}(F = V, Ts, T) \leftarrow\\&\quad \mathsf{terminatedAt}(F = V, T_f),\\&\quad T_s < T_f < T.\end{aligned} \tag{3}$$

not represents "negation by failure", which provides a form of default persistence – inertia of fluents. According to Axiom (1), $F = V$ holds at time T if $F = V$ held initially and has not been broken since. According to Axiom (2), $F = V$ holds at time T if the fluent F has been initiated to value V at an earlier time T_s, and has not been broken since. Axiom (3) dictates that a period of time for which $F = V$ holds is broken at T_f if $F = V$ is terminated at T_f.

In this work we assume that LLEs and HLEs, as defined in an event recognition context, correspond respectively to Event Calculus events and fluents. Given the domain-independent axioms of the Event Calculus, the construction of HLE patterns consists in defining *domain-specific* rules which describe initiation and termination conditions.

Various types of uncertainty exist in event-related domains, such as erroneous or missing LLEs [1]. To address uncertainty, in [18] the Event Calculus has been ported in the probabilistic logic programming language ProbLog [13]. ProbLog is a probabilistic version of Prolog, where standard Prolog facts and rules may be annotated with probabilities. Probabilistic facts are expressions of the form $p_i :: f_i$, where f_i is a Prolog fact and p_i is a real number in the interval $[0, 1]$. Probabilistic facts in ProbLog represent random variables with an independence assumption, thus a rule defined as a conjunction of a set of probabilistic facts, has a probability that is equal to the product of the probabilities of these facts. When a predicate appears in the head of more than one rule, then its probability is computed by calculating the probability of the implicit disjunction created by the multiple rules. ProbLog facts with no probability are implicitly given a probability of 1. In addition to probabilistic facts, ProbLog supports probabilistic rules as well, that is, expressions of the form $p_0 :: h \leftarrow f_1, \ldots, f_n$. Intuitively, such a rule means that if $\wedge_{j=1}^{n} f_j$ is true and each f_j has probability p_j, then h is true with probability $p_h = \prod_{j=0}^{n} p_j$. For a non-ground probabilistic expression $p :: e$ (rule or fact), the probability p applies to all possible groundings of e.

ProbLog supports several forms of probabilistic inference, such as computing the success probability of a query (the overall probability of a query being true) or finding the most likely explanation of a query (the proof with the highest probability). Given a program P and a query q, computing the success probability of q can be achieved by summing the probabilities of all subprograms L that entail q. This is a hard task even for small problems, since it involves a number of summands which is exponential in the size of the Herbrand base of the initial program. Thus in practice, ProbLog uses Binary Decision Diagrams (BDDs) [4] and techniques from dynamic programming to efficiently address computations through different proofs of a query, allowing the implementation to scale to queries containing thousands of different proofs [13].

3 Event Definitions for Activities of Daily Living

In this section we present our approach to detecting activities of daily living (ADL) and characterizing functional ability. ADL typically refers to the fundamental self-care activities, such as getting out of bed, walking around etc. ADL detection is of particular importance in assisted living applications, since the capacity to perform such activities has been confirmed in numerous studies to have broad implications for functioning, reflecting a person's ability to live independently. Given the fact that disability or functional impairment is usually closely related to a person's inability to perform these and other basic tasks without assistance, the requirements of the ADL use-case in USEFIL involve, in addition to ADL detection, estimation of the ADL's score in the *Barthel Index* [8].

The Barthel Index, commonly acknowledged as the "golden standard" for functional ability, consists of 10 items that measure a person's daily functioning. The items include feeding, moving from wheelchair to bed and return, grooming, transferring to and from a toilet, bathing, walking on level surface, going up and down stairs, dressing, and continence of bowels and bladder.

In USEFIL, obtaining a Barthel score for some of these ADL is subject to restrictions/limitations, mainly due to monitoring insufficiency, related to the unobtrusiveness requirement[4]. Table 1 presents three ADL for which the monitoring equipment suffices in order to formulate interesting event patterns, namely *transfer*, *mobility* and *stairs*. *Transfer* (or "changing position") refers to the ability of a person to get up from bed and lie down, stand up from a chair and sit down. The corresponding scores in the Barthel Index range from 0 to 3, as shown in Table 1. Since the goal is to determine user's level of independence and possibly detect signs of functional decline, we assume that the user is able to perform to some extend. As a result, we do not score *transfer* with a 0. *Mobility* ADL refers to the ability of a person to walk adequately well, while *stairs* ADL refers to a person's ability to walk stairs upwards/downwards. The respective scores in the Barthel Index range are presented in Table 1. Similar to *transfer*, we do not score *mobility* with 0 and 1; we also do not score *stairs* with 0.

3.1 Complex Events

As mentioned in Section 2, an HLE definition consists of Event Calculus rules expressing the conditions in which the HLE is initiated and terminated. Consider the event hierarchy presented in Figure 1, developed in collaboration with the USEFIL experts. The leaves in this tree-like hierarchy represent sensor-level data (LLEs), which are obtained by applying various aggregation and transformation techniques on raw sensor measurements, while each node represents an HLE. According to this representation, in order to obtain a Barthel score for the transfer ADL (root node), one should try to infer whether the user changed position, while receiving help for this task, and also take into account the ease and safety with which the user performs.

[4] For these ADL a surrogate, indirect index is obtained, which measures the increase in frequency of performing the ADL when someone else is in the house, in the long term.

ADL	Barthel scores	Related sensors
Transfer	0: unsafe - no sitting balance* 1: major help (one or two people, physical), can sit 2: minor help (verbal or physical) 3: independent	WWU Kinect camera Microphones
Mobility	0: immobile* 1: wheelchair independent, including corners, etc.* 2: walks with help of one person (verbal or physical) 3: independent (but may use any aid, e.g., stick)	WWU Kinect camera Microphones
Stairs	0: unable 1: needs help (verbal, physical, carrying aid) 2: independent up and down	WWU Kinect camera Microphones

Table 1. ADL, their respective scores in the Barthel Index and sensors which provide relevant LLEs.

Position change (ex. from sitting to standing, or from lying to sitting) depends on the corresponding HLEs, which in turn are inferred from sensor-level LLEs. Help is inferred by carer detection, which in turn may be inferred from kinect evidence (more than one persons) or microphone evidence (more than one speakers), and also by user-carer proximity, a value in meters, also provided by the kinect. The ease and safety with which the user transfers serves as a surrogate for the fact that we have no way to discriminate between major and minor help, as required by the Barthel standard (see definitions for Barthel scores of 1 and 2 in Table 1). Instead, we use the contextual knowledge of the ADL, that is, the user's speed and balance while changing position, as an indirect indication for the user's functional ability, which in turn is utilized in the extraction of the Barthel score for the ADL.

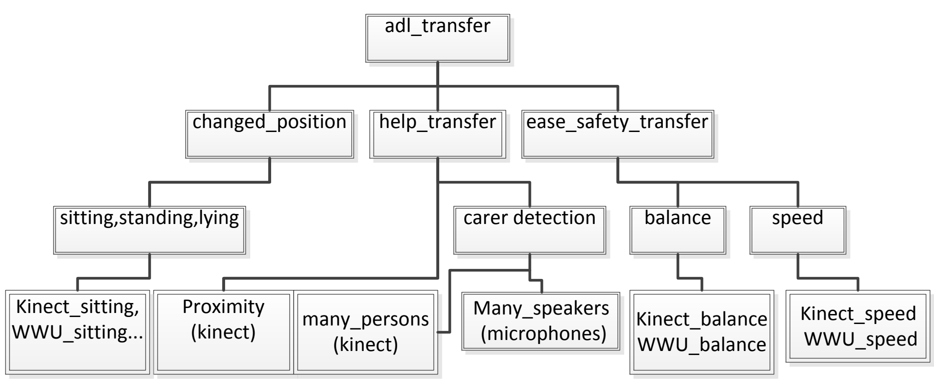

Fig. 1. An event hierarchy for the *transfer* ADL.

We next describe the construction of an Event Calculus program for Barthel-scoring the *transfer* ADL, starting from the bottom level (sensor data) of the event hierarchy and moving up to the root node of the target event.

The second from the bottom level of the event hierarchy consists of HLEs which result mostly from the combination of sensor-level data. For instance, *sitting* is defined in terms of two sensor-level events, one coming from the kinect and one from the wrist wearable unit, as the following two rules dictate:

$$\begin{array}{ll} \mathsf{initiatedAt}(sitting = true, T) \leftarrow & \mathsf{initiatedAt}(sitting = true, T) \leftarrow \\ \quad \mathsf{happensAt}(kinect\_sitting, T). & \quad \mathsf{happensAt}(wwu\_sitting, T). \end{array} \tag{4}$$

Note that by means of such rules, the probability of an HLE at time T results by the combined probabilities of all related evidence, up to that time. For instance from the following sensor evidence[5]

$$\begin{array}{l} 0.8 :: \mathsf{happensAt}(kinect\_sitting, 10) \\ 0.7 :: \mathsf{happensAt}(wwu\_sitting, 11) \end{array} \tag{5}$$

and the two rules in (4), the probability of *sitting* HLE at time 12 (i.e the probability of $\mathsf{holdsAt}(sitting = true, 12)$) is $0.8 + 0.7 - 0.8 \cdot 0.7 = 0.94$, provided that the HLE has not been terminated in the meantime.

The first HLE in the next level of the event hierarchy (see Figure 1) is recognized once the user changes position:

$$\begin{array}{l} \mathsf{initiatedAt}(changed\_position = true, T) \leftarrow \\ \quad \mathsf{initiatedAt}(sitting = true, T), \\ \quad \mathsf{terminatedAt}(standing = true, T). \end{array} \tag{6}$$

There are similar axioms for the remaining pose combinations. The *help_transfer* HLE definition is as follows:

$$\begin{array}{l} \mathsf{initiatedAt}(help\_transfer = true,\ T) \leftarrow \\ \quad \mathsf{happensAt}(proximity(Value),\ T), \\ \quad Value \leq 1, \\ \quad \mathsf{holdsAt}(carer\_detected = true,\ T). \end{array} \tag{7}$$

that is, help is inferred if a carer has been detected and user-carer proximity is less than 1 meter.

Let us now turn to the *ease_safety* HLE. As mentioned earlier, this HLE represents the ease and safety with which the user changes position, as indicated by speed and balance measurements, and its purpose is to serve as an indication for the user's functional ability, related to the *transfer* ADL. The speed and balance indications are provided by the kinect and WWU as a set of LLEs, which represent *steady* or *unsteady* balance and *fast, slow*, or *normal* speed. The task of combining these LLEs into a single indication for the user's functional abilities is not straightforward. For instance, a *fast* or *normal* transfer speed is an indication for an easy transfer. However, it may be accompanied by *unsteady* balance, an indication of unsafe transfer, in which case there is no interpretation of this contextual knowledge which says something meaningful for the user's functional abilities (in other words, an easy but unsafe transfer does not provide any insight for the user's ability w.r.t the *transfer* ADL). We address this issue as follows: First, we define values for the *ease_safety* HLE, which follow the actual Barthel scores for the *transfer* ADL. Intuitively, a score of 1 for *ease_safety* is to

[5] Recall that a statement of the form $p :: a$ in ProbLog means that a is true with probability p (see Section 2).

be interpreted as a difficult and unsafe transfer, a score of 3 as an easy and safe one and a score of 2 as something between the extreme cases.

We associate each LLE related to speed and balance to the *ease_safety* HLE in the obvious way according to which *fast* or *normal* speed, or *steady* balance indicates ease and safety (thus a value of 3 for *ease_safety* HLE), while *slow* speed or unsteady balance indicates difficulty or unsafety (thus a value of 1 for *ease_safety* HLE). This can be formulated as a set of rules of the following form (we omit the whole set of rules for brevity):

$$\begin{aligned}&\mathsf{initiatedAt}(\mathit{ease\_safety} = \mathit{3}, T) \leftarrow\\&\quad \mathsf{happensAt}(\mathit{kinect\_balance\_steady}, T).\end{aligned} \tag{8}$$

Then the average value of 2 for *ease_safety* is defined by means of a probabilistic rule, which weights the conjunction of the extreme cases at time T, by the confidence value of the most probable one. To do so, we use the concept of a *intentional probabilistic fact* [11] in ProbLog, i.e a probability associated to a rule, which is not defined explicitly, but is calculated at runtime, based on a number of constraints which must be satisfied, as in the following rule:

$$\begin{aligned}&P :: \mathsf{initiatedAt}(\mathit{ease\_safety} = \mathit{2}, T) \leftarrow\\&\quad P_1 :: \mathsf{holdsAt}(\mathit{ease\_safety} = \mathit{1}, T),\\&\quad P_2 :: \mathsf{holdsAt}(\mathit{ease\_safety} = \mathit{3}, T),\\&\quad P \;\texttt{is}\; \max\{P_1, P_2\}.\end{aligned} \tag{9}$$

P in Rule (9) is an intentional probability. By Rule (9) an average score will be attributed (with a significant probability) to the *ease_safety* HLE, only if the extreme scores for the HLE, both have significant probabilities, which means that contradictory evidence w.r.t ease and safety has been received.

For the definition of the top-level HLE in the hierarchy, we use two auxiliary HLEs, *transf_help* and *transf_no_help*, which are initiated by the conjunction of *changed_position* and *help_transfer*, or the negation of the latter. The definitions of the *adl_transfer* HLE is given disjunctively, for each different Barthel score. The definition of a 1-score is as follows:

$$\begin{aligned}&\mathsf{initiatedAt}(\mathit{adl\_transfer} = \mathit{1}, T) \leftarrow\\&\quad \mathsf{initiatedAt}(\mathit{transf\_help} = \mathit{true}, T),\\&\quad \mathsf{holdsAt}(\mathit{ease\_safety} = \mathit{1}, T).\end{aligned} \qquad \begin{aligned}&\mathsf{initiatedAt}(\mathit{adl\_transfer} = \mathit{1}, T) \leftarrow\\&\quad \mathsf{initiatedAt}(\mathit{ease\_safety} = \mathit{1}, T),\\&\quad \mathsf{holdsAt}(\mathit{transf\_help} = \mathit{true}, T).\end{aligned} \tag{10}$$

$$\begin{aligned}&p^1_{max} :: \mathsf{initiatedAt}(\mathit{adl\_transfer} = \mathit{1}, T) \leftarrow\\&\quad \mathsf{initiatedAt}(\mathit{transf\_help} = \mathit{true}, T),\\&\quad \mathsf{holdsAt}(\mathit{ease\_safety} = \mathit{2}, T).\end{aligned} \qquad \begin{aligned}&p^1_{max} :: \mathsf{initiatedAt}(\mathit{adl\_transfer} = \mathit{1}, T) \leftarrow\\&\quad \mathsf{initiatedAt}(\mathit{ease\_safety} = \mathit{2}, T),\\&\quad \mathsf{holdsAt}(\mathit{transf\_help} = \mathit{true}, T).\end{aligned} \tag{11}$$

$$\begin{aligned}&p^1_{min} :: \mathsf{initiatedAt}(\mathit{adl\_transfer} = \mathit{1}, T) \leftarrow\\&\quad \mathsf{initiatedAt}(\mathit{transf\_help} = \mathit{true}, T),\\&\quad \mathsf{holdsAt}(\mathit{ease\_safety} = \mathit{3}, T).\end{aligned} \qquad \begin{aligned}&p^1_{min} :: \mathsf{initiatedAt}(\mathit{adl\_transfer} = \mathit{1}, T) \leftarrow\\&\quad \mathsf{initiatedAt}(\mathit{ease\_safety} = \mathit{3}, T),\\&\quad \mathsf{holdsAt}(\mathit{transf\_help} = \mathit{true}, T).\end{aligned} \tag{12}$$

The definition consists of a set of crisp and a set of probabilistic rules, which aim to account for the inherent uncertainty of the Barthel scoring task. Rules

(10) state that if help was offered and the user's functional abilities as indicated by the *ease_safety* HLE are minimum, then a Barthel score of 1 should be attributed to the *transfer* ADL. On the other hand, Rules (11) state that if the functional abilities are not minimum, then the score may still be 1 (since help was offered), however with a reduced confidence, reflected in the probability p^1_{max}. Similarly, Rules (12) state that if the functional abilities are the highest possible, then a Barthel score of 1 should be attributed to *transfer* with an even smaller confidence (min and max in the probabilities denote that p_{min} is intended to be smaller than p_{max}). The definition for a 3-score is similar. The difference is that the *transf_no_help* auxiliary HLE is utilized, instead of the *transf_help* one. Finally, Rules (13)-(15) provide a definition for a 2-score in the Barthel index:

$$
\begin{array}{ll}
p^2_{max} :: \mathsf{initiatedAt}(\mathit{adl\_transfer} = \mathit{2}, T) \leftarrow & p^2_{max} :: \mathsf{initiatedAt}(\mathit{adl\_transfer} = \mathit{2}, T) \leftarrow \\
\quad \mathsf{initiatedAt}(\mathit{changed\_position} = \mathit{true}, T), & \quad \mathsf{initiatedAt}(\mathit{ease\_safety} = \mathit{2}, T), \\
\quad \mathsf{holdsAt}(\mathit{ease\_safety} = \mathit{2}, T). & \quad \mathsf{holdsAt}(\mathit{changed\_position} = \mathit{true}, T).
\end{array}
\tag{13}
$$

$$
\begin{array}{ll}
p^2_{min_1} :: \mathsf{initiatedAt}(\mathit{adl\_transfer} = \mathit{2}, T) \leftarrow & p^2_{min_1} :: \mathsf{initiatedAt}(\mathit{adl\_transfer} = \mathit{2}, T) \leftarrow \\
\quad \mathsf{initiatedAt}(\mathit{transf\_no\_help} = \mathit{true}, T), & \quad \mathsf{initiatedAt}(\mathit{ease\_safety} = \mathit{1}, T), \\
\quad \mathsf{holdsAt}(\mathit{ease\_safety} = \mathit{1}, T). & \quad \mathsf{holdsAt}(\mathit{transf\_no\_help} = \mathit{true}, T).
\end{array}
\tag{14}
$$

$$
\begin{array}{ll}
p^2_{min_2} :: \mathsf{initiatedAt}(\mathit{adl\_transfer} = \mathit{2}, T) \leftarrow & p^2_{min_2} :: \mathsf{initiatedAt}(\mathit{adl\_transfer} = \mathit{2}, T) \leftarrow \\
\quad \mathsf{initiatedAt}(\mathit{transf\_help} = \mathit{true}, T), & \quad \mathsf{initiatedAt}(\mathit{ease\_safety} = \mathit{3}, T), \\
\quad \mathsf{holdsAt}(\mathit{ease\_safety} = \mathit{3}, T). & \quad \mathsf{holdsAt}(\mathit{transf\_help} = \mathit{true}, T),
\end{array}
\tag{15}
$$

Recall that the intention in USEFIL is for a 2-score to account for "vague" situations, where help may have been provided or not, but the functional abilities of the user do not allow to classify her as fully independent or not. For example, if the user manages to stand up from the sitting position with no help, but with very low balance, it is unsafe to classify the user as independent. Instead, an intermediate score of 2 is an indication for the medical stuff in USEFIL that the user may need assistance for that particular ADL. Thus Rule (13) attributes a score of 2 by means of functional ability only, without taking help into account. Then, Rules (14) and (15) weight contradicting cases with a 2-score, that is, cases where functional ability is the lowest, but transfer was achieved with no help (Rules (14)) and cases where functional ability is the best possible, but help is inferred with a significant probability (Rules (15)).

The presented rules allow for the possibility that the *adl_transfer* HLE has more than one values at a particular time point. The obvious way to select between competing scores is to keep the one with the highest probability. However, delivering all the inferred Barthel scores allows a broader view of the user's status, which may be useful to the medical personel. In particular, the intended behavior of the formulated knowledge is to provide a dominant Barthel score for the ADL, but also provide additional indications via the less probable scores, particularly in the "vague" cases mentioned earlier.

The probabilities in the knowledge base may be tuned manually or may be learned using ProbLog's parameter learning utilities. For this work the probabilities where defined by experts, while we further tuned the values manually using synthetic data. In future work we will additionally use machine learning techniques to refine the probabilities from data.

The process of constructing event definitions for the other two ADL mentioned earlier, namely *mobility* and *stairs* is similar. Details are omitted due to space limitations.

4 Empirical Evaluation

USEFIL is an ongoing research project and real data is not yet available. Moreover, since ADL Barthel scoring is an empirical task, designed to be carried out by humans (medical personnel), by means of observing a patient, neither normative data, nor annotated datasets are available. Thus in order to validate that the formulated knowledge behaves as expected we performed experiments using synthetic data that simulate particular situations.

We defined a number of scripts, as presented in Table 2 for transfer, in order to annotate the generated datasets. The notation $score_1 \rightarrow score_2$ in Table 2 means that $score_1$ is the dominant score (the one with the highest probability), while $score_2$ is the second best score, which may be used as an additional indication for the user's functional status. For instance $3 \rightarrow 2$ means that the user is able to transfer with no help but there are some indications of functional decline, while a score of $2 \rightarrow 3$ is the opposite, that is, there are serious indications of functional decline, however the user may transfer with no assistance. We refer to such cases as *soft* Barthel scores, in contrast to the *hard* scores of 1 and 3. In Table 2 we order the various cases from the best possible, in terms of functional ability, to the worst possible.

We generated 50 instances of each script in Table 2 (a total of 600 instances) as follows: We defined a temporal window within which we assume that evidence related to transfer may result to the recognition of the *adl_transfer* HLE. This window was set to 15 seconds. To generate an instance of a script we generated related LLEs, randomly timed across the 15 second window, so that the values of the generated LLEs comply with the particular script. For example the following set of LLEs is an instance of case 1 in Table 2 for changing position from sitting to standing:

$$
\begin{aligned}
&0.784 :: \mathsf{holdsAt}(kinect\_sitting = true, 10).\\
&0.854 :: \mathsf{holdsAt}(wwu\_sitting = true, 11).\\
&0.478 :: \mathsf{happensAt}(many\_persons, 12).\\
&0.756 :: \mathsf{holdsAt}(proximity = 2.4, 15).\\
&0.553 :: \mathsf{happensAt}(many\_speakers, 16).\\
&0.786 :: \mathsf{holdsAt}(transfer\_speed = fast, 17).\\
&0.324 :: \mathsf{holdsAt}(transfer\_balance = steady, 21).\\
&0.788 :: \mathsf{holdsAt}(wwu\_standing = true, 24).\\
&0.698 :: \mathsf{holdsAt}(kinect\_standing = true, 25).
\end{aligned}
$$

Indeed, the above data indicate that the user is sitting at time 10 and standing at time 25 (thus *changed_position* will be recognized). In addition the *many_persons*

Case	Transfer	Help	Speed	Balance	Barthel score
1	yes	no	fast	steady	3
2	yes	no	normal	steady	3
3	yes	no	slow	steady	$3 \rightarrow 2$
4	yes	no	fast	unsteady	$3 \rightarrow 2$
5	yes	no	normal	unsteady	$3 \rightarrow 2$
6	yes	no	slow	unsteady	$2 \rightarrow 3$
7	yes	yes	fast	steady	$2 \rightarrow 1$
8	yes	yes	normal	steady	$2 \rightarrow 1$
9	yes	yes	slow	steady	$1 \rightarrow 2$
10	yes	yes	fast	unsteady	$1 \rightarrow 2$
11	yes	yes	normal	unsteady	$1 \rightarrow 2$
12	yes	yes	slow	unsteady	1

Table 2. Script definitions for different parameters of the *adl_transfer* HLE.

and *many_speakers* LLEs indicate the presence of a carer and the *proximity* LLE indicates that user-carer distance does not qualify for inferring physical help (it is larger that the threshold of 1 meter). Moreover speed and balance LLEs are valued as in case 1 of Table 2.

Instances were generated in a sorted fashion and between two scripts there is a period of time during which nothing happens, or the user is assumed to walk around the house. Thus the generated data form a (temporally) sorted stream. A large number of random USEFIL LLEs, irrelevant to the *transfer* task, was also added in the data. Time in the data ranges in the interval $[0, 10000]$ with a step of 1 (that is, the temporal distance between two consecutive events is 1). Four different datasets were generated in this manner: In the first one, the generated LLEs are *crisp* (i.e their probability is 1.0). In the second, noise in the form of random probabilities was injected in the speed and balance LLEs, while all other LLEs were crisp (*smooth noise*). In the third dataset all generated LLEs were noisy (*strong noise*). In the fourth dataset noise was injected to all LLEs as in the *strong noise* case, and additionally, up to 3 LLEs related to a particular script could be randomly omitted, or have different values from the ones required by the script (*strong-incomplete noise*). The purpose of the last dataset was to simulate more realistic cases where data may be missing due to hardware malfunction or delayed delivery, or they may have erroneous values with significant probability. The experiments consisted in evaluating the predictive accuracy of the presented event definitions for the different cases of noise.

As mentioned in Section 3, for these experiments, the parameters (probabilities) of the event definitions were manually determined based on expert knowledge. An HLE is recognized at time T if its probability at time T exceeds a *recognition threshold* p_1. In order to disambiguate between hard and soft Barthel scores, as presented in Table 2, we additionally defined a lower probability threshold p_2. In order to attribute a strong Barthel score $score_1$ to a transfer activity at time T, the probability of $score_1$ at T should be the maximum of the probabilities of all other Barthel scores, and it should exceed the threshold p_1. Moreover, the probability of the second most probable score $score_2$ should be

	crisp		smooth		strong		strong_incomplete	
	precision	recall	precision	recall	precision	recall	precision	recall
Transfer								
$transfer_3$	1.0	1.0	0.961	1.0	0.783	0.693	0.723	0.686
$transfer_{3\rightarrow 2}$	1.0	1.0	0.975	0.96	0.574	0.568	0.498	0.573
$transfer_{2\rightarrow 3}$	1.0	1.0	0.695	0.925	0.407	0.84	0.413	0.764
$transfer_{2\rightarrow 1}$	1.0	1.0	0.957	0.926	0.260	0.129	0.218	0.145
$transfer_{1\rightarrow 2}$	1.0	1.0	0.988	0.88	0.444	0.24	0.384	0.114
$transfer_1$	1.0	1.0	0.943	1.0	0.944	0.48	0.784	0.426

Table 3. Precision and recall for all identified variations of Barthel scores for the *adl_transfer* HLE, and for 4 different levels of noise.

smaller than the threshold p_2. In the opposite case, the activity will be attributed a soft score of $score_1 \rightarrow score_2$. For these experiments the above thresholds were set to $p_1 = 0.5$ and $p_2 = 0.2$ respectively.

Our experimental results are presented in Table 3. The results indicate that the formulated knowledge is able to classify correctly the level of functional dependency related to the *transfer* activity for the crisp case. It also achieves good results in the smooth noise case. Note that a perfect recall is achieved for the hard scores of 1 and 3 in the smooth noise case, that is, all activity instances of scores 1 and 3 were correctly classified. This is because the *help_transfer* HLE is crisply recognized in this case (the relevant LLEs are crisp), thus 1-score and 3-score activity instances are recognized with an increased probability by means of the crisp rules of the form (10) (and respective rules for a 3-score). Noise in the recognition of the *ease_safety* HLE is responsible for the erroneous classifications of activity instances, particularly for the soft score cases, where an activity instance contained contradictory evidence regarding balance and speed. Score $2 \rightarrow 3$ has the worst precision due to a large number of false positives ($3 \rightarrow 2$ and $2 \rightarrow 1$ instances incorrectly attributed with a score of $2 \rightarrow 3$).

Precision and recall drop significantly in the two remaining cases of noise. Hard scores (3 and 1) are classified relatively well, with the exception of the low recall for 1-scores in the strong noise case, which is attributed to an increased number of false negatives, that is, 1-score activity instances which were incorrectly classified as $1 \rightarrow 2$. Due to the increased noise and contradictory evidence for the *ease_safety* HLE in the generated activity instances, the predictive accuracy for some of the ambiguous (soft) scores is particularly low. This indicates the need for adjustment and refinement of the parameters (weights) of the knowledge base, an issue to be addressed in future work by means of machine learning. The worst recognition results are achieved with the fourth dataset, where in addition to the increased level of erroneous classification of functional ability, a number of transfer activities were not recognized at all, due to missing LLEs.

5 Related Work

A number of logic-based approaches for reasoning support in ambient intelligence and assisted living applications have been proposed. In [9] the authors present an approach based on if-then rules for the detection of elders' activity related to ADL and possible emergency conditions. The knowledge base in [9] was expert-engineered, while its construction was assisted by semi-supervised learning techniques (clustering and mixture models).

In [12] a method is presented for the recognition of HLEs using rules that impose temporal and spatial constraints between LLEs. Some of the constraints in the event definitions are optional, and as a result, an HLE may be recognized from incomplete information, but with lower confidence. The confidence of an HLE increases with the number of relevant (optional) LLEs. Due to noisy or incomplete information, the recognized HLEs may be logically inconsistent with each other. The method resolves these inconsistencies using the confidence, duration and number of involved LLEs.

In [15] the system SINDI (Secure and INDependent lIving) is proposed, which relies on Answer Set Programming (ASP) [2], a logic programming paradigm based on the stable model semantics. SINDI addresses uncertainty by means of a number of ASP features such as non-deterministic choice rules and cardinality constraints. ASP reasoning on sensory evidence extracts a number of "indicators" [15], relevant to interesting cases which are subject to monitoring, as for example ADL and sleep quality. These indicators are correlated in a dependency graph, which is reasoned upon, again with ASP, to deduce signs of improvement or aggravation of the monitored condition.

In [5, 6] the authors present a method for reasoning and decision-making support in a smart home setting. They use low-level and high-level ontologies to represent the domain, and interesting "situations" [5] to be recognized, respectively. Description logic reasoners and SWRL[6], which offer the ability to define temporal relations between entities, are utilized to detect target situations. Decision-making is assisted by an influence diagram based on Markov Logic Networks [16]. Ontological modelling and automata-based reasoning are utilized in [3], towards the detection of potential emergency situations for the elderly, while an ontology/description logic-based approach to seniors' ADL detection is presented in [7].

Common in all the above approaches is a limited/restricted handling of uncertainty. [9], [3] and [7] cannot handle uncertainty whatsoever, while [12] lacks a formal probabilistic semantics. [15] does not support probabilistic reasoning and relies on (crisp) ASP constructs in order to address uncertainty. The approach in [5, 6] is not able to handle uncertainty at the level of sensory data.

In contrast, the work presented here addresses uncertainty by means of a formal probabilistic semantics (ProbLog is based on the distribution semantics [17]), while it preserves the power of logic programming. The probabilistic version of the Event Calculus utilized in this paper was recently introduced in [18], in an effort to deal with uncertainty in activity recognition applications. The aim of our work is to evaluate the probabilistic Event Calculus in the context of a large, distributed monitoring system for assisted living.

[6] http://www.w3.org/Submission/SWRL/

6 Conclusions

We presented a Complex Event Recognition approach to detecting Activities of Daily Living and the level of functional ability, as defined by the Barthel index. The presented work is part of a real-world, unobtrusive, distributed monitoring system which is being developed in the FP7 project USEFIL, and involves various components such as multimedia processing and decision support. Our work is part of a research agenda that aims at evaluating the use of the Event Calculus in large distributed applications.

Our approach builds on previous work and proposes a logical framework based on the Event Calculus, properly extended in order to account for noise and uncertainty, by means of probabilistic reasoning with ProbLog. This framework exhibits a formal (probabilistic) semantics, and supports the representation of complex temporal phenomena for event recognition.

Further work includes experimentation with the real datasets that will be collected during the pilot studies of USEFIL. Moreover, to improve event recognition accuracy, we will employ techniques for weight learning and refinement (as opposed to setting the weights of rules manually), and abduction to deal with noise in the form of missing LLEs.

7 acknowledgement

The research leading to these results has received funding from the European Union, Seventh Framework Programme (FP7/2007- 2013), under grant agreements n^{o} 288532 (USEFIL) and n^{o} 619435 (SPEED). The first author would like to thank the Hellenic Artificial Intelligence Society for additional financial support.

References

1. Alexander Artikis, Marek Sergot, and Georgios Paliouras. A logic programming approach to activity recognition. In *Proceedings of the 2nd ACM international workshop on Events in multimedia*, pages 3–8. ACM, 2010.
2. Piero Bonatti, Francesco Calimeri, Nicola Leone, and Francesco Ricca. Answer set programming. In *A 25-year perspective on logic programming*, pages 159–182. Springer-Verlag, 2010.
3. Juan A Botia, Ana Villa, and Jose Palma. Ambient assisted living system for in-home monitoring of healthy independent elders. *Expert Systems with Applications*, 39(9):8136–8148, 2012.
4. Randal E Bryant. Graph-based algorithms for boolean function manipulation. *Computers, IEEE Transactions on*, 100(8):677–691, 1986.
5. Pedro Chahuara, François Portet, and Michel Vacher. Context aware decision system in a smart home: Knowledge representation and decision making using uncertain contextual information. *ARCOE-12*, page 52, 2012.
6. Pedro Chahuara, François Portet, and Michel Vacher. Making context aware decision from uncertain information in a smart home: A markov logic network approach. In *Ambient Intelligence*, pages 78–93. Springer, 2013.

7. Liming Chen, Chris D Nugent, and Hui Wang. A knowledge-driven approach to activity recognition in smart homes. *Knowledge and Data Engineering, IEEE Transactions on*, 24(6):961–974, 2012.
8. C. Collin, DT. Wade, S. Davies, and V. Horne. The barthel ADL index: a reliability study. *Disability & Rehabilitation*, 10(2):61–63, 1988.
9. Siddharth Dalal, Majd Alwan, Reza Seifrafi, Steve Kell, and Donald Brown. A rule-based approach to the analysis of elders activity data: Detection of health and possible emergency conditions. In *AAAI Fall 2005 Symposium*, 2005.
10. Opher Etzion and Peter Niblett. *Event processing in action*. Manning Publications Co., 2010.
11. Daan Fierens, Guy Van den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt. Inference and learning in probabilistic logic programs using weighted boolean formulas. *arXiv preprint arXiv:1304.6810, to appear in Theory and Practice of Logic Programming (TPLP)*, 2013.
12. Chrysi Filippaki, Grigoris Antoniou, and Ioannis Tsamardinos. Using constraint optimization for conflict resolution and detail control in activity recognition. In *Ambient Intelligence*, pages 51–60. Springer, 2011.
13. Angelika Kimmig, Bart Demoen, Luc De Raedt, Vítor Santos Costa, and Ricardo Rocha. On the implementation of the probabilistic logic programming language problog. *Theory and Practice of Logic Programming*, 11(2-3):235–262, 2011.
14. Robert Kowalski and Marek Sergot. A logic-based calculus of events. In *Foundations of knowledge base management*, pages 23–55. Springer, 1989.
15. Alessandra Mileo, Davide Merico, and Roberto Bisiani. Reasoning support for risk prediction and prevention in independent living. *Theory and Practice of Logic Programming*, 11(2-3):361–395, 2011.
16. Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.
17. Taisuke Sato. A statistical learning method for logic programs with distribution semantics. In *Proceedings of the twelfth international conference on logic programming (ICLP-1995)*, pages 715–729, 1995.
18. Anastasios Skarlatidis, Alexander Artikis, Jason Filippou, and Georgios Paliouras. A probabilistic logic programming event calculus. *Journal of Theory and Practice of Logic Programming (TPLP)*, 2014.