Event Recognition For Assisted Independent Living

Nikos Katzouris National Centre for Scientific Research (NCSR) "Demokritos" Athens 15310, Greece
nkatz@iit.demokritos.gr

Alexander Artikis National Centre for Scientific Research (NCSR) "Demokritos" Athens 15310, Greece
a.artikis@iit.demokritos.gr

Fillia Makedon Dep. of Computer Science & Engineering (CSE) University of Texas at Arlington Arlington, Texas 76019-0015
makedon@cse.uta.edu

Vangelis Karkaletsis National Centre for Scientific Research (NCSR) "Demokritos" Athens 15310, Greece
vangelis@iit.demokritos.gr

George Paliouras National Centre for Scientific Research (NCSR) "Demokritos" Athens 15310, Greece
paliourg@iit.demokritos.gr

ABSTRACT

We present the application of a recently proposed probabilistic logical formalism, on the task of sensor data fusion in the USEFIL project. USEFIL seeks to extract valuable knowledge concerning the well-being of elderly people by combining information coming from low-cost, unobtrusive monitoring devices. The approach we adopt to device its data fusion component is based on the Event Calculus and the stochastic logic programming language ProbLog and aims towards constructing a semantic representation of the received data, usable by a Decision Support System that will assist elderly people in their every day activities and will provide to doctors, relatives and carers insights on the user's behaviour and health.

Categories and Subject Descriptors

I.2.1 [**Applications and Expert Systems**]: Medicine and science

General Terms

Event Recognition, probabilistic reasoning, data fusion

Keywords

Assisted living, unobtrusive monitoring, knowledge representation and reasoning

1. INTRODUCTION

USEFIL[1] is an on-going FP7-funded project aiming to address the gap between technological research advances and the practical needs of elderly people by developing advanced but affordable in-home unobtrusive monitoring and web communication solutions. A key innovation of USEFIL is the usage of low cost non-specialized monitoring equipment that will assist the elderly in maintaining their independence and daily activities. Research in USEFIL is focused into two directions. The first one is developing systems that enable social awareness, allowing elderly people to feel in touch with their relatives, carers and doctors, through friendly and easy to use interfaces. The second one is engineering services that can unobtrusively monitor elderly behavioral indicators including health vital signs, functional and emotional status. Overall, the aim is to extend the time that older people can live independently in their homes, feeling safe and socially connected, while also limiting increases in public expenditure.

The USEFIL architecture consists of a number of components that are installed within the users' house, as well as components running on a remote server. The in-house hardware includes various interfaces, such as a slate tablet PC and/or, a web TV that provide to the user specific functionality for communicating with her relatives, cares or doctor, checking her health record, getting reminders about her medication and getting engaged into special-purpose games. In addition, it includes a number of low-cost sensors distributed within the house, including a depth camera (e.g. Kinect) next to the TV, a hidden camera, as well as a light wrist watch worn by the elderly. This hardware silently monitors the user's activity and uploads information relevant to her health to the USEFIL server. The aim is to provide long-distance access to a Decision Support System (DSS), which uses the uploaded data, together with long-term data accumulated over time and the electronic health record of the user, to draw the attention to changes and/or trends of the user's behavior and health.

In order to utilize sensor data, coming from heterogeneous resources like the ones mentioned above into operational knowledge, a data fusion step is necessary. USEFIL's data fusion component is responsible for interpreting sensor data into a semantic representation of the user's current status, in the form of a "snapshot". Such snapshots aggregate data over periods of time short enough so that they can be considered as a unit by the DSS and long enough to not swamp the DSS with unnecessary detail. In this paper we describe our approach to data fusion in the context of USEFIL. In particular, we employ a recently proposed probabilistic logic pro-

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

PETRA '13, May 29 - 31 2013, Island of Rhodes, Greece

http://dx.doi.org/10.1145/2504335.2504361

gramming formalism to represent and reason about USEFIL knowledge and perform real-time *Event Recognition*, that is, detection of event occurrences that satisfy certain patterns, from properly represented temporal sensor data. In the rest of this paper we present our approach and provide evidence supporting its applicability in the USEFIL project.

2. EVENT RECOGNITION

Event Recognition (ER) [4, 8, 9] refers to the automatic detection of event occurrences within a system. From a sequence of properly represented *Low Level Events* (LLEs), for example sensor data, an ER system recognizes *High Level Events* (HLEs) of interest, that is, events that satisfy some pattern. Consider, for example, the recognition of attacks on nodes of a computer network given the TCP/IP messages, the recognition of suspicious trader behaviour given the transactions in a financial market, the recognition of whale songs given a stream of whale sounds, and the recognition of human activities given multimedia content from surveillance cameras.

ER systems with a logic-based representation of event structures [3] 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 [10, 1]. The Event Calculus [6] in particular is a prominent logical formalism that has been excessively used for detecting occurrences of composite events from data streams of simple events.

2.1 The Event Calculus

For an introduction to the basic syntax and semantics of logic programming the reader is referred to [7]. For the rest of this paper, following Prolog convention, predicates and ground terms start with a lower case letter and variable terms start with a capital letter.

The EC is a temporal logic formalism for reasoning about events and their effects. It allows to model the evolution of a system in time by expressing the fact that certain properties of the system (*fluents*) are true, due to *events* that occur at certain *time points* and effect the fluents' truth value. The core *domain-independent axioms* of the EC incorporate the common sense *law of inertia*, according to which fluents persist over time, unless they are affected by an event. Informally, according to the EC axioms fluent F has the value V at time T if $F = V$ has been *initiated* by an event at some earlier time-point, and not *terminated* by another event in the meantime — law of inertia. The core axioms of the EC are presented below:

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

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

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

$$\begin{aligned}&\text{broken}(F = V_1,\ T_s,\ T) \leftarrow \\ &\quad \text{initiatedAt}(F = V_2,\ T_f), \\ &\quad V_1 \neq V_2, \\ &\quad T_s < T_f < T\end{aligned} \tag{4}$$

not represents "negation by failure", which provides a form of default persistence — inertia — of fluents. According to rule (1), $F = V$ holds at time-point T if $F = V$ held initially and has not been broken since. According to rule (2), $F = V$ holds at time-point T if the fluent F has been initiated to value V at an earlier time T_s, and has not been broken since. According to rule (3), a period of time for which $F = V$ holds is broken at T_f if $F = V$ is terminated at T_f. Rule (4) dictates that if $F = V_2$ is initiated at T_f then effectively $F = V_1$ is terminated at T_f, for all other possible values V_1 of F. Rule (4) therefore ensures that a fluent cannot have more than one value at any time.

2.2 USEFIL knowledge in the EC

Performing Event Recognition using the EC as the logical engine means to derive HLEs (which are represented by properties/fluents) that hold at certain time points by means of LLEs (events) that occur at certain time points and initiate or terminate the HLEs. In USEFIL, a third type of events, together with HLEs and LLEs is identified, namely Mid-Level Events (MLEs). These event occurrences span over a period of time (represent fluents) and thus they differ from LLEs, which are typically instantaneous events and are represented in the EC by happensAt/2 predicates. On the other hand, MLEs, which just as HLEs are represented in the EC by holdsAt/2 predicates, are often involved in the definition of higher-level events. In other words MLEs are HLEs in their own right, but at the same time they are part of an HLE hierarchy that leads to the recognition of higher-level events.

For instance, medical experts in USEFIL tell us that a reduced response to external stimuli is a strong indication for depressed mood. This knowledge may be represented in the EC by a rule[2] of the form:

$$\begin{aligned}&\text{initiatedAt}(\mathit{depressed}(\mathit{Person}) = \mathit{true},\ T) \leftarrow \\ &\quad \text{holdsAt}(\mathit{reduced\_response}(\mathit{Person}) = \mathit{true},\ T).\end{aligned} \tag{5}$$

which states that knowledge of the fact that the fluent *reduced_response*(*Person*) has the value *true* at some time point T, initiates a period of time for which the fluent *depressed*(*Person*) has the value *true*. On the other hand, reduced response to external stimuli is in itself an HLE which may be defined by the following formalization of an expert-provided knowledge:

[2] Such expert-provided rules are often called *domain-specific axioms*, in contrast to the *domain-independent axioms* presented in the previous section.

$$\text{initiatedAt}(reduced\_response(Person) = true, T) \leftarrow \text{holdsAt}(room\_change(Person) = low\_frequency, T). \quad (6)$$

$$\text{initiatedAt}(reduced\_response(Person) = true, T) \leftarrow \text{holdsAt}(reaction\_phone\_calls(Person) = slow, T). \quad (7)$$

$$\text{initiatedAt}(reduced\_response(Person) = true, T) \leftarrow \text{holdsAt}(unanswered\_phone\_calls(Person) = high, T). \quad (8)$$

$$\text{initiatedAt}(reduced\_response(Person) = true, T) \leftarrow \text{holdsAt}(reaction\_time\_door\_rings(Person) = slow, T). \quad (9)$$

Thus, reduced response to external stimuli is an MLE, which leads to the recognition of higher level events, like depressed mood, but at the same time may itself be recognized by other MLEs, in particular, the body conditions in rules (6)-(9), each of which is a different MLE. In turn, each of these MLEs in the body of rules (6)-(9) may be directly defined by means of LLEs provided by the monitored resources in the USEFIL setting. This hierarchy of HLEs which ultimately depend on properly represented sensor data (LLEs) is the basis of the data fusion functionality.

2.3 Probabilistic Event Calculus

Classical Logic Programming does not support probabilistic reasoning. However reasoning under uncertainty is of utmost importance in domains where one seeks to extract knowledge from data under different levels of certainty. A first case for this is the presence of noise in the data, the most common scenario in handling data from real-life sensors, particularly in cases like USEFIL's, where one depends on low-cost hardware. In such cases each piece of knowledge that is present in the incoming data may be accompanied by a number, i.e, a probability that signifies a level of certainty to the fact that this knowledge is indeed true.

Another case in which uncertainty handling significantly improves representation and reasoning are those where we have different degrees of certainty in the formalized knowledge. Consider for example the knowledge described in Table 1 concerning evidence that supports the fact that an elderly is depressed, as formulated by medical experts in the USEFIL project. The first row for instance states that a slow or low-stride gait may be an indication for depressed mood. The "average weight" that accompanies gait=slow/low-stride is a confidence value that signifies how certain we may be in that depressed mood is the case, provided that we observe gait=slow/low-stride. Thus, evidence with higher weight than gait, such as lights on/off=long_time for instance, are stronger indications for depressed mood.

These "degrees of correlation" between LLEs and MLEs (in our case gait, or lights on/of) and HLEs (depressed mood) may not be expressed and reasoned upon in the EC, where a period of time for which depressed mood holds may be equally initiated both by gait and lights on/of via EC domain-specific axioms of the form[3]:

[3] Note that in rules (10) and (11) gait is written in event notation, i.e. as a predicate $gait(value, Person)$ that is true at some time point, instead of writing it in fluent notation of the form $gait(Person) = slow$ for example. This is because gait is an LLE, that is, an event that occurs instantaneously and not a fluent whose value may persist for a certain period of time.

Table 1: Depressed mood definitions in USEFIL

Evidence	Value	Weight
gait	slow or low-stride	average
step count decrease	true	average
lights on/off	long time/all the time	high
tv on/off	long time/all the time	high
total time outdoors	low	high
reduced response to stimuli	true	high
socially inactive	true	high
crying	true	high
yell/shout	true	high
tired pose	true	high

$$\text{initiatedAt}(depressed(Person) = true, T) \leftarrow \text{happensAt}(gait(slow, Person), T). \quad (10)$$

$$\text{initiatedAt}(depressed(Person) = true, T) \leftarrow \text{happensAt}(gait(low\_stride, Person), T). \quad (11)$$

To address such issues in the USEFIL we employ a recently proposed [11] probabilistic version of the EC, based on the stochastic logic programming language ProbLog [5], a probabilistic extension of Prolog. ProbLog differs from Prolog in that it allows for probabilistic facts, which are facts of the form $p_i :: f_i$. In the expression $p_i :: f_i$, p_i is a real number in the range [0, 1] and f_i is a Prolog fact. If f_i is not ground, then the probability p_i is applied to all possible groundings of f_i. Classic Prolog facts are silently given probability 1.

Probabilistic facts in a ProbLog program represent random variables. Furthermore, ProbLog makes an independence assumption on these variables. This means that a rule which is defined as a conjunction of n of these probabilistic facts has a probability equal to the product of the probabilities of these facts. When a predicate appears in the head of more than one rules then its probability is computed by calculating the probability of the implicit disjunction created by the multiple rules. For example, for a predicate p with two rules $p \leftarrow l_1$ and $p \leftarrow l_2, l_3$, the probability $P(p)$ is computed as follows:

$$\begin{aligned} P(p) &= P((p \leftarrow l_1) \vee (p \leftarrow l_2, l_3)) = \\ &= P(p \leftarrow l_1) + P(p \leftarrow l_2, l_3) - P((p \leftarrow l_1) \wedge (p \leftarrow l_2, l_3)) = \\ &= P(l_1) + P(l_2) \times P(l_3) - P(l_1) \times P(l_2) \times P(l_3) \end{aligned}$$

In addition to probabilities attached to ground facts, ProbLog allows for probabilities attached to rules as in $p_i :: p \leftarrow l_1, \ldots, l_n$, $p_i \in [0, 1]$. In this case, the attached probability expresses the confidence that one has to the corresponding rule.

For further details on ProbLog representation and inference mechanisms, including negation handling, that is, reasoning probabilistically in the absence of knowledge and a caching mechanism for efficient reasoning described in [11] the reader is referred to [5, 11].

3. PROBABILISTIC INFERENCE IN USEFIL

In this section we describe USEFIL knowledge representation in a probabilistic context and we present a concrete inference example to demonstrate ProbLog's functionality. The following ProbLog program

expresses a portion of the expert-provided premises and corresponding confidence values that are involved in the recognition of the depressed mood HLE and participate in the scenario that we discuss in this section.

$$0.45 :: \text{initiatedAt}(depressed(Person) = true,\ T) \leftarrow \text{happensAt}(gait(Person, slow),\ T). \quad (12)$$

$$0.73 :: \text{initiatedAt}(depressed(Person) = true,\ T) \leftarrow \text{happensAt}(missed\_calls(Person, Calls, T),\ Calls > 3. \quad (13)$$

$$0.87 :: \text{initiatedAt}(depressed(Person) = false,\ T) \leftarrow \text{happensAt}(time\_outdoors(Person, TimeOut), T)\ TimeOut > 4. \quad (14)$$

Rule (12) states that observing a slow gait initiates a period of time for which the depressed mood HLE holds with probability 0.45. It is thus an indicator of depression of average weight. Similarly, rule (13) states that if the number of missed calls exceed a certain threshold, then we have a strong indication that depressed mood holds for a period of time, as shown by the attached probability. Finally, rule (14) states that the total time spent outdoors may, with high probability, initiate a period of time for which depressed mood is believed false.

Using this simple knowledge base it is possible to reason upon a user's psychological state in the USEFIL setting. Suppose for the shake of demonstration that USEFIL's data fusion component aggregates sensor data every three hours. Thus time stamps in the incoming data will represent inference points, which collectively form an inference cycle. The period of time that follows after the i-th inference point and precedes $i+1$ will be denoted by i^+. For instance, if an inference cycle starts at 7 am the first inference point denoted by 1 corresponds to 7 am and i^+ is the period of time up to 10 am. Based on the incoming data, after the reasoning process that takes place at the i-th inference point some HLEs may be initiated (resp. terminated), and will continue to hold (resp. not hold) throughout i^+ until proper evidence is provided at $i+1$.

Assume that the following probabilistic observations arrive at the data fusion component during the 1rst, 2nd and 3rd and inference points of a day's inference cycle:

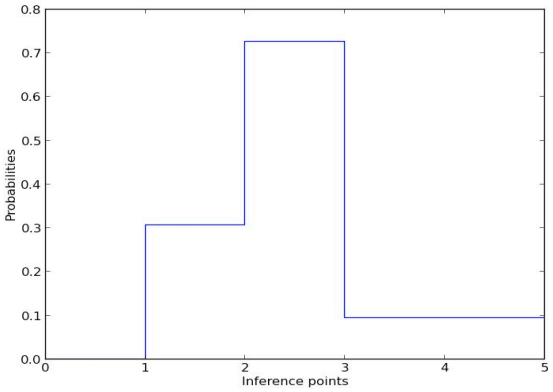
$$\begin{aligned} &0.68 :: \text{happensAt}(gait(user1, slow),\ 1)\\ &0.83 :: \text{happensAt}(missed\_calls(user1, 4),\ 2)\\ &0.93 :: \text{happensAt}(time\_outdoors(user1, 5),\ 3) \end{aligned} \quad (15)$$

The observed slow gait at the 1rst inference point will fire rule (12) and $holdsAt(depressed(user1) = true, 1^+)$ will be derived from the probabilistic EC engine with probability $0.45 \cdot 0.68 = 0.306$. Three hours later, at the 2nd inference point, *user*1 is reported to have missed 4 phone calls with probability 0.83. This knowledge will fire rule (13) from which there will be subsequently derived that $holdsAt(depressed(user1) = true, 2^+)$ with an increased probability:

$$\begin{aligned} P(init) &= P(init_1 \vee init_2) = \\ &= P(init_1) + P(init_2) - P(init_1 \wedge init_2) \\ &= 0.306 + 0.83 \cdot 0.73 - 0.306 \cdot 0.83 \cdot 0.73 = 0.726 \end{aligned}$$

where $P(init)$ in the above equation is the probability that depressed mood is initiated at the 2nd inference point, and $P(init_1)$ and $p(init_2)$ are the initiation probabilities derived from rules (12) and (13) respectively.

Figure 1: PrboLog inference



This increase in the initiation probability is an interesting feature of ProbLog. In the context of event recognition it may be interpreted by the fact that providing continuous indication that an HLE has occurred increases its probability, since we are are more inclined to believe that the HLE does actually hold.

Finally, evidence at the third inference point states, with probability 0.93, that *user*1 has spent 5 hours out of the house. This fact will cause rule (14) to significantly reduce the probability of depression: Since the incoming data terminate the depressed mood HLE with probability 0.87, the probability of $holdsAt(depressed(user1) = true, 3^+)$, that is, the probability of believing that *user*1 is depressed after the 3rd inference point is $1 - 0.87 = 0.13$. The resulting probability equals to the product of the prior probability, 0.726, times the probability that *depressed mood* is the case after inference point 3, thus $0.726 \cdot 0.13 = 0.094$.

Probabilistic inference with this small example is presented graphically in Figure 1. The probability of depressed mood is initially zero, increases in the first and second inference points, (and also remains constant between them) and finally decreases after the third inference point. Note that just as probabilities increase with positive evidence, they also decrease with continuous negative evidence. That is, if more negative observations for depression mood are presented at subsequent inference points, the probability in Figure 1 would eventually approach zero.

4. CURRENT STATUS AND FUTURE WORK

USEFIL is an ongoing project and real data for evaluating the proposed approach will be available shortly. However in [11], this approach has been evaluated on large-scale temporal data from a video surveillance application, as defined in the CAVIAR[4] project. Thus in order to obtain a preliminary evaluation of the adopted approach we performed experiments with CAVIAR data, at an in-home USEFIL machine, with promising results, given the fact that this machine is a low-cost computer with limited resources and it is not dedicated to data fusion, but it is also used for various other USEFIL-related tasks. In addition, much higher efficiency is expected to be achieved, since current work in progress involves the extension of the efficient EC dialect for real-

[4] http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

time Event Recognition, presented at [2], in order to support probabilistic reasoning, towards an efficient logical formalism for reasoning under uncertainty. This work-in-progress is expected to allow for real-time data fusion from large-scale sensor data, even with the limited resources provided in the USEFIL setting.

Acknowledgments

The research work presented in this paper is funded by the FP7 project USEFIL.

References

[1] A. Artikis, G. Paliouras, F. Portet, and A. Skarlatidis. Logic-based representation, reasoning and machine learning for event recognition. In *Proceeding of Destributed Event Based Systems (DEBS)*, pages 282–293, 2010.

[2] A. Artikis, M. Sergot, and G. Paliouras. Run-time composite event recognition. In *International Conference on Distributed Event-Based Systems (DEBS), ACM*, 2012.

[3] A. Artikis, A. Skarlatidis, F. Portet, and G. Paliouras. Logic-based event recognition. *Knowledge Engineering Review*, 27(4), 2012.

[4] O. Etzion and P. Niblett. *Event processing in action*. Manning Publications Co., 2010.

[5] A. Kimmig, B. Demoen, L. De Raedt, V. S. Costa, and R. Rocha. On the implementation of the probabilistic logic programming language problog. *Theory and Practice of Logic Programming*, 11(2-3):235–262, 2011.

[6] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):67âĂŞ96, 1986.

[7] J. Lloyd. *Foundations of Logic Programming*. Springer, 1987.

[8] D. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc, 2001.

[9] D. Luckham and R. Schulte. *Event processing glossary version 1.1*. Event Processing Technical Society, 2008.

[10] A. Paschke. ECA-ruleML: An approach combining ECA rules with temporal interval-based KR event logics and transactional update logics. Technical report, Technische Universitat Munchen, 2005.

[11] A. Skarlatidis, A. Artikis, J. Filippou, and G. Paliouras. A probabilistic logic programming event calculus. *Theory and Practice of Logic Programming*, to appear, 2013.