Accurate Temporal Relationships in Sequences of User Behaviours in Intelligent Environments

Asier Aztiria and Juan Carlos Augusto and Rosa Basagoiti and Alberto Izaguirre

Abstract Intelligent Environments are supposed to act proactively anticipating user's needs and preferences in order to provide effective support. Therefore, learning user's frequent behaviours is essential to provide such personalized services. In that sense, we have developed a system, which learns those frequent behaviours. Due to the complexity of the entire learning system, this paper will focus on discovering accurate temporal relationships to define the relationships between actions of the user.

1 Introduction

Ambient Intelligence (AmI) [6] [14] [15] can be understood as 'a digital environment that proactively, but sensibly, supports people in their daily lives' [2]. Some of the potential benefits that this technology can bring to people in their daily lives include making an environment more comfortable, safer and more energy efficient. In order to achieve these objectives, the environment should learn patterns of the user which means that the environment has to gain knowledge about the preferences, needs and habits of the user in order to be in a better position to assist the user adequately [9].

Let us consider the following scenario, which exemplifies a common behaviour of a user. *On weekdays Michael's alarm clock goes off ('Alarm on') few minutes after 08:00am. Approximately 10 minutes after getting up he usually steps into the bathroom ('Bathroom on') and (4 seconds after) he turns on the lights of the bathroom ('BathLights on') if the bathroom is dark (bathroom light level $<$10). On Tuesdays, Thursdays and Fridays he usually takes a shower ('Shower on'); Michael prefers the temperature of the water to be around 24-26 degrees Celsius in the winter and around 21-23 degrees Celsius in the summer. Before he leaves the bathroom ('Bathroom off')he turns off the lights ('BathLights off').*

Asier Aztiria
University of Mondragon, e-mail: aaztiria@eps.mondragon.edu

Juan Carlos Augusto
University of Ulster, e-mail: jc.augusto@ulster.ac.uk

Rosa Basagoiti
University of Mondragon, e-mail: rbasagoiti@eps.mondragon.edu

Alberto Izaguirre
University of Mondragon, e-mail: aizagirre@eps.mondragon.edu

Knowing users' frequent behaviours allows the environment to act intelligently and proactively. In Michael's case, it could mean that the environment automatically turns on and off the lights, sets the temperature of the water and so on. Unlike previous systems, *automation* of actions and/or devices in intelligent environments is based on learned patterns, making sure they adapt to users' common behaviours. Knowledge extracted from these patterns can also be used in order to *understand* his behaviour. For example, the analysis of frequent interaction with objects and devices in the house can facilitate the detection of unhealthy habits (e.g., the system detects that Michael does not brush his teeth in the mornings). Making the environment more efficient in terms of *saving energy* (e.g. by turning off the lights when he leaves) or *increasing safety* (e.g. turning off the water or issuing alarms when detecting that Michael left it on and he will not return soon) are other dimensions of daily life that can be supported by the Intelligent Environment thanks to the knowledge it has collected.

In order to achieve these objectives, we have developed a software which allows an Intelligent Environment to discover frequent behavioural patterns, and we have named it Sequential Patterns of User Behaviour System (SPUBS). Due to the complexity of SPUBS, this paper will focus on the step of discovering accurate temporal relationships between actions. The reminder of the paper is organized as follows. Section 2 describes briefly the SPUBS. Section 3 summarizes previous related work done in learning highlighting the task of discovering temporal relationships. In Section 4 we explain our approach for discovering accurate temporal relationships. Finally Section 5 shows the results of the validation experiments and Section 6 the conclusions.

2 Sequential Patterns of User Behaviour System

Sequential Patterns of User Behaviour (SPUBS) is a system that discovers user's common behaviours and habits from data recorded by sensors. The core of the system is the learning algorithm $\mathscr{A}_{SPUBS}$, which combined with a language $\mathscr{L}_{SPUBS}$, allows to discover and represent the patterns. This algorithm is divided in four steps which represent the four logical steps to discover frequent and comprehensible patterns. A brief description of each module follows:

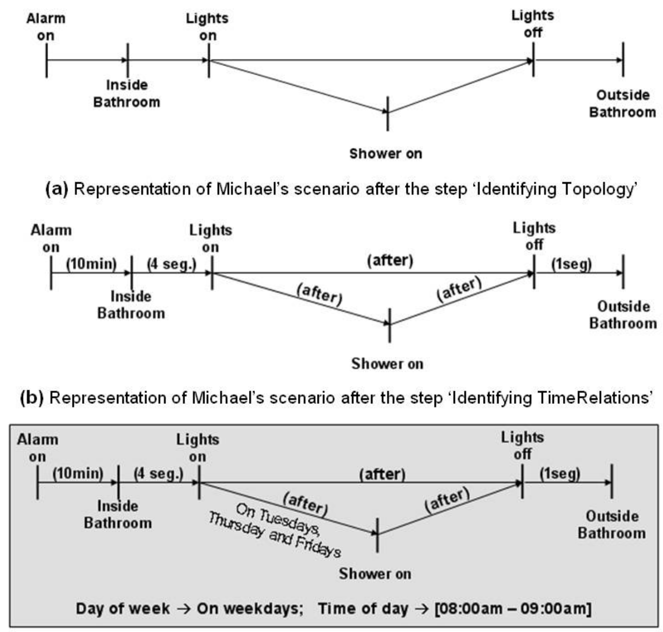
Identifying Frequent Sets: The objective of this first step is to identify the sets of actions that frequently occur together (for further details see [4]). In Michael's scenario this step discovers that Michael usually carries out the following actions together:

'Alarm on', 'Bathroom on', 'Bathroom off', 'BathLights on', 'BathLights off', 'Shower on' and 'Shower off'

Identifying Topology: This step discovers in what order the user usually carries out the sets of actions discovered in the first step, giving it a sense of sequence (for further details see [5]). Thus, in Michael's scenario it discovers that the activation of the alarm clock comes first, then he goes into the bathroom, then he turns the lights on and so on. See Figure 1(a) for the output achieved after this step.

Identifying Time Relations: The objective of this step is to relate the actions in terms of time. In contrast with the previous step this step tries to relate the actions as accurately as possible (for further details see Section 4). See Figure 1(b) for the output achieved after this step.

Identifying Conditions: The last step of the algorithm is to discover specific conditions for situations where different options are, and general conditions in order to contextualize the whole sequence. In Michael's scenario, on the one hand it defines that he only has a shower on Tuesdays, Thursdays and Fridays. On the other hand this step also defines that the whole sequence occurs only on weekdays and it starts few minutes after 08:00am and it finishes before 09:00am. See Figure 1(c) for the output achieved after this step.



(a) Representation of Michael's scenario after the step 'Identifying Topology'

(b) Representation of Michael's scenario after the step 'Identifying TimeRelations'

(c) Representation of Michael's scenario after the step 'Identifying Conditions'

Fig. 1 Evolution of Michael's scenario through the $\mathscr{A}_{SPUBS}$

Although the whole algorithm is already developed, because of space restrictions, this paper will be focused on the first step of 'Identifying Time Relations'.

3 Related Work

Learning is an essential feature in any Intelligent Environment. However, it has not been devoted as much attention in the literature as it may require. Next, different approaches are analysed emphasizing the way they discover and represent the temporal relationships between actions.

So far, most of the applications [8] [10] [13] where a learning process was involved have related an action to the global situation of that moment, instead of relating actions among them. Representing user's behaviours by means of sequences, as well as creating complete and comprehensible representation, allows us to relate actions in terms of time. The group that has been working in MavHome and Casas projects [12] was one of the first groups that emphasized the importance of associating actions and their temporal relationships. They started using time intervals associated with activities rather than instantaneous time points. They used Allen's temporal logic [1] in order to represent time intervals, producing fairly intuitive sequences of actions. In that sense, considering Michael's example, the system would be able to detect that he first gets up, then he goes into the bathroom and then he turns on the lights. This system only considers Allen's temporal logic relations (which define relations qualitatively), ruling out quantitative relations, which allow us to define more accurate relationships. In that sense, Aztiria et al. [3] suggested a process to discover quantitative temporal relationships, but they only applied it to one-to-one relationships.

4 Identifying Time Relations

The step of 'Identifying topology' provides a sequential representation of patterns. This way of representation implies a first definition of temporal relationships due to the fact the sequence itself defines a temporal order of actions. In that sense, all relationships defined by the sequences are *qualitative* relationships defined by the term '*after*'.

Qualitative relationships allow us to understand the logic order of actions within a sequence. Even so, such relationships could be better defined if we were able to discover a regular *quantitative* relationship. Compared to *qualitative* relationships, *quantitative* relationships provide a higher quality information, being possible to use them for other purposes. One of those purposes can be the automation of devices, which is possible with *quantitative* relationships. We will consider Michael's behaviour of turning on the light of the bathroom 4 seconds after he goes into (defined by ActionPattern 0). If such a relationship is defined by means of a *qualitative* term like 'after', the system cannot infer when it has to turn on the light because it does not know if the time delay is 4 seconds, 5 minutes or 2 hours. However, *quantitative* relationships (4 seconds in Michael's case) allows the system to turn on the light at right time.

```
(ActionPattern 0)
  ON occurs (Bathroom, On,t0)
  IF context (Bathroom Light level is (<,10))
  THEN do (On, BahtLights, t) when t=t0+4s
```

Therefore, the objective of this step of 'Identifying Time Relations' is to discover *quantitative*, or accurate temporal relationships, to better define user's behaviours. For that, SPUBS includes two different algorithms. Following, these two algorithms as well as the process of getting necessary information will be described.

4.1 Data Collection for Identifying Time Relations

The first step to discover such relationships will be to collect necessary data. Relationships to study will already be defined by the topology. Thus, for each relationship defined by the topology we will collect particular time relationships of all occurrences.

Let us consider again Michael's example and the relationship between the actions 'Bathroom on' and 'BathLights on'. For each case where a occurrence of 'Bathroom on' is followed by a occurrence of 'BathLights on', SPUBS collects the time distance between both actions. Let us consider time relationships of Michael's example are depicted in Figure 2.

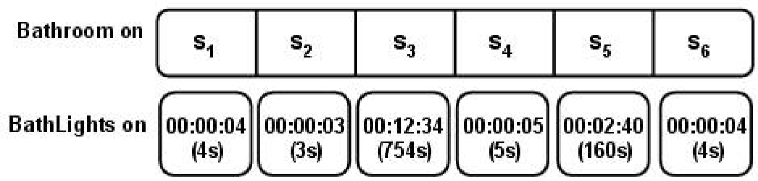


Fig. 2 Time Distances between occurrences of 'Bathroom on' and 'BathLights on'

Once temporal distances are collected, the next step is to identify interesting time relationships. For that, SPUBS includes two algorithms, 'Basic Algorithm' and 'EM Algorithm', so that the user of the system may choose any of them to identify such relationships. Both algorithms are based on the same idea of grouping occurrences taking into account their similarities and deciding if a group represents an interesting time relationship.

4.2 'Basic Algorithm' for Identifying Time Relations

The technique to make groups could be as complex as we can imagine and different techniques could be suggested to accomplish this task. SPUBS includes a very basic algorithm, which groups values that are within a range established by:

$$[min, max] = \overline{x} \pm (\overline{x} * tolerance) \qquad where \qquad \overline{x} = \frac{\sum_{i=1}^{n} a_i}{n} \tag{1}$$

with: tolerance = tolerated deviation from $\overline{x}$ (%); a_i = time distance of an element; and n = number of elements

If a value does not fulfil the requirements to join any group, a new group is created using that value as group's mean value. Every time a new value is added to a group the mean value of that group is recalculated.

Considering Michael's example and time distances depicted in Figure 2, the process of making groups would be (let us consider 50% as tolerance percentage):

(e_1,4s); There is no group; create(group0, $\bar{x}$ (4s), [2,6])

Tolerance = 4 $\pm$ (4 * 0.5)

(e_2,3s); 3=[2,6]; join(group0, $\bar{x}$ (3.5s), [1.75,5.25])

Tolerance = 3.5 $\pm$ (3.5 * 0.5)

(e_3,754s); 754$\neq$[1.75,5.25]; create(group1, $\bar{x}$ (754s), [377,1132])

Tolerance = 754 $\pm$ (754 * 0.5)

(e_4,5s); 5=[1.75,5.25]; join(group0, $\bar{x}$ (4s), [2,6])

Tolerance = 4 $\pm$ (4 * 0.5)

(e_5,160s); 160$\neq$[2,6] and 160$\neq$[377,1132]; create(group2, $\bar{x}$ (160s), [80,240])

Tolerance = 160 $\pm$ (160 * 0.5)

(e_6,4s); 4=[2,6]; join(group0, $\bar{x}$ (4s), [2,6])

Tolerance = 4 $\pm$ (4 * 0.5)

Once groups are created, the following step is to decide what groups define interesting time relationships. For that, SPUBS's user must define a minimum confidence level. Thus, those groups that cover more instances than established by the minimum confidence level would be considered as interesting, being their mean value the temporal relationship. In Michael's case, let us consider a minimum confidence of 25%.Thus, the only group considered as interesting would be 'group 0', which covers 4 out of 6 occurrences.

4.3 'EM Algorithm' for Identifying Time Relations

Besides the 'Basic Algorithm', SPUBS includes another algorithm, which creates clusters of time distances based on the Expectation-Maximization (EM) algorithm [11]. The basic idea of the EM algorithm is to estimate the maximum likelihood between parameters. An important advantage of this algorithm is that it automatically calculates the necessary number of groups and includes each occurrence in its corresponding group.

Once groups are created by the EM algorithm, the process of deciding what groups defines an interesting time relationship is same as 'Basic Algorithm'.

5 Validation and Results

In order to validate the system we applied it to artificial data generated at the University of Ulster and then to a real dataset collected from Washington State University's (WSU's) Smart Apartment. The data collected in the WSU smart apartment [7] represents participants performing five Activities of Daily Living (ADLs) in the apartment: making a phone call, washing hands, cooking, taking medicine/eating the food and cleaning (See Table I for actions involved in each activity).

In all, the sensors installed in WSU smart apartment are:

Table 1 Actions involved in each ADL.

Activity	Involved Actions
Making a phone call	Phone Book (ON) –>Phone (ON) –>Phone (OFF)
Washing hands	Water (ON) –>Water (OFF)
Cooking	Cabinet (ON) –>Raisins (ON) –>Oatmeal (ON) –>Measuring spoon (ON) –>Bowl (ON) –>Sugar (ON) –>Cabinet (OFF) –>Water (ON) –>Water (OFF) –>Pot (ON) –>Burner (ON) –>Burner (OFF)
Taking medicine and Eating	Cabinet (ON) –>Medicine (ON) –>Cabinet (OFF) –>Water (ON) –>Water (OFF) –>Cabinet (ON) –>Medicine (OFF) –>Cabinet (OFF)
Cleaning	Water (ON) –>Water (OFF)

- 14 sensors on objects such as phone, medicine container or cabinet.
- 27 motion sensors.

As the set of actions involved in these 5 ADLs and the order of such actions were known in advance, we knew what patterns should be discovered by SPUBS. As we expected, the steps of 'Identifying Frequent Sets' and 'Identifying Topology' discovered the actions involved in the sequence as well as their order. Once the topology was discovered we applied both algorithms, described in Section 4, to discover temporal relationships. The results obtained in that process showed that it was possible to discover an accurate temporal relationships in 11 out of 17 relationships. Following some of the ActionPatterns, i.e. parts of the sequence, discovered by the system are represented using $\mathscr{L}_{SPUBS}$:

```
(ActionPattern 1)
  ON occurs (Phone, On,t0)
  IF (by default = true)
  THEN do (Off, Phone, t)
       when t=t0+50s

(ActionPattern 2)
  ON occurs (Cabinet, On,t0)
  IF (by default = true)
  THEN do (On, Medicine, t)
       when t=t0+3s

(ActionPattern 3)
  ON occurs (Medicine, On,t0)
  IF (by default = true)
  THEN do (Off, Cabinet, t)
       when t=t0+2s

(ActionPattern 4)
  ON occurs (Cabinet, Off,t0)
  IF (by default = true)
  THEN do (On, Water, t)
       when t=t0+16s
```

6 Conclusions

Intelligent Environments need to know the common behaviours and preferences of the user in order to meaningfully act. Representing such common behaviours by means of sequences facilitates their understanding. Besides, such a representation allows us to relate such actions in terms of time. In that sense, SPUBS includes two algorithms to get more accurate relationships between actions. Accurate temporal relationships, represented by means of numerical values, defines better such relationships than using qualitative values, due to the fact they allow the system to use the sequences for additional purposes, for example to automate devices' activation.

References

1. J. Allen. Towards a general theory of action and time. In *Artificial Intelligence*, volume 23, pages 123–154, 1984.
2. J. C. Augusto and D. J. Cook. Ambient intelligence: applications in society and opportunities for ai. In *20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, 2007.
3. A. Aztiria, J. C. Augusto, A. Izaguirre, and D. J. Cook. Learning accurate temporal relations from user actions in intelligent environments. In *Proceedings of the 3rd Symposium of Ubiquitous Computing and Ambient Intelligence*, volume 51/2009, pages 274–283, 2008.
4. A. Aztiria, A. Izaguirre, R. Basagoiti, J. C. Augusto, and D. J. Cook. Discovering of frequent sets of actions in intelligent environments. In *Proceedings of the 5th International Conference on Intelligent Environments*, pages 153–160, 2009.
5. A. Aztiria, A. Izaguirre, R. Basagoiti, J. C. Augusto, and D. J. Cook. Automatic modeling of frequent user behaviours in intelligent environments. In *Proceedings of the 6th International Conference on Intelligent Environments (Submitted)*, 2010.
6. D. Cook, J.C. Augusto, and V.R. Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277–298, 2009.
7. D. Cook and M. Schmitter-Edgecombe. Activity profiling using pervasive sensing in smart homes. *IEEE Transactions on Information Technology for Biomedicine*, 2008.
8. D. J. Cook and S. K. Das. How smart are our environments? an updated look at the state of the art. In *Pervasive and Mobile Computing*, volume 3, pages 53–73. Elsevier Science, 2007.
9. M. Galushka, D. Patterson, and N. Rooney. Temporal data mining for smart homes. Designing Smart Homes. The Role of Artificial Intelligence, pages 85–108. Springer-Verlag, 2006.
10. H. Hagras, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, and H. Duman. Creating an ambient-intelligence environment using embedded agents. *IEEE Intelligent Systems*, 19(6):12–20, 2004.
11. R. Hogg, J. McKean, and Allen Craig. *Introduction to Mathematical Statistics*, pages 359–364. Pearson Prentice Hall, 2005.
12. V. R. Jakkula, A. S. Crandall, and D. J. Cook. Knowledge discovery in entity based smart environment resident data using temporal relation based data mining. In *7th IEEE International Conference on DataMining*, pages 625–630, 2007.
13. M. C. Mozer, R. H. Dodier, M. Anderson, L. Vidmar, R. F. Cruickshank, and D. Miller. *The neural network house: an overview*, pages 371–380. Current trends in connectionism. Erlbaum, 1995.
14. H. Nakashima, H. Aghajan, and J.C.Augusto. *Handbook on Ambient Intelligence and Smart Environments*. Springer Verlag, 2009.
15. M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, 1991.