# Automatic Modeling of Frequent User Behaviours in Intelligent Environments

Asier Aztiria, Alberto Izaguirre, Rosa Basagoiti
University of Mondragon
Spain

Juan Carlos Augusto
University of Ulster
United Kingdom

Diane J. Cook
Washington State University
USA

***Abstract*—Intelligent Environments depend on their capability to understand and anticipate user's habits and needs. Therefore, learning user's common behaviours becomes an important step towards allowing an environment to provide such personalized services. Due to the complexity of the entire learning system, this paper will focus on the automatic discovering of models of user's behaviours. Discovering the models means to discover the order of such actions, representing user's behaviours as sequences of actions.**

***Keywords: Intelligent Environments, Learning behavioural patterns, workflow mining.***

## I. INTRODUCTION

Intelligent Environments defined as digital environments that proactively, but sensibly, support people in their daily lives [1] can be considered as a promising opportunity to use technology for the benefit of society with a range of applications being explored [2]. Some of the potential benefits that this technology can bring to our daily lives include making the environment we live and work in 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 [3,4].

Let us consider the following scenario describing typical activities related to the preparation of breakfast. We tag some of those steps so that they can be used later on to explain how our algorithm works. *On weekdays after getting up and getting washed, Michael goes into the kitchen to have breakfast. He usually has breakfast some coffee with milk and some juice. For that, he firstly opens the cupboard ('Cupboard on') and gets a cup ('Cup on'). Once he gets a cup and closes the cupboard ('Cupboard off'), he opens the fridge ('Fridge on') and gets the milk bottle ('Milk on') as well as the bottle of juice ('Juice on'). Then, he closes the fridge ('Fridge off') and opens the cupboard again ('Cupboard on') to get some coffee ('Coffee on') and the medicine container ('Medicine Container on'). Once he has the necessary ingredients he starts preparing the breakfast.*

If a system can learn the way the user behaves in some particular circumstances then it can act upon the environment to intelligently assist with daily life activities. For example, in Michael's scenario, once the system knows his habits it can act proactively in order to make his life easier and safer. In this case acting proactively could mean the environment could remind him that he has to take the pills. Knowing his habits and preferences the environment can also *support and facilitate some activities* through house automation. For example, knowing everyday he has coffee with milk and some juice as breakfast the fridge can predict when he will need to buy those products and order them from the supermarket automatically. The knowledge acquired from the patterns can also be used in order to *understand his behaviour*. For example, let us assume Michael is an older citizen with memory problems and his house is a smart home that helps him and his family along the day. Michael's relatives or caregivers can analyze if he takes his medication and observe minimal nutrition and hygiene guidelines. Making the environment more efficient in terms of *energy saving* (e.g., by managing lighting and heating) or *increasing the safety* (e.g., by identifying that usually Michael takes no more than 15 minutes when it goes to the toilet in the night) are other services the environment can provide if it is aware of the common behaviours of the user.

The remainder of the paper is organized as follows. Section II describes the system, SPUBS, we have developed to learn behavioural patterns. Section III summarizes previous related work done in topology or workflow mining, focusing on Intelligent Environments. In Section IV we explain our strategy to discover the topology behind a group of actions. Section V explains other steps which are part of the overall approach. Section VI shows the results of the validation experiments and finally Section VII provides the conclusions of this research.

## II. 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. To tackle different aspects of the problem with different strategies and algorithms we have created a three-layered architecture (see Figure 1). This modular approach also facilitates the organization of our system to distinguish those aspects related to particular environments from those aspects that can be generalized for all environments.

The core of the SPUBS system is the learning algorithm $A_{SPUBS}$ which combined with a language $L_{SPUBS}$ allows to discover and represent the patterns. It is worth mentioning that the learning layer, in contrast with the two other layers, is independent of the specific environment we are dealing with. In other words, the algorithm created in this layer as well as the language used in order to represent the patterns can be used in any environment.

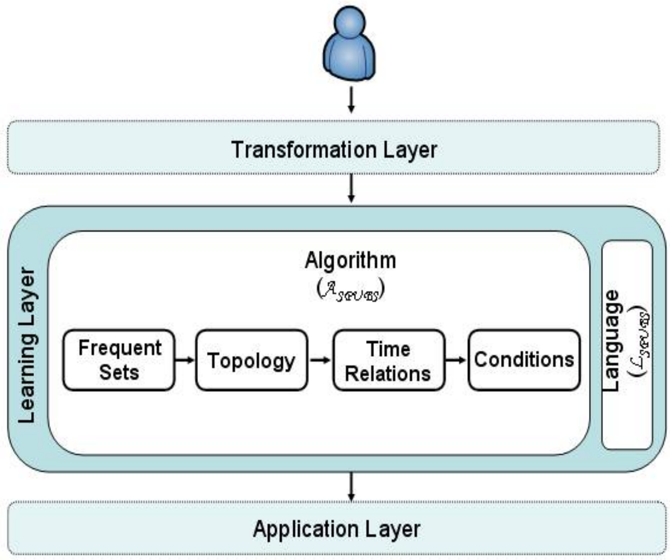


**Figure 1: Global Architecture of SPUBS**

Figure 1 also shows the internal structure of the algorithm ($A_{SPUBS}$) which is the core of the whole system. 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:

'Cupboard on', 'Cupboard off', 'Cup on', 'Fridge on', 'Fridge off', 'Milk on', 'Juice on', 'Coffee on' and 'Medicine Container on'

**Identifying Topology:** This step discovers in what order the user usually carries out the sets of actions discovered in the first step, giving them a sequential or, more typically, a quasi-sequential structure (for further details see Section IV). Thus, in Michael's scenario it discovers that the user usually opens the cupboard and then he gets the cup and so on. See Figure 2(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. It further refines the previous step trying to relate the actions as accurately as possible. In Michael's scenario this step discovers how long it takes Michael from he opens the cupboard until he gets a cup. Thus, knowing what the typical time relation is between two actions allows the system to perform more accurate actions. See Figure 2(b) for the output achieved after this step.

**Identifying Conditions:** The last step of the algorithm is to discover specific and general conditions in order to contextualize the whole sequence. In Michael's scenario, on the one hand it defines when he takes the pill and when he does not. On the other hand this step also defines that the whole sequence occurs only on weekdays and it happens between 08:15AM and 09:00AM. See Figure 2(c) for the output achieved after this step.

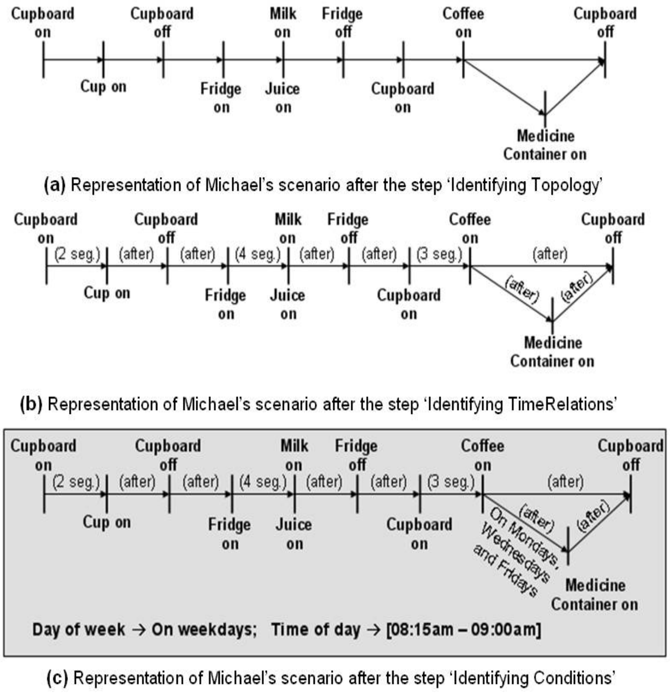


**Figure 2: Michael's sequence's evolution through $A_{SPUBS}$**

## III. 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 [5]. This section provides a brief account of some interesting related approaches with emphasis on the way they model user's behaviours.

So far, most of the applications where a learning process was involved have related an action to the global situation of that moment instead of relating actions among them. Such approaches create independent pieces of information instead of a complete and comprehensible representation of the user's behaviour. Hence some research groups started the creation of sequences that relate user's actions. The MavHome and Casas projects [6] [7] were amongst the first ones to highlight the importance of this, but they only discovered one to one associations instead of the whole sequence. Aztiria et al. [8] suggested a process to discover sequences of user actions.

Automatic modelling of user behaviour, by means of sequences of actions, although highly desirable is not available yet. There are tools that allow the definition of user behaviour manually but it requires too much effort and it is against the idea of Intelligent Environment of being unobtrusive, flexible and adaptive. An interesting exception is [9], where Fernández et al. use workflow mining techniques [10] to infer human behaviour models.

## IV. IDENTIFYING TOPOLOGY

The step 'Identifying Frequent Sets' discovers what sets of actions occur frequently together. In order to properly model user's behaviours, it is necessary to define the order of such actions. Representing the user's behaviours by means of ordered sequences of actions facilitate their understanding. Moreover, SPUBS provides a language [8] to formalize such patterns with the purpose of facilitating their use in tasks of prediction and automation of future actions. A previous step in our algorithm (see Section II) will have already discovered frequent sets of actions. Necessary data to discover the topology will be collected from the data coming from the Transformation Layer. Let us consider that in Michael's case, the step "Identifying Frequent Sets" discovered a frequent set with the following actions; *'Cupboard on', 'Cupboard off', 'Cup on', 'Fridge on', 'Fridge off', 'Milk on', 'Juice on', 'Coffee on' and 'Medicine Container on'*. Once the frequent sets of actions are identified, for each of them we collect the particular sequences where actions included in the frequent set happened, keeping the original order of actions. Let us consider some particular sequences in Michael's case following the *"time;device;status"* template:

2009-11-9, Monday (S1)
*08:16:42;Cupboard;on*
*08:16:44;Cup;on*
*08:17:01;Cupboard,off*
*08:17:50;Fridge,on*
*08:17:54;Milk,on*
*08:17:55;Juice,on*
*08:18:02;Fridge,off*
*08:19:24;Cupboard,on*
*08:19:27;Coffee,on*
*08:19:28;MedicineContainer;on*
*08:19:55;Cupboard,off*

2009-11-10, Tuesday (S2)
*08:28:41;Cupboard;on*
*08:28:44;Cup;on*
*08:28:46;Cupboard,off*
*08:30:57;Fridge,on*
*08:31:00;Juice,on*
*08:31:01;Milk,on*
*08:31:05;Fridge,off*
*08:33:08;Cupboard,on*
*08:33:10;Coffee,on*
*08:33:58;Cupboard,off*

2009-11-11, Wednesday (S3)
*08:49:29;Cupboard;on*
*08:49:30;Cup;on*
*08:49:54;Cupboard,off*
*08:50:26;Fridge,on*
*08:50:31;Milk,on*
*08:50:50;Fridge,off*
*08:52:08;Cupboard,on*
*08:52:12;Coffee,on*
*08:52:20;MedicineContainer;on*
*08:52:56;Cupboard,off*

2009-11-12, Thursday (S4)
*08:24:37;Cupboard;on*
*08:24:40;Cup;on*
*08:25:02;Cupboard,off*
*08:25:27;Fridge,on*
*08:25:30;Juice,on*
*08:25:32;Milk,on*
*08:25:58;Fridge,off*
*08:28:27;Cupboard,on*
*08:28:31;Coffee,on*
*08:29:12;Cupboard,off*

2009-11-13, Friday (S5)
*08:22:51;Cupboard;on*
*08:22:52;Cup;on*
*08:23:20;Cupboard,off*
*08:23:57;Fridge,on*
*08:24:02;Milk,on*
*08:24:30;Juice,on*
*08:24:31;Fridge,off*
*08:24:33;Cupboard,on*
*08:24:35;Coffee,on*
*08:24:50;MedicineContainer;on*
*08:24:51;Cupboard,off*

### A. Basic Methodology

As pointed out in Section III, few groups have dealt with this problematic in Intelligent Environments. Due to this fact, we have analysed other meaningful domains where user's actions have been used to extract models of behaviour. One of the closest domains we found has provided relevant results is the area of Workflow Mining [10-12], where they discover process models from event logs. Both domains are very similar except we will have actions of the user instead of event logs. Still, the objectives are very closely related in both domains. Even so, due to the nature of Intelligent Environments, we have considered that the Workflow Mining algorithm must be modified in various aspects. We explain these changes in the following subsections.

Due to the modifications we have carried out, the basic algorithm have also been modified. Thus, the only task of this first step is to represent the input data in a sequenced way, keeping all the actions as well as all relationships. Considering Michael's example, the output of this first step is represented in Figure 3.

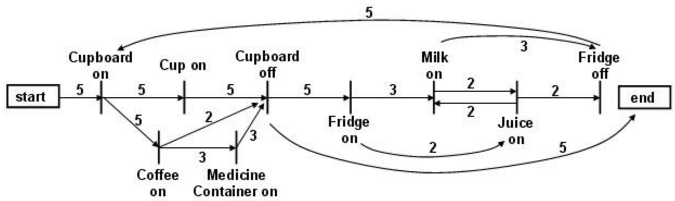


**Figure 3: Michael's sequence's basic representation**

The output of this first step shows a sequence where all actions as well as all relationships are represented. The numbers shown over the arrows indicate the frequency of each relationship. For example, the action 'Milk on' comes after 'Fridge on' three times (S1,S3 and S5), whereas 'Juice on' comes after 'Fridge on' twice (S2 and S4). Frequency of a relationship indicates the importance of that relationship in the behaviour of the user.

*B. Repetitive Actions*

Unlike other domains where an action is unique and there are no more than one occurrence of such an action per sequence, in Intelligent Environments there could be different occurrences of the same action. In fact, the nature of different occurrences will probably be different, i.e. the objective or the meaning of different instantiations will be different within the sequence. Thus, there could be repetitive actions in the same sequence. This is due to the fact that the user can do the same action with different purposes. An example of it is the action of 'Cupboard on' in Michael's sequence. He firstly opens the cupboard to get a cup, but after closing it, he opens it again to get some coffee and the medicine container. Considering the action 'Cupboard on' as unique make difficult the understanding of the sequence as it is.

Being aware of the possible existence of more than one instance of the same action, we have developed a methodology to:

- Decide how many instances of each action are necessary.
- Define the nature of particular sequences' actions.

If an action is repeated within a sequence it will probably be due to the fact the occurrences of that action are most likely to have different purposes (i.e. there are subtle variations in the context related to intentions in each of the instances). For example, in Michael's example the action 'Cupboard on' has the purpose of getting a cup in the first occurrence and getting coffee and the medicine container in the second one. We realized that seeing the action in context (e.g., by looking at the previous and next actions surrounding that action under analysis) provided us valuable information about the nature of a particular action.

A decision to make is the number of previous and next actions to consider. SPUBS, by means of a GUI, allows the user of the system to select from 1 to 4 actions to consider. The greater the number of actions we consider, more accurate the definition of the nature will be but of course that can have a negative impact in the time the process takes. Once the number of previous and next actions to be considered has been decided, SPUBS uses techniques of clustering to create groups. In that sense, SPUBS offer different options:

- To define manually the number of groups or clusters to create, considering a.) the average number of occurrences of an action; b.) the maximum number of occurrences of an action.
- To define automatically the number of groups or clusters using the EM algorithm [13].

Considering Michael's case, let us consider two previous and two next actions of particular instances of 'Cupboard on'.

| | Previous 2 | Previous 1 | Next 1 | Next 2 |
|---|---|---|---|---|
| 1) | ---- | ---- | Cup on | Cupboard off |
| 2) | Juice on | Fridge off | Coffee on | Medicine on |
| 3) | ---- | ---- | Cup on | Cupboard off |
| 4) | Milk on | Fridge off | Coffee on | Cupboard off |
| 5) | ---- | ---- | Cup on | Cupboard off |
| 6) | Milk on | Fridge off | Coffee on | Medicine on |
| 7) | ---- | ---- | Cup on | Cupboard off |
| 8) | Milk on | Fridge off | Coffee on | Cupboard off |
| 9) | ---- | ---- | Cup on | Cupboard off |
| 10) | Juice on | Fridge off | Coffee on | Medicine on |

It is clear that previous and next actions provide interesting information to create groups of particular instances with similar purposes. In Michael's case, all different options provided by SPUBS give the same result, where instances 1,3,5,7 & 9 create one group, whereas instances 2,4,6,8 & 10 create the second group. Thus, from this point on, there will be two different actions of 'Cupboard on'. 'Cupboard off' is another action which needs repetitive actions.

Taking into account the repetitive actions, Michael's sequence's representation is:

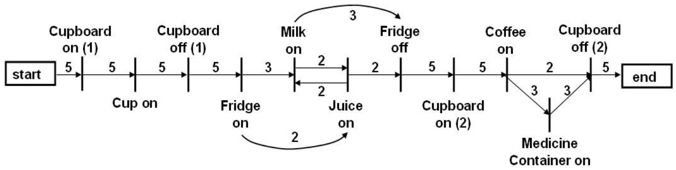


**Figure 4: Michael's sequence with repetitive actions**

*C. Unordered Subsets of Actions*

Different works of Workflow mining also propose the idea of parallel subsets of actions. The idea is to allow the system to represent those actions that occur simultaneously. In our system, the system of sensors is aimed at detecting those actions that occur at a specific moment, represented by means of a timestamp. The duration of an action can be extracted calculating the difference between the starting and finishing times.

As simple actions do not have duration, it is impossible the occurrence of parallel actions. Even so, the idea of parallel action can be applied to discover unordered subsets of actions. An unordered subset of actions represents a set of actions where it has not been possible to define a well-ordered sequence. This is typically the case when we consider actions which can be performed in various orders without affecting the final result. Michael's scenario offer us a typical example of such cases when he opens the fridge he sometimes takes the bottle of milk first and then the bottle of juice (S1 and S5) and some other times he first takes the bottle of juice (S2 and S4).

As in parallel actions of the Workflow mining case, the representation of unordered sets of actions shows bidirectional relationships between such actions (See the relationship between 'Milk on' and 'Juice on' in Figure 3). SUPBS includes a set of parameters that can be used to decide what bidirectional relationships must be considered as unordered sets of actions. These parameters are necessary due to the fact that the frequency of those relationships must be taken into account to decide if a bidirectional relationship represents an unordered set of actions or it does not.

Michael's behaviour, including unordered sets of actions (in this scenario there is only one unordered set, with 'Milk on' and 'Juice on' actions), will be represented as in Figure 5.

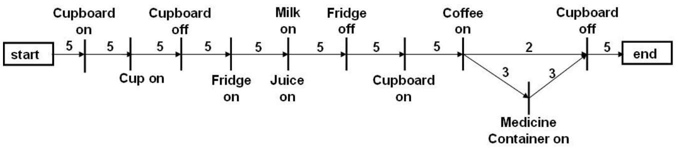


**Figure 5: Michael's sequence with unordered actions**

*D. Granularity and Allowed Maximum Granularity*

Once the behaviour of the user has been discovered and represented, SPUBS allows the user of the system to select the granularity of the sequence. The selected granularity indicates the threshold for the relationships. Thus, relationships with lower frequency than such a threshold will not be represented. Therefore, a high granularity will provide a more general representation of the sequence, whereas, a low granularity will provide a more complex sequence.

A selection of the granularity allows the user of the system to see the sequences with different complexities depending on his/her interests. It is clear that selecting granularity without any limit can result in illogical sequences, where actions are not related to any other action, or sequences where there is not a path from 'start' to 'end'. Michael's sequence can be an example of it if we select a granularity of '5'. In that case, Michael's sequence would be as shown in Figure 6.

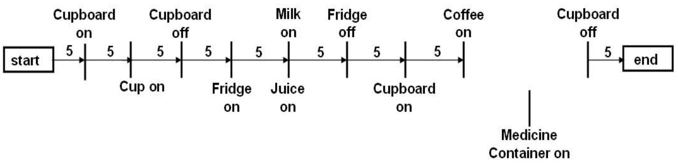


**Figure 6: Michael's sequence without Allowed Maximum Granularity parameter**

In order to avoid illogical representations, SPUBS includes a heuristic Uniform-cost search [14], which calculates the maximum granularity that allows, at least, one path from 'start' to 'end'. Thus, the system does not allow the user to select a higher granularity level than this parameter, making sure that sequences will maintain a minimum logic in all cases. In Michael's case, the 'Allowed Maximum Granularity' value is '3' and SPUBS would not allow higher granularity than 3. Michael's sequence's most general representation would be as shown in Figure 7.

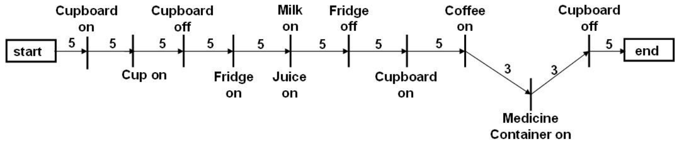


**Figure 7: Michael's sequence with Allowed Maximum Granularity parameter**

## V. FOLLOWING STEPS

It has been mentioned above that all steps of the SPUBS are already developed. Due to space restrictions is not possible to provide more details of the other steps. The aim of each step has been defined in Section II. This section briefly reinforces and clarifies the connection of the step described in the previous section with the rest of the process

Once the Frequent Sets and their topologies are discovered, the next step is to identify frequent time relationships among the actions (See Figure 2(b)). Different time distances are created taking into account their similarity. Thus, if there is any group which includes more instances than the demanded minimum level, the average value of that group is considered as a relevant value to define the time relation between those actions. SPUBS considers two types of time relationships, on one hand qualitative relationships (e.g., 'after'/'before') and on the other hand quantitative relationships (e.g., '2 seconds', '5 minutes'). The usefulness of each type of relation is different. Both help to characterize the user's habits, qualitative relation skeleton of a higher level activity whilst quantitative relationships can be used to automate interventions from the house to assist the user.

Finally, conditions are discovered (See Figure 2(c)) by using classification techniques. Conditions are of outmost importance when the topology indicates there is more than one option. We consider each option as a class and the sequences of that option as the instances of that class. Thus, first we collect the context (e.g. temperature) and temporal information (e.g. day of the week) of each of the instances and then we use that information to discover the conditions for each option.

## VI. RESULTS

The SPUBS has been validated by applying it to artificial data generated by the authors and then to real datasets collected from MavPad [15] and the WSU Smart Apartment [16]. The MavPad dataset was more relevant to the step 'Identifying Frequent Sets' as there were unknown frequent sets and that was a good test for the algorithm. Once frequent sets were discovered we identified the topology of such frequent sets. In that sense, it was possible to identify topologies as well as the parameter 'Allowed Maximum Granularity' for all the frequent sets.

For the step ‘Identifying Topology’, the dataset collected from WSU Smart Apartment was much more interesting. The data represented participants performing five ADL activities in the apartment (making a phone call, washing hands, cooking, eating and cleaning in that order). In our validation tests we have used data collected from 20 participants. All them carried out the activities in the same order so that we knew in advance the topology SPUBS should discover.

In those 5 activities there were involved 24 actions such as getting the medicine container, getting a pot, opening/closing a tap or switching on/off the burner. The ‘Identifying Frequent Sets’ step discovered 24 frequent actions. In order to define repetitive actions, we have considered four previous and four next actions, and the number of repetitive actions was defined by the average occurrence of that action. SPUBS discovered the need of defining repetitive actions of some actions such as opening/closing the tap or opening/closing the cabinet. This is because the participants needed water to carry out different activities such as cooking or cleaning. Once definitive actions were defined, the next step was to define the unordered subsets of actions. SPUBS discovered some subsets of unordered actions, mainly actions involved in the activity of cooking. For example, some participants added first sugar whereas some other ones added first raisins.

Once repetitive and unordered actions were defined, SPUBS represented the behaviour of the participants in a comprehensible (human-readable) way. SPUBS provided a topology where the 5 activities involved in the behaviour of the user were represented sequentially. Finally, SPUBS also calculated the value of the ‘Allowed Maximum Granularity’, which was ‘4’.

## VII. CONCLUSIONS

Intelligent Environments need to know the common behaviours and preferences of the user in order to perform meaningful interventions. Representing such common behaviours by means of total/partial orders facilitates their understanding as well as their use for other purposes such as the automation of actuation. Discovering models of user’s behaviour that capture the typical sequences of action defining everyday activities is a fundamental step. Representing such knowledge requires several considerations. More than one instantiation of actions are necessary to represent occurrences of the same action with different purposes. Unordered subset of actions representing sets of actions where it is not possible to define well-ordered sequences is normal. Finally, the system also needs to make sure there is a coherent representation of all possible interesting sequences.

## ACKNOWLEDGMENTS

This work was partially supported by Basque Government grant PC2008-28B.

## REFERENCES

[1] J.C. Augusto, “Ambient intelligence: The confluence of pervasive computing and artificial intelligence,” in Intelligent Computing Everywhere, A. Schuster, Ed. Springer, 2007, pp. 213–234.

[2] V.Callaghan, A. Kameas, A. Reyes, D. Royo and M. Weber, “Proceedings of the 5th International Conference on Intelligent Environments”, IOS Press, 2009.

[3] M. Galushka, D. Patterson, and N. Rooney, “Temporal data mining for smart homes,” ser. Designing Smart Homes. The Role of Artificial Intelligence, J. C. Augusto and C. D. Nugent, Eds. Springer- Verlag, 2006, pp. 85–108.

[4] A. Aztiria, A. Izaguirre, R. Basagoiti, J.C. Augusto, and D.J. Cook, “Discovering of Frequent Sets of Actions in Intelligent Environments” Proceedings of the 5th International Conference on Intelligent Environments, 2009 pp 153-160.

[5] A. Aztiria, A. Izaguirre, J.C. Augusto “Learning patterns in Ambient Intelligence environments: A Survey”. Artificial Intelligence Review (accepted)

[6] 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, 2007, pp. 625–630.

[7] E. Heierman, M. Youngblood, and D. J. Cook, “Mining temporal sequences to discover interesting patterns,” in KDD Workshop on Mining Temporal and Sequential Data, 2004.

[8] A. Aztiria, A. Izaguirre, R. Basagoiti, J.C. Augusto, “Learning about preferences and common behaviours of the user in an intelligent environment”, Behaviour Monitoring and Interpretation – BMI - Smart Environments, “Ambient Intelligence and Smart Environments” book series, ed. Björn Gottfried, Hamid Aghajan, V.3, 2009, pp. 289-315.

[9] C. Fernández, J.P. Lázaro, and J.M. Benedí, “Workflow Mining Application to Ambient Intelligence Behavior Modeling”. In Proceedings of the 5th international on Conferenceuniversal Access in Human-Computer interaction. Part Ii: intelligent and Ubiquitous interaction Environments (San Diego, CA, July 19 - 24, 2009). C. Stephanidis, Ed. Lecture Notes In Computer Science, vol. 5615. Springer-Verlag, Berlin, Heidelberg, 2009, 160-167.

[10] W. van der Aalst, A. Weitjers, and L. Maruster, “Workflow mining discovering process models from event logs,” IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 9, 2004, pp. 1128–1142.

[11] L.Wen,W. van der Aalst, J.Wang, and J. Sun, “Mining process models with non-free-choice constructs,” *Data Mining and Knowledge Discovery*, vol. 15, no. 2, 2007, pp. 145–180.

[12] G. M. Youngblood, D. J. Cook, and L. B. Holder, “Managing adaptive versatile environments,” in IEEE International Conference on Pervasive Computing and Communications, 2005.

[13] I.H. Witten, and E. Frank, “Data Mining: Practical Machine Learning Tools and Techniques”. 2nd ed. Elsevier, 2005

[14] R. Stuart, P. Norvig, “Artificial Intelligence: A modern appraoch”. 2nd ed. Prentice Hall, 2003.

[15] G. M. Youngblood, D. J. Cook, and L. B. Holder, “Managing adaptive versatile environments,” in IEEE International Conference on Pervasive Computing and Communications, 2005.

[16] D. Cook and M. Schmitter-Edgecombe. “Activity profiling using pervasive sensing in smart homes”. IEEE Transactions on Information Technology for Biomedicine, 2008