2012 Eighth International Conference on Intelligent Environments

A System to Learn Frequent Behavioural Patterns

Asier Aztiria, Rosa Basagoiti
Electronics and Computing Dep.
Mondragon University
Mondragon, Spain

Juan Carlos Augusto
Sch. of Computing and Mathematics
University of Ulster
Belfast, U.K.

Gonzalo Espejo
Dep. of Languages and Computer Systems
University of Granada
Granada, Spain

***Abstract*— Intelligent Environments (IEs) are expected to support people in their daily lives. To achieve this goal, the environment should learn how to react to the actions and needs of the users, and this should be achieved in an unobtrusive and transparent way. In order to provide personalized and adapted services, it is necessary to know the preferences and habits of users. Thus, the ability to learn patterns of behaviour becomes an essential aspect for the successful implementation of IEs. This paper presents a system, Learning Frequent Patterns of User Behaviour System (LFPUBS), that discovers user's frequent behaviours. LFPUBS was validated using data collected from real environments.**

Keywords- Frequent behaviours, Machine learning techniques, Intelligent Environments.

I. INTRODUCTION

Intelligent Environments (IEs) 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. 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 references, needs and habits of the user in order to be in a better position to assist the user adequately. Knowing users' frequent behaviours allows the environment to act intelligently and proactively. Knowledge extracted from these patterns can also be used in order to understand user's behaviour. For example, the analysis of frequent interaction with objects and devices in the house can facilitate the detection of unhealthy habits. Making the environment more efficient in terms of *saving energy* (e.g. by turning off the lights when the user leaves) or *increasing safety* (e.g. turning off the water or issuing alarms when detecting that the user 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.

One of the main characteristics of IEs is the key role that the user plays as the focus of the entire process. In other words, the process starts by collecting data about the user and the environment in which the user is situated, and it finishes by acting intelligently for the user. In order to achieve these objectives, a software, called Learning Frequent Patterns of User Behaviour System (LFPUBS), has been developed, which allows an Intelligent Environment to discover frequent behavioural patterns. This paper explains the basic components of LFPUBS and how to use the system.

II. LEARNING FREQUENT PATTERNS OF USER BEHAVIOUR

Learning Frequent Patterns of User Behaviour System (LFPUBS) 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 A_{LFPUBS}, which combined with a language L_{LFPUBS}, 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.

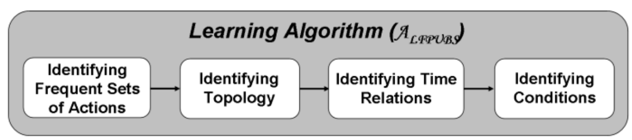

Figure 1: Algorithm of LFPUBS

Developed algorithm allows to be used in different ways. On the one hand, it can be embedded within other systems so that their functionalities can be used adding the LFPUBS as an extension (.jar) of new systems. On the other hand, it can be used by means of a Graphical User Interface (GUI) that allows us to execute different steps of the algorithm step by step.

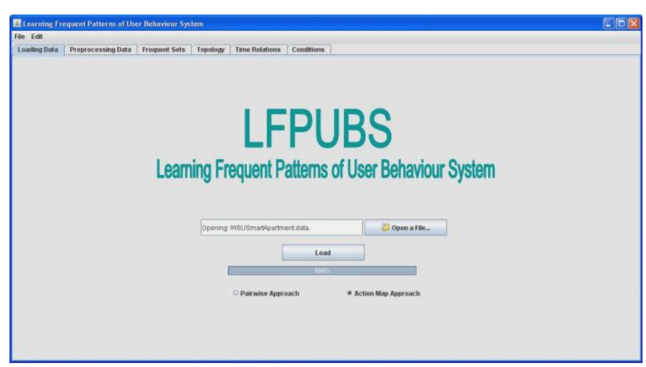

Figure 2: Initial screen of the GUI

All the parameters necessary to carry out the learning process, and defined below, can be defined by means of the GUI. Following, each step of the process, along with its correspondent GUI screenshot, will be explained.

978-0-7695-4741-1/12 $26.00 © 2012 IEEE
DOI 10.1109/IE.2012.33

A. *Identifying Frequent Sets of Actions*

The objective of this step is to discover the sets of actions that frequently occur together (Frequent Sets). The underlying idea of the first step is both simple and efficient. Defining a demanded minimum level (minimum confidence level), it discovers all those sets of actions that occur more times than the minimum level. These sets of actions are treated as Frequent Sets. To discover Frequent Sets in large amounts of data, the Apriori algorithm [2] was used. As shown in Figure 3, LFPUBS allows the definition of the parameters considered in order to discover Frequent Sets.

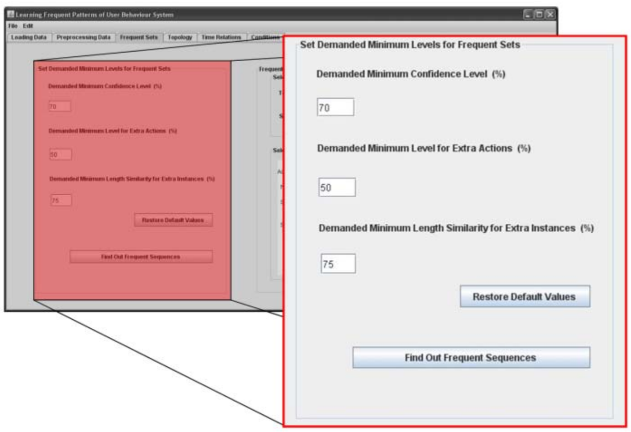

Figure 3: Defining parameters for Frequent Sets.

B. *Identifying Topology*

The step ‘Identifying Frequent Sets’ discovers which sets of actions frequently occur together. In order to properly model the user’s behaviours defined by such sets of actions, it is necessary to define the order of such actions. The goal of this step is to discover the frequent order (defined as Topology) of the actions in the behaviour of the user. Few groups have dealt with this problem in IEs, so that, other meaningful domains in which user’s actions have been used to extract models of behaviour have been analysed. In that sense, one of the closest domains is the area of Workflow Mining [3] in which process models are discovered from event logs. Both domains are equal, with the only difference being that instead of event logs, LFPUBS considers the actions of the user. Even so, because of the nature of IEs, some particularities must be taken into account.

First of all, considering the actions involved in the Frequent Set, all the relations defined by the data are represented. The objective is to define the initial probabilities, $P_0 = \Pr(fi_0)$ and the transition probabilities for each relationship $P = [P_{k,j}]$ where $P_{k,j} = \Pr(f_{i,j} | f_{i,k})$. Although this step does not discover anything, at this point, it is important to highlight that it provides the first formal representation of the behaviour.

Repetitive Actions

Unlike other domains in which an action is unique and there is no more than one occurrence of each action in a pattern, in IEs, there could be different occurrences of the same action. In fact, the nature of repetitive occurrences will probably be different because the user can do the same action with different purposes. Thus, it is necessary to identify repetitive actions and create different instantiations of them.

Considering the possible existence of more than one instance of the same action, a methodology to automatically discover such situations has been developed. It is based on the idea that the meaning, and by extension, the nature of an action is mainly defined by the previous and next actions. In other words, the occurrence of an action is related to the previous and next actions because the set of those actions will probably follow a specific objective. Thus, the nature of different actions is defined by creating groups of actions that take into account the similarities among the previous and next actions of their occurrences. LFPUBS includes two different techniques to create groups:

- Manually define the number of groups or clusters to create, considering a.) the average number of occurrences of an action in a pattern and b.) the maximum number of occurrences of an action in a pattern.
- Automatically define the number of groups or clusters using the EM algorithm [4].

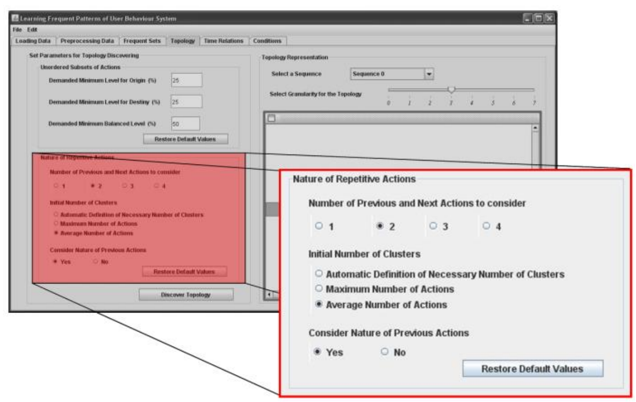

Figure 4: Defining parameters for repetitive actions.

Unordered Subsets of Actions

Different works of Workflow mining also suggested the idea of parallel subsets of actions. The same idea can be applied to discover unordered subsets of actions. An unordered subset of actions represents a set of actions in which it has not been possible to define an order for such actions. As in the parallel actions of Workflow mining cases, the representation of unordered set of actions shows bidirectional relationships between such actions. To decide whether a bidirectional relationship (let us say between A and

B) must be considered as an unordered set of actions, LFPUBS includes a set of parameters. The GUI allows the definition of such parameters:

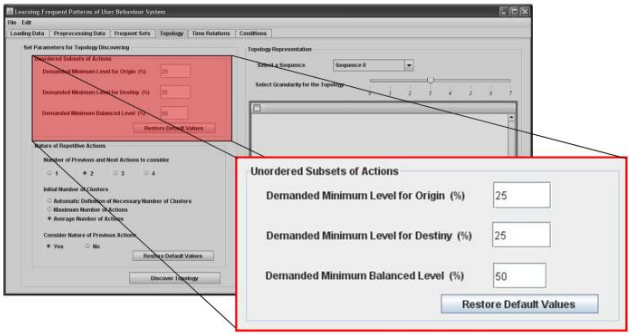

Figure 5: Defining parameters for unordered actions.

C. Identifying Time Relations

It is clear that the Topology defines a first temporal representation of the frequent behaviour by means of qualitative relations (using the term 'after') and their probabilities. The objective of this step is to discover frequent quantitative Time Relations between actions. For that purpose, LFPUBS includes two different algorithms - the 'Basic Algorithm' and the 'EM Algorithm' - so that the user of the system may choose either of them to identify such quantitative relations. Both algorithms are based on the same idea of grouping occurrences by taking into account their similarity and deciding whether a group represents a quantitative Time Relation. For more details about these algorithms see [5].

D. Identifying Conditions

Once Topology and Time Relations have been identified, behaviours are represented in a comprehensible way. Even so, a final step that identifies Conditions for each behaviour is necessary in order to create accurate representations of the behaviours of the user.

On the one hand, conditions are needed when one action is followed by two (or more) different actions, i.e. sometimes the user does one thing whereas some other times he/she does a different action. In those cases, it is necessary to identify under what conditions each of those relations is true. For that, for each possible relation a table is created. In each table the occurrences covered by that relation are collected, together with the calendar and context information collected when such occurrences happened. Once the tables are created, separating both tables by using the information they contain allows one to discover conditions. In that sense, the task of separating can be solved by treating it as a classification problem. The JRip Algorithm [6] was used to accomplish this task.

On the other hand, it is necessary to define the general context in which a behaviour occurs. General Conditions refer to calendar and context information that allows the user of the system to understand under what conditions the whole behaviour occurs. In this research work, only calendar information ('Time of Day' and 'Day of Week') has been considered. In order to identify General Conditions, defined by the terms 'Time of Day' and 'Day of Week', a very basic strategy has been adopted. Such a strategy is based on covering all the occurrences. Thus, all occurrences are covered by the range defined by 'Time of Day' and 'Day of Week' terms.

III. USING PATTERNS TO INTERACT WITH THE USER

Once pieces of knowledge about users' frequent behaviours have been learned, they can be used for different purposes. An important aspect of IEs has to do with their interaction with users, a key element in the process of efficiently applying the extracted knowledge. Given the importance of users for the success of an IE, it is essential that there be a friendly and easy way for the user to interact with the environment. As a first approach, a speech-based HCI system has been developed. The goal of this system was to allow users to give their feedback about discovered behaviours before using these patterns of behaviour to automate the activation/deactivation of devices.

Different environments and different objectives require the development of different interaction systems. The methodology to develop an HCI system where users' frequent behaviours are involved is the same in all applications. Depending on the nature and the objectives of each environment, it will be necessary to modify the possible questions as well as the options given to the user, but the technology will remain untouched in all the applications. In this case, the chosen speech synthesizer has been FreeTTS 1.2 while Sphinx- 4 has been chosen as the speech recognizer. For more information see [7].

IV. VALIDATION

LFPUBS has been validated using datasets collected from many real environments.

A. Using LFPUBS in Smart Offices

A dataset has been collected in an office of the University of Ulster, which has been equipped with different sensors. Installed sensors are mainly motion sensors, as well as sensors to monitor doors and lamps.

Figure 6: ZWave equipment. Clockwise, from top left: PIR, door switch, dimmer, control box.

Once data were collected from sensors, LFPUBS was applied in order to discover frequent behaviours. As expected LFPUBS was able to discover recognizable behaviour. The most representative one showed how the user of the smart room switched on and off the lamp of his desk when he used to go in and out the room. Although it seems that this pattern does not represent any new knowledge, LFPBUS was also able to discover the time relations between the actions. Thus, it was discovered that the user, on average, switched on the lamp 8 seconds after he went into the office. The representation of the behaviour as well as the time relation between actions can be seen in Figure 7.

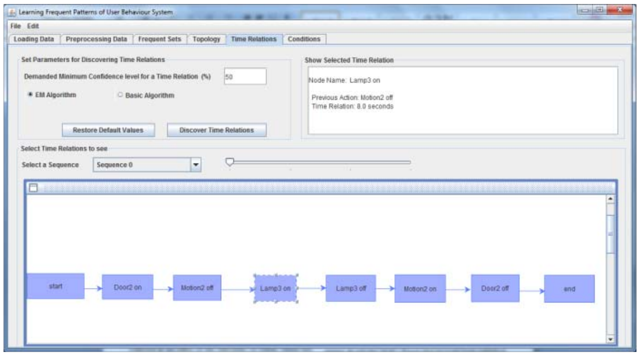

Figure 7: Pattern representing a frequent behaviour.

B. Using LFPUBS in Smart Homes

LFPUBS was also used in order to discover frequent behaviours in smart homes [8]. Data collected in the WSU Smart Apartment represented participants performing the same five ADLs (Activities of Daily Living) in the apartment, so the frequent behaviours that the LFPUBS should discover were known in advance. The actions involved in each one of the activities are shown in Table 1.

Table 1: Actions involved in each ADL

Activity	Actions Involved
Make a phone call	'PhoneBook on' → 'Phone on' → 'Phone off'
Wash hands	'Water On' →'Water Off'
Cook	'Cabinet On'→'Raisins On'→'Oatmeal On' →'MeasuringSpoon On'→'Bowl On' →'Sugar On'→'Cabinet Off'→'Water On' →'Water Off'→'Pot On'→'Burner On' →'Burner Off'
Eat	'Cabinet On'→'Medicine On'→'Cabinet Off' →'Water On'→'Water Off'→'Cabinet On' →'Medicine Off'→'Cabinet Off'
Clean	'Water On' →'Water Off'

In this case, due to the fact it was known the behaviour carried out by the participants in advance, this validation was an acid test for the steps of 'Identifying Frequent Sets of Actions' and 'Identifying Topology' that had to identify and model such a behaviour.

The step of 'Identifying Frequent Sets of Actions' was able to discover the actions involved in the pattern (specified in Table 1), considering 60% as the minimum parameter. Then, the 'Identifying Topology' step identified the repetitive actions as well as the unordered actions. For example, the actions 'Water On' and 'Water Off' were involved in activities such as 'Wash hands', 'Cook', 'Eat' and 'Clean'. The nature and the purpose of such actions in each one of the activities is different; therefore, identifying repetitive actions was an important step to correctly model users' behaviours. In the case of the actions 'Water On' and 'Water Off' LFPUBS was able to define that four different 'Water On' and 'Water Off' actions were needed. When it comes to unordered actions, LFPUBS discovered that some of the users took out the raisins first and then the oatmeal, and others did the opposite.

Although the topology itself defined the qualitative Time Relations, quantitative Time Relations were discovered using the 'Basic Algorithm'. Considering all the relations defined by the Topology, the 'Basic Algorithm' was able to identify quantitative Time Relations in 25 out of 29 (86%) cases. Finally, Specific and General Conditions were identified. Regarding the Specific Conditions, it is true that very few situations demanded Specific Conditions (only three). It is worth noting that using only calendar information, it was possible to identify conditions in two out of three (67%) cases. The identified General Conditions indicated when the participants performed such actions. Thus, it was discovered that all of the actions were carried out on weekdays between 10:45 a.m. and 18:15 p.m.

V. CONCLUSIONS AND FUTURE WORKS

One of the main assumptions of IEs is that the environment must adapt itself to the needs and preferences of users. In order to achieve real adaptive environments, a system that learns user's frequent behaviours has been developed. This system can be either embedded into bigger systems or used by means of a GUI. This GUI allows the user of the system to define the necessary parameters to carry out the process.

Future work includes extending the system in order to consider other types of information such as health and emotional status of the user, as well as including the online adaptation of the patterns within LFPUBS.

REFERENCES

[1] J. C. Augusto, Ambient Intelligence: the Confluence of Ubiquitous/ Pervasive Computing and Artificial Intelligence, ser. Intelligent Computing Everywhere. Springer London, 2007, pp. 213–234.

[2] R. Agrawal and R. Srikant, "Mining sequential patterns," in Pro. 11th International Conference on Data Engineering, 1995, pp. 3–14.

[3] 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, pp. 1128–1142, 2004.

[4] R. Hogg, J. McKean, and A. Craig, Introduction to Mathematical Statistics. Pearson Prentice Hall, 2005, pp. 359–364.

[5] A. Aztiria, J.C. Augusto, R. Basagoiti and A. Izaguirre "Accurate Temporal Relationships in Sequences of User Behaviours in Intelligent Environments". Int. Symp. on Ambient Intelligence (ISAmI), 2010.

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

[7] A. Aztiria, A. Izaguirre, R. Basagoiti, and J. Augusto, Learning about preferences and common behaviours of the user in an intelligent environment, ser. Behaviour Monitoring and Interpretation-BMI-Smart environments, AISE Series. IOS Press, 2009, pp. 289–315.

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