A Bayesian Network and Rule-base Approach Towards Activity Inference

Venet Osmani, Sasitharan Balasubramaniam, Dmitri Botvich
Telecommunications Software and Systems Group
Waterford Institute of Technology
Waterford, Ireland
{vosmani, sasib, dbotvich} @tssg.org

Abstract—**In this paper we describe an activity recognition system capable of monitoring user activities in home environments. Activities are monitored by processing information from various sensors embedded in the environment that provide information pertinent to user's actions. We utilise the concept of self-organising Object Networks to gather and hierarchically process information related to user actions in a distributed manner. This information is then fed to the Decision Module which matches this information in the user's Activity Map in order to deduce user's activity. The Decision Module comprises a Bayesian Network coupled with a rule-based engine which is used to provide accurate activity inference process.**

Keywords-component; Bayesian networks; activity recognition; activity inference;

I. INTRODUCTION

More than ever the importance of leading a healthy lifestyle is being emphasised with various campaigns aimed at encouraging healthy eating and exercise. One important aspect of increasing physical activity is the ability to monitor activity levels. Monitoring and correcting activity levels, coupled with other factors such as proper diet, provide a clear picture of one's overall health state and can also serve as an encouragement tool to further physical activity levels. Physical activity is not limited to intense exercises only, rather its definition has been extended to common household activities [1] which is the focus of this paper. Therefore, automatically recognising user activities is an important aspect of home activity monitoring and a specialised research field named Activity Recognition is concerned with this issue. Activity Recognition is a research trend in Pervasive Computing that studies the ability of computing devices to monitor the user and the environment, and infer user's activities based on events triggered by user's actions. By observing the user's actions, much can be learned about user's behavioural patterns and anticipation of future actions.

This paper describes an Activity Recognition system capable of inferring user activities based on the user's interactions with various objects in the environment. The objects within the environment include various devices, and we envisage an environment embedded with different sensors. In addition, sensors are also embedded in devices in the living environment such as floor sensors, appliances' or utensils' sensors or in any other device that the user may manipulate. Processing information generated from these sensors provides an understanding of users' actions. However, information generated as a result of user actions can be potentially vast thus creating a management challenge. To counter this issue we have taken a multi-level hierarchical activity inference process that is distributed and supports self-organisation of various objects into an emergent object network structure. Since user actions are typically recorded through primitive sensors our hierarchical Object Network structure allows processing of low-level information by recursively composing the low-level data into higher level abstractions of information. This information is then utilised by our Decision Module that has the final responsibility of inferring users' activities. Decision Module can infer a number of common household activities; however, the system can be easily extended to recognise a large number of diverse activities. In this paper we particularly focus on the Decision Module and its ability to receive events from the object network and infer user activities using Bayesian Network as well as rule engine.

The rest of this paper is organised as follows. Section II describes the most relevant work in the area of activity recognition while section III describes the overall architecture and its constituents. Section IV provides results pertaining to our activity recognition, while section V draws the concluding remarks.

II. RELATED WORK

A number of research projects have observed and recorded information pertaining to user actions or mobility such that user activity inference can take place. Monitoring user activities through a vanilla mobile phone has been a research focus of [2]. The authors rely on GSM cells' signal fluctuations which are fed into a neural network to infer three user mobility states, namely stationary, walking and driving. While the approach is appealing, authors had encountered accuracy problems. This is mainly due to the fact that GSM cells tend to have a relatively low granularity thus providing coarse user mobility information. Another project with similar aspiration involving data from a GPS device coupled with Bayesian filtering has been presented in [3]. Authors infer three modes of transportation, namely bus ride, walking on foot and taking a car. The prediction accuracy was relatively low, which prompted authors to augment the prediction model with bus routes. Bao and Intille in [4] propose recognising human

1-4244-0264-6/07/$25.00 ©2007 IEEE

activities based on accelerometers. Authors report recognition accuracy up to 95%. However, their approach, based solely on accelerometers only, limits the number of activities the system can recognise. In [5] an RFID reader mounted on hand glove records information about objects being manipulated by a user and this information is fed to an activity inference engine. A model of activities is obtained through web data mining techniques especially mining the *how-to* websites. While the authors report positive results, there are two disadvantages to this approach; the inconvenience of wearing a glove and the centralised architecture design. While the first problem can be somewhat alleviated considering the technology trends in miniaturisation (authors report working on an RFID bracelet to replace the glove [6]) the second problem poses a greater challenge for scalability.

This critique of the reviewed systems has shown that each is lacking one or more features in order to create a robust and accurate activity recognition system. Majority of these systems rely on a single technology. Such reliance decreases the richness of information generated as a result of user behaviour which limits the number of activities that can be recognised. Our design presented in the following sections aims to alleviate these issues by employing a number of concepts that are mapped to various components of the architecture.

III. ARCHITECTURE

Our architecture depicted in Figure 1 outlines the main components, namely the Object Networks, hierarchical structures that provide an efficient processing platform of information generated as a result of user actions, Activity Map (AM) that acts as a repository of user's activities and Decision Module (DM) that works in conjunction with AM in order to infer user's activities based on the events filtered through the object network.

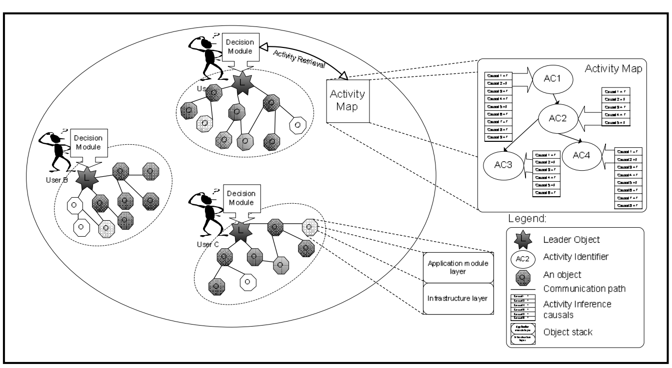

Figure 1. Overall Architecture

Our aim is to create a decentralised activity inference architecture that can efficiently and accurately infer user's activities. The process of activity inference is engaged when a user enters a particular domain and his/her mobility stabilises at a particular location. At this point, devices surrounding the user self-organise into a hierarchical object network structure that observes user's actions. Information pertaining to user's actions is processed throughout the hierarchy, up to the DM that infers user's activities by drawing out a generic set of events that contribute to an activity from a centralised repository – the Activity Map.

A. *Object Networks and Self-organisation mechanism*

Object networks interconnect various devices that can provide information pertinent to user's actions. An object network is an overlay network, inspired from the functionalities of sensor networks that have the ability to self-organise through peer to peer interactions. However, our concept of object networks extends beyond sensor networks; an object network includes various devices with different capabilities and functionalities. An object is any artefact that may provide information relevant to the activity inference process with a particular focus on the objects that user may interact with (e.g. electronic devices, appliances, utensils).

Employing the concept of object networks has two primary purposes. Firstly, object networks provide an efficient processing platform for context information generated from low-level sensors as a result of user actions dictated by a specific behaviour. This is achieved through the hierarchical structure created as a result of the self-organisation process where no centralised control exists. Secondly, object networks significantly lessen the gap that exists between the low-level information generated from various sensors and the ability of this information to represent high-level events such as user's goals or actions. This is achieved through increasing the level of abstraction of context information whereby low-level information is processed, filtered and composed throughout the object network hierarchical structure. Increasing the level of abstraction of context information directly benefits the activity inference process, since DM is shielded from the low-level details associated with raw information generated from low-level sensors (such as data sampling, data representation, composition, filtering). This also has the added benefit of making the inference process more efficient due to processing of lower volume of information in comparison with vast amount of raw data generated from a large number of sensors. The hierarchical processing structure is created automatically based on local object to object interaction rules. Typically low-level sensors sit at the bottom of the structure, while the *leader object*, chosen as a result of an election process takes the highest position at the top of the structure (see Figure 1). An object network 'follows' the user as s/he changes the position within a domain. Such behaviour stems from the fact that user activities are highly localised and typically involve manipulation of objects in the close proximity.

Object networks self-organise into Context Zones and as such form an emergent hierarchical structure. Self-organisation process is dictated by local object to object interactions governed by the information dependencies between objects. The process of self-organisation, including a formal description of Context Zones and their creation is detailed in [7].

B. *Decision Module*

Decision Module depicted in Figure 2 is housed in the leader object and represents the last stage of the inference process. The main components of the Decision Module are the Activity Map and the Rule Engine module. The overall activity inference and refinement process is determined by a number of

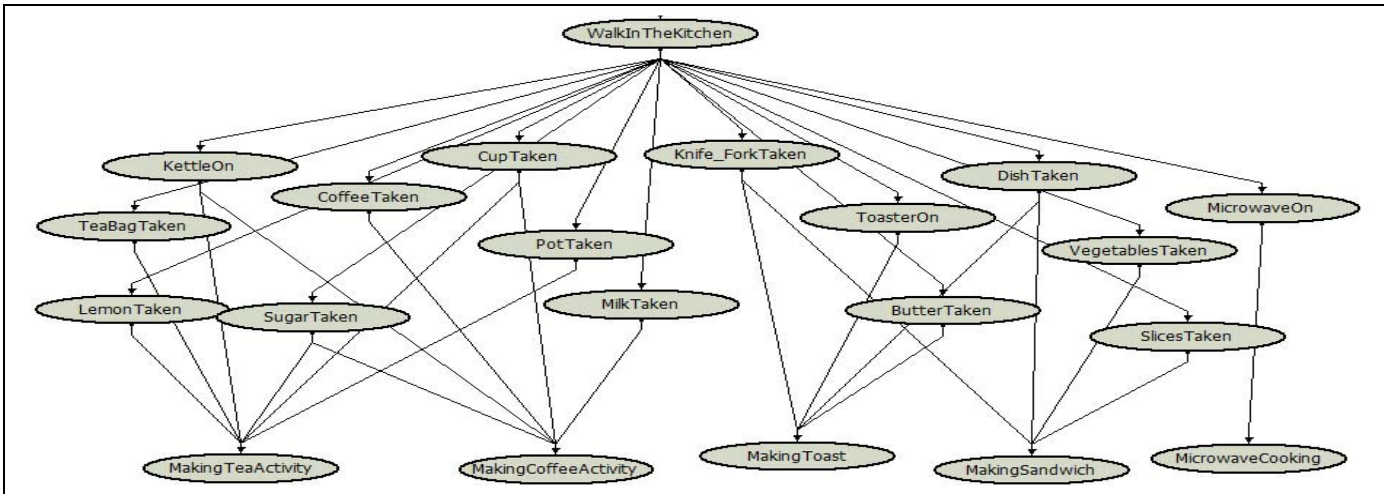

Diagram A. Part of user's AM

sub-modules that are organised within the *Rule Engine module*. The main task of the Decision Module is to evaluate events from the object network and match them to the Activity Map in order to infer user's activity. The AM stores activities that a user has performed in addition to representing relationships between each activity and its *causals*. Causals are events from the object network that represent the facts that have to be true for an activity to take place. An example of the Activity Map is illustrated in Diagram A. AM contains the knowledge about user activities and thus is specific to a user. The internal structure of an AM corresponds to a directed graph where each arc is assigned a probability value. This essentially forms a Dynamic Bayesian Network (DBN). Information generated in the object network is regarded as Bayesian evidence that is constantly fed to the DBN to infer a set of likely activities, where we apply the Junction Tree Algorithm. However, the DBN has to be refined in order to adapt to the changing user behaviour. The AM is stored within user's environment and as soon as the user enters a domain a subset of AM is retrieved that details the activities that can be performed within the user's current domain. A user entering a domain initiates the creation of an object network surrounding the user. The leader object then uses ***Activity Map download*** module from the Rule Engine to determine the subset of AM that corresponds to the user's domain. Once the subset of AM has been downloaded, the Decision Module begins receiving events from the object network.

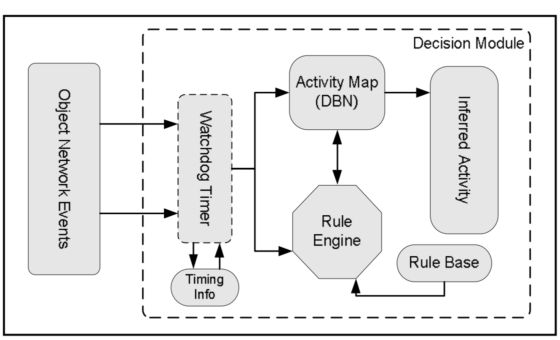

Figure 2. Decision Module

During the refinement phase, object network events are organised into user behaviour sequences that typically differ from each other, despite representing the same activity. Thus the ***Sequence Extraction*** module from the Rule Engine is used to obtain the most likely sequence, which represents the most common behaviour of a particular user for an activity. Most likely sequence is then used to refine the DBN. A sequence represents an *ordered set* of objects that user has manipulated during the course of carrying out an activity. A particular user may produce different sequences that represent the same activity. The differences in these sequences reflect the variations in user's behaviour. An example sequence of a particular user may be KettleOn, CupTaken, CoffeeTaken, MilkTaken, SugarTaken which corresponds to Making Coffee activity (see Diagram A).

From a set of obtained sequences we then select the most likely sequence which is the one that has occurred most frequently. The most likely sequence is used to extract the activity causals – typically the objects that the user has manipulated – and use these causals to refine the DBN. The refinement process involves changing the conditional probability of causals with respect to a specific activity. For each causal in the most likely sequence, we update the conditional probability of that causal in relation to the activity the causal is part of. The refinement process thus allows DBN to dynamically adapt to user behaviour changes as we show in the results section. The Decision Module uses ***DBN Refinement*** module from the Rule Engine to update the conditional probability of the causals represented in the most likely sequence.

In order to ensure an accurate activity inference process, the Decision Module uses ***Watchdog Timer*** *module* to detect and avoid sporadic events. These events occur when user intends to carry out an activity, however his/her attention focus swiftly changes, for instance due to an event requiring attention (e.g. telephone ring). In such instances we temporarily suspend the inference process and attempt to deduce the new activity the user has engaged in.

Orchestration of these modules and their utilisation at different stages of DBN refinement and inference process provide for an accurate activity recognition process due to continuous user behaviour adaptation and also ensure that the inference process is not carried out unnecessarily.

IV. Simulation Results

For our simulation setup we have analysed a number of home activities and recorded each activity's details. We then used this information to create a behaviour model for a user. We performed two sets of tests, which include tests to evaluate the activity recognition, and tests for evaluating the same activities under altered user behaviours.

A. Activity Recognition Accuracy

Modelling user behaviour is realised using a Markov Chain model. An example for a Markov Chain model for making coffee activity is illustrated in Figure 3 Markov Chain models have been used to model user behaviour in a number of research projects [8, 9] since their probabilistic nature allows capturing the dynamics of user behaviour. Once setup, the Markov Chain user behaviour model is fed to the Test Agent (TA). The TA is responsible for extracting a sequence from the Markov Chain model through the *Sequence Extraction* module which represents a set of objects that user manipulates during the course of performing an activity. Due to the probabilistic nature of the model, the set of sequences obtained will each differ reflecting the manner in which humans conduct activities – variation is almost always present. Obtained sequence from the TA is fed to the Decision Module and reflects a particular way in which the user performs an activity. An important advantage of this approach is that the simulation process is entirely transparent to the Decision Module, since TA generates events in the same manner as they would have been generated should a real user perform a particular activity.

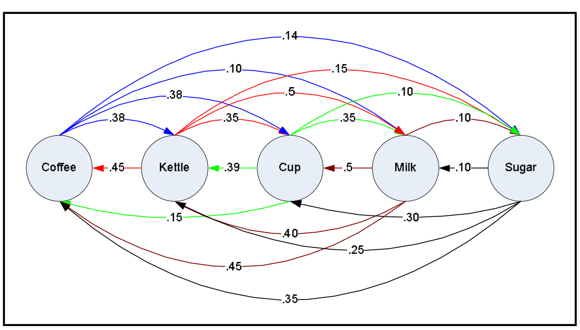

Figure 3. Part of the Markov Chain model of user behaviour

Our simulation setup consists of four activities performed at home. The number of iterations for training the DBN for each activity is set to 20 – equivalent to the user repeating a particular activity. The iteration step is set to 4 indicating that for every 4 sequences obtained we calculate the most likely sequence that is used to refine the DBN. Initially the DBN structure contains causals that have neutral impact, thus $P(A \mid c = T)$ is set to 0.5 (A represents an activity and c is a causal of that activity). However, the refinement process adjusts this value for each causal in the most likely sequence. The results for the four activities we have simulated are shown in Figure 4, 5, 6, and 7. As it can be seen from the results some of the activities have refined very well while in others, the DBN at instances was confused. For example Making Sandwich and Making Toast activity have had a steady increase in the inference probability. However, in Making Coffee and Making Tea activity the refinement process had a number of spikes. These are evident in Figure 4, Refinement 1 with causals 3 and 4 whereby the inference probability value had a fall from 0.575 to 0.5. Also a similar pattern can be observed in Figure 7, whereby Refinement 3 in causals 2 and 3 had a drop in the inference probability from 0.595 to 0.53. This was unexpected since each causal should increase the inference probability. The main reason for this behaviour lies in the number of causals for individual activities. While Making Sandwich and Making Toast activities only had 4 causals, the other two activities Making Coffee and Making Tea activities had 5 and 6 causals respectively. While this increase in number of causals may not be large, it becomes significant when considering the number of combinations. For example in Making Tea activity, the user had 6! = 720 unique ways of performing this activity compared to only 4! = 24 for Making Toast and Making Sandwich activities. Such large variation tended to confuse the DBN at times, hence the spikes in the refinement graphs.

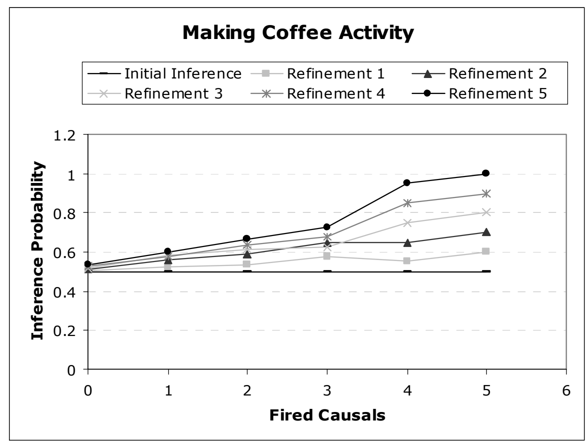

Figure 4. Making Coffee activity refinement

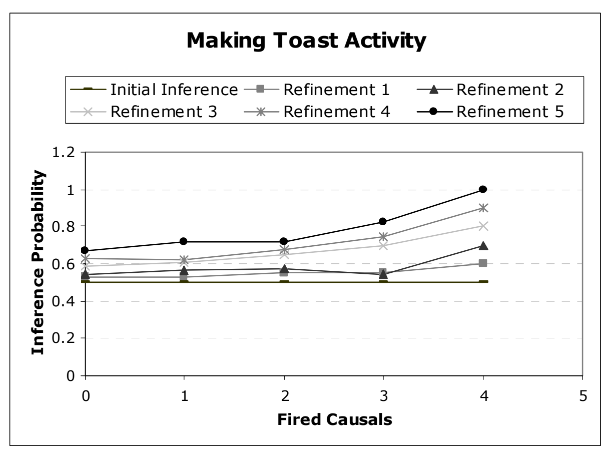

Figure 5. Making Toast activity refinement

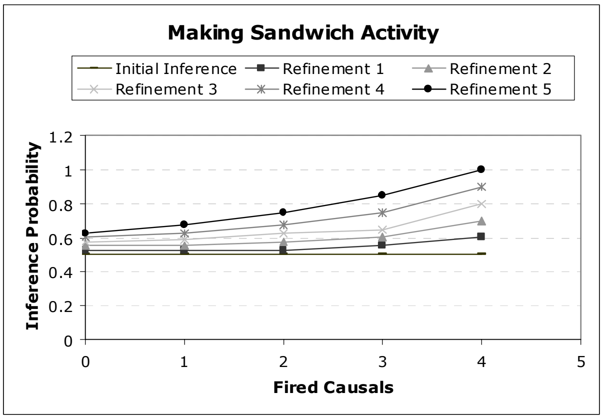

Figure 6. Making Sandwich activity refinement

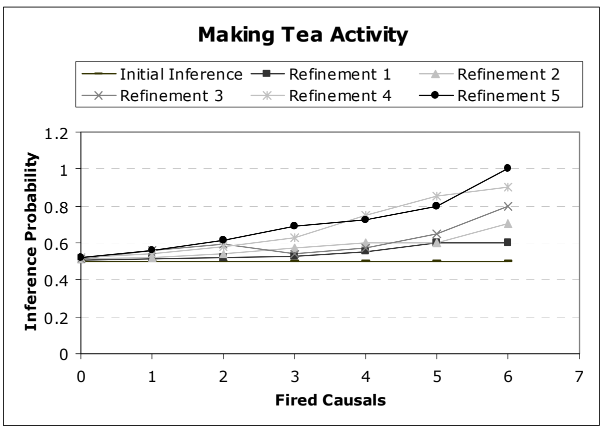

Figure 7. Making Tea activity refinement

B. Activity Recogntion accuracy for altered behaviour

We then changed the user behaviour model to measure the performance of the DBN in adapting to new behaviour for two activities namely making sandwich and making coffee activity. Even though the alternative user behaviour, as expected, generated a different DBN refinement pattern for the same activities in Figure 8 and Figure 9 the DBN adapted well to behaviour changes.

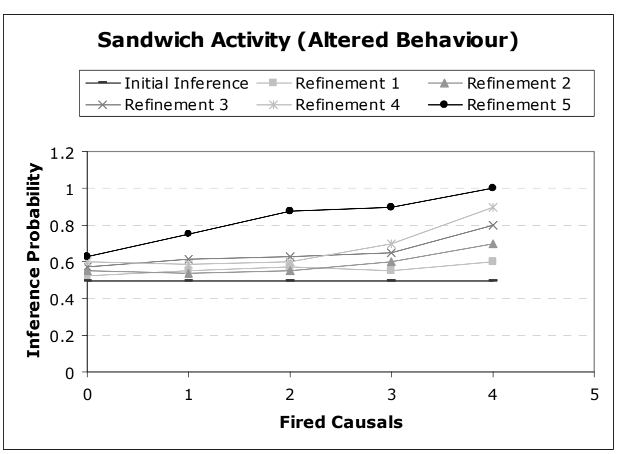

Figure 8. Making Sandwich Activity with altered behaviour

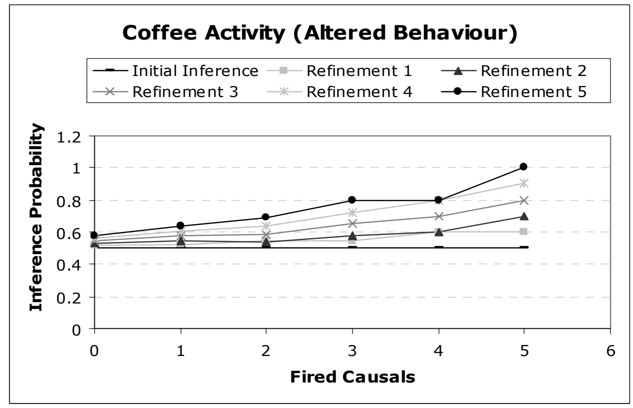

Figure 9. Making Coffee Activity with altered behaviour

V. CONCLUSION

Our system presented in this paper can accurately deduce user's activities solely based on the objects user manipulates. Our results show that DBN refinement process adapts to user behaviour changes therefore providing an accurate activity inference process. In this respect the Rule Engine plays a significant role. While we have focused mainly on home environments, our activity recognition architecture is flexible enough to be deployed in other domains and also can be extended to recognise an arbitrary number of activities.

VI. REFERENCES

[1] US Gov. Centers for Disease Control and Prevention, " Prevalence of Physical Activity, Including Lifestyle Activities Among Adults", MMWR August 15, 2003, Vol. 52, No. 32, 2003

[2] Julie Maitland, Scott Sherwood, Louise Barkhuus, Ian Anderson, Malcolm Hall, Barry Brown, Matthew Chalmers, and Henk Muller., "Increasing the Awareness of Daily Activity Levels with Pervasive Computing", 1st International Conference on Pervasive Computing Technologies for Healthcare 2006, IEEE press, 2006

[3] Don Peterson, Lin Liao, Dieter Fox, and Henry Kautz, "Inferring High-Level Behavior from Low-Level Sensors", In Proc. of UbiComp 2003, LNCS vol 2864, pp. 73-89, 2003

[4] Ling Bao and Stephen S. Intille, "Activity Recognition from User-Annotated Acceleration Data", Proceedings of PERVASIVE 2004, vol. LNCS 3001, A. Ferscha and F. Mattern, Eds. Berlin Heidelberg: Springer-Verlag, 2004, pp. 1-17, 2004

[5] Danny Wyatt, Matthai Philipose, and Tanzeem Choudhury, "Unsupervised Activity Recognition using Automatically Mined Common Sense", In Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI 2005), pp. 21-27, 2005

[6] Shiaokai Wang, William Pentney, Ana-Maria Popescu, Tanzeem Choudhury, and Matthai Philipose, "Commonsense-based Joint Training of Human Activity Recognizers", In proc. of International Joint Conference on Artificial Intelligence (IJCAI 2007), 2007

[7] Venet Osmani, Sasitharan Balasubramaniam, and Dmitri Botvich, "Self-Organising Object Networks using Context Zones for Distributed Activity Recognition", In Proc. of Second International Conference on Body Area Networks (BodyNets'07), ACM Press, to appear, 2007

[8] Artem M. Sokolov, "An adaptive detection of anomalies in user's behavior", In proc. of International Joint Conference on Neural Networks, vol. 4, pp.2443- 2447, ISBN: 0-7803-7898-9, 2003

[9] Helmut Hlavacs and Gabriele Kotsis, "Modeling User Behavior: A Layered Approach", In proc. of 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, pp.218-225, ISBN: 0-7695-0381-0, 1999