Context Management Support for Activity Recognition in Health-Care

Venet Osmani and Sasitharan Balasubramaniam

Telecommunications Software and Systems Group
Waterford Institute of Technology
Cork Road, Waterford,
Ireland
+353 51 30 2941
{vosmani, sasib}@tssg.org
http://www.tssg.org

Abstract. This paper addresses issues in human activity recognition with a particular focus in the health-care domain. The architecture presented is a non-obtrusive system combining distributed and centralised paradigms based on the concept of object networks that are created on the basis of user-to-object and object-to-object interactions. The activity inference process is distributed throughout the environment and is carried out by the inference engine that operates over the object networks. This process actively exploits various events generated from the object network, describing the proximity relationships, or depicting specific object functionality in addition to the state of the environment to infer user's current activity. Moreover we utilise information representing user's behaviour history as well as behaviour of similar users in equivalent context to advance the activity inference process.

1. Introduction

The last few years have seen tremendous research development in the field of Pervasive Systems. These researches have defined converged network infrastructures (e.g. WiFi networks, GPRS), software architectures (e.g. distributed middleware) as well as context information models to support pervasive computing applications in smart environments. The pervasive smart environments contain smart devices that have the ability to support user's daily life activities through efficient context evaluation mechanism that support pre-defined activities for different user's requirements. At the same time application adaptation for these activities is also required in response to changes from the environment.
Current research focus in pervasive systems has shifted towards supporting activity recognition for users based on their daily lifestyle activities and interaction behaviour with smart devices in the computing environment. Activity recognition has tremendous potential to support pervasive applications, especially in the health care domain. An example of applying activity recognition towards health care includes supporting the elderly in their everyday tasks and also monitor their behaviour for

signs of dysfunction, commonly known as dementia, and correcting their behaviour in order to sustain a normal lifestyle. Alzheimer's Association defines Dementia as the loss of intellectual functions (such as thinking, remembering, and reasoning) of sufficient severity to interfere with a person's daily functioning [1]. Patients that suffer from Dementia result in various severity degree of impairment of ADL (Activities of Daily Living). Therefore, an Activity Recognition system has a tremendous potential in supporting Dementia patients by providing audio/visual cues and orientation directions on the basis of the activity they are aiming to achieve. Certainly, there are many applications of activity recognition, beyond health-care that span across diverse domains. Simply enhancing one's life by providing pertinent information with respect to the current activity is a worthy objective to pursue, especially if the system is effectively invisible to the user.

Although, numerous developments have been made towards defining context-aware pervasive systems, little attention has been focused on defining mechanisms to automatically recognise and infer user activities. Current mechanisms for activity recognition have numerous drawbacks which include, (i) static activity recognition mechanisms that are defined by human operators, or (ii) limited recognition mechanisms that rely on specific technologies (e.g. RFID readers).

In this paper we introduce a combination of centralised and distributed context management architecture that supports various device technologies intertwined in the physical environment to infer user's activities based on the user-to-object and object-to-object interactions. Using this concept we seek to create an object network that supports context evaluation for activity inference and, self-learning and refinement mechanisms to support varied user behaviour.

The rest of this paper is organised as follows. We present and critique the most relevant architectures in Section 2, to follow up with the description of our overall architecture and integral components in Section 3. While Section 4 illustrates a motivating scenario, Section 5 draws the conclusive remarks from our ideas and presents the future plans.

2. Related Work

Inferring user's activity has been a research area for some time; however, recently this research domain received particular attention due to the potential it offers to various applications (e.g. Health-care). Specifically, reliable monitoring of ADL has multiple applications ranging form supporting the habitants of an elderly home up to providing pertinent information for health care workers, such as doctors, nurses, ambulance officers at an accident scene, etc.

A significant number of researches in the area of activity recognition have provided various means to infer user's activity and we briefly describe the most relevant systems. Guralnik and Haigh [2] describe the approach of collected data from a set of houses instrumented with a number of motion detection sensors. The captured information is fed to statistical algorithms that are used to extract the behaviour patterns of the house occupants. However authors do not describe any means to attach semantic information to the extracted patterns. There is no automatic recognition of user behaviour; the actual interpretation of the patterns is left up to the human operator.

Korhonen et al. [3] have devised a rather simplistic, but commercially exploitable system based on a wrist device. The device is worn by an elderly patient to monitor their well-being by evaluating context information through temperature, wrist movement and skin conductivity sensor. At the same time, the device also features a panic button. Triggered alarms that may result from prolonged non-movement are transmitted to a base station that routes the request to the appropriate operator. The system is limited because of the number of sensors utilised and does not provide facilities to be augmented by other types of information.

Another initiative in activity inference comes from University of Aarhus in Denmark [4]. Authors describe the issues that surround the activity inference, with a special focus on healthcare. Inferring user's activity based on the set of artefacts and other context information was found to be difficult, since activities are triggered by sources that are too complex to capture. At the same time the authors do not consider previous user behaviour or behaviour of other users in a similar context, which creates the necessary ground for learning.

University of Washington and Intel Research have devised an activity inference engine based on the 'Invisible Man' theory they have developed which states that activities are well characterised by the objects that are manipulated during their performance [5]. An RFID reader mounted on a glove records the information about objects being manipulated by a particular user which is fed to an activity inference engine. Activities are modelled through the web data mining techniques using 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) the second problem poses a greater challenge. While this problem may not be essential in home environments, a scalable architecture becomes critical, when considering workplace domains, for example hospitals where number of users may range in the thousands.

Overall the systems presented in this section lack one or more features to reliably infer user's activities by utilising diverse technologies, while catering for diverse user behaviour.

3. Architecture

The previous section has described some drawbacks of current activity recognition systems. Establishing a context-aware system across heterogeneous domains is a very challenging task. The problem is further exacerbated in cases when multiple users with potentially diverse behaviours come into play. In such cases, adaptation of the computing environment to suit individual user's needs (learning) while scaling to a vast range of devices, becomes a very important issue that have to be considered on the onset of a system design. Static architectures adhering to centralised structures will not fulfil these requirements.

Our system, presented in the next section, provides a solution that brings the realisation of these goals a step closer. In our activity recognition system, we believe that solely depending on sensor events is not substantial enough to infer activities. We believe that incorporating user behaviour is crucial towards accurate activity inferencing (e.g. different users may perform same activities but may only interact with common subset of sensors). Therefore, our systems supports (i) evaluating various deduced context information that is not limited to static sensor information, (ii) decentralised inferencing technique from objects surrounding user environment (iii) supporting experience transfer between heterogeneous domains (iv) incorporates learning techniques for both existing and user entering new domains.

The architecture depicted in Figure 1 is distributed in nature in order to facilitate potential large number of devices and users, specifically in cases where a centralised approach would pose a serious bottleneck once the number of context information types and users in the environment increase along with being error prone in the event of failures.

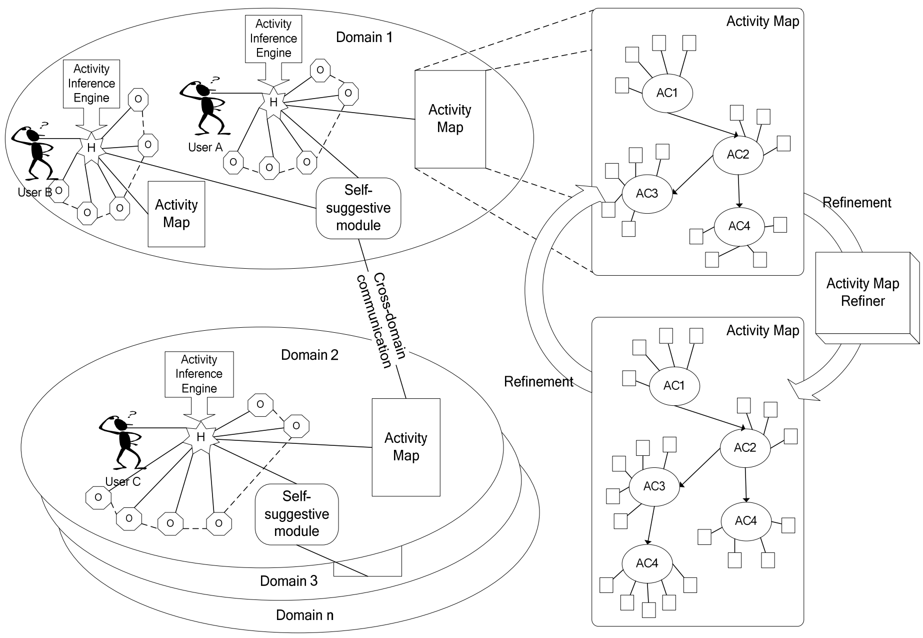

Fig. 1. Overall Architecture

In order to counter this problem we utilise the concept of domain object networks. An object is any entity that contributes to the activity inference process. We move away from the idea of a centralised inference engine and we distribute this process throughout the environment.

Inferring user's activity is carried out by the Activity Inference Engine (AIE) which involves acquiring and processing of information produced from multiple sources, divided into three components: object networks, Activity Map (AM) and information produced from the Self-Suggestive Module (SSM). A detailed description of each component follows in the next section.

3.1 Object Networks

The idea of object networks has been inspired from the functionalities of robots and sensor networks, which have capabilities to self-organise and cooperate collectively towards achieving variety of goals. Objects are considered everyday artefacts, such as laptops, PDAs, sensors, tables, beds, chairs or any other artefact that may be considered relevant in the process of inferring user's activity. Also the human is considered to be a part of the object network. An object network is specific to a domain. Typically the interaction between the neighbouring objects takes the form of the '*proximity_to*' relationship. This entails that objects in the vicinity can 'smell' each other and establish an interaction network.

We assume that each object has communication capabilities in addition to an embedded two-layer stack that is composed of the *infrastructure layer* and the *application module layer*. The discovery and election algorithms are contained in the infrastructure layer, which additionally houses an object profile. The profile plays an important role in the discovery process, since it contains identity information about the object, which is evaluated during the election process. The application software module hosts software modules that are loaded dynamically in response to the object usage.

3.1.1 Object Network Interaction

The election process is necessary to select the object that will carry out the activity inference process. In order to fully harness the processing power available within an object network and ensure an efficient activity inference process, the election between the objects takes place. Obviously not every object will have the processing and communication capabilities to run the AIE. Therefore the participants in the election are limited to objects that are capable of housing and running the AIE. As a result the object with the highest processing and communication power in accordance with the information supplied from the object profile bears the role of the *leader object*.

The AIE running in the leader object receives events from the object network upon completion of the election process. Therefore, whenever the '*proximity_to*' relationship holds true between objects, it causes an event to be reported back to the elected object. The '*proximity_to*' relationship will hold true in all cases where there is an intersection between the operating radii of two or more objects belonging to an object network. The operating radius is specific to a particular object, where an example may be the viewing area of a laptop or a PDA, a certain distance between medical officers and a medical tray or standing area near a patient's bed.

However, the events generated from the object network are not limited to proximity events only. The object profile also defines events related to the object's functionality with respect to the current object usage. An example may include the currently active application on a PDA or even state of a patient (e.g. awake vs. sleeping). As such the interaction between the objects relieves the processing burden on the activity inference engine, which otherwise would need to keep track of the location of each object and monitor the conditions when two or more objects are in proximity while querying each object for the information it provides.

3.2 Activity Map

The Activity Map (AM) is a repository containing the history of recorded activities for a particular user. The AM, which is specific to a particular user and is located in each domain, contains the sequence of activities a user has performed and also defines the relationships between different activities. AM relies on the idea that humans typically perform a sequence of activities drawn from a finite activity set in order to achieve a certain goal. This idea is supported by psychological studies stating that many of the human behaviours are hierarchically structured [6]. The AM is a directed graph structure where each arc connecting two activities is assigned a probability value. These values are continuously refined over time by the Activity Map Refiner (AMR) and represent the probability of the next activity following the current activity (see Figure 2).

Since the sequence of activities is determined through probability relationship, we have selected the Bayesian network to model the AM. Context information held in an AM is evaluated by the leader object to infer the user's activity.

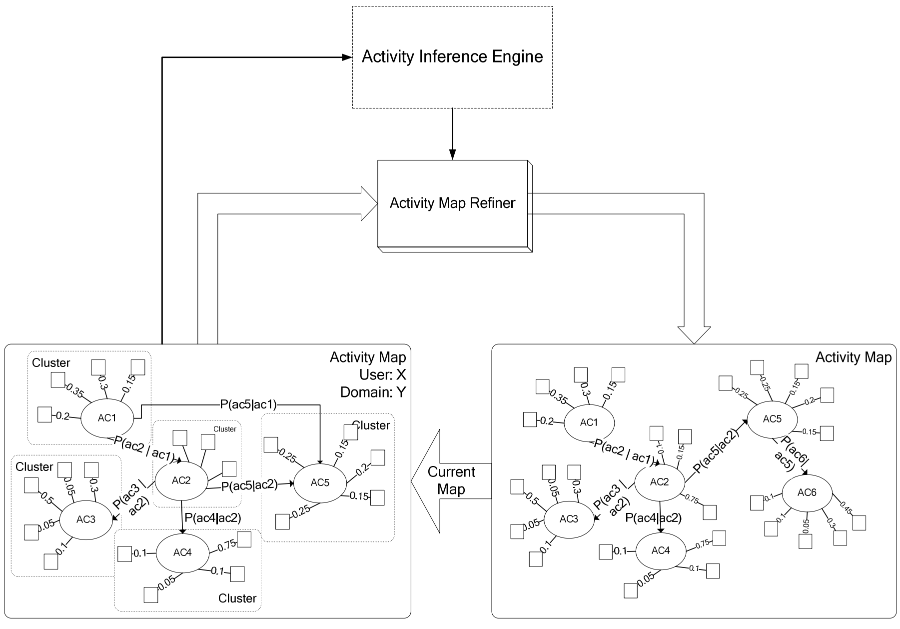

Fig. 2. Graphical Representation of an Activity Map

The interaction mechanism of the leader object with the AM is two fold. Although, the leader object evaluates context information contained in the AM repository, the leader object also updates this repository with the inferred activities. Therefore, an inferred activity is submitted back to the AM to be processed by the AMR, which may change the latest relationships between the recorded activities in the repository, update the relationship probabilities or transform the current AM by adding or removing activities. These mechanisms create the basis for constant learning of user behaviour and adaptation according to the behavioural changes.

In the event where the current activity has two or more diverging nodes with the same probability value, signifying that two or more future activities have the same chance of occurring, the leader object uses the information from the object network to discriminate between the competing activities. Given that each activity in the AM has a different set of context information associated with it, the leader object compares the current state of the user's environment (user's context) with the actual competing activities and makes the decision based on pattern matching.

For example, in the evenings Dr. James will regularly prescribe medicines or look through the patients' lab results. His AM shows that each activity has an equivalent possibility of occurring, given the current state of the environment established on the basis of context information originating from the devices of the object network. However, prescribing the medicine is carried out through his PDA, whereas the patient's results are displayed on the large wall-mounted touch sensitive display in his office to ensure convenient viewing of detailed information such reports typically contain. As a consequence, the leader object will discriminate through these activities based on the objects handled (e.g. PDA versus the wall-mounted screen), albeit Dr. James' AM indicates that he has an equal likelihood of engaging in either activity.

The Activity Map component allows for an important feature of a context-aware system, the capability of activity prediction. In our case, the information stored on the Activity Map, provides the basis to predict subsequent activity. Predicting user's activity is an important feature of a context-aware system, since the services and resources that are likely to be used can be also predicted, resulting in behaviour that is closer to meeting the user expectations from an intelligent system.

3.3 Inferring user's activity

Upon completion of the election process, the leader object executes the AIE, which is the main component of the system containing the logic to enable the activity deduction process. This is based on the information sourced from the object networks as well as the AM, combined into the activity model to facilitate the inference process. The activity model is based on our earlier work on context definition [7] and is purposely kept simple in order to create a lightweight AIE that can support devices of different processing capabilities (see Figure 3).

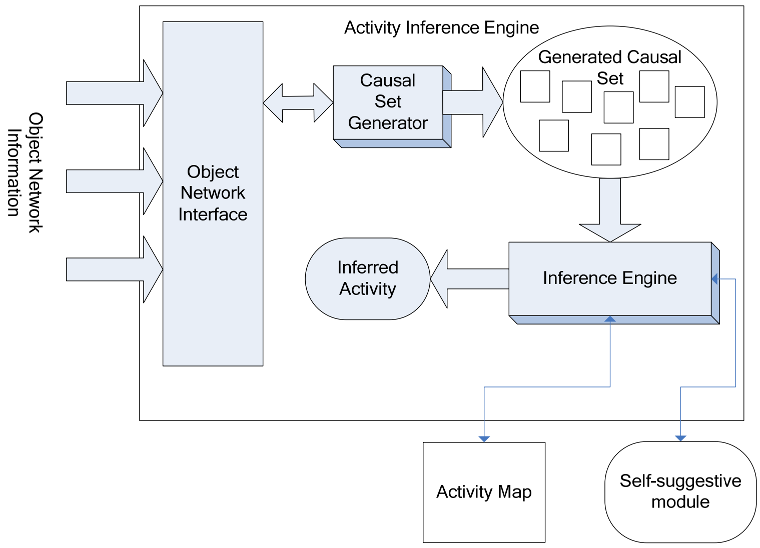

Fig. 3. Activity Inference Engine

Essentially it consists of an activity identifier and a set of context information associated with this activity. An instantiation of this model results in an *activity cluster* that comprises an activity identifier containing the semantic description of the activity and it also features the *causal set*, which defines the context information that has to be true for an activity to be performed.

An activity cluster, similar to the AM, is also modelled using a Bayesian network. Logically an AM contains a set of activity clusters for a particular user. The central node in the Bayesian model is represented by the activity itself and the elements from the causal set are represented as edge nodes. Each edge node is associated with a weight which determines the impact a certain causal has on the activity. The impact values are refined over time to closely accommodate the user behaviour. The state of each causal being true or false is typically determined from the context information produced by the object network. The events reported to the leader object are utilised to determine the status of each causal until such time that the number of causals with the status set to true while considering the impact of each causal, is sufficient to infer the user's activity with a reasonable accuracy.

Object network communication with the AIE is accomplished through the Object Network Interface. This component listens for the various events produced from the object network for example, object A is in the vicinity of the object B and forwards them to the Causal Set Generator (CSG). The CSG has the responsibility to carry out conversion of events to a higher abstraction level resulting in the *generated causal set*. The CSG converts events from the object network to a set of *assertions*. Therefore, in the above example the event is converted into proximity(object_a, object_b) assertion. The conversion logic is facilitated by the information contained in the object profile. The object profile in addition to afore-mentioned information also specifies the object's functional capabilities. As a result objects are aware of the functionality they provide. This awareness enables an object to monitor interaction with the user and other objects in the network and supply object's usage information to the leader object by means of object network events. For instance, an element of the object network, an anaesthesia machine switched on – dripping a particular type of anaesthetic – generates an event that conveys information regarding the current functionality of the machine which is processed by the CSG and converted to an assertion to ultimately deduce surgeon's activity.

The inference engine performs a two phase process - matching and selection. In the first phase, each causal from the generated causal set is matched with the causals within individual activity clusters. A *selection value* is produced for each cluster where there is at least one causal that corresponds to an element in the generated causal set. Activities with the selection value higher than a particular threshold (e.g. 0.7) are selected to create the *potential activity set* and in simplistic cases the activity with the highest value is selected to represent user's current activity. The selection of the activity would complete the second phase.

However, there are cases where the elements of the potential activity set will have the same or nearly equivalent selection values. In order to resolve this ambiguity the engine harvests historical information regarding the recorded user behaviour. In such occasions, the role of the AM becomes primary due to the fact that it holds information about the most-likely activity based on the previous activity the user was engaged in. This information is combined with the potential activity set towards resolving the ambiguity, a process that entails amplification of the selection value of a particular activity that matches the information obtained from the AM. This mechanism results in the selection of the most probable activity.

3.3.1 Self Suggestive module

However, in the process of inferring user's activity there are instances when the user behaviour history information is not always available. In such occasions the inference engine may not produce a reliable result since the information provided solely by the object network may result in ambiguity, as pointed out in the previous section. For example a user may just have started using the system, implying that there will be no history of recorded activities. Also the user may be engaged in a specific activity for the first time, so there will be no record describing this new activity in the user's particular AM.

In order to address these problems we have devised a method based on a technique proven very successful with significantly positive results in the e-commerce web, known as Collaborative Filtering (CF). The CF algorithms are based on a simple intuition, which is predictions for a user should be based on the preference patterns of other people who have similar interests [8]. Systems implementing the CF technique typically perform the following steps, (i) the system creates and maintains a user profile, (ii) the user profile is compared to profiles of other users, and weighs the user profile based on degree of similarity, and (iii) considers a set of the most similar profiles, and uses information contained in these profiles to recommend items to the user [9].

We have applied a similar idea to activity inference process. The main difference is that CF has been designed to make recommendations for a particular item to the user, whereas in our case the system recommends a potential user activity to itself such that the resulting value from executing the algorithm is fed back to the system.

The self-suggesting aspect works by observing other users, who are similar to the current user, where the similarity degree is computed. What follows is that similar users tend to perform similar activities within sufficiently similar context, especially in rather structured domains like workplaces. Therefore, we use this observation to aid the inference process by processing behaviour history records of other users given that this information satisfies the condition that users in question have a sufficiently high degree of similarity.

The activities performed by these users in a context similar to the context of the current user (e.g. domain, location, manipulated objects, time, etc) are observed and this information is utilised to auto-suggest an activity for the current user. The information produced from the SSM is considered another causal, and similar to the information generated from the AM is given a probability value that represents the degree of impact this information has on the activity inference process. We envision that self-suggesting feature will have a significant impact in system learning. Over time, the information about the new activity will be propagated to other users' AM by means of our collaborative filtering-like technique.

Utilising the above components creates the necessary ground to enable activity inference and support users with their everyday tasks. It should be noted that our approach does not necessarily rely on having a user profile, especially information that is considered more sensitive such as user's preferences, even though the existence of a profile will augment the inference process. However, given the fact that our architecture is geared towards supporting healthcare staff, certain information about the users typically are available in an organisation-wide domain such as identification, position, rank, etc. Since the topic of user monitoring is quite delicate, we believe that our approach provides a sensible trade-off between the user's privacy and normal system operation, especially in light of the fact of non-necessity of prior user profiling.

4. Motivating Scenario

We now present a scenario in which our architecture augments the activities of the hospital staff in their everyday care practices. In this scenario we assume that the patients are fitted with various monitoring devices that monitor the vital signs of the patient conditions. At the same time, the doctors and nurses have hand-held wireless computing devices (PDA) at their disposal to help them carry out their tasks. The PDA devices are not personal in the sense that they permanently belong to a doctor or a nurse; rather when member of staff acquires a particular PDA, a personal relationship is established, given that the PDA will recognise the new user. In addition, a location-tracking system monitors the movement of the patients and the hospital staff.

As Dr. James walks around the hospital hall, his PDA displays the scheduled appointments that rearrange dynamically according to the patients' location and attention requirements to ensure the doctor's time is utilised with the highest efficiency. Glancing on the monitor located near the door of the patient’s room, the doctor is presented with the patients' details and historical graphs of their condition recorded overnight. Patients currently not in the room are marked with a coloured strip so that Dr. James can decide whether to visit the next room. The system exhibits this behaviour due to the fact that his PDA has inferred that Dr. James is currently engaged with ‘*doing the rounds*’ activity established on the basis of information obtained from the object network created by the location tracking system, agenda creating systems, and the PDA. This ‘*doing the rounds*’ activity is determined from the events generated between the different objects.

As soon as the doctor enters the patient's room, the displayed details of patients' information on the monitor at the door fades away, leaving only the detail of information required by the nurses. At this point Dr. James' PDA has negotiated with the computing artefacts in the room and the election process between the different objects has resulted in his PDA changing the role of the leader object. Consequently the responsibility for coordination and decision making (running the AIE) has been delegated to the more powerful on-board computer in the patient’s bed. The events generated from the objects within the network have been matched to causal set for the activities. The activity ‘*Evaluating patient conditions*’ is inferred. This activity leads to the Electronic Patient Record (EPR) being retrieved and presented in the nearby interactive wall-mounted display. The activity also retrieves images of the recent CT scans as well as graphs exhibiting the previous 7-day medical history with an overview of the patient’s condition. The doctor consults with the patient and in light of the results shown, decides to prescribe a lighter medical treatment since the patient has shown signs of improvement. The doctor modifies the EPR through the interactive display and changes to the patient’s details are committed.

Upon detecting modifications of the drug combination, an alert is sent to the PDA of the nurse currently in duty with the information regarding the new therapy. The nurse handling the medical tray walks into the medicine room, and regulates the new therapy while her PDA infers the ‘*medicine administration*’ activity for the patient in question. The nurse, the medicine tray, medicine cabinet and the PDA amongst other computing entities have created an object network and the ‘*drug selection*’ activity was inferred. The drug selection activity allows each drug placed on the tray to be checked against the doctor’s recommendations. When she finishes with the drug selection, the drug details along with the nurse ID are automatically displayed in her PDA and she submits this information to the drug administration log.

This scenario has shown the process of object formation supporting the location changes of the doctor. Following the formation of object network, automatic assertion of activities based on generated events of the different objects is triggered to select the corresponding activity. The scenario has also shown how changes made by an activity can lead to invoking other activities of other users within the same domain.

5. Conclusion

In this paper we have presented a distributed architecture that facilitates the enhancement of user actions by inferring activities the user is currently engaged in.

While the importance of activity inference has been recognised for context-aware systems, and substantial work has been done in this area, the current approaches in the literature lack capabilities to infer user's activity from multitude sources of information.
Our architecture represents advancement over the systems surveyed, since it uses multiple and diverse information sources to deduce user's activity. While the idea of observing objects manipulated by a user to infer the user's activity has been presented by others, our architecture makes further advancements. These advancements are not limited to objects within an environment, but also include the previously conducted activities as well as the behaviour of other users in similar context is considered in the process. Furthermore our architecture does not rely on a central server to carry out the activity inference; rather this process has been distributed throughout the environment. We could find no references to any system that incorporates the concepts presented in this paper. Currently we are in the process of implementing the architecture described above and aim to deploy the system through the Smart Hospital test-bed located in our research centre.

6. Acknowledgements

This work is supported by the Irish Higher Education Authority (HEA) under the PRTLI "M-Zones" initiative aiming to develop novel information and communications management technology to support dynamic, integrated management of participants, information appliances, and smart space infrastructure.

7. References

[1] The Alzheimer's Association, "Glossary", 2006, Available: http://www.alz.org/Resources/Glossary.asp#d.

[2] Valerie Guralnik and Karen Zita Haigh, "Learning Models of Human Behaviour with Sequential Patterns", Proceedings of the AAAI-02 workshop 'Automation as Caregiver', pp. 24-30, 2002, Available: http://www.cs.cmu.edu/~khaigh/papers/AAAIWS02-VGuralnik.pdf.

[3] Ilkka Korhonen, Paula Paavilainen, and Antti Särelä, "Application of ubiquitous computing technologies for support of independent living of the elderly in real life settings", UbiHealth 2003: The 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications, 2003, Available: http://www.healthcare.pervasive.dk/ubicomp2003/papers/Final_Papers/2.pdf.

[4] Jakob Bardram and Henrik Bærbak Christensen, "Open Issues in Activity-Based and Task-Level Computing", First International Workshop on Computer Support for Human Tasks and Activities, CfPC PB-2004-60, 2004, Available: www.pervasive.dk/publications/files/open_issues_bardram.pdf.

[5] Matthai Philipose, Kenneth P.Fishkin, Mike Perkowitz, Donald J. Patterson, Dieter Fox, Henry Kautz, and Dirk Hähnel, "Inferring Activities from Interactions with Objects", IEEE Pervasive Computing, vol. 03, no. 4, pp. 50-57, October-December, 2004, Available: http://seattleweb.intel-research.net/projects/activity/pics/philipose_pervasive_adl.pdf.

[6] Jeff Zacks and Barbara Tversky, "Event Structure in Perception and Cognition", Psychological Bulletin, 127, 3-21, 2001, Available: http://www.apa.org/journals/bul.html.

[7] Venet Osmani and Sven van der Meer, "Context? Yes but to whom?" International workshop on Context in Mobile HCI in conjunction with MobileHCI'05, Salzburg, Austria, 2005, Available: http://mobilehci.icts.sbg.ac.at/context/papers/osmani_vandermeer.pdf.

[8] Koji Miyahara and Michael J. Pazzani, "Collaborative Filtering with the Simple Bayesian Classifier", Pacific Rim International Conference on Artificial Intelligence, 2000, Available: www.ics.uci.edu/~pazzani/Publications/IPSJ.pdf.

[9] Upendra Shardanand and Pattie Maes, "Social Information Filtering: Algorithms for Automating "Word of Mouth"", Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, 1995, Available: http://www.cs.ubc.ca/~conati/532b/papers/chi-95-paper.pdf.