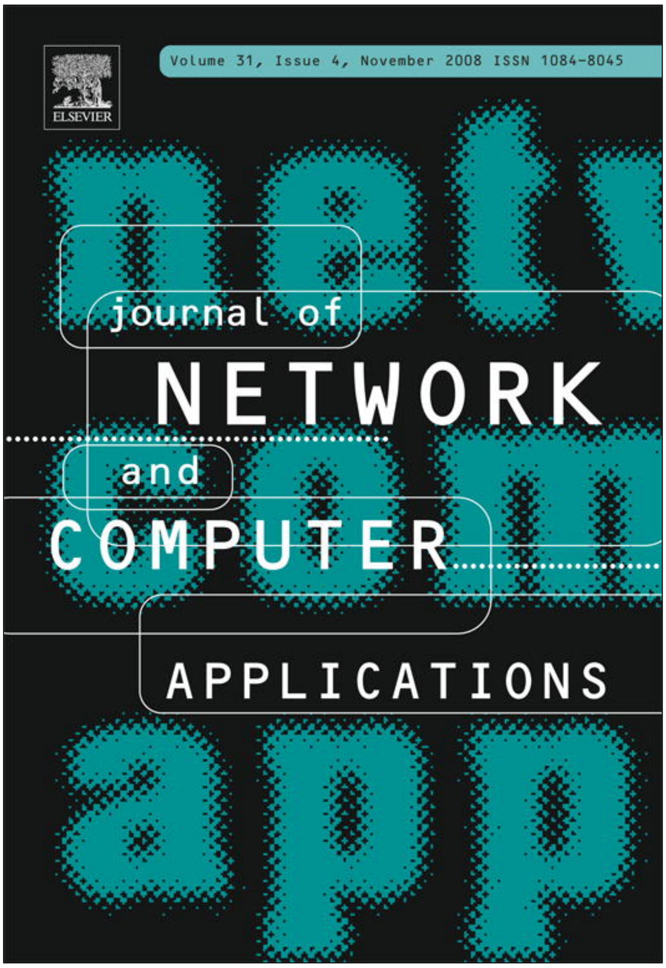
ELSEVIER
Volume 31, Issue 4, November 2008 ISSN 1084-8045
journal of
NETWORK
and
COMPUTER
APPLICATIONS

Available online at www.sciencedirect.com

ScienceDirect

Journal of Network and
Computer Applications 31 (2008) 628–655

Journal of NETWORK and COMPUTER APPLICATIONS

www.elsevier.com/locate/jnca

Human activity recognition in pervasive health-care: Supporting efficient remote collaboration

Venet Osmani*, Sasitharan Balasubramaniam, Dmitri Botvich

Telecommunications Software and Systems Group, Carriganore West Campus, Waterford Institute of Technology, Cork Road, Waterford, Ireland

Received 24 May 2007; received in revised form 14 October 2007; accepted 1 November 2007

Abstract

Technological advancements, including advancements in the medical field have drastically improved our quality of life, thus pushing life expectancy increasingly higher. This has also had the effect of increasing the number of elderly population. More than ever, health-care institutions must now care for a large number of elderly patients, which is one of the contributing factors in the rising health-care costs. Rising costs have prompted hospitals and other health-care institutions to seek various cost-cutting measures in order to remain competitive. One avenue being explored lies in the technological advancements that can make hospital working environments much more efficient. Various communication technologies, mobile computing devices, micro-embedded devices and sensors have the ability to support medical staff efficiency and improve health-care systems. In particular, one promising application of these technologies is towards deducing medical staff activities. Having this continuous knowledge about health-care staff activities can provide medical staff with crucial information of particular patients, interconnect with other supporting applications in a seamless manner (e.g. a doctor diagnosing a patient can automatically be sent the patient's lab report from the pathologist), a clear picture of the time utilisation of doctors and nurses and also enable remote virtual collaboration between activities, thus creating a strong base for establishment of an efficient collaborative environment. In this paper, we describe our activity recognition system that in conjunction with our efficiency mechanism has the potential to cut down health-care costs by making the working environments more efficient. Initially, we outline the activity recognition process that has the ability to infer user activities based on the self-organisation of surrounding objects that user may manipulate. We then use the activity recognition information to enhance virtual collaboration in order to improve overall efficiency of tasks within a hospital environment. We have

*Corresponding author.
E-mail addresses: vosmani@tssg.org (V. Osmani), sasib@tssg.org (S. Balasubramaniam), dbotvich@tssg.org (D. Botvich).

doi:10.1016/j.jnca.2007.11.002

analysed a number of medical staff activities to guide our simulation setup. Our results show an accurate activity recognition process for individual users with respect to their behaviour. At the same time we support remote virtual collaboration through tasks allocation process between doctors and nurses with results showing maximum efficiency within the resource constraints.

Keywords: Human activity recognition; Activity inference; Collaborative environments; Pervasive health-care; Pervasive Computing; Ubiquitous computing

1. Introduction

High availability and small form factor of wireless devices has given rise to a vast number of applications within Pervasive Computing (Weiser, 1991) research. The ultimate goal of Pervasive Computing is to drastically improve user's life by supporting aspects of their lifestyle through an invisible computing environment. Monitoring user behaviour can assist the user in a vast number of day-to-day activities from food preparation up to various training regimes, encouraging users to live a healthy lifestyle. While, leading a healthy lifestyle is an important factor to a healthy life, conditions such as chronic illnesses or accidents still occur, requiring medical attention and care. Providing patient care in recent years has left medical institutions with an ever increasing challenge, mainly due to the rising costs of health-care. Demographic changes are attributed as one of the leading contributors, putting health-care institutions under strain. Increased life expectancy has led to the median age being pushed higher, thus resulting in an increased number of senior citizens. In general, senior citizens are much more vulnerable to chronic diseases than the rest of the population, hence requiring medical care more often. In the US, the proportion of the population greater than 65 is projected to increase from 12.4% in 2000 to 19.6% in 2030 which translates into an estimated 71 million elderly by 2030 (US.C.B., 2007). This increase will have a direct impact in health-care institutions, specifically considering the fact that the health-care costs per capita for persons over 65 years are three to five times greater in comparison with the health-care costs of persons under 65 (Jacobzone and Oxley, 2002). To address the rising costs, the public financing of long-term care for the period 2000–2020 is projected to increase 20–21% in the UK and the US, while this figure in Japan is estimated to be 102% (JP and DW, 2000).

When considering above figures it becomes evident that medical institutions are actively seeking cost-cutting solutions. One avenue being explored is in the latest technological advances, specifically Pervasive/Ubiquitous Computing (Weiser, 1991). Pervasive computing has already given rise to a number of novel medical applications. Tele-medicine, real-time telemetry, network-based haptic devices, and remote surgery are just some of the applications enabled by advances in Pervasive Computing. Doctor-centric applications are also in the research focus, enabling doctors to conduct their activities more efficiently. For example, the embedded computing environment can intelligently respond and adapt to changes in doctor's activities. Such adaptations range from simple phone call redirection, for example when the doctor is engaged in a crucial activity such as surgery, up to advanced applications that can display current patient state, medical history and any other information relevant to the doctor while the doctor is engaged in patient examination. Clearly, the enabling platform for these applications is a flexible and robust activity recognition system that has the ability to continuously infer medical staff activities with a high accuracy.

Activity recognition is a research strand of Pervasive Computing that is concerned with 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. Having continuous knowledge of activities of medical staff, the health-care institutions can greatly enhance hospital processes and utilisation of their staff. This has the potential to cut down associated costs by increasing the efficiency of the working environments as well as support various other applications and information relevant to the current user's activity. There have been a number of studies (Centeno et al., 2003; Martínez-García and Méndez-Olague, 2003; Wijewickrama and Takakuwa, 2006) with the objective of improving collaboration efficiency within various departments of health-care institutions such as emergency room or surgery theatre. Typically, these studies manually observe processes within the hospital, where activities and allocation of doctors and nurses and other information are fed to a simulation engine such as (Corporation, 2007; CreateASoft, 2007) in order to correct inefficiencies and increase productivity.

However, manual data collection typically involves researchers spending a considerable amount of time and effort to record the relevant data. The amount of labour required to aggregate the data renders the manual technique quite inefficient. More importantly, however, efficiency is not a static measure and tends to decrease over time since it is dictated by the dynamic behaviour of the medical staff. Therefore, the manual technique becomes unfeasible to often collect staff activity collaboration data in order to maintain high efficiency.

With these issues in mind, we focus this paper on our activity recognition system and its applicability to the health-care domain. We seek to study the interaction between medical staff through the inference of their activities and provide means to make this interaction more effective. Our solution is unique in that it creates a synergy between the activity recognition mechanism that provides information about activities of the health-care staff and the optimisation process that uses this information to enable an efficient collaborative environment. Also, having an automated solution allows for a continual adaptation to the behaviour dynamics of the medical staff, which ultimately provides a maximum collaboration efficiency that cannot be achieved or is unfeasible with the manual solutions that we review in the next section. Our case study is focused on the recognition of activities of primarily nurses and doctors and we make the point that utilising information about activities of the medical staff can enhance their tasks and applications and at the same time support remote doctor–nurse collaboration, by making this process more efficient.

The rest of this paper is organised as follows. We first review the most relevant work in the two main focus areas, namely activity recognition and collaborative health-care. Then in Section 3 we present our activity inference architecture, the description of the main components and their interaction. Section 4 deals with collaborative activities and how they are supported by our activity recognition architecture, while Section 5 presents the simulation results of our work. Section 6 draws the main conclusions.

2. Related work

Pervasive health-care paradigm (Bardram et al., 2006; Saranummi et al., 2006; Varshney, 2003) has recently emerged as a distinct research strand within Pervasive

Computing. There are a number of aspects in the focus of Pervasive health-care, such as improving the general health (Maitland et al., 2006; Oliver and Kreger-stickles, 2006), monitoring patients (Bouchard et al., 2007; Karl et al., 2006; LeBellego et al., 2006; Osmani et al., 2007), enabling an efficient schedule of the health-care staff (Centeno et al., 2003; Driver et al., 2006) or even recognising their activities (Bardram and Christensen, 2007; Favela et al., 2006). While Pervasive health-care encapsulates a number of application domains, our focus is specifically on two aspects: (i) activity recognition of medical staff which provides continuous information about the current activities of doctors and nurses and (ii) utilising this information to provide an efficient collaboration process between doctors and nurses, allowing medical staff to utilise their time in the most efficient manner and collaborate in a remote virtual environment while being physically distant. Therefore, in the next section, we review and critique the most relevant activity recognition systems and then we describe the ongoing research into enabling efficient collaborative environments in the health-care domain.

2.1. Activity recognition

The ability to recognise human activities is a key factor if computing systems are to interact seamlessly with the user's environment. Research into enabling computer systems to recognise human activities has emerged as an application domain of computer vision research. The strong interest in this domain has been motivated by the desire to improve machine to human interaction that offers many promising applications (Gavrila, 1999).

However, the more recent trends in human activity recognition have witnessed the appearance of another strand in this domain. Technological advancements and steady form-factor miniaturisation have enabled dense instrumentation of our living environments with a large variety of multimodal sensors and actuators. Such environments possess the ability to monitor user's behaviour and provide information pertaining to user's actions, which is then filtered, processed and composed in order to infer user's activities. As such human activity recognition can be divided into two major approaches, namely machine-vision-based recognition and sensor-based recognition. Of course, this division is by no means strict and hybrid approaches also exist, for example, Oliver and Horvitz (2005).

The main focus of our paper though is on the latter, sensor-based approach. Therefore, we now review the most relevant systems in this domain.

Guralnik and Haigh (2002) describe the approach of collected data from a set of living environments instrumented with a number of motion detection sensors. The captured information is fed to statistical machine learning algorithms that are used to extract the behaviour patterns of the house occupants. However, reliance solely on the motion sensors is insufficient to deduce activities with high accuracy and also makes it very difficult to understand specific user behaviours. In Kern et al. (2003), authors describe a hardware platform equipped with three-dimensional accelerometers. However, results reported show only a small number of simple activities that are recognised including sitting, standing, walking, handshaking, which may be attributed to using only one type of sensors. The framework is heavily centralised with no support for personalisation to suit specific user behaviour. Bao and Intille (2004) also propose recognising human activities based on accelerometers. Authors report recognition accuracy up to 95%. However, their approach limits the number of activities the system can recognise. Another initiative in activity inference comes from University of Aarhus in Denmark (Bardram and Christensen, 2004).

Although authors describe the issues that surround the activity inference, with a special focus on health-care, 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.

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 (Wyatt et al., 2005). 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; Wang et al., 2007) the second problem poses a greater challenge for scalability. While this issue may not be essential in home environments, a scalable architecture becomes critical, when considering workplace domains, for example, hospitals where number of users as well as devices and sensors may range in the order of thousands.

Overall the systems presented in this section lack one or more features to infer a large number of user's activities. More importantly the majority of these systems rely on a single technology that effectively decreases the richness of information generated as a result of user actions and behaviour, which limits the number of activities that can be recognised. Our design aims to alleviate these issues by employing a number of novel concepts that are mapped to various components of the architecture.

2.2. Collaborative health-care

The concept of collaborative health-care environments has gained increased attention, particularly due to the potential they offer in delivering quality care and lower the costs by increasing efficiency. Cost-cutting measures are necessary if the hospitals are to remain competitive in light of increased demand for resources. A number of research projects have studied ways to improve efficiency amongst medical staff. For example, Martínez-García and Méndez-Olague (2003) argues that activities being performed and their associated information, such as where, when and how these activities are performed have to be analysed in order to provide a complete analysis of the medical staff with patient interaction process. However, a major issue is that the interaction has to be manually observed and also entered into the simulation tool, so that the interaction process can be made more efficient. By changing various parameters through the simulation tools, the authors report an increase of 20% of patients provided with medical attention. Authors in Wijewickrama and Takakuwa (2006) aim to reduce the weighted average patient waiting time by simulating the patient care provisioning process. The data collection was performed manually through interviewing doctors, nurses and also some data was collected through observation. Since this was a modern hospital, some data was already available, which primarily was data pertaining to patient appointment details and check in times. The simulation and optimisation developed by the authors could greatly benefit from an automatic activity recognition, especially in recognition of activities of doctors while engaged with patients. Through application of optimisation in considering a number of scenarios using the collected data, the authors report 59.95% reduction in the weighed average patient waiting time. However, a change in the simulation parameters, for example introduction of new treatments, would require the pertinent data to be collected manually again. Another effort that aims to increase efficiency of health-care institutions by optimising the staff scheduling is described in Centeno et al. (2003). Authors report a 28% improvement over the existing schedule obtained through application of Integer Linear Programming (ILP). However, the data is entered manually and as with the previous systems. Responding to changes requires manual effort, making the process costly.

Therefore, an activity inference system becomes essential in enabling automatic gathering of activity information thus diminishing the effort required for the vital part of the simulation—collaborative activities and interaction information. In the following sections, we provide the details of how our architecture realises an accurate and robust activity inference process and how this architecture supports remote collaborative environments.

3. Architecture overview

Previous section critiqued some of the most relevant work in the area of activity recognition and its application to health-care. Overall the research literature presented lacks one or more features that present a significant barrier towards providing a holistic approach to activity recognition. This is due to the fact that activity recognition architectures face a set of stringent requirements in order to intelligently and efficiently infer dynamic user behaviour with a high accuracy. Device heterogeneity is an issue that must be considered at the onset of design. Also, reliance on a single sensing technology or standard will have severe negative impact on interoperability with devices that a user may own in the future. Centralised knowledge processing ultimately creates a bottleneck, potentially weakening the system performance and in severe cases resulting as a burden rather than a supporting tool in the user's day-to-day goals and activities. An important issue that has to be addressed lies in the continually changing user behaviour. Everyday experience reveals that user behaviour patterns are not static; rather they tend to vary with time. Therefore, mechanisms that cater for this dynamicity are essential and have to be incorporated into the architecture design of an activity inference system. Addressing these challenging issues becomes paramount in the system design in order to ensure a high likelihood of user acceptance, which may ultimately determine the difference between acceptance success and failure.

Our architecture brings the realisation of these challenges one-step closer. Our system supports the following characteristics: (i) evaluating deduced context information that is not limited to static sensor information but from various sensors and objects within the living environment; (ii) context information processing in a distributed, hierarchical manner resulting in a high level of abstraction of context information; (iii) support self-organisation of devices into object networks that infer user activities; (iv) decentralised user activity inference technique through the objects in the surrounding user environment; (v) incorporating learning techniques for both existing and new users entering a domain; and (vi) enabling remote collaborative working environments. In order to tackle these issues we have defined a number of components comprising the activity inference architecture, namely object networks that provide an efficient processing platform of information generated as a result of user actions, Activity Map (AM) that acts as a

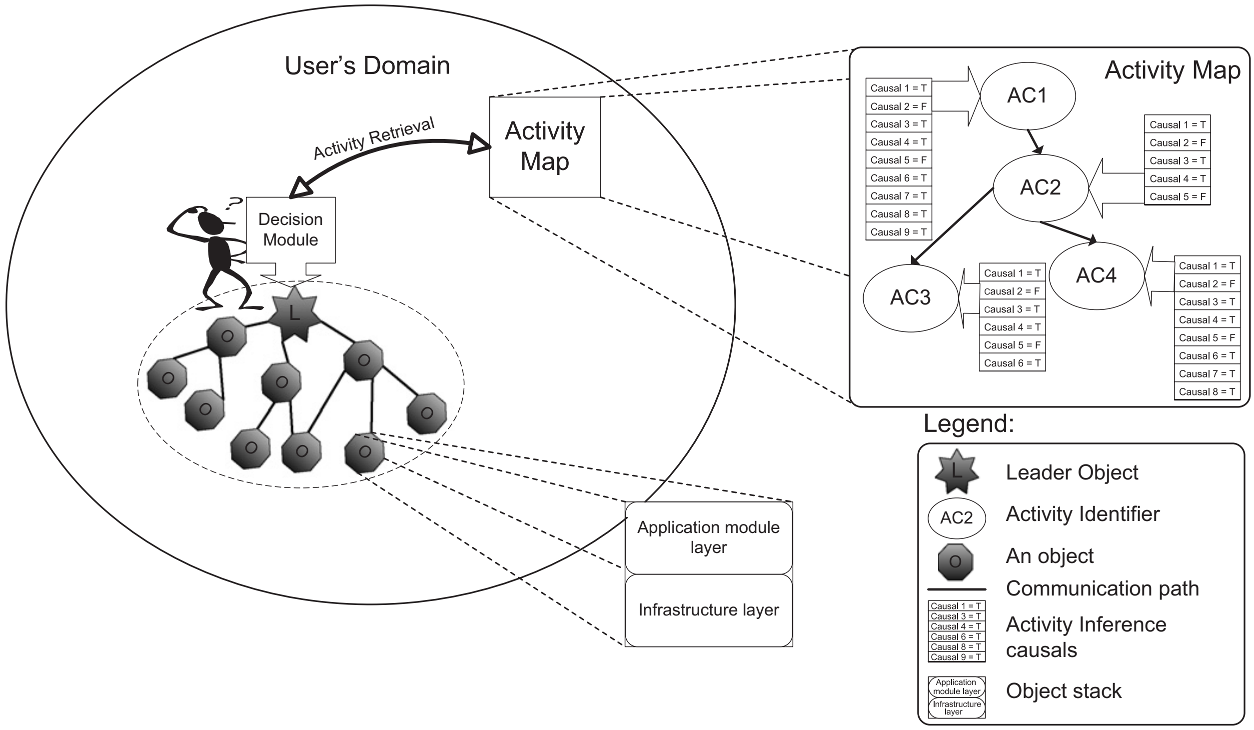

Fig. 1. Architecture for activity recognition.

repository of user's activities and a 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. These components are illustrated in Fig. 1.

Our aim is to create a decentralised mechanism to infer user activities. Activity inference starts when user enters a particular location and stabilises his/her mobility, at which point the objects surrounding the user will collectively self-organise into an object network. The object network in conjunction with the DM will infer user's activities by drawing out a generic set of events that contribute to an activity from a centralised repository—the AM. The sections that follow describe each component in detail and their functionality.

3.1. Object networks

The object networks concept has been devised on the basis of functionalities of sensor networks that have the ability to self-organise through peer-to-peer interaction; however, our concept is more generic in nature. Object networks extend beyond sensors and sensor interaction; an object network is an overlay network that encapsulates various devices with different functionalities and processing capabilities that provide crucial information pertaining to user's actions and environment status. This includes 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. Laptops, PDAs, medical monitors, and instruments) or just plain low processing sensors. Once the user's location stabilises, the objects dynamically organised into an emergent object network hierarchical structure.

Utilisation of object networks is two-fold. 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 sampling, data representation, composition, filtering). This also has the added benefit of making this 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 the local object-to-object interaction rules. The 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.

3.1.1. Object network constituents

Each element of an object network is equipped with a sensing mechanism that allows it to discover other objects in the vicinity and query their services. This mechanism assumes existence of communication capabilities in addition to an embedded dual-layer stack (see Fig. 1) that houses the object logic. The stack comprises an *infrastructure layer* where object sensing and discovery algorithms are contained and its primary role is the establishment of the object network. This layer also coordinates an election process between the elements of an object network to determine the object with the highest capabilities—the *leader object*. The second layer within the stack is known as *application module layer* with the main responsibility of executing dynamically loadable modules, referred to as *object roles*. An object role specifies all or part of a functionality of an object and also provides an interface to tap into this functionality. Object roles are semantically described and are akin to services running on a device. An object is not limited to single role and may contain a set of roles executed in parallel (especially, objects with higher processing capabilities—e.g. PDAs).

Application module layer also includes a set of *role dependency rules* that determine the dependencies of a particular role on information generated from other roles, such that the role in question can be fulfilled. For example, in a health-care environment, a PDA in fulfilling its 'patient's state' role may be dependent on information generated from other objects such as body worn temperature sensor, blood pressure sensor and/or heart monitor where the PDA can automatically retrieve information from these devices when in proximity.

However, information from other object roles has to be semantically understood in order to be utilised. Therefore, *context evaluation rules* determine the semantics of the exchanged information between the dependent roles of various objects. The context evaluation rules in effect infer a deduction through processing and composition of the information received from dependent roles of various objects. In the example above, the PDA will engage the context evaluation rules to process information from dependent object roles in order to deduce the patient's state.

3.1.2. Object network self-organisation

As stated above, object network constituents have the ability to self-organise in order to create a hierarchical, tree-like structure based on the local object-to-object interaction rules. Typically, low-level sensors will sit at the bottom of the hierarchical structure while the leader object, chosen as a result of the election process, will be at the top (refer to Fig. 2). The leader object's main responsibility is to evaluate the activity of the user.

The initial structure-less object network, converges into a hierarchical structure, is based on the consumer/provider relationships that are established on the basis of local role dependency rules. This in turn enables creation of Context Zones depicted in Fig. 2.

Before we delve into description of Context Zones, we first describe the concept of consumer/provider roles that steer the establishment of Context Zones.

3.1.2.1. Consumer/provider role dependencies. Prior to object network self-organisation process taking place, the objects must select other objects in vicinity to create relationships with, which serve as a basis for self-organisation. These relationships are specifically established between objects' roles and enable an object role to select a set of objects in vicinity in order to create an overlay link with the associated roles. Creation of these overlay links is driven by consumer/provider role dependencies.

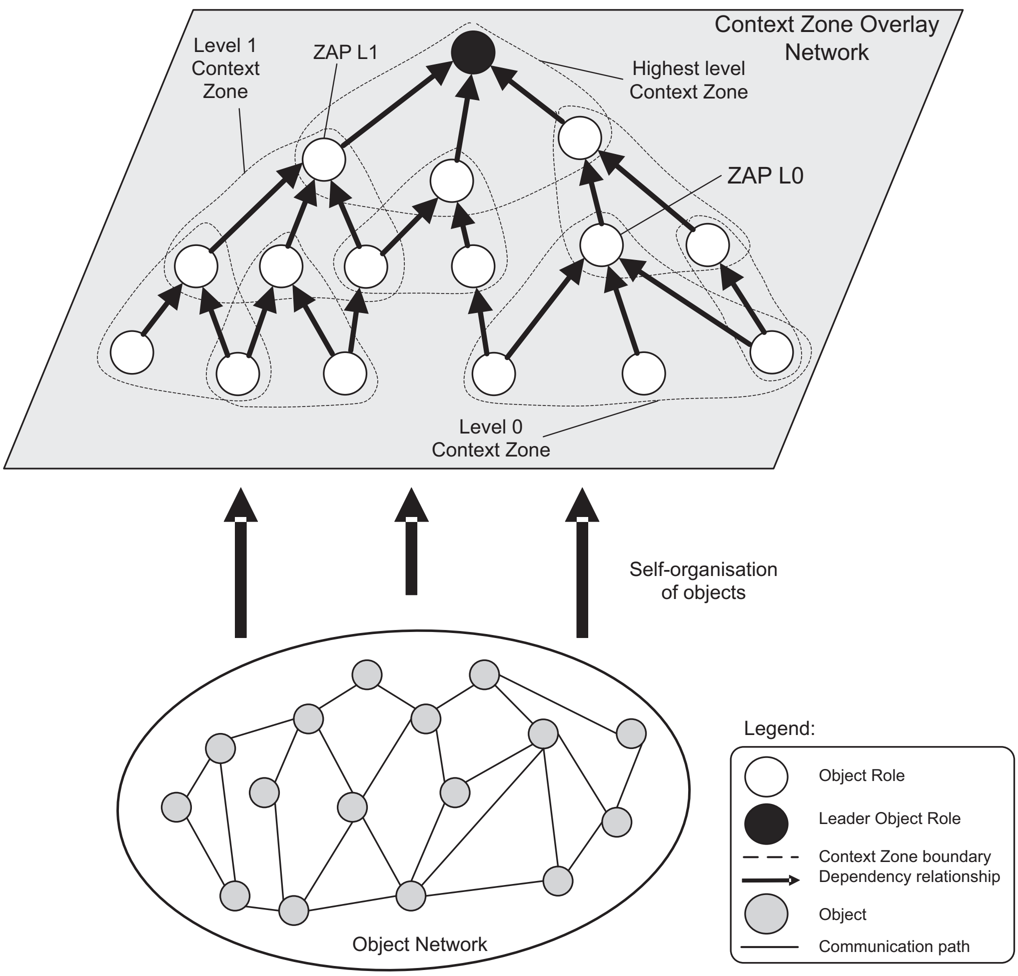

Fig. 2. Hierarchical Context Zone structure.

Typically a role may be an *information consumer role*, *information provider role* or both. The role(s) that each object can take depend on object's processing capability and functionality. Low-level sensors adhere to information provider roles only since sensors are self-sufficient and typically do not require information generated elsewhere. The leader object houses the ultimate consumer role that gathers information generated throughout the hierarchy in order to deduce user activities. All other object network roles positioned in between, are *both* consumer and provider roles to enable information flow from the lowest level of hierarchy up to the leader object.

3.1.2.2. Role dependency establishment. A consumer role's objective is to fulfil its dependencies; therefore, during object network formation each consumer role broadcasts a semantic query to other objects within a specific hop count. The query is based on the role's dependency rules and contains semantic description of the information required by the role so that it can be successfully fulfilled. In the example above, the PDA engaging the '*patient state*' role may have role dependency rules stating that this role requires information about temperature, blood pressure and heart monitor data. In order to fulfil its role, the PDA sends a semantically described query to other objects in the vicinity requesting roles that can provide information about temperature, blood pressure and heart beat information for a particular patient to respond back. If there are objects in the vicinity that house roles that can fulfil these requirements a response is sent back. This response reaches the querying object and a dependency relationship is established between the respective roles, i.e. the 'patient state' role and the corresponding objects' information provider roles (e.g. temperature, blood pressure or heart monitor data roles). If no response is received the PDA will broaden its query radius by increasing the number of hops so that the query can reach a larger number of objects until a response is received, otherwise the role cannot be fulfilled.

3.1.2.3. Context Zones. Establishing dependency relationship between roles through the consumer/provider relationship allows the formation of Context Zones, which are the main ingredient in the emergence of a global hierarchical object network role structure (see Fig. 2). The idea of Context Zones has been inspired from research work on Semantic Overlay Networks (Crespo and Garcia-Molina, 2002; Klein et al., 2003). While, the goal of semantic overlay networks is to closely group semantically similar services in order to optimise routing of search queries, our aim is different. We seek to create an efficient hierarchical object network overlay structure to enable abstraction of context information such that this information is used in activity inference process.

Context Zones are automatically created based on object-to-object interaction, where this interaction is governed by local role dependency rules. "A Context Zone is created for each consumer role and it groups together provider roles that can fulfil information requirements of the consumer role as defined in consumer role dependency rules". The consumer role of a Context Zone is known as *Zone Access Point* (ZAP) since it contains knowledge regarding semantic descriptions of all roles within the Context Zone. Before we delve into the formal definition of the notion of Context Zones, we need to begin with the basic notation.

Definition 1. (Basic notation)

An object network T is a set of artefacts having communication capabilities such that each artefact $o_i \in T$ can exchange information with another artefact $o_j \in T$ using an underlying pre-agreed or translated protocol (indices i and j are used to distinguish between different objects).

An object network bears a high similarity with a peer-to-peer ad-hoc network where no network structure is imposed and each node (an object) has the capability to sense other nodes in the close proximity and query the services (roles) they provide. An object role r_n^C of an object $o_i \in T$ is defined as

$$o_i \langle r_n^C \rangle.$$

Here, o_i represents the object where the role is housed, top index of the role indicates whether the role is a consumer role (C) or provider role (P) while the bottom index n indicates a single role from the total roles N of object o_i, and is used to distinguish between multiple roles within the same object. In our example above, the object o_i may be the PDA having 'patient state' role r_n^C as an information consumer role.

For each consumer role, a *dependency relationship* d ($\xrightarrow{d}$) is established with one or more provider roles as determined by the role dependency rules. The set of all *provider* roles, where a dependency relationship is established, is denoted with Θ and is called the *dependency set*

$$o_i \langle r_n^C \rangle \xrightarrow{d} \Theta \bullet \left(o_j \langle r_n^P \rangle, o_{j+1} \langle r_n^P \rangle, \ldots, o_k \langle r_n^P \rangle \right) \in \Theta.$$

In this instance the 'patient state' role of the PDA will establish dependency relationships with the respective provider roles of the body temperature, blood pressure and heart monitor (in this case, single role objects) as defined in the 'patient state' role dependency rules. Once the dependency relationship is established, the provider roles are added to the 'patient state' dependency set Θ.

Definition 2. (Context Zone)[1]

Based on the notation described above, we can now formally define the concept of Context Zone. A Context Zone groups together object roles that can fulfil information requirements of a consumer role. At the same time a Context Zone encompasses the consumer role along with the provider roles from the dependency set of the consumer role. The definition of a Context Zone Z_s is as follows:

$$Z_s = \left(o_i \langle r_n^C \rangle, \Theta \right) \Leftrightarrow o_i \langle r_n^C \rangle \xrightarrow{d} \Theta.$$

A Context Zone is represented as a tuple containing the consumer role $o_i \langle r_n^C \rangle$ and a set of roles contained in Θ with which a dependency relationship has been established.

Context Zone is complete when the dependency set of the consumer role has been satisfied, in other words the dependency set Θ contains all the roles specified by the consumer role dependency rules. For example, the 'patient state' role will establish a Context Zone with its dependencies, namely roles that provide body temperature, blood pressure and heart monitor information.

At the same time, each role of an object can take the status as the ZAP and is denoted[2] as (see also Fig. 2)

$$o_i \langle r_n^C \rangle \perp Z_s.$$

[1] In what follows, we use the $\bullet$ symbol to denote "such that" which corresponds with Z formal specification language conventions (Spivey, 1992).

[2] The $\perp$ symbol is not used in the mathematical context.

Following our example, the ZAP in this instance is the consumer role, namely the 'patient state' role since it contains information about other roles (temperature, blood pressure and heart monitor roles) within the Context Zone.

Creation of multiple Context Zones results in an emergent hierarchical object network structure, where the highest position in the hierarchy is taken by the leader object. The ZAPs of Context Zones can be further self-organised to support higher level Context Zones of the hierarchy. The Context Zone created by the 'patient state' role along with other roles, for example, 'medical history analysis' or 'disease knowledgebase', may become part of a higher level Context Zone created by say 'therapy recommendation' role. In this case, the 'patient state' role becomes a provider role, whereas it was a consumer role within its Context Zone. The emerging hierarchical structure comprises a number of levels, determined by the number of the Context Zone levels created which we now formally define.

Definition 3. (Context Zone levels)

Self-organisation of object network leads to formation of multiple Context Zones. These Context Zones can fit in specific levels of the hierarchy, governed by roles' information dependencies. We formally define Context Zone levels with the following: for a Context Zone q at a level x, Z_q^x, the following must hold true:

$$Z_q^x = \left(o_i\langle r_n^C\rangle, \Theta\right) \Leftrightarrow \left(\forall r_m^P \in \Theta, \exists o_n \in T \bullet o_n\langle r_m^P\rangle Z_u^{x-1}\right).$$

That is, all the dependencies of the Context Zone (Z_q^x) ZAP, contained in the Θ set must be ZAPs of lower level Context Zones Z_u^{x-1}. However, there are two exceptions to this rule:

(i) *Level 0 Context Zone* that states that each of the dependencies of the Context Zone ZAP must be roles that have no further dependencies; and
(ii) *Leader object Context Zone* that states that the Context Zone ZAP must not be part of another Context Zone, that is the ZAP cannot assume a provider role. This then ensures that the leader object Context Zone sits at the highest level of the hierarchy.

Thus far, we have formally defined the concepts that enable object network self-organisation from local object-to-object interactions. The basis for establishment of dependency relationships lies in matching semantic queries sent by consumer roles with the information offered by provider roles within the object network. Therefore, a successful matching of information needs of a consumer role with the roles providing the requested information can only be achieved if there exists a semantic description of requested and provided information, which in our case translates to semantic role description.

3.1.2.4. Semantic role description. As can be seen from the above description, the basis of Context Zone formation is the consumer/provider dependency relationship between roles. However, matching the needs of a consumer role with the information generated by provider roles requires role semantic description. Role semantic description allows computation of semantic similarity between the requested information and the provided information. Much work has been done in area of semantic description of services (for an overview see Cabral et al., 2004). Therefore, we can use any of the methodologies proposed in the current literature on semantic service description ranging from service taxonomies to Ontologies to describe object roles.

However, rather than opting for a particular semantic role description, we have chosen to make the process of object network organisation to be independent of any specific semantic role description. In our solution, inspired from Klein et al. (2003), the self-organisation process is able to handle any semantic role description, where the only requirement is that the semantic role description of choice must support a semantic distance function. The distance function is used to calculate the degree of semantic similarity between a set of roles. The output of the semantic distance function determines whether the needs of a consumer role are closely enough matched with a provider role, effectively determining whether a dependency relationship can be established between a consumer role and a provider role.

One of the main reasons for taking this approach is that we feel it is highly unlikely that one semantic description will be agreed upon across diverse domains. Therefore, pre-selecting a specific semantic role description would severely limit the applicability scope of object networks. In addition, diverse domains have specific requirements that may render a particular semantic role description unsuitable for a specific domain.

A generic representation of the distance function is as follows:

$$dist(o_i\langle r_n^C\rangle, o_j\langle r_n^P\rangle) \rightarrow \delta.$$

The semantic similarity coefficient δ is calculated between an information query from a consumer role r_n^C housed in the object o_i and information from a provider role r_n^P housed in a different object o_j. There are numerous examples of semantic service descriptions that meet our criteria in providing the distance function. For instance, in service taxonomies the distance function can be defined by counting the number of edges traversed when getting from one service description to another. In case of more complex semantic service descriptions such as Ontologies utilizing OWL, work already exists to calculate the distance function (see Shancheng et al., 2006, for an example).

Clearly, the implementation of the distance function is highly dependent on the semantic role description of choice and also the domain in which an object network has been deployed. However, in general terms, we envisage that a *semantic boundary value* β will be defined to indicate the similarity threshold, such that two roles are semantically similar if $\delta \leqslant \beta$ and thus allowing dependency relationship to be established.

3.2. Decision Module

DM is one of the main components of the overall architecture, and is executed by the leader object with the responsibility of deducing user activities. At the same time, the DM constantly adapts and refines the AM to the dynamics of the user behaviour. Deduction of activities is performed on the basis of information pertaining to user actions. This information is generated from the low-level sensors and processed throughout the object network hierarchy up to the leader object to be utilised by the DM.

The DM, illustrated in Fig. 3, is an orchestration of two chief components, namely the AM that serves as a repository of user activities and the Rule Engine that enables continuous adaptation to user behaviour changes through refinement of the AM. Watchdog Timer is used to detect sporadic events that may occur due to users shifting focus.

The AM is a repository that is specific to a user within a domain and stores activities that a user has performed. The concept of AM is based on the idea that users typically perform

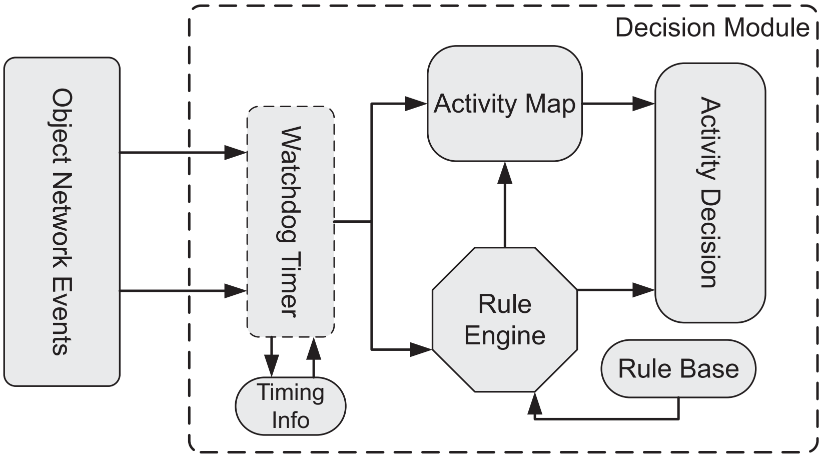

Fig. 3. Decision Module.

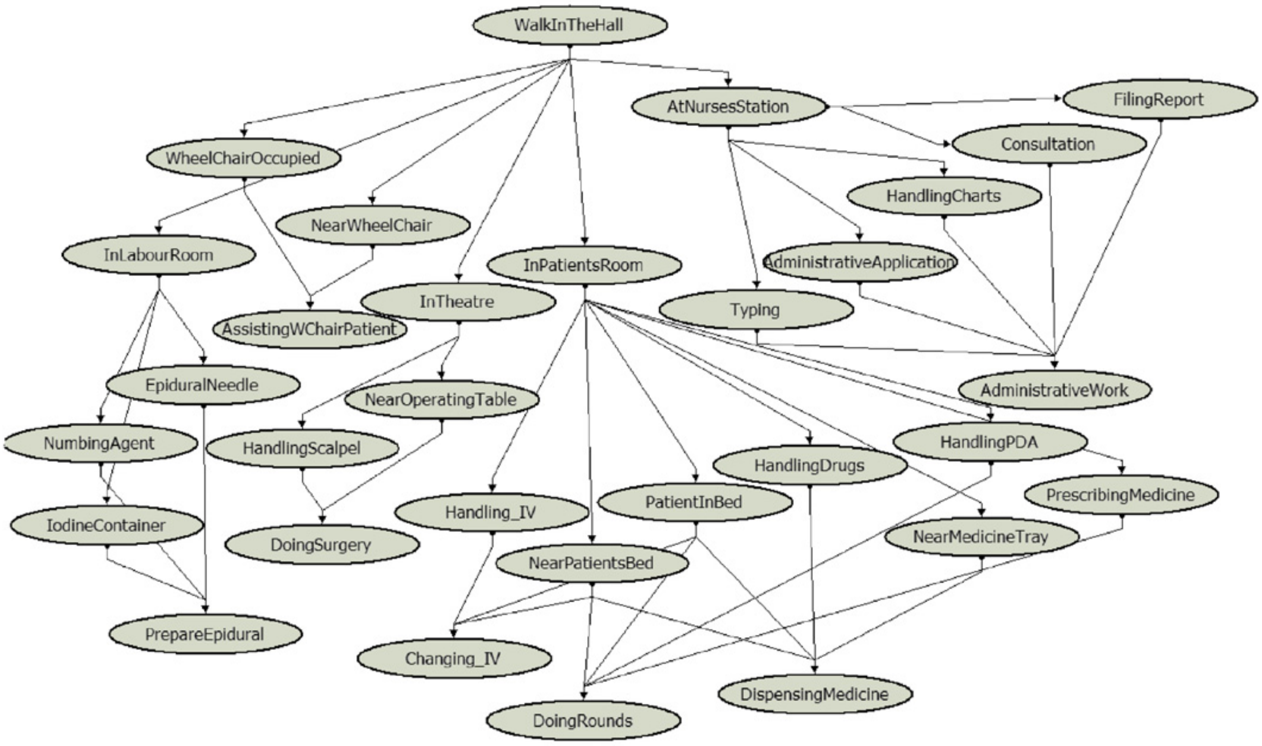

Fig. 4. Part of a user's AM showing DBN.

activities drawn from a finite set in order to achieve a particular goal. An example of part of an AM is illustrated in Fig. 4. The AM is stored centrally within the user's working environment akin to user's profile (for example, within the hospital server). Upon creation of an object network surrounding the user, subset of the AM is retrieved that corresponds to the activities that can be performed within the user's current domain, for example, within a specific hospital ward.

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). The AM represents the relationships between an activity and a *causal*. Causals are events from the object network that are used as evidence in inferring the activity a particular user is engaged in. *Causal set* determines the facts that have to be true for an activity to take place. Received information from the object network is essentially evidence that is constantly fed to the DBN to infer a set of likely activities, where we apply the Junction Tree Algorithm.

The DBN requires constant refinement such that it can adapt to dynamically changing user behaviour. The refinement process involves updating the Bayesian Network belief values across nodes. This process is determined on the basis of user behaviour such that the DBN reflects the changes in this behaviour.

We observe user behaviour through the events from object network in order to obtain a set of sequences. A sequence represents an ordered set of events φ raised as a result of user actions while performing a specific activity A. A sequence is denoted with $\lambda_A = \langle \varphi_1, \ldots, \varphi_n \rangle$ and reflects user behaviour's actions when manipulating objects. For example, a sequence is created when a nurse manipulates the following objects in order: Iodine Container, Numbing Agent and Epidural Needle (refer to Fig. 4). We analyse the set of obtained sequences and select the most likely sequence for an activity. The most likely sequence is the one that has occurred most frequently during our observations, denoted with λ_A^1 which represents the most typical user behaviour. For instance, in the example presented above, the nurse may handle the three objects in this sequence (Iodine Container, Numbing Agent and Epidural Needle) 85% of the time when preparing an epidural procedure. Upon selection of the most likely sequence, we then extract the activity causals which are used to refine the DBN.

Therefore, considering the most likely sequence with n number of causals we refine the DBN as follows:

$$\forall \varphi_n \in \lambda_A^1 \bullet n \in (1 \ldots \#\lambda_A^1) \rightarrow P(A|\varphi_n).$$

For every object network event (causal) φ in the most likely sequence, we increase the DBN conditional probability of the node that corresponds to that causal with respect to activity A where the $P(A|\varphi_n)$ is refined. The refinement process thus allows DBN to dynamically adapt to user behaviour changes as we show in Section 5.

However, there are times when a user may start an activity that s/he may not always complete. In fact some activities may only be started, for example, by manipulating the initial object, and not completed due to user changing their focus attention (e.g. a sudden emergency). For instance, in a simple activity ‘making tea’, the user may intend to prepare a tea by taking the cup out from the cupboard when the door bell rings, thus making tea activity is not completed or is left hanging. Clearly, the activity recognition system should not attempt recognising ‘making tea’ activity, but rather temporarily terminate the activity since it was interrupted.

In order to detect these sporadic events and also increase the activity recognition accuracy, we have incorporated a minimum duration value for various causals implemented through the *Watchdog Timer* component in Fig. 3. The minimum duration value is particular to a casual and is not a constant. This value will differ for each user, based on the user's behaviour. Therefore, when we observe the sequence of user actions, we also record the amount of time the user spends manipulating a specific object or the time between object network events, stored in the *Timing Info* module in Fig. 3. These records are then used to adjust the minimum duration value for each casual. When an event from the object network indicates that the user is performing an action, for example manipulating an object, we do not start the activity inference process until the minimum time has elapsed. An exception to this rule occurs when the user generates another object network event before the minimum time expiration. If the second object network event is a causal belonging to the same activity as the first causal, then the activity inference process can proceed, since the change in time indicates a variation in the user behaviour. This variation is then reflected back into adjustment of the minimum time for the first causal. Otherwise a sporadic event has occurred and thus the first causal is ignored. The second causal is now regarded as the start of another activity having the same rules applied regarding the minimum time. The process is then repeated for each new and subsequent causal. As such the minimum duration rules in conjunction with the DBN refinement ensure an accurate and adaptable activity inference process.

4. Collaborative activities

Technology already plays an important role in our everyday lives, while humans continue to be social creatures that aim to interact in multimodal ways. Human interaction implies existence of activities that are not performed in isolation; rather they require cooperation amongst interacting users. These activities play a significant role especially in working environments, thus researches have investigated various ways in which technology can be used to support these activities. A dedicated research field called Computer Supported Collaborative Work (CSCW) has risen as a result.

CSCW has been investigated extensively in the last two decades (Grudin, 1994). The idea is based on studying the manner in which technology can support users in their work and has been defined by Peter and Carstensen (1999) as addressing ''how collaborative activities and their coordination can be supported by means of computer systems''. CSCW emerged as a result of realising the vast potential of technology and its advances in enabling coherent and effective conduction of work activities. It has been applied to large number of domains, from teleconferencing (Quemada et al., 1996) up to commercial aeroplane production lines (Laborie et al., 2005). Our solution for activity recognition can support CSCW, where a typical interaction of CSCW environments participants is illustrated in Fig. 5. User interacts with the various technologies which, based on the mechanism described in the previous section, have the ability to infer user's activity. This interaction provides a window to the 'world' of other participants enabling exchange of information pertaining to the collaborative activities.

For instance, CSCW used in aeroplane assembly lines requires a high degree of coordination such that the final product is assembled with high precision. Therefore, continuous knowledge of the staff activities provides a solid basis to monitor the assembly process and schedule various tasks such that the overall process is completed in the shortest possible time.

While CSCW environments have a wide applicability scope, we focus our paper in the health-care domain. We aim to show how activity recognition can support health-care CSCW environments to improve the efficiency of the day-to-day collaboration activities of doctors and nurses. In the current practice, a nurse or a set of nurses typically follow the doctor while s/he interacts with the patients and take notes of the doctor's recommendations (Sweet and Norman, 1995). Then these recommendations are carried out for each individual patient based on the urgency. In some practices, the doctor may give his recommendations solely to the head nurse which then allocates various tasks to other nurses. In either case the nurses' time is not efficiently utilised. For example, the idle time

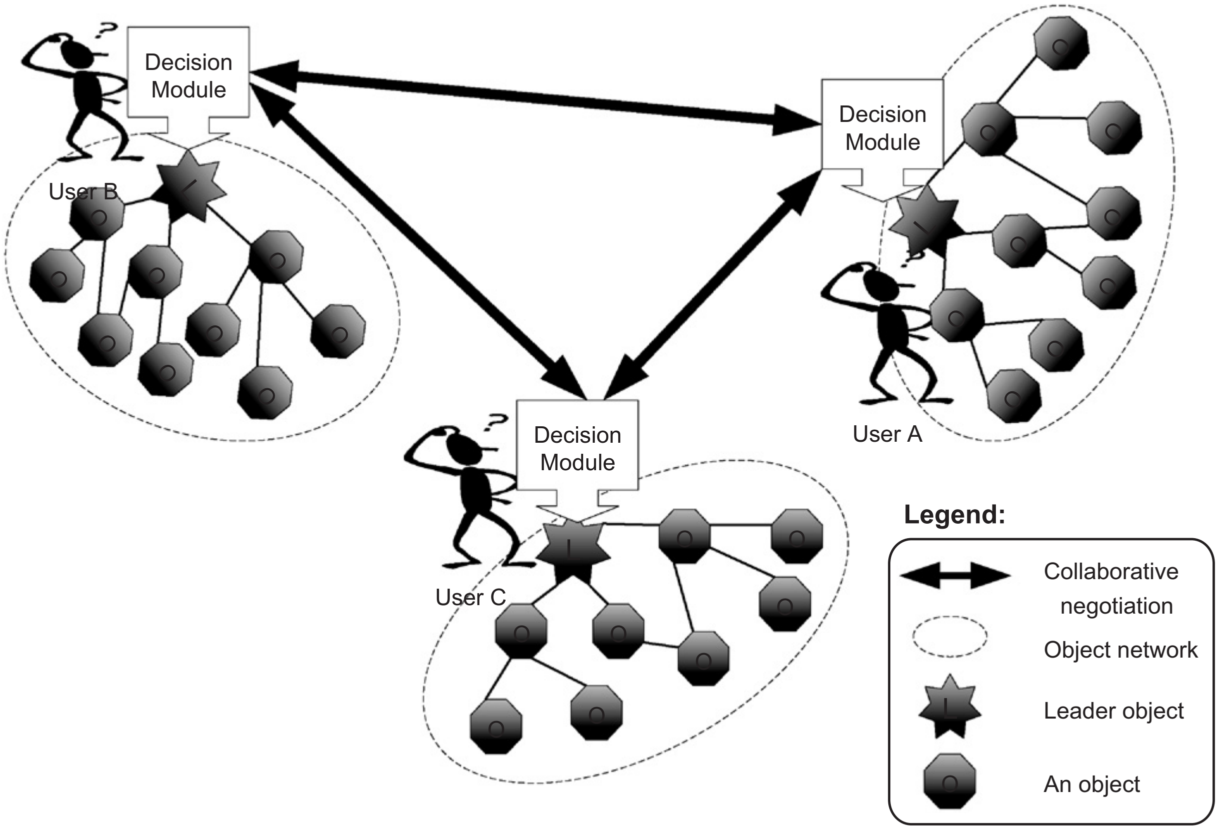

Fig. 5. Sample interaction in a CSCW environment.

while the doctor is making diagnosis could be spent on other activities in patient care. Also in case of multiple doctors the head nurse may be under pressure to note all the recommendations given, therefore, decreasing response time and increasing the chance of errors in patient treatment.

Our system, on the other hand, can utilise the activity recognition system to deduce each individual activity and through collaborative negotiation protocol support remote virtual collaboration. In this instance, the doctor does not allocate a task to a specific nurse or request the head nurse to do so; rather the doctor's request is allocated seamlessly to a specific nurse that is selected based on the various activities of the various nurses.

The activity of each nurse is provided by our activity recognition system through inference of medical staff activities. Therefore, a doctor's requests can be routed to an appropriate nurse without a need for a mediating person or requiring a nurse to shadow a doctor during his activities. Based on this transparent concept, when doctor is ready to make a request for a particular patient (e.g. dress the wound), he simply enters the appropriate information on his mobile device. This request is then allocated to a nurse. The decision as to which nurse receives a particular request is agreed after considering a number of activity metrics.

Clearly, a naïve scheme of allocating the doctor's request would be to simply select a nurse based on her current activity, for instance the nurse with the lowest activity priority (activity priority value is assumed to be predetermined). However, such scheme may bias the workload towards less experienced nurses, since nurses with high experience tend to carry out high priority activities. Also doctor's requests may be such that they must be carried out by an experienced nurse. Therefore, our devised solution takes into account three primary factors, namely experience, workload and activity priority. These factors were chosen primarily because they correspond to the overall work state of a nurse and also they can be calculated directly from the activity recognition process. The workload metric can be obtained by measuring the number of activities a particular nurse has carried out during her shift, while experience is a long-term monitoring of nurse's activities. The actual allocation process and the associated sequence diagram will be elaborated on in the next section.

4.1. Request allocation process

The process of allocating a doctor's request to a nurse essentially translates to an optimisation problem. The objective of the optimisation function is to maximise the efficiency of the medical ward within the resource constraints imposed by the number of nurses available. We define efficiency as the ratio of the sum of currently attended activities' priority values with the total number of activities' priority values as follows:

$$\varepsilon = \frac{\sum A_n^R}{\sum A_n^R + \sum A_n^S}, \tag{1}$$

where A_n^R represents an activity that is currently performed by a medical staff, and A_n^S represents a suspended activity waiting to be attended to. In order to calculate the efficiency we have defined a virtual *activity queue* that holds information about activities that nurses and doctors are currently performing in addition to activities that have been suspended or any new activities that are generated due to a doctor's request. In other words to ensure the highest efficiency, the allocation process's objective is to suspend only the activities with lowest priority values subject to available resources and constraints associated with these resources, such as nurse workload and experience.

Once the request is entered, the doctor's mobile device broadcasts a request information about activities of the nurses in the vicinity. This broadcast message is transmitted to the leader object of each of the nurses' object network that is in close vicinity to the doctor. The leader object of the various nurse's object networks will respond back to the doctor's mobile device of their current activity. Upon receiving this information, the mobile device uses a linear objective function to select the most appropriate nurse. This selection is based on three criteria incorporated into the objective function as follows:

$$v = \alpha E + \beta(1 - W) + \delta(1 - P), \tag{2}$$

where E, W, and A represent nurse experience, workload and current activity priority, respectively, along with their weight coefficients α, β, and δ. Combining these values into the objective function yields an availability value v that measures the degree of readiness of a nurse to cater for doctor's request. Clearly, not all activities performed by a nurse can be interrupted. Activities with high priority, for instance assisting in a surgical theatre, cannot be interrupted; therefore, we have defined an activity priority threshold. Activities that are above the threshold cannot be interrupted and the threshold value is represented as another constraint in the objective function.

Once the nurse's leader object accepts the request from the doctor's mobile device, an alert is sent to the nurse's mobile device to suspend her current activity and process the new activity.

However, as soon as the nurse's activity is suspended, the efficiency value ε is lowered by the activity priority value of the suspended activity. Due to this decrease, the allocation process must ensure that another nurse with a lower activity priority value takes over the suspended activity. Replacing the suspended activity with another activity of lower priority has the effect of increasing the efficiency, thus bringing the ε ratio closer to the peak efficiency. However, since the second nurse is assigned to take over a lower priority suspended activity, the alert sent to the nurse's device will only require the nurse to perform the activity once her current activity is completed. This will avoid blindly asking nurses to change from one activity to another frequently, which incurs a large overhead primarily in the time spent changing from one activity to another which may also have unpleasant effects on the nurses caused by the frequent activity changes.

The overall allocation process is illustrated through an example where the sequence diagram is shown in Fig. 6.

The sequence diagram highlights the tasks that have to be accomplished to respond to doctor's request and allocate activities to available nurses. The doctor has just finished reading the latest medical headlines on his wall screen and is now preparing for his daily routine of patient assessment of medical conditions. During his rounds he visits each patient individually and based on the physical examination and patient's monitoring history he can recommend various actions to be taken, such as altering the current therapy or various recovery tasks. The activity recognition system has inferred the doctor's activity

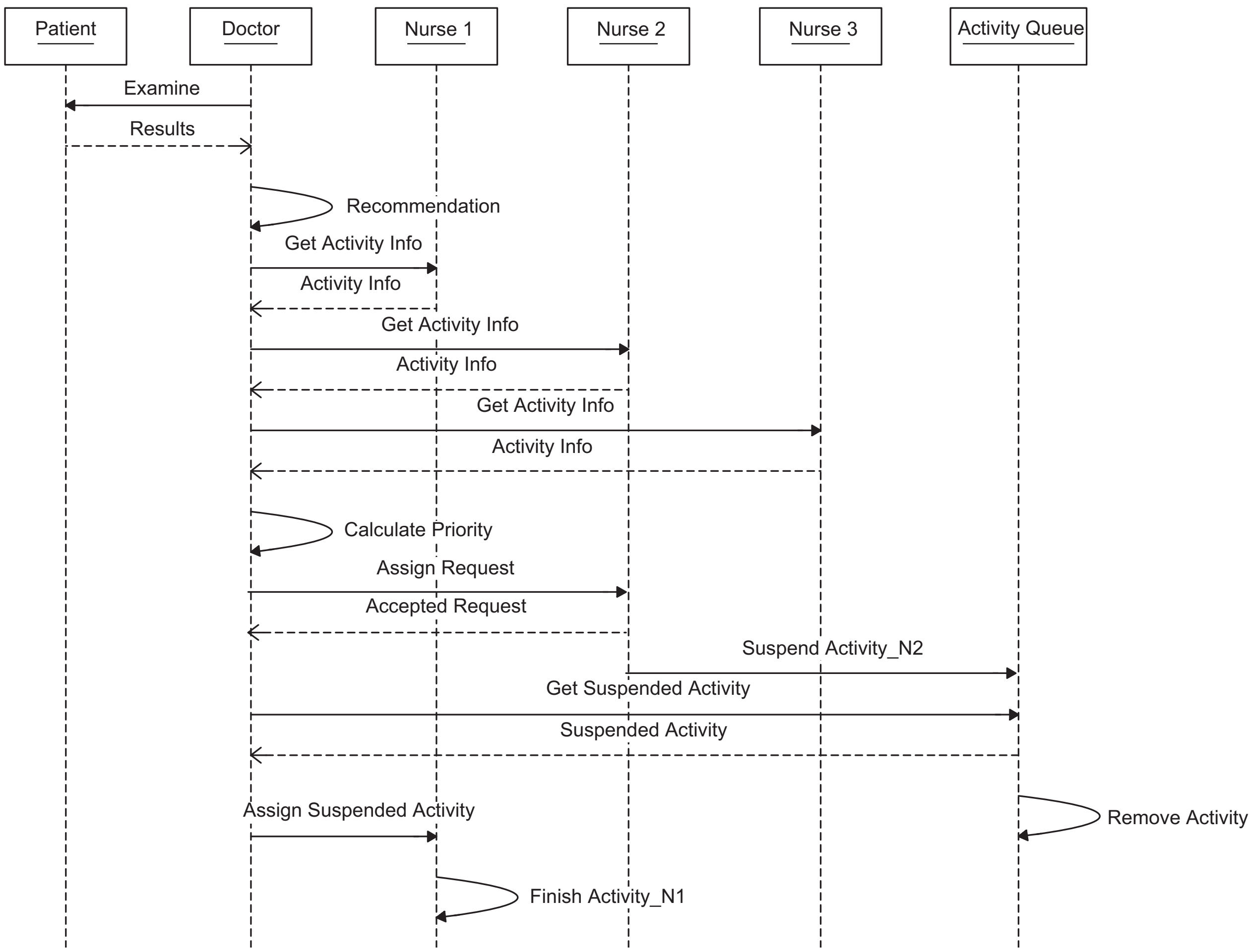

Fig. 6. Sequence diagram of the allocation process.

of '*doing rounds*' based on events from his surrounding object network (e.g. handling PDA and moving through Zones of patient's bed). He is currently examining a patient that has been in the hospital for a number of days after being involved in a serious motorbike accident. The vital signs are stable; however, his leg wound is not healing properly. The doctor notices areas of necrotic tissue around the wound. He enters in his mobile device the immediate request to redress the wound after treating it with Dakin's solution. His request is of high priority since if left untreated the patient runs the risk of developing sepsis. This action signals the doctor's mobile device to send for a nurse that can complete this emergency request. Clearly, at this point the efficiency has been lowered since there is new unattended activity added to the queue. Incidentally, doctor's mobile device has broadcast the request to all DMs of nurses' leader object within a particular radius. The DMs of the nurses within vicinity respond with the current deduced activity from the activity recognition system. This allows doctor's mobile device to subscribe to the updates of current activities of the nurses in vicinity in order to select a nurse to complete his request. This information shows that Nurse 3's availability is the highest. She is currently preparing a patient for an epidural. Since this procedure cannot be interrupted the request is not sent to Nurse 3, due to the fact that her activity priority falls above the threshold. Nurse 2 is currently examining the IVs of the patients in the intensive care ward. Her activity has lower priority than the activity of Nurse 3 and also due to her experience Nurse 2 is chosen to carry out doctor's request. She has just finished replacing the catheter from the hand vein of a patient when she receives the request to attend to doctor's recommendations. She clicks the accept prompt on her mobile device. However, doctor's mobile device has inferred that Nurse 2 will not complete her activity for all patients in the ward and puts her activity in a suspended state while attempting to find another nurse that can take over the remaining patients. At this point the efficiency slightly increases, since the doctor's request is now removed from the activity queue and Nurse 2's activity is put on the queue. The increase in efficiency stems from the fact that Nurse 2's activity has lower priority value in comparison with doctor's request. The newest update of nurses' activities reveals that Nurse 1 is just about to finish a routine administrative work of the day. Therefore, the doctor's mobile device assigns the suspended activity in the activity queue to Nurse 1 so that she can attend to as soon as her activity is finished. At this point, Nurse 2 has left to attend to doctor's request; while a short time later Nurse 1 begins examining IVs of the remaining patients.

This scenario has presented how the activity recognition systems can help support virtual negotiation between different object networks to help determine what each nurse is currently performing and who to assign the new task to. The system mimics an invisible shadow that evaluates the users' activities automatically and performs communication in a peer–peer fashion in order to maintain the overall efficiency at the highest level possible.

5. Simulation results

We now provide the performance evaluation results of our system. Since the collaborative activities are supported by the activity recognition system, as detailed in the previous description, we have divided the results in two logical sections, namely the performance evaluation of the activity recognition system and also evaluation of the efficiency of the collaborative environments. We will demonstrate our solution through three types of simulations, namely (i) activity recognition accuracy with respect to learning,

(ii) management of Context Zones with respect to user's behaviour, and (iii) remote virtual collaboration.

5.1. Activity recognition

In order to test the performance of the system, we have analysed activities of the medical staff and recorded the manner in which these activities were performed. We then fed this data directly to the Test Agent (TA) object. Initially, the TA loads up a scenario that specifies its behaviour. The scenario controls what objects typically take part when performing an activity and their status during that time.

User actions and the resulting behaviour are encoded using a Markov Chain based model. Markov Chain models have been used to model user behaviour in a number of research projects (Hlavacs and Kotsis, 1999; Sokolov, 2003) since they allow representation of the dynamics of user behaviour. In our model, the Markov Chain *state space* is used to define the set of objects that are manipulated during the course of performing an activity. In addition, the *transition matrix* defines the probability of user manipulating an object n given that s/he is currently manipulating another object m throughout the state space. A Markov Chain exists for each simulated activity and each Markov Chain model represents the behaviour of a single user. It should be noted that Markov Chains are created based on the observation of the manner in which an activity is conducted including objects manipulated and their frequency of manipulation.

Each Markov Chain is loaded into the TA which has the responsibility to extract a *sequence* from the model. The sequence refers to an ordered set of objects that the user has manipulated while performing an activity. The probabilistic nature of the model caters for variation in the user's behaviour where the sequences obtained may differ reflecting the way that humans conduct activities—variation is almost always present. A sequence from the TA is fed to the DM and reflects a particular manner in which the user performs an activity. An important advantage of this approach is that the simulation process is entirely transparent to the object network and the leader object, since TA generates events in the same manner as they would have been generated should a real user perform a particular activity.

We now present the simulation results related to performing a number of medical activities. Our simulation consists of 20 iterations for each activity, resulting in 20 sequences—equivalent to the user repeating a particular activity. The iteration step is set to 4, which means that for every four sequences obtained, we calculate the most likely sequence which we use to refine the DBN (refer to Section 3.2). The initial structure of the DBN contains activity causals for each activity with the probability impact of each causal $P(A|c = T)$ set to a neutral value of 0.5 (A represents an activity and c is a causal of that activity). The DBN refinement is carried out by increasing the probability impact of particular causals based on the most likely sequence. Based on our chosen parameters, during the course of 20 iterations, the DBN is refined five times.

We have setup our simulation environment to recognise seven activities. However, we only show the results pertaining to the scenario, namely recognition and evaluation of four activities, depicted in Fig. 7. These simulations are carried out with respect to the DBN in Fig. 4.

Closely examining these refinements it can be seen that 'Administrative Work' activity only achieved inference probability of 0.748, despite the fact that the same simulation parameters, such as number of iterations and iteration step, were used for each activity.

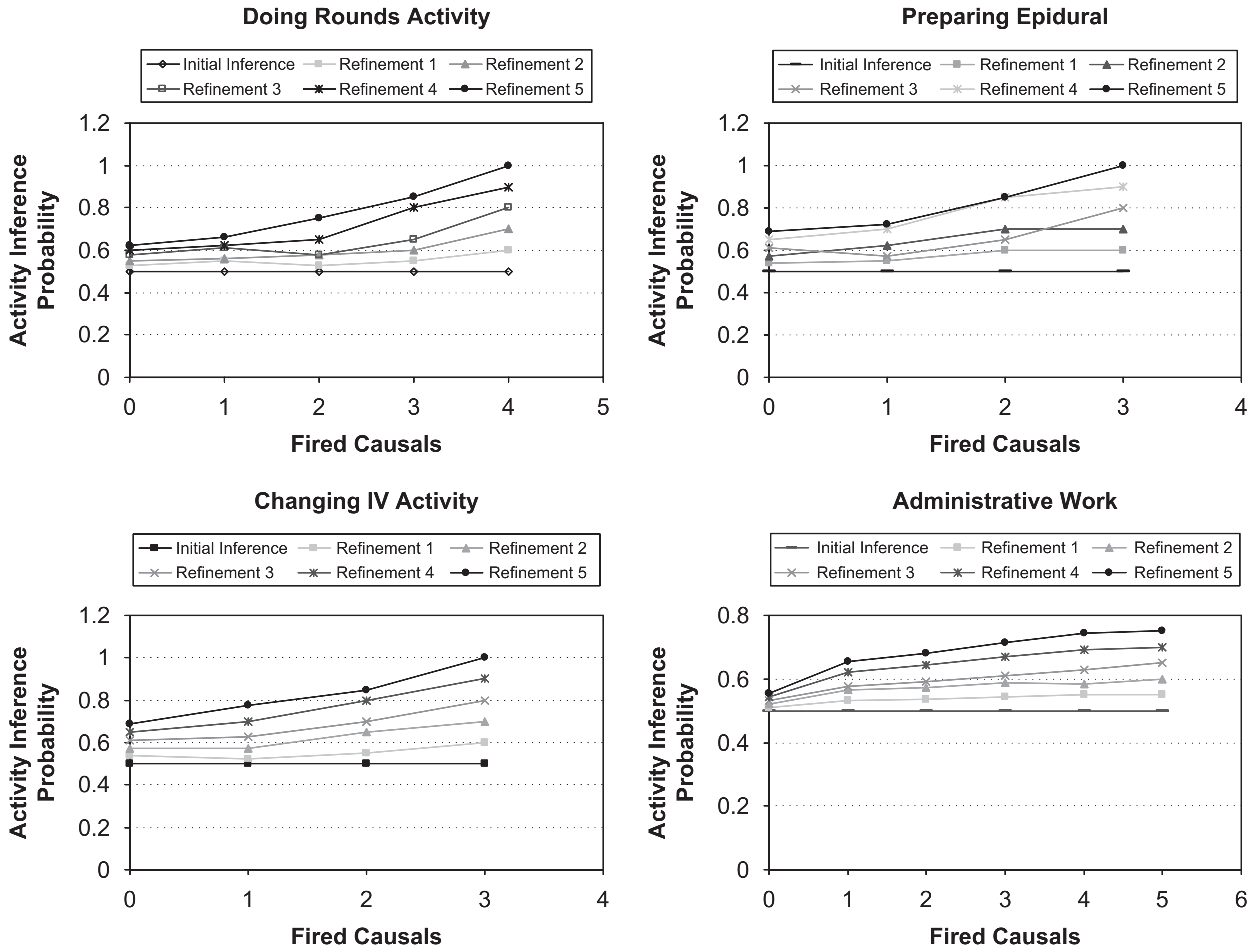

Fig. 7. Impact of DBN refinement in activity inference.

We attribute the inference probability problem to the number of causals. This activity is more general than other activities since larger number of objects is used to define this activity, as opposed to manipulating intravenous line in ‘Changing IV’ activity for example. Thus, ‘Administrative Work’ requires larger number of causals (see Fig. 4). Based on the probability of 0.748 for activity ‘Administrative Work’, the most common sequence of causals used to refine the DBN was Typing, HandlingCharts, Consultation, FilingReport, AdministrativeApplication, in this order, which occurred 80% of the time.

In our simulation setup, Administrative Work had five causals compared to three causals for Changing IV and Preparing Epidural activities or four causals for Doing Rounds activity, which meant that there was much more room for variation. In fact, theoretically this variation translates to $v = 5!$ or 120 combinations, whereas much smaller difference exists between the other activities, only $v = 4!$ or $v = 3!$ which is 24 or 6 variations, respectively. Clearly, our Markov Chain model adds predictability that reduces this variation.

As such, increasing the number of iterations and the number of refinements mitigates this problem as can be seen from Fig. 8. For the results in Fig. 8, we have increased the number of iterations from 20 to 32 such that the number of refinements has also increased from 5 to 8. Changing these parameters, we obtained the inference probability close to 0.9.

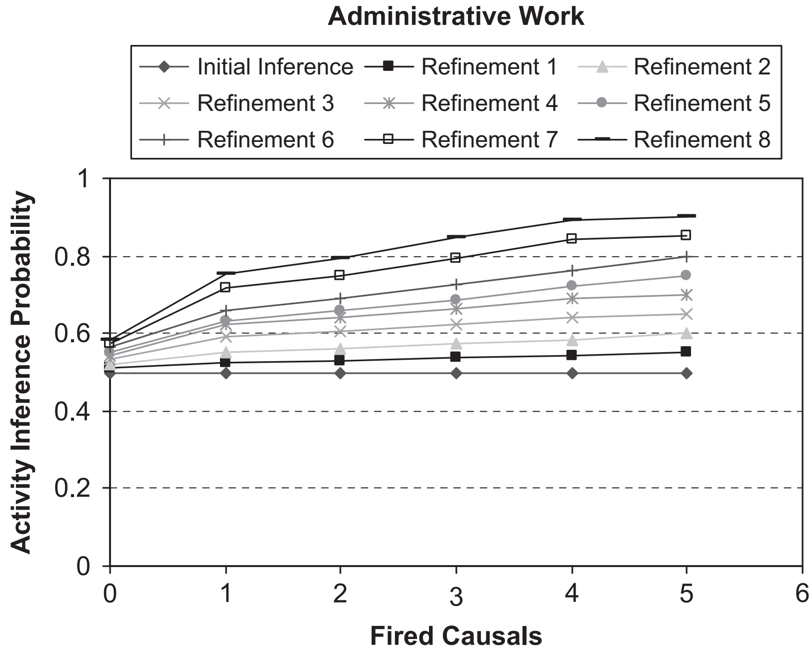

Fig. 8. Administrative work activity refinement.

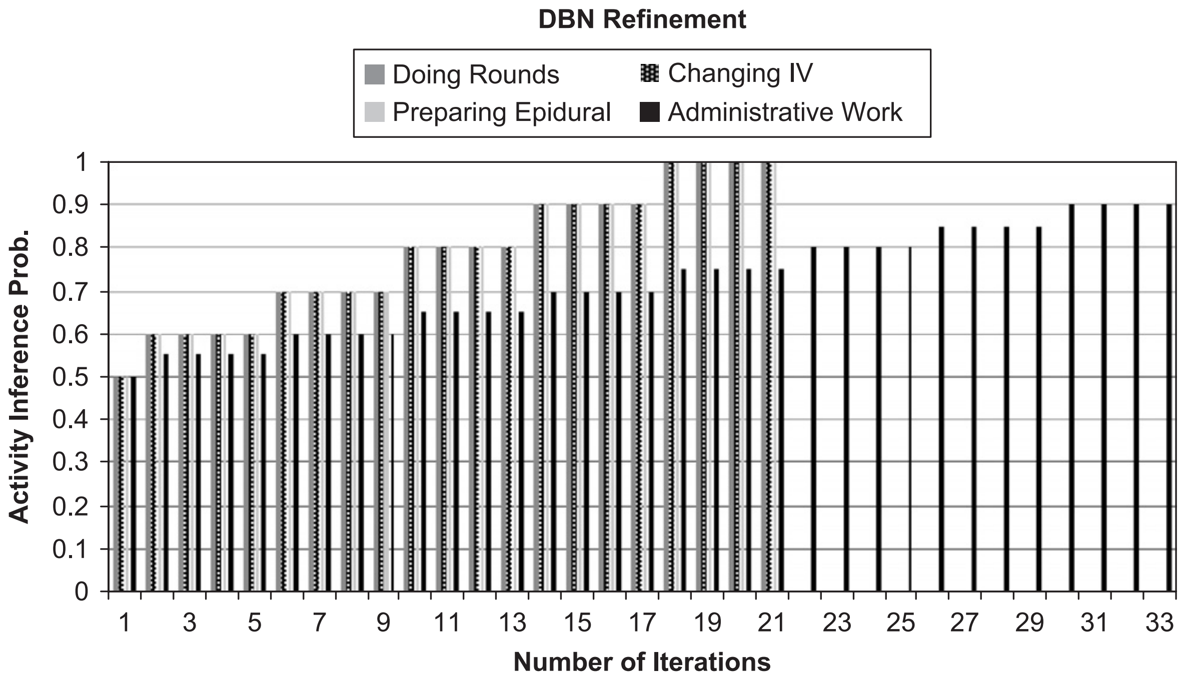

Fig. 9. Training effort required for each activity.

Our results show that some activities take longer to train than others, a behaviour that is primarily determined by the number of activity causals. The overall training effort required for each activity is shown in Fig. 9.

While the first three activities trained well to achieve satisfactory inference probability close to 1, the 'Administrative Work' activity required further training. Clearly, we could have decreased the iteration step from 4 to say 2 which would have given us 10 refinements within 20 iterations. However, the problem with this approach was that the system had to infer the most likely sequence from a pool of only 2 sequences which tended to confuse the refinement process.

5.2. Context Zone formation

Once we have refined the activities and trained the DBN, the system knows the associated objects of a particular user behaviour. This information is used to efficiently predict formation of Context Zones. Establishment of Context Zones is a prerequisite to a successful activity inference process. Context Zone formation is another part of the simulation process that is concerned with measuring the performance of the system with regards to the management of objects (e.g. sensors, actuators, PDAs) interconnected within the environment. We are primarily concerned with the number of Context Zones created as a response to user activities and we aim to reduce this number through prediction of the user's actions and hence Context Zone formation based on the most likely sequence. There are two benefits associated with this reduction. Firstly, the Context Zones can be established in advance, thus increasing the responsiveness of the activity recognition since there is no delay in Context Zone formation. Secondly, by establishing only the Context Zones that are necessary, we reduce the number of messages exchanged between sensors thus minimise the sensor energy expenditure.

Initially, we create all possible Context Zones in the environment, since the system does not have any knowledge about activities a user may perform. Then as DBN is refined, we use the knowledge about the most likely sequence to predict user's activities and establish specific Context Zones in advance. Context Zone formations are highly localised since they depend, amongst other parameters, on the physical position of the sensors. For example, a Context Zone created within the Patient's Room typically contains objects that are local to the Patient's Room domain. However, this does not preclude Context Zones from interacting with other Context Zones outside a specific domain.

With no knowledge about user behaviour the total number of Context Zones created amounts to 29 in order to recognise each of the four activities. For our simulation, we use a three level object network hierarchy, therefore the number of Context Zones is highly correlated to the number of causals.

However, as we learn user behaviour by analysing the most likely sequences, this number gets dramatically reduced as our simulation results show.

For example, as soon as the doctor walks in the Patient's Room (Fig. 10), the number of Context Zones is reduced from, initially 17, to only 6. This is due to the fact that Doing Rounds activity, according to the most likely sequence, requires formation of six Context Zones. Therefore, we create those Zones in advance in order to infer doctor's activity. Other Context Zones are created in conjunction with our prediction of medical staff activities, which is depicted in Fig. 11. The graph shows that number of Zones gets mostly reduced in the Patients' Room when a doctor or a nurse enters to carry out an activity. In contrast in Nurses' station and Labour Room, the Context Zone prediction has a lower effect. This is due to the fact that higher number of activities can be carried out in the Patients' Room compared with other locations as defined by our simulation setup. Therefore, the number of Zones created can be reduced to a higher extent in comparison with other locations. Clearly, the higher the number of activities recognised, the higher the sensor energy savings.

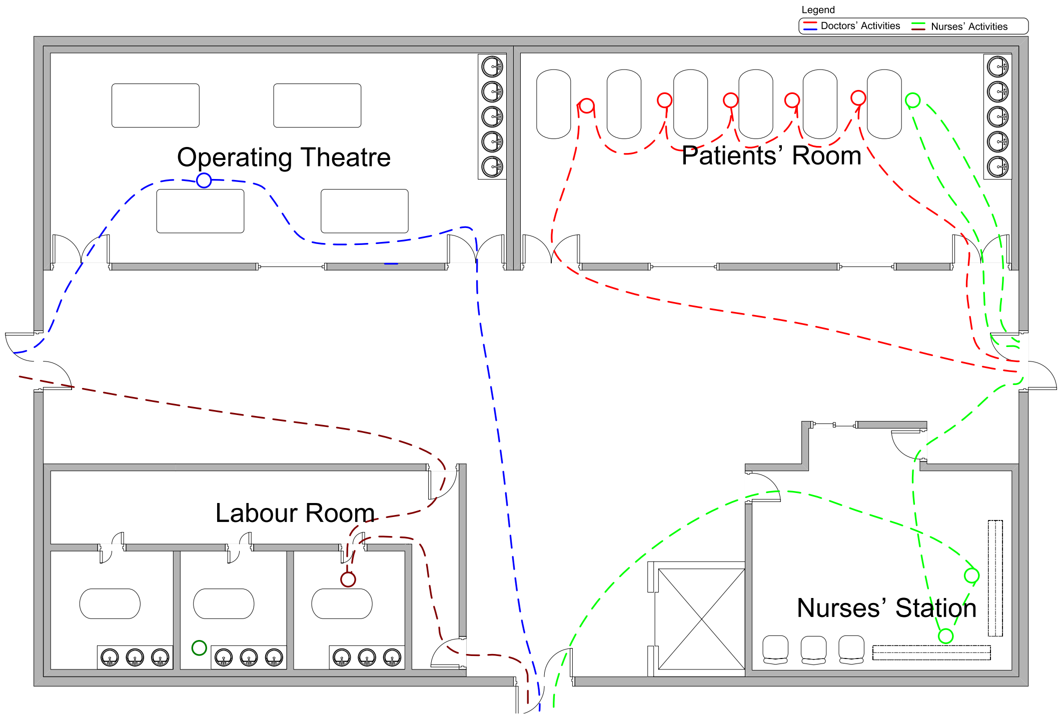

Fig. 10. Section of a hospital floor and staff mobility tracks.

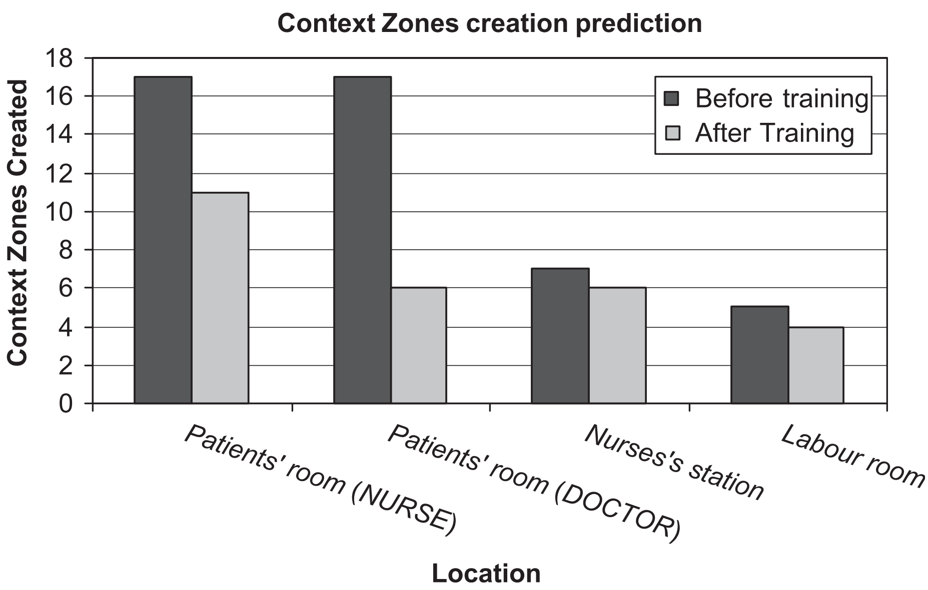

Fig. 11. Context Zones creation prediction after training.

5.3. Collaboration

We have also tested the efficiency of the doctor–nurse collaboration based on the scenario depicted in Section 4, and also based on the sequence of operations in Fig. 6. The results are shown in Fig. 12.

At the point when all the four activities are attended to (cooperative event 4), the efficiency is at the highest value. The doctor's request for the wound redress causes a new

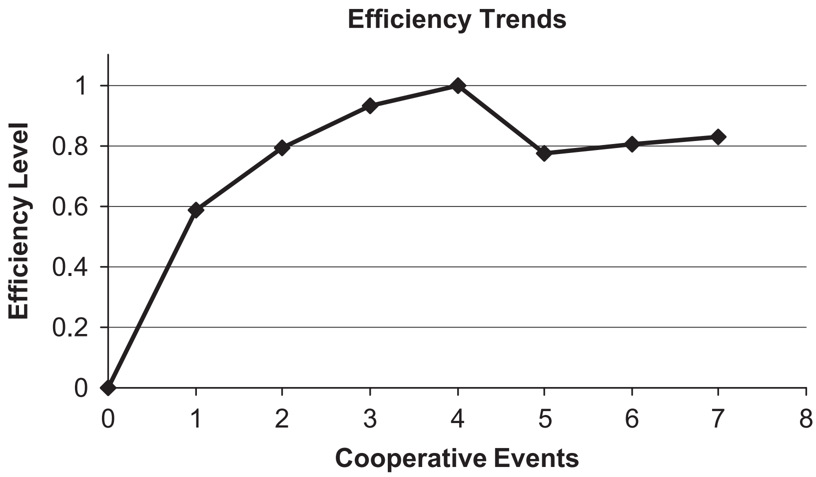

Fig. 12. Overall efficiency of collaboration activities.

activity to be created which is initially unattended (event 5). Therefore, the efficiency drops, according to Eq. (1) defined in Section 4.1, since there is a new request in the activity queue. Once the doctor's request is assigned to Nurse 2 and the nurse responds (event 6), causes efficiency increased since doctor's request has been removed from the queue while activity of Nurse 2 that has lower priority is put into the activity queue. As illustrated in Fig. 6, as Nurse 1 finishes her current activity and attends to the activity of Nurse 2 (event 7), it causes the efficiency value to increase again following a similar pattern.

6. Conclusion

Increasing costs of health-care provisioning is becoming an ever growing challenge for health-care institutions. Growing number of elderly population and other factors, such as reduced funding are contributing to this challenge. However, employing the latest technological advancements can enable these institutions to become more effective in patient care provisioning. We have shown how activity recognition process can be applied in health-care environments. We have also shown our synergetic approach in which an accurate activity recognition process supports remote working collaborative environments by routing various request to different users on the basis of their current activity.

References

Bao L, Intille SS. Activity recognition from user-annotated acceleration data. In: Ferscha A, Mattern F, editors. Proceedings of PERVASIVE 2004, vol. LNCS 3001. Berlin, Heidelberg: Springer; 2004. p. 1–17.

Bardram J, Christensen HB. Open issues in activity-based and task-level computing. In: Proceedings of first international workshop on computer support for human tasks and activities, CfPC PB-2004-60, 2004.

Bardram JE, Christensen HB. Pervasive computing support for hospitals: an overview of the activity-based computing project. IEEE Pervasive Comput Mag 2007;6(1):44–51.

Bardram JE, Hansen TR, Mogensen M, Soegaard M. Experiences from real-world deployment of context-aware technologies in a hospital environment. In: Proceedings of Ubicomp 2006, LNCS vol. 4206, 2006.

Bouchard B, Giroux S, Bouzouane A. A keyhole plan recognition model for Alzheimer's patients: first results. J Appl Artif Intell (AAI) 2007;21(7):623–58.

Cabral L, Domingue J, Motta E, Payne TR, Hakimpour F. Approaches to semantic web services: an overview and comparisons. In: Proceedings of first European semantic web symposium, ESWS 2004, LNCS vol. 3053; 2004. p. 225–39.

Centeno MA, Giachetti R, Linn R, Ismail AM. A simulation-ILP based tool for scheduling ER staff. In: Proceedings of 2003 winter simulation conference, vol. 2; 2003. p. 1930–8. ISBN: 0-7803-8131-9.

Corporation P. ProModel solutions-healthcare, ⟨http://www.promodel.com/products/edsimulator/⟩; 2007 [last accessed 14.05.07].

CreateASoft I. Healthcare process simulator, ⟨http://www.createasoft.com/processImprovementSimulator/leanProcessSimulationSoftware/SimCareProcessSimulator7.2.html⟩; 2007 [last accessed 14.05.07].

Crespo A, Garcia-Molina H. Semantic overlay networks for P2P systems. Technical report, Computer Science Department, Stanford University, October 2002.

Driver C, Linehan E, Spence M, Tsang SL, Chan L, Clarke S. Facilitating dynamic schedules for healthcare professionals. In: Proceedings of 1st international conference on pervasive computing technologies for healthcare, Innsbruck, Austria, 2006.

Favela J, Tentori M, Castro LA, Gonzalez VM, Moran EB, Martínez-García AI. Estimating hospital workers activities and its use in context-aware hospital applications. In: Proceedings of first conference in pervasive healthcare, Salzburg, Austria, 29 November–1 December 2006.

Gavrila DM. The visual analysis of human movement: a survey. Comput Vis Image Understanding 1999;73(1): 82–98.

Grudin J. Computer-supported cooperative work: history and focus. IEEE Comput 1994;27(5):19–26.

Guralnik V, Haigh KZ. Learning models of human behaviour with sequential patterns. In: Proceedings of the AAAI-02 workshop 'Automation as Caregiver'; 2002. p. 24–30.

Hlavacs H, Kotsis G. Modeling user behavior: a layered approach. In: Proceedings of 7th international symposium on modeling, analysis and simulation of computer and telecommunication systems, 1999. p. 218–25. ISBN: 0-7695-0381-0.

Jacobzone S, Oxley H. Ageing and health care costs. Internationale Politik und Gesellschaft Online (International Politics and Society) 1/2002, 2002.

JP K, DW F. Current and future public health challenges. J Am Med Assoc (JAMA) 2000;284:1696–8.

Karl BC, Finkelstein SM, Robiner WN. The design of an Internet-based system to maintain home monitoring adherence by lung transplant recipients. IEEE Trans Inf Technol Biomed 2006;10(1):66–76.

Kern N, Schiele B, Schmidt A. Multi-sensor activity context detection for wearable computing. In: Proceedings of European symposium on ambient intelligence, EUSAI03, LNCS vol. 2875; 2003. p. 220–32.

Klein M, Konig-Ries B, Obreiter P. Service rings—a semantic overlay for service discovery in ad hoc networks. In: Proceedings of 14th international workshop on database and expert systems applications (DEXA'03); 2003. p. 180.

Laborie F, Chatty S, Reyterou C. Coordination and collaboration environments for production lines: a user acceptance issue. In: Proceedings of the 2005 ninth European conference on computer-supported cooperative work; 2005. p. 407–26.

LeBellego G, Noury N, Virone G, Mousseau M, Demongeot J. A model for the measurement of patient activity in a hospital suite. IEEE Trans Inf Technol Biomed 2006;10(1):92–9.

Maitland J, Sherwood S, Barkhuus L, Anderson I, Hall M, Brown B, Chalmers M, Muller H. Increasing the awareness of daily activity levels with pervasive computing. In: Proceedings of 1st international conference on pervasive computing technologies for healthcare 2006, IEEE, 2006.

Martínez-García A, Méndez-Olague R. Process improvement with simulation in the health sector. e-Health: application of computing science in medicine and health care; 2003. p. 204–19. ISBN: 970-36-0118-9.

Oliver N, Horvitz E. A comparison of HMMs and dynamic Bayesian networks for recognizing office activities. In: Proceedings of the tenth conference on user modeling (UM 2005), LNCS vol. 3538; 2005. p. 199–209.

Oliver N, Kreger-Stickles L. Enhancing exercise performance through real-time physiological monitoring and music: a user study. In: Proceedings of 1st international conference on pervasive computing technologies for healthcare 2006, IEEE, 2006.

Osmani V, Balasubramaniam S, Botvich D. Self-organising object networks using context zones for distributed activity recognition. In: Proceedings of second international conference on body area networks (BodyNets'07), in cooperation with ACM SIGCHI, 2007. p. 152. ISBN: 978-963-06-2193-9.

Peter H, Carstensen KS. Computer supported cooperative work: new challenges to systems design. In: Kenji I, editor. Handbook of human factors/ergonomics. Tokyo: Asakura Publishing; 1999. p. 619–36 [2003].

Quemada J, Miguel TD, Azcorra A, Pavón S, Salvachúa J, Petit M, et al. ISABEL: a CSCW application for the distribution of events. Multimedia telecommunications and applications. In: Proceedings of third international COST 237 workshop, LNCS vol 1185; 1996. p. 137–53.

Saranummi N, Korhonen I, Kivisaari S, Ahjopalo H. A framework for developing distributed ICT applications for health. In: Proceedings of 1st transdisciplinary conference on distributed diagnosis and home healthcare, IEEE, Arlington, VA, USA; 2006. p. 137–43.

Shancheng T, Yi Q, Wei W. An inexact matching method based on ontology and semantic distance for resource discovery and interaction. In: Proceedings of the 2006 international conference on pervasive systems & computing, PSC 2006; 2006. p. 108–13.

Sokolov AM. An adaptive detection of anomalies in user's behavior. In: Proceedings of international joint conference on neural networks, vol. 4; 2003. p. 2443–7. ISBN: 0-7803-7898-9.

Spivey M. The Z notation: a reference manual, 2nd ed. Prentice-Hall; 1992.

Sweet SJ, Norman IJ. The nurse–doctor relationship: a selective literature review. J Adv Nurs 1995;22(1):165–70.

US.C.B. IDB aggregations—access options. US Census Bureau. Table 094. Population by age and sex, 〈http://www.census.gov/ipc/www/idbagg.html〉; 2007 [last accessed 14.05.07].

Varshney U. Pervasive healthcare. IEEE Comput 2003;36(12):138–40.

Wang S, Pentney W, Popescu A-M, Choudhury T, Philipose M. Commonsense-based joint training of human activity recognizers. In: Proceedings of international joint conference on artificial intelligence (IJCAI 2007), 2007.

Weiser M. The computer for the 21st century. Scientific American; 1991. p. 94–100.

Wijewickrama AKA, Takakuwa S. Simulation analysis of an outpatient department of internal medicine in a university hospital. In: Proceedings of 2006 winter simulation conference WSC 06; 2006. p. 425–32. ISBN: 1-4244-0500-9.

Wyatt D, Philipose M, Choudhury T. Unsupervised activity recognition using automatically mined common sense. In: Proceedings of the twentieth national conference on artificial intelligence (AAAI 2005); 2005. p. 21–7.