Self-Organising Object Networks using Context Zones for Distributed Activity Recognition

Venet Osmani, Sasitharan Balasubramaniam, Dmitri Botvich

Telecommunications Software and Systems Group,
Waterford Institute of Technology,
Waterford, Ireland
+353 51 30 2902

{vosmani, sasib, dbotvich} @tssg.org

ABSTRACT

Activity recognition has a high applicability scope in patient monitoring since it has the potential to observe patients' actions and recognise erratic behaviour. Our activity recognition architecture described in this paper is particularly suited for this task due to the fact that collaboration of constituent components, namely Object Networks, Activity Map and Activity Inference Engine create a flexible and scalable platform taking into consideration needs of individual users. We utilise information generated from sensors that observe user interaction with the objects in the environment and also information from body-worn sensors. This information is processed in a distributed manner through the object network hierarchy which we formally define. The object network has the effect of increasing the level of abstraction of information such that this high-level information is utilised by the Activity Inference Engine. This engine also takes into consideration information from the user's profiles in order to deduce the most probable activity and at the same time observe any erratic or potentially unsafe behaviour. We also present a scenario and show the results of our study.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical Sciences – *health, medical information systems.*

General Terms

Algorithms, Management, Performance, Experimentation, Human Factors

Keywords

Activity, recognition, inference, health-care, monitoring

1. INTRODUCTION

Miniaturisation and proliferation of wireless devices coupled with a steady increase in their capability has lead to a significant growth in Pervasive Computing research. Pervasive Computing Environment paradigm aims to improve users' lifestyles, such as supporting user's computing application, observing her goals and needs and providing assistance anywhere, anytime. Recently, a significant attention has been focused towards Pervasive Healthcare, where the aim is to create a collaborative computing environment supporting efficient patient diagnosis for medical officers, or providing assistance for patients suffering form various medical disorders, including cognitive disorders. Considering the fact that life expectancy is increasing, coupled with the rising costs of health care, Pervasive Healthcare will play an important role and impact in the very near future, where applications such as telemedicine and remote diagnosis can be performed. Due to the limited funding for public hospitals, various governments are implementing plans to create technology that provides intelligent diagnosis on behalf of medical doctors, allowing non-life threatening diagnosis to be performed by computing devices.

Activity Recognition as a research trend in Pervasive Computing can bring closer realisation of this vision. Activity Recognition is the ability for computing devices to monitor the user and environment and infer user's activity based on events triggered by user's mobility or actions. Crucial to realisation of such vision is integrating knowledge about many aspects of the user's environment and profiles. By observing interaction of the user with the environment to infer user activities, much can be learned about the user's behavioural patterns and anticipation of future intentions.

In this paper, we propose an Activity Recognition system capable of observing and inferring user activities based on the user's interactions with body-worn objects and various objects in the environment as well as integrating other context information such as user's preferences and profiles. The objects within the environment include various devices, where we envision an environment that has sensors and actuators embedded in each device, living environment (e.g. floor sensors, fabric sensors), or even on the users (e.g. Body Area Networks). We do not rely on a specific technology (e.g. RFID [1]) to provide the necessary information about the user actions; rather we take a multi-level hierarchical inference process that is distributed and supports self-organisation of various objects (e.g. sensors, actuators, utensils) and at the same time evaluates specific context information relevant to the user. Since user actions are typically recorded through primitive sensors, we place a significant focus into our platform that allows processing of this information by recursively composing the low-level data into higher level abstract information that are ultimately utilised by our Activity Inference

BODYNETS 2007, June 11-13, Florence, Italy

DOI 10.4108/bodynets.2007.152

Engine, thus ensuring an efficient activity inference process. While activity recognition is applicable to a large number of domains, the primary focus of this work is on applying activity recognition for health-care application. Our case study describes applicability of activity recognition in assisting patients suffering from an early stage of dementia, a progressive brain dysfunction that affects various intellectual processes including thinking, decision making, reasoning and spatial orientation [2]. Activity recognition system has the potential to initially infer users' activities based on their location as well as objects within the vicinity of that location, and guiding them through the correct steps in completing an activity whenever an erroneous behaviour is detected. This in turn reduces the cost of supervision by caregivers and most importantly ensures that the patients live a relatively normal and independent life.

The rest of this paper is organised as follows. We first review and critique the ongoing research that address the problem of Activity Recognition in Section 2. Then we present our architecture, components and a description of each in Section 3. Section 4 deals with the simulation scenario and we also present the results obtained, while Section 5 provides the concluding remarks.

2. RELATED WORK

Research in activity recognition has seen two major strands in terms of the employed methodology. The first approach relies on video feeds that are processed and analysed to infer what the user is currently doing (e.g. [3-5]). While the second approach involves activity inference based on Body Area Networks and information about objects in the environment, manipulated as a result of a user action. Our focus is on the latter methodology and we review the most relevant systems.

Guralnik and Haigh [6] 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 it is very difficult to understand specific user behaviours. In [7] authors describe a hardware platform equipped with 3-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. Also the framework is heavily centralised with no support for personalising to suit specific user behaviour. Bao and Intille [8] 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 [9]. Although 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 that may aid in adapting the system to particular users.

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 [1]. 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) 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 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 behaviour which limits the number of activities that can be recognised. Our design presented in the next sections aims to alleviate these issues by employing a number of novel concepts that are mapped to various components of the architecture.

3. ARCHITECTURE

The previous section highlighted some of the drawbacks of current Activity Recognition systems. In general Activity Recognition architectures have to confront a set of relatively strict constraints in order to intelligently and efficiently meet users' goals and expectations. Device heterogeneity is an issue that have to be considered on the onset of the system design. Relying on a single technology or standard will severely impair interoperability with devices that a user may have in the future, resulting in weak likelihood of user acceptance of the system. 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. Another issue which has to be addressed lies in the continual changing user behaviour. Everyday experience shows that user behaviour does not follow the same pattern. Therefore, this dynamic behaviour has to be incorporated into the system design. In addition, specific context information in the profile of users must be integrated to support their requirements and needs.

Designing a system within these stringent set of requirements is indeed a challenging task. However, our architecture presented in this section 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 body-worn sensors and objects within the living environment; (ii) context information processing in a distributed, hierarchical manner resulting in a high level abstraction of context information; (iii) support self-organisation of devices into object networks that infer user activities (iv) pruning irrelevant information while actively utilising the pertinent context information through our concept of role fitness (v) decentralised user activity inference technique through the objects in the surrounding user environment (vi) supporting behaviour knowledge transfer between heterogeneous domains (vii) incorporating learning techniques for both existing and new users entering a domain. The focus of this paper is in the organisation of object networks to support inference of human activities within the constraints considered above, where we describe the functionalities for (i), (ii), (iii), and (v). In order to tackle these issues we have defined a number of components comprising the overall architecture, namely Object Networks, Activity Map (AM), Activity Inference Engine (AIE) illustrated in Figure 1. For an overview of these components see [10][18]. The following sections will describe each component and their functionality.

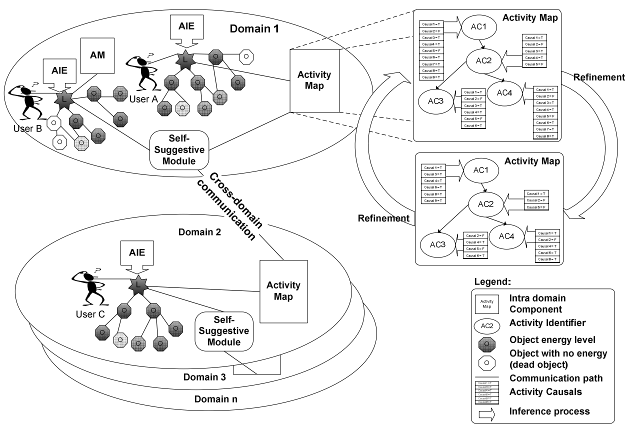

Figure 1 Overall Architecture for Activity Recognition

3.1 Object Networks

The *Object Network* is an overlay network of interconnection between various devices worn by the user and within the vicinity of the user, and provides the necessary and relevant context information depicting environment status and user actions. An object is any artefact that can contribute to the activity inference process, and has the ability to self-organise with various objects through peer to peer interaction. This concept has been primarily inspired from the functionalities of sensors which have the capability to recognise other sensors and self-organise. However, an object network has the ability to form an interconnected overlay with various devices of different functionalities and processing capabilities. This includes objects that a user may interact with (e.g. laptops, PDAs, medical monitors, instruments) since they provide pertinent information relating to the user activities.

Utilising object networks has a two-fold benefit. Firstly, object networks enable an efficient processing of context information generated from low-level sensors as a result of specific human behaviour which dictates the interaction sequence with a set of objects. Object networks achieve this through the distributed nature of the objects that take part in an object network where no centralised control exists. Secondly, object networks bridge the gap between low-level raw sensor information and high level information, such as user's actions or goals by increasing the level of abstraction of context information. Increasing the level of abstraction of context information directly benefits the activity inference process, since the AIE is shielded from low-level details associated with the raw information generated from sensors (such as sampling, data representation, composition, filtering) when inferring users' activities and also increases the efficiency of the inference process. This allows the AIE to process a smaller set of pertinent information as opposed to a potentially vast volume of raw information generated from a large number of sensors. Object networks achieve this through formation of a hierarchical object network structure created as a result of the local interaction rules between objects.

Object networks are dynamically created and typically involve objects within a close proximity to the user. As such an object network will 'follow' the user as s/he changes position within a domain. Such object network behaviour stems from the fact that human activities are highly localised, in other words performing activities typically involves manipulation of objects in close proximity.

3.1.1 Object Network Architecture

An object network is specific to a domain, where each object is equipped with a mechanism that allows sensing and discovering other objects and as such establishing an interaction network. We assume that objects have communication capabilities in addition to an embedded dual layer stack composed of the *infrastructure layer* and the *application module layer*. Object sensing and discovery algorithms are housed in the infrastructure layer which is primarily concerned with the establishment of the object network. The infrastructure layer is also responsible for executing and coordinating the election process between the elements of an established object network that determines the object with the highest capabilities to execute the AIE – the *leader object*. The application module layer includes dynamically loadable modules which are referred to as *object roles*. An object role specifies part or overall object functionality of an object and 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 that can be executed in parallel.

In addition to object roles the application module layer also houses a set of *role dependency rules.* Role dependency rules determine if a particular role is dependent on information generated from other role(s) such that the role in question can be successfully fulfilled. For example a PDA in deducing patient's state may rely on information from various body-worn sensors, such as temperature, blood pressure or heart monitor to fulfil its 'patient state' role.

While role dependency rules dictate object to object interaction, the *context evaluation rules* determine the semantics of the exchanged information between roles of various objects. Context evaluation rules are used to interpret context information received form another object's role with which a dependency relationship has been established. In effect the context evaluation rules infer an actual deduction through processing and composition of the information received from dependent roles of various objects. For example the PDA in the example above uses context evaluation rules to determine the patient's state based on the information received from other object roles housed in body-worn sensors, namely temperature, blood pressure and heart monitor. Collaboration of the above components creates a platform for an efficient information exchange between roles and an effective interaction between the objects assembling the object network.

3.2 Self-organisation of Object Networks

In order to efficiently process context information and to provide pertinent information to the activity inference process, objects within a domain will self-organise to create a hierarchical, tree-like structure based on local interaction rules. Context information may be generated as a result of user interaction with the objects in the environment or in response to the changes detected by body worn sensors. Typically low-level sensors will sit at the bottom of the structure while the *leader object* that has been chosen through the election process takes the highest position at the top of the structure and has the responsibility to execute the AIE. This is illustrated in Figure 2.

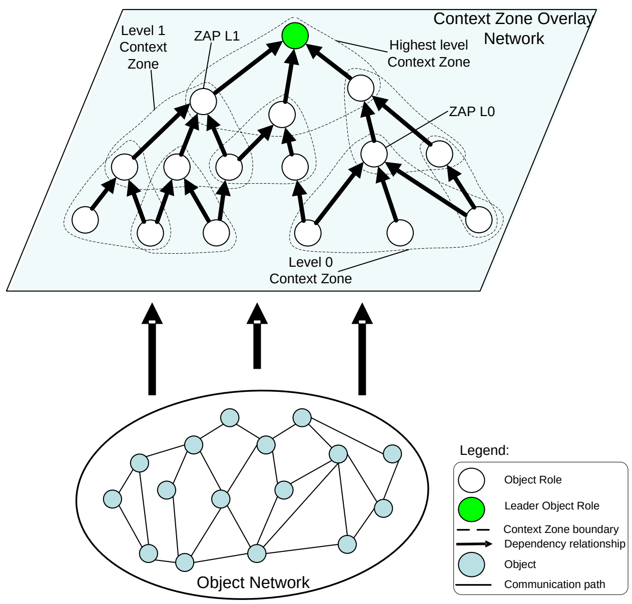

Figure 2 Creation of Context Zones

The initial structure-less ad-hoc network converges into a hierarchical structure based on consumer/provider relationships which are established on the basis of local role dependency rules. This in turn enables formation of *Context Zones* (Figure 2).

3.2.1 Object Network Role Dependency

In general, an object role may be an *information provider role*, *information consumer role* or both. Low-level sensors adhere to information provider roles only since they are self-sufficient and typically do not require information generated elsewhere. Leader object houses the ultimate information consumer role that gathers information generated from various object roles to deduce user activities. All other object network roles that are positioned in between are both consumer and provider roles such that they enable information flow from the lowest level of hierarchy up to the leader object.

Each consumer role uses the object's communication interface to broadcast a semantic query within a small number of hops to other objects in the vicinity. The query contains semantic description of the type of information required by the role so that it can be successfully fulfilled. Role information requirements are specified within the role dependency rules. In our example above it may be that the '*patient state deduction*' role which is a consumer role executed by a PDA may have role dependency rules stating that this role requires temperature information, blood pressure and heart monitor data. In order to fulfil this 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 to the querying object and a *dependency relationship* is established between the respective roles, i.e. the 'patient state deduction' role and the corresponding objects' information provider 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.

3.2.2 Context Zones

Establishing dependency relationship between roles allows formation of Context Zones which are the main ingredient in the emergence of a global hierarchical object network role structure (see Figure 2). The idea of Context Zones has been inspired from the work on Semantic Overlay Networks [11],[12]. 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 its 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 P is a set of artefacts having communication capabilities such that each artefact $o_i \in P$ can exchange information with another artefact $o_j \in P$ using an underlying pre-agreed or translated protocol.

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 'sniff' other nodes in the close proximity and query the services (roles) they provide.

An object role r_n^P of an object $o_i \in P$ is defined as

$$o_i \left\langle r_n^C \right\rangle$$

The top index indicates whether the role is a consumer role (C) or provider role (P) while the bottom index distinguishes multiple roles within the same object.

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 called the *dependency set*

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

Definition 2 (Context Zone)[1]

Based on the 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_p is as follows:

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

A Context Zone is complete when the dependency set Θ of the consumer role has been satisfied. The Zone Access Point (ZAP) r_n^C of a Context Zone Z_p is denoted as follows[2]:

$$o_i\langle r_n^C\rangle \perp Z_p$$

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 are further self-organised to support higher level Context Zones of the hierarchy. This hierarchical structure contains a number of levels determined by the number of the Context Zone levels and now we formally define these levels.

Definition 3 (Level 0 Context Zone)

A Level 0 Context Zone is created when role dependencies of its ZAP have no dependencies of their own. In other words all elements of the dependency set must be solely provider roles that have no further dependencies. Thus:

$$Z_p^0 = \left(o_i\langle r_n^C\rangle, \Theta\right) \Leftrightarrow o_i\langle r_n^C\rangle \xrightarrow{d} \Theta \wedge$$
$$\left(\forall r_m^P \in \Theta, \exists o_n \in P \bullet o_n\langle r_m^P\rangle \xrightarrow{d} \Theta \wedge \Theta = \{\varnothing\}\right)$$

As stated before low-level sensors typically adhere to provider roles only and as such have no other dependencies, hence it is also intuitive to create a Level 0 Context Zone at this level of the hierarchy. There may be multiple Context Zones within the same level and for this reason we use the lower index to distinguish zones on the same level. Membership of a role in a Context Zone is not exclusive thus allowing the possibility of overlapping zones.

[1] In what follows we use the $\bullet$ symbol to denote "such that" which corresponds with Z formal specification language conventions [13]

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

The ZAPs of Level 0 Context Zones are then organised to support higher level Context Zones (Level 1), in effect becoming provider roles for the zones in question.

Definition 4 (Context Zones for Level 1 and above)

A Level 1 Context Zone is created when all elements of the consumer role dependency set, are ZAPs of a Level 0 Context Zone.

$$Z_q^1 = \left(o_i\langle r_n^C\rangle, \Theta\right) \Leftrightarrow o_i\langle r_n^C\rangle \xrightarrow{d} \Theta \wedge$$
$$\left(\forall r_m^P \in \Theta, \exists o_n \in P \bullet o_n\langle r_m^P\rangle \perp Z_p^0\right)$$

Therefore every element of the consumer role dependency set Θ must satisfy the condition of being the ZAP of a lower level zone.

Further levels of Context Zones are defined in a similar manner. For example a Level 2 Context Zone would be defined in terms of Level 1 Context Zones. A generic definition of the Context Zones above Level 0 is as follows:

$$Z_q^x = \left(o_i\langle r_n^C\rangle, \Theta\right) \Leftrightarrow o_i\langle r_n^C\rangle \xrightarrow{d} \Theta \wedge$$
$$\left(\forall r_m^P \in \Theta, \exists o_n \in P \bullet o_n\langle r_m^P\rangle \perp Z_p^{x-1}\right)$$

The hierarchical levels of Context Zones continue until the top of the hierarchy is reached at the leader object role.

Definition 5 (Leader object role)

The leader object role $o_l\langle r_n^C\rangle$ must be ZAP of the highest level Context Zone, each of its dependencies must be ZAPs of Context Zones one level below and at the same time the leader object role cannot be a provider role, since there are no requirements to process information beyond the leader object. Therefore for the highest Level *x* of a Context Zone in an object network, the following has to be true for an object role to become the leader object role:

$$o_l\langle r_n^C\rangle \perp Z_a^x \wedge \neg\exists\, o_l\langle r_n^P\rangle \wedge o_l\langle r_n^C\rangle \xrightarrow{d} \Theta \bullet$$
$$\left(\forall r_m^P \in \Theta, \exists o_n \in P \bullet o_n\langle r_m^P\rangle \perp Z_p^{x-1}\right)$$

We have formally defined the concepts that enable object network self-organisation from local object interactions. The basis for creating dependency relationships lies essentially in matching the semantic queries for information sent by consumer roles with the information offered by the provider roles within the 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.

3.3 Semantic Role Description

Matching information needs of a consumer role with the information provided by provider roles is essentially achieved by computing the semantic similarity between the requested information and the provided information. However, in order to enable computation of semantic similarity, an important requirement is that the semantic role description must be common across all roles.

Much work has been done in area of semantic description of services (for an overview see [14]). 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 we have decided to take a different approach. Rather than opting for a particular semantic role description we have chosen to make the process of object network organisation to be independent of a specific semantic role description. This is inspired from the work in [12]. In our solution 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 then used to calculate the degree of semantic similarity between a set of roles. The output of the semantic distance function is used to determine 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 the consumer role and the 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 limit the applicability scope of object networks. In addition, diverse domains have specific requirements that may render a particular semantic role description unsuitable for the domain in question.

A generic representation of the distance function is as follows:

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

The semantic similarity coefficient δ is calculated between a query from a consumer role r_n^C housed in the object o_i and a provider role r_n^P housed in another object o_j.

There are numerous examples of semantic service descriptions that meet our criteria for 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 already exist to calculate the distance function (see [15] 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 $\delta \leq \beta$ and thus a dependency relationship can be established.

3.4 Activity Map and Activity Inference

Once an object network has been formed, the leader object will begin inferring user's activities. Inference is based on evaluating events from the object network and matching them to the Activity Map. The Activity Map is a repository that is specific to a user within a domain and stores activities that a user has performed along with the relationship between the activities. The concept of Activity Map is based on the idea that users typically perform activities drawn from a finite set in order to achieve a particular goal. An example of part of an AM is illustrated in Figure 3.

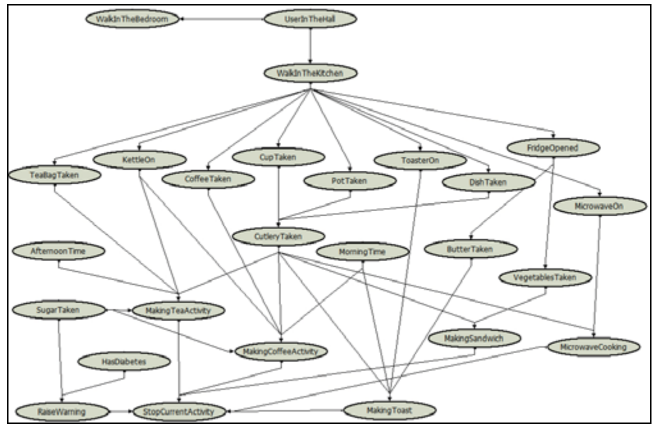

Figure 3 Part of the user's AM

The internal structure of an AM corresponds to a directed graph where each arc connecting neighbouring activities is assigned a probability value. In addition to representing relationship between tasks, the AM also 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. In order to determine user activities we infer the relationship between an activity's causal set and events from objects using Decision Module within the AIE, which is illustrated in Figure 4.

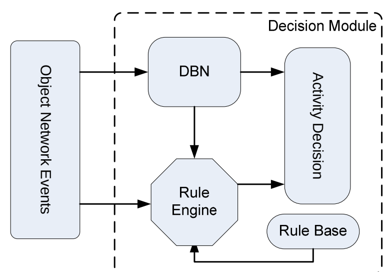

Figure 4 Decision Module

Our Decision Module, includes a *Dynamic Bayesian Networks* (DBN) and a rule engine. Information contained within an AM is utilised by the leader object that actively receives information processed by the object network pertaining to user actions and behaviour. Received information is essentially evidence that is constantly fed to the DBN to infer a set of likely activities, where we apply the Junction Tree Algorithm. Another decision layer is handled by a rule-base contained in the decision module (see Figure 4). The rules are used to encode sequence of objects manipulated within a time threshold specified per activity. This is to ensure that activity inference is not carried out based on sporadic events from the object network, for instance unintentionally taking a tea bag with no intention of making tea.

Also we use the rules to specify the actions that need to be taken in the event of detection of potentially harmful actions, such as the one described in our scenario.

4. CASE STUDY AND SIMULATION

In this section we elaborate on the application scenarios of activity recognition. Our scenario concentrates on supporting a patient suffering from the early symptoms of a Dementia. Dementia encompasses a number of diseases that cause progressive decline in cognitive functions due to neuron degeneration in the brain. The most well known type of dementia is Alzheimer's disease [16] which is a behaviour altering disease that severely impairs patients' Activities of Daily Living (ADL) [17].

4.1 Dementia patient support scenario

John is 75 years old, and for a number of years he has lived with type II diabetes. After his wife passed-away John moved to a community retirement home which has been equipped with state of the art patient monitoring and support system. As he is approaching the age of 80 John is becoming more and more dependent on the health-care staff at the centre. He has problems recalling names of people close to him and from time to time he also gets confused and de-orientated about his surroundings. The head psychiatrist has received John's reports from the patient monitoring system and has decided to conduct an interview with him. John is showing symptoms of an early stage of dementia and thus he can no longer sustain his normal behaviour. As a result the patient monitoring system is setup such that in addition to constantly monitoring John's behaviour it also provides the necessary support whenever an erratic behaviour is detected.

It's morning time, and as John is waking up, the system monitors his actions and updates the activity daily log. John gets up from his bed and walks towards the kitchen, wanting to prepare his morning tea. An object network has been created around his current location that includes body-worn sensors and objects in the surroundings such as a kettle, pots, cup holders, tea, coffee and sugar jars, milk dispenser, toasters, a microwave, an oven cooker and a smart fridge which has been chosen as the leader object. The leader object connects to the central server to retrieve John's personal profile information. While in the kitchen John can be engaged in any number of activities for instance making a sandwich, heating the soup, making a coffee or any other activity that may involve the nearby objects. However in this instance John has decided to take a cup, activating the embedded cup sensor. Information from the sensor is processed by the cup holder causing an event to be sent to the fridge describing John's action. Incidentally the fridge has downloaded part of John's Activity Map that details his activities within the kitchen domain. At this point the system does not contain enough information to reliably deduce user's activity since holding a cup causal is common to a number of activities specified in the John's AM for this domain. However, when John takes out a tea bag from the jar the weight sensor causes an event to be processed and sent to the leader object. This event has a high impact in inferring the 'morning tea' activity as defined in the AM. Coupled with the previous event the Activity Inference Engine executed on the smart fridge deduces with a high probability user's activity and as such can guide John in the event an erratic behaviour is detected. John carries on with putting the tea bag in the cup and boiling the water. However he becomes confused and reaches for the sugar jar. Since this action is specified as potentially harmful due to his diabetes, and in accordance with John's medical profile, the system generates an audio cue warning Join that this action is dangerous. At the same time the kitchen display shows a warning sign while instructing him to take milk as he always did. John abides by the instructions given to him and a behaviour log is created to be later reviewed by the medical staff.

The above scenario represents a small subset of the typical support that our implementation can provide for patient monitoring and support. We present preliminary results in the following section.

4.2 Simulation Scenario

In our scenario presented above a number of Context Zones are created in response to John's position in the kitchen (see Figure 5). For example a Level 0 Context Zone is created between the cup holder and the nearby kitchen utensils such as cups, pots and dishes. One of the roles defined in the cup holder object is 'utensils monitor' which is a consumer role responsible for processing motion sensor information from each kitchen utensil and inferring high-level events, such as a cup has been taken. Role dependency rules of this role state that this consumer role requires information describing motion status of a utensil. Therefore, in order to fulfil this role, the cup holder broadcasts a query to other objects in the proximity requesting roles that provide information describing motion status of a utensil. Roles that provide such information, namely embedded sensors in cups and pots respond back to the query originator. Upon receiving a response the cup holder role establishes a dependency relationship with each respondent utensil role. Once the cup holder object has processed all responses a Level 0 Context Zone is established between the provider role of the cups and the 'utensils monitor' role which also becomes the Zone Access Point of this Context Zone (see Definition 2). In a similar manner the role that provides information about the status of electrical objects in the kitchen, housed in the microwave object has established a Level 0 Context Zone with kettle's status role, toaster's status role and cooker's role.

Once Level 0 Context Zones have been established, each corresponding ZAP of these zones waits for requests from other objects to join a higher level Context Zone as provider roles. At John's current position there are only two levels of Context Zones, thus ZAPs do not receive a request to join another context zone and as such assume that their respective zones must be one level below the highest level context zone. Therefore, each ZAP sends its role advertisements to the leader object in order to complete the object network. The leader role housed in the fridge responds back and establishes dependency relationships with each ZAP. When all the dependencies are established the leader object begins receiving high level events derived from low-level sensors thus inferring user's activities.

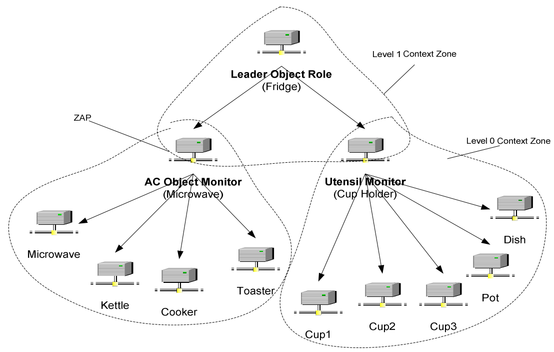

Figure 5 Context Zone formation

4.3 Performance Evaluation

We have tested the performance of the system using simulated data based on the observation of real life activities. We have monitored the behaviour of a number of people and recorded the characteristics of the objects users manipulated while performing their activities. 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 what their status is during that time. An important advantage of this approach is that the simulation process is entirely transparent to the object network and the leader object, since events are generated in the same manner as they would have been generated should a real user perform a particular activity.

For this paper we present preliminary simulation results of performing two activities. Both activities relate to the above specified scenario and are simulated within the context specified by the scenario. At John's location there are five possible activities specified in the user's AM, namely 'making tea', 'making coffee', 'microwave cooking', 'making toast' and 'making sandwich'. We simulate the actions of 'making tea' and 'making sandwich' activity, which have been fed to the TA. Our aim is to show the results of activities being discriminated by the Decision Module as the evidence is received from the object network and at the same time evaluating other context information such as John's medical profile. For the first activity we obtained the results shown in Figure 6:

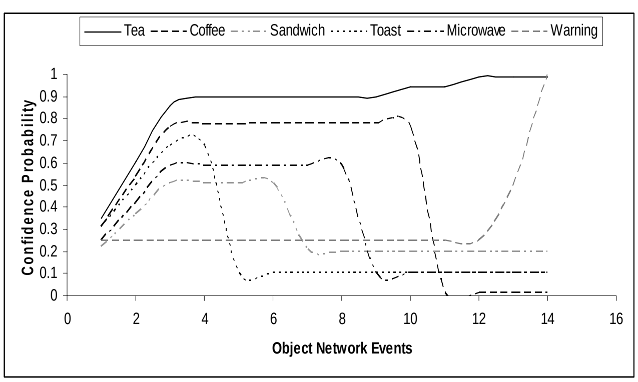

Figure 6 Activity 'making tea'

As it can be seen from the graph, all other activities either are discriminated early or have a low probability of occurring. However, making tea and making coffee activity continue having a high probability for some time. Such behaviour occurs because the two activities have a large number of objects in common which makes it more difficult to distinguish between these two activities in comparison with other activities. In addition the common objects shared between both activities have a high impact in the inference of these activities.

However, the discriminatory event is the fact that John has taken a tea bag, which dramatically lowers the probability of 'making coffee' activity occurring, while increasing the confidence of the other activity. Therefore, these results show that DBN is able to quickly narrow the number of potential activities and also discriminate between the remaining ones.

Also in our scenario above, John is diabetic sufferer and as such we have encoded events that pose a high risk for John (Figure 3). In this case, except for information from objects, we also integrate other context information into user's AM, for instance information from user's medical profile that allows close monitoring of patient. As depicted in Figure 3 we incorporate medical conditions, such as diabetes in this instance, and actions that are potentially harmful to patients with these conditions, such that *P(raise_warning | has_diabetes, sugar_taken)* is very high. For example John's actions are monitored in the course of carrying out various activities (e.g. making tea) and since he uses sugar, this causes a warning message to be raised (at event 12). The rule engine evaluates the set of activities inferred by the DBN and performs a final decision which is to warn John to stop his current activity and notify John's carers.

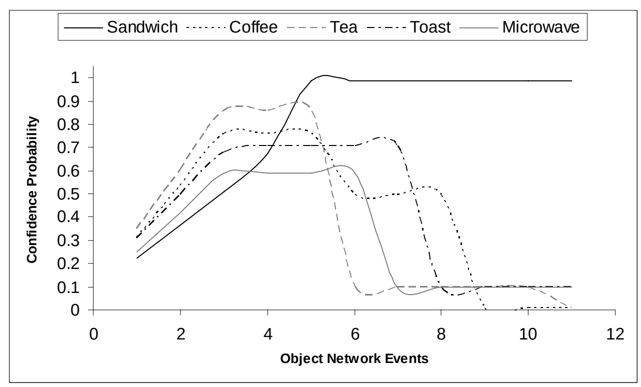

Figure 7 Activity 'making sandwich'

A similar pattern follows the making sandwich activity (Figure 7), however because this activity has a small number of events shared with other activities, it can be inferred very early as evidenced from the graph. The average probability achieved in inferring both activities is 98%.

5. CONCLUSION

In this paper we have presented flexible and scalable activity recognition architecture that is able to process information from various sensors in a distributed manner. In addition to the information generated from the user interaction with the environment, we also take into account user's medical profile such that we can infer potentially erratic or hazardous behaviour. Object network self-organisation provides a scalable context information processing platform which dramatically reduces the burden on the AIE. Our preliminary results show that we can discriminate activities at the early stages, thus ensuring an accurate activity inference process. In the future we aim to conduct further experiments with various levels of activity complexities and also study the impact of the rule base in the inference process.

6. REFERENCES

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

[2] The American Geriatrics Society, "The American Geriatrics Society - Education - Dementia", http://www.americangeriatrics.org/education/forum/dementia.shtml, Last Accessed 01/02/07, 2007

[3] James L. Crowley, "Context Aware Observation of Human Activity", PSIPS 2004, Oulu, Finland, 2004

[4] Dhruv Mahajan, Nipun Kwatra, Sumit Jain, Prem Kalra, and Subhashis Banerjee, "A Framework for Activity Recognition and Detection of Unusual Activities", Proceedings of the Fourth Indian Conference on Computer Vision, Graphics & Image Processing (ICVGIP 2004), pp. 15-21, 2004

[5] Thomas B. Moeslund and Erik Granum, "A survey of computer vision-based human motion capture", Computer Vision and Image Understanding, vol. 81, issue 3, pp. 231-268, 2001

[6] 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

[7] Nicky Kern, Bernt Schiele, and Albrecht Schmidt, "Multi-Sensor Activity Context Detection for Wearable Computing", In Proc. European Symposium on Ambient Intelligence, EUSAI03, LNCS vol. 2875, pp. 220-232, 2003

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

[9] 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

[10] Venet Osmani and Sasitharan Balasubramaniam, "Context Management Support for Activity Recognition in Health-Care", Pervasive 2006 Workshop Proceedings (ISBN 978-3-00-018411-6), pp.453-465 T. Strang, V. Cahill, & A. Quigley, eds in 3rd International Workshop on Tangible Space Intiative, 2006

[11] Arturo Crespo and Hector Garcia-Molina, "Semantic Overlay Networks for P2P Systems", Technical report, Computer Science Department, Stanford University, October 2002,

[12] Michael Klein, Birgitta Konig-Ries, and Philipp Obreiter, "Service Rings - A Semantic Overlay for Service Discovery in Ad hoc Networks", 14th International Workshop on Database and Expert Systems Applications (DEXA'03), p.180, 2003

[13] Mike Spivey, "The Z Notation: A Reference Manual, Second Edition", Prentice Hall, ISBN: 0-139-78529-9, 1992

[14] Liliana Cabral, John Domingue, Enrico Motta, Terry R. Payne, and Farshad Hakimpour, "Approaches to Semantic Web Services: an Overview and Comparisons", First European Semantic Web Symposium, ESWS 2004, LNCS vol. 3053, pp. 225-239, 2004

[15] Tang Shancheng, Qian Yi, and Wang Wei, "An Inexact Matching Method Based on Ontology and Semantic Distance for Resource Discovery and Interaction", Proceedings of the 2006 International Conference on Pervasive Systems & Computing, PSC 2006, pp. 108-113, 2006

[16] The Alzheimer’s Association, "What is Alzheimer's?" http://www.alz.org/alzheimers_disease_what_is_alzheimers.asp, Last Accessed 01/02/2007, 2007

[17] KATZ S, FORD AB, MOSKOWITZ RW, JACKSON BA, and JAFFE MW, "Studies of Illness in the Aged. The Index of ADL: A Standardized Measure of Biological and Psychological Function", JAMA. 1963;185: 914-919, 1963

[18] Venet Osmani, Sasitharan Balasubramaniam, and Tao Gu, "A Distributed Hierarchical Structure for Object Networks Supporting Human Activity Recognition", In Proc. of 9th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services, MMNS 2006, LNCS vol 4267, Dublin, Ireland., 2006