DiCPA: Distributed Context Processing Architecture for an Intelligent Environment

Teddy Mantoro, Chris. W. Johnson
Department of Computer Science, Australian National University, ACT 0200, Australia
{teddy,cwj}@cs.anu.edu.au

Keywords, Distributed Context Processing Architecture, Location Awareness, User Mobility, Context Awareness, Ubiquitous Computing, Intelligent Environment.

Abstract
This paper describes a distributed context processing architecture for an Active Office as an implementation model of an Intelligent Environment. It describes the IE repository, IE resolution, knowledge-based and resources manager as an integrated part of context awareness mechanism which plays an important role in an Active Office. It proposes how the scalable distribution of context and user model information is managed in an Active Office. It illustrates an implementation of the architecture through an example of how to get connected in an unfamiliar Intelligent Environment domain.

1. INTRODUCTION

In an Intelligent Environment (IE), the computing and sensing devices are ubiquitous. However, computer and communications systems today are underused because the range of control mechanisms and application interfaces is too diverse. Context awareness mechanisms could therefore be the best way to implement computer applications in IE. It is necessary to consider the mechanism that might allow users to manipulate systems in simple and ubiquitous ways and make computers more aware of the facilities in their surroundings [11,27].

These awareness-mechanisms are bringing the computer into user's daily activities. In this paper we explore the computers capability to recoqnise user's location, activities and the social context defined by the presence of other people, in order to assist people with a variety of activities at work.

An Intelligent Environment (IE) is a ubiquitous computing environment with significant processing done by various sensors (fixed and mobile sensors) and using a DiCPA architecture. A DiCPA architecture is a scalable context processing architecture which at least contains an IE repository, a resources manager, a knowledge base and a global/local resolution object naming scheme.

This paper describes an Active Office as an implementation model of an IE. For the Active Office an IE is the collection of detectors and sensors, in a local physical vicinity which corresponds to a 'local domain' of the IE (i.e. IE domain), with descriptions and communication for these entities regarded as services. In order for an Active Office to provide services to the users, the Active Office can be design to detect its current state/context and determine what actions to take based on the context [9,15]. So, we define an Active Office as a normal office, which consists of several normal rooms with minimal additional decorations (minimal intrusive detectors and sensors, without explicitly badging people) and using DiCPA architecture to manage and respond to rapidly changing aggregation sensor data.

The DiCPA architecture is based on Merino services layer architecture [15]. It could manage the interaction between users and the environment in the IE domain by handling raw sensor information and making transformations between the raw physical devices/sensors within the environment and the application programs. The infrastructure supports interoperability of context sensors/widgets and applications on heterogeneous platforms [9,15].

An IE domain has an important role as an administrative domain. It contains an IE repository, a resources manager, a knowledge base and various sensors. Every object in an IE has a local name and a global name. It needs to register in a resolution server for a persistent mapping purpose in an IE repository. Hence, we can access every registered object from an unfamiliar Active Office (other namespace) by giving the resolution protocol, the resolution server and the object's global name to the resources manager and the knowledge base.

To support an Active Office infrastructure, we could apply a WVLAN together with a General Packet Radio Service (GPRS) or a Universal Mobile Telecommunications Systems (UMTS). We implemented WiFi (802.11b,b+,g), Bluetooth, IrDA and GPRS (GSM network) integrate them with a local wired network. We used a MAC address of Wifi and Bluetooth to identify users, and these technologies also enable us to determine user location in an indoor Active Office [16,17,18].

This paper describes the requirement for recognising user locations and activities in an Active Office environment. First, we present related work in the IE area, followed by an overview of the Merino architecture, the description of the DiCPA distributed context processing architecture, IE domain, IE resolution, IE repository, user modelling, resources manager, knowledge base, and user's activities. In user's activities we describe how a user model which is characterised by user identification and authentification, user profile, network access and service adaptation to user environment. Then, we discuss a scenario for the purpose of recognising user's location and user's activities. Finally, a summary will be presented in the last part of this paper.

2. RELATED WORK IN IE

Since 1992, Mozer et.al. [21] has been developing intelligent behaviour in an entire adaptive environment in the University of Colorado. In 1996, while Coen et.al. [6,7] starting building intelligent rooms in the MIT AI Lab. Kidd et.al. built the 'aware home' in the Georgia University in 1997 [14], and at the end of 2000 Cisco Inc. built a broadband internet home (ihome)[5].

Mozer's focus is on adaptability, building the intelligence into various sensors (such as thermometers) and effectors (such as a heating and ventilation system) so that it can adapt to the preferences of the house residents. One of his important questions is how the intelligence can infer the preferences of its resident, such as by learning patterns in its occupants' behavior to adjust the setting of lights, ventilation and thermostats [21].

Much of Coen's efforts centre on the two intelligent rooms he built in the MIT AI Labs dealing with the integration of various sensing modalities such as vision and speech [4,6,7]. In the aware home, Kidd focuses on finding frequently lost objects such as keys, wallets, glasses and remote control using small radio-frequency tags attached to each object, and how to track the object using a long-range indoor positioning systems. The aware home allows support of 'everyday cognition' for the elderly using sensing technology [14].

The Broadband Internet Home built by Cisco Inc. is essentially a home with broadband connections that provided an application to monitor and control the house from virtually anywhere, anytime, over the internet [5]. It does not use context aware applications.

Abowd (1996) and Flanagan (1999) showed it is not only offices and homes that can be intelligent, but also rooms with large numbers of people such as lecture halls/classrooms and conference venues. Abowd et. al. has made substantial effort to develop an educational environment such as Classroom 2000 at Georgia University, that automatically creates records linking simultaneous stream of information, such as what the teacher is saying, while a student is writing down her notes on a digital pad [1,2], whereas Flanagan focusses on recognising automatically who in the audience is asking a question, pointing the video camera to that position and person and using a microphone array to filter out sounds coming from elsewhere in the large lecture hall at CAIP Center, Rutgers University [10]. He divides the problem into two parts, identifying the spatial location of the sound source, and then extracting the desired sound signal out of the collection of sounds being received by the microphones. He points out how microphone and digital signal processing technologies are enabling the creation of such intelligent rooms.

Recent technical advanced technology in Active Badge/Bat (Cambridge), Wearable Computing (University of South Australia), Cricket (MIT), and Smart Floor are also enabling the creation of such Intelligent Environments [11,12,23,22,26]. These advanced technology have demonstrated the potential of context awareness application, but have also shown that these kinds of systems are still extremely difficult to design, develop and maintain [13].

3. MERINO: IE SERVICE LAYERS ARCHITECTURE

The Merino architecture has two important parts. The first part is the managing of the interaction between physical devices/sensors within the IE environment and the program application, to manage the interaction between the user and the environment. This part contains its abstraction layers: Core Sensors and Device Layers, Device Abstraction Layers, Context and the highest abstraction level, i.e. Smart Environment Agent Layers (Kummerfeld, Quigley, et al. 2003).

The Sensor Layer is the innermost service layer that represents the range of sensors, which detects the physical environment. The next layer is the Context Layer, which performs core task of filtering and aggregating the raw data from the Sensor Layer and also ensures interoperability between sensors.

The second part consists of an IE Repository and a User Model. An IE Repository is a key element that unifies and manages the whole collection of objects in the environment. The Context Layer interacts with the IE Repository. This handles a global name structure for objects and it manages the naming/subnaming authority.

A User Model overlaps with the IE Repository. The Merino architecture treats them almost the same; the main difference is that the User Model concerns people data only. The User Model contains personal identity, subject to security and to privacy to confirm to the emerging environment, in a local and a global space. On the other hand, the IE Repository holds context information from the Context Layer. Data from the IE Repository which is associated with the user will be stored in the User Model.

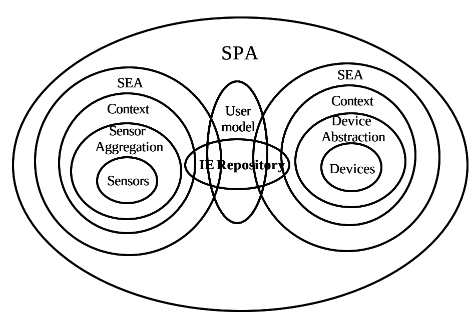

Figure 1. Merino service layers architecture for the Intelligent Environment

For example, data from a movement detector, a Wifi or a Bluetooth proximity detector, and a keyboard activity monitor will be held in the IE Repository, but once it is associated with an individual user, it will be kept in the User Model.

The device and the Device Abstraction Layer are motivated by the need to send data to low level devices in the IE using some device attribute, e.g. a phone number or a Bluetooth MAC address in a mobile phone. A Smart Environment Agent may determine that the context implies "a meeting is in progress", and communicate with the phone to change its state, e.g. to request for switching to a silent mode.

4. DiCPA: A SCALABLE IE DISTRIBUTED CONTEXT PROCESSING ARCHITECTURE

A DiCPA is a scalable context processing architecture based on Merino service layer architecture for IE. The key role of the DiCPA is an IE domain. An IE domain is an administrative domain, which at least contains an IE repository, a resources manager, a knowledge base and various sensors (above the level of dumb sensor that communicate using standard protocol).

A resources manager uses sensors/widget agent to have a direct communication between sensors and devices to the IE repository. A resources manager also distributes sensor data or aggregate sensor data to other IE repository in other IE domains. The structure of an IE domain can be hierarchical (it can have a single higher level IE domain or have several lower level IE domains) or scattered without any level (figure 2). In a DiCPA architecture, the communication between IE domains is based on a request from each resource manager (peer-to-peer basis). The resources manager will send a request to the resolution server to get all information about an object. The resolution server accepts the object's global name and uses persistent mapping from the object's global name to send the persistent location (persistent URL) of the object to the resources manager. Finally, the resources manager sends a request to a relevant resources manager in other IE domain to get the object.

Every object in the IE domain is defined to be a self-describing data-structure. This means that the structures as well as the value are always included in the object's content. This approach has two advantages: firstly, it avoids the complexity of separate schemes and maintaining objects across multiple distributed servers and secondly, it permits meta-tools, e.g. a browser to manipulate the object without knowledge of the specific contents [24]. A particular object has a unique identifier and is located using a search that starts in the local IE domain, and expands to adjacent and higher lever IE domain such as resolution server (using persistent URL or Handle system).

The communication within an IE domain uses logical multicasting. It means that the communication between IE repository and sensors could be implemented using IP multicast with well known group-content-filtering in the receivers or message server distributed content-routing and filtering (such as Elvin or Spread) for an automatic update of object content [25].

For example, consider a person moving between locations. Anna works in the Computer Science Department. The CS department (IE) domain contains the home server for her personal user model. When she walks to the nearby Department of Engineering building, the CS department context layer will use aggregate sensor data to determine that Anna is not present, and other aggregating sensors in the Department of Engineering will detect her presence as an unknown object and ask their local resources manager for the address of the server for this object. The local resources manager multicasts the request to the resolution server and discovers the correct home server for the object – her user model. The object's location information in the home server is then updated accordingly [18].

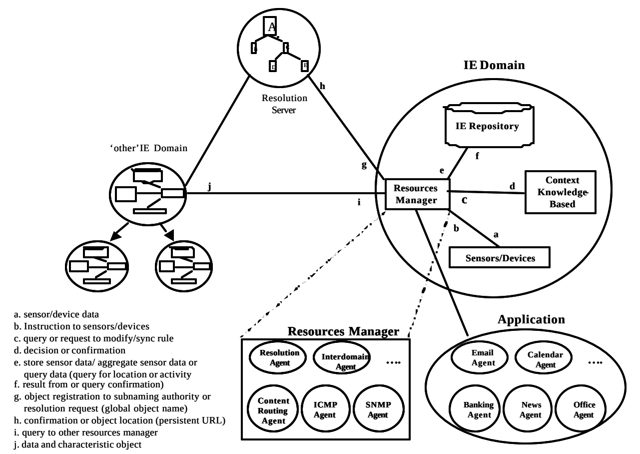

Figure 2. DiCPA: Distributed Context Processing Architecture for an Active Environment

4.1 IE Repository

The IE repository keeps all objects and their characteristics. It holds the rapidly changing state of the objects. It contains all relevant data within the IE domain, e.g. sensor data, aggregate sensor data, context data and rich context data. The IE repository uses distributed object database for the Context Repository and the User Model purposes. It also contains user profiles, registered devices (server, workstation, bridge, router, switch, printer, phone, fax), sensors, network (type - wired or wireless, subnetting, segmentation), data management, location mapping (sensors/devices signal-strength mapping to physical location), and other services data (monitoring data (SNMP/MRTG), hosts, etc.).

Every object in the IE repository has an identifier which is registered in the resolution server for IE resolution purposes, e.g. from URN to URLs.

The resources manager manages the communication between IE repositories and sensors/devices using logical multicasting and the communication between IE repositories using peer-to-peer (P2P) systems. These P2P systems offer distributed hash table (DHT) functionality and exact-match query facility. These systems are extremely scalable; lookups can be resolved in log n overlay routing hops for an overlay network of size n hosts. Exact match lookups are suitable for fetching objects or resolving domain names. To do the lookups, extended SQL can be used.

We use DHTs not only as an indexing mechanism for all object in IE domain, but also as a network routing mechanism.

4.2 IE Resolution

In the IE domain, all objects can be distributed to other physical location or logical location. Every object has a global resolution name and the global resolution name need to be registered in the resolution name server, e.g. PURL or Handle Server for persistent global mapping/resolution purposes. The resolution mechanism is identical to the DNS for the Naming Authority Pointer (NAPTR) Resources Record (RR) to delegate the lookup's name [8,20]. We use the Unique Resolution Name (URN) as a global unique name and the Unique Resolution Locator (URL) as a locator which locates any object, anywhere, anytime.

The purpose of the IE resolution is to identify if an object or a user changes its location, especially for recording the dynamic location due to the user mobility in an unfamiliar Active Office.

4.3 Resources manager

A resources manager is a network management map application that provides information about status of object (devices/sensors) in the IE network. The resources manager detects, controls, manages and concludes the functionality of all objects in the IE domain.

The resources manager will show any failures in the devices/sensors, access points, hubs, bridges and routers. Furthermore, the resources manager will detect any traffic problems and will identify what, where, cause and time-length of problems.

The resources manager has a capability to control and manage the functionality of an IP network using the Internet Control Message Protocol (ICMP) and to 'trap' mechanism using the Simple Network Management Protocol (SNMP), so that failsafe mechanism can be implemented. To set up the resources manager, a set policy (knowledge-based) that determines the acceptable levels of traffics, broadcasts and errors on any devices/sensors at any segments needs to be established.

The resources manager knows all existing objects in the IE domain by querying to the IE repository (context repository or user model layer). If the object does not belong to its IE domain, the resources manager has the mechanism to find the object by querying to the resolution server to get the 'home server' and continue querying to the 'home server' about the details and characteristics of the object if it exist. If it does not exist, the resources manager will tag it as an unidentified object.

4.4 Knowledge-Based Context

A complex infrastructure and service environment requires more shared knowledge for the entities to be able to invoke communication sessions. The knowledge-based service must lead to the growth of services in IE. The entities required a knowledge-based context i.e. an advance knowledge-based based on context service and a shared knowledge-based. The Context knowledge-base requires a protocol, e.g. JINI, JXTA, UPnP, to invoke the services.

Entities are able to support a knowledge-based context which represents the capability of the entities and associated objects to create relations between them. Entities can exchange knowledge-based, and merge with the existing knowledge based and then interpret the knowledge-based context in the end of the services.

When the knowledge-based is merged, it could be growth very big, then the knowledge-based will be needed to shrink, extract and filter to have a better performance but without loosing the essence. Multiple Classification Ripple Down Rules (MCRDR) and Self Organizing Map (SOM) of Artificial Neural Network (ANN) component can be a best candidate to filter 'rule' to produce knowledge maps and capable to extract raw and semantic rule features from sensors/devices .

The knowledge-based context contains context rules of the interaction between the user and the Active Office environment. Rules are used to represent heuristics which specify a set of action that needs to be performed in a given situation.

A generic rule is composed of and *if* portion and a *then* portion.

The *if* portion of a rule is a series of patterns which specify the facts from sensors in IE where the rule would be applicable in the IE domain. The process of matching the facts to the patterns (pattern-matching) is done by an inference engine agent, a specific agent in the resources manager. The inference engine agent will automatically match the facts against the patterns and determine which rules are applicable to the context.

The *then* portion is a set of actions that need to be executed when the rule is applicable to the situation (context). The actions of applicable rules are executed when the inference engine agent is being instructed to begin the execution. The inference engine agent selects the rules, and then the actions of the selected rule are executed. The *if* portion detects its current status and *then* portion determines what actions to be taken based on the context.

The context knowledge-based also has a dynamic knowledge cooperation to allow the rules and the sensor/devices to dynamically affect the actions service selection process within the IE domain or the inter IE domain. The mechanism of the context knowledge-based described above can be implemented using a rule engine and a scripting language, e.g. a Java Expert System Shell (JESS), a Prolog (such as SICtus prolog), a CLIPS, or other function languages.

4.5 Resources Manager Applications

The application consists of user agents including email agent, calendar agent, banking agent, news agent and office agent. The application will be in users' handheld computer to communicate to the Active Office through the resources manager. As shown in figure 2, the application lies outside of the IE domain. It makes it easier for the user to move from one room to the other, even to other unfamiliar IE domain.

The resources manager manages service delivery to the user. All information in user model can be retrieved from IE repository and be stored directly to the user's application by resources manager and applications' agent. The application allows the user to use available features in Active Office based on users' role. The user could also query using context user interface.

5. USER ACTIVITY

Since an Active Office is also a ubiquitous computing environment, sensors, actuators, simple push buttons, sliders, and computer accesses are available in every area. Users can be identified by their mobile computing devices (PDA/handheld), image recognition, or by their activities of accessing available resources anywhere, anytime. We used precise sensors and proximate sensors to locate user's location and collected user's location from both type of sensors[18].

In Active Office, user have regular work schedule. User has routine activity that can be predicted his location in certain time. Users' activity can be represented by user mobility, and user mobility can be seen from user's changing location in significant scale. So, in Active Office once we can capture user location then we can have a pattern of users' mobility [16,17,18].

As described in [3,16] users can be characterized by: identification and authentification, user profile, users' terminal and users' access network characteristics, and service adaptation to user environment.

Firstly, a user identity (unique identifier) and authenticity to prove user as whom he/she claims to be, needs to be recognised before the user accesses the service in a personalised way. The use of login id and password is sufficient as the user authentification.

Secondly, a user profile defines the user's preferred terminal environment, which includes individual display setting, network connection, printer connection, program items, screen color, windows size and position and other specified settings.

The user profile is created based on a user profile skeleton. The user profile can be local, roaming and mandatory. A local user profile is created when the user logs on for the first time; a roaming user profile is created in a 'home' server and is available every time the user logs on to any computer on the network; and a mandatory user profile is a roaming profile that can be used to specify a particular setting for an individual or an entire group of users. The user profile shall be a unique and independent of any access, even if several implementation scenarios are established.

When a user logs on to the system, a user profile which contains a default user setting will be created to allow the user's access to the environment. Once user profiles are created in a certain IE domain, they can be automatically accessed, including the relevant authorization and authentication, by other IE domains if and only if trust between domains is activated. When the user uses his mobile host (PDA) at the first time in the IE domain, the user profile will be defined and relevant data will be synchronized from the server to the PDA.

An example structure of a user profile:

User		User preferences
	LoginID	User Login Id
	Password	Password
	User description	User description
Home Network		Access to home network
	Server name/IP address	Home server where the profile created
	Trusted Key	To establish trusted relation between machines (to 'home' server)
Access Network		Access to network
	Network type	Type of Access Network
	Availability QoS	Availability QoS
Terminal		Terminal setting
	Terminal type	Terminal type (i.e. desktop, notebook, PDA)
	Hardware configuration	The existing hardware
Service Session		Service Session
	Service type	Service type when invoked service
	Service status	Temporary service status

Thirdly, when the user changes from PDA to other terminals, the user's environment may significantly be changed due to different accesses and terminal capabilities. It would influence the way user services are delivered. The procedure to enable network entities to determine the service presentation and capabilities, along with the current access and terminal characteristics of the users will be considered.

Fourthly, a service adaptation has a capability to allow user to customise services in the user terminal in order to adapt service execution and presentation with a common "look and feel" to the user's current environment. The bit rate and quality of the access link have major impact on Quality of Services, especially in the service adaptation to user terminal. When user accesses his IE domain or an unfamiliar IE domain, the service adaptation should deliver the same contents, but can be through a different service execution and adaptation.

6. APPLICATION SCENARIO

In this part, we will describe a scenario of how users and devices in an unfamiliar IE domain get connected.

In the Active Office, we assume that the user brings a portable computer, i.e. a mobile host, rather than simple personal assistant or a pager or phone. The user's mobile host has reasonable capabilities in processing power and memory/storage, remote access and execution mechanism such as Remote Procedure Call (RPC) so that fixed local server could run any applications, which are unavailable on the mobile host. A user, for example, wants to deliver a presentation from a file at a remote server using facilities in an unfamiliar Active Office with an IE domain.

We will describe how this unfamiliar Active Office responds to the user's presence and requests. We assume that an unfamiliar Active Office has a fixed server through a wired LAN which connects to the Internet and the server has a reasonable service to connect to the mobile host through a wireless connection, e.g. IEEE 802.11b, 802.11g, Bluetooth or IrDA, as part of the network infrastructure.

First, the user logs in to his PDA. When the user arrives at an unfamiliar Active Office, his PDA will broadcast its unique hardware identifier i.e. vendor-supplied MAC or BDA address. The local server has a DHCP and a Network Address Translation (NAT) service. The NAT uses IP Masquerading to translate IP addresses. After a handshaking process, the server will send a local IP address to the PDA. At this stage the PDA has established a connection to the network (figure 3).

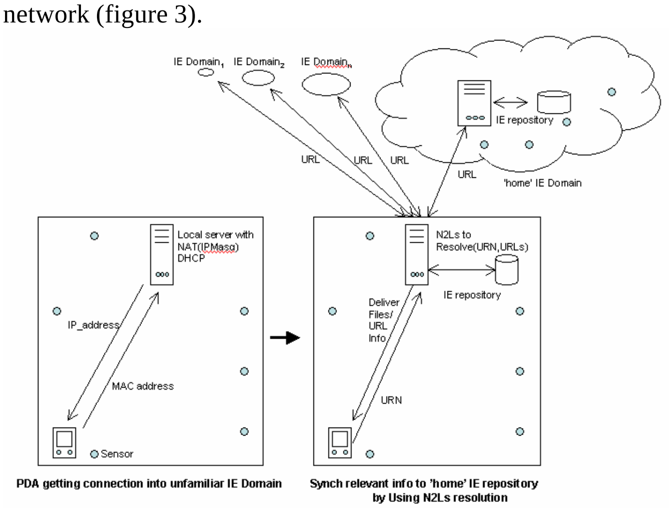

Figure 3. Getting connection into unfamiliar IE domain

There are several ways to get connected to the home server, e.g. using NIS(YP), NIS+, or mounting a directory to the home server. However these ways are not appropriate for an Active Office purposes since we only have limited information, i.e MAC address, instead of home server, login id and password which required for using NIS+ or mounting directory. Therefore we use the resolution server to get connected to the 'home server'. When the IE domain identifies an unknown object (the MAC address of the PDA), its resources manager will send this MAC address in a query to the resolution server to retrieve a URN. For example:

http://resolution.anu.edu.au/1030.52/00-80-BD-08-08-08

protocol	resolution-server	subnaming-authority	MAC-address/ BDA

The response from the resolution server, for example is *jblog@inul.org*.

The resources manager will send another query using this URN to get URL resolution.

For example:

GET /usi-res/N2L?urn:1030.52:jblog@inul.org HTTP/1.0

The result would be:

 # urn: 1030.52:john.blog@inul.org
http://www.inul.org/people/john.blog.html
http://www.inul.org/pub/presentation.pdf
ftp://ftp.inul.org/pub/presentation.ppt

The resources manager in the unfamiliar Active Office then will get the information, such as the person's name and administrative domain and where the file for the presentation is available. When the resources manager stores the location information to the user's PDA, at the same time the resources manager in the 'home' domain changes the location status in the IE repository (user model layer) [17,18].

We have implemented user's location based on history data (predicted user location) and develop user interface called SpeechCA using Java Speech API[16]. We also have studied service delivery based on user location and user's mobility in an Active Office, so, when user moved to 'other' IE domain, the user still has full access to internet but incoming and outgoing data it's used his IE domain using WiFi and printing to closed printer in 'other' IE domain using Bluetooth [17,18]. We used three different operating systems (Solaris, Linux and Windows servers) on the three buildings. These implementation is a part 'proof of concept' of the distributed architecture of the IE.

7. SUMMARY AND FURTHER STUDY

This paper proposed a DiCPA architecture - a distributed context processing architecture - for an IE which is simple, efficient, scalable, fault tolerant and applicable to be implemented across a range of heterogeneity computing platforms.

This paper has described the requirement for scalable context processing in an Active Office using DiCPA architecture as an implementation model of IE. We also presented an overview of the Merino service layer architecture, and distributed context processing architecture which then followed by a description of IE domain, IE resolution, IE repository, user modeling, resources manager, knowledge based, application and interoperation between sensors.

We showed how the scalable distribution of the context and the user model information are managed for recognising user activities. We also described an illustration of how users get connected in an unfamiliar Active Office in other IE domain and how to monitor user activities.

Further research topics that can be considered from this work are the modelling of sharing context rules between IE Domains in the distributed context knowledge-base and the requirement to have a flexible and efficient communication between resources managers in several IE domains to form a society of IE domains

Acknowledgements

The authors would like to acknowledge the support of the Smart Internet Technology CRC.

References

1. Abowd, G. D. (1999). "Classroom 2000: An Experiment with the Instrumentation of a living Educational Environment." IBM Systems Journal (Special Issue on Pervasive Computing), Vol. 38 No. 4.
2. Abowd, G. D., C. G. Atkeson, et al. (1996). Teaching and Learning a Multimedia Authoring: The Classroom 2000 Project. Proceeding of the ACM Multimedia'96 Confereence.
3. Bougant, F., F. Delmond, et al. (2002). "The User Profile for the Virtual Home Environment." IEEE Communication Magazine (January 2002): 93-98.
4. Brooks, R. A. (1997). The Intelligent Room Project. Proceedings of the second International Cognitive Technology Conference (CT'97), Aizu, Japan.
5. Cisco-Inc. (2000). iHome: Australia's first internet-enabled home, www.iHome.com.au. 15 May 2002.
6. Coen, M. H. (1998). Design Principles for Intelligent Environment. Proceedings of the 1998 National Conference on Artificial Intelligence (AAAI-98).
7. Coen, M. H. (1999). "The Future of Human-Computer Interaction, or how I learn to stop worrying and love my Intelligent Room." IEEE Intelligent Systems (March/April 1999).
8. Daniel, R. and M. Mealling (1997). Resolution of Uniform Resource Identifiers using the Domain Name System. RFC 2168.
9. Dey, A. K., G. D. Abowd, et al. (1999). A Context-Based Infrastructure for Smart Environments. 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99).
10. Flanagan, J. L. (1999). "Autodirective Sound Capture: Towards Smarter Conference Rooms." IEEE Intelligent Systems (March/April 1999): 16-18.
11. Harter, A. and A. Hopper (1994). A Distributed Location System for the Active Office, IEEE Network Vol 8, No 1 pp.62-70.
12. Harter, A., A. Hopper, et al. (2001). "The Anatomy of a Context-Aware Application." Wireless Networks **1**: 1-16.
13. Hong, J.I., J.A. Landay. (2001) An Infrastructure Approach to Context-Aware Computing. Human-Computer Interaction (HCI) Journal. 16(2-3)
14. Kidd, C. D., R. Orr, et al. (1999). The Aware Home: A living Laboratory for Ubiquitous Computing Research. in Proceedings of the Second International Workshop on Cooperative Buildings - CoBuild'99. Pittsburgh, PA.
15. Kummerfeld, B. Quigley A. et al.(2003). Merino: Towards an intelligent environment architecture for multi-granularity context description. Workshop on User Modelling for Ubiquitous Computing.
16. Mantoro, T. and C. W. Johnson (2003). Location History in a Low-cost Context Awareness Environment. Workshop on Wearable, Invisible, Context-Aware, Ambient, Pervasive and Ubiquitous Computing, Australian Computer Science Communications (ACSW 2003), Volume 25, Number 6, Adelaide, Australia. pp 153-158.
17. Mantoro, T. (2003). User Location and Mobility for Distributed Intelligent Environment. Adjunct Proceedings, The Fifth International Conference on Ubiquitous Computing (UbiComp 2003), Seattle, Washington, USA, 12-15 October, 2003. pp. 255-256
18. Mantoro, T., Johnson, C. W. (2003). User Mobility Model in an Active Office. The First European Symposium on Ambient Intelligence (EUSAI 2003), LNCS 2875, Springer, Eindhoven, The Netherlands, 3-4 November 2003. pp 42-55
19. Manzoni, P. and J. Cano (2003). "Providing interoperability between IEEE 802.11 and Bluetooth protocols for Home Area Networks." Computer Networks **42**(1): 23-37.
20. Mealling, M. and R. Daniel (2000). The Naming Authority Pointer (NAPTR) DNS Record. Updated RFC 2168.
21. Mozer, M. C. (1999). "An Intelligent Environment Must be Adaptive." IEEE Intelligent Systems (March/April 1999): 11-13.
22. Orr, R. J. and G. D. Abowd (2000). The Smart Floor: A mechanism for Natural User Identification and Tracking. Atlanta, Georgia, Georgia Institute of Technology.
23. Priyantha, N. B., A. Chakraborty, et al. (2000). The Cricket Location-Support System. 6th ACM international Conference on Mobile Computing and Networking (ACM MOBICOM), Boston, MA.
24. Schilit, W. N. (1995). A System Architecture for Context-Aware Mobile Computing. The Graduate School of Arts and Sciences. Colombia, Colombia University**:** 144.
25. Segall, B., D. Arnold, et al. (2000). Content Based Routing with Elvin4. in Proceding Australian Unix and Open Systems Users Groups 2000 (AUUG2K).
26. Thomas, B. H., V. Demczuk, et al. (1998). A Wearable Computer System with Augmented Reality to Support Terrestrial Navigation. In Proc. 2nd International Symposium on Wearable Computers, IEEE Computer Society, Pittsburgh, Pennsylvannia.
27. Weiser, M. (1993). "Some Computer Science Issues in Ubiquitous Computing." Communications of the ACM **6**(7): 75-84.