Creating an Ambient Intelligence Network using Insight and Merged Reality Technologies

Dr. Ralph Moseley
School of Science & Technology
Middlesex University
London, UK
r.moseley@mdx.ac.uk

***Abstract*—Humans live and work in environments which are essentially "dumb", though recently, due to information networks, devices within these areas have increasingly become connected. The system presented here builds on previous work to create an ambient intelligence zone using facets of a merged reality system and a new process based on recognition/insight patterns. When combined, agents within the system communicate and react as one to form a responsive ambient intelligence at a given location.**

***Keywords*— Ambient Intelligence, Robot Autonomy, Merged Realities, Intelligent Systems**

I. Introduction

Current technologies are largely disparate in terms of communication and lack the ability to share resources en-masse. The Internet of Things (IoT) [1] is a step toward sharing of information between devices. The IoT is a common term used to describe the capacity of the internet to act as a network for a multitude of devices. These devices range from personal computers, cameras and pace makers, through to power station controllers. The internet acts as a basic communication mechanism. There is also the idea of the internet for robots[2][3].

Ambient intelligence (AmI)[4] refers to electronic environments that are sensitive and responsive to the presence of people. In an ambient intelligence world, devices work in concert to support people in carrying out their everyday life activities, tasks and rituals in easy, natural way using information and intelligence that is hidden in the network connecting these devices. As these devices grow smaller, more connected and more integrated into our environment, the technology disappears into our surroundings until only the user interface remains perceivable by users.

The ambient intelligence paradigm builds upon pervasive computing, ubiquitous computing, profiling practices, context awareness, and human-centric computer interaction design and is characterized by systems and technologies that are:

- Embedded: The networked devices are integrated into the environment
- Context aware: The communicating devices recognize the user within a situational context
- Personalized: the devices can be altered to fit to a user's needs
- Adaptive: the devices can change in response to the user
- Anticipatory: the devices can anticipate the user's requirements without complex mediation

Here a previously developed system called VECSED - Virtual Environment for Control, Simulation and Electronic Deployment [5] is built upon. This software not only allows resources (processes etc.) to be shared along with information but utilizes Virtual Reality as a modelling tool and space in which to understand the environment in which it works [6].

What the current system will do:

- Allow devices (robots, computers, sensors, …) real and virtual to communicate
- To share (computing, processing power) resources
- Allow real space to be mapped (manually) into virtual space
- To track real objects in virtual space

The experiments here are to explore how the system can be made to:

- Automate mapping of real space into virtual space
- Track entities through the space and reconfigure or task the controlled area
- Anticipate patterns of behaviour
- Extend the sharing of resources (power, mechanical capabilities …)

The objective is to be able to track an agent (which could be human or otherwise) not connected to the system, predict behaviour and in some sense react appropriately.

II. VECSED – THE CORE SYSTEM

A. Basic description

VECSED works by creating a network which produces a bridge between real life and virtual devices. Each device communicates with a controller holding a database which it relates its resources and functionality to. Other information, such as geospatial location (or relative coordinates in a virtual environment) and time last seen, are also stored.

It is possible, using VECSED, to map between the virtual environment and the real environment in real time. Along with this it is then possible to track a real agent and map it to a model in 3d virtual space.

Novel human input systems have been created to feed data streams into VECSED and produce a biofeedback system, essentially producing an adaptive environment in an immersive world.

B. The Experiment

In this experiment it is necessary that:

- Non-system agents are detected, then identified, where possible and assigned an identifier.
- Non-system agents are recorded going through an area controlled by the system and predictions of actions created on this basis built on probability. Probabilities are stored in specific units known as Insight modules.
- An avatar is produced to represent the non-system agent in the 3D virtual space, in real time.

C. Sensing

The system must recognize basic activities in the area: An object when entering the environment interacts; this could be physically (by moving things around) or via information exchange (throwing switches, entering data, voice). An object in repetition, forms a pattern or signature of behaviour which can be identified [7].

An object when entering an environment could be identified through visual or auditory (for example, voice).

An object can be classified as a type and then uniquely identified as an individual.

Once this entity is categorized in the database it can be given particular pathways of actions within the controlled environment.

An agent could also be identified purely by the actions they perform and thereby create a type in itself – a ghost, where a physical form would be hard to detect, given the sensors available.

Sensors required:

Light sensors (camera, simple trigger)

Sound sensors (samplers, triggers)

Motion detection

Triggers (Weight, switches, …)

III. THE PRACTICAL EXPERIMENTS

The basic idea here is to test the premises set out in this paper, building on previous development work with VECSED.

To do this, a physical and virtual space is also required. The VECSED system will be used as the core of the system which will allow devices to communicate, deploy resources and build a virtual replica to work within. Any physical device also has its own simulacrum within the virtual space which is linked via VECSED. Any information about the environment and states of devices etc. is also transferred into the virtual map.

A. Robots

To explore the central ideas in this paper several robots (see figures 1 and 2) and devices were built with various sensors and capabilities, to implement the idea of ambient robotics [8][9]. A basic robot type consisted of a two-wheel base, motor driver unit, CPU card (Arduino based), a prototyping card and a WIFI card. This base unit was originally developed for work in another project developing a merged reality virtual assistant [10]

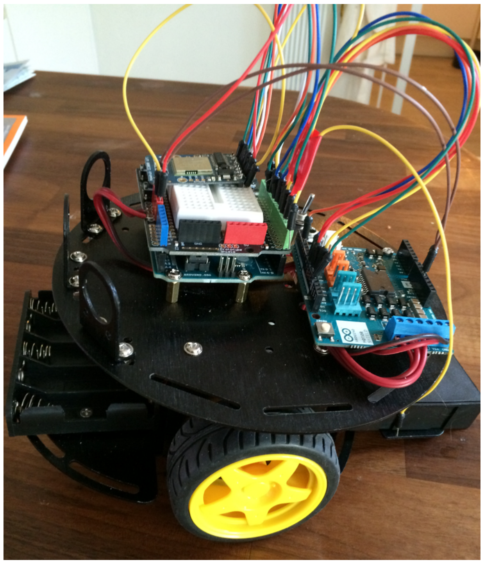

Fig. 1. One of the initial baseline bot units

From this basic unit sensors and other additions were added:

Ultrasonic range finder

Temperature / Humidity sensor

PIR Sensor

Light level detection sensor

Bluetooth Low Energy capability [11]

The initial robot consisted of an Arduino UNO, a prototyping board (or prototyping shield, designed by DFRobot) through which control was administered to the Adafruit CC3000 WIFI break-out board, and Arduino motor controller board (known as a Motorshield, in this case, Adafruit, v2.3) which was connected to the base unit's motors.

This base unit could connect itself via any WIFI to the VECSED server, and connecting itself into this network from anywhere in the world (as long as there was a known WIFI connection). Once an initial handshake occurred it would report its details; IP address, host, and capabilities. This information is stored in the VECSED server database for communication among its peers.

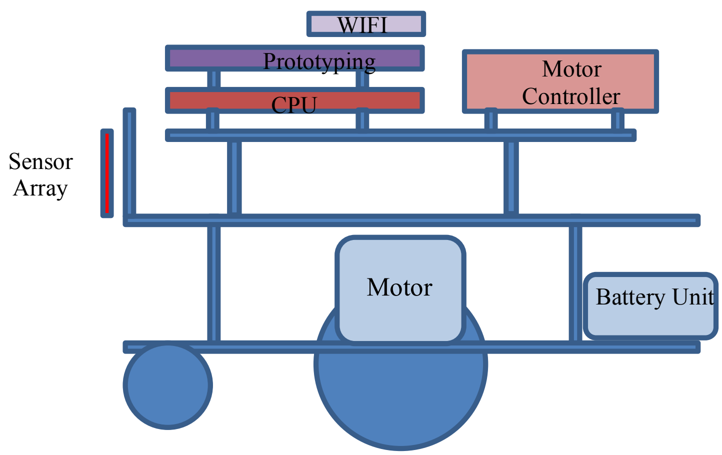

Fig. 2. Schematic diagram of the baseline robot

In practice, the building of the initial robot presented a few difficulties; the normal stacking of such boards created particular input output problems with the lines required so a prototyping board was utilized to breakout various connections from the UNO.

Another difficulty met was the WIFI board which frequently hung; a work-round consisted in sensing there was a persistent connection and when dropped a reboot would be induced thus clearing the problem with the chip. Later robots used a different board to get around this problem and also, explored here, is how to make the system much more resilient to problematic communications giving options to entities that belong to it.

The base unit actually consumed more power than anticipated which seemed in the main part to be the WIFI board rather than the motors.

Larger capacity boards were used in further robots, such as the Arduino 2560 which offers greater memory for programs and data, as well as the critical input / output lines for attaching further sensors and equipment.

The later base unit was upgraded to a BLuno MEGA 2560 which included the increased capacity of the Arduino 2560 with the added benefit of Bluetooth Low Energy 4.0 for an additional supported communication method [12][13][14]. This gave the main processing board of the basic robot a much better extended capability which included an ATmega2560 with a clock frequency of 16MHz, 8KB SRAM static storage, a flash memory of 256KB and 4KB EEPROM memory capacity. It includes 54 digital I/O (14 of which are PWM outputs) and 16 analogue inputs. Bluetooth, as stated is built in together with support for HID and iBeacons.

This was more than enough for the purposes of the sensors, the basic firmware and system for communication with VECSED [15].

It was made possible for a base unit to be directly controlled via the web at least for initial test purposes and before hooking into the core system VECSED.

Base units were then equipped with more unique and individual capabilities for the purposes of experimentation.

- Bot1: **Sentry** – motion sensors, light
- Bot2: **Mapper** – Range finder, bump sensors
- Bot3: **Communicator**
- Bot4: **Refueler**
- Bot5: **Environmental** – temperature, humidity
- Bot6: **Sampler** – photographic, auditory, camera

IV. AN INTELLIGENT SPACE

One of the key ideas of this paper is the reproduction of real life (RL) space inside a virtual one, which is then used as a work space and map for processes and devices. Here, we use OpenSimulator [16] as the virtual world system which can run independently of other public spaces such as Second Life (SL) or, be linked. OpenSimulator allows more control, is free to use and has more resources available than immediately accessible commercial virtual spaces. The OpenSimulator system was deployed on a machine along with VECSED. An interface was developed between the main server and the virtual world allowing automated building and deployment of scripted processes. This could, due to how similar the two systems are be easily swapped to run in SL if desired.

A physical zone was created within a room to test the capabilities of the robots and devices. This initial space was an area 5m x 5m.

A. System updates

The updated VECSED server works as before in terms of setting up communication. That is, when a real or purely virtual device is turned on or instantiated, the program running in that device will attempt to initiate contact with the VECSED central server. Once established the details of its network properties, IP or host location etc. are shared so further information is easily swapped. These details are stored in the server with other details about that system. This initial communication is a simple registering, where it can be checked if it has previously had contact and identified itself. A device will then have to authenticate itself to be a part of the network.

A RL device can then communicate in the network between other RL devices and/or virtual components. It can, in effect, have a virtual component itself with which to interact with in the intelligent space. They act as true transponders to one another and are not simply a transmitter or receiver.

A virtual device can switch a real mechanism on or off and a RL device can act on a virtually instantiated entity.

The robots in the network are connected through the system and have their own true internet similar to other ideas [17], though in this case, they do have a virtual component too.

Once connected to the central system, as described here and in the more detailed previous paper on this matter, individual tasks were given to the robots.

The mapper bot was set to explore the space as nothing was made known to the central system or the bots themselves. This bot used ultrasonic range finders and bump sensing to determine the area it found. This information was relayed to the networked system which produced a suitable 3D virtual area.

Once the basic metrics of the controlled space were determined the second bot was set into action – the environment bot. This determined features of the space such as temperature, humidity and light/shadow regions. Again, this information was fed into the virtual space.

The space was then investigated by the sampler bot which took photos and listened to the environment. Baseline levels of noise were recorded. It could be possible to take the idea here even further where the sampler bot actually takes photographic samples for textures to replicate in the virtual world.

The refueler bot was used by the system to store power for the other bots, when patterns of light were recognised and maps, the refueler would enter the areas determined and collect power via its solar panels. An interesting aspect of the system was to keep track of the energy as a resource which can be collected for use.

These devices collect all the metrics for a given area which is then translated into the virtual space; it is possible you could have one bot that did all these tasks, although the loading from what was seen here, would have been quite high on the battery.

Once the zone is set up all devices which operate within in it are registered. The various sensors keep track of objects or agents [18], such as unauthorised humans [19] entering and leaving the zone and watch for any communication. Changes in the physical zone are duplicated into the virtual. Device statuses, as we have seen previously, are also replicated. Information about the zone is available to any connected device which has authenticated with the system and network. So, any device may enquire as to the metrics of the room, or environmental properties that have been recorded.

B. Bluetooth Low Energy - Identification

One aspect which was explored was that of identifying individuals entering the space. In this case, they were identified when they enter the space by an app on their phones which via BLE provided a transponder identification “blip”. The bots can then perform pre-programmed functions according to the particular person that has entered the space. In this case, lights were activated, an entry logged and broadcast through the network. Basically any function of the bots can be performed should a specific individual enter the zone. This could be automated mechanical functions or internet based such as a Skype broadcast.

Bluetooth Low Energy (BLE) 4.0, is particularly useful. Also known as Bluetooth Smart, while using the same 2.4 GHz radio frequencies as Classic Bluetooth, BLE uses a much simpler modulation system as well as a less complex protocol setup. BLE is cited as having low power (operating for months or years on a small button cell), a very small circuit footprint and compatible with a large range of devices such as mobile phones, computers and specialised products such as fitness bands. It is possible for a device to not require pairing and other aspects of the Classic Bluetooth and thereby simplifying the process of establishing links – and therefore perfect for the kind of experiments and tasks created here.

Using BLE it was possible to analyse when a person entered the area, once in the area other mechanisms could be used for tracking the individual. This included the use of the ultrasonic sensors.

Once in the area, a notifier app was made to deliver a data payload, which included an identifier for authentication and a list of tasks specific to that individual. This data payload is uploaded to the main system which as long as authenticated will activate the task list.

The notifier app was developed in Android Studio but could easily be targeted to other formats such as Apple iOS. It was in effect acting as a form of RFID device, though it was possible to store a large amount of information rather than just identifying codes. As the strength of signal can be worked out through all nearby receivers It was also possible to determine proximity to, and in, the area concerned. This was useful to determine that someone had actually neared the area, entered it and also whether they had left.

C. Adding Resilience to the System

The co-operating bots [20] normally share their own WIFI network which they sign into on boot-up. A secondary system was explored utilising the BLE only. Bots had their WIFI capability turned off and instead were made to sign on to the system via their BLE links. This worked well using less power and added a secondary communication path making the system more resilient. Interfaces between the components were upgraded whereby the switch between the two mechanisms was instant should the main form drop-out. This was particularly useful after some of the WIFI cards seemed prone to rebooting after entering a panic state. The BLE system as a backup allowed seamless operation even in those bots that suffered problems of this kind.

D. Anticipatory Processing – Intelligence with Insight

When an entity is authenticated to an area, the information about that entity is loaded into the system and any reconfiguration to that space will take place. It is also possible to configure "macro" gestures or triggers within the space. For example, when switch A is thrown and in a particular place then bots initiate a sequence of activity which could be turning on lights, activating a stream or suchlike. These patterns of behaviour and triggers were stored in a database and the algorithms developed were called Insight modules.

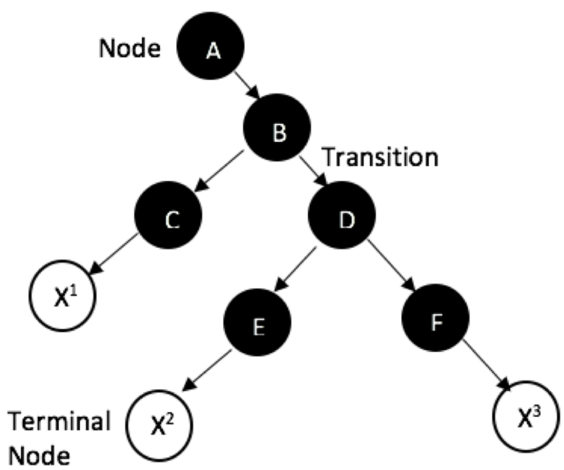

Fig 3. Simple input patterns and terminal event

Figure 3 describes how events, visualized as nodes, form sequences of actions leading to a terminal point where some action has been completed. For example, the pattern (or signature) of end point X^2 is a transition of nodes between A, B, D and E. The end point itself may be the culmination of some action(s) in the target area such as crossing the area and closing curtains and turning on a light.

Input event	Switch A	Switch B	Movement Sensor C	Movement Sensor D	Terminal Event
	X			X	X1
		X		X	X2
	X	X	X		X3

Fig 4. Simple input patterns and terminal event

Input event	Switch A	T^1	Switch B	T^2	Movement Sensor C	T^3	Movement Sensor D	T^4	Terminal Event	TO
	X	1					X	2	X1	10
			X	3			X	4	X2	8
	X	3	X	1	X	4			X3	12

Fig 5. Input patterns with added transition time

The resulting basic pattern of these transitions are further described in figure 4, showing example inputs leading to terminal events. Temporal attributes are then the next to consider. This was done by adding actual transition times and time out (TO) values (see figure 5) where an end-point has been reached, giving a unique 'signature' to a particular set of actions.

The system then becomes event aware and builds patterns of learned experience [21] as logical pathways in probabilistic networks. Other modules could be added to expand this ability with other avenues of learning too [22]. The Insight module in effect generates tables of logic patterns as detected by actions within the designated area and assigns a level of probability as that causal chain is repeated, a weighting being placed on each possible branch point. Complete sequences become a recognizable action which is an enumerated within a database of possible actions. From a human perspective, this may be "enter the area, switch lights a and b on and send a message to Peter" or "enter the area, move object X to location Y". The connections between each element being given the probabilistic weighting depending upon the number of times this pathway has been followed before.

It would be possible to create user-defined patterns, rather than observed, which the system "learns" and then incorporates into the modules. All bots and devices have access to this information and could be set to anticipate or respond to the signature of a learned pattern.

E. Spatial Modes

The area itself could be seen as entering several different modes of operation:

1) Initial

This mode is the initial set-up phase; an area shape and size is properly determined. It is mapped into the virtual world and a representation is built, in this case in OpenSim, but could equally be SL which shares a common structure and code.

2) Housekeeping

The area is maintained and observed; data is collected. Resources collected.

3) Entry Detected

An object, or entity is detected moving into the area. If the entity is known and carrying the notifier app an authentication process is run, which is similar to all devices which join the network. If the authenticated correctly, a payload is delivered which carries the instructions for system reconfiguration or tasks.

4) Reconfiguration/Response

The reconfiguration can typically take place after a user enters the area. This may be for specific help to that individual – turning on lights, communication streams, anything that can be automated in effect.

V. CONCLUSIONS

There are several levels to this enhancement of the VECSED system:

- Automatic mapping of physical to virtual
- Bots that can work together to fulfil some purpose
- ID authentication within a specific area
- Delivery of the data payload and communication to central system
- Reconfiguration of physical and virtual space
- Automated learning of patterns of recognizable behaviour within an area according to probability
- Use of BLE as backup or main communication system should WIFI be limited or not available
- Distribution and collection of resources, automatically, such as power
- Remote entry into virtual space by individual via viewer or virtual headset

These enhancements draw on several key, and relatively new technologies. These include development of better and more capacity processor boards such as the BLuno MEGA 2560 with BLE; the integration of virtual reality hardware and software and more capable mobile devices such as phones complete with Bluetooth and app development.

The central system needed only additions in terms of communications and some of the advanced features such as organising of devices and resources much of the rest, authentication, data collection and communication between the virtual and RL remained as discussed in the previous paper.

Several concepts have been developed and recognized within this work:

Physical interaction. A human can enter the space and interact with the devices in that space (which is replicated into the virtual). The mechanisms in that space can be trained to recognise the activities and assist or respond.

Remote interaction via avatar. A human can interact remotely with the physical and virtual space by visiting the space using only an avatar, logged into the space.

Remote interaction via headset. A human could easily visit the replicated virtual space and interact with it using appropriate VR headsets and controllers. He could in effect, step into one of the bots (or multiple, as they are networked), such as the camera bot and drive it, watching through its camera stream.

Furthermore, the system was made intelligent in the sense of being able to build its own context, collect, and more importantly respond to, causal patterns of agents which occur within a designated area.

REFERENCES

[1] A. Whitmore, A. Anurag, The Internet of Things—A survey of topics and trends, Information Systems Frontiers 17.2, 2015, 261-274.

[2] Robo-earth, http://roboearth.org/, Retrieved 25th September 2016

[3] R. Majumdar, Robots at the Edge of the Cloud, International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Springer Berlin Heidelberg, 2016.

[4] J.C. Augusto, N. Hideyuki, and A. Hamid. Ambient intelligence and smart environments: A state of the art, Handbook of ambient intelligence and smart environments. Springer US, 2010. 3-31.

[5] R.Moseley, Merged Reality Systems: Bringing together automation and tracking through immersive geospatial connected environments, Proceedings Science and Information Conference (SAI), IEEE, 2015, ISBN 978-1-4799-8547-0

[6] C. Ramos, J.C. Augusto, D. Shapiro, Ambient Intelligence: The next step for Artificial Intelligence, IEEE Intelligent Systems, 2008, ISSN: 1541-1672

[7] D.U. Atmojo, "System-level approach to the design of ambient intelligence systems based on wireless sensor and actuator networks." Journal of Ambient Intelligence and Humanized Computing 6.2 (2015): 153-169.

[8] N. Verstaevel, Principles and experimentations of self-organizing embedded agents allowing learning from demonstration in ambient robotics, Future Generation Computer Systems (2016).

[9] D. Jalamkar, A. A. Selvakumar. Use of Internet of Things in a Humanoid Robot-A Review, Advances in Robotics & Automation (2016).

[10] J. Kaur, SPECIE - Two Worlds Unite: Working Toward Socially Connected Robot for Real World Navigation and Communication, UG Final Thesis, Middlesex University, 2015

[11] C. Gomez, J. Oller, and J. Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology, Sensors 12.9 (2012): 11734-11753.

[12] K.Townsend, C.Cuff, R. Davidson, Getting started with Bluetooth Low Energy, O'Reilly, 2014, ISBN: 978-1-491-94951-1

[13] A.Allan, D. Coleman, S. Mistry, Make: Bluetooth, Maker Media, 2016, ISBN: 978-1-4571-8709-4

[14] J. Lee, Jin-Shyan, S. Yu-Wei, and S. Chung-Chou, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi." Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE. IEEE, 2007.

[15] K. Murakami, Supporting robotic activities in informationally structured environment with distributed sensors and rfid tags, Journal of Robotics and Mechatronics 21.4 (2009): 453.

[16] OpenSimulator, http://opensimulator.org/wiki/Main_Page, Retrieved 25th September 2016

[17] E. Cervera, The robot programming network, Journal of Intelligent & Robotic Systems 81.1 (2016): 77-95.

[18] J. Gascueña, IDK and ICARO to develop multi-agent systems in support of Ambient Intelligence, Journal of Intelligent & Fuzzy Systems 28.1, 2015, 3-15.

[19] J. Liu, W. Meng, F. Bo, iBotGuard: an Internet-based intelligent robot security system using invariant face recognition against intruder, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 35.1 (2005): 97-105.

[20] Y. Tang, Tracking control of networked multi-agent systems under new characterizations of impulses and its applications in robotic systems, IEEE Transactions on Industrial Electronics 63.2, 2016,1299-1307.

[21] A. Yachir, Event-Aware Framework for Dynamic Services Discovery and Selection in the Context of Ambient Intelligence and Internet of Things, IEEE Transactions on Automation Science and Engineering 13.1, 2016, 85-102.

[22] N.R. Ramli, R. Sazalinsyah, O. Mashanum, Learning in Immune Network Algorithm for Multi-Robot Cooperation, Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 8.2, 2016, 111-116.