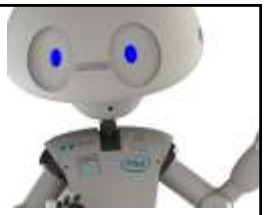
KAU Training Course



Intelligent Environments Training Package 2

Course Lecturer:
Vic Callaghan

Lab Trainers:
Bo Yao (Marc Davies in Essex)

http://victor.callaghan.info

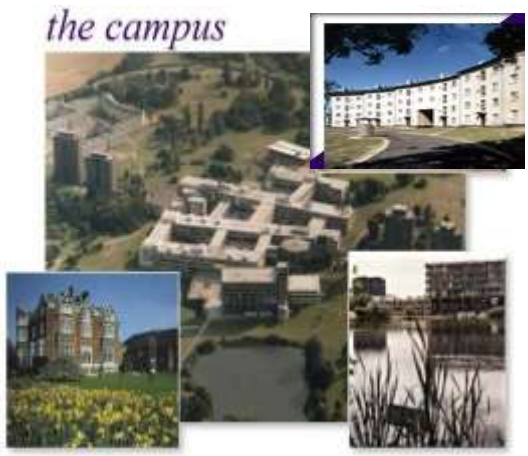
http://victor.callaghan.info/publications/misc/2014_KAU_Training-IE_June2014.pdf
http://victor.callaghan.info/publications/misc/2014_KAU_Training-IE_Dec2014.pdf

Essex University

Recap: About Me

- Professor of Computer Science at Essex University
- Member of Intelligent Environments Group, Digital Lifestyles Centre & School of Computer Science and Electronic Engineering
- Expertise in intelligent environments artificial intelligence, Agents, Embedded-Computing, End-User Programming, Mixed Reality & Internet-of-Things, Product Innovation.
- Part of organizational team for numerous conferences, workshops, journals

the campus



- ***Parkland of 200 acres***
- ***Royal Charter in 1965***
- ***11,939 students***
- ***27% post graduates***
- ***42% overseas (130 countries)***
- ***Ranked 9th in UK for research***

Essex University

2

Recap: Reading & Information

- Course notes – are available to download from:
  - http://victor.callaghan.info/publications/misc/2014_KAU_Training-IE_June2014.pdf
  - http://victor.callaghan.info/publications/misc/2014_KAU_Training-IE_Dec2014.pdf
- Lab Notes (written by Marc Davies)
- Lab Guidance (provided by Bo Yao)
- Research papers (http://victor.callaghan.info/publications/)

Recap: Some Useful Papers

Papers– You will find a lot of useful papers at **http://victor.callaghan.info/publications/**

- **General Overview of Area** – Victor Callaghan, "*Intelligent Environments*", Chapter 5 in book called 'Intelligent Buildings' published by ICE Publishing, August 2013, available from http://victor.callaghan.info/publications/2013_IntelligentBuidings(IntelligentEnvironments).pdf
- **General introduction to embedded agents** – Callaghan V, et-al *"Programming iSpaces: A Tale of Two Paradigms*", in Springer-Verlag book Intelligent Spaces: January 2006 available from http://victor.callaghan.info/publications/2006_API06(ProgrammingISpacesATale).pdf
- **Introduction to Adjustable Autonomy** – Matthew Ball, Vic Callaghan, "*Managing Control, Convenience and Autonomy – A Study of Agent Autonomy in Intelligent Environments*", Journal of Ambient Intelligence and Smart Environments, IOS Press, pp 159 – 196, Volume 12, 2012, available from http://victor.callaghan.info/publications/2012_IOS12(ManagingControlConvenience).pdf
- **Introduction to Programming-by-Demonstration** – J. Chin, V. Callaghan, G. Clarke, "*End-user Customisation of Intelligent Environments*". In the handbook of Ambient Intelligence and Smart Environments, Springer, 2010, Spring, pp 371-407, available from http://victor.callaghan.info/publications/2009_JAISE09(EndUserCustomisation).pdf

Recap: Some Useful Videos

General – you will find a number of videos at **http://victor.callaghan.info/videos/**

- **iSpace** – an intelligent environment at Essex, (shown as part of EU project, Atraco) – http://www.youtube.com/watch?v=W2-wVlh4zFY
- **Embedded-Agents** – an old but informative look at embedded agent design http://www.youtube.com/watch?v=Ej3Hs4sC2DM
- **Adjustable Autonomy Agent** – introduction to variable intelligence http://www.youtube.com/watch?v=C_0yW6dQdZ4
- **Pervasive Interactive Programming** – demonstration of users programming smart spaces http://www.youtube.com/watch?v=TP56Y1Gh-3Y
- **Personal Operating Spaces** – an idea for mobile virtual spaces http://www.youtube.com/watch?v=AcFXeMDkTig
- **Blended Reality Desk** – Mixed reality using the desk in the new KAU lab http://youtu.be/rIjqUBvLQoo

Recap: Training Packages

- Training Package 1 – **June 2014**
  - **Focus** – Understanding and building stand-alone intelligent environments

- Training Package 2 – **December 2014**
  - **Focus** – Understanding and building connected (scaled-up) things, agents and intelligent environments.

THIS IS THE TRAINING PACKAGE WE ARE FOLLOWING FOR THE NEXT 3 DAYS

Training Package 2 – Outline Syllabus

Part 4

1. Introduction to multi-agent environments (rationale)
2. Architectures and topologies for embedded multi-agent systems
3. Network standards for embedded multi-agent systems

Part 5

1. Communication languages for embedded multi-agent systems
2. Configuring embedded multi-agent systems
3. Ontology for embedded multi-agent systems

Part 6

1. Debugging Multi-agent system
2. Human aspects of embedded multi-agent systems
3. Product Innovation

Training Package 2 – Learning Outcomes

- After completing this package, participants will be expected to be able to:

1. Explain the issues governing the specifications of embedded multi-agents architectures (software and hardware).
2. Explain the issues involved in the practical development of embedded multi-agent products and environments.
3. Design & program embedded multi-agent environments.
4. Debug Multi-agent system

Recap: Lab Work (Uses Buzz-Boxes)



- Desk-top Intelligent Environments (eg iSpaces)
- FortiTo – Mexican Company run by Victor Zamudio (ex Essex PhD student))
- Upgradable with 30+ modules (including mobile phone)
- Programmed in in, Java, C, or Python etc

Training Package 2 – Lab Work

- **Training Exercises**
  - **Lab Exercise 4** – Introduction to multi-behaviour agents
  - **Lab Exercise 5** – Introduction to multi-agents and multi-spaces.
  - **Lab Exercise 6** – An open innovation assignment (create a design of your own).

Part 4

Overview

- About 'Training Package 2' (just did that)
- Introduction to multi-agent environments (rationale)
- Network standards for intelligent environments
- Topologies for embedded multi-agent systems

Joining an Essex-KAU Experiment (1 hour)


The evaluation is divided in 3 parts:

- **First 10 minutes** – Preliminary online survey to determine your familiarity and experience with Mixed Reality and Smart Objects.
- **Second 35 minutes** – Simple programming in collaboration with other student located at Essex using a virtual world & BuzzBox.
- **Final 15 minutes** – Short questionnaire about your experience.

Total duration of the activity: 60 minutes



If you are interested in taking part in this experiment have a look at this video:

http://youtu.be/rIjqUBvLQoo

and contact Anasol Pena-Rios at acpena@essex.ac.uk

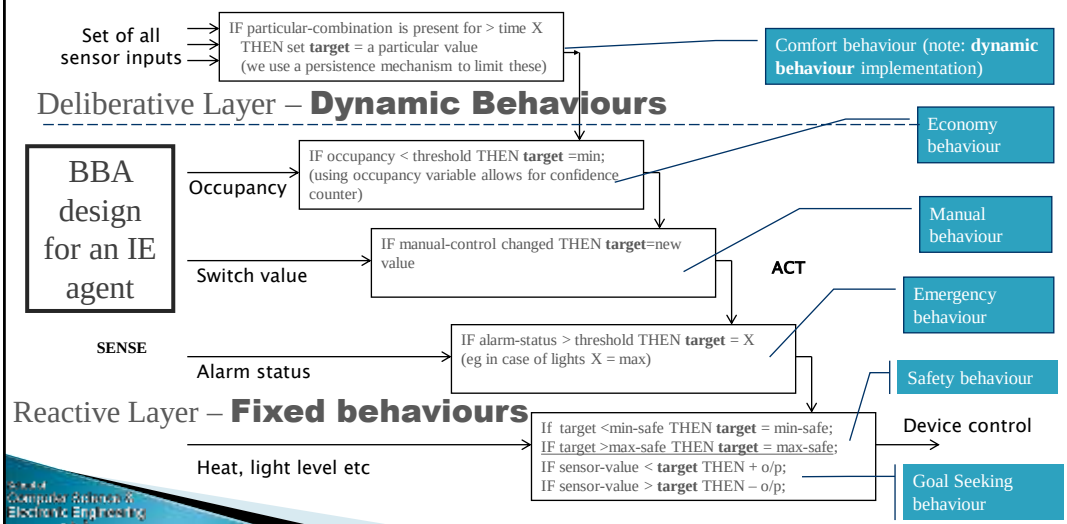
Recap: Training Package 1

- Introduced Intelligent Environments
- Explained embedded computer systems, pervasive/ubiquitous computing & the Internet-of-Things
- Learnt how to program a rule-based multi behaviour stand-alone embedded-agent (in the labs implemented learning one single rule/behaviour to react to a single threshold (eg temperature).

See, http://victor.callaghan.info/publications/misc/2014_KAU_Training-IE_June2014.pdf

Recap: Rule based behaviour agent

Learnt how to design this behaviour based embedded-agent



Multi Agents Rationale

- **Premise** – "*groups of agents/artefacts can achieve more than single isolated entities*" ?
  - **In nature** – groups of people, herds of animals, colonies of insects
  - **In Artificial Agents** – cooperation, collaboration and coordination mechanisms well documented but *pervasive computing* has slightly different needs to general AI (eg no negotiation on role, as that is fixed by embedding agent into artefact)
  - **In Internet-of-Things** – 'Virtual 'Appliances' constructed from coordinating groups of gadgets (eg incoming telephone call from person X resulting in raising light, muting TV and switching on video recorder)
  - **ScaleUp** – coordination takes place between whole environments (eg rooms, buildings, virtual and mixed realities etc)
- Ultimately coordination targeted at ad-hoc structures of vast numbers of agents

Some Coordination Examples

- Some intelligent environment problems that can't be solved (or efficiently managed) without co-operation. There are endless possibilities such as:
  - Meta Appliances – Applications – how users can creatively program collections of coordinating appliance or application functions to produce classic or novel functions (eg virtual TVs etc)
  - Safety Appliances – eg how cooker based agent might determine it should switch off (when you are in bed or left house!)
  - Smart Appliances – eg using other artefacts to help a given gadget make a better decision (eg smart air-conditioners can save up to 40% more energy)

– DIY – Try to think of more examples

Question 1 – what network?

Media & Performance

- Wireless (radio, infrared etc)
- Wired (cat-5 etc)
- Powerline (x10 etc)

IP or not IP?

- Good economies of scale in IP world.
- Could we use IP networks (depends how important real-time is)?
- IP 4 non-deterministic but IP 6 can offer real-time

Protocol & Performance

- **Data-Network** – Large bandwidth but bulky implementation and not necessary real-time (used for large files and media streaming, etc).
  - Performance measured in throughput (bandwidth).
- **Field-bus** – Low bandwidth but lightweight implementation and real-time (used for appliance status, commands etc)
  - performance measured in transactions per sec & response time

IE Network Standards

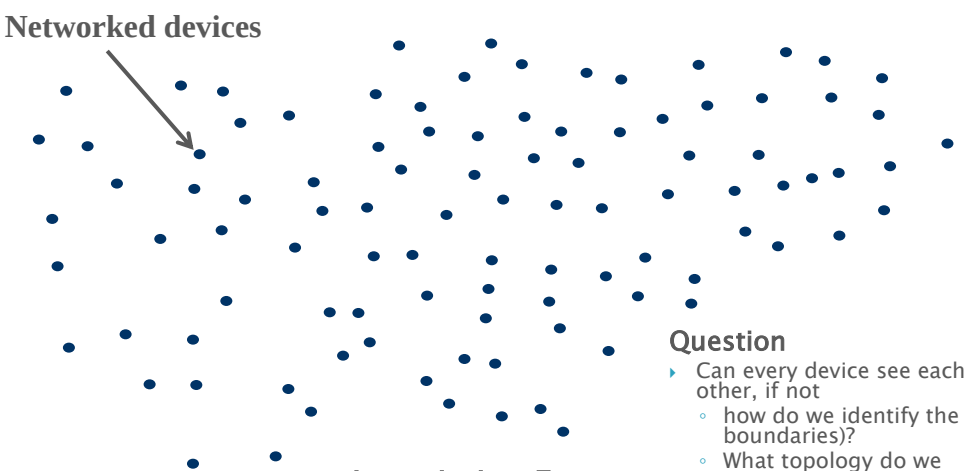
- **40+ standards. The leading ones being:**
  - **Lonworks** - Popular commercial standard. Supports power-line, wired, wireless. Focus of LonWorks is "Neuron" chip, which acts as a network node & micro-controller (plus a fuzzy-control like language)
  - **KNX** (formerly known as EIB) – originated as the European Intelligent Buildings standard and is perhaps the main modern contender to Lonwork with numerous manufactures offering devices (eg ABB, Jung, Siemens). It is a non-proprietary standard managed under the auspices of the EIBA
  - **BACnet** - An ASHRAE (American Society of Heating, Refrigerating & Air-Conditioning Engineers) standard modelled on Open Systems Interconnection (OSI) basic reference model. By way of example, Honeywell use both BACnet and LonWorks in their products.

IE Network Standards

- Continued ….

- **DALI** (Digital Addressable Lighting Interface) – International Standard (IEC 62386) for lighting control systems.
- **CEbus** – An EIA (Electronics Industry Association) standard. Covers devices that communicate through power line wires, low voltage twisted pairs, coax wires, infrared, RF, and fibre optics
- **Smart-House** – Developed by the NAHB (National Association of Homes Builders) for building into new houses.
- **EHS** – European Home Systems standard grew out of EU research projects. It is managed by the EHSA (which includes, British Gas, Deutsche Telekom, EDF, Electrolux, ENEL, Schneider Electric, Siemens, Trialog).
- **X10** – Oldest commonly available IB technology allowing limited control of common household control devices, through power line.

Question 2 – what topology?

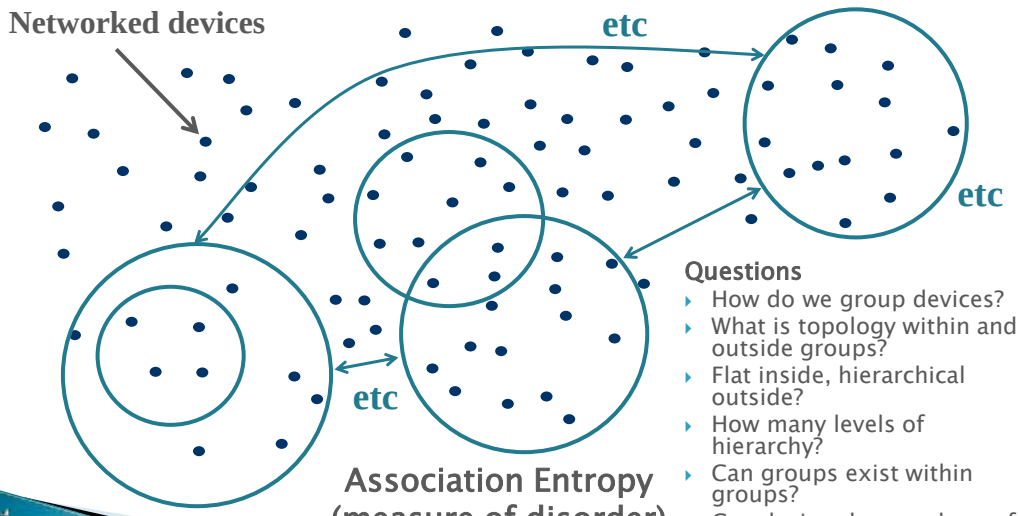


Association Entropy
(measure of disorder)

Question

- Can every device see each other, if not
  - how do we identify the boundaries)?
  - What topology do we impose?

Question 2 – what topology?



Questions

- How do we group devices?
- What is topology within and outside groups?
- Flat inside, hierarchical outside?
- How many levels of hierarchy?
- Can groups exist within groups?
- Can devices be members of more that one group?

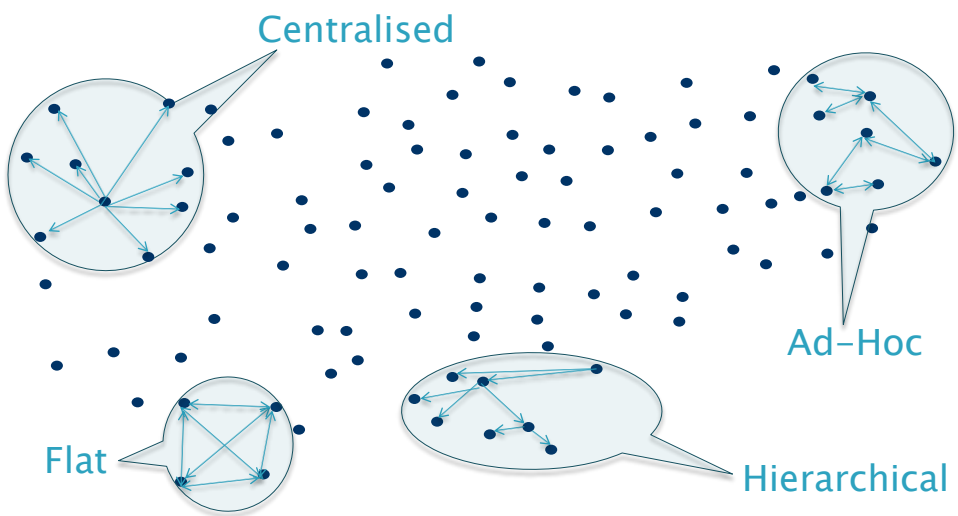
Association & Coordination

- A core principle in Intelligent Environments is that networked devices can coordinate actions to create meta-functions (sometimes called meta appliances/applications MAps or virtual appliances)
- How are devices associated together (what groupings or communities, what mechanism)?
  - Promiscuous (eg connected/disconnected algorithms)
  - Directed (eg ontology driven)
- How are devices programmed to produce the meta-functionality required by the user?
  - Autonomous Agent Learning (as in earlier notes)
  - End-User Programming

How Do We Group Devices?

- **Option 1** – "Hard-wire" appliances into groups that make some intuitive sense (eg topologies that mirror physical world):
  - Space (eg room) centric
  - Person centric
  - Technology (eg network) centric
  - Function centric (eg GadgetWorld, Virtual Appliance etc)
- **Option 2** – Get an agent to automatically form connections for a appliance to improve its performance (possible but complicated & part of research)
- **Option 3** – Use a user/type/ontology directed mechanism

Topology – some options



Topology: Centralisation Versus Distribution

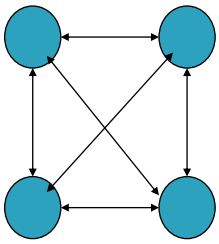
- **Case for Control Distribution (one extreme):**
  - **Performance –** better performance due to concurrent processing
  - **Scalable** –as processors and physical system expand in tandem and thus well matched to *dynamic (changeable) computing environments* (eg mobile and nomadic pervasive computing)
  - **Better fit** for systems with spatially dispersed control needs (eg ships, buildings, dispersed organisations)
  - **Less wiring & bandwidth demands** (ie I/O closer to processor)
  - **Reliability** – no single dependency, fewer connections, less susceptible to emi but more recent evidence has shown they are vulnerable to Zamudio's '**cyclic instabilities**'!

Topology: Centralisation Versus Distribution

- **Case for Control Centralisation (another extreme)**
  - **Simpler Architecture** (easier to understand & programme, not vulnerable to Zamudio instabilities)
  - **Cheaper** (less processors and programming)
  - **More efficient computer system management** (less anarchical)
  - **Facilitates Data mining** (ie better supports commercial needs, social research etc)

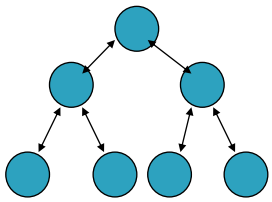
Question 2 – what topology?

- Flat Network
  - Co-operation between units of equal importance
  - The most complex approach but with significant performance advantages that are making it a target for future systems
  - "*The problem of flat structures is they are somewhat analogous to anarchies which can border on the chaotic, akin to an ill-functioning government*".
  - Structure is one solution, but what structure?



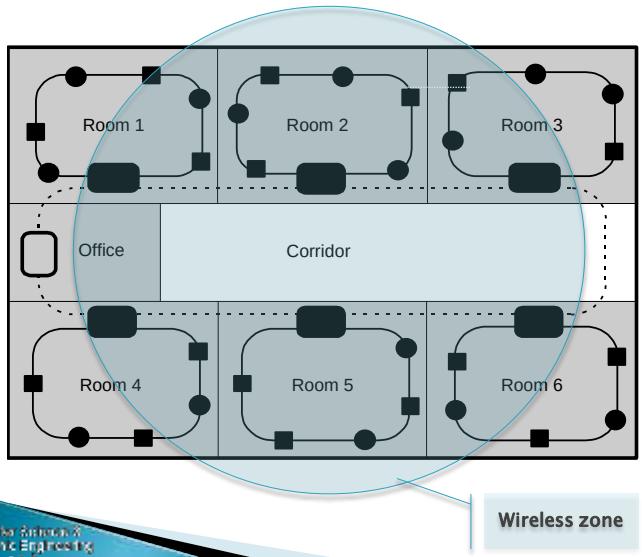
Question 2 – what topology?

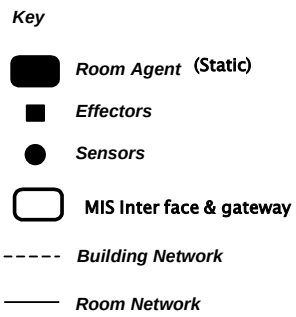
- Hierarchical Network
  - Co-operation between units of unequal (parent-child relationship)
  - The simplest approach and historically, the way most complex systems such as armies, governments, networks and hence IIEs have been implemented.



  - "*The problem of controlling a sensory-interactive system in similar in many respects to that of controlling any complex system such as an army, government, business or biological organism*" … "*the command & control structure for such systems is invariably a hierarchy wherein goals or tasks are selected at the highest level and decomposed into a sequence of sub-tasks*" (adapted from Rzeski p283). Hierarchies (both physical & logical/represent one possible solution.

Topology Options – Space Centric Model





Rooms are natural way of partitioning peoples behaviour & agents types (reflecting the architects intentions)

Essex University

29

Notes on Space Centric Model

- **Justification** – based on view that that physical and logical unit of an intelligent building is a single room (the justification being that that rooms are basic building blocks of larger buildings and communities … and are the unit of functional decomposition used by the architect).
- **Room Level** – room agent containing a number of interacting behaviours and connected to sensor/effectors via local network.
- **Building Level** –room agents connected via enterprise network (eg IP) and to other buildings via a gateway
- **Agent Level** – Person's behaviour stored on network (eg cloud, agents etc) and passed between agents as needed.

Essex University

30

Topology Options – Technology Centric

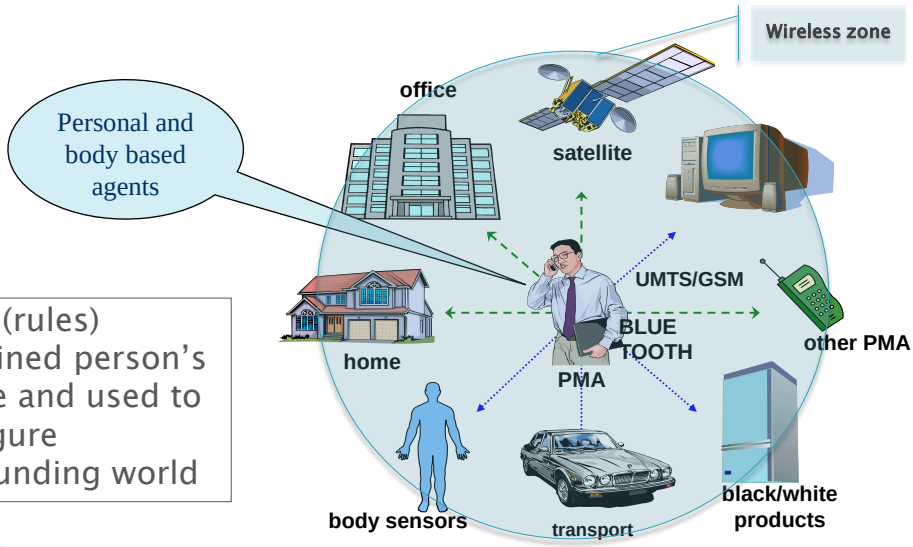
- In the workplace, the contenders for centralised agents include:
  - Network providers of gateways/routers
  - Telcos/ISP (gateways/routers) for SMEs
  - Utility companies

- In the home, the contenders for centralised agents include:
  - Telcos/ISP (gateways/routers)
  - Games boxes
  - Utility companies

Notes on Technology Centric Model

- Justification – Based on similar but conflicting views that:
  - Communication architecture best reflects computational and embedded-agent needs
  - Utility providers (eg smart meters) & or service providers (eg games, ISPs) are already in homes and therefor available
- Telcos have strong argument as communication device is present is present in all computational spaces (eg body, room, floor, building, vehicle etc)
- Assumes gateways and agents can be integrated on same platform.
- Can derive additional advantages such as data-mining.

Topology Options – Person Centric

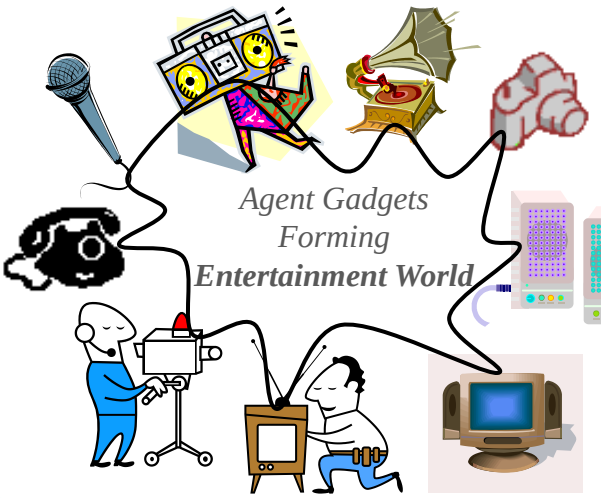


Maps (rules) contained person's phone and used to configure surrounding world

Notes on Person Centric Model

- **Justification** – based on view that that agents act for people, therefore should be with them.
- **Body Level** –gadgets connected together and to user via wireless network (depends on good wireless networking coverage, arbitration and security)
- **Room & building Level** – connected via wired networks. Behaviours (rule sets) stored and transported in phone (back up in cloud).

Topology Options – Function Centric



Virtual Appliances

Based on view appliances can be deconstructed to basic functions and reconstructed to produce meta functions or virtual appliances (eg virtual TV)

Notes on Function Centric Model

- **Justification** – based on view that that appliances are collections of functionality where each can be offered separately (decomposition) and recombined to create virtual appliance as needed (a better model for the internet age). A sort of crowd sourced innovation for appliances!
- **Room Level** – Groups of networked decomposed functions connected to form required appliances (MAps stored locally).
- **Wide Area Level** – MAps stored locally, in phones or the cloud which are mobile between locations and tradable.

End of Part 4

That's it for Part 4 !

Part 5

Overview

- Languages & communication for multi embedded-agents
- ASL (Agent Semiotic Language)
- Community Formation and co-ordination in embedded multi-agent systems
- Ontology and Taxonomies

These slides are available from:
- http://victor.callaghan.info/publications/misc/2014_KAU_Training-IE_June2014.pdf
- **http://victor.callaghan.info/publications/misc/2014_KAU_Training-IE_Dec2014.pdf**

Joining an Essex-KAU Experiment (1 hour)

The evaluation is divided in 3 parts:

- **First 10 minutes** – Preliminary online survey to determine your familiarity and experience with Mixed Reality and Smart Objects.
- **Second 35 minutes** – Simple programming in collaboration with other student located at Essex using a virtual world & BuzzBox.
- **Final 15 minutes** – Short questionnaire about your experience.

Total duration of the activity: 60 minutes

If you are interested in taking part in this experiment have a look at this video:

http://youtu.be/rIjqUBvLQoo

and contact Anasol Pena-Rios at **acpena@essex.ac.uk**

Essex University

Inter-Agent Communication

- Pervasive Devices need to communicate with each other to achieve coordination
- Two popular approaches:
  - Semiotic Based Communication (eg sign/state based mechanism)
  - Language Based Communication (eg semantic/grammar based mechanism)
- Some Issues
  - No negotiation regarding which agent does some jobs (as embedded-agent physically tied to machine)
  - Instability (distributed schemes, without management can misbehave)
  - Limited computational resources (need lightweight mechanisms)

Essex University

40

Agent Communication Languages

- Labrou defines **agent communication languages (ACLs)** as *"a collection of message types, each with a reserved meaning"*
  - ACLs are not concerned with the physical exchange, over networks,
  - ACLs are concerned about the content (eg promise of future action)"
- 20+ ACLs (and other agent communication frameworks) !
  - All about message passing.
- **KQML** (Knowledge Query and Manipulation Language)
  - High-level agent communication language
  - Based on speech act theory (messages are actions or communicative acts, as they are intended to perform some actions by virtue of being sent)
  - Twelve reserved performatives, (in seven categories)
- **FIPA** (Foundation for Intelligent Physical Agents)
  - A high-level ACL,
  - Based on speech act theory.

KQML & FIPA differ in details of semantics (Syntax is almost identical)

ACLs & Embedded-Agents

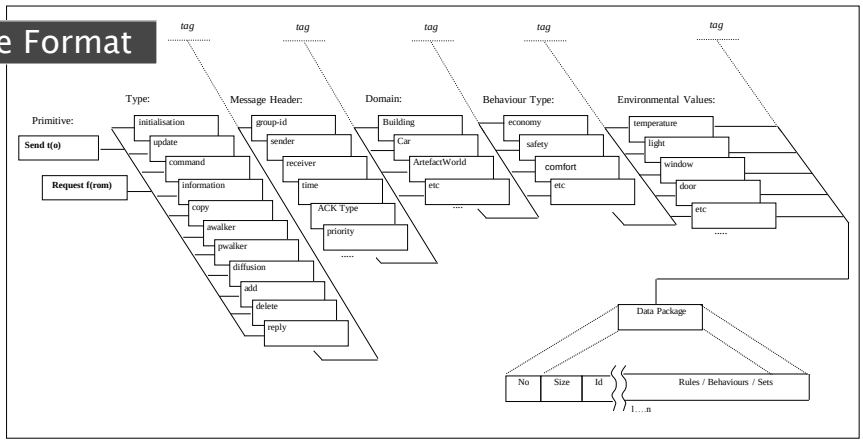
- Conventional Agent ACLs (eg KQML, PIPA etc) not useful for embedded-agents because:
  - computational footprints too large (10's of MB !)
  - Too many and irrelevant primitives
  - Overly complex structure (highly developed)
  - Don't easily provide feedback on whether message received, understood or executed
  - No convenient mechanisms for detecting or registering new agents on the network (often use Java frameworks such as Jackal and Jafmas)

DIBAL: an ACL for embedded agents

- **DIBAL** (Distributed Intelligent Building Agent Language) for small embedded-agents
- Uses **hierarchical tagged field format** allowing more efficient and flexible message size packaging.
  - Small fixed size header containing
    - two primitives (send, request)
    - 20 type flags (11 currently defined)
  - Variable size and hierarchical tagged fields for:
    - Domain
    - Behaviour
    - Sensor-parameter ID
    - data

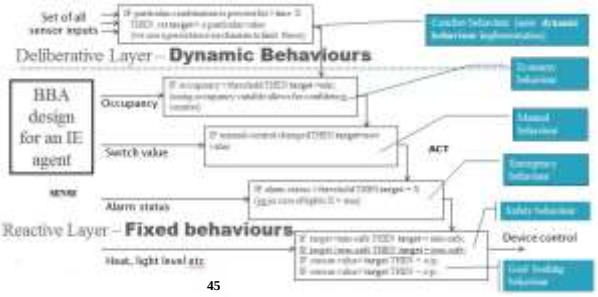
DIBAL Message Structure

Message Format

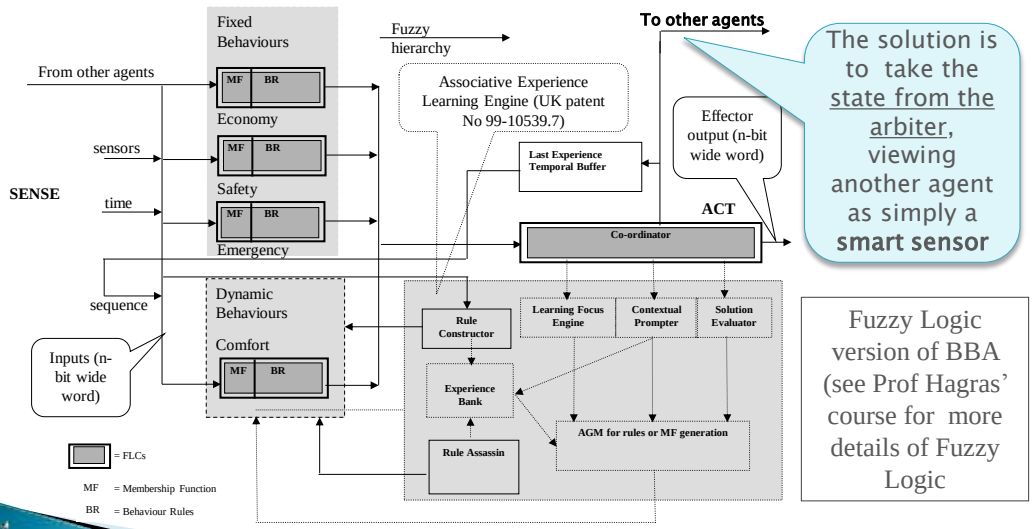


Language and Embedded-Agents

- But ….
- The embedded-agents we have created have little/no language processing ability!
- The we are trying to build them from minimal hardware (slow processors, small memories) while demanding the work in real-time.
- Is there a solution?



A solution: state as a communication language?



The solution is to take the state from the arbiter, viewing another agent as simply a **smart sensor**

Fuzzy Logic version of BBA (see Prof Hagras' course for more details of Fuzzy Logic

ASL Based Communication

- **ASL** (Agent Semiotic Language) – name for sign/state messaging
  - Semiotics – the study of signs and symbols in various fields
  - Agent presents *status* (sign) based on a compressed view of sub-sensory states making it like a super-sensor (ie no message passing)
  - Agent processes data from other agent as though it was a sensor; a **super-sensor** or smart-sensor (learns to weight value of data for any particular stimuli-action set)
  - Compressed view of Agent state available in BBA from the arbiter state

- **Advantages**:
  - Simple (little overheads,
  - Compact (compresses data)
  - Generic & scale independent

- **Disadvantages**:
  - No higher order semantics
  - Non interactive (listening only)
  - Implies large i/p vectors

Some Communication Mechanisms

- Some Embedded-Agent Co-ordination Mechanisms:
  - **Status exchange** (ie compressed room data, pseudo reasoning)
  - **Weighted status exchange** (ie attaching a value to data)

ASL mechanism (non interactive – no semantics)

ACL mechanisms (interactive communication – with some semantics)

  - **Walkers** (eg passive and active virtual occupants)
  - **Behaviour migration** (eg following specific occupants, initialising agents)
  - **Knowledge diffusion** (eg Chinese whispers; passing non-specific rules that can lead to better rules cf learning by forgetting)

Some ASL Mechanisms (uni-directional)

- **Status exchange**
  - Only "compressed" set of information needs to be broadcast (eg "room X, occupied by Y, probability Z)
  - Probability explicitly used in pseudo reasoning mechanisms such as confidence counters (eg Mataric, Sharples)
- **Weighted Status**
  - Value of differing sensors or devices may differ, for given decision
  - Could use fuzzy-logic to model this
  - Simpler approach is to use "**Confidence Counters**" (see Mataric)
    - Pseudo reasoning (eg reasoning on whether to turn on smart heater, say)
    - Gather information from various sources (*including remote appliance/agents)* dynamically assigning a weight (multiplier; high confidence high values, low confidence low values)
    - Incrementing / decrementing counter (confidence counter)
    - Decisions made by monitoring whether threshold exceeded
    - Compatible with compact agents mechanisms (minimalist design)

Some ACL Mechanisms (bi-directional)

- **Behaviour migration**
  - Behaviours are effectively self-programmed control algorithms (eg personalised sets of rules)
  - Useful to automatically move with occupant rather than reprogramming
- **Knowledge diffusion**
  - Request initialisation behaviours (rules) from artefacts in locality corresponding nearest match for task/person; use, adapt and make available these rules for similar requests (ie each "knowledge process hop" potentially modifies data or methods)
  - As knowledge (rules) defuse through network they are adapted and evolved (a type of learning with some GA, evolving, like qualities!).
  - Akin to the transfer of tools via knowledge movement in early civilisation

Some Other Mechanisms

- **Walkers** – mobile processes that fake sensors to simulating occupants or other actions
  - **Type 1** : Active Walker (ASL) – data sent to an agent to simulate an occupant (eg cause preparatory "go to bed" mode by sending virtual person to room)
  - **Type 2**: Passive Walker (ACL) – is a request sent to agent to get status report (eg monitor alarms, debugging etc)

Configuring embedded multi-agent systems

- Two main approaches
- **Manual**
  - Example, End user programming (see Part 1 notes)
  - Puts user in control (good for privacy, trust etc)
  - Releases user creativity (enables them to have a more active role)
  - Potential to disrupt current market
  - Need intuitive methods (non-technical users can't write programs)
- **Automatic Approaches**
  - Directed Association
  - Random Association

Manual Configuration Methods (recap)

- From Part-1 we discussed configuration based on:
  - The use of a graphical editor to manually interact with a list of networked devices (eg based on middleware discovery or ontology)
  - Physically touch (or otherwise indicate eg switching on/off) devices to be linked (see related topic of "Tangible Computing" by, Ishii at MIT, which seeks to give data and soft components physical form)

Automatic Promiscuous Association

works like GA (very slow)

Type 1 – Initially Fully Disconnected

- make a random (or type/ontology directed) connection (novelty!)
- learn new sensory-effector instances
- compare to previous performance (based on user over-rides)
- Kill/Keep

Type 2 – Initially Fully Connected

- similar to the disconnected but with the first step being a random (or informed disconnection (eg from low weighting rules, or ontology)
- Parsing during idle periods is possible

Directed Association (eg ontology driven)

- **Directed Association** – based on making informed connection decision (which may or may not be followed by informed disconnections)

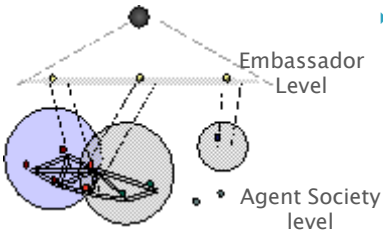
Contrast to '**Promiscuous Association**' which makes uninformed connection decisions (eg random) followed by informed disconnection

- Informed decision based on "reasoning" about devices capabilities based on a suitable description
- The description (and the reasoning engine) is the key to this approach.
- Based on ontology (ways of being) from semantic web

Directed Association Mechanisms

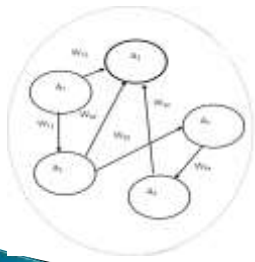
- Two approaches:
  - **Device CV (ontology)** – select (prioritise) connections to try based on reasoning about information provided via a device CV that focuses on their capabilities and possibilities rather than low device specifications
    - The description (and the reasoning engine) is the key to this approach.
    - Based on ontology ideas found semantic web
  - **Friend-0f-friend** – rather than selecting potential associations for investigation in a random way; select '*connections of connected devices*' (prioritising connections that occur in more that one connected device)
- Can be fully-automated (no manual intervention) or semi-automated (prompts user)

Example of Association Algorithm



Hierarchical clustering model

- **Automatic Association works by:**
  - learning a rule that specifies how much the weight of an input should be increased or decreased in proportion to the product of its association with the activation of the desired output
  - Hebbian learning is often summarised as "The theory is often summarized as "Cells that fire together, wire together"

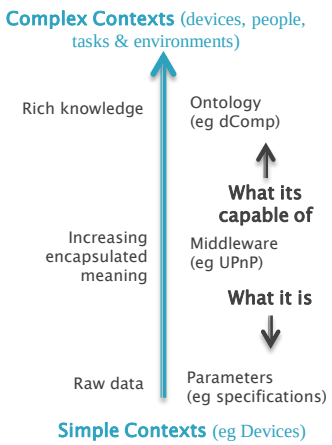


Hebbian learning association weights adjusted via:

- ***Step 0***: *Initialize all weights: wAi = 0 (i=1 to n)*
- ***Step 1***: *Initialize the learning rate a = 0.1*
  wij(q) = wij (q-1) + aai(q) pij(q)
- ***Step 2***: *For each pre-associative and post-associative agent pair (i,j=1 to n), a : p, do following:*
- ***Step 3***: *On a regular basis apply threshold, t=0.1, to determine redundant associations; if redundant associations > 0 then do Step 4, otherwise go to Step 2.*
- ***Step 4***: *Adjust every agent's rule base according to the importance of the associations*

H. Duman et-al, *"A Multi-Society based Intelligent Association Discovery and Selection for Ambient Intelligence Environments"*. ACM Trans on Autonomous & Adaptive Systems, Vol.5, No.2 pp.40-74 May 2010.

Types of descriptions in IEs



- Traditional AI components (eg reasoners) require descriptions of devices, people, tasks environments to operate.
- Manifestations vary from technical specifications, through middleware descriptions to ontologies
- To date, only ontology is suitable for describing all facets of an intelligent environment.

Ontologies: an intelligent environments view

- Ontology
  - In philosophy it refers to the study of the nature of 'being', or 'reality', but more practically, it **involves describing categories and relationships**. In technology it's a way of defining a body of knowledge about an area based on terms and relationships.
  - Knowledge, infers the inclusion of meaning, such as capabilities of a device, rather than just what the device is.
- Why use an ontology?
  - To automate (or semi-automate) pervasive computing need to encapsulate knowledge about devices and their capabilities.
    - Built in semantics makes reasoning function possible (needed to automate systems)
  - Also, lack of a standard way of describing such knowledge is seen as an obstacle to interoperability and market success.
- Why Don't use an ontology
  - They can be huge and computationally intensive.

Ontology Examples

- Ontology Web Language (OWL)
  - Developed by the Semantic Web Initiatives & sponsored by the World Wide Web Consortium (W3C)
  - Provides a framework for web asset management
  - Based on Resource Description Framework (RDF) – a syntax used in the Semantic Web for representing data which conforms to XML rules
  - Supported by Semantic Web tools such as HP Jena and inference engines such as RACER, F-OWL, ConstructTM
- dComp ontology
  - Developed for intelligent Environments and supports device decomposition and communities (Virtual Appliances).
  - 10 classes – 3 adopted the current standard ontology SOUPA (a context-awareness model developed by UbiComp)
  - midComp ontology; a lighter version of dComp

What does an ontology look like?

dComp Classes

DCOMPDevice Class
DCOMPHardware Class
DCOMPService Class
DCOMPperson Class
Rule Class
Preference Class
DCOMPCommunity Class
Action Class
Time Class
Policy Class

dComp device description

```
<device:AudioDevice rdf:ID="TestDevice12">
<device:hasDeviceInfo>
 <device:DeviceInfo>
<device:friendlyName>TestDevice12</device:friendlyName>
<device:DeviceUUID>0</device:DeviceUUID>
<device:DeviceType>urn:schemas-upnp-org:TestDevice12:1</device:DeviceType>
<device:DeviceModelURL>http://TestDevice12URL/</device:DeviceModelURL>
<device:DeviceModelNumber rdf:datatype="&xsd;double">0.0</device:DeviceModelNumber>
 </device:DeviceInfo>
 </device:hasDeviceInfo>
 <hw:componentOf>
  <hw:RAM rdf:about="#JCTestMemory2"/>
 </hw:componentOf>
 <serv:hasDCOMPService>
 <!-- can have more than 1 service -->
  <serv:AudioService rdf:about="#JCAudioService01"/>
</serv:hasDCOMPService>
<!-- 2nd service -->
<serv:hasDCOMPService>
 <serv:AudioService rdf:about="#JCAudioService02"/>
 </serv:hasDCOMPService>
<!-- 3rd service -->
<serv:hasDCOMPService>
 <serv:AudioService rdf:about="#JCAudioService03"/>
</serv:hasDCOMPService>
</device:AudioDevice>
```

XML device description

```
<resource>
<hasa>
<uuid>0</uuid>
<description>
<name />
<serial_number />
<OS />
<RAM>128M</RAM>
</description>
<service required="true">
<in required="false">
<protocol>local:jack:3.5mm</protocol>
<format>stream:audio:analog:line</format>
<parameters>
<range name="channels" min="1" max="2"/>
</parameters>
</in><in required="false">
<protocol>local:optical:tos</protocol>
<format>stream:audio:digital:spdiff</format>
<parameters>
<range name="channels" min="1" max="6"/>
</parameters>
</in><in required="false">
<protocol>network:shoutcast</protocol>
<format>stream:audio:digital:icecast</format>
</in></service>
<service required="true"><in>
<protocol>network:TCP</protocol>
<format>control:volume</format>
<parameters>
<range name="volume" min="0" max="100" required="false"/>
</parameters>
<returns>
<range name="volume" min="0" max="100"/>
</returns>
</in></service>
</hasa>
</resource>
```


Multi Environment Tests

- Some tests for building applications of multi intelligent environments include the:
  - **Travelling Occupant** – how the agents would act intelligently to support regular physically dispersed multi agent activity, enabling agents to be pre-emptive across dispersed spaces (eg person travelling from office to home, a night-watchman making his rounds)
  - **Erratic Occupant** – how agent deals with non-cyclic behaviour such as avoiding erratic agent behaviour when office temporally vacated (eg photo copying, at meeting etc).
  - **Vacated Occupant** – how an agent separates periods of learning diverse activity (eg vacation/ non-vacation, illness etc)

Ambient Intelligence Taxonomy

Focus of the AI

- Type 1 – control (optimising actuator management)
- Type 2 – association (optimising connections or associations)

Learning Mechanism

- Mode 1 – "explicit" (manual)
- Mode 2 – "implicit" (automated)

Example: 11 would be a programming-by-demonstration agent

Summary

- Collaboration between embedded-agents can be obtained very simply by:
  - Making agent's behaviour arbitration output available as input to other agents
  - Adding a dynamic association method (random or directed) that allows agent to explore which collaborations beneficial.
- Capitalises on reasoning (rule-set interplay) and learning (rule formation) that already exists in behaviour based embedded-agents

End of Part 5

That's it for Part 5 !

Part 6

Overview

- Debugging multi-agents
- Human Related Issues
- Product Innovation (agent & Internet-of-Things)

These slides are available from:
- http://victor.callaghan.info/publications/misc/2014_KAU_Training-IE_June2014.pdf
- **http://victor.callaghan.info/publications/misc/2014_KAU_Training-IE_Dec2014.pdf**

Joining an Essex-KAU Experiment (1 hour)

The evaluation is divided in 3 parts:

- **First 10 minutes** – Preliminary online survey to determine your familiarity and experience with Mixed Reality and Smart Objects.
- **Second 35 minutes** – Simple programming in collaboration with other student located at Essex using a virtual world & BuzzBox.
- **Final 15 minutes** – Short questionnaire about your experience.

Total duration of the activity: 60 minutes

If you are interested in taking part in this experiment have a look at this video:

http://youtu.be/rIjqUBvLQoo

and contact Anasol Pena-Rios at acpena@essex.ac.uk

Essex University

Debugging Distributed Embedded-Agents

- Developing & debugging situated distributed embedded-agents offers challenges:
  - Complex large scale system
    - Lots of agents and sensors
    - Widely distributed asynchronous & concurrent real-time events
    - Size of system (eg building etc) cannot move to lab (& situated / embodied aspects plus human behaviour crucial to operation)
  - Significance lies in distributed "event & activity" correlation
    - System (mix of people and machines) is running non-deterministic "intelligent" processes (how do we know what to expect?)
    - Actions may result from other agents (but which?)
    - The apparently incorrect behaviour or action may be correct, but odd, characteristic of learning (but how do we know; c-dilemma?)

Brooks Axiom "*the world is is its own best model*" … means development largely in-situ

Essex University

68

Debugging Distributed Embedded-Agents

- Conventional development and debugging methods for embedded computers include:
  - Trial & Error (Compile, download & observe … repeat)
  - Emulation Systems (like simulation but using modified versions of the real product)
  - Xdev
- The "observe" part of debugging is in need of better tools eg correlated graphical displays of multi-agent activity, use of AI based reasoning to analyse activity etc
  - more research is needed as bad tools often result in struggling for longer than necessary on faults).

Debugging Distributed Embedded-Agents

- Some solutions being researched for MAS include:
  - Synchronised Xdev (cf multi-target VxWorks)
  - Instrumented code netf seeds (cf printf statements)
  - Novel monitors (eg graphically-enriched, Java assisted, AI)

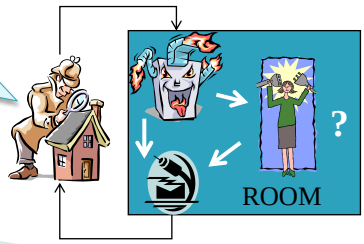
See '*Dependable computing needs pervasive debugging*' Tim Harris Proceedings of the 2002 ACM SIGOPS European Workshop. Can be found on http://www.cl.cam.ac.uk/research/srg/netos/pdb/

- Investigating debugging embedded multi-agent systems could make an interesting research!

Developing Distributed Embedded-Agents

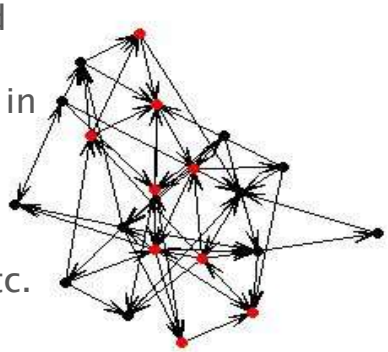
- **Development** – Developing agents in simulated environment could reduce development cycle by accelerating time.
- **Simulation** – difficult (s-paradox: non-deterministic humans in the simulation loop!)
  - Occupant (human) is critical part of model (simulation) but occupant reaction is non-deterministic – a simulation of a person would itself be equivalent to making us?
  - Control Loop – action of agent alters world, altered world provokes occupant reaction that may change world which in turn may cause reaction from agent etc (cf "chicken & egg")

*How do we model non-deterministic "**human transformation**"? See earlier slide for "human cross-coupling" and "agent interference"*



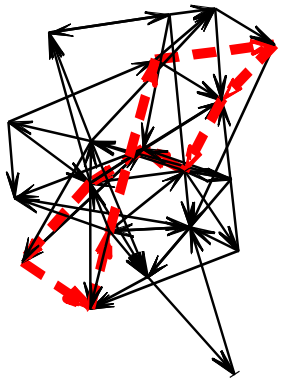
Debugging – Cyclic Instability 1

- Pervasive computing devices and embedded-agents have inter-dependent rule sets can interact in unexpected and unwanted ways such as lights cycling on/off
- Temporal delays due to network latency, speeds of processing, etc. could result in some device receiving old information, or being involved in feedback loops contributing to unstable behaviour.



Victor Zamudio and Victor Callaghan , "*Understanding and Avoiding Interaction Based Instability in Pervasive Computing Environments*", International Journal of Pervasive Computing and Communications, Vol. 5 Issue: 2, pp.163 – 186, 2009

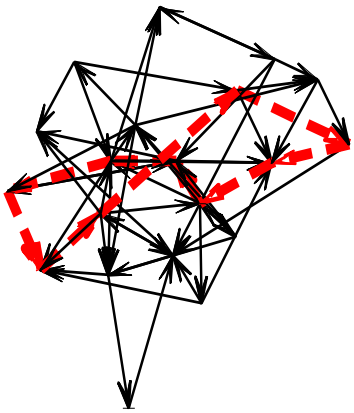
Debugging – Cyclic Instability 2



- Doesn't occur in centralized systems – only distributed systems (which is a new and growing trend in pervasive computing and ambient intelligence).
- Complex system theory shows it is not possible to predict theoretically whether an arbitrary set of rules will suffer from instability.
- Variables include:
  - number of agents,
  - Topology
  - Perturbations
  - initial states
  - Coordination rules

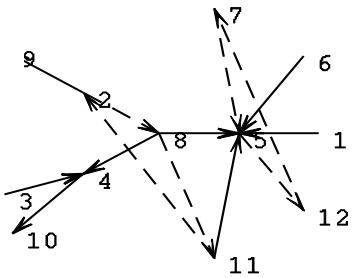
Debugging – Cyclic Instability 3

- High Level Algorithm (based on detection of loops in the Interaction Network)
- For each loop
  1. Find the node member of the loop which minimizes a "functionality function" (ie the one that has fewer descendants). In terms of the services provided to the user, this is very important, as it minimize user dissatisfaction.
  2. Lock this node
- Step (2) includes learning from the user their preferred "locking preferences"



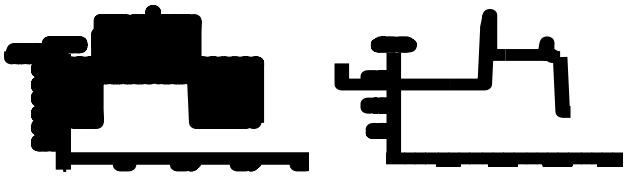
Debugging – Cyclic Instability 4
(example of 2 non-coupled cycles)

Number of nodes		12
Topology		{{1,5},{2,8},{2,9},{3,4}, {4,10},{5,12},{6,5},{7,5}, {8,4},{8,5},{8,11},{11,2}, {11,5},{12,7}}
Number of Cycles		2
Cycles		{11,2,8,11}, {7,5,12,7}
Coupled		No
Rules of Interaction		{1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0}
Locking vector		{1,0,1,1,1,1,0,1,1,1,1,1}
Oscillations	*Before*	Yes
	After	No



node-functionality lists:
{{11, 3}, {2, 1}, {8, 5}} and
{{7, 0}, {5, 0}, {12, 0}}

2 and 7 are locked
locking vector: {1,0,1,1,1,1,0,1,1,1,1,1}



Human Aspects

Human-Agent Interface

- Agent's interface can be implicit (invisible sensors) or explicit (manual control, speech, gesture etc).

Human-Agent Modes

- Pre-emptive Assistance
  - Agent transparently controls buildings services (eg heat) without user intervention
- Assistance-on-Demand
  - An occupant calls for assistance by pressing button on pendent/watch. Agent offers successive options on each button press (eg 3) before giving up and going into learning mode only.
- Alert-on-Exception (abnormality detection)
  - An occupant is alerted to his abnormal behaviour (eg as leaves house with appliances left on)

Human Aspects

Autonomy & Privacy

- Our homes are some of the most private spaces in our lives, do we want
  - Autonomous networked agents recording our behaviours
  - Communicating our behaviors elsewhere
- Privacy of Individual threatened by:
  - Network access by third parties
  - Agents learning private behaviors
  - Lack of transparency and explicitly control over system (ie what, when and for whom is personal information being gathered)
- End-user programming and adjustable autonomy empower the user

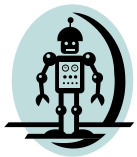
See: Callaghan V, Clarke G, Chin J "*Some Socio-Technical Aspects Of Intelligent Buildings and Pervasive Computing Research*", Intelligent Buildings International Journal, Vol 1 No 1, 2008

Human Aspects – Fears of Too Much Intelligence?

- The arguments for and against AI?
  - Sam Williams' book "**Arguing A.I.**" presents both sides of AI debate describing Ray Kurzweil's view that machines capable of surpassing human intelligence are inherently good and inevitable, versus Bill Joy's argument that has more sinister implications, such as AI moving beyond the control of human beings. It reports McCarthy's view that faster machines alone will not usher in human-level intelligence, while Lanier is reported as arguing AI has led to poorly designed user interfaces, among other things. Other figures profiled in the book include Alan Turing, David Hilbert, Hubert Dreyfus, and Marvin Minsky, and science fiction such as "2001: A Space Odyssey" is also credited as an influence. The book is aimed at providing non-expert readers with a road map for this debate.
  - Ray Kurzweil's book, "**The Singularity Is Near**" is also worth reading if you are interested in the Singularity debate.

Human Aspects

- Asimov's Rules for Robots are:



1. Protect Humans,
2. Obey Humans,
3. Protect Itself

Clark R "*Asimov's Laws of Robotics*", IEEE Computer Dec 1993, p53)

The films, "Demon Seed", "The Tower" & "2001A Space Odyssey" portrayed intelligent environments as dangerous

- SX IE Law for IBs are:
  1. Protect the habitat (and as a consequence, the occupants)
  2. Obey authorised stakeholders (commonly building occupants – obey humans)
  3. Maximise comfort for individual occupants.
  4. Economise energy

What laws might be devised for general pervasive computing?

Rapid Prototyping & Internet-of-Things

Some Buzz Board modules

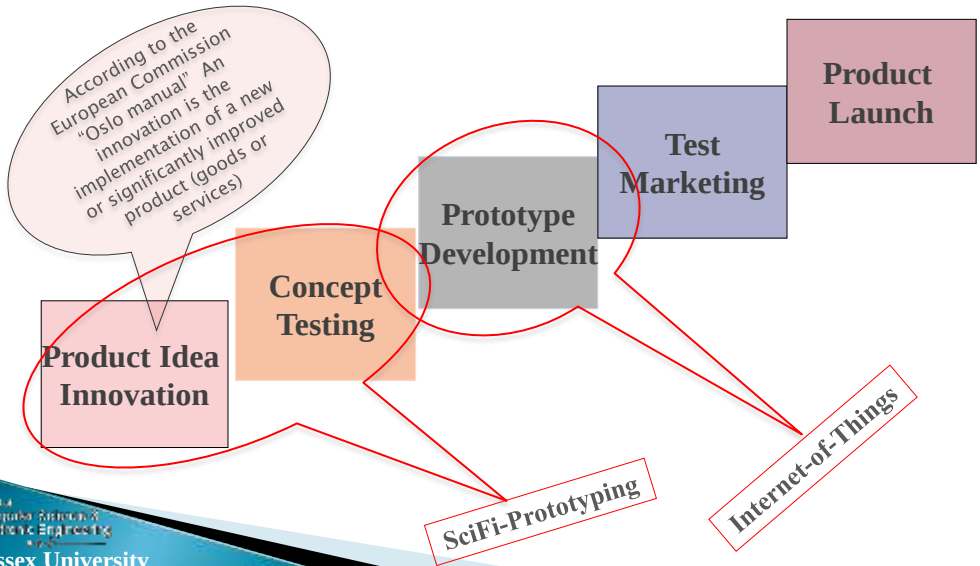
Examples: robot & Internet Radio

"Teaching Next Generation Computing Skills; The Challenge of Embedded Computing", Intelligent Campus 2011 (iC'11)

Buzz Boards

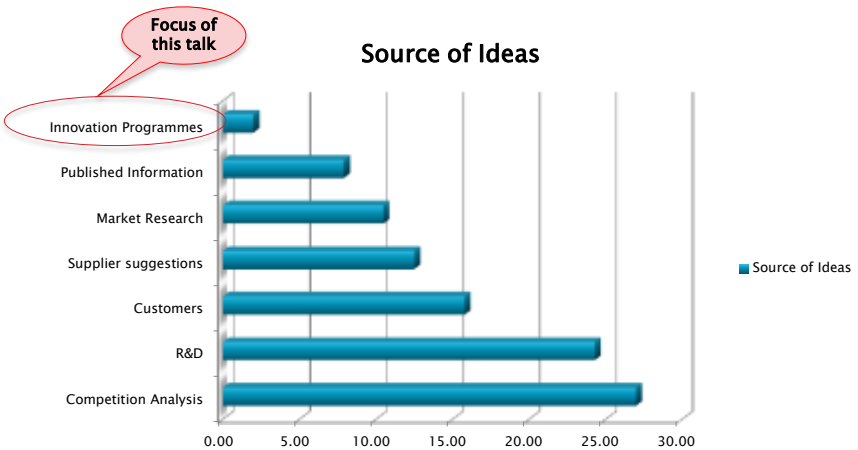
- Internet-of-Things networked modules can be assembled like multi-agents.
- Works by plugging some of the 30+ modules together to create, not only the KAU BuzzBox, but almost anything such as robots, radios, cameras etc
- Used for rapid protoyping for product innovation.

Product Development Process



Essex University

Where Do New Product Ideas Come From?



Essex University

The ingredients of a creative mind

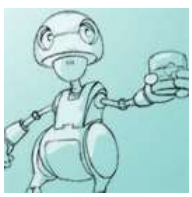
Joy of discovery
Sense of challenge
Curiosity
Imagination
Wonder
Playfulness
Questioning
Inquisitiveness
Willingness to take risks
Desire to interrelate
Desire to learn
Tolerance of ambiguity
Positive discontent
Desire to see things differently
Willingness to experiment

Drawing by Paul Rumsey
(www.paulrumsey.co.uk/)

Essex University

Science Fiction Prototyping

- Core methodology is the use of science fiction to motivate and direct research into business & science via science fiction prototypes (*imaginative stories about business or products in the future*)
- Prototypes grounded in existing research, and written for explicit purpose of acting as prototypes for people to explore a wide variety of futures.
- 'Prototypes' can be created by scientists and business professionals to stretch their work or by, for example, writers, school children and members of the public to influence the work of researchers.
- Outcomes of SFPs are used to create new products or businesses.



www.creative-science.org

Essex University

An Example–Exploring AI & The Future of Learning

From "Creative-Science 2010"

iPods were effectively small cocoons; something like a comfortable armchair enclosed within a sound-proof egg-like structure packed with sophisticated but largely invisible technology that included immersive mixed reality and sophisticated AI. When participating in a movie (the industry had long dropped the word "watching" which describing these new immersive movies) the immersive reality technology aimed to make the participant feel as though they were truly part of a fictional physical world.

Intel Creative Science Foundation – Promoting Technology Innovation through Science Fiction Prototyping

Additive Technology ePod-4

In this increasingly competitive world, where knowledge determines success, your child deserves the very best education available and that is Addictive Technology's **ePod-4**

Pioneering research by Benjamin S. Bloom in the 1980s (and supported by all work since) proved that students who receive one-on-one tuition learn at least an order of magnitude better than grouped students. If you want to give your child the best one-to-one education in the world, give them an Addictive Technology's **ePod-4**

Education:

- Super-Intelligent Artificial Teachers
- Personalised one-to-one tuition (the gold standard)
- Teacher's avatar has visualisation powers that don't exist in physical space
- Available 24 hours a day, 365 days a year
- Learning environment (avatar, surroundings, lessons) can be tailored for each student
- Unwavering attention and happy disposition
- Compelling content combined with contextual delivery
- Teachers available in different cultures, ages, sexes and form

Technology

- Free-Will 3 © - Quantum processor (upgradable)
- My-Mind 1.2 © - Evolving Persona Engine (customizable)
- Flame 5 © - EmotionWare
- Get Real 8.2 © - Mixed Reality Cocoon
- Real-Touch © iSkin & Haptics
- Ghost 4.1 © – 3D Imaging & Audio
- SentiNet © - Knowledge Engine

Addictive Technology, Zizhu Science Park, No. 880 Zi Xing Road, Minhang, Shanghai 200241, China

[Callaghan V, (2010). *Tales From a Pod*. In Creative-Science 2010 (CS'10). Kuala Lumpur, Malaysia: IOS Press, pp. 1-10.

Notes on SFP Example

- Called "Tales from a Pod" & presented at CS'10.
- Took a speculative look at how AI and mixed reality might change the education business in 2050.
- It imagined a future time when
  - Interactive computer games merged with cinema to provide "*immersive movies*" (audience were no longer passive observers – unless they wished to be), offering highly personalised experiences
  - Used high-tech-environments called '*education pods*', called ePods.
  - The *technological singularity* had been reached, meaning machine intelligence was equal or surpassed that of people.
  - The technology used to create an "intelligent teacher avatar" that "lives" in virtual reality pod.
- Its lots of fun inventing the future – try it!

http://dces.essex.ac.uk/Research/iieg/papers/TalesFromAPod(Paper).pdf

Final Summary!

- **Summary** of main points covered
- Part 1covered design of:
  - Single Intelligent Environments
  - Single behaviour based agents
  - **Key learning outcome** is rule based behaviour system can create computationally compact but powerful AI
- Part 2 covered:
  - The principles of Multi-agent architectures
  - The design of connected multi-agents
  - **Key learning outcome** is collaboration can be implemented very simply by offering arbitrator output to other agents.

End of Part 6

That's it!