Universidad de Deusto
Deustuko Unibertsitatea

mobility research lab

Sistemas Distribuidos Seguros y Ubicuos

Dr. Diego Lz. de Ipiña Gz. de Artaza
http://paginaspesonales.deusto.es/dipina
http://www.morelab.deusto.es
http://www.ctmd.deusto.es

4. Computación Ubicua
   - Concepto y áreas relacionadas: sentient computing, interfaces inteligentes, Ambient Intelligence
   - Sistemas sensoriales: cómo proporcionar capacidades sensoriales a dispositivos computacionales
     - Location-Aware Computing
   - Middleware para Computación Ubícua
     - Descubrimiento e interacción con servicios externos
     - Ejemplos de aplicaciones de computación ubicua
   - Escenarios de aplicación: Hogar, laboratorios, Intelligent Transport Systems
   - Interacción natural con el entorno: NFC
5. Middleware semántico para Sistemas Ubicuos
   - Web semántica: qué es y para qué sirve
   - Servicios Web y Web Services Extensions (WS-*)
   - Servicios Web Semánticos (OWL-S)
   - Aplicabilidad de la Web Semántica y los Servicios Web a los Sistemas Ubicuos
6. Líneas de investigación abiertas
   - Tendencias y problemas a resolver en sistemas distribuidos
   - Áreas a resolver para hacer realidad la visión de computación ubicua

Universidad de Deusto
Deustuko Unibertsitatea

mobility research lab

Tema 4 – Ubiquitous Computing

Dr. Diego Lz. de Ipiña Gz. de Artaza
http://paginaspesonales.deusto.es/dipina
http://www.morelab.deusto.es
http://www.ctmd.deusto.es

Contents

Universidad de Deusto
Deustuko Unibertsitatea

mobility research lab

4.1 – Conceptos

Dr. Diego Lz. de Ipiña Gz. de Artaza
http://paginaspesonales.deusto.es/dipina
http://www.morelab.deusto.es
http://www.ctmd.deusto.es

Ubiquitous Computing

Universidad de Deusto
Deustuko Unibertsitatea

- Concepto definido por Mark Weiser en 1991:
  - M. Weiser “The computer for the 21st century”, Scientific American, pp. 94-100, Septiembre 1991
- Creación de entornos saturados de sistemas informáticos y comunicaciones integrados con los humanos
  - Se puede ver como una evolución de los sistemas distribuidos
  - Sinónimos: Calm Computing & Invisible Computing
- La tecnología es invisible pero ofrece servicios
  - Soporta movilidad
  - Integrada en la infraestructura
  - Proactiva

Context-Aware Computing

Universidad de Deusto
Deustuko Unibertsitatea

- Conciencia del contexto: context-awareness
  - ¿Cómo se representa el contexto? ¿Dónde se almacena?
  - ¿Cuándo se actualiza? ¿Cúantos recursos se dedican a ello?
  - ¿Qué técnicas de localización se emplean?
- Privacidad y confianza:
  - ¿Qué técnicas de autenticación utilizar?
  - Efecto “Big Brother”
- Abstracción del contexto
  - Filtrado
  - Interpretación
  - Anotación semántica

Sentient Computing

Universidad de Deusto
Deustuko Unibertsitatea

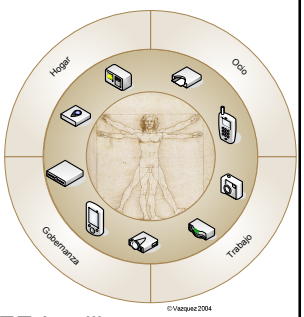
- Ability of computing systems to *detect*, *interpret* and *respond* to aspects of the user’s local environment
- Provides computing systems with awareness of their surrounding environment so that they can change their behaviour according to the observed situation
  - Devices themselves perceive the dynamic world they are part of through sensors
    - Computer systems and applications, when provided with a certain degree of perception, are made more natural, flexible, and adaptable
- Reference: Hopper, A., "The Clifford Paterson Lecture, 1999 Sentient Computing". Philosophical Transactions of the Royal Society London, vol. 358, no. 1773, pp. 2349-2358, August 2000

Pervasive Computing

- More systems oriented rather than user oriented (interaction, intelligence)
  - Term promoted by IBM
- Embraces several computing disciplines:
  - Distributed computing
  - Mobile computing
  - Sensor networks
  - Human-computer interaction
  - Artificial intelligence
- Reference: M. Satyanarayan “Pervasive Computing: Vision and Challenges”, IEEE Personal Communications, pp. 10-18, Agosto 2001

Ambient Intelligence

- Visión donde los humanos están rodeados de dispositivos computacionales y redes de ordenador accesibles por Interfaces Inteligentes
  - Promovido por el grupo de consejo ISTAG de la Comisión Europea
- AmI se centra en el usuario:
  - User friendliness
  - Provisión efectiva y distribuida de servicios
  - Mejora las capacidades de usuarios
  - Soporta sus interacciones de una manera natural
- Los sistemas deben ser:
  - sensibles al contexto
  - reactivos
  - interconectados
  - Inteligentes



- Referencia: N. Shadbolt, Ambient Intelligence, IEEE Intelligent Systems, Vol. 2, No.3, July/August 2003

Smart Object & Smart Space

- Smart Object → everyday objects augmented with computational services
- Smart Space → smart object ecosystems

Áreas de Interés

- Requisitos para cumplir la visión de AmI:
  - Hardware no intrusivo
    - miniaturización, nanotecnología, dispositivos inteligentes, sensores
  - Middleware e Infraestructura de comunicación móvil y fija transparente
  - Redes dinámicas y masivas de dispositivos computacionales y sensoriales
    - redes ad-hoc de sensores inalámbricos
  - Interfaces de interacción naturales e inteligentes
    - agentes inteligentes, interfaces multi-modal, sensibilidad al contexto
  - Robustez y Seguridad
    - privacidad, autenticación
  - Sistemas de razonamiento y aprendizaje basados en IA

Tecnologías Relacionadas

- TCP/IP, GPRS, GPS, UMTS
- Agents, Tuple Spaces, RPC, CORBA, Web Services, OSGi
- RFID, Computer Vision, UWB
- IPv6, DHCP, HTTP
- X10, Lonworks, EIB/KNX
- Bluetooth, Zigbee
- RDF, OWL, JENA, Semantic Web, Neural nets

Smart Service Samples

- Telephone Call Routing Application
- Print to the nearest printer
- Teleporting: reallocate a user's desktop to her new location
- Follow-me multimedia objects (nomadic user videoconference)
- Stick-e notes for ecological work or as reminders in a library
- Drawing on a computer and displaying it in the closest screen
- Forget-me-not system
- Augmented reality context-aware museum guides

Acceptability

- Issues of privacy and security
  - Intrusiveness
- Benefits must outweigh the drawbacks

Universidad de Deusto
Deustuko Unibertsitatea

mobility research lab

4.2. Context-Aware Computing

Dr. Diego Lz. de Ipiña Gz. de Artaza
http://paginaspesonales.deusto.es/dipina
http://www.morelab.deusto.es
http://www.ctmd.deusto.es

Definition of Context

- *"Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves".*
  - Dey A.K. and Abowd G.D. "Towards a Better Understanding of Context and Context-Awareness". Proceedings of the CHI 2000 Workshop on The What, Who, Where, When, and How of Context-Awareness, The Hague, Netherlands, April 2000
- Some attributes of context are:
  - Identity
  - Spatial information (location, orientation, speed and acceleration)
  - Temporal details (time of day, date and season of year)
  - Environmental situation (temperature, air quality, light and noise level)
  - Social interaction (who we are with and people that are nearby)
  - Resources that are nearby (accessible devices and hosts)
  - Capabilities of resources (display size, sound capabilities, video input)
  - Physiological measurements (blood pressure, heart rate, respiration)
  - Activity (talking, reading, walking, running)
  - Schedules and agendas.
  - Services locally available

Modelling Context

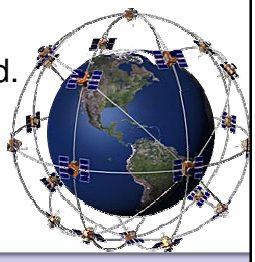
- Few attributes of context other than *location* are used in actual applications
  - Other important attributes of context are identity, activity and time
- Context is difficult to use by applications [Dey+00] for the following reasons:
  - Acquired from non-traditional devices
  - Must be abstracted to make sense to applications
  - Must be acquired from multiple distributed and heterogeneous sources
  - Dynamic nature, detection and reaction in real-time
- Needs:
  - Sensor fusion to merge data coming from heterogeneous systems
  - Frameworks to interpret raw data and generate higher level context understable by apps

Location-Aware Computing

- When we move, our context changes; the available computing resources surrounding us vary, as do our social interactions
  - Justifies the comprehensive research in systems that automatically locate people, equipment and other assets
- A system is considered location-aware if its behaviour is dependant on the position of objects in the environment
- Provision of a reliable location-tracking system is critical
  - Outdoors: GPS achieves 5m accuracy
  - Indoors: existing location technologies present a variety of drawbacks
    - high cost, custom-built hardware nature, and difficulties in their deployment and operation

GPS

- Triangulation from 24 satellites in active orbit are the basis of the GPS
- Radio signals are sent from orbiting satellites to Earth
- A GPS receiver measures distance using the travel time of radio signals from the ground to satellites.
- GPS radio signals are between 1,227 & 1,575 MHz
- GPS uses very precise clocks to measure the travel time between the ground and the satellite
- The exact location of the satellites are needed.
- Galileo European System



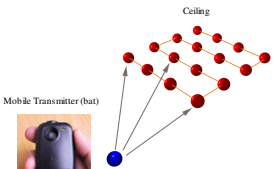
Indoor Location-Systems

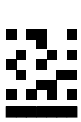
Universidad de Deusto
Deustuko Unibertsitatea

Technology	Physical Medium	Tagged (Active/Passive)/ Untagged	Ident. /Orie.	Physical/ Symbolic	Centralised / Local Location Computation	Location Granularity / Frequency	Off-the-shelf/ Bespoke Hardware Requirements	Cost	Limitations
Active Badge	Infrared	Tagged (Active)	I	S: room ID	Centralised	Room size, once every 10 secs	B: infrared sensors + badges	Maintenance costs	Sunlight and fluorescent light interference, thick tags (1 cm)
Bat/Ubisense	Ultrasonic / UWB	Tagged (Active)	I/O?	P: 3D coordinates	Centralised	3 cm (95% cases), 150 Hz per room	B: ultrasound sensor per ceiling tile + badges	Maintenance costs, expensive ($1,500 per room)	Very cumbersome to deploy. Thick (1.5 cm) t ags
Cricket/ Hexamite	Ultrasonic	Tagged (Active)	I	PS: knows distance to given base station	Local	30 cm (98% cases)	B: ultrasound beacons + receivers	$10 each beacon and receiver.	No central management, receiver computation
PinPoint's 3D-iD	RF	Tagged (Active)	I	P: distance to base station	Centralised	1-3 metres	B: base stations + tags	Expensive ($10,000 for 8-antenna kit)	Complex infrastructure
RADAR/ Nibble/ EKAHAU	802.11 RF	Tagged (waveLan card)	I	P: distance to base station	Centralised	3-4.3 metres (50% cases)	O: Wavelan Base Stations + PC cards	802.11 LAN installation + $100 per wireless NIC	Only attachable to devices with wireless NIC
GPS	Radio	Tagged (Active)	None	P: latitude, altitude, longitude	Local	1-5 metres (95-99% cases)	O: only requires purchase of GPS unit.	$100 per GPS receiver	Does not work indoors
Active Floor	Weight cells	Untagged	I?/O	P: exact position within floor	Centralised	Locates centre of mass (1cm), very fast (1KHz).	B: specially designed floor sensor grid.	Data acquisition card is expensive ($1,000)	Expensive, difficult to infer identity
Flock of Birds	Magnetic	Tagged (Active)	I/O	P: 6 degrees of freedom	Centralised	1 mm, 1ms, 0.1° (nearly 100 %), 100 Hz	B: specially build controller and sensors	Expensive hardware (from $1,500 to $90,000)	Control unit tethered, precise installation, high cost
EasyLiving	Vision	Untagged	I?/O	P: distance and orientation with respect to camera	Centralised	Within 5 cm, 7Hz	O: PCs+cameras	Uses very expensive cameras ($6,000)	Line of sight, high processing power, no identification, tolerates few people in room
CyberCode / ARToolKit	Vision	Tagged (Passive)	I/O	P: distance and orientation with respect to camera	Centralised	Within 1 cm, 1 degrees of error, 20 Hz	O: PCs+cameras	Very low, printed tags + cheap CCD cameras (around $50 each)	Line of sight, relatively high processing power, requires objects to be very close to camera

Indoor Location Systems

Universidad de Deusto
Deustuko Unibertsitatea





a) CyberCode Tag

b) ARToolkit Marker

Requirements for an Ideal Sensor System

- Use commonly available, off-the-shelf, technology.
- Require low maintenance costs and non-technical expertise.
- Provide fine-grained location information at a high update rate.
- Be minimally obtrusive.

- **The current trend is to go for Zigbee and RFID-based location!**

Technology	Physical Medium	Tagged (Active/ Passive)/ Untagged	Ident. /Orie.	Physical/ Symbolic	Centralised / Local Location Computation	Location Granularity / Frequency	Off-the-shelf/ Bespoke Hardware Requirements	Cost	Limitations
TRIP	Vision	Tagged (Passive)	I/O	P: distance and orientation with respect to camera	Centralised	3% average error on location and 2% on orientation, 16 Hz	O: PCs+cameras	Very low, printed tags + cheap CCD cameras (around $50 each)	Line of sight, relatively high processing power

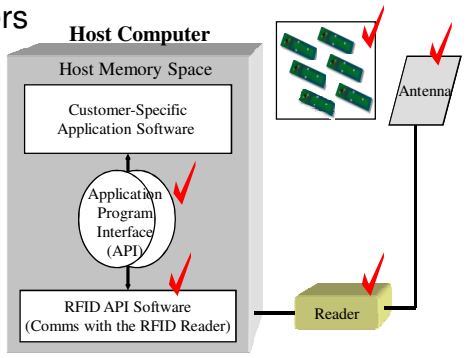
What is RFID?

- RFID uses radio-frequency waves to transfer data between a reader and a movable item to identify, categorize, track...
- RFID is fast, reliable, and does not require physical sight or contact between reader/scanner and the tagged item
- Linking RFID to outdoor GPS tracking and cellular systems is required for a complete solution

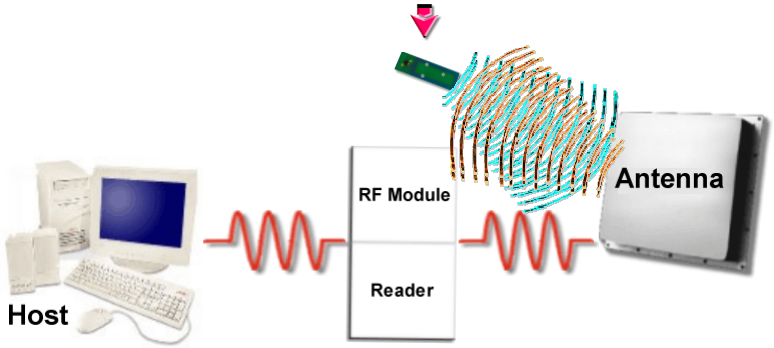
- Standards:
  - ISO 14443 A – a proximity card used for identification that usually uses the standard credit card form factor
  - EPC Global – standard for global RFID usage, family of coding schemes created as an eventual successor to the bar code – http://www.epcglobalinc.org/home

An RFID System

- One or more RF tags
- Two or more antennas
- One or more interrogators
- One or more host computers
- Appropriate software



An RFID System



Why is RFID in such a hype?

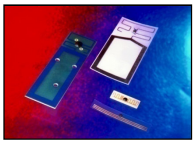
Universidad de Deusto
Deustuko Unibertsitatea

- After decades of technology refinement these radio tags have now become very small and inexpensive
- A great deal of innovation has brought us to the point where the silicon core of an radio tag is now 0.4mm
- The antenna can be printed onto a product's packaging at time of manufacture
- Currently radio tags can be purchased for less that €0.25 in quantity.
  - RFID is small enough, fast enough, and cheap enough to do real work with today's technology.

RFID Tag Types

Universidad de Deusto
Deustuko Unibertsitatea

- Active
  - Tag transmits radio signal
  - Battery powered memory, radio & circuitry
  - High Read Range (< 100m)
- Passive
  - Tag reflects radio signal from reader
  - Reader powered
  - Shorter Read Range (5 cm - 5 m)
- Features:
  - Memory
    - Size (16 bits - 512 kBytes +)
    - Read-Only, Read/Write or WORM
    - Type: EEProm, Antifuse, FeRam
  - Arbitration (Anti-collision)
    - Ability to read/write one or many tags at a time
  - Frequency
    - 125KHz - 5.8 GHz
  - Physical Dimensions
    - Thumbnail to Brick sizes
  - Price (€0.50 to €250)



Frequency Bands

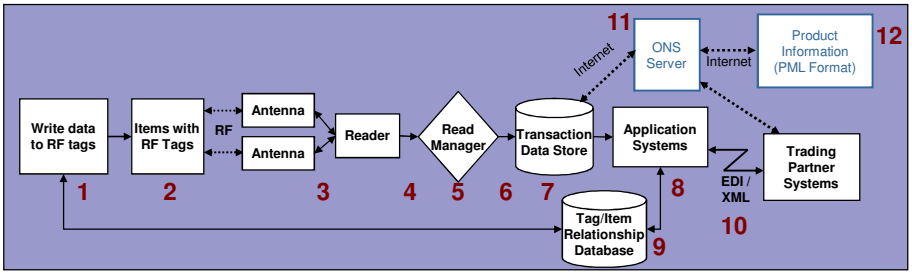
- LF : 125 KHz is used for short distance applications (1-50 cm): smart cards for access control, not absorbed by water
- HF: 13,56 MHz is used for short distance applications (1-100 cm): smart cards for access control, healthcare: faster than LF, more power
- UHF: 860-960 MhZ : Wal-Mart mandates, Target mandates, RFID supply chain management applications, distribution, eventually in a store at the item level (< 2m), bad behavour with human flesh and water.
- **Reference**: “RFID: A Technology Overview” at http://www.informit.com/articles/printerfriendly.aspx?p=413662

Frequency Bands

Referencia:
http://www.autoid.org/presentations/sep/**RFID_Basics**_qed.ppt

Frequency	Regulation	Range	Data Speed	Comments
125-150 kHz	Basically unregulated	Å 10 cm	Low	Animal identification and factory data collection systems
13.56 MHz	ISM band, differing power levels and duty cycle	< 1m	Low to moderate	Popular frequency for I.C. Cards (Smart Cards)
433 MHz	Non-specific Short Range Devices (SRD), Location Systems	1 Ğ 100 m	Moderate	Asset tracking for U.S. DoD (Pallets)
860-930 MHz	ISM band (Region 2); increasing use in other regions, differing power levels and duty cycle	2 Ğ 5 m	Moderate to high	EAN.UCC GTAG, MH10.8.4 (RTI), AIAG B-11 (Tires)
2450 MHz	ISM band, differing power levels and duty cycle	1 Ğ 2 m	High	IEEE 802.11b, Bluetooth, CT, AIAG B-11

RFID Use Chain



- Physical Markup Language: describes physical objects for industrial, commercial and consumer applications
  - http://web.mit.edu/mecheng/pml

RFID is still not perfect

- Does not work well if the tagged object is made of RF opaque material like metals, several type of liquids, carbon
- RF can penetrate only up to a limited depth of a material. So it may be impossible to read all the case tags of a pallet, even if the cases are made of RF-friendly materials
- The number of tags that can be simultaneously read/written is limited to about 50 – 100 / second
- The following environmental factors can hamper proper reading:
  - Speed with which the object is moving
  - Human bodies (composed largely of water)
  - Presence of RF interference, moisture and metals
- Tag technology is not mature, intensive research is being conducted both at the theoretical (e.g., antenna design) and manufacturing (e.g., material used, processing techniques)

Why is RFID important for AmI?

- Good contactless mechanisms to uniquely identify objects and even provide some metadata bound to them
- Enabling technology for Internet of Things vision
- Very important to combine it with mobile RFID readers which are transported by us everywhere at anytime: NFC mobile devices

Management of Context Information

- Requirements of Context Management Software:
  - Collection of raw sensor information
  - Transformation of raw context into an application-understandable format
  - Dissemination to interested applications is necessary.
- Some examples of Context Management Software:
  - The Active Badge Distributed Location Service
  - Situated Computing Service
  - SPIRIT
  - Stick-e note architecture
  - MicroSoft EasyLiving
  - HP CoolTown
  - Context Toolkit

Context Management Research

- Traditional object-oriented:
  - Context Toolkit
  - Project Aura
  - Oxygen
  - Equator
  - Gaia
- Semantic Web-based:
  - SOUPA, COBRA, Task Computing

Universidad de Deusto
Deustuko Unibertsitatea

mobility research lab

4.3. Middleware for Ambient Intelligence

Dr. Diego Lz. de Ipiña Gz. de Artaza
http://paginaspesonales.deusto.es/dipina
http://www.morelab.deusto.es
http://www.ctmd.deusto.es

Definition

- Software infrastructure which provides services to the user by targeting software towards a specific context
- Issues tackled:
  - Application adaptability
  - Mobile services
  - Semantic service discovery
  - Context-aware user interfaces

Problems to be tackled

- Explosive growth in computation, communication, information and integration technologies
  - computing & communication is ubiquitous
- Pervasive ad hoc “anytime-anywhere” access environments
  - ubiquitous access to information
  - peers capable of producing/consuming/processing information at different levels and granularities
  - embedded devices in clothes, phones, cars, mile-markers, traffic lights, lamp posts, medical instruments …
- “On demand” computational/storage resources, services

Middleware for AmI: Challenges

Universidad de Deusto
Deustuko Unibertsitatea

- Individual system elements increasingly difficult to maintain and operate
  - large numbers of configuration/tuning parameters
- Heterogeneous systems are becoming increasingly connected
  - integration becoming ever more difficult
- Behaviors, execution context, interactions not known a priori
  - increasingly dynamic, opportunistic and unanticipated

AmI Middleware Issues

Universidad de Deusto
Deustuko Unibertsitatea

- Very large scales
  - million of entities
- Ad hoc (amorphous) structures/behaviors
  - p2p/hierarchical architecture
- Dynamic
  - entities join, leave, move, change behavior
- Heterogeneous
  - capability, connectivity, reliability, guarantees, QoS
- Unreliable
  - components, communication
- Lack of common/complete knowledge
  - number, type, location, availability, connectivity, protocols, semantics, etc.

Requirements for AmI Middleware

Universidad de Deusto
Deustuko Unibertsitatea

- Capture, interpret, model and reason about contextual information
- Allow spontaneous discovery of services based on user context
- Ease the dynamic deployment, interoperability, interaction and coordination among shared services

Discovery and Interaction in AmI

Universidad de Deusto
Deustuko Unibertsitatea

- **Discovery** → a mechanism to discover through ad-hoc or wireless networking the computing services exported by surrounding smart objects
- **Interaction** → a mechanism to interact with those discovered services, so that the represented objects adapt to the user's commands
- Types:
  - Discovery protocols supported by mobile code (Jini)
  - Discovery protocol integrated with specific interaction protocols (UPnP)
  - Interaction independent discovery protocols (SLP)
- Reference:
  - Grace P., Blair G.S., Samuel S., 2005, "A Reflective Framework for Discovery and Interaction in Heterogeneous Mobile Environments", ACM SIGMOBILE, vol. 9, no. 1
  - Service Discovery in Pervasive Computing Environments, October-December 2005 (Vol. 4, No. 4) pp. 81-90

Properties of AmI Middleware

Universidad de Deusto
Deustuko Unibertsitatea

- Pervasive Computing is different from Distributed Computing:
  - Context-awareness
  - Dynamism
  - Heterogeneity
- Issues in a pervasive computing environment:
  - Mobility
  - Disconnection
  - Dynamic introduction and removal of devices
  - Merging of the physical environment with the computational infrastructure are common and affect the underlying middleware infrastructure.
  - Different devices might be connected to different networks, with different latency and bandwidth.
- We need middleware that:
  - Provide mechanisms for handling disconnection
  - Addressing fault tolerance, and
  - Adapting to a number of issues related to diversity
  - Evolving and self-healing

Middleware Examples

Universidad de Deusto
Deustuko Unibertsitatea

- Non-AmI specific:
  - Aspect-Oriented Programming
  - OSGi
  - UPnP
  - Semantic Web Services: Task Computing
- AmI-specific:
  - Autonomic computing
  - COBRA, SOUPA
  - EMI2lets

Aspect-Oriented Programming (AOP)

Universidad de Deusto
Deustuko Unibertsitatea

- Programación orientada a aspectos (AOP):
  - Concepto
  - Aplicabilidad de AOP
  - Herramientas: AspectJ

AOP: Concepto

Universidad de Deusto
Deustuko Unibertsitatea

- OOP modela comportamiento en una jerarquía de objetos → es vertical
- AOP factoriza conceptos comunes a varias aplicaciones (*cross-cutting concerns*) → es horizontal
  - Ej. Logeo, Seguridad, Transacciones
  - Creado por Gregor Kiczales en Xerox PARC
- Objetivos:
  - Separar conceptos comunes a aplicaciones
  - Evitar solapamiento de funcionalidad
- **¡¡¡No substituye OOP, la complementa!!!**

AOP: Concepto

- Un concepto (*concern*) puede clasificarse en:
  - Concepto fundamental de negocio (*core business concern*)
    - Procesar pagos en un sistema de tarjetas de crédito
  - Concepto general (*system concern*)
    - Logeo, transacciones, autenticación, seguridad, etc.
    - Aparecen en muchas aplicaciones
- OOP hace que se repita funcionalidad de sistema en muchos módulos:
  - Mayor complicación del código
  - Diseño e implementación más difícil
  - Más difícil la evolución

AOP: Aspecto

- Una implementación de AOP encapsula la funcionalidad común a varias aplicaciones a través del concepto de Aspecto (*aspect*)
  - Aspecto = módulo de código que factoriza funcionalidad común correspondiente a requisitos no funcionales
    - Compuesto de:
      - Consejos o advices (funcionalidad adicional a añadir) y
      - Puntos de unión (join points), lugares dónde se añade esa funcionalidad extra
  - Ejemplo: un aspecto de seguridad puede incluir un consejo de seguridad que añade instrucciones de comprobación al comienzo de los métodos `a()`, `b()` y `c()` de una clase

AOP: Motivación

Universidad de Deusto
Deustuko Unibertsitatea

- Consideremos una aplicación bancaria:

```
void transfer(Account fromAccount, Account
  toAccount, int amount) {
  if (fromAccount.getBalance() < amount) {
    throw new InsufficientFundsException();
  }
  fromAccount.withdraw(amount);
  toAccount.deposit(amount);
}
```

- Normalmente, "corrompemos" código con detalles no funcionales: autorización, logeo, transacciones, etc.

AOP: Motivación

Universidad de Deusto
Deustuko Unibertsitatea

```
void transfer(Account fromAccount, Account toAccount, int amount) {
  if (!getCurrentUser().canPerform(OP_TRANSFER)) {
    throw new SecurityException();
  }
  if (amount < 0) {
    throw new NegativeTransferException();
  }
  if (fromAccount.getBalance() < amount) {
    throw new InsufficientFundsException();
  }
  Transaction tx = database.newTransaction();
  try {
     fromAccount.withdraw(amount);
     toAcount.deposit(amount);
     tx.commit();
     systemLog.logOperation(OP_TRANSFER, fromAccount, toAccount, amount);
  }
  catch(Exception e) {
     tx.rollback();
  }
}
```

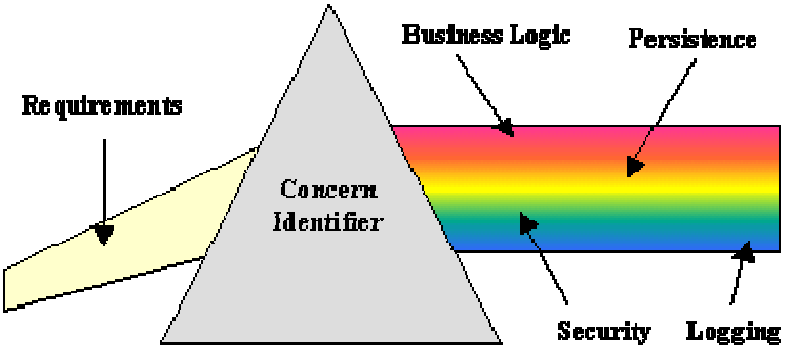
AOP: Motivación

Universidad de Deusto
Deustuko Unibertsitatea

- El código anterior mezcla:
  - Business logic concerns y
  - Cross-cutting concerns
- Consecuencias:
  - Dificultad para cambiar implementación de cross-cutting concerns
    - Los conceptos comunes están desperdigados por el código → ¡¡¡FALTA DE MODULARIZACIÓN!!!
- AOP pretende resolver esta situación

AOP: Motivación

Universidad de Deusto
Deustuko Unibertsitatea



JPM: Entrelazando AOP y OOP

Universidad de Deusto
Deustuko Unibertsitatea

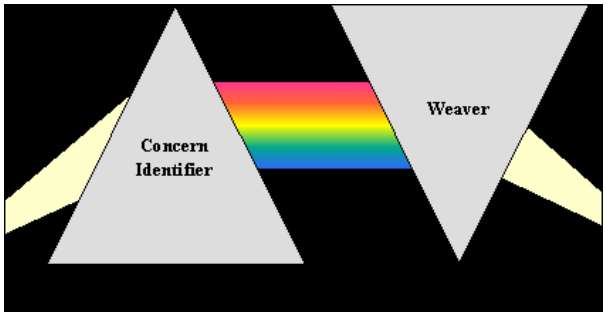
- El Joint Point Model (JPM) define cómo un aspecto interactúa con un programa, mediante:
  - Puntos de unión (*Joinpoints*) → dónde puede aplicarse el aspecto
  - Puntos de corte (*Pointcuts*) → cuándo, conjunto de joinpoints, consultas sobre ellos
  - Consejos (*advices*) → cómo, funcionalidad a añadir en los joinpoints

Weaving: "Tejiendo programas"

Universidad de Deusto
Deustuko Unibertsitatea

- ¿Cómo inyectar consejos en puntos de unión de un programa?
  - Mediante un preprocesador → complica desarrollo
  - Un postprocesador binario → complica desarrollo
  - Un compilador específico a AOP → AspectJ
  - Durante la carga de clases → lento
  - En tiempo de ejecución → lento
- Esto es un tema de investigación actualmente

Weaving: “Tejiendo programas”



Un lenguaje AOP: AspectJ

- Permite la definición de aspectos en un programa Java:
  - **Declaraciones entre-tipos**: añade métodos, campos o interfaces a clases

```
aspect VisitAspect {
  Point.acceptVisitor(Visitor v) {
    v.visit(this);
  }
}
```

  - **Pointcuts**: permite a un programa especificar un conjunto de puntos de unión

```
pointcut set() : execution(* *.set*(..) ) && this(Point);
```

  - **Advice**: permiten a un programador especificar las acciones a ejecutar cuando un pointcut se evalua a `true`

```
after () : set() {
  Display.update();
}
```

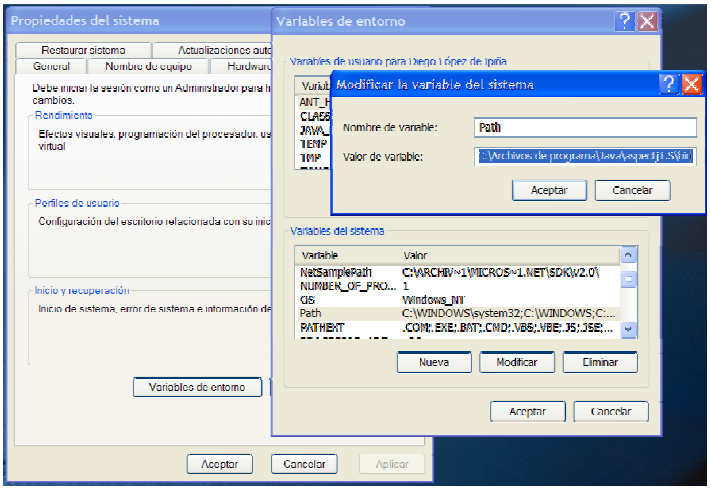
Instalación AspectJ

Universidad de Deusto
Deustuko Unibertsitatea

- Descargarlo de: http://www.eclipse.org/aspectj/
- Instalar: `java -jar aspectj-DEVELOPMENT-20060426104617.jar`
- Configuración:
  - Añadir `C:\Archivos de programa\Java\aspectj1.5\lib\aspectjrt.jar` a tu `CLASSPATH`.
    - Contiene las clases requeridas por cualquier programa que utiliza el compilador `ajc`.
  - Modificar tu `PATH` para incluir `C:\Archivos de programa\Java\aspectj1.5\bin`.
    - `Ajc` y `AjcBrowser`

Instalación AspectJ

Universidad de Deusto
Deustuko Unibertsitatea



Ejemplo AspectJ

- Añadir capacidad de logeo con aspectos a una clase:

```
public class ClassExample {
   public static void method1() {
      System.out.println("¡Hola ESIDE!");
   }

   public static void method2() {
      System.out.println("¡Hola revista ESIDE!");
   }

   public static void saySomething(String x) {
      System.out.println(x);
   }

   public static void main(String[] args) {
      method1();
      method2();
      saySomething("¡Aprende AOP!");
   }
}
```


Ejemplo AspectJ


```
// AspectExample.aj
public aspect AspectExample {
    public pointcut methodCall(): call(public* ClassExample.*());
    public pointcut methodCallArg(String a): call(public* ClassExample.*(String)) &&
    args(a);

    before(String a): methodCallArg(a) {
          System.out.print("\n -- ClassExample.");
          System.out.println(thisJoinPointStaticPart.getSignature().getName() + "(" + a +
    ") empezando--");
    }

    after(String a): methodCallArg(a) {
          System.out.print("\n -- ClassExample.");
          System.out.println(thisJoinPointStaticPart.getSignature().getName() + "(" + a +
    ") ejecutado--");
    }

    before(): methodCall() {
          System.out.println("\n -- ClassExample." +
    thisJoinPointStaticPart.getSignature().getName() + " empezando --");
    }

    after(): methodCall() {
          System.out.println(" -- ClassExample." +
    thisJoinPointStaticPart.getSignature().getName() + " ejecutado--\n");
    }
}
```

Ejemplo AspectJ

Universidad de Deusto
Deustuko Unibertsitatea

- El anterior código hace lo siguiente:
  1. Define **un nuevo aspecto** con el nombre **`AspectExample`**
  2. Define **dos puntos de unión** para:
     - Métodos públicos de `ClassExample` sin argumentos (**`methodCall`**) y
     - Métodos que aceptan como argumento un `String` (**`methodCallArg`**).
  3. Define **cuatro consejos** (fragmento de código) a ejecutar:
     - Antes de la invocación de los métodos públicos de `ClassExample` con un argumento de tipo `String`.
     - Después de que se ejecuten los métodos que tienen un `String` como parámetro.
     - Antes y después de que se invoquen los métodos de `ClassExample` sin argumentos.
- La siguiente línea de código devuelve el nombre del método a invocar:

```
thisJointPointStaticPart.getSignature().getName()
```


Ejemplo AspectJ

Universidad de Deusto
Deustuko Unibertsitatea

- Para compilar este código y luego ejecutarlo escribe:

```
ajc AspectExample.aj ClassExample
java ClassExample
```

- El resultado será:

```
-- ClassExample.method1 empezando --
¡Bienvenidos a Araba Enpresa Digitala!
 -- ClassExample.method1 ejecutado--
 -- ClassExample.method2 empezando --
¡Bienvenidos a Últimas Tendencias en Desarrollo de
  Software Empresarial: SOA y Web 2.0!
 -- ClassExample.method2 ejecutado--
 -- ClassExample.saySomething(¡Aprende AOP!) empezando--
¡Aprende AOP!
 -- ClassExample.saySomething(¡Aprende AOP!) ejecutado–
```

- Sin tener que cambiar una sola línea de `ClassExample` hemos sido capaces de añadir capacidad de logeo y testear esa clase

Ejemplos Pointcuts

- Se pueden usar los operadores relacionales ||, && y ! para combinarlos

`call(void MyClass.myMethod(..))`	Call to `myMethod()` in `MyClass` taking any arguments, with `void` return type, and any access modifiers
`call(* MyClass.myMethod(..))`	Call to `myMethod()` in `MyClass` taking any arguments returning any type
`call(MyClass+.new(..))`	Call to any `MyClass` or its subclass's constructor. (Subclass indicated by use of '+' wildcard)
`call(public * com.mycompany..*.*(..))`	All public methods in all classes in any package with `com.mycompany` the root package
`execution(* *.myMethod(..))`	Execution of `myMethod()` in any class in default package
`execution(MyClass.new())`	Execution of any `MyClass` constructor taking no arguments
`get(PrintStream System.out)`	Execution of read-access to field out of type `PrintStream` in System class
`set(int MyClass.x)`	Execution of write-access to field `x` of type `int` in `MyClass`
`handler(RemoteException)`	Execution of catch-block handling `RemoteException` type
`this(JComponent+)`	All the joinpoints where `this` is instanceof `JComponent`
`target(MyClass)`	All the joinpoints where the object on which the method is called is of type `MyClass`
`args(String,..,int)`	All the joinpoints where the first argument is of `String` type and the last argument is of `int` type

Ejemplo advices

- Se puede asociar código en AOP antes, después o durante un joinpoint:

```
before() : call(public * MyClass.*(..)) {
        System.out.println("Before: " + thisJoinPoint + " " +
System.currentTimeMillis());
}

after() : call(public * MyClass.*(..)) {
        System.out.println("After: " + thisJoinPoint + " " +
System.currentTimeMillis());
}
```

- El siguiente consejo substituye invocaciones a `Connection.close()` por añadir la conexión a un pool en caso de haberse permitido pooling:

```
void around(Connection conn) : call(Connection.close()) &&
target(conn) {
    if (enablePooling) {
        connectionPool.put(conn);
    } else {
        proceed();
    }
}
```

AOP en la Empresa

- Usos empresariales:
  - Añadir `ThreadPools` a una aplicación
  - Logeo y depuración
  - Lazy creation/initialization
  - Cacheo
- Muchos servidores de aplicaciones y frameworks incorporan capacidades AOP:
  - Spring (http://www.springframework.org/)
  - Jboss (http://java-source.net/open-source/aspect-oriented-frameworks/jbossaop)

AOP: Conclusión

- Ventajas AOP:
  - Código menos enmarañado, más natural y más reducido.
  - Mayor facilidad para razonar sobre los conceptos: separados y mínimas dependencias
  - Código más fácil de mantener
  - Se consigue que un conjunto grande de modificaciones en la definición de una materia tenga un impacto mínimo en las otras.
  - Se tiene un código más reusable y que se puede acoplar y desacoplar cuando sea necesario.
- Desventajas:
  - Depuración de código más difícil
  - Captura no intencionada de puntos de unión

Research in AOP

Universidad de Deusto
Deustuko Unibertsitatea

- Portal of Aspect-Oriented Software Development Conference:
  - http://aosd.net/
- The AOP Alliance:
  - http://aopalliance.sourceforge.net/
- Can AOP be used in AmI Middleware?
- Example paper:
  - Tal Cohen and Joseph (Yossi) Gil , “AspectJ2EE = AOP + J2EE: Towards an Aspect Based, Programmable and Extensible Middleware Framework “, ECOOP} 2004 – Object-Oriented Programming, {LNCS} 3086, Springer-Verlag, 2004

Referencias

Universidad de Deusto
Deustuko Unibertsitatea

- AOP
  - I want my AOP!, Part 1, Separate software concerns with aspect-oriented programming
    - http://www.javaworld.com/javaworld/jw-01-2002/jw-0118-aspect.html
  - I want my AOP!, Part 2, Learn AspectJ to better understand aspect-oriented programming
    - http://www.javaworld.com/javaworld/jw-03-2002/jw-0301-aspect2.html
  - I want my AOP!, Part 3, Use AspectJ to modularize crosscutting concerns in real-world problems
    - http://www.javaworld.com/javaworld/jw-04-2002/jw-0412-aspect3.html

OSGi (Open Services Gateway Initiative)

Universidad de Deusto
Deustuko Unibertsitatea

- Una **framework Java para el desarrollo de aplicaciones desplegables remotamente**, que requieren:
  - Robustez
  - Elevada distribución
  - Heterogeneidad (diferentes dispositivos)
  - Colaboraciones

- Resuelve el problema de **desplegar muchos programas independientes en grandes sistemas distribuidos**, proveyendo:
  - Un sistema operativo para programas
  - Un formato para la descarga de código ejecutable
  - Un mecanismo para descubrir otros programas
  - Estandarización de APIs para promocionar la reutilización
  - Gestión eficiente del ciclo de vida de las aplicaciones descargadas

OSGi (Open Services Gateway Initiative)

Universidad de Deusto
Deustuko Unibertsitatea

- Product is the OSGi Service Platform Specification
  - Currently in release 4
  - OSGi Alliance website: http://www.osgi.org/
  - Specification available at: http://www.osgi.org/osgi_technology/license_agreement.asp
    - OSGi_R4.core.pdf (2.06 MB) -- R4 Core Specification
    - OSG_ R4.cmpn.pdf (3.47 MB) -- R4 Service Compendium

- Industry Alliance
  - 30+ companies are members

Alpine Electronics Europe Gmbh , Aplix Corporation , BMW Group , Computer Associates , Deutsche Telekom AG , Echelon Corporation , Electricité de France (EDF) , Ericsson Mobile Platforms AB , Esmertec , Espial Group, Inc. , ETRI Electronics and Telecommunications Research Institute , France Telecom , Gatespace Telematics AB , Gemplus , Harman/Becker Automotive Systems GmbH , Hitachi, Ltd. , IBM Corporation , Industrial Technology Research Institute , Insignia Solutions , Intel Corporation , KDDI R&D Laboratories, Inc. , KT Corporation , Mitsubishi Electric Corporation , Motorola, Inc. , NEC Corporation , Nokia Corporation , NTT , Oracle Corporation , Panasonic Technologies, Inc. , ProSyst Software GmbH , Robert Bosch Gmbh , Samsung Electronics Co., Ltd. , SavaJe Technologies, Inc. , Sharp Corporation , Siemens AG , Sun Microsystems, Inc. , Telcordia Technologies, Inc. , Telefonica I+D , TeliaSonera , Vodafone Group Services Limited

OSGi (Open Services Gateway Initiative)

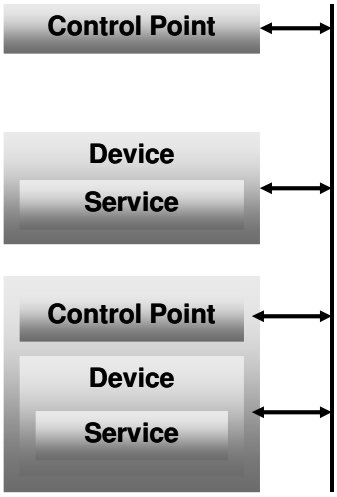
- REVISAR APÉNDICE A. OSGi

UPnP: Universal Plug & Play

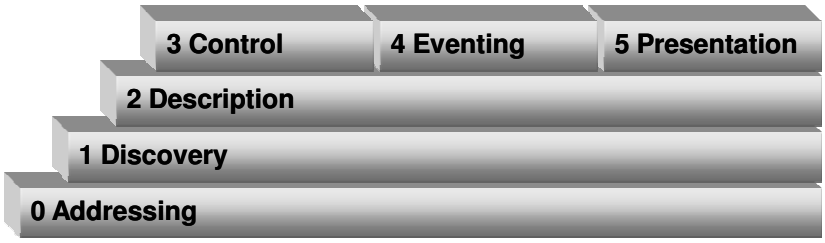
- Regulated by the UPnP Forum – http://www.upnp.org
- Addresses distributed control by humans of networked devices that can come and go dynamically
  - Allows devices to connect seamlessly and to simplify the implementation of networks in the home and corporate environment
- For that purpose, it defines protocols:
  - Discovery of devices on a local network
  - A notification mechanism with publish/subscribe
- And a language
  - A formal language that defines the actions and variables of a device

UPnP

- Describe the protocols for communication between:
  - **Control points**
    - Controller, usually client
  - **Device**
    - Controlled, usually server
  - An actual device might contain both functions

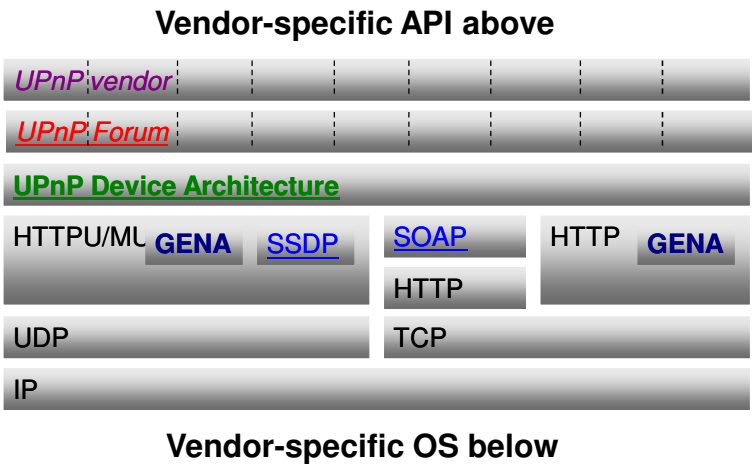


Steps to UPnP Networking



1. Control point and device get addresses
2. Control point finds interesting device
3. Control point learns about device capabilities
4. Control point invokes actions on device
5. Control point listens to state changes of device
6. Control point controls device and/or views device status using HTML UI

UPnP Protocol Stack



OSGi vs. UPnP

- There is no "versus" here. OSGi is fully complimentary to UPnP. No overlap.
  - UPnP = set of protocols to discover and communicate with networked devices
    - UPnP Implementations could use OSGi as execution environment like they could use Windows, Linux or QNX operating systems
- OSGi = environment for Java programs to execute in a well defined and managed environment
  - OSGi implementations could use UPnP (or Jini, or SLP, or Salutation) to discover and use networked devices

OSGi & UPnP: Comparison

OSGi	UPnP
Java	XML DTD
Executing code	Communications
Behaviour (Code)	Declarative
Program-Program oriented	User oriented
Standardizing Java interfaces	Standardizing XML templates
Service is local and fast	Service is remote and slow to execute

Semantic Web & AmI

- AmI needs Semantic Web technologies to deal with its inherently ad-hoc relationships between users, devices and services
  - Allows the user to be “out of the loop”
  - The environment can anticipate based on user preferences, context and profiles

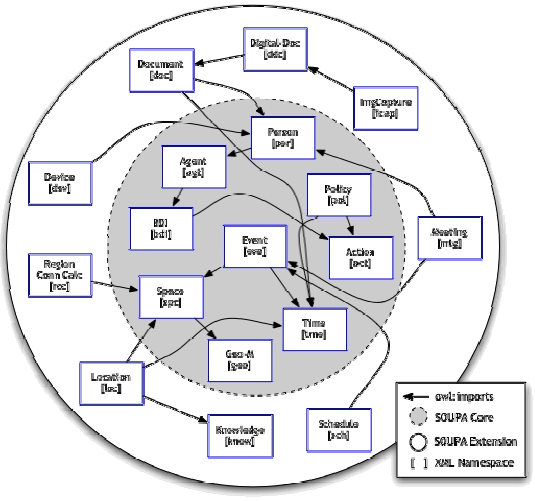
SOUPA: Standard Ontology for Ubiquitous and Pervasive Apps

Universidad de Deusto
Deustuko Unibertsitatea

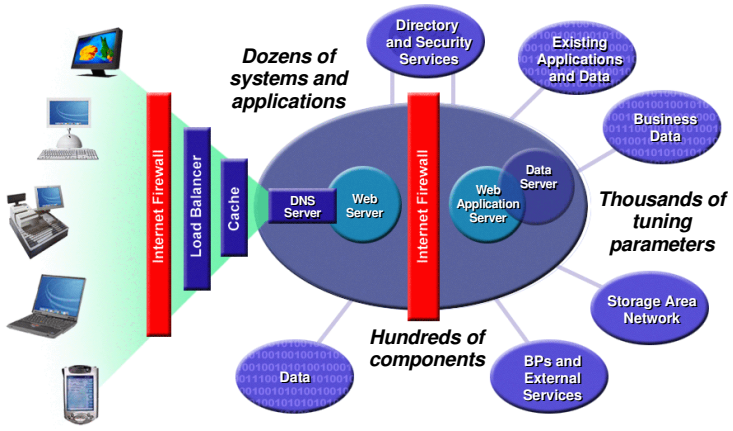
- An ontology for supporting knowledge sharing, context reasoning and interoperability in open and dynamic AmI environments
  - Expressed using OWL
  - Combines useful vocabularies from different consensus ontologies (FOAF, DAML-Time, OpenCyc, etc)
  - Defines vocabularies applied to any pervasive computing application: person, agent, belief-desire-intention, action, policy, space, time and event.
- CoBrA is an agent architecture for Smart Spaces
  - A Context Broker running in a resource rich machine:
    - Acquires and maintains context knowledge
    - Reasons about the info
    - Detects and resolves inconsistencies
    - Protects user privacy

SOUPA

Universidad de Deusto
Deustuko Unibertsitatea



Autonomic Computing



Autonomic Computing: Self-managing Computing Systems

- Administration of individual systems is increasingly difficult
  - 100s of configuration, tuning parameters for DB2, WebSphere
- Heterogeneous systems are becoming increasingly connected
  - Integration becoming ever more difficult
- Architects can't intricately plan interactions among components
  - Increasingly dynamic; more frequently with unanticipated components
- More of the burden must be assumed at run time
  - But human system administrators can't assume the burden; already
    - 6:1 cost ratio between storage admin and storage
    - 40% outages due to operator error
- We need self-managing computing systems
  - Behavior specified by sys admins via high-level policies
  - System and its components figure out how to carry out policies

Evolving Towards Self-Management

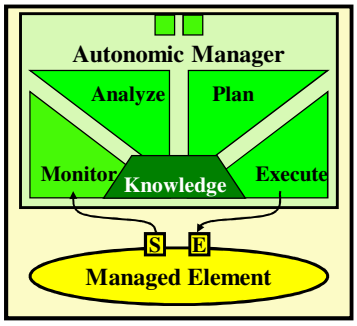
Universidad de Deusto
Deustuko Unibertsitatea

	Today	The Autonomic Future
Self-configure	Corporate data centers are multi-vendor, multi-platform. Installing, configuring, integrating systems is time-consuming, error-prone.	Automated configuration of components, systems according to high-level policies; rest of system adjusts automatically. Seamless, like adding new cell to body or new individual to population.
Self-heal	Problem determination in large, complex systems can take a team of programmers weeks	Automated detection, diagnosis, and repair of localized software/hardware problems.
Self-optimize	WebSphere, DB2 have hundreds of nonlinear tuning parameters; many new ones with each release.	Components and systems will continually seek opportunities to improve their own performance and efficiency.
Self-protect	Manual detection and recovery from attacks and cascading failures.	Automated defense against malicious attacks or cascading failures; use early warning to anticipate and prevent system-wide failures.

Autonomic Computing: Architecture

Universidad de Deusto
Deustuko Unibertsitatea

- Autonomic Elements (AE): atoms of autonomic computing
- An AE contains
  - Exactly one autonomic manager
  - Zero or more managed element(s)
- AE is responsible for
  - Managing own behavior in accordance with policies
- Interacting with other autonomic elements to provide or consume computational services



An Autonomic Element

Task Computing

- Fills the gap between:
  - Tasks (what user wants to do)
  - Services (what is available)
    - Devices, Web Services, Applications
- A Task Computing Framework (TCF) is a framework that supports Task Computing, by providing support for:
  - The workflows of Task Computing, i.e., at a minimum, Discovery, followed by Composition and Execution
  - Semantic description of tasks and services
  - Specification, Execution and Re-Usability of tasks by end-users
  - Manipulation including creation and disposal of services by the end-users

Task Computing

- Nuevo paradigma → cómo interactúan los usuarios con dispositivos y servicios.
- Pretende rellenar el hueco entre lo que el usuario quiere hacer y los servicios existentes.
- Se centra en el qué y no en el cómo.
- Permite que los usuarios hagan composición de servicios on-the-fly.

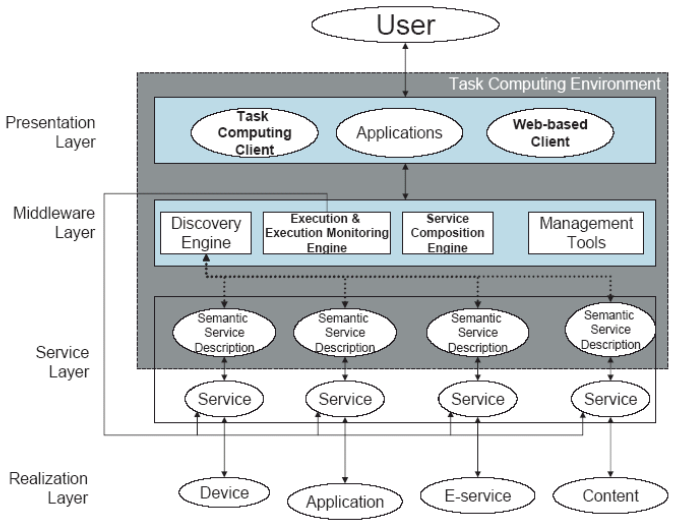
Posibles escenarios

Universidad de Deusto
Deustuko Unibertsitatea

- Enviar el video de seguridad captado por una cámara a cualquier número de dispositivos (TV, monitores, PDA-s, etc…) sin conectar ningún cable.
- Llamar a un número que se disponga en un PIM (Personal Information Management) usando un teléfono de una sala de reuniones que se visita por primera vez.
- Mostrar en el navegador Web información sobre el tiempo en la localización de un contacto metido en el PIM.

Arquitectura

Universidad de Deusto
Deustuko Unibertsitatea



MORE LAB mobility research lab

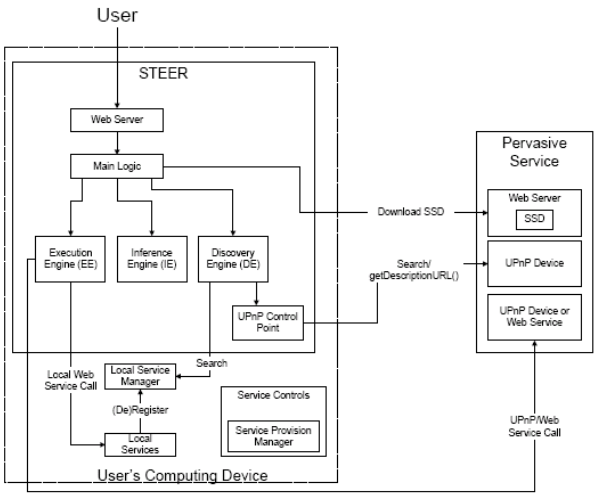
Task Computing Framework (TCF)

- Contiene el workflow: Descubrimiento, composición y ejecución.
- Permite la descripción semántica de servicios y tareas.
- Permite la especificación, ejecución y reutilización de tareas por parte de los usuarios.
- Permite la manipulación de los servicios.

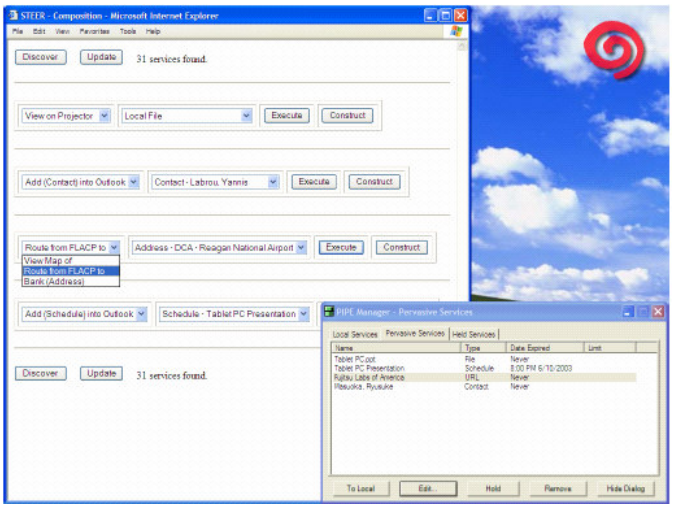
Task Computing Enviroment (TCE)

- Se compone de:
  - Clientes (TCC)
  - Servicios descritos semánticamente (SDS)
  - Mecanismos de descubrimiento (SSDM)
  - Servicios de Control (SC)
- Hace uso de tecnologías de servicios web semánticos:
  - DAML-S
  - OWL-S
  - WSDL
  - SOAP

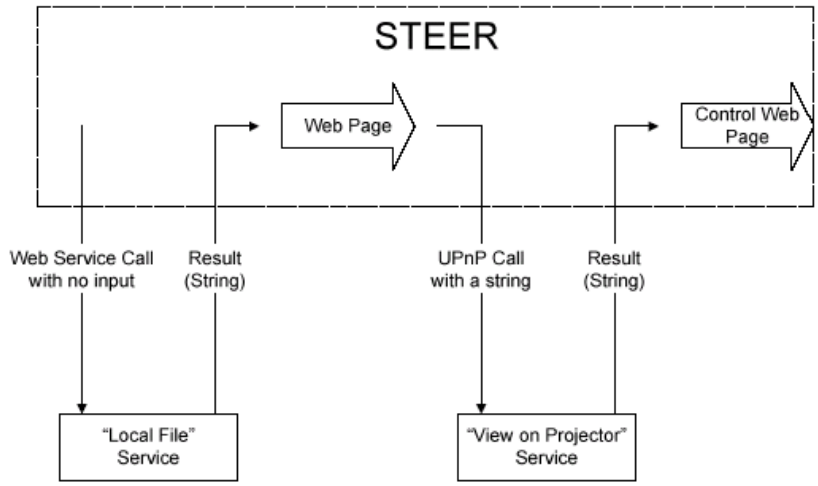
STEER (Semantic Task Execution editoR)



STEER



STEER



Ejemplos de composición de servicios

- Contact Provider (on Bob's PDA) + Add Into Contact List
- Local File + View On Proyector
- Local File + Bank (File)
- Schedule from Scheduler + Bank (Schedule) → Schedule Provider (created by Bank (Schedule)) + Add into Schedule List

Referencias

- Ryusuke Masuoka, Mohinder Chorpa, Yannis Labrou, Zhexuan Song, Wei-lun Chen, Lalana Kagal, and Tim Finin, "*Policy-based Access Control for Task Computing Using Rei*," Policy Management for the Web (PM4W), A WWW2005 Workshop, 14th International World Wide Web Conference May 10, 2005, Chiba, Japan
- Ryusuke Masuoka, Yannis Labrou, and Zhexuan Song, "*Semantic Web and Ubiquitous Computing - Task Computing as an Example* -" AIS SIGSEMIS Bulletin, Vol. 1 No. 3, October 2004, pp. 21 - 24
- Zhexuan Song, Yannis Labrou and Ryusuke Masuoka, "*Dynamic Service Discovery and Management in Task Computing*," pp. 310 - 318, MobiQuitous 2004, August 22-26, 2004, Boston, USA
- Ryusuke Masuoka and Masanobu Yuhara, "*Task Computing - Filling the Gap between Tasks and Services*," FUJITSU, pp. 376 - 383, July 2004
- Ryusuke Masuoka, Bijan Parsia, Yannis Labrou and Evren Sirin, "*Ontology-Enabled Pervasive Computing Applications*," IEEE Intelligent Systems, vol. 18, no. 5, Sep./Oct. 2003, pp. 68-72.
- Ryusuke Masuoka, Bijan Parsia and Yannis Labrou, "*Task Computing - the Semantic Web meets Pervasive Computing*," 2nd International Semantic Web Conference (ISWC2003), 20-23 October 2003, Sanibel Island, Florida, USA
- Ryusuke Masuoka and Yannis Labrou, "*Task Computing - Semantic-web enabled, user-driven, interactive environments* ," WWW Based Communities For Knowledge Presentation, Sharing, Mining and Protection (The PSMP workshop) within CIC 2003, June 23 - 26, 2003, Las Vegas, USA
- http://taskcomputing.org

EMI2lets: An AmI-specific Middleware

- Latest mobile devices used mainly for communication, entertainment or as electronic assistants
- However, their increasing:
  - Computational power
  - Storage
  - Communications (Wi-Fi, Bluetooth, GPRS)
  - Multimedia capabilities (Camera, RFID reader)
  - Extensibility
- Ideal to act as intermediaries between us and the environment:
  - Aware (Sentient) Devices
  - Powerful devices
  - Always with us anywhere at anytime
- **Our mobile devices can turn into our personal butlers**

Motivation

- Our goals are to **build Smart Spaces** and **transform mobile devices into Universal Remote Controllers** of Anything Anywhere at Anytime
  - Mobile devices equipped with Bluetooth, cameras, barcode, GPS or RFID are sentient devices
    - http://www.ctmd.deusto.es/mobilesense
  - A **Smart Space** is a container, either indoors or outdoors, of Smart Objects
  - A **Smart Object** is an everyday object (e.g. door) or device augmented with some computational service.
- Definition of suitable AmI architectures may be a good starting point to make AmI reality

EMI²lets Platform I

- EMI²lets is a middleware to facilitate the development and deployment of mobile context-aware applications for AmI spaces.
- This software platform provides the infrastructure to:
  - convert physical environments into AmI spaces
    - augment daily life objects with computational services
  - transform mobile devices into remote controllers of those Smart Objects

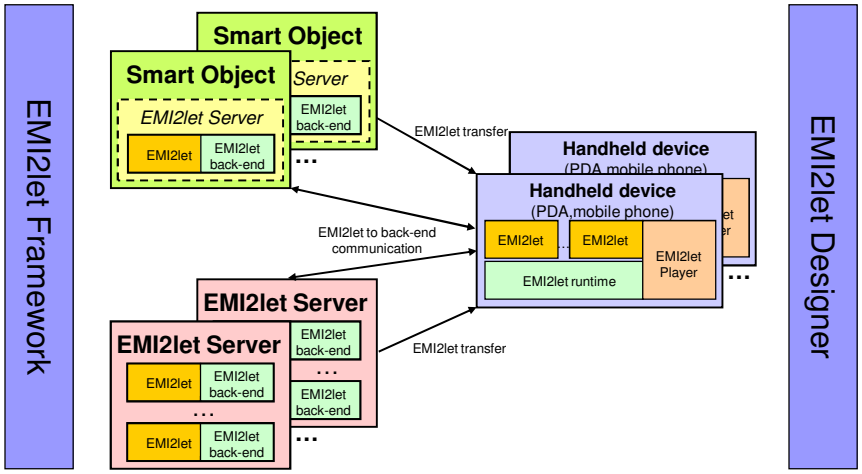
EMI²lets Platform II

- EMI²lets is an AmI-enabling middleware
  - addresses the **service discovery and interaction** aspects required for **active influence** on EMI²Objects
- Follows a Jini-like mechanism and Smart Client paradigm
  - once a service is discovered, a proxy of it (an EMI²let) is downloaded into the user's device (EMI²Proxy).
  - An **EMI²let** is a mobile component transferred from a Smart Object to a nearby handheld device, which offers a graphical interface for the user to interact over that Smart Object

Properties

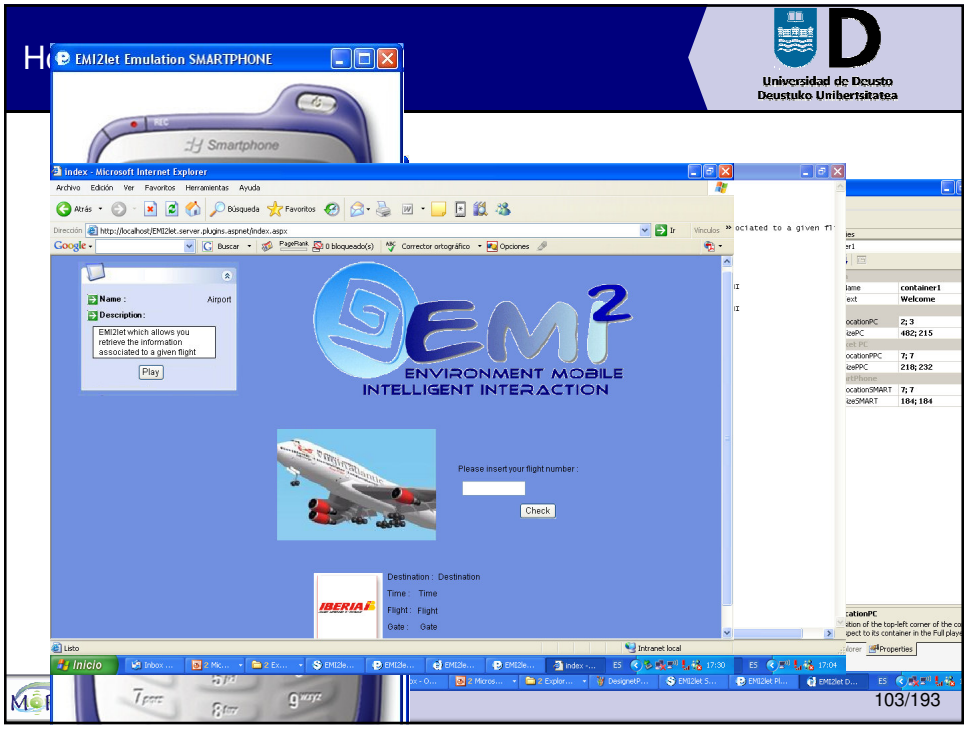
1. Transform mobile devices into remote universal controllers of Smart Objects
2. Enable local (Bluetooth, Wi-Fi) and global access (GPRS/UMTS) to Smart Objects
3. Develop middleware independent of a particular discovery or interaction mechanism.
   - Abstract the programmer from the several available discovery (Bluetooth SDP or wireless UPnP discovery) and interaction mechanisms (RPC or publish/subscribe).
   - Allow this middleware to easily adapt to newly emerging discovery (e.g. RFID identification) and interactions means
4. Make use of commonly available hardware and software features in mobile devices
5. Generate software representatives (proxies) of smart objects which can be run in any platform
   - **"write once run in any device type"** philosophy

EMI²lets Deployment

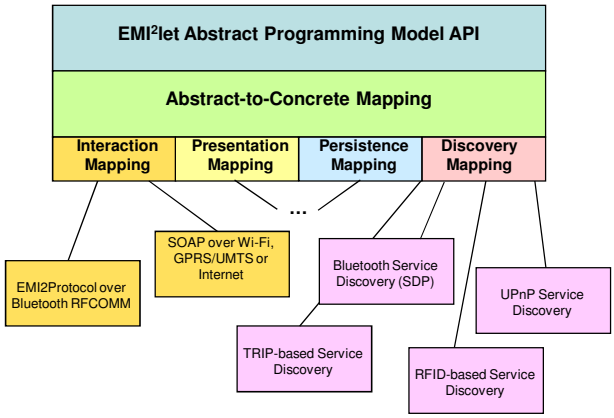


EMI²lets Platform Deliverables

- The EMI²lets platform consists of the following elements:
  - A programming framework
  - An integrated development environment, named EMI²let Designer
  - A runtime environment installed on EMI²let-aware devices for executing downloaded code.
  - An EMI²let Player to discover, download, verify and control the execution life of a downloaded EMI²let.
  - An EMI²let Server which acts as repository of EMI²lets and as running environment of EMI²lets server-sides



EMI²lets Internal Architecture



EMI² Internals

- 3-tier software architecture
- EMI² framework defines **4 programming abstractions**:
  - Discovery
  - Communication
  - Presentation
  - Persistency
- An **EMI2let plug-in** = abstraction implementation
  - Common plug-ins: Bluetooth, Wi-Fi, UPnP
  - Special purpose: TRIP (Target Recognition using Image Processing)
- **Assembly fusion** at runtime
  - Reflection does the magic!!!

EMI²lets Implementation

- Reflection is used to verify that the code arriving as part of an EMI2let complies with the EMI2lets framework and can be trusted.
- The `EMI2let` base class defines a set of methods that rule the life cycle of an EMI2let:
  - `Start, Pause, Destroy, NotifyDisconnected`
- ... and its metadata:
  - `GetUUDI`
  - `SetProperty/GetProperty`
  - `GetAddresses`
- The binary code downloaded is linked dynamically (late bound) with the runtime installed in the target device
- Our first implementation has been done in .NET available for PC, PDA and Mobile Phone

EMI²lets Applications

- We have created EMI2lets for different application domains:
  - Accessibility: blind (bus stop), deaf (conference)
  - Home/office automation: comfort (lights), entertainment (WMP), surveillance (camera)
  - Industry: robot
  - Public spaces: restaurant, parking, airport

EMI²lets Applications

Conclusion

- EMI²lets = middleware providing universal active influence to mobile devices over Smart Objects:
  - Transforms mobile devices into universal remote controllers.
  - Enables both local and global access to those Smart Objects (anywhere/anytime).
  - Independent and extensible to the underlying service discovery and interaction, graphical representation and persistence mechanisms.
  - Enables AmI spaces using conventional readily-available hardware and software.
  - Follows a “write once run in any device type” philosophy
- EMI²lets won the Spanish Imagine Cup 2005

References

- OSGi
  - OSGi Alliance “About the OSGi Service Platform”
    - http://www.osgi.org/documents/osgi_technology/osgi-sp-overview.pdf
- UPnP Forum
  - UPnP Forum, “Understanding UPnP: A White Paper”
    - http://www.upnp.org/resources/whitepapers.asp
- SOUPA
  - Semantic Web in UbiComp
    - http://pervasive.semanticweb.org/
- Task Computing
  - Fujitsu, http://taskcomputing.org/
- Autonomic Computing
  - IBM – creating self-managing computing systems
    - http://www-03.ibm.com/autonomic/
- CoBrA – Context Broker Architecture -- http://cobra.umbc.edu/
- EMI2lets: a Reflective Framework for Enabling AmI, Diego López de Ipiña, Juan Ignacio Vázquez, Daniel García, Javier Fernández, Iván García, David Sainz and Aitor Almeida, Journal of Universal Computer Science (J.UCS), vol. 12, no. 3, pp. 297-314, March 2006
  - http://paginaspersonales.deusto.es/dipina/#publications

Universidad de Deusto
Deustuko Unibertsitatea

mobility research lab

4.4. Escenarios de Aplicación: Laboratorio, Hogar, ITS

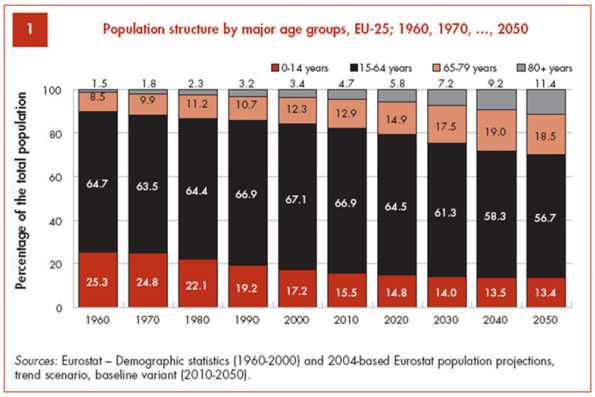
Dr. Diego Lz. de Ipiña Gz. de Artaza
http://paginaspesonales.deusto.es/dipina
http://www.morelab.deusto.es
http://www.ctmd.deusto.es

Infrastructural Support for Ambient Assisted Living

Universidad de Deusto
Deustuko Unibertsitatea

- Motivation
- The ZAINGUNE project
- Platform architecture
- Multi-modal human-environment interaction
- Rule-based environment reasoning
- Conclusion and future work

Motivation



Motivation

- Some facts:
  - **By 2020, 25%** of the EU's population will be **over 65**
  - **Spending** on pensions, health and long-term care is expected **to increase by 4-8% of GDP in coming decades**
    - Total expenditures tripling by 2050
  - **Older Europeans are important consumers** with a wealth over €3000 billion
- **Ambient Assisted Living (AAL)** is a European Union initiative to address the needs of the ageing European population
  - Elderly people should be able of living longer in their preferred environments, to enhance the quality of their lives
  - Costs for society and public health systems should be reduced
    - **http://www.aal-europe.eu/**
- **To make AAL reality** → important to devise new easily-deployable middleware and hardware infrastructure

The ZAINGUNE Project

Universidad de Deusto
Deustuko Unibertsitatea

- **Aims to provide the software/hardware infrastructure (platform) required to easily deploy assistive services for elderly people @home**
  - **http://www.tecnologico.deusto.es/projects/zaingune**
- **HOW?**
  - With an **OSGi gateway powered by a rule-based reasoning engine** which allows the coordination and cooperation of the home sensing and actuation devices

- Consortium composed by:

Tecnológico Fundación Deusto Teknologikoa Deustu Fundazioa

ECDOA

visesa

irontec Internet y Sistemas sobre GNU/Linux

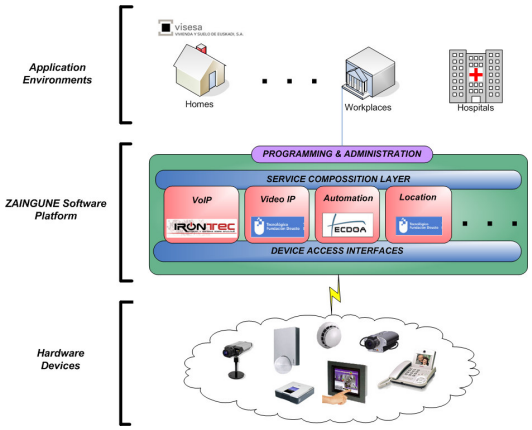
ZAINGUNE Goals

Universidad de Deusto
Deustuko Unibertsitatea

- **Heterogeneous device support:**
  - Agotek's gerontek, Asterisk IP phones, IP cameras, KNX-EIB devices, ...
- Model **assistive environments** as a set of **cooperating services**
- **Programmability through** a **SOA**-based approach.
- **Apply natural explicit interaction mechanisms**:
  - Easy to use gadget-based and secure front-end, phone-mediated interaction, ...

MORE LAB mobility research lab Infrastructural Support for Ambient Assisted Living

ZAINGUNE Multi-layered Architecture



ZAINGUNE Multi-layered Architecture

1. The **hardware device layer** is composed of the sensors and actuators which populate an environment
2. The **software platform layer** transforms the functionality provided by the devices managed by Layer 1 into software services (**bundles**) which can be combined to give place to more advanced services
   - Every device within an AAL environment (home, residence, hospital) is encapsulated as a bundle or execution unit within our OSGi environment.
   - It includes two **core bundles**:
     - **ZainguneController** – core component of ZAINGUNE server, manages and controls access to the components (OSGi bundles) supported by ZAINGUNE.
     - **ZainguneServlet** – behaves as an **Web Service/OSGi gateway** exporting the OSGi bundle functionality through Web Services and **generates web front-ends** (based on JavaScript X library) of every bundle
3. The **applications environment layer** includes all the possible application scenarios for the ZAINGUNE infrastructure
   - **Public housing flat for disabled or elderly people**, hospitals, residences and so on

ZAINGUNE Platform Deliverables

Universidad de Deusto
Deustuko Unibertsitatea

- **Software**
  1. An **OSGi-based extensible environment controlling server**
  2. A **rule engine embodying the intelligence (implicit reactivity) offered by an environment** instrumented by our infrastructure.
     - Allows for environment reactive configuration as a set of IF-THEN rules.
  3. An **advanced web-gadget-based control dashboard** which allows for **explicit service triggering** both **locally** (in-house) or **remotely** (family, caregivers).
     - A subset of the functionality is also accessible through voice commands and phone keystrokes based on Asterisk VoIP technology.
- **Hardware**
  1. A **custom-built alert bracelet through which emergency** help can be sought or relevant messages can be presented to the patient.

Multi-modal Environment Interaction

Universidad de Deusto
Deustuko Unibertsitatea

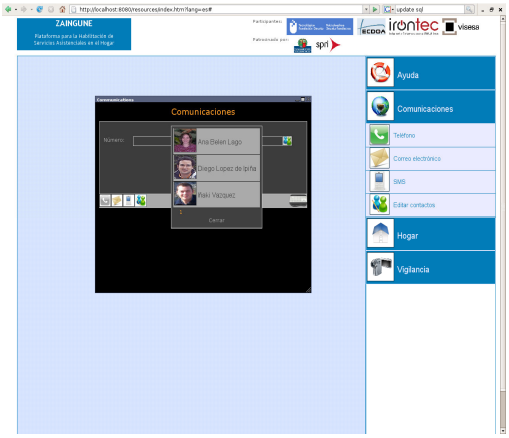
- **Web gadget-based interaction** – an easy to use web gadget-based environment controller divided into the following sections:
  - **Help** – single button to request help
  - **Communications** – call by photo, email and SMS
  - **Home control** – control of every device by container
  - **Surveillance** – both local and remote IP camera control
- **Phone touchpad- and voice-based interaction** – the integration of Asterisk in ZAINGUNE provides:
  - feedback through phone speakers,
  - house control through keystrokes and
  - voice commands
- **Alert bracelet-based interaction** – special purpose device designed for assistance seeking and alert notification

Multi-modal Environment Interaction

Universidad de Deusto
Deustuko Unibertsitatea

Multi-modal Environment Interaction

Universidad de Deusto
Deustuko Unibertsitatea



mobility research lab
Infrastructural Support for Ambient Assisted Living

ZAINGUNE Alert Bracelet

Universidad de Deusto
Deustuko Unibertsitatea

- A **custom-built device combining** an **organic screen** (µOLED-96-G1 of 4D Systems) with a **WSN mote** based on Mica2DOT capable of displaying messages broadcasted by nearby motes.
- Every inhabitant may carry an alert bracelet for:
  - **Assistance seeking**
  - **Alert notification**
- A future work option is to add living signal monitoring sensors (e.g. Nonin 4100 Avant Module) to such device

Intelligence through Rule-Based Reasoning

Universidad de Deusto
Deustuko Unibertsitatea

- The adoption of a **rule-based engine** in ZAINGUNE offers two main advantages:
  - **Decouples environment configuration from programmability**
  - **Enables environment-initiated proactive reactions**
- **Environment intelligence is encapsulated as a set of rules which trigger when certain sensorial situations are matched**
  - LHS represents sensing conditions whilst RHS depicts actions to be undertaken when the LHS situations are matched
- This rule-based paradigm is employed to configure the reactive behaviour of a ZAINGUNE-controlled environment:
  - efficient management of energy resources
  - security at home or
  - danger situation prevention

JBoss™ Drools

MORE LAB mobility research lab Infrastructural Support for Ambient Assisted Living

Rule-reasoning Example

```
rule "Flooding Event"
   no-loop true
   activation-group "eib"
   salience 10
   when
      event: ZainguneEvent()
      houseInfo: ZainguneHouseInfo()
      eventSender: EventSender()
      eval(event.getTopic().equals("es/deusto/tecnologico/osgi/eventadmin/eib/VALUE_CHANGE"))
      eval(houseInfo.checkDeviceType((String)event.getProperty("name"), "FloodingSensor"))
      eval("On".equals((String)event.getProperty("newValue")))
   then
      String topic = "es/deusto/tecnologico/zaingune/emergency/send";
      Hashtable<String, String> properties = new Hashtable<String, String>();
      ZainguneDeviceInfo deviceInfo =
    houseInfo.getDeviceInfoByName((String)event.getProperty("name"));
      ZainguneRoomInfo roomInfo = houseInfo.getDevicesRoom(deviceInfo.getName());
      properties.put("location", roomInfo.getName());
      properties.put("emergency_type","flooding");
      properties.put("message", "Flooding in room " + roomInfo.getName());
      Event outputEvent = createEvent(topic,properties);
      eventSender.sendEvent(outputEvent);
      retract(event);
end
```


Conclusions

- ZAINGUNE provides several **easily-deployable ICT infrastructure contributions for their progressive adoption** at elderly people's homes
- Our **main outcome** is an **OSGi platform powered by a rule-based reasoning engine which integrates** a **KNX/EIB** automation bus, **VoIP** and **VideoIP** infrastructure to configure more aware and reactive homes.
- An **assortment of multi-modal explicit interaction mechanisms** to request services from the environment have been shown: touch screen-based web gadget-based dashboard, an alert bracelet or VoIP phone-mediated interaction

Future Work

- Development of a new low-cost easily deployable indoor location system based on WSN technology
- Development of new vital sign/location monitoring bracelets
- Creation of a wizard through which ZAINGUNE can be parameterised for a new environment
- The integration of workflow mechanisms to assemble complex advanced services from basic ones without requiring code modification and
- **The evaluation of our infrastructure in a real home deployment and its extension to multi-user residence environments.**

ZAINGUNE Website

Acknowledgements

Infrastructural Support for Ambient Assisted Living
http://www.tecnologico.deusto.es/projects/zaingune
Dr. Diego López-de-Ipiña, Xabier Laiseca, Ander Barbier, Unai Aguilera, Aitor Almeida, Pablo Orduña and Dr. Juan Ignacio Vazquez
dipina@eside.deusto.es

Thanks to the Industry, Commerce and Tourism Department of Basque Government for sponsoring this work through grant IG-2007/00211

ZAINGUNE Project site: http://www.tecnologico.deusto.es/projects/zaingune

Semantically Dynamic Infrastructure for Intelligent Environments

Dr. Diego López-de-Ipiña, Aitor Almeida, Unai Aguilera, Iker Larizgoitia, Xabier Laiseca, Pablo Orduña, Ander Barbier, Juan Ignacio Vázquez

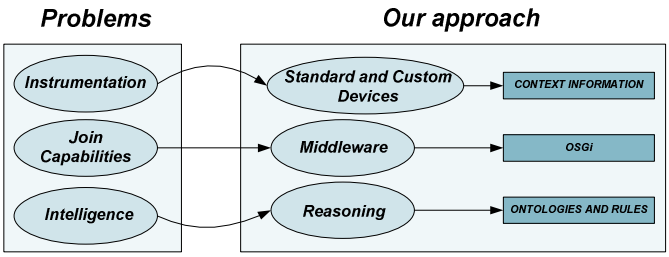
http://www.tecnologico.deusto.es/projects/smartlab/

Outline

- Motivation and Objectives
- System architecture
  - Layers
  - Examples
- Conclusions
- Questions

Motivation and Objectives

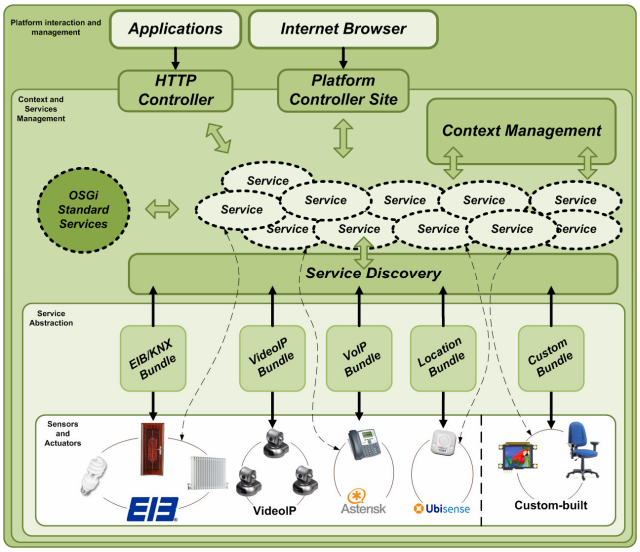
- **Middleware support for intelligent environment provision in AAL:**
  - Monitoring context
  - Determine the user activity and high level context
  - Adapt the environment to the user



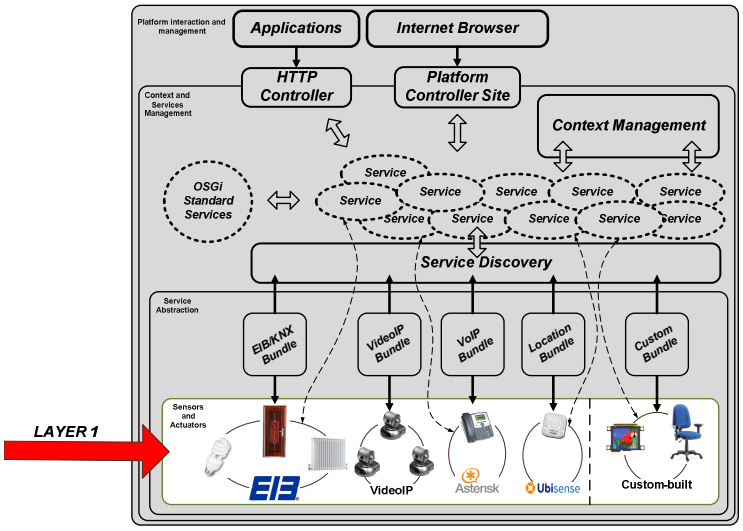
OSGi Review

- *OSGi technology is a Java-based Universal Middleware which provides a service-oriented, component-based environment for developers and offers standardized ways to manage the software lifecycle.*
- It provides standard services that can be used to develop context-aware applications:
  - Event services
  - Configuration services
  - Components Lifecycle automatic management

SmartLab Architecture



Layer 1: Sensors and Actuators



Layer 1: Sensors and Actuators

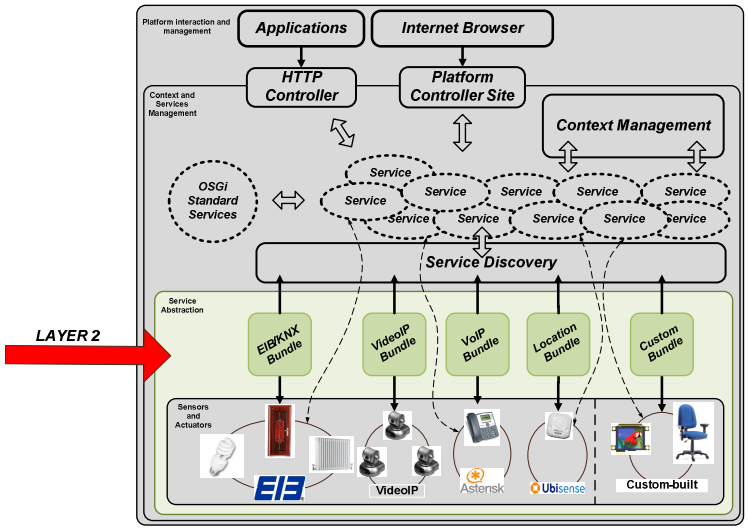
- Device assortment common in home and caring environments:
  - EIB/KNX
  - Asterisk VoIP
  - VideoIP Cameras
  - Indoor Location System (Ubisense)
  - People wandering devices (Gerontek)
  - Custom-built Devices (WSN)
    - Chair
    - Display bracelet
    - Container
- Every system has its own control interface
  - How do we interconnect all of them?







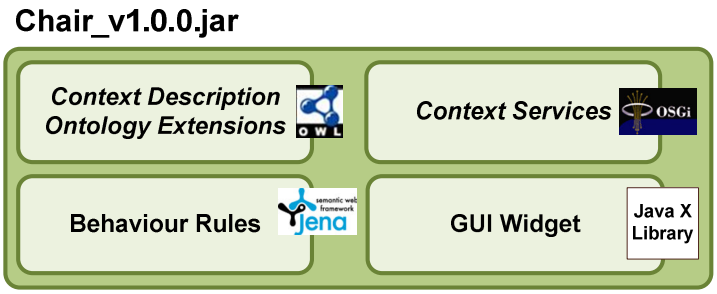

Layer 2: Service Abstraction



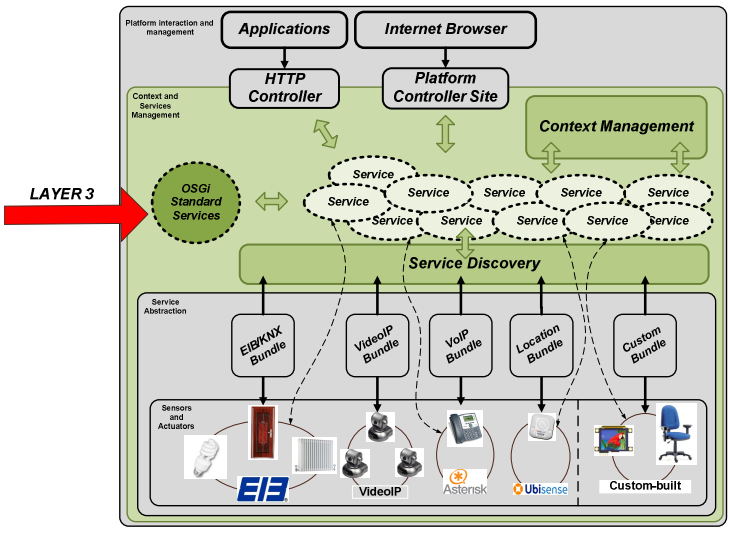
Layer 2: Service Abstraction

- Every device or system provides certain functionalities that we must transform into software services inside OSGi.
  - Each device must *provide* a control *bundle* acting as a proxy inside the OSGi platform.
  - All the native services of each device are wrapped in OSGi services.
    - *EIB/KNX Bus → BinaryLight, DimmableLight, Alarm, DoorSensor*
    - *VideoIP HTTP Cameras → CameraController*
    - *VoiceIP Asterisk Server → AsteriskController*
    - *Gerontek Server → GerontekController*
    - *Ubisense COM Server → UbisenseController*
    - *Custom-Built Devices → SmartChair, SmartContainer*

Layer 2: Bundles' anatomy



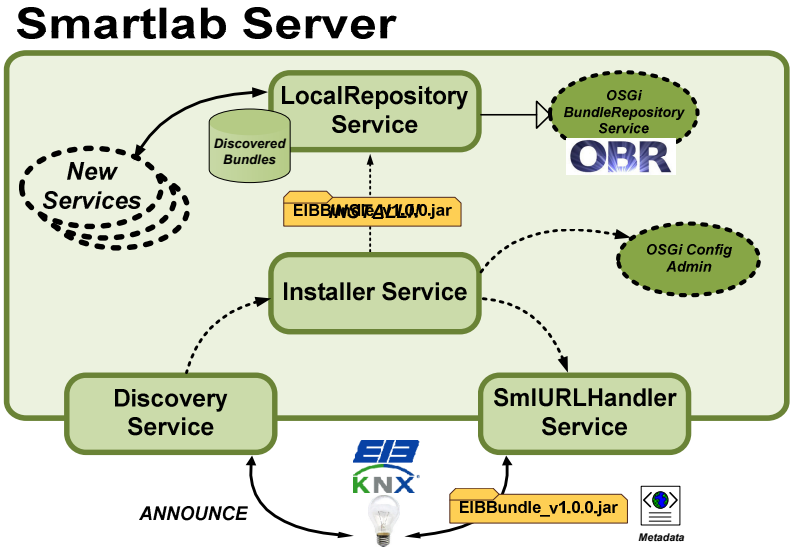
Layer 3: Service and Context Management



Layer 3: Service Management

- Discovery service
  - Simple multicast protocol to obtain the bundles automatically.
- Installer service
  - Decides whether the bundle should be installed or not.
- Local Repository service
  - Extends the OBR service to provide a local cache for the discovered bundles.

Example: Service Management



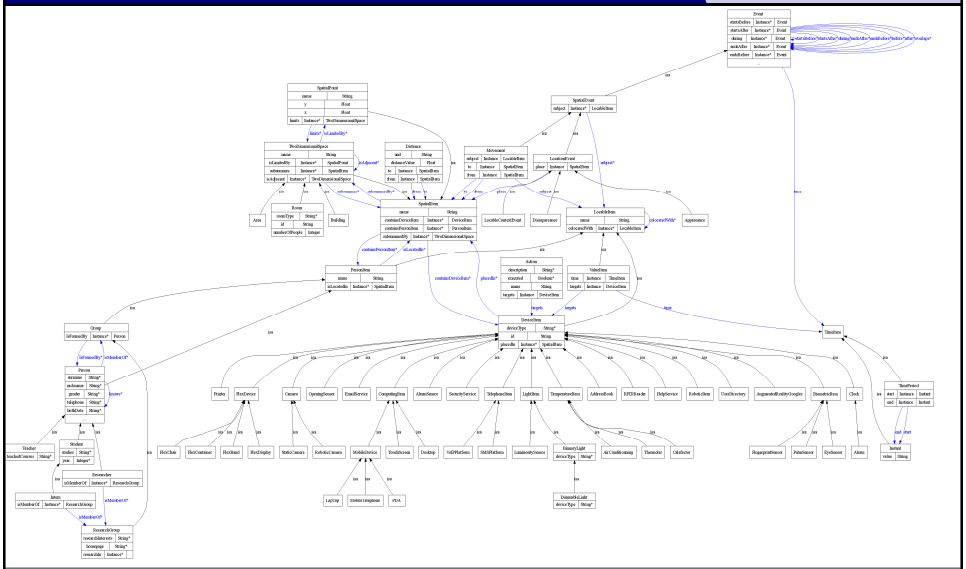
Layer 3: Context Management

Universidad de Deusto
Deustuko Unibertsitatea

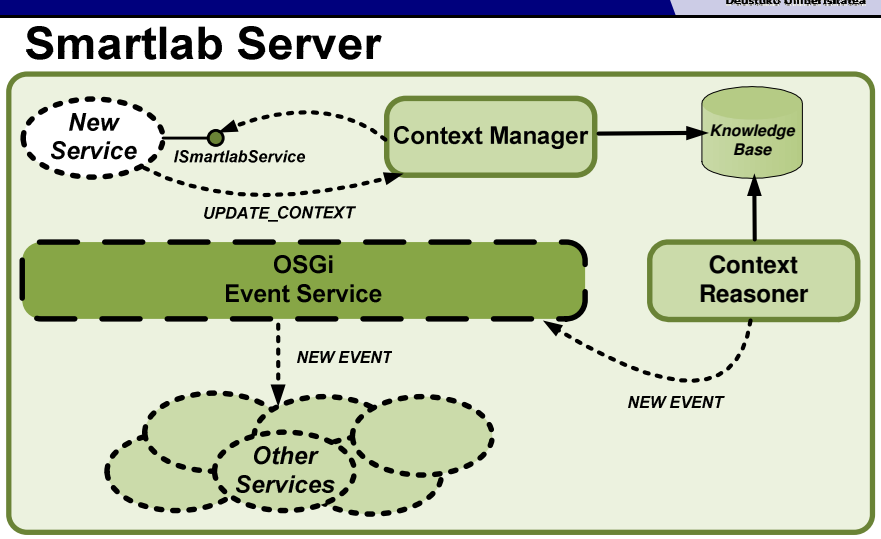
- Context information modelled with an ontology
  - Base core
  - Time and space relations
  - Events
- New services might extend the knowledge base
  - Classes and instances
  - Behaviour rules
- Convert inferred information in OSGi events to which the different services can register.
  - React accordingly to specific events.

Layer 3: Ontology

Universidad de Deusto
Deustuko Unibertsitatea



Example: Context Management



Context Management

- Two knowledge generation methods in SmartLab:
  - **Ontological reasoning**
    - Makes use of RDF (rdf:domain), RFS (rdfs:subPropertyOf) y OWL (owl:TransitiveProperty) predicates
    - Allows to infer implicit knowledge
  - **Rule-based reasoning**
    - Allows defining relationship among entities in ontology
- Three types of inference:
  - **Semantic rules** – enable making ontological reasoning based on RDF and OWL theoretical models
  - **Knowledge extraction rules** – extract new knowledge from ontology's implicit one
  - **Event-inferring rules** – generate aggregated events from the context in the knowledge base

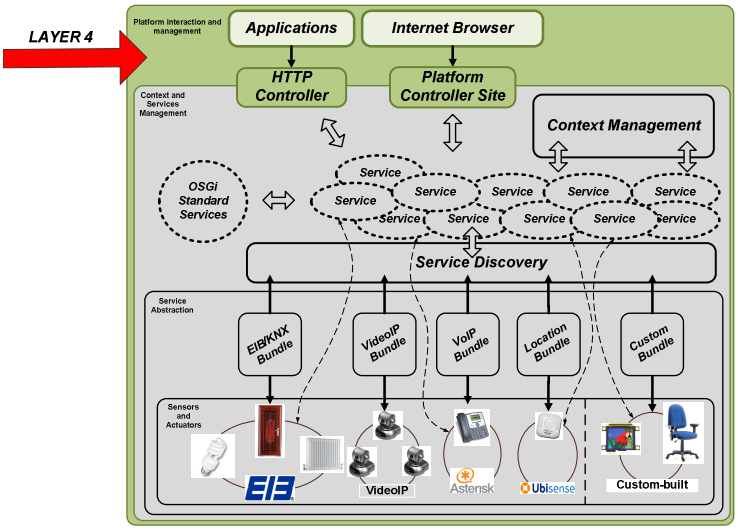
Event-inferring Rule Example

```
[EVENT-Meeting:
(?meetingArea rdf:type
   <http://deusto.es/smartlab.owl#MeetingArea>),
(?meetingArea <http://deusto.es/smartlab.owl#containsPersonItem>
   ?p1),
(?meetingArea <http://deusto.es/smartlab.owl#containsPersonItem>
   ?p2),
 (?p1 <http://deusto.es/smartlab.owl#name> ?name1),
(?p2 <http://deusto.es/smartlab.owl#name> ?name2),
notEqual(?name1, ?name2),
makeTemp(?meetingEvent)
->
(?meetingEvent
   rdf:type<http://deusto.es/smartlab.owl#LocableMeetingEvent>),
(?meetingEvent <http://deusto.es/smartlab.owl#place> ?meetingArea),
(?meetingEvent <http://deusto.es/smartlab.owl#name> ?name1),
(?meetingEvent <http://deusto.es/smartlab.owl#name> ?name2)
]
```


Performance Results

	5 Devices	10 Devices	20 Devices	50 Devices	100 Devices
Average Inference Time Jena (ms)	1854	2620	3671	9944	27827
Average Inference Time SWRLTab+Jess (ms)	922	1063	1391	3953	12844
Average triples Jena	866	965	1164	1656	2329
Average triples SWRLTab+Jess	3259	3464	3874	5104	7154
Average classes Jena	112	123	143	203	303
Average classes SWRLTab+Jess	127	127	127	127	127
Average individuals Jena	66	84	121	194	269
Average individuals SWRLTab+Jess	83	103	143	263	463
Average rules Jena & SWRLTab+Jess	29	35	45	75	125

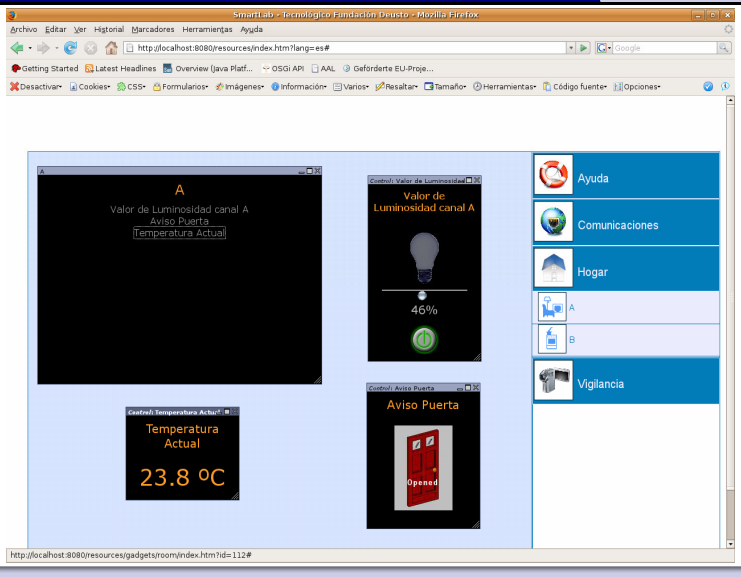
Layer 4: Service programmability, Management and Interaction



Layer 4: Service programmability, Management and Interaction

- Implicit interaction
  - Context management generates events and some services are invoked automatically.
- Explicit interaction
  - HTTP interface inside OSGi to invoke any service that exposes remote methods
  - Dashboard-like GUI based on widgets (javascript cross-browser library) that are loaded when the services are active.

Control GUI



Conclusions

- Several extensions to the OSGi framework to support intelligent environment instrumentation have been presented:
  - Devices or environment services expose a special control bundle.
  - OSGi bundles are discovered and act as a proxy providing semantic enhanced services.
  - These services populate the system with new context information in order to infer new knowledge and generate events.
  - Different services can register to receive context events and react to them accordingly.
  - The platform knowledge has been modelled using ontologies and rules that can be extended and updated dynamically.
  - For explicit interaction we have a HTTP interface or a Dashboard GUI based on widgets that can be used to interact with the platform.
- We expect to deploy this infrastructure in AAL environments
  - Preventive care, security and safety, environment management, telemonitoring

Intelligent Transport Systems

Universidad de Deusto
Deustuko Unibertsitatea

- Compared with other Smart Spaces such as meeting rooms and houses, the automobile's interior is mobile, small, must frequently communicate with the outer environment
  - A car is a kind of mobile sensor platform
- Why?
  - traffic congestion worldwide
  - synergy of new information technologies for simulation, real-time control and communications networks
- How can technology help reducing transportation's economic and environmental impact?
  - Traffic estimation and prediction systems
- Services offered:
  - Dynamic traffic management
  - Variable message signs
  - Ramp metering
  - Dynamic speed signals over motorways and warning messages for extreme weather conditions

Benefits of ITS

Universidad de Deusto
Deustuko Unibertsitatea

- Increased road network capacity
- Reduced congestion and pollution
- Shorter and more predictable journey times
- Improved traffic safety for all road users
- Lower vehicle operating costs
- More efficient logistics
- Improved management and control of the road network (both urban and inter-urban)
- Increased efficiency of the public transport systems
- Better and more efficient response to hazards, incidents and accidents
- Remote maintenance services

Applications of ITS

- Aim: more efficient, safe and environmentally friendly use of the road network:
  - Urban congestion reduction, intelligent traffic management service undertakes decisions
  - Onboard displays can receive warnings from this system
  - Freight and fleet applications

ITS Service Topologies

- Inter-vehicle Services: messages regarding in-vehicle sensors for accident warning, info on traffic jams, direct collision warning
- Services of Road Side Units (RSU) communication to increase safety
- Portal-based services, server-based infrastructure providing services such as parking or hotels.

Universidad de Deusto
Deustuko Unibertsitatea

mobility research lab

4.5. Practical Case: Visual Sensing and Middleware Support for Sentient Computing

Dr. Diego Lz. de Ipiña Gz. de Artaza
http://paginaspesonales.deusto.es/dipina
http://www.morelab.deusto.es
http://www.ctmd.deusto.es

Introduction

Universidad de Deusto
Deustuko Unibertsitatea

Goals:

- build **Sentient Spaces** = *computerised* environments that *sense* & *react*
- close gap between user and computer by using **context**
- make **ubiquitous computing** reality through **Sentient Computing**

Sentient Computing = **computers + sensors + rules**:

- distributed sensors capture context, e.g. temperature, identity, **location**, etc
- rules model how computers **react** to the stimuli provided by sensors
- 3 phases: (1) context capture, (2) context interpretation and (3) action triggering

To make viable widespread adoption of Sentient Computing, we propose:

- location sensor deployable everywhere and for everyone
- middleware support for easier sentient application development:
  - rule-based monitoring of contextual events and associated reactions
  - user-bound service lifecycle control to assist in action triggering

Laboratory for Communications Engineering (LCE)
Cambridge University Engineering Department
England, UK

AT&T Laboratories
Cambridge

Basque Government
Education Department

TRIP: a Vision-based Location Sensor

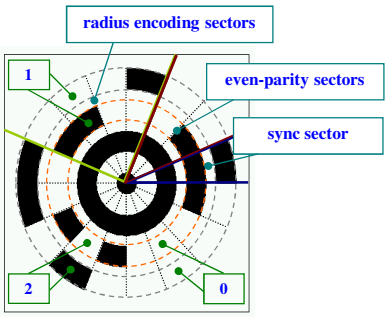
Universidad de Deusto
Deustuko Unibertsitatea

"Develop an easily-deployable location sensor technology with minimum hardware requirements and a low price"

- TRIP (Target Recognition using Image Processing):
  - identifies and locates tagged objects in the field of view of a camera
- Requires:
  - off-the-shelf technology: cameras+PC+printer
  - specially designed 2-D circular markers
  - use of well-known Image Processing and Computer Vision algorithms
- Cheap, easily deployable can tag everything:
  - e.g. people, computers, books, stapler, etc
- Provides accurate 3-D pose of objects within 3 cm and 2° error

TRIPcode 2-D Marker

Universidad de Deusto
Deustuko Unibertsitatea



* 10 2011 221210001
TRIPcode of radius 58mm and ID 18,795

- 2-D barcode with ternary code
- Easy to identify bull's-eye:
  - invariant with respect to:
    - Rotation
    - Perspective
  - high contrast
- 2 16 bit code encoding rings:
  - 1 sector synchronisation
  - 2 for even parity checking
  - 4 for bull's-eye radius encoding
  - 3^9 = 19,683 valid codes

Target Recognition Process

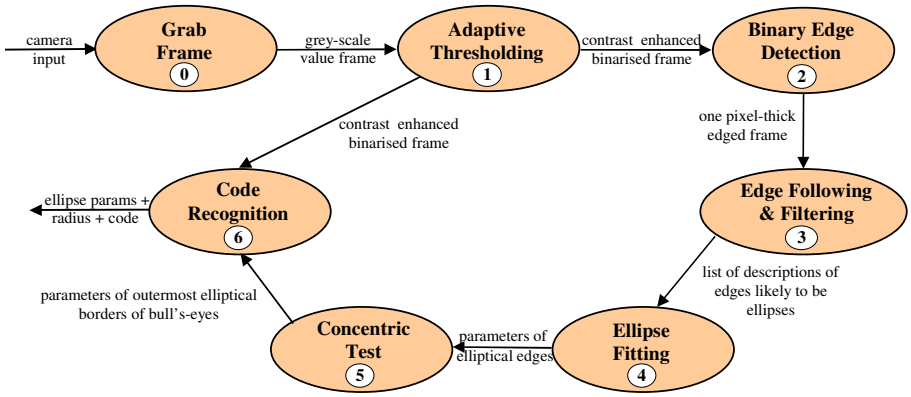


Image Acquisiton (Stage 0)

- A digital image is represented by a numerical matrix, *I*, with *n* rows and *m* columns, where $p_n=[x,y]$ denotes the image value (brightness) at pixel (*x*, *y*) (*x*-th column and *y*-th row), and encodes the intensity recorded by the photo-sensors of the CCD array contributing to that pixel.
  - TRIP makes exclusive use of monochromatic images, where p_n represents a greyscale value in the range 0 (black) to 255 (white)

Adaptive Thresholding (Stage 1)

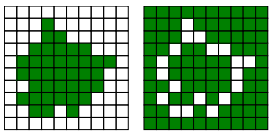
Universidad de Deusto
Deustuko Unibertsitatea

- Image segmentation technique that aims to separate objects from the background
- Removes the effects of uneven lighting, glints and shadowing in the field of view and increases the overall contrast of the image
- Wellner's adaptive thresholding method varies the threshold value used to transform a pixel into black or white value based on the background illumination of that pixel

Binary Edge Detection (Stage 1)

Universidad de Deusto
Deustuko Unibertsitatea

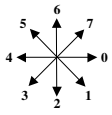
- Edges correspond to the pixels at or around which the image values undergo a sharp variation.
  - Edge detection techniques apply the geometric interpretation of the *gradient* (the measure of change in a function) to an image, expressed as the rate of change of the grey levels in it.
- *Heuristic applied:*
  - *"A pixel in a binary image is an edge point if it has black intensity value and a 4-connected neighbour pixel with white intensity value and an 8-connected one with black intensity value"*

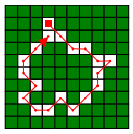


Edge Following & Filtering (Stage 3)

Universidad de Deusto
Deustuko Unibertsitatea

- All the 8-connected chains of edgels previously located are followed in a clockwise fashion, producing for each edge a list of ordered point locations
  - The circular bull's-eyes of TRIPtags are observed in the captured frame as elliptical due to perspective projection
    - This stage applies a filtering process to every edge tracked, retaining only the ones plausibly belonging to an ellipse





Ellipse Detection (Stage 4)

Universidad de Deusto
Deustuko Unibertsitatea

- An ellipse detection method is applied which seeks for each of these edges a conic function representing an ellipse, given by the following implicit equation
  - Fitzgibbon A.W. and Fisher R.B. "A Buyer's Guide to Conic Fitting". Proceedings of the 5th British Machine Vision Conference, pp. 513-522, 1995.
- Goal: minimize the Euclidean distance between points belonging to the ellipse, *p*, and the measured points, *pi*, in the image

$$\min_{a} \sum_{i=1}^{N} \left\| p - p_i \right\|^2$$

Concentric Test (Stage 5)

Universidad de Deusto
Deustuko Unibertsitatea

- The ellipse parameters obtained in the previous stage are compared to identify concentric ellipses likely to form part of the projection of candidate targets' bull's-eyes.

Code Deciphering (Stage 6)

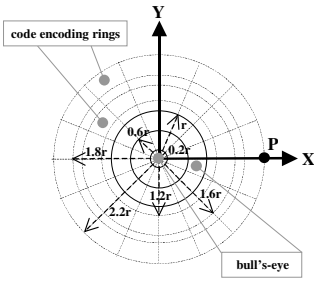
Universidad de Deusto
Deustuko Unibertsitatea

- A code extraction stage is applied to the candidate TRIP sightings.
  - Applies an efficient pixel-sampling mechanism on the binary image result of the Adaptive Thresholding stage, based on the parameters of the outermost ellipse of the projection of a candidate TRIPcode
- The outcome of the target recognition process is a list of TRIP sighting descriptions containing:
  - A target's ternary identifier
  - The radius of its central bull's-eye's outermost circular border
  - The geometric properties of this border's elliptical projection in the image (x1, y1, a b, and θ)
  - The six parameters of the conic corresponding to the bull's-eye's outermost circular border's elliptical projection and
  - The location in the image of the bottom outermost corner of the synchronisation sector of a TRIPtag.

Target Pose Extraction (Stage 7)

Universidad de Deusto
Deustuko Unibertsitatea

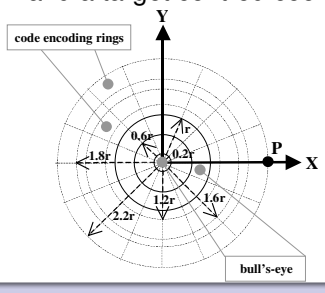
- Applies a model-based object location technique
- The 'TRIPtag object model' (see Fig. 3) is composed of:
  - the outermost circular border of the central bull's-eye and
  - a point outside the bull's-eye (denoted by *P* in), lying on the *X*-axis of the target centred coordinate system. This point corresponds to the bottom outermost corner of the synchronisation sector of a TRIPtag



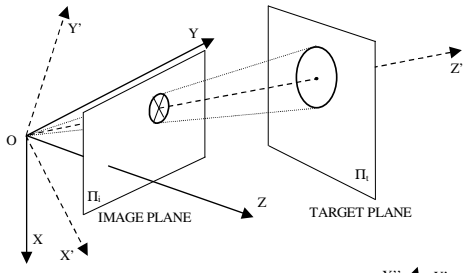
Target Pose Extraction (Stage 7)

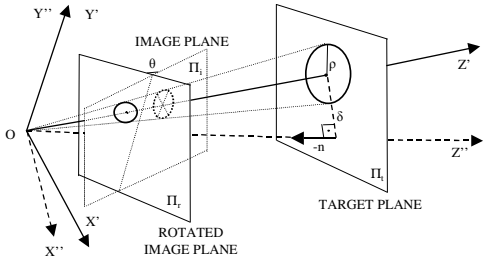
Universidad de Deusto
Deustuko Unibertsitatea

- The POSE_FROM_TRIPTAG method establishes a transformation that back-projects the elliptical projection of the outermost border of a target into its actual circular form, of known radius, in the target plane.
  - Returns the translation vector *[Tx Ty Tz]* and rotation angles *(α, β, γ)* that define the rigid body transformation between a camera coordinate system and a target centred coordinate system.



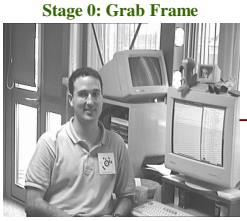
Geometry POSE_FROM_TRIPTAG method



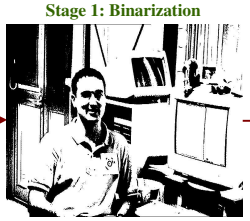


Target Recognition Process

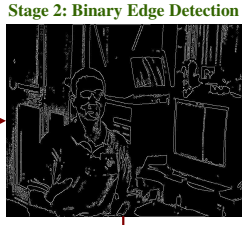
Stage 0: Grab Frame



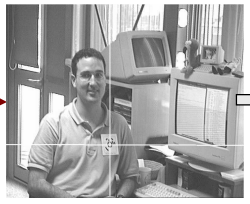
Stage 1: Binarization



Stage 2: Binary Edge Detection



Stage 3: Edge Following & Filtering



Ellipse params:
x (335.432), y (416.361) pixel coords
a (8.9977), b (7.47734) pixel coords
θ (15.91) degrees

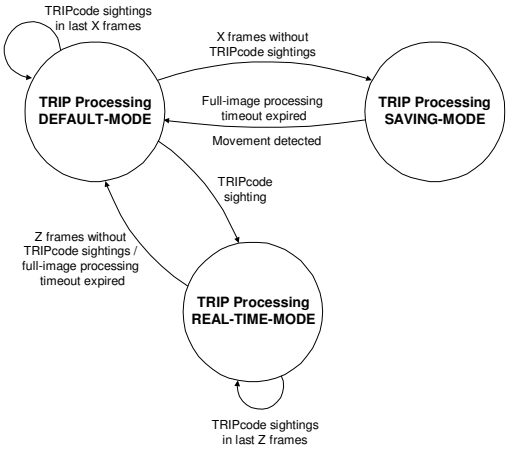
Bull's-eye radius: 0120 (15 mm)
TRIPcode: 002200000 (1,944)

Translation Vector (meters):
(T_x=0.0329608, T_y=0.043217, T_z=3.06935)

Target Plane Orientation angles (degrees): (α=-7.9175, β=-32.1995, γ=-8.45592)

Stages 4-7: Ellipse Fitting, Ellipse Concentricity Test, Code Deciphering and POSE_FROM_TRIPTAG method

TRIP Sensor Operation Modes

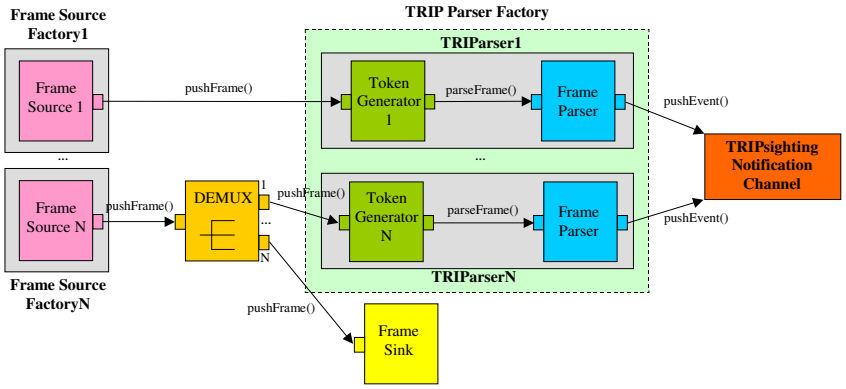


TRIP: a Distributed Sensor System (I)

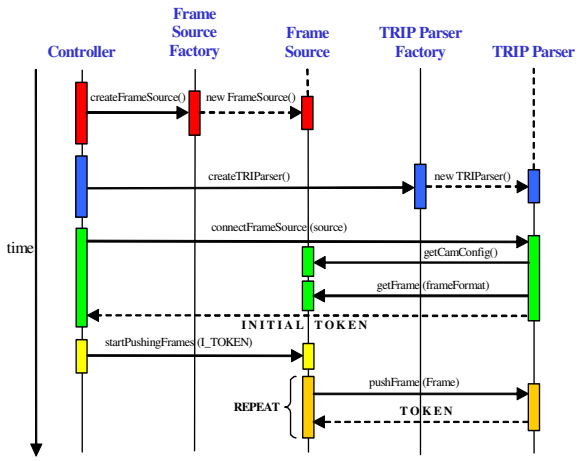
- TRIP C++ library and TRIP Directory Service
- Java package wrapping the TRIP C++ library
- CORBA-based TRIParser component:
  - accepts video frames from distributed frame sources
  - provides synchronous and asynchronous interfaces for video parsing
  - frame sources push frames using a token-based protocol for image flow control
  - pushes a TRIPevent per target sighting into a CORBA Notification Channel:

```
struct TRIPevent {
   double timestamp;
   unsigned long cameraID;
   string TRIPcode; // code ternary representation
   TargetPosition position; // (xpos, ypos, zpos) vector
   TargetOrientation orientation; // (α, β, γ) angles
};
```

TRIP: a Distributed Sensor System (II)



TRIP: a Distributed Sensor System (III)



A Rule Paradigm for Sentient Computing

Universidad de Deusto
Deustuko Unibertsitatea

- Sentient systems are reactive systems that perform actions in response to contextual events
  - Respond to the stimuli provided by distributed sensors by triggering actions to satisfy the user's expectations based on their current context, *e.g.* their identity, location or current activity
- Issues:
  - Development of even simple sentient application usually involves the correlation of inputs provided from diverse context sources
- Observation:
  - *Modus operandi* of sentient applications follows a common pattern:
    - Wait until a pre-defined *situation* (a composite event pattern) is matched to trigger an *action* → *Event-Condition-Action* model

ECA Rule Matching Engine

Universidad de Deusto
Deustuko Unibertsitatea

- Sentient Applications respond to an Event-Condition-Action (ECA) model:
  - monitor contextual events coming from diverse sources
  - correlate events to determine when a contextual situation occurs:
    - e.g. IF two or more people in meeting room + sound level high THEN meeting on
  - ineffective to force every app to handle same behaviour separately
- Solution → ECA Rule Matching Service:
  - accepts rules specified by the user in the ECA language

    `<rule> ::= {<event-pattern-list> => <action-list> }`
  - automatically registers with the necessary event sources
  - notifies clients with aggregated or composite events or executes actions when rules fire:
    - *aggregated event* = new event summarizing a situation
    - *composite event* = batch of events corresponding to a situation

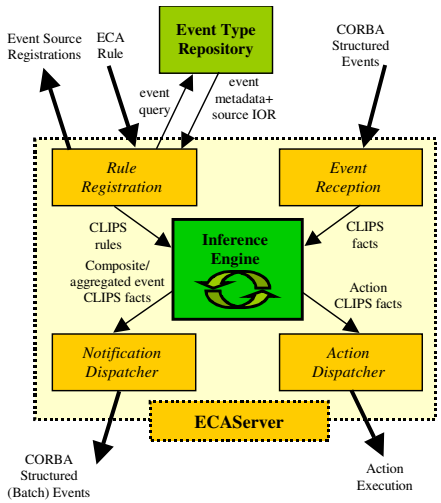
A Language for Specifying ECA Rules

Universidad de Deusto
Deustuko Unibertsitatea

```
<rule> ::= within INTEGER <pattern_list> => <action_list> |
   <pattern_list> => <action_list>
<pattern_list> ::= <pattern_list> <connective> <pattern_list> |
   (<pattern_list>) | not <pattern_list | test(<expr>) |
   <event_pattern>
<event_pattern> ::= <EVENT_ID>(<name_value_list>) |
   <VARIABLE>:<event_pattern) | query <event_pattern>
<connective> ::= and | or | before | after
<name_value_list> ::= <name_value> |
   <name_value_list>, <name_value>
<name_value> ::= (<name_value>) | <CLASS_ID> <VARIABLE> |
   <CLASS_ID> <DATA_VALUE>
<expr> ::= <expr> and <expr> | <expr> or <expr> | not <expr> |
   <expr> <relation_op> <expr> | <expr> ~ <expr> |
   <expr> <arith_op> <expr> | - <expr> | (<expr>) |
   <DATA_VALUE> | <VARIABLE> |
   curtime | date | year | month | dayofweek
<relation_op> ::= = | <> | < | > | <= | >=
<arith_op> ::= + | - | * | /
<action_list> ::= <action_list> <action_stmt> | <action_stmt>
<action_stmt> ::= notifyEvent(<event>); |
   fireAction(<ACTION_NAME>(<params_list>)) |
   <VARIABLE> := <expr>;
<event> ::= <EVENT_ID>(<params_list>)|
   eventBatch(<event_var_list>)
<event_var_list> ::= <event_var_list>, <VARIABLE> | <VARIABLE>
<params_list> ::= <params_list>, <param_value> | <param_value>
<param_value> ::= <VARIABLE> | <DATA_VALUE>
<CLASS_ID> ::= [a-zA-Z][a-zA-Z0-9_]* (\.[a-zA-Z][a-zA-Z0-9_]*)*
<VARIABLE> ::= \?[a-zA-Z][a-zA-Z0-9_]*
<DATA_VALUE> ::= INTEGER | FLOAT | STRING
<EVENT_ID> ::= <CLASS_ID>$<CLASS_ID>
<ACTION_NAME> ::= STRING
```


ECA Service Architecture

Universidad de Deusto
Deustuko Unibertsitatea



Building a Sentient Jukebox with ECA Service

"If it is Monday, a lab member is logged in and either he is working or it is raining outside, then play some cheerful music to raise the user's spirits"

```
within 15000 {/* Enforce events occur in 15 secs time span*/
  query PCMonitor$logged_in(user ?userID, host ?hostID) and
  test(dayofweek = "Monday") and
  Location$presence(user ?userID) before
  /* a presence event must occur before any event on its RHS */
  ((PCMonitor$keyboard_activity(host ?hostID, intensity ?i) and
    test(?i > 0.3)) or
   (query WeatherMonitor$report(raining ?rainIntensity) and
    test(?rainIntensity > 0.2)))
=>
  notifyEvent(Jukebox$play_music(?userID, ?hostID, "ROCK"));
}
```


Mapping of Rules to CLIPS


```
(assert (rule (ruleID 0) (ruleRegTime 1005472984621)))
(defrule rule0
  (PCMonitor$logged_in (user ?userID) (host ?hostID)
                       (timestamp ?time0#))
  (test (eq (dayofweek) "Monday"))
  (Location$presence (user ?userID) (timestamp ?time1#))
  (test (> ?time1# 1005472984621))
  (test (> ?time1# (- (curtime) 15000)))
  (or (and (and (PCMonitor$keyboard_activity (host ?hostID)
                              (intensity ?i) (timestamp ?time2#))
                (test (> ?time2# 1005472984621))
                (test (> ?time2# (- (curtime) 15000)))
                (test (> ?time2# ?time1#)))
           (test (> ?i 0.3)))
      (and (WeatherMonitor$report (raining ?rainIntensity)
                                  (timestamp ?time3#))
           (test (> ?rainIntensity 0.2))))
=>
  (bind ?currentTime# (curtime))
  (bind ?factID0# (assert (Jukebox$play_music# 0 ?currentTime#
                                          ?userID ?hostID "ROCK")))
  (notify-event ?factID0#))
```

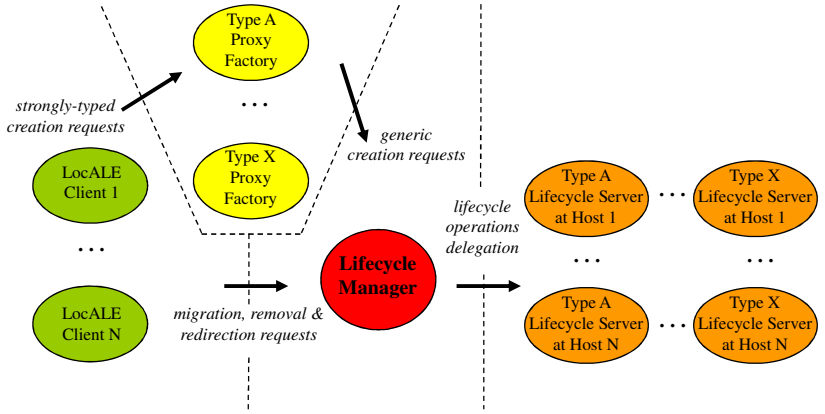
LocALE Framework

- Need to provide support for reactive behaviour of sentient systems:
  - e.g. user-bound service activation after aggregated event arrival
- LocALE = CORBA-based solution to object lifecycle & location control:
  - hybrid of CORBA's Object LifeCycle Service and Implementation Repository
  - addresses location-constrained service activation, deactivation and migration
  - adds mobility, fault-tolerance and load-balancing to objects in a location domain
  - generates permanent object references (independent of object network location)
  - undertakes transparent client request redirection upon object's location change
  - useful for third-party object location controllers:
    - e.g. "migrate the TRIP parser to another host when the used host owner logs in"

Location-constrained Object Lifecycle Control

- Why is CORBA location transparency not always desirable?
  - sometimes want to control where objects are first located and then relocated
    - e.g. load-balancing or follow-me applications
- LocALE provides apps with location-constrained object lifecycle-control:
  - apps specify on distributed object creation its initial location:
    - within a host, e.g. `hostDN("guinness")`
    - any host in an spatial container (room), e.g. `roomID("Room_1")`
    - in any location domain's host, e.g. `hostDN("ANY")` or
    - in one of a given set of hosts, e.g. `hostGroup("heineken", "guinness")`
  - … and restrictions under which an object can later be moved and/or recovered:
    - `LC_CONSTRAINT(RECOVERABLE | MOVABLE)` # any host of location domain
    - `LC_CONSTRAINT(RECOVERABLE_WITHIN_ROOM | MOVABLE_WITHIN_ROOM)`

LocALE $3^{1/2}$ tier Architecture



LocALE 3½-tier architecture

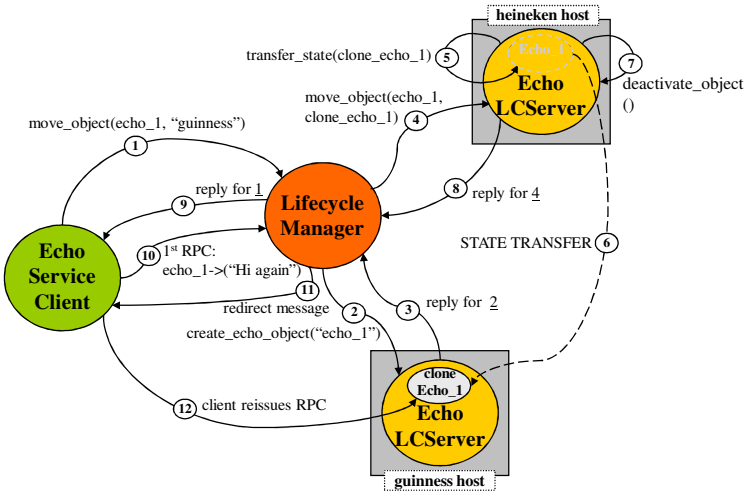
LocALE IDL Interface

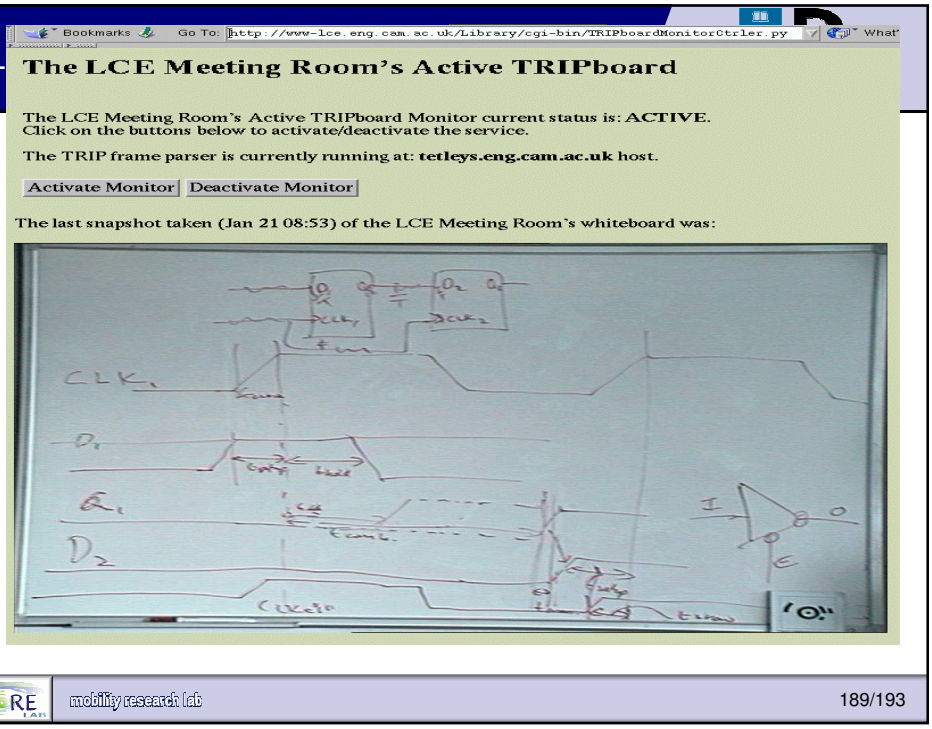
```
interface LCServer {
   void move_object(in LocALE_Object objRef,
                    in LocALE_Object cloneObjRef)
      raises(UnKnownObj, WrongObjType, InvalidClone);
   void destroy(); // Method to destroy LCServer
};

interface LocALE_Object {
   void destroy();
   void transfer_state(in LocALE_Object cloneRef)
      raises (InvalidClone);
   readonly attribute HostDN currentLocation;
};
```

LocALE IDL Interface (cont)

```
interface LCManager {

    LocALE_Object create_object(in Kind k, in AnyList
    params, in Location where, in ObjectLCPolicyList
    policyList) raises (NoLCServer, InvalidParam,
    NoExistingLocation, WrongPolicySpecification,
    ObjectLimitReached);

    void move_object(in LocALE_Object objRef, in Location
    where) raises (UnKnownObject, NoExistingLocation,
    NoPossibleMigration);

    void remove_object(in LocALE_Object objRef) raises
    (UnKnownObject);

    LocALE_ProxyFact find_proxy_factory(in Kind k) raises
    (UnknownType);
};
```


LocALE Lifecycle Control Flow





Follow-Me Audio

Universidad de Deusto
Deustuko Unibertsitatea

- Provides mobile users with music from the nearest set of speakers
- MP3 decoder and player follow the user to his new location.
- Uses TRIP as a real-time location and music selection device
- Uses ECA Service to register contextual situations to be monitored
- Uses LocALE's migration support

Conclusions

Universidad de Deusto
Deustuko Unibertsitatea

- Assortment of technologies to make Sentient Computing available to everyone:
  - TRIP 3-D location distributed sensor
  - rule-based programming paradigm for sentient applications
  - LocALE object lifecycle- and location-control middleware
  - couple of sentient applications as ‘proof of concept’

Universidad de Deusto
Deustuko Unibertsitatea

MORE LAB mobility research lab

Tema 4 – Computación Ubicua

Dr. Diego Lz. de Ipiña Gz. de Artaza
http://paginaspesonales.deusto.es/dipina
http://www.morelab.deusto.es
http://www.ctmd.deusto.es

Trabajos

Universidad de Deusto
Deustuko Unibertsitatea

- Exposición de 15 minutos sobre los siguientes temas:
  - RFID + Internet of Things: Capacidades de codificación de las etiquetas RFID más populares en el mercado, mecanismos de codificación eficiente