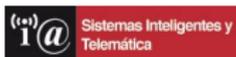


Inteligencia Ambiental basadas en Tecnologías de la Web Semántica

Juan A. Botía and José T. Palma, 2010.



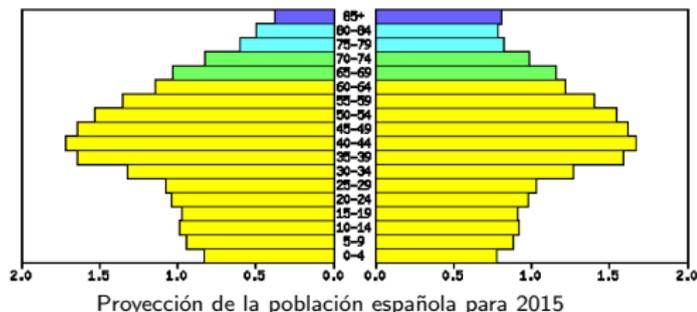
- 1 Planteamiento del Problema
 - El Problema del Envejecimiento
 - Ambient Assisted Living
 - Inteligencia Ambiental
- 2 El proyecto DIA, modelo conceptual
- 3 El proyecto DIA, ingeniería
- 4 Conclusiones

Aging Society

- Es un hecho contrastado que la edad media de la población mundial está aumentando a medida que la esperanza de vida es cada vez mayor
- Es el cambio demográfico más serio que tendremos que afrontar en los próximos años
 - Entre 2008 y 2060 el número de personas mayores de 65 años va a pasar del 17% al 30%
 - Se prevé que para el 2060 sólo un tercio de la población tenga menos de 80 años
 - Y a partir de 2015 los fallecimientos superarán a los nacimientos.

La situación con los mayores en España

España también es un país *de edad avanzada* y el problema se va a agravar en los próximos años debido, en parte, a unos índices de natalidad bajísimos



Implicaciones de este cambio demográfico

- Estos cambios demográficos implicarán importantes cambios en el comportamiento social, estilos de vida y la propia consideración que se tienen las personas mayores:
 - Podrán vivir de manera independiente por más tiempo
 - Podrán trabajar durante más tiempo
 - Sin embargo, el aumento de la esperanza de vida trae consigo un incremento de la prevalencia de problemas de salud, sobre todo mentales.
- Esta situación tendrá fuertes implicaciones económicas para los gobiernos
- Pero también supone una importante fuente de oportunidades de innovación y negocio.

El programa AAL de la UE

AAL (*Ambient Assisted Living*) o
ICT-enabled assisted living

Todo está en
<http://www.aal-europe.eu/>

Objetivo *The objective of the AAL Joint Programme is to enhance the quality of life of older people and strengthen the industrial base in Europe through the use of Information and Communication Technologies (ICT).*

Aproximadamente 700M(euros) a
gestionar entre 2008 y 2013



Objetivos del programa AAL

Definición del concepto AAL, en base a lo que se busca conseguir

- Aumentar el tiempo que las personas mayores viven de manera independiente
- Apoyar el mantenimiento de una buena salud y vida funcional en los mayores
- Fomentar mejores estilos de vida en individuos con riesgo
- Aumentar la seguridad, prevenir el aislamiento social de la persona mediante el mantenimiento de la red de contactos del individuo
- Apoyar a cuidadores, familias y organizaciones relacionadas.

¿Cómo?

- Potenciando el uso de *Sistemas Inteligentes* para conseguir un estilo de vida mejor, más sano y seguro, en el entorno en el que está se desarrolla.
- Integrando conceptos, productos y servicios que se interconectan para mejorar el entorno social en el que operan.
- Reduciendo los costes sociales y sanitarios de los gobiernos
- Con una fuerte inversión en I+D+i y apoyando a los esfuerzos de los miembros de la UE

¿En qué dominios de aplicación?

- **AA4Persons (AALhome, AALMobile):** Conseguir una mejor y mas sana calidad de vida diaria por más tiempo por medio del uso de la tecnología y manteniendo un alto grado de independencia, autonomía y dignidad.
 - AAL en Salud
 - Gestión sanitaria personalizada (dentro y fuera del hogar)
 - Tele-monitorización y autogestión de enfermedades crónicas
 - Asistencia a los proveedores sanitarios
 - AAL en el Hogar: Garantizar que la persona se sienta segura en casa.
 - AAL y los sistemas Biorobóticos: Asistencia a personas impedidas
 - AAL y servicios orientados a las personas: compras, cuidados personales, alimentación, interacción social y comunicación

¿En qué ámbitos?

- **AAL en la sociedad:** Permanecer socialmente activo y creativo por más tiempo.
 - Inclusión social: participación en distintas actividades, creatividad, aficiones, deporte
 - Entretenimiento
 - Movilidad
 - Soporte a la movilidad física
 - AA Driving: tanto personal como público
- **AALwork:** Permanecer laboralmente activo y productivo por más tiempo, mejorando la calidad del trabajo y con un mejor balance trabajo/tiempo libre.
 - Acceso al puesto de trabajo
 - Garantizar condiciones laborales óptimas.
 - Asistencia en el desarrollo de las tareas
 - Prevención de problemas de salud y posibles lesiones.

¿En qué ámbitos tecnológicos?

- **Sensores:** Corporal (invasiva/no invasiva), en los electrodomésticos o en el entorno (en el hogar, en el exterior, en el vehículo, ...)
- **Sistemas de razonamiento:**
 - Recolectando, procesando y analizando datos.
 - Transformando la información en conocimiento.
 - Integración de capacidades de razonamientos en la infraestructura sensorial
- **Actuadores:** control automático (local o remoto) por retroalimentación instantáneo o diferido, interfaces multimodales.
- **Comunicaciones:** sistemas de comunicaciones dinámicos.
- **Interacción:** Sistemas de interacción inteligentes que adaptados a las capacidades de los usuarios.

¿Qué problema queremos resolver?

- El problema que intentamos resolver se encuadra dentro de ALL@Home: desarrollar un sistema que detecta problemas en personas mayores independientes que viven solas de manera no intrusiva.
 - El usuario target es la persona mayor que vive sola y de manera independiente
 - Perseguimos una funcionalidad clara: detección automática de caídas y desvanecimientos (e.g. problemas coronarios imprevistos)
 - Restricciones: nada de sensores *wearables*, nada de cámaras (evitamos el efecto *big brother*)
 - Ausencia de requerimiento de respuesta rápida
 - **Tesis**: es posible diseñar un sistema inteligente que detecte esas situaciones, con 0.0 **falsos negativos** y **falsos positivos** minimizados

¿Qué áreas están implicadas?

- **Inteligencia Ambiental:** Parte de las ciencias de la computación que se centra en el análisis, diseño y desarrollo de sistemas que son sensibles y pueden responder de forma autónoma a las personas, además de adaptarse a su comportamiento.
- Todo es posible gracias a un entorno en el que las personas están rodeadas de dispositivos inteligentes integrados en los objetos de uso diario y al integración de diferentes tecnologías:
 - Computación Ubicua
 - Interfaces Inteligentes
 - Computación Sensible al Contexto (Context-Aware Computing)

Computación Ubicua

- **Objetivo:** facilitar el uso de los sistemas computacionales haciendo que estos sean accesibles desde cualquier lugar y de forma invisible al usuario.
 - Las capacidades de procesamiento están distribuidas e integradas en objetos de uso diario.
 - a través de un conjunto de dispositivos de procesamiento interconectados y distribuidos por el entorno en que se desarrollan las actividades diarias.
 - Para ello se requiere la percepción del entorno en el que los usuarios y dispositivos operan

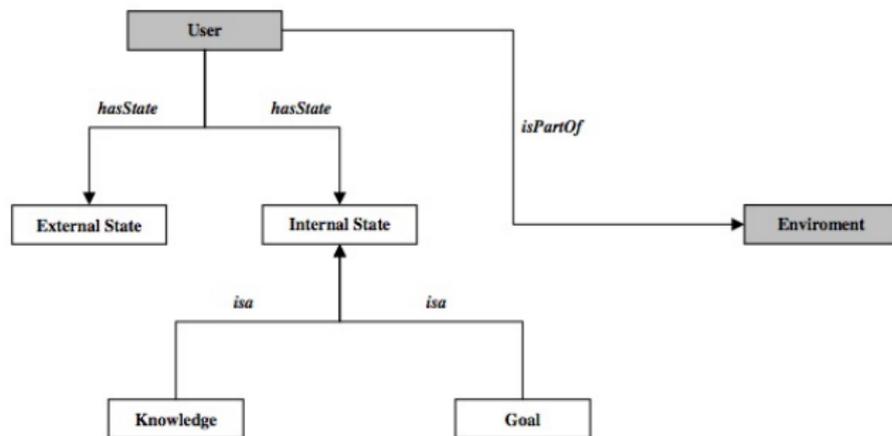
Computación Sensible al Contexto

- **Objetivo:** permitir que los dispositivos puedan proporcionar mejores servicios a los usuarios mediante la utilización de la información de contexto disponible.
- **¿Cómo?**
 - Mediante una mejor **prestación de servicios**
 - Gracias a una **adaptación proactiva** del uso, acceso, estructura y comportamiento
 - De la información, los servicios, las aplicaciones y el entorno físico
 - En función de la **información de contexto disponible**

¿Qué es el Contexto?

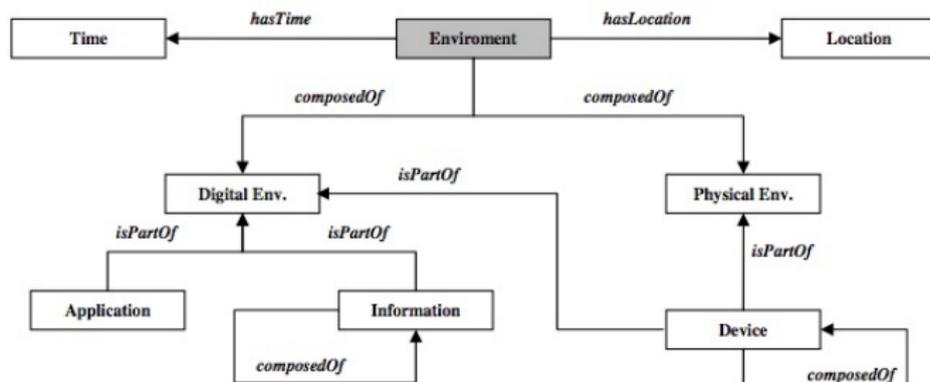
- Se entiende por Contexto cualquier información que puede ser usada para caracterizar el estado de una entidad.
 - **Entidad:** persona, lugar u objeto relevante para la interacción entre el usuario y las aplicaciones, siendo ambos también parte del contexto.
- El que algo sea considerado como contexto depende de la forma en qué es usado en el proceso de interpretación, no a sus propiedades intrínsecas.

¿Qué es el Contexto?



[Soylu et al., 2009]

¿Qué es el Contexto?



[Soylu et al., 2009]

Componentes de una Aplicación Sensible al Contexto

- **Modelado y Representación del Contexto:** Cómo el contexto es modelado y representado de forma explícita
- **Adquisición del Contexto:** Cómo la información de contexto es adquirida
- **Razonamiento y Abstracción sobre Contexto:** Procesamiento de la información de contexto
- **Diseminación del Contexto:** Cómo se almacena y distribuye la información del contexto

Modelado y Representación de Contexto: Requerimientos

- Tiene que ser **Estructurado**
- Debe poder **Intercambiarse**
- Se tiene que poder **Componer o Descomponer**
- Debe ser **Uniforme**
- Debe ser **Extensible y Dinámico**
- Debe ser fácilmente **Estandarizable**
- Debe ser **Abierto**
- **Mínimo**

¿Porqué Usar Ontologías

- Proporcionan la capacidad **expresiva** necesaria
- Ofrecen un marco **estructurado y estándar** para la representación de información/conocimiento
- Facilitan la **compartición y reutilización** de la información/conocimiento
- Pueden ser **extendidas** dinámicamente
- Ofrecen mecanismos básicos de **razonamiento**
- Pueden trabajar con información **incompleta**

¿Porqué Usar tecnologías de la Web Semántica

- Los lenguajes que ofrecen, OWL y RDF, permiten añadir información de forma dinámica
- Están diseñados para trabajar en entornos distribuidos
- Se ofrecen herramientas que permiten el modelado y el razonamiento
- En el caso de OWL, permite la descripción de reglas (SWRL)
- Permiten reutilizar trabajos previos
 - Por ejemplo, podemos basarnos en ontologías de carácter general: SUMO, Cyc or DOLCE
 - Aunque en la mayoría de los casos se adopten soluciones ad-hoc: CDF, C-OWL, Context, FOAF, GUMO

DIA (Dispositivo Inteligente de Alerta)

Objetivo del proyecto: mediante técnicas de computación ubicua, no invasivas, desarrollar un sistema que detecta problemas en personas mayores independientes que viven solas

Componentes



GSI & AIKE



Motivación

- El usuario target es la persona mayor que vive sola y de manera independiente
- Perseguimos una funcionalidad clara: detección automática de caídas y desvanecimientos (e.g. problemas coronarios imprevistos)
- Restricciones: nada de sensores *wearables*, nada de cámaras (evitamos el efecto *big brother*)
- Ausencia de requerimiento de respuesta rápida
- Tesis: es posible diseñar un sistema inteligente que detecte esas situaciones, con 0.0 **falsos negativos** y un número de **falsos positivos** razonable

Definiendo el entorno

- Definimos la estructura del problema como una tripla

$$(Env, U, Ap)$$

- Ahora podemos definir el entorno $Env = \{d_1, d_2, \dots, d_n\}$
 - definimos cada sensor como $d : \mathcal{E} \rightarrow L$ en donde \mathcal{E} denota el entorno físico, $L = \{yes, no\}$
- Definimos entonces el detector de caídas como $Ap : L^n \rightarrow L$, en donde n es el número de sensores
- Asumimos aquí que el problema es markoviano

Teniendo en cuenta el pasado

Si, en un tiempo t , hay inactividad en la casa, ¿está el atendido ausente o ha sufrido un problema que lo inmoviliza en el suelo?

- Optamos por refinar Ap como

$$Ap(d_1(t), d_1(t-1), \dots, d_2(t), d_2(t-1), \dots, d_n(t), d_n(t-1), \dots),$$

en donde $d_i(t')$ es la lectura del sensor i -ésimo en el instante t'

- ¿Cuál debería ser la naturaleza de AP?
 - Las entradas y salidas son categóricas
 - Representaciones basadas en reglas o árboles de decisión se obtienen a partir de aprendizaje mediante ejemplares (inducción)
 - Podemos intentar diseñarlo a mano, al menos inicialmente

¿Qué podemos asumir del entorno?

Ejemplo altamente probable de una casa de atendido



¿Qué podemos asumir del entorno?

El sistema instalado



Si incorporamos el sentido común en Ap

Posible definición simbólica de Ap

If $(a_{living}(t, t - \alpha) = no \wedge a_{bedrom}(t, t - \alpha) = no \wedge$
 $a_{bath}(t, t - \alpha) = no \wedge a_{kitchen}(t, t - \alpha) = no \wedge$
 $p_{bedroom}(t) = no \wedge p_{living}(t) = no)$
 $\vee p_{bedroom}(t, t - \beta) = yes \vee p_{living}(t, t - \beta) = yes$
then *Yes*
else *No*

en donde

- $a_{lugar}(t, t - \alpha)$ es una función booleana que indica si hubo actividad en el intervalo $[t - \alpha, t]$ en *lugar*
- $p_{lugar}(t, t - \beta)$ es análoga con la presión (cte.) en un mueble ubicado en *lugar*

Los parámetros α y β

Son dos parámetros importantes

- El parámetro α
 - Asociado al sensor de actividad
 - Dependiente de la habitación?
 - Modula la reactividad del sistema ante inactividad
 - Compromiso entre reactividad y nivel de falsas alarmas
- El parámetro β complementa a α
 - Asociado al sensor de presión
 - Dependiente del mueble en donde es instalado
 - Ayuda en descansos prolongados (e.g. una simple siesta)
 - La ingeniería del sensor ha de ser cuidadosa

Conclusiones sobre el diseño de Ap

- Es una definición sencilla → es posible validar por experto
- Hace uso mínimo de tres tipos de sensores, no *wearables*
- Elimina la posibilidad de que ocurran falsos negativos
- ¿Qué ocurre con los falsos positivos?
- Hay una importante componente de historicidad

Críticas al diseño

Jugamos con dos suposiciones básicas

- 1 Ap se debería desactivar automáticamente cuando el atendido está fuera de casa
 - De ahí que existan sensores de puerta
- 2 Se supone que Ap está diseñado para cuando el atendido está solo en casa
 - Cuando hay algún acompañante Ap es innecesario
 - ¿Qué ocurre con las mascotas?

El modelo de Ap genérico

Dado que tenemos un entorno genérico

$Env = (a_1, a_2, \dots, a_n, p_1, p_2, \dots, p_m)$ con n sensores de actividad y m sensores de presión, podemos definir Ap para dicho Env como

If $(a_1(t, t - \alpha) = no \wedge a_2(t, t - \alpha) = no \wedge \dots \wedge a_n(t, t - \alpha) = no$
 $p_1(t) = no \wedge p_2(t) = no \wedge \dots \wedge p_m(t) = no)$
 $\vee p_1(t, t - \beta) = yes \vee p_2(t, t - \beta) = yes \vee \dots \vee p_m(t, t - \beta) = yes$
then *Yes*
else *No*

Por tanto, podemos decir que es válido para cualquier tipo de hogar

Personalizando A_p

Principal hipótesis: cada atendido tiene su propio ritmo de vida diaria, por lo que podemos definir el entorno como

$$\begin{aligned}
 Env_{U_i} = & ((a_1, \alpha_1(U_i)), (a_2, \alpha_2(U_i)), \\
 & \dots, \\
 & (a_n, \alpha_n(U_i)), (p_1, \beta_1(U_i)), (p_2, \beta_2(U_i)), \\
 & \dots, \\
 & (p_m, \beta_m(U_i))),
 \end{aligned}$$

en donde a_i es el i -ésimo sensor de actividad, p_j es el j -ésimo sensor de presión y $\alpha_i(U_k)$ y $\beta_j(U_k)$ indican que los timers de actividad y presión dependen de cada atendido U_k

Pero, ¿y cómo se construye un sistema así?

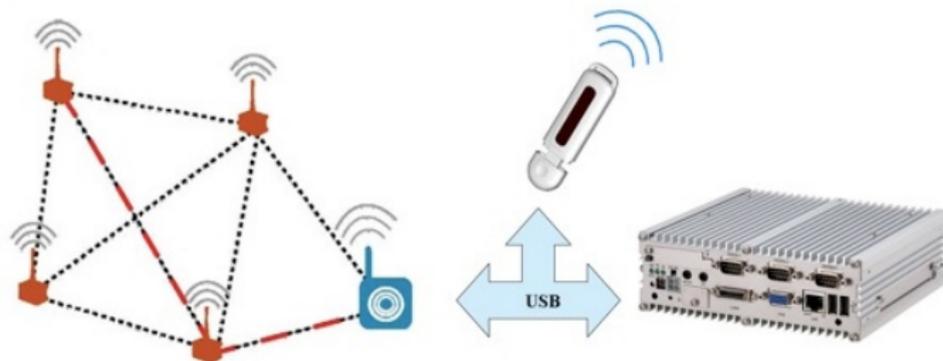
Paradigmas de referencia

Computación ubicua (Weiser, M. 1991): cjto. de dispositivos y programas orientados a aumentar el bienestar y la productividad del usuario de forma inadvertida.

Inteligencia Ambiental (Zelka & Epstein, 1998): énfasis en sistemas ubicuos que interaccionan con el usuario de forma natural y sin dificultades

Disciplinas/tecnologías: miniaturización, redes de sensores, mecanismos de coordinación, interacción hombre-máquina, adaptación, sistemas fiables, etc

Arquitectura hardware

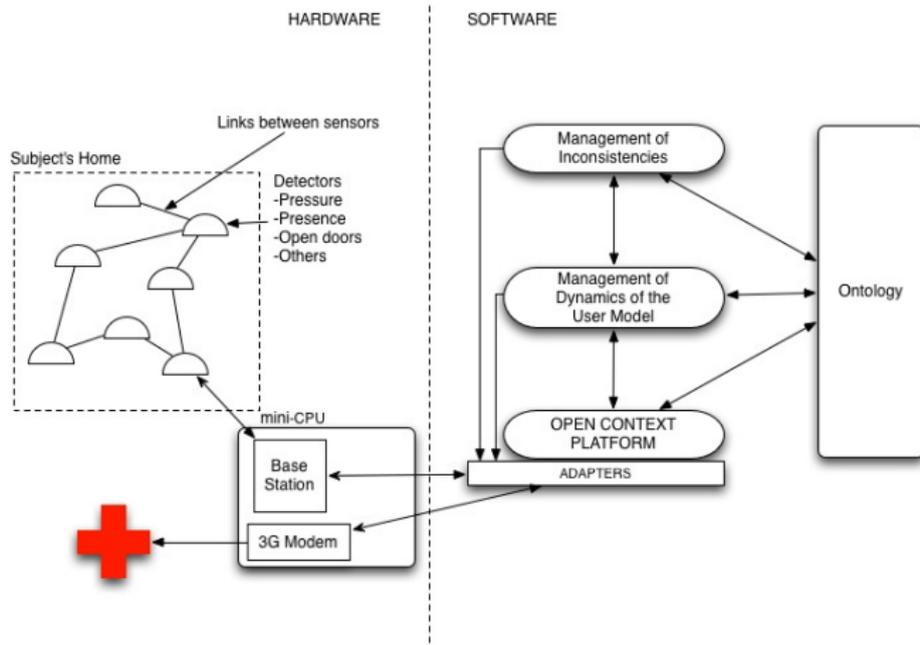


<< Motes >>	
HARDWARE	PDS200
SOFTWARE	Tiny-OS + XPDS200.2
TASKS	Data Collecting + Mesh Support

<< WSN Base >>	
HARDWARE	PDSB100
SOFTWARE	Tiny-OS + XmeshBase
TASKS	Routing Management + WSN-Pc Interface

<< Processing Station >>	
HARDWARE	miniPC - PC/104
SOFTWARE	Linux + DIA Application SW
TASKS	Patient Monitoring + Alert Callings + System Survey

Arquitectura completa

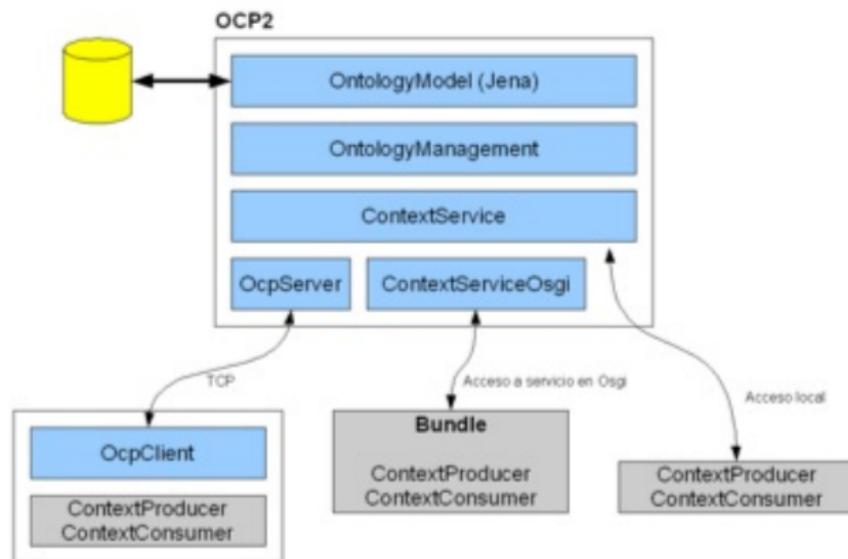


El software: capa básica

OCP (*Open Context Platform*)

- Middleware para gestión de información de contexto
- Coordinación mediante el mecanismo productor/consumidor
- Representación del dominio mediante Web Semántica
- Implementación basada en Jena/Pellet, 100% Java
- Integrado en OSGi
- Función en PRODIA: obtener, almacenar y significar la información del entorno

OCP, arquitectura



Ontologías en el dominio UbiCom

Hay un número importante de ontologías para sistemas UbiCom

- COBRA-ONT (2003)
- GAIA (2003)
- SUMO (2003)
- SOUPA (2004)
- CODAMOS (2004)
- SOCAM (2004)
- CONON (2004)
- ULCO (2004)

¿Qué describen?

Todas incluyen conceptos para modelar

- Espacio físico: la localización es un activo importante en UbiCom
- Agentes: los agentes son dinámicos (i.e. software/humanos) y su contexto cambia por tanto
- Actividad del agente: activo importante para el contexto
- Eventos: sin duración, indican cambios en el estado de las cosas
- Tiempo: el tiempo es importante para ordenar eventos cuando sea posible
- Reusan otras: FOAF, TIME, OpenCyc, OpenGIS

Nuestra propuesta

En la versión 1.0

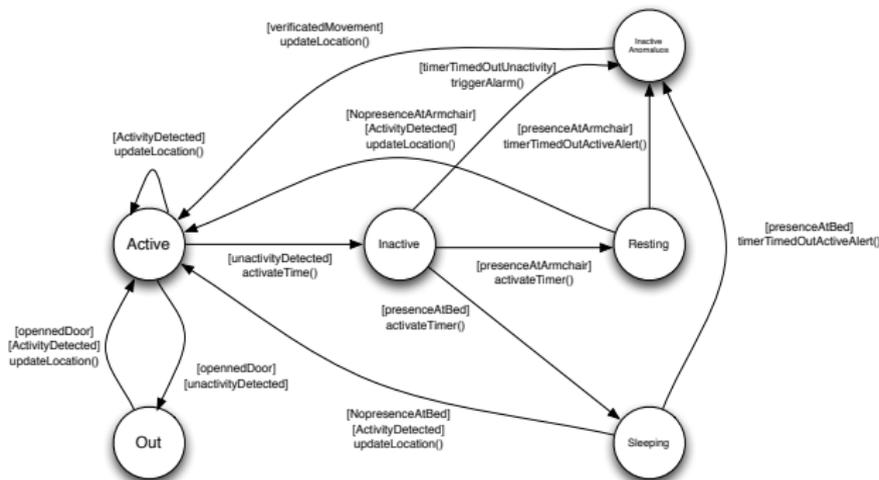
- OCP incluye únicamente la entidad ContextEntity
- Reusamos elementos de SOUPA
- Ampliamos con elementos espaciales para representación de hogares
- Estado del atendido

En la versión 2.0

- Partimos de nuestra propia ontología UbiCom (similar a las anteriores), se denomina OCP-CORE
- Se particulariza para dominios concretos (e.g. hospital, hogar, etc.)

El software: capa de modelado de usuario

Utilizamos un AFD centrado en estados de actividad del usuario



Transiciones disparadas por reglas diseñadas a mano

El software: capa de modelado de usuario

Un ejemplo de regla usada como transición al estado activo

$$\begin{aligned} & \textit{Attended}(?x) \wedge \textit{location}(?x, ?y) \wedge \textit{MovementSensor}(?s) \wedge \\ & \quad \textit{detectMovement}(?, \textit{true}) \wedge \textit{inside}(?s, ?r) \wedge \\ & \quad \textit{HabitSpaceInBuilding}(?r) \wedge \textit{haveConnection}(?r, ?c) \wedge \\ & \quad \textit{connectionWith}(?c, ?c1) \wedge \textit{equal}(?y, ?c1) \\ & \quad \Rightarrow \\ & \quad \textit{state}(?x, \textit{"Active"}) \wedge \textit{location}(?x, ?r) \end{aligned}$$

La representación interna es OWL + SWRL y se utiliza Jena + Pellet para el motor de reglas

Implementando los α y β

En realidad, el autómata implementado es un autómata temporizado

- Cada vez que se sale de un estado se desactiva el correspondiente temporizados que corresponde a un α y quizás a un β
- Cuando se llega a un estado, se activa su correspondiente α y quizás algún β
- Si se cumple algún temporizador se realiza transición a un estado anómalo

El software: capa de meta-razonamiento

El AFD puede tener inconsistencias (i.e. dado a un estado, es posible pasar a más de uno)

Se necesita un mecanismo para disparar alarmas

Esta capa se encarga de ambas cosas, resolviendo las inconsistencias también mediante reglas, para dar una posible explicación

Una vez construido, ¿cómo lo validamos?

¿Qué tipo de sistema es este?

Sistema artificial complejo

- Fuerte componente hardware
- Dependiente de comunicaciones wireless
- Software complejo, varios niveles de ejecución
- Fuerte componente de iHCI
- Varios equipos de desarrollo segregados por niveles

¿Qué podemos testear para validar en la UMU?

- Comunicación hardware-software (i.e. desde estación base mediante driver al software de tres niveles)
 - Simulación de entorno y atendido
- Arquitectura de tres niveles
 - Generación de historias y sus correspondientes outputs (simulador)
- Sistema completo mediante living-lab

Living-labs para validación de sistemas Aml

Citando la definición de W. Mitchell, del MediaLab en el MIT, *los living labs representan una investigación centrada en el usuario para el seguimiento mediante percepción, prototipado, validación y refinamiento de soluciones complejas en contextos cambiantes continuamente, de la vida real.*

Utilizamos una casa real, un actor, y diseñamos experimentos para comprobar la efectividad del sistema

Estructura del sistema de validación

Dos ubicaciones diferentes

- Una casa real, situada en Murcia
- Una cabina de realización, situada en la Fac. Informática de la UMU

Tres actores diferentes

- Actor-atendido en casa real
- Director del experimento en UMU
- Anotador del experimento en UMU

Tecnologías utilizadas

Skype, ssh y un procesador de textos

The image shows a computer screen with two overlapping windows. The background window is a terminal titled 'prodis' with the path '/ProDiaRunMir'. It displays XML-RDF data representing sensor readings and reasoning steps. The data includes attributes for location, temperature, sensor movement detection, and voltage, along with reasoning logs from a 'Reasoner'. The foreground window is a Skype video call with a participant named 'Emilio Iborra'. The call interface shows a video feed of a person in a room, a smaller inset of the same person, and standard call controls like a volume slider, microphone, and video toggle.

```
[OCP : Day 10 , 13 : 04 : 33] :Atributelocation : dormitorio

[OCP : Day 10 , 13 : 04 : 33] :Atributotemperature : 22

[OCP : Day 10 , 13 : 04 : 33] :Sensor : sensor320 detected MOVE: false
  [ Reasoner : Day 10 , 13 : 04 : 33] :Reasoning...
  [ Reasoner : Day 10 , 13 : 04 : 34] :Rules deducted
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.0="http://www.owl.ontologies.com/ProDia.owl#j.0" >
  <rdf:Description rdf:about="http://www.owl.ontologies.com/ProDia.owl#maria"
    <j.0:neuEvent rdf:datatype="http://www.w3.org/2001/XMLSchema#string">PressBed
  </rdf:Description>
</rdf:RDF>

  [ Reasoner : Day 10 , 13 : 04 : 34] :End Reasoning...
  [ Reasoner : Day 10 , 13 : 04 : 34] :New Event PressBed

[OCP : Day 10 , 13 : 04 : 34] :Attributeidentify : 32

[OCP : Day 10 , 13 : 04 : 34] :Attributevoltage : 2896
  [ Reasoner : Day 10 , 13 : 05 : 08] :Timer: dormitorioHidda:TimerActiveAnomalous finalize
  [ Reasoner : Day 10 , 13 : 05 : 08] :Reasoning...
  [ Reasoner : Day 10 , 13 : 05 : 08] :Rules deducted
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.0="http://www.owl.ontologies.com/ProDia.owl#j.0" >
  <rdf:Description rdf:about="http://www.owl.ontologies.com/ProDia.owl#mariaana"
    <j.0:neuEvent rdf:datatype="http://www.w3.org/2001/XMLSchema#string">PressBed</j.0:neuEvent>
  </rdf:Description>
</rdf:RDF>
```

Lecciones aprendidas del living-lab

Experimentar en entornos reales es muy útil porque

- Hay errores de diseño que difícilmente se detectan de otra forma
 - Subestimamos el problema del alcance cruzado de los sensores
 - Los sensores de presión iniciales no funcionaron tan bien como se esperaba
 - Descubrimos zonas muertas y el patrón de actividad anómala
- Te permite tener un demostrador para mostrar a posibles clientes

Pero, ¿cómo se conecta esto con el resto del mundo?

- Creamos una central de alarmas
- Utilizamos tecnología de telefonía móvil (3G, GPRS) para conexión
- Aumentamos las capacidades de autodiagnóstico del sistema
- Integramos el producto en un producto global de teleasistencia

Problemas actuales: el mantenimiento!!!

El mantenimiento de estos equipos es algo delicado

- Alto número de componentes por casa
- Equipos con un coste de fabricación razonable (hardware industrial?)
- Funcionando 24x7x12
- Personal no cualificado
- Problemas de índole social

Problemas actuales: la supervisión!!!

Este problema no se reduce a algo tecnológico: fuerte componente social

- El sistema convive con un humano
- Se instala en todo tipo de casas (condiciones ambientales variables y no supuestas de antemano)
- El problema de las mascotas
- La casuística de cada lugar es muy específica

Conclusiones

- Las personas mayores son un dominio incipiente para las TIC
- La cooperación universidad-empresa puede generar innovación con facilidad y en poco tiempo
- Hay que tener en cuante a las personas en los diseños sotware!!



Referencias

Muy recomendables

- [Soylu et al., 2009] Ahmet Soylu, Patrick De Causmaecker and Piet Desmet. **Context and Adaptivity in Pervasive Computing Environments: Links with Software Engineering and Ontological Engineering.** *Journal of Software*, vol. 4, no. 9 pp. 992-1013 (2009).
- [Euzenat et al., 2008] Jerome Euzenat, Jerome Pierson and Fano Ramparany. **Dynamic Context Management for Pervasive Applications.** *The Knowledge Engineering Review*, vol 23:1, pp. 21-49 (2008).

Para indagar algo más en el trabajo

- [Botía et al., 2009] Juan A. Botía, Ana Villa, Jose T. Palma, David Pérez, Emilio Iborra. **textbfDetecting domestic problems of elderly people: simple and unobstrusive sensors to generate the context of the attended.** *First Internationa Workshop on Ambient Assisted Living, IWAAL*, 2009. Salamanca, Spain.
- [Palma et al., 2009] José Palma, Juan Botia, Ana Villa, David Pérez, Emilio Iborra. **An Ontology Based Approach to the Detection of Domestic Problems for Independent Senior People.** *International Work-conference on the Interplay between Natural and Artificial Computation*, 2009.

Muchas gracias!!

Juan A. Botía y José T. Palma
{juanbot, jtpalma}@um.es

El resto del equipo: David Pérez, Ana Villa, Francisco Luque,
Miguel A. Sotomayor, Adrián Nicolás, Sergio Fructuoso,
Francisco López, Juan Verdú, Antonio Rosa, Juan Zapata,
Xavier Cerdá, Emilio Iborra, José Iborra

Curso de IA en la Web, UNED 2010

