EUSKO JAURLARITZA
GOBIERNO VASCO

Programa Saiotek 2006

SMARTLAB

Entorno de Trabajo Inteligente Colaborativo y Programable

Ontologías para modelar un entorno de trabajo inteligente

Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa

HISTORIAL DE CAMBIOS

Versión	Descripción	Autor	Fecha	Comentarios
V0.1	Versión inicial con varias ontologías: FOAF, Dublín Core, Basic Geo, DAML Time entry sub-ontology, OpenCyc y MoGATU BDI	Aitor Almeida	11/05/2007	
V0.2	Añadidas SOUPA Core, SOUPA Extension, CONON, CoDAMoS y comparativa	Aitor Almeida	17/05/2007	
V0.3	Añadido razonamiento sobre el contexto	Aitor Almeida	22/05/2007	

TABLA DE CONTENIDOS

1 INTRODUCCIÓN

En este documento se recogen las ontologías que podrían servir para modelar un entorno de trabajo inteligente. Estas ontologías contemplan varios aspectos:

- El contexto físico: Tiempo, espacio, personas…
- El contexto organizativo: Reuniones, eventos, acciones, políticas, agentes…
- La taxonomía de dispositivos disponibles en el entorno.

2 FOAF

FOAF (Friend-of-a-Friend) [FOAF] permite modelar a **personas y las relaciones entre ellas**. El elemento central dentro de la ontología son las personas: a que grupo pertenecen, cuales son sus datos, que otras personas conocen…

En la especificación [FOAFESP] se detallan 5 categorías para los elementos que forman la ontología:

- FOAF Basics
- Personal Info
- Projects and Groups
- Documents and Images
- Online Accounts/IM

2.1 FOAF Basics

Los elementos que entran dentro de esta categoría son:

- *Agent*: Un agente (persona, grupo, software u objeto físico)
- *Person*: Representa a una persona, tiene diferentes propiedades para almacenar la información. Es una subclase de foaf:agent
- *Name*: Es el nombre de algo (persona, agente, etc…)
- *Nick*: el mote de un agente.
- *Title*: Dr., Mr., Ms., etc…
- *Homepage*: Una homepage. Es un documento Web de acceso público pero no tiene porque ser una página HTML.
- *Mbox*: Una cuenta de correo personal. Ya que en FOAF las cuentas de correo deben

de ser únicas permiten diferencias a un agente de otro.

- *mbox_sha1sum*: es el SHA1 de una cuenta de correo. Se usa para cuando se quiere hacer público un perfil FOAF pero se quiere mantener la privacidad de la dirección de correo.

- *Img*: Una imagen que puede usarse para representar algo.

- *depiction (depicts)*: Es la relación entre una cosa y la imagen (foaf:Image) que lo muestra.

- *Surname*: El apellido de una persona.

- *family_name*: El family name de una persona.

- *Givenname*: El givenname de una persona.

- *firstName*: El first name de una persona.

2.2 Personal Info

Los elementos de esta categoría permiten definir información adicional sobre un agente.

- *Weblog*: El Weblog de un agente. Se contempla que agentes software, grupos y objetos físicos también puedan tener también un Blog.

- *Knows*: Esta relación permite modelar a que personas conoce una persona.

- *Interest*: Una URI que representa una página con los intereses de una persona.

- *currentProject*: El proyecto en el que está trabajando una persona.

- *pastProject*: Un proyecto en el que ha trabajado una persona con anterioridad.

- *Plan*: Contiene la misma información que un fichero '.plan' para poder responder a un comando finger.

- *based_near*: Permite relacionar dos entidades espaciales mediante información

geográfica.

- *workplaceHomepage*: La homepage del lugar de trabajo.
- *workInfoHomepage*: Página Web donde se describe el trabajo de una persona.
- *schoolHomepage*: Página Web de un organización donde estudia o ha estudiado una persona.
- *topic_interest*: Uno de los intereses de la persona. Marcado para su borrado por estar poco definida.
- *Publications*: Un enlace a las publicaciones de la persona.
- *Geekcode*: GeekCode [GEEK] de una persona.
- *myersBriggs*: Clasificación Myer-Briggs [MYBRI] de una persona.
- *dnaChecksum*: Checksum del DNA de una persona. Es un huevo de pascua sin utilidad real.

2.3 Projects and Groups

Los elementos en esta categoría permiten definir proyectos, grupos y organizaciones y sus miembros.

- *Project*: Representa a un esfuerzo personal o colectivo a lo largo del tiempo. Todavía no está relacionada correctamente con *currentProyect* y *pastProyect*.
- *Organization*: Una organización es mas “sólida” que foaf:Group, ya que este último permite colecciones de personas más ad-hoc.
- *Group*: Una colección de agentes.
- *Member*: una relación para indicar que se pertenece a un grupo.
- *membershipClass*: Indica una clase de individuos que son miembros de un grupo.

- *fundedBy*: Quien aporta fondos para un grupo o proyecto. Esta propiedad es experimental.

- *Theme*: un tema. Esta propiedad no está suficientemente especificada y no suele usarse.

2.4 Documents and Images

Los elementos de esta categoría sirven para definir documentos e imágenes.

- *Document*: Un documento, sus características no se encuentran todavía demasiado definidas, por lo que podría ser interesante enlazarlo con Dublin Core

- *Image*: Una imagen.

- *PersonalProfileDocument*: Un foaf:document que usa RDF para describir al foaf:maker de otro documento.

- *topic (page)*: Un tema de una página o documento.

- *primaryTopic*: El tema principal de una página o documento.

- *tipjar*: Contiene información sobre como pagar a un agente o comprar un documento. Pueden ser datos informales (“Mándame una postal”) o datos que puedan ser leídos por una máquina como una cuenta PayPal.

- *sha1*: Checksum SHA1 de un documento.

- *made (maker)*: Esta relación indica que un agente ha hecho algo.

- *thumbnail*: thumbnail de una foto.

- *logo*: Un logo para representar algo.

2.5 Online Accounts/IM

En esta categoría se definen cuentas para diferentes servicios online.

- OnlineAccount: Una cuenta online.
- OnlineChatAccount: Una cuenta online de chat.
- OnlineEcommerceAccount: Una cuenta en una página que permite vender y/o comprar productos.
- OnlineGamingAccount: Una cuenta en una página de juego online.
- holdsAccount: relación que indica que un agente tiene una cuenta en concreto.
- accountServiceHomepage: HomePage del proveedor de servicios de una cuenta.
- accountName: identificador de una cuenta online.
- icqChatID: ID de icq.
- msnChatID: ID de msn.
- aimChatID: ID de aim.
- jabbered: ID de Jabber.
- yahooChatID:id de Yahoo Chat.

3 DUBLIN CORE

El Dublin Core Metadata Element Set (DCES) [DCES] permite anotar los **metadatos de recursos digitales**. Aunque no es una ontología en puede ser expresada mediante una fácilmente.

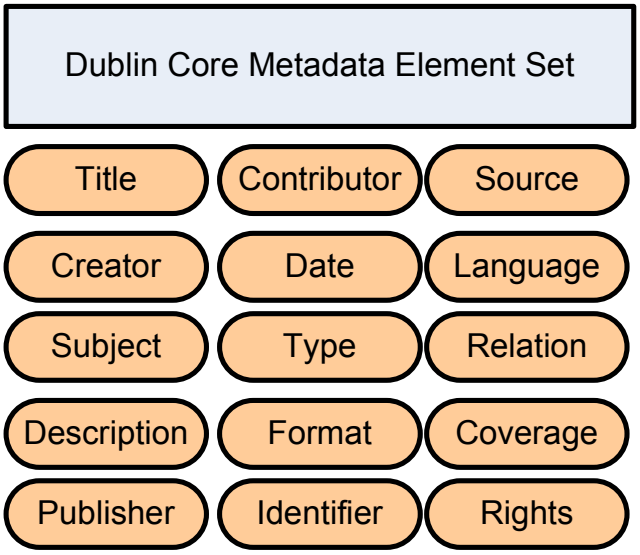

fig 1 Elementos de Dublin Core

Los elementos de DCES son:

- *Subject*: Uno de los temas del recurso. La especificación aclara: “Typically, Subject will be expressed as keywords, key phrases or classification codes that describe a topic of the resource”.

- *Language*: El lenguaje del recurso

- *Coverage*: La extension o ámbito del recurso. La especificación aclara: “Typically, Coverage will include spatial location (a place name or geographic coordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity)”.

- *Rights*: Información sobre la licencia de uso del recurso.

- *Title*: El nombre dado a un recurso.

- *Type*: El tipo del recurso.
- *Format*: El formato del recurso. La especificación aclara: “Recommended best practice is to select a value from a controlled vocabulary (for example, the list of Internet Media Types defining computer media formats)”.
- *Identifier*: Un identificador no ambiguo del recurso. La especificación aclara: “Formal identification systems include but are not limited to the Uniform Resource Identifier (URI) (including the Uniform Resource Locator (URL)) [. . .]”.
- *Contributor*: La entidad encargada de contribuir al recurso.
- *Creador*: La entidad encargada de crear el recurso.
- *Date*: Un periodo de tiempo asociado al recurso.
- *Description*: La descripción del recuso.
- *Publisher*: La entidad responsable de publicar el recurso.
- *Relation*: un recurso relacionado.
- *Source*: fuentes utilizadas para la creación del recurso.

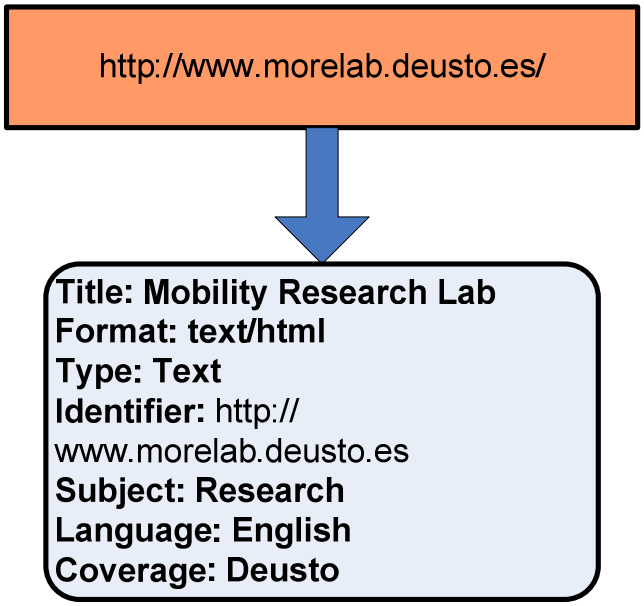

fig 2 Recurso anotado con Dublin Core

4 DAML-TIME ENTRY SUB-ONTOLOGY

DAML Time [DAMLT] permite expresar conceptos temporales. La sub-ontología creada por Feng Pan y Jerry Hobbs [SUBONT] es un subconjunto de la ontología inicial que permite expresar los conceptos necesarios en la mayoría de aplicaciones.

Lo conceptos temporales contemplados son:

- *Instant*: Un instante de tiempo.
- *Interval*: Un intervalo de tiempo.
- *InstantEvent*: Un evento que sucede en un momento dado.
- *IntervalEvent*: Un evento que sucede en un intervalo de tiempo.
- *TemporalEntity*: Una colección de instantes y/o intervalos.
- *Event*: Una collection de InstantEvent y/o IntervalEvent.
- *InstantThing* : Una colección de InstantEvent y/o Instant.
- *IntervalThing*: Una colección de IntervalEvent y/o Interval.

Las relaciones topológicas entre conceptos son:

- *Before*.
- *After*.
- *Begins*.
- *Ends*.
- *Inside*.

Las relaciones entre conceptos son:

- *intEquals*
- *intBefore*
- *intMeets*
- *intOverlaps*
- *intStarts*
- *intDuring*

- *intFinishes*
- *intAfter*
- *intMetBy*
- *intOverlappedBy*
- *intStartedBy*
- *intContains*
- *intFinishedBy*
- *startsOrDuring*
- *nonoverlap*

También existen los conceptos de año, mes, día, hora, minuto, segundo y zona horaria.

5 OPENCYC

OpenCyc es la versión open source de Cyc, un proyecto de inteligencia artificial que pretende crear una ontología que sea capaz de contener el conocimiento necesario para emular el “sentido común” usado en la vida diaria con el objetivo de razonar de manera parecida a las personas. OpenCyc engloba un conjunto completo de términos de Cyc así como millones de asertos sobre ellos.

Las categorías son las siguientes: Fundamentals, Top Level, Rule Macro Predicates, Time and Dates, Spatial Relations, Quantities, Mathematics, Microtheories and Contexts, Groups, "Doing", Transformations, Changes Of State, Transfer Of Possession, Movement, Parts of Objects, Composition of Substances, Agents, Organizations, Actors, Roles, Professions, Emotion, Propositional Attitudes, Social, Biology, Chemistry, Physiology, General Medicine, Materials, Waves, Devices, Construction, Financial, Food, Clothing, Weather, Geography, Paths and Traversals, Transportation, Information, Perception, Agreements, Linguistic Terms, Documentation, Database, Lexicon, Other Medical.

La ontología OpenCyc es tremendamente extensa, pero para poder modelar el contexto de un entorno de trabajo inteligente son interesantes dos de sus categoría: *Time and Date* y *Spatial propierties and relations*.

5.1 Spatial propierties and relations

Esta categoría sirve para modelar el contexto espacial de manera simbólica. Contiene elementos para describir las siguientes características:

- Superficies, portales y cavidades.
- 63 atributos de formas (arcos, polígonos, etc…)
- Diferentes tipos de simetría espacial.
- Vocabulario de dirección y orientación.
- Posiciones relativas de los objetos.
- Cercanía y orientación.

- Pertenencia.
- Predicados “In-“ (~ 60).
- Predicados de conexión(~ 65).
- Relaciones mereologicas (conjunto-partes)

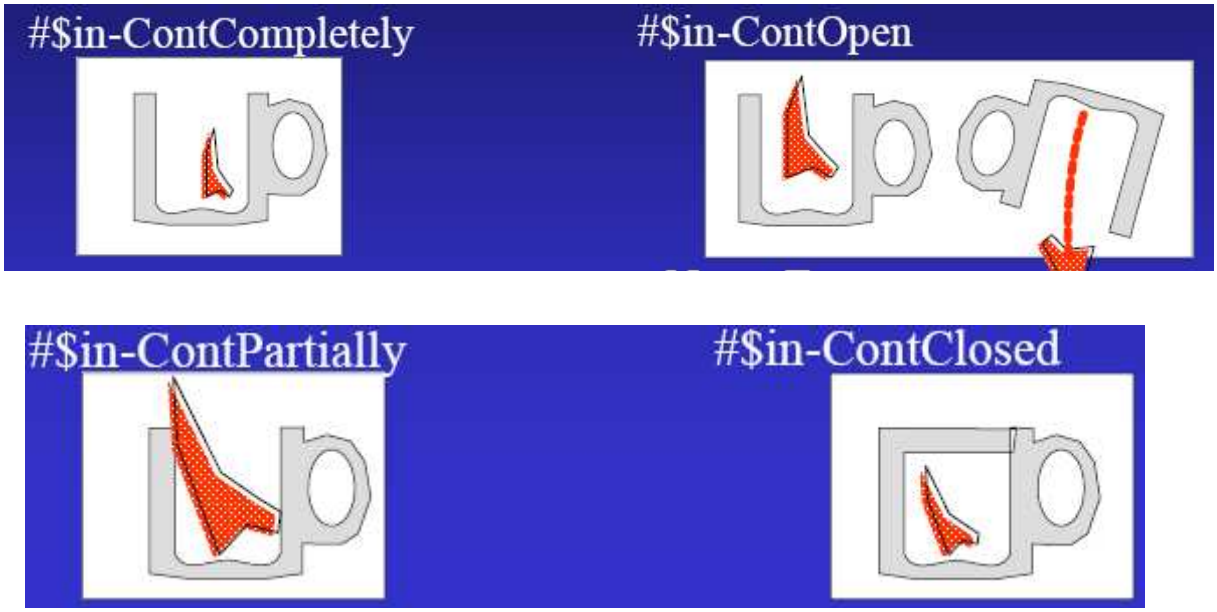

fig 3 Ejemplos de predicados "In"

En esta categoría hay cientos de entidades y propiedades, estando descritas en el vocabulario de OpenCyc [OCYCVOC].

5.2 Time and Date

Las entidades en esta categoría permiten expresar momentos e intervalos de tiempo y relaciones entre ellos.

Las relaciones definidas son:

- *temporalBoundsIntersect*
- *temporallyIntersects*
- *startsAfterStartingOf*

- *endsAfterStartingOf*
- *endsAfterEndingOf*
- *startingDate*
- *temporallyContains*
- *temporallyCooriginating*
- *temporalBoundsContain*
- *temporalBoundsIdentical*
- *startsDuring*
- *overlapsStart*
- *startingPoint*
- *simultaneousWith*
- *alter*
- *includedInIntervalType*
- *subsumesIntervalType*
- *intersectsIntervalType*
- *subsumedByIntervalType*
- *followingIntervalType*

5.3 Device

Aunque existe una categoría device esta engloba cualquier objeto físico, existiendo subclases para muchos tipos de objetos (aperos de labranza, vehículos, dispositivos electrónicos…). Esta categoría es demasiado amplia y poco específica para poder modelar la taxonomía de dispositivos de un entorno de trabajo inteligente.

6 BASIC GEO

Es un vocabulario RDF básico creado por la W3C [GEO] que permite expresar la latitud y la longitud en formato WGS84. Las clases del vocabulario son:

- *SpacialThing*: Cualquier cosa con extensión espacial.
- *Point*: un punto expresado mediante coordenadas.

Las propiedades son:

- *Latitude*: En grados decimales.
- *Longitude*: En grados decimales.
- *Altitude*: Metros decimales sobre el elipsoide de referencia local.
- Lat/*long*: representación separada por comas.

7 MOGATU BDI

MoGATU BDI [BDI] es una ontología usada para representar a usuarios y agentes software inteligentes. La ontología contempla las siguientes entidades:

- *Agent*: Cualquier usuario o agente software inteligente. Permite indicar sus *believes* (los asertos que forman el conocimiento del agente), *desires* (las condiciones finales que quiere lograr) e *intentions* (el plan para lograr los deseos). También permite indicar si un agente tiene un objetivo.

- *Statement, Condition, Conditional Statement*: Declaraciones simples (siempre son verdaderas) o condicionales (que el agente puede elegir si creer o no dependiendo de la condición).

- *Action, Plan*: Un agente debe de efectuar acciones para cambiar de estado, las acciones definen una precondición, un efecto y una prioridad. La precondición puede ser simple o estar compuesta por múltiples *and* y *or*. Un plan es una secuencia ordenada de acciones.

- *Policy, Preference, Modality*: Un agente debe de seguir las políticas existentes mientras intenta lograr sus objetivos. Una política permite expresar precondiciones y poscondiciones para una acción, su modalidad (Right, Prohibition, Obligation, Dispensation) y si es una preferencia (política local de un agente).

- *Belief, Desire, Intention, Goal*: Son las creencias, deseos, intenciones y metas de un agente.

- *Priority*: permite a un agente especificar la importancia relativa de sus deseos, metas, intenciones, acciones, preferencias y políticas. Para hacer esto se utilizan tres relaciones: <*prio:lessImportantThan*>, <*prio:asImportantAs*> y <*prio:moreImportantThan*>.

- *Time*: MoGATU BDI también define el concepto de tiempo para permitir a los agentes especificar cuando y cada cuanto tiempo realizar una acción.

8 SOUPA

SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications) [SOUPA] es un conjunto de ontologías orientadas a aplicaciones de pervasive computing. Se compone de dos documentos: SOUPA Core and SOUPA Extension. SOUPA Core define un vocabulario que es universal para todo tipo de aplicaciones relacionadas con pervasive computing. SOUPA Extension esta pensado para modelar aplicaciones específicas.

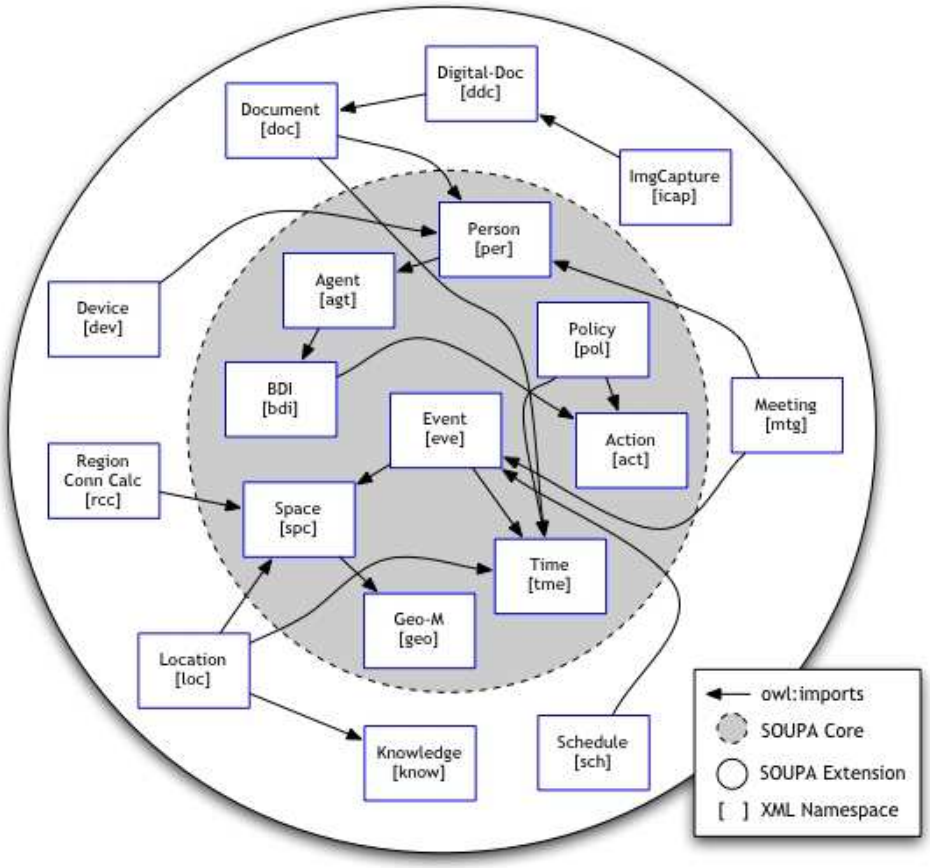

fig 4 SOUPA Core y SOUPA Extension

Parte del vocabulario de SOUPA ha sido extraído de otras ontologías, pero en vez de importarlas directamente se mappean los términos usando *owl:equivalentClass* y *owl:equivalentOntology*. Al hacerlo de esta manera se limita la sobrecarga de importar la ontología completa cuando sólo se van a utilizar algunos términos de la misma.

SOUPA se compone de dos conjuntos de ontologías: SOUPA Core y SOUPA Extension. El núcleo pretende definir un vocabulario genérico universal para aplicaciones de pervasive computing. Las extensiones definen vocabulario adicional para aplicaciones específicas.

8.1 SOUPA Core

El núcleo tiene ontologías para modelar las siguientes características: Personas, agentes, BDI (belief-desire-intention), acciones, políticas, tiempo, espacio y eventos.

8.1.1 Personas

Esta ontología define el vocabulario necesario para expresar la información de contacto y el perfil de una persona. Es equivalente a *foaf:Person*, por lo que todas las propiedades aplicables en FOAF (ver sección 2) también lo son en SOUPA.

```
<per:Person>
  <per:firstName
     rdf:datatype="&xsd;string">Harry</per:firstName>
  <per:lastName
     rdf:datatype="&xsd;string">Chen</per:lastName>
  <per:gender rdf:resource="&per;Male"/>
  <per:birthDate
     rdf:datatype="&xsd;date">1976-12-26</per:birthDate>

  <per:homepage
     rdf:resource="http://umbc.edu/people/hchen4"/>
  <foaf:weblog
     rdf:resource="http://umbc.edu/people/hchen4"/>

  <per:hasSchoolContact rdf:resource="#SchoolContact"/>
  <per:hasHomeContact rdf:resource="#HomeContact"/>

  <foaf:workplaceHomepage
     rdf:resource="http://ebiquity.umbc.edu"/>
  <foaf:workplaceHomepage
     rdf:resource="http://www.umbc.edu"/>
  <foaf:workplaceHomepage
     rdf:resource="http://www.cs.umbc.edu"/>
</per:Person>
```

8.1.2 Acción

Esta ontología permite definir acciones: quien la realiza, quien la recibe, el objeto implicado, la localización, la hora en la que se ejecutará, el instrumento que usará el actor… No es necesario que todas las propiedades contengan algún valor.

8.1.3 Política

Permite representar políticas de seguridad y privacidad así como un mecanismo basado en

lógica descriptiva para razonar sobre ellas. Esta ontología se encuentra influenciada por el lenguaje de políticas Rei [REI]. Estas políticas se aplican a las acciones definidas con la ontología anterior, indicando que se permite o deniega. Esta ontología también permite definir metadatos sobre la política creada (quien la creó, cuándo…)

```
<owl:Class rdf:ID="ShareHarryLocInfoWithEBMembers">
    <owl:intersectionOf rdf:parseType="Collection">
       <owl:Class rdf:about="&act;Action"/>

       <owl:Restriction>
         <owl:onProperty rdf:resource="&act;actor">
         <owl:hasValue>
            <agt:Agent rdf:about="ctb@cobra1.cs.umbc.edu"/>
         </owl:hasValue>
       </owl:Restriction>

       <owl:Restriction>
          <owl:onProperty rdf:resource="&act;target"/>
          </owl:allvaluesFrom
            rdf:resource="#LocationContextOfHarry"/>
       </owl:Restriction>

       <owl:Restriction>
          <owl:onProperty rdf:resource="&act;recipient"/>
          <owl:allValuesFrom
             rdf:resource="&eb;EbiquityMembers"/>
       </owl:Restriction>
    </owl:intersectionOf>
</owl:Class>
```

8.1.4 BDI

Esta ontología permite modelar el “estado mental” de los agentes software y usuarios del sistema mediante belief-desire-intention basándose en MoGATU BDI. Permite definir hechos (una subclase de *rdf:statement*) y planes, que son una subclase de *act:action* con las propiedades extras de *bdi:precondition* y *bdi:effect*. Además los desires de un agente pueden no ser realizables o estar en conflicto entre ellos.

8.1.5 Agent

Representa a un agente con las propiedades *agt:relieves*, *agt:desires*, *agt:inteds*.

8.1.6 Time

Esta ontología permite definir instantes, intervalos y relaciones entre ellos. Puede ser usada para describir las propiedades de diferentes eventos que ocurran. Esta ontología hace uso del vocabulario de DAML-time (ver sección 4). Para expresar el tiempo se hace uso del tipo de datos *xsd:dateTime*.

```
<tme:from>
  <tme:TimeInstant>
    <tme:at rdf:datatype="xsd;dateTime">
      2004-02-01T12:01:01
    </tme:at>
  </tme:TimeInstant>
</tme:from>
<tme:to>
  <tme:TimeInstant>
    <tme:at rdf:datatype="xsd;dateTime">
      2004-02-11T13:41:21
    </tme:at>
  </tme:TimeInstant>
</tme:to>
</tme:TimeInterval>
```

Para describir las relaciones entre dos instantes la ontología define las siguientes propiedades: *tme:before, tme:after, tme:beforeOrAt, tme:afterOrAt*, y *tme:sameTimeAs*. Para describir las relaciones entre objetos temporales diversos (instantes, intervalos…) las propiedades definidas son: *tme:startsSoonerThan, tme:startsLaterThan, tme:startsSameTimeAs, tme:endsSoonerThan, tme:endsLaterThan, tme:endsSameTimeAs, tme:startsAfterEndOf,* and *tme:endsBeforeStartOf.* En próximas versiones se espera poder soportar el vocabulario de RDF Calendar para poder definir sucesos recurrentes.

8.1.7 Space

Esta ontología está diseñada para permitir el razonamiento que implique relaciones espaciales entre varios tipos de regiones geográficas, expresadas tanto por valores simbólicos como por coordenadas geoespaciales. Adopta parte del vocabulario de OpenCyc (ver sección 5) y de OpenGIS [OGIS]. Aunque soporta coordenadas en forma de latitud y longitud no soporta el uso de coordenadas dentro de habitaciones o edificios que pudieran ser extraídas de un sistema de localización indoors.

8.1.8 Event

Permite modelar actividades que tienen componentes espacio-temporales, por ejemplo el descubrimiento de un dispositivo Bluetooth a una hora en cierto lugar.

```
<owl:Class rdf:ID="DetectedBluetoothDev">
  <rdfs:subClassOf
   rdf:resource="&eve;TemporalSpatialEvent"/>
</owl:Class>

<owl:ObjectProperty rdf:ID="foundDevice">
  <rdfs:domain
  rdf:resource="#DetectedBluetoothDev"/>
</owl:ObjectProperty>

<DetectedBluetoothDev>
  <spc:hasCoordinates>
    <geo:LocationCoordinates>
      <geo:longitude rdf:datatype...>
        -76.7113
      </geo:longitude>
      <geom:latitude rdf:datatype...>
        39.2524
      </geom:latitude>
    </geo:LocationCoordinates>
  </spc:hasCoordinates>

  <foundDevice rdf:resource="url-x-some-device"/>
  <tme:at>
    <tme:TimeInstant>
      <tme:at rdf:datatype="xsd;dateTime">
        2004-02-01T12:01:01
      </tme:at>
    </tme:TimeInstant>
  </tme:at>
<DetectedBluetoothDev>
```

8.2 SOUPA Extension

Este conjunto de ontologías se han definido para cumplir con dos objetivos:

1. Definir un vocabulario extra para permitir el modelado de aplicaciones relacionadas con habitaciones para reuniones inteligentes.

2. Demostrar como se puede ampliar SOUPA Core.

Los vocabularios definidos son:

- *Meeting & Schedule.*
- *Document & Digital Document.*
- *Image Capture.*
- *Region Connection Calculus.*
- *Location.*

9 CONON

CONON [CONON] es una ontología para modelar el contexto en aplicaciones de pervasive computing creada para ser utilizada junto a SOCAM [SOCAM]. La ontología se divide en dos conjuntos de ontologías, el primero de ellos se encarga de capturar los conceptos generales comunes a todas las aplicaciones y el segundo es específico del dominio.

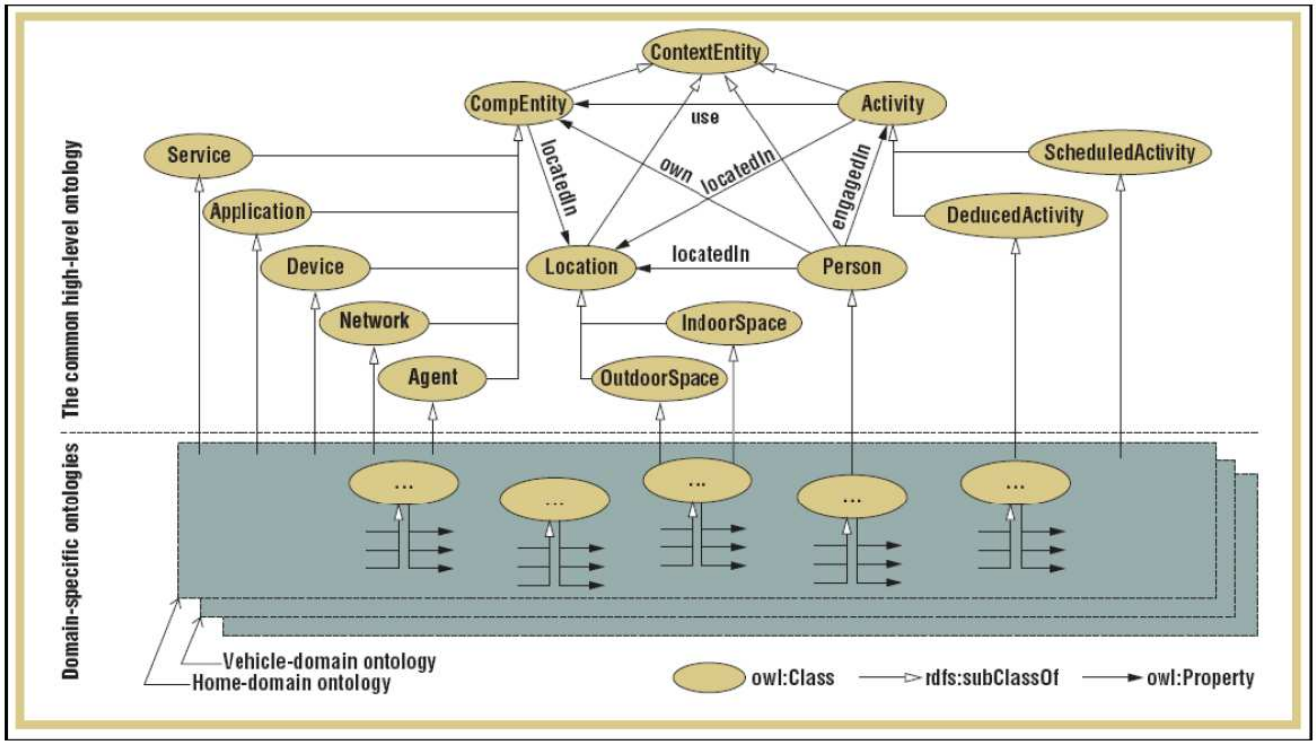

fig 5 Ontología general

Para la ontología general del contexto se definen cuatro entidades fundamentales:

- Localización
- Usuario
- Actividad
- Entidad Computacional

Un ejemplo de ontología específica del dominio sería la siguiente:

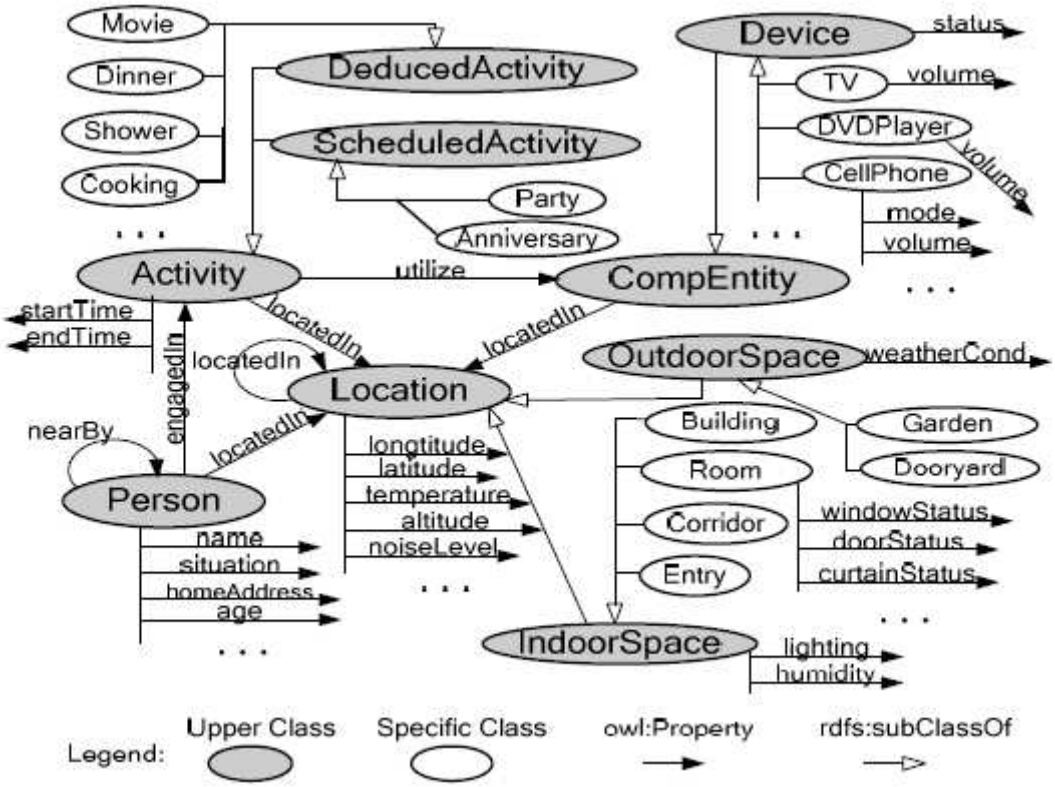

fig 6 Ontología específica del dominio Home

10 CODAMOS

La ontología [CODONT] creada para el proyecto CoDAMoS [CODAMOS] trata de crear un infraestructura adaptable y flexible para aplicaciones de Inteligencia Ambiental (AmI) sensibles al contexto. Para ello se centra en cuatro entidades principales: *User, Service, Platform* y *Enviroment*.

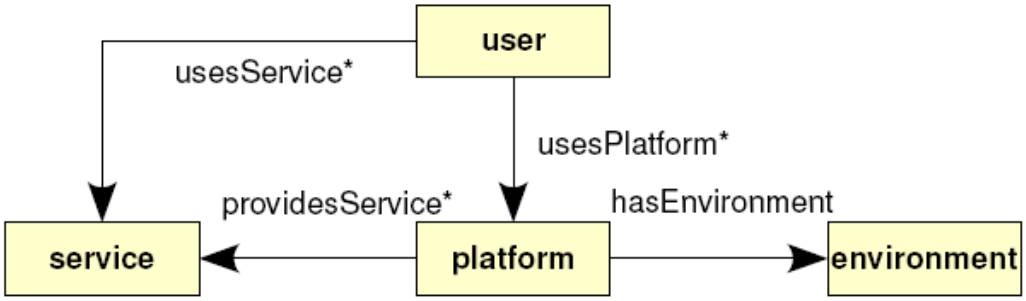

fig 7 Entidades principales

10.1 User

El usuario desempeña un papel fundamental en AmI, siendo los dispositivos los que se tienen que adaptar al usuario y no al revés. En esta ontología la información del contexto sólo es importante si afecta al usuario. La entidad User tiene propiedades como *preferentes* (que cambia con la situación), *profile* (que se mantiene en el tiempo), *task* (que se puede dividir en *activities*), role y *mood* (que afecta a las preferencias).

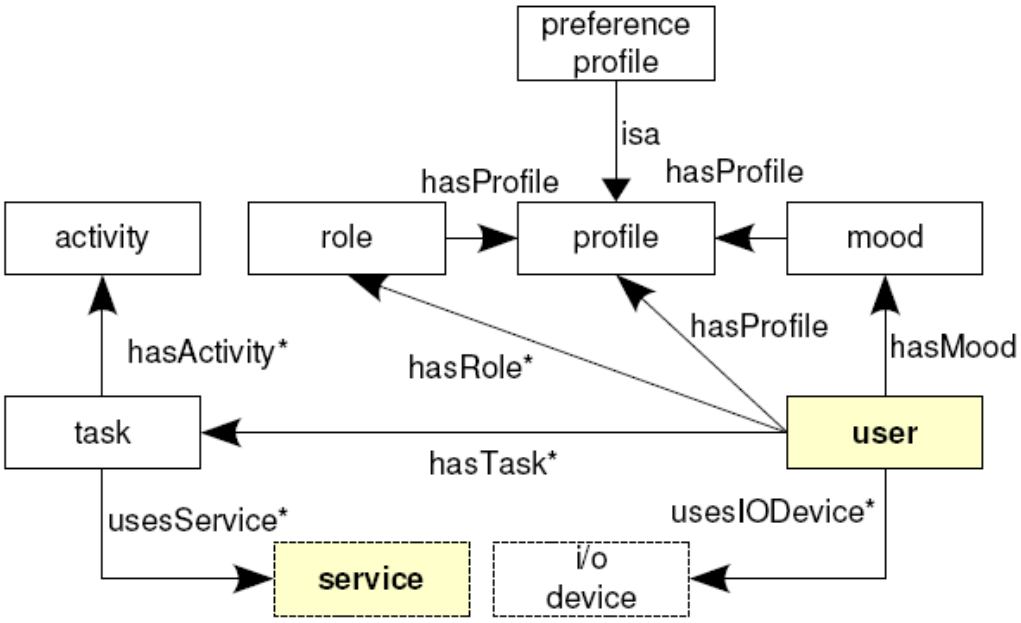

fig 8 Conceptos de la ontología User

10.2 Enviroment

Es importante que el usuario tenga información sobre el entorno ya que puede influenciar las decisiones a tomar. Dado que es imposible describir todo lo que rodea a un usuario CoDAMoS distingue tres aspectos importantes del entorno: *location*, *time* y *enviromental conditions* (meteorología, temperatura…).

Un aspecto importante de la información del entorno es que puede ser obtenida de múltiples fuentes y estas pueden tener diferentes niveles de precisión. Por eso es necesario que esto quede reflejado en la ontología.

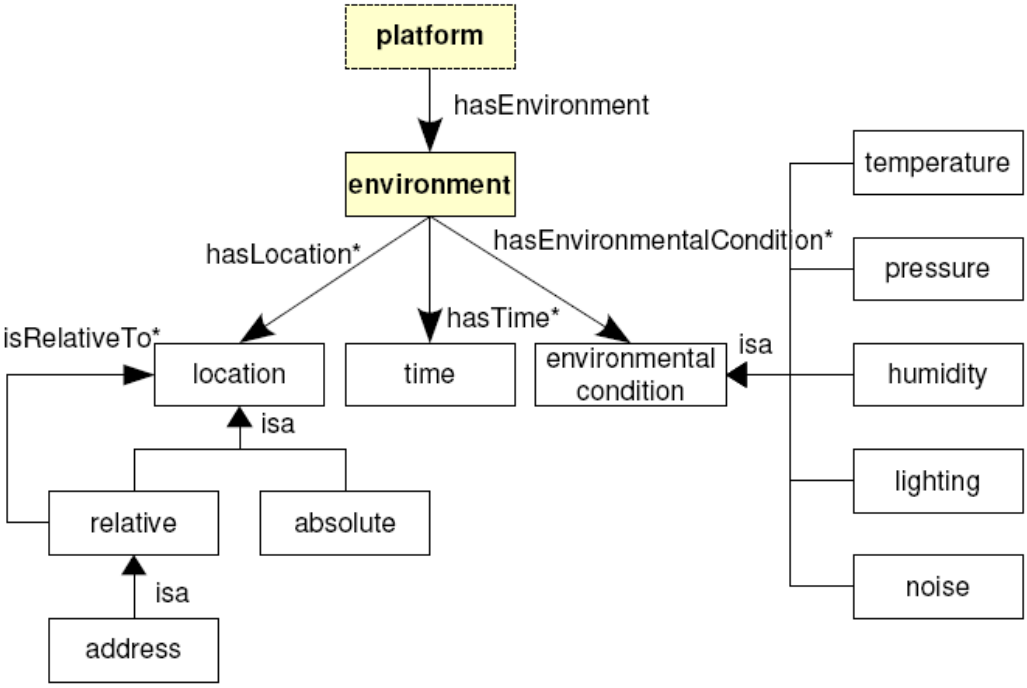

fig 9 Conceptos de la ontología Enviroment

10.3 Platform

Esta sección de la ontología describe tanto el software como el hardware que están disponibles para el usuario y servicios en un dispositivo. Es necesario conocer el software disponible en un dispositivo para poder inferir si un servicio puede ejecutarse en el y para poder construir servicios específicos para un dispositivo automáticamente.También es necesario conocer los recursos hardware de un dispositivo para poder conocer que capacidades tiene y que servicios se adaptan al mismo.

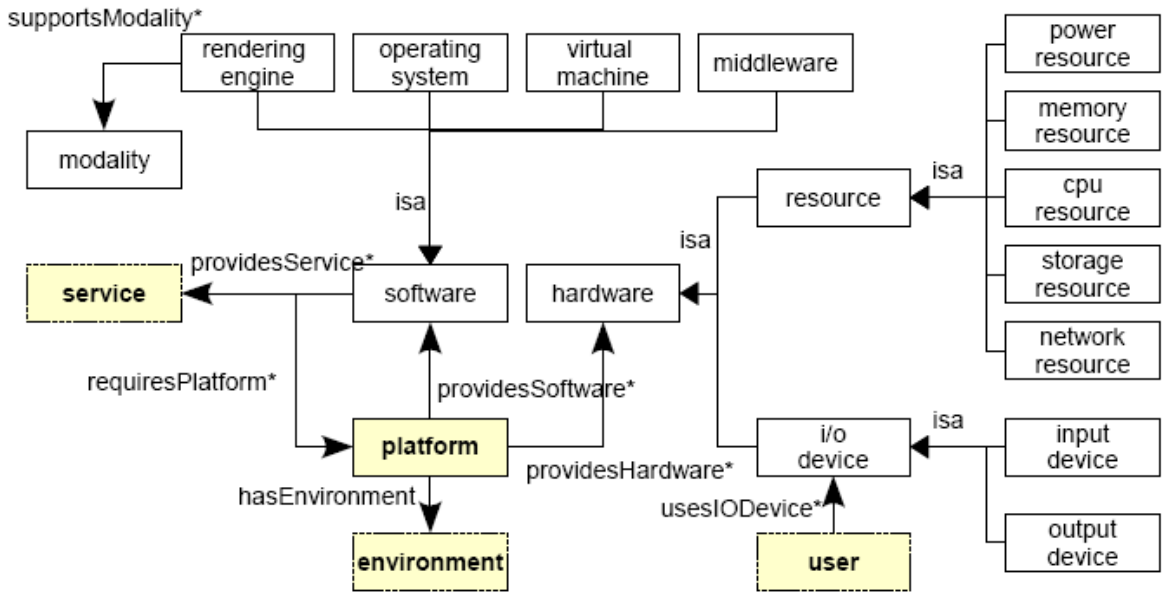

fig 10 Conceptos de la ontología Platform

10.4 Services

En esta ontología se describen entidades computacionales que ofrecen una funcionalidad específica. Para describir un servicio se usan tres aspectos del mismo:

- *Service profile*: Una descripción human-readable con la funcionalidad del servicio.
- *Service model*: Describe el workflow del servicio.
- *Service grounding*: detalles sobre la implementación del servicio que especifican el protocolo de comunicación, el formato de los mensajes…

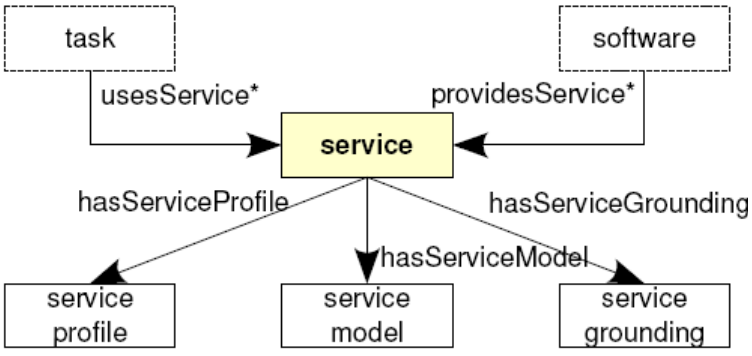

fig 11 Conceptos de la ontología Service

11 DIFERENCIAS EN EL CONTEXTO: SOUPA, CONON, CODAMOS

Las tres ontologías analizadas contienen elementos comunes en sus ontologías, así como vocabulario único de cada una:

	SOUPA	CONON	CoDAMoS
Elementos Principales	• Person • Agent • BDI • Policy • Event • Action • Time • Space	• Location • Person • Activity • Computing Entity	• User • Service • Platform • Location • Time • Enviromental condition

Tabla 1 Elementos principales de las ontologías

11.1 Elementos comunes

Hay ciertas entidades comunes en las tres ontologías:

- *Space/Location*: La localización es uno de los aspectos más importantes del contexto, tanto en interiores como en exteriores. Además es importante que el vocabulario permita establecer relaciones espaciales para poder inferir nuevo contexto a partir de ellas.

- *Time*: Tanto SOUPA como CoDAMoS definen vocabulario para expresar instantes y periodos de tiempo. Además SOUPA define relaciones temporales que permiten establecer un workflow de las diferentes acciones. CoDAMoS tiene implícitas este tipo de relaciones en el workflow del servicio, no siendo tan versátil.

- *Person/User*: Dado que para AmI la persona es el centro del sistema las tres ontologías permiten modelar a los usuarios. Tanto SOUPA (con BDI) como CoDAMoS (con la propiedad *mood*) permiten además modelar el estado mental de los usuarios.

- *Action/Activity*: SOUPA y CONON definen un vocabulario para poder expresar las acciones que realizarán los usuarios.

- *Computing entity/Platform*: CONON y CoDAMoS permiten definir los dispositivos que

se pueden encontrar en el entorno. CoDAMoS se centra en sus características hardware y software pero define una entidad *Service* para poder modelar funcionalidades.

11.2 Elementos únicos

Hay elementos que sólo se encuentran en una ontología:

- *Policy*: Este es un elemento único de SOUPA que permite definir políticas de seguridad y privacidad.
- *Event*: Este elemento de SOUPA permite modelar eventos que ocurren en el sistema.
- *Service*: Este elemento de CoDAMoS permite definir las funcionalidades de un servicio.
- *Enviromental Condition*: Sólo CoDAMoS contempla otras características del contexto como temperatura, condiciones meteorológicas, ruido, iluminación…

12 RAZONAMIENTO SOBRE EL CONTEXTO

Tanto SOCAM [SOCAM] como CoBrA [COBRA] definen dos actividades indispensable a la hora de razonar sobre el contexto: comprobar incosistencias e inferir nueva información sobre el contexto a partir de la ya conocida.

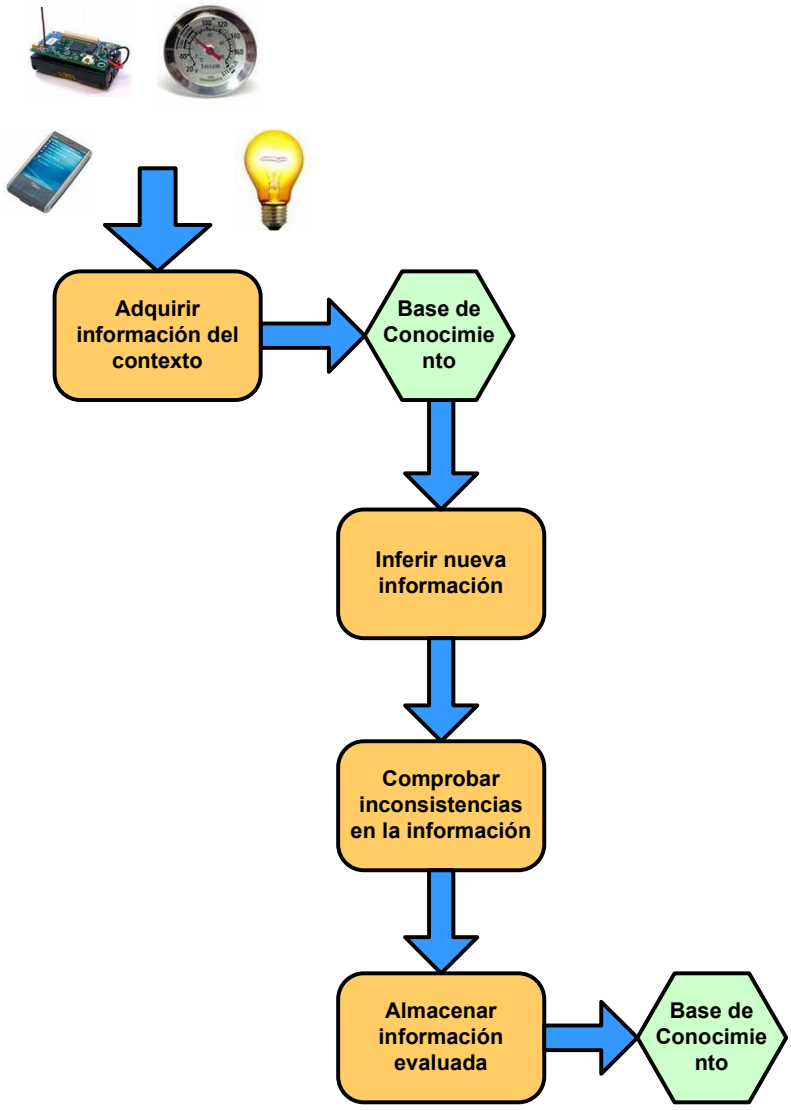

fig 12 Razonamiento sobre el contexto

Además en ambos proyectos el servicio central de razonamiento de contexto no se encarga de ejecutar acciones, sólamente se asegura de mantener actualizada la información del contexto creando nuevo contexto explicito en base al contexto implicito.

La labor de realizar acciones en base al contexto recae sobre otros servicios, agentes o

aplicaciones que usan el servicio central, teniendo estos reglas para sus dominios específicos.

Application	Rule
Healthcare	**If** (John's blood pressure exceeds the threshold) ∨ (John's heartbeat is abnormal) ∨ (socam:temperature(John, greaterThan(101F) **Then** alert hospital emergency department
Memory aid	**If** socam:status(E-prescription, VALID) ∧ (socam:time(Local, XX:YY) matches the time indicated in the E-prescription) **Then** prompt to take medicines
Energy saving	**If** (¬socam:locatedIn(John, Room) ∧ socam:hasLightinglevel(Room, HIGH) **Then** turn off the light

fig 13 Reglas para acciones específicas de cada dominio de varios servicios de SOCAM

12.1 Inferir nuevo contexto

Tanto CoBrA como SOCAM infieren nuevo contexto de dos maneras: Utilizando las relaciones semánticas definidas en OWL (lógica descriptiva) y con reglas heurísticas asociadas a cierto dominio.

12.1.1 Razonamiento basado en relaciones semánticas

OWL define una serie de relaciones semánticas entre clases y propiedades:

- SubclassOf: `(?a rdfs:subClassOf ?b) and (?b rdfs:subClassOf ?c) implies (?a rdfs:subClassOf ?c)`
- DisjointWith: `(?C owl:disjointWith ?D) and (?X rdf:type ?C) and (?Y rdf:type ?D) implies (?X owl:differentFrom ?Y)`
- Symmetric propierty: `P(x,y) iff P(y,x)`
- Functional propierty: `P(x,y) and P(x,z) implies y = z`
- InverseOf: `P1(x,y) iff P2(y,x)`

- Inverse functional propierty: `P(y,x) and P(z,x) implies y = z`
- Transitive propierty: `P(x,y) and P(y,z) implies P(x,z)`
- Etc…

Un ejemplo del uso de estas relaciones sería marcar la relación *esUn* como transitiva, teniendo los triples *esUn (Usuario, Humano)* y *esUn(Humano, Animal)* se podría inferir nueva información, con el resultado de *esUn(Usuario, Animal)*. SOCAM hace uso de estas propiedades de manera muy parecida como puede verse en la siguiente tabla:

INPUT	DL Reasoning Rules	(?P rdf:type owl:TransitiveProperty) $\wedge$ (?A ?P ?B) $\wedge$ (?B ?P ?C) $\Rightarrow$ (?A ?P ?C) (?P owl:inverseOf ?Q) $\wedge$ (?X ?P ?Y) $\Rightarrow$ (?Y ?Q ?X)
	Explicit Context	<owl:ObjectProperty rdf:ID="locatedIn"> <rdf:type="owlTransitiveProperty"/> <owl:inverseOf rdf:resource="#contains"/> </owl:ObjectProperty> <Person rdf:ID="Wang"> <locatedIn rdf:resource="#Bedroom"/> </Person > < Room rdf:ID="Bedroom"> < locatedIn rdf:resource="#Home"/> </ Room>
OUTPUT	Implicit Context	<Person rdf:ID="Wang"> <locatedIn rdf:resource="#Home"/> </Person > <Building rdf:ID="Home"> < contains rdf:resource="#Bedroom"/> < contains rdf:resource="#Wang"/> </Building> <Room rdf:ID="Bedroom"> < contains rdf:resource="#Wang"/> </Room>

fig 14 Razonamiento sobre localización mediante el uso de una ontología

Evidentemente para poder hacer uso de estas caracteristicas de OWL es necesario que la ontología esté creada usando estas relaciones semánticas. Por ello es necesario diseñar la ontología con cuidado intentando maximizar el conocimiento que podrá ser inferido.

12.1.2 Reglas específicas del dominio

Definir reglas específicas del dominio permite mayor flexibilidad a la hora de inferir nuevo contexto. Ejemplos de estas reglas se pueden encontrar en CoBrA y SOCAM:

Situation	Reasoning Rules
Sleeping	(?u locatedIn Bedroom) ∧ (Bedroom lightLevel LOW) ∧ (Bedroom drapeStatus CLOSED) ⇒ (?u situation SLEEPING)
Shower-ing	(?u locatedIn Bathroom) ∧ (WaterHeater locatedIn Bathroom) ∧ (Bathroom doorStatus CLOSED) ∧ (WaterHeater status ON) ⇒ (?u situation SHOWERING)
Cooking	(?u locatedIn Kitchen) ∧ (ElectricOven locatedIn Kitchen) ∧ (ElectricOven status ON) ⇒ (?u situation COOKING)
Watching-TV	(?u locatedIn LivingRoom) ∧ (TVSet locatedIn LivingRoom) ∧ (TVSet status ON) ⇒ (?u situation WATCHINGTV)
Having-Dinner	(?u locatedIn DiningRoom) ∧ (?v locatedIn DiningRoom) ∧ (?u owl:differentFrom ?v) ⇒ (?u situation HAVINGDINNER)

fig 15 Reglas para inferir nuevo contexto en SOCAM

```
(?x tme:before ?y) <-
     (?x rdf:type tme:IntervalThing), (?x tme:ends ?endsX),
     (?y rdf:type tme:IntervalThing), (?y tme:begins ?beginsY),
     (?endsX tme:before ?beginsY).

(?x tme:inside ?y) <-
    (?x rdf:type tme:InstantThing),
    (?y rdf:type tme:IntervalThing),
    (?y tme:begins ?beginsY), (?y tme:ends ?endsY),
    (?beginsY tme:before ?x), (?x tme:before ?endsY).

(?x tme:overlaps ?y) <-
    (?x rdf:type tme:ProperIntervalThing),
    (?y rdf:type tme:ProperIntervalThing),
    (?x tme:begins ?beginsX), (?x tme:ends ?endsX),
    (?y tme:begins ?beginsY), (?y tme:ends ?endsY),
    (?beginsY tme:before ?endsX),
    (?beginsX tme:before ?beginsY),
    (?endsX tme:before ?endsY).

(?x tme:overlappedBy ?y) <- (?y tme:overlaps ?x).
```

fig 16 Reglas para inferir nuevo contexto temporal en CoBrA

13 REFERENCIAS

[FOAF] FOAF project. Online, accedida el 9/may/2007. URL: http://www.foaf-project.org/

[FOAFESP] Especificación de FOAF. Online, accedida el 9/may/2007. URL: http://xmlns.com/foaf/0.1/

[GEEK] GeekCode. Online, accedida el 9/may/2007. URL: http://www.geekcode.com/geek.html

[MYBRI] Clasificación de la personalidad Myer-Briggs. Online, accedida el 9/may/2007. URL: http://www.myersbriggs.org/

[DCES] Dublin Core Metadata Element Set, Version 1.1 (RDF). Online, accedida el 10/may/2007. URL: http://dublincore.org/documents/dces/

[DAMLT] DAML Time ontology. Online, accedida el 10/may/2007. URL: http://www.cs.rochester.edu/~ferguson/daml/

[SUBONT] Entry sub-ontology if DAML Time. Online, accedida el 10/may/2007. URL: http://www.isi.edu/~pan/damltime/time-entry.owl

[GEO] Basic GEO. Online, accedida el 10/may/2007. URL: http://www.w3.org/2003/01/geo/

[OCYC] OpenCyc. Online, accedida el 11/may/2007. URL: http://www.opencyc.org/

[CYC] CycCorp. Online, accedida el 11/may/2007. URL: http://www.cyc.com/

[OCYCVOC] OpenCyc Vocabulary. Online, accedida el 11/may/2007. URL: http://www.cyc.com/cycdoc/vocab/vocab-toc.html

[BDI] MoGATU BDI. Online, accedida el 14/may/2007. URL: http://mogatu.umbc.edu/bdi/

[SOUPA] Harry Chen, Filip Perich, Tim Finin, and Anupam Joshi. SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications, In Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous 2004), Boston, MA, August 22-26, 2004.

[REI] L. Kagal, T. Finin, and A. Joshi. A Policy Based Approach to Security for the Semantic Web. In 2nd International Semantic Web Conference (ISWC2003),

September 2003.

[OGIS] S. Cox, P. Daisey, R. Lake, C. Portele, and A. Whiteside. Geography Markup Language (GML 3.0). In OpenGIS Documents. OpenGIS Consortium, 2003.

[CONON] Wang XH, Zhang DQ, Gu T, Pung HK. Ontology Based Context Modeling and Reasoning using OWL. Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004.

[SOCAM] Tao Gu, Hung Keng Pung, Da Qing Zhang. Toward an OSGi-Based Infrastructure for Context-Aware Applications. Pervasive Computing, 2004.

[CODONT] Davy Preuveneers, Jan Van den Bergh, Dennis Wagelaar, Andy Georges, Peter Rigole, Tim Clerckx, Yolande Berbers, Karin Coninx, Viviane Jonckers, Koen De Bosschere. Towards an Extensible Context Ontology for Ambient Intelligence. Proceedings of Ambient Intelligence: Second European Symposium, EUSAI 2004.

[CODAMOS] The CoDAMoS Project: Context-Driven Adaptation of Mobile Services. Online, accedida el 17/may/2007. URL: http://www.cs.kuleuven.ac.be/distrinet/projects/CoDAMoS/ (2003).

[COBRA] H. Chen. An Intelligent Broker Architecture for Pervasive Context-Aware Systems. PhD thesis, University of Maryland, Baltimore County, 2004.