

## Una ontología para representar el contexto en sistemas de computación académicos universitarios

Lenin Erazo<sup>a</sup>, Lourdes Illescas<sup>b</sup>, Mauricio Espinoza<sup>b</sup>

<sup>a</sup> Escuela de Ingeniería de Sistemas, Universidad del Azuay, Av. 24 de mayo 7-77, Cuenca, Ecuador  
lerazo@uazuay.edu.ec

<sup>b</sup> Departamento de Ciencias de la Computación, Universidad de Cuenca, Av. 12 de abril, Ciudadela Universitaria, Cuenca, Ecuador  
{lourdes.illescasp, mauricio.espinoza}@ucuenca.edu.ec

**Resumen.** El uso de dispositivos móviles, sean estos teléfonos inteligentes o tabletas, está propiciando una nueva generación de aplicaciones que explotan la información contextual del usuario y de los demás elementos involucrados. Estas aplicaciones son útiles en múltiples dominios, como por ejemplo el ambiente académico universitario, el cual requiere de sistemas de información “inteligentes”, que tomen en consideración los elementos del entorno como personas, lugares, recursos didácticos, actividades, etc., para enriquecer la experiencia del usuario, mediante la provisión de información y/o servicios relevantes con su situación actual. Una de las actividades en el desarrollo de este tipo de aplicaciones es definir un modelo que represente de manera formal la información contextual. A pesar de que en la literatura existen múltiples propuestas para modelar el contexto usando modelos semánticos formales (ontologías), aún no es posible identificar un modelo estándar que pueda ser utilizado en diferentes aplicaciones de dominio. Este trabajo describe el proceso metodológico utilizado durante la construcción de una ontología de contexto para el dominio académico universitario, así como el modelo ontológico resultante de este proceso.

**Palabras Clave:** ontología, sensibilidad al contexto, computación ubicua, sistema académico universitario.

### 1 Introducción

En la actualidad los dispositivos móviles, sean teléfonos inteligentes o tabletas, han sido adoptados en forma mayoritaria por la sociedad. Se puede apreciar este hecho mirando a nuestro alrededor cómo las personas hacen uso de estos dispositivos, casi en cualquier lugar, para realizar cada vez más operaciones. La razón por la que los dispositivos móviles han tenido gran aceptación es la evolución y la convergencia tecnológica, específicamente, la mejora considerable de la infraestructura de comunicaciones en velocidad, fiabilidad y cobertura; la reducción del tamaño y el aumento del rendimiento de los dispositivos móviles; la integración de sensores e identificadores digitales a estos dispositivos; y, la presencia de aplicaciones informáticas (software) cada vez más avanzadas, lo que hace

factible que las personas puedan resolver una mayor cantidad de actividades tanto personales como de negocios desde casi cualquier lugar y circunstancia.

Tal como lo había anticipado Weiser [1], con la popularidad de los dispositivos móviles se está viviendo un periodo de transición desde la era de la computadora personal hacia una nueva era denominada la computación ubicua, la cual se perfila como la computación del futuro y, cuyo propósito es la integración de dispositivos inteligentes alrededor de todos los escenarios donde interactúa el ser humano.

El contexto y el conocimiento del entorno son componentes claves en la computación ubicua, siendo los modelos de representación de contexto insumos fundamentales para el diseño e implementación de aplicaciones sensibles al contexto. Sin embargo en la actualidad la programación de este tipo de aplicaciones es compleja y laboriosa debido a la utilización de modelos tradicionales para la representación del contexto, que carecen de formalidad, extensibilidad, expresividad y semántica, a su vez no permiten modelar las relaciones complejas ni los aspectos de temporalidad que existen en la información contextual, y no facilitan la estandarización e interoperabilidad entre las diversas fuentes de contexto (sensores, bases de datos, agentes, servicios), de tal forma que puedan compartir los datos de contexto e inferir información contextual implícita de más alto nivel. Una de las alternativas que mejor se adapta a las características de los datos contextuales y que se perfila como la más idónea es el uso de modelos semánticos formales para construir colecciones de ontologías que permitan modelar el contexto y capturar su conocimiento.

A pesar de que en la literatura es posible identificar una gran variedad de ontologías que modelan el contexto bajo diferentes enfoques [2, 3, 4, 5, 6, 7, 8] no es posible hablar aún de un modelo consensuado que pueda ser utilizado ampliamente para la modelación del contexto en múltiples aplicaciones. Además, de los modelos ontológicos existentes sólo unos pocos están disponibles para ser estudiados en detalle y sobre todo para poder ser reutilizados.

El presente trabajo tiene como objetivo describir el proceso utilizado para la construcción de un modelo ontológico que permita representar el conocimiento contextual de los elementos involucrados en un ambiente académico universitario, tales como: personas, lugares, recursos didácticos, actividades, etc. La modelación contextual de estos elementos puede influenciar positivamente en aquellos sistemas de computación que proveen de información o servicios académicos en función de las condiciones del entorno.

Para alcanzar este propósito en la sección 2 se comienza con un breve estudio de los términos ontología, contexto y sensibilidad al contexto como paso previo a la descripción de los trabajos existentes relacionados con la representación del contexto usando ontologías (ver sección 3). En la sección 4, se describe el proceso metodológico usado en la construcción de la ontología de contexto, así como el modelo ontológico resultante de este proceso. Finalmente, en la sección 5 se presentan las conclusiones y líneas de trabajo futuro.

## **2 Conceptos básicos**

En esta sección se describen los términos relacionados con el trabajo propuesto. Primero se define brevemente el término ontología desde la perspectiva de las ciencias de la computación, y la terminología fundamental usada para describir una ontología. Luego se describen los términos contexto y sensibilidad al contexto, usados para entender los tipos de contexto y el alcance de las entidades empleadas para representar el contexto.

## 2.1 Ontología

Una ontología como se define en [9], es una especificación formal y explícita de una conceptualización compartida. Aquí, *conceptualización* se refiere a un modelo abstracto de algún fenómeno en el mundo y a la identificación de los conceptos relevantes de ese fenómeno. *Explícita* significa que el tipo de conceptos que se utilizan, así como las restricciones de su uso, se definen de forma explícita. *Formal* hace referencia al hecho de que la ontología debe ser legible por una máquina. *Compartida* refleja la noción de que una ontología captura conocimiento consensuado, es decir, que no es privado, sino aceptado por un grupo.

Existen diferentes formalismos de representación del conocimiento para la definición de ontologías. Sin embargo, la mayoría de trabajos comparten el siguiente conjunto mínimo de componentes:

- *Clases*: representan conceptos que se toman en un sentido amplio, es decir, que pueden representar conceptos abstractos o conceptos específicos. Las clases de la ontología por lo general se organizan en taxonomías donde es posible aplicar el mecanismo de herencia.
- *Relaciones*: representan un tipo de asociación entre los conceptos de dominio. Las ontologías por lo general contienen relaciones binarias. El primer argumento se conoce como el dominio de la relación, y el segundo argumento es el rango. Las relaciones binarias se utilizan a veces para expresar atributos de un concepto. Los atributos se suelen distinguir de las relaciones porque su alcance es un tipo de dato, tales como: cadena de texto, numérico, etc., mientras que el rango de una relación es un concepto.
- *Instancias*: se utilizan para representar elementos o individuos en una ontología.

El lenguaje propuesto por la W3C<sup>1</sup> para representar explícitamente el significado de los términos y sus relaciones en una ontología, es OWL [10]. Este lenguaje provee tres sub-lenguajes: OWL Lite, OWL DL y OWL Full, cada uno de los cuales agrega más expresividad a los términos. La mayor parte de las ontologías pueden ser descritas usando la terminología propuesta en OWL Lite. Para facilitar el entendimiento de las siguientes secciones, en lo que respecta a la estructura de las ontologías analizadas y de la ontología propuesta en este trabajo, a continuación se describe brevemente la terminología fundamental usada por el lenguaje OWL:

---

<sup>1</sup> El Consorcio WWW, en inglés: World Wide Web Consortium, es un consorcio internacional que genera recomendaciones y estándares que aseguran el crecimiento de la World Wide Web a largo plazo.

- *owl:Class*: permite definir una clase, la cual posee un grupo de instancias que comparten algunas propiedades.
- *rdfs:subClassOf*: permite definir jerarquías de clases.
- *rdf:Property*: las propiedades se pueden utilizar para declarar relaciones entre instancias (*ObjectProperty*) o desde instancias a valores tipo dato (*DatatypeProperty*).
- *rdfs:subPropertyOf*: permite definir jerarquías de propiedades.
- *rdfs:domain*: el dominio de una propiedad limita las instancias sobre las cuales una propiedad puede ser aplicada.
- *rdfs:range*: el rango de una propiedad limita las instancias que la propiedad puede tener como valor.

Aquellos lectores interesados en estudiar las características adicionales que proveen los otros sub-lenguajes de OWL pueden consultar [11].

## 2.2 ¿Qué es contexto?

El término contexto fue introducido en el área de la computación por primera vez por Schilit y Theimer en 1994 [12], y lo definen como la ubicación, la identidad de la gente y de los objetos cercanos, así como los cambios sobre estos objetos. A partir de este momento en la literatura científica se encuentran innumerables definiciones de contexto, por ejemplo en el libro de Krumm [13] se realiza un interesante análisis comparativo de varias definiciones, las cuales se pueden clasificar en dos grupos. El primer grupo incluye definiciones enfocadas en enumerar ejemplos de las entidades que pueden ser parte del contexto, tales como: ubicación, identidad del usuario, actividad, temporalidad, estado emocional, objetos y personas en el entorno del usuario, etc. En el segundo grupo se incluyen definiciones sinónimas, que buscan una equivalencia con la palabra contexto, considerándolo como el entorno del usuario o el ambiente de la aplicación.

Una de las definiciones más aceptada por la comunidad científica es la de Dey [14], que define el contexto como: “*Cualquier información que puede ser usada para caracterizar la situación de una entidad. Una entidad puede ser una persona, un lugar o un objeto que es considerado relevante para la interacción entre el usuario y la aplicación misma*”.

Al ser el contexto un concepto amplio, un sinnúmero de entidades pueden formar parte del mismo, de ahí que se han propuesto varias formas de clasificarlo. Sin embargo, en base al análisis realizado por Poslad [15] se han identificado como principales los siguientes tipos de contexto:

- *Contexto geográfico (localización)*: denota la ubicación de un dispositivo, y en consecuencia la del usuario que lo utiliza. La ubicación puede ser en el interior de una edificación o en lugares externos, asimismo, puede ser lógica o física. La ubicación lógica utiliza datos discretos simbólicos (ej. en la biblioteca, aula 101, laboratorio 2, etc.), por lo que requiere de un conocimiento adicional para asignarle significado al mismo. Mientras que la ubicación física emplea sistemas de

posicionamiento como GPS. Adicionalmente se pueden manejar datos de proximidad de una entidad en particular como dispositivos o personas.

- *Contexto físico*: son consideradas como parte de este contexto las propiedades físicas de usuarios, edificios, objetos, dispositivos, redes, recursos y servicios que rodean a la aplicación, tales como: presión arterial, temperatura, nivel de ruido, humedad, intensidad de luz, velocidad de la red, disponibilidad de servicios, etc. La información de este contexto es proporcionada por sensores desplegados en un entorno particular.
- *Contexto de tecnologías de información (TI)*: abarca la infraestructura de TI tanto a nivel de software como hardware. Incluye aplicaciones y servicios informáticos, agentes inteligentes, dispositivo, redes, entre los más importantes.
- *Contexto temporal*: comprende cualquier tipo de información temporal: hora del día, fecha, semana, estación del año, calendario de actividades/eventos, etc.
- *Contexto de usuario*: incluye información relacionada con: identidad o perfil del usuario, preferencias, situación actual del usuario, actividad, dispositivos que lleva un usuario en cada momento con sus características y configuraciones, entre las más importantes.
- *Contexto sobre actividades/eventos*: describe la situación actual o futura (planificada) de un usuario. Por ejemplo, el profesor enseña a los estudiantes el manejo de un editor de texto como Word.
- *Contexto social*: describe las personas y las instituciones con las que un usuario se relaciona. En particular, describe grupos de gente como: familias, círculos de amigos, compañeros de trabajo, etc.
- *Contexto organizacional*: denota información de un contexto particular relacionado con los procesos de negocio. Un contexto organizacional puede ser información de un empleado referente a: institución en la que trabaja, cargo, responsabilidades, actividades y a quién reporta sus actividades.

Una vez identificadas las diferentes categorías del término contexto, a continuación se aborda la definición del término sensibilidad al contexto.

### 2.3 ¿Qué es sensibilidad al contexto?

Existen varias definiciones de computación sensible al contexto, según el análisis realizado en el libro de Krumm [13], se pueden clasificar en dos categorías. La categoría *using context* lo define como la capacidad de los dispositivos informáticos para detectar, interpretar y responder a los aspectos del ambiente local de un usuario y de los propios dispositivos de computación; mientras que la categoría *adapting context* lo define como aquellas aplicaciones que pueden cambiar dinámicamente o adaptar su comportamiento en base al entorno de sí misma y del usuario.

Una definición que engloba todo tipo de aplicaciones sensibles al contexto, desde aquellas que proporcionan información del contexto al usuario hasta las que brindan

servicios o se adaptan de acuerdo al contexto, es la Dey y Abowd en 2000 [16], que define a las aplicaciones sensibles al contexto como: “*Un sistema que utiliza la información del contexto para proveer información o servicios relevantes al usuario, donde la relevancia depende de la actividad del usuario*”.

Una vez definidos los términos relevantes para este trabajo, en la siguiente sección se abordan los trabajos relacionados con la definición de modelos de representación de contexto basados en ontologías.

### 3 Trabajos relacionados

Desde hace dos décadas aproximadamente, varias investigaciones se han realizado en la rama de la computación sensible al contexto, lo que demuestra que es una característica clave en las aplicaciones ubicuas y móviles. Particularmente en lo que respecta al modelado del contexto basado en ontologías se pueden destacar los siguientes trabajos de investigación:

#### 3.1 Ontology for context-aware pervasive computing environments (COBRA-ONT)

El trabajo de Chen, Finin y Joshi, en [2], es una colección de ontologías expresadas en OWL para describir información del contexto en espacios inteligentes. La versión 0.2 de COBRA-ONT, que se muestra en la Figura 1 está categorizada en cuatro temáticas relacionadas: i) ontología sobre lugares físicos, ii) ontología sobre agentes (esté termino abarca tanto a humanos como software), iii) ontología sobre el contexto de ubicación de los agentes, y iv) ontología para describir las actividades de los agentes.

**Ontología sobre lugares físicos.** Esta ontología permite modelar lugares físicos, y es capaz de representar en forma jerárquica qué lugares están contenidos en otros lugares. Por ejemplo, el campus de una universidad puede contener edificios, y a su vez estos últimos pueden contener aulas. La clase de más alto nivel es *Place* que representa un lugar físico; dispone de un conjunto de propiedades para describir una ubicación, tales como: *Longitude*, *Latitude* y *String Name*. Esta clase se subdivide en dos subclases *CompoundPlace* y *AtomicPlace* que representan los dos posibles tipos de lugares. La clase *CompoundPlace* puede contener otros lugares, mientras que la clase *AtomicPlace* no lo puede hacer.

Las propiedades de clase que permiten representar las relaciones de contención entre los lugares son: *SpatiallySubsumes* y *isSpatiallySubsumedBy*, y están definidas como propiedades inversas. En la clase *AtomicPlace* la propiedad *SpatiallySubsumes* es restringida a cardinalidad 0 y el rango de la propiedad *isSpatiallySubsumedBy* es limitado a *CompoundPlace*. En la clase *CompoundPlace* el rango de la propiedad *SpatiallySubsumes* es limitado a *Place* y el rango de la propiedad *isSpatiallySubsumedBy* es limitado a *CompoundPlace*.

CoBrA Ontology Classes		CoBrA Ontology Properties	
"Place" Related	Agents' Location Context	"Place" Related	Agent's Location Context
Place AtomicPlace CompoundPlace Campus Building AtomicPlaceInBuilding AtomicPlaceNotInBuilding Room Hallway Stairway OtherPlaceInBuilding	ThingInBuilding SoftwareAgentInBuilding PersonInBuilding ThingNotInBuilding SoftwareAgentNotInBuilding PersonNotInBuilding	latitude longitude hasPrettyName isSpatiallySubsumedBy spatiallySubsumes accessRestricted- ToGender lotNumber	locatedIn locatedInAtomicPlace locatedInRoom locatedInRestroom locatedInParkingLot locatedInCompoundPlace locatedInBuilding locatedInCampus
Restroom Gender LadiesRoom MensRoom ParkingLot	Agent's Activity Context	"Agent" Related	Agent's Activity Context
"Agent" Related	PresentationSchedule Event EventHappeningNow PresentationHappeningNow RoomHasPresentationHappeningNow ParticipantOfPresentation- HappeningNow SpeakerOfPresentationHappeningNow AudienceOfPresentationHappeningNow  PersonFillsRoleInPresentation PersonFillsSpeakerRole PersonFillsAudienceRole	hasContactInformation hasFullName hasEmail hasHomePage hasAgentAddress  fillsRole isFilledBy intendsToPerform desiresSomeone- ToAchieve	participatesIn  startTime endTime Location hasEvent hasEventHappeningNow invitedSpeaker expectedAudience presentationTitle presentationAbstract presentation eventDescription eventSchedule
Agent Person SoftwareAgent Role SpeakerRole AudienceRole IntentionalAction ActionFoundInPresentation			

Fig. 1. Lista completa de clases y propiedades de COBRA-ONT v0.2 [2]

Las subclases predefinidas de *AtomicPlace* son: *Room*, *Hallway*, *Stairway*, *Restroom* y *ParkingLot*, y las subclases predefinidas de *CompoundPlace* son: *Campus* y *Building*.

Para representar el lugar donde se puede realizar un evento (ej. una reunión) se ha definido la propiedad *Has-Event* cuyo dominio es *Place* y el rango es *Event*. Una instancia de la clase *Event* puede ser asociada a un intervalo de tiempo.

**Ontología sobre agentes.** La clase de más alto nivel es *Agent* y tiene dos subclases disjuntas denominadas *Person* y *SoftwareAgent* que representan a los agentes humanos y software respectivamente. Entre las propiedades que se han definido para describir el perfil de un agente, tenemos: *HasContactInformation*, *HasFullName*, *HasHomePage* y *HasAgentAddress*.

Cada agente tiene asociado un rol en un evento (ej. conferencista, participante, etc.). El rol de un participante se define con la propiedad *FillsRole* que tiene como dominio la clase *Agent* y como rango la clase *Role*. La ontología tiene para la clase *Role* predefinidas dos subclases: *SpeakerRole* y *AudienceRole*.

En la ontología la clase *Role* sirve para caracterizar las posibles intenciones de un agente, a través de la propiedad *IntendsToPerform* que tiene como dominio la clase *Role* y como rango la clase *IntentionalAction*. A su vez, un agente puede desear que otros agentes

realicen ciertas actividades para alcanzar sus objetivos propuestos. Para definir la acción que un agente desea que realice otro agente se tiene la propiedad *DesiresSomeoneToAchieve*.

**Ontología sobre el contexto de ubicación de los agentes.** El contexto de ubicación es una colección de conocimiento dinámico que describe la localización de los agentes en el tiempo. La propiedad *LocatedIn* representa la ubicación de un agente. Para determinar si un agente está ubicado en un *AtomicPlace* o *CompoundPlace*, se definen dos subpropiedades de *LocatedIn* denominadas: *LocatedInAtomicPlace* y *LocatedInCompoundPlace*. La primera propiedad es definida con un rango restringido a la clase *AtomicPlace* y la segunda con un rango restringido a la clase *CompoundPlace*. Adicionalmente, a partir de estas propiedades se restringe aún más el lugar físico en el que un agente puede estar, mediante la definición de las propiedades: *LocatedInRoom*, *LocatedInRestroom* y *LocatedInParkingLot* como subpropiedades de *LocatedInAtomicPlace*; y, *LocatedInCampus* y *LocatedInBuilding* como subpropiedades de *LocatedInCompoundPlace*.

A su vez, la ontología permite categorizar a los agentes dependiendo de su ubicación. Por ejemplo, se tienen las subclases predefinidas *PersonInBuilding* o *SoftwareAgentInBuilding* para representar a las personas o software que están localizadas en un edificio respectivamente.

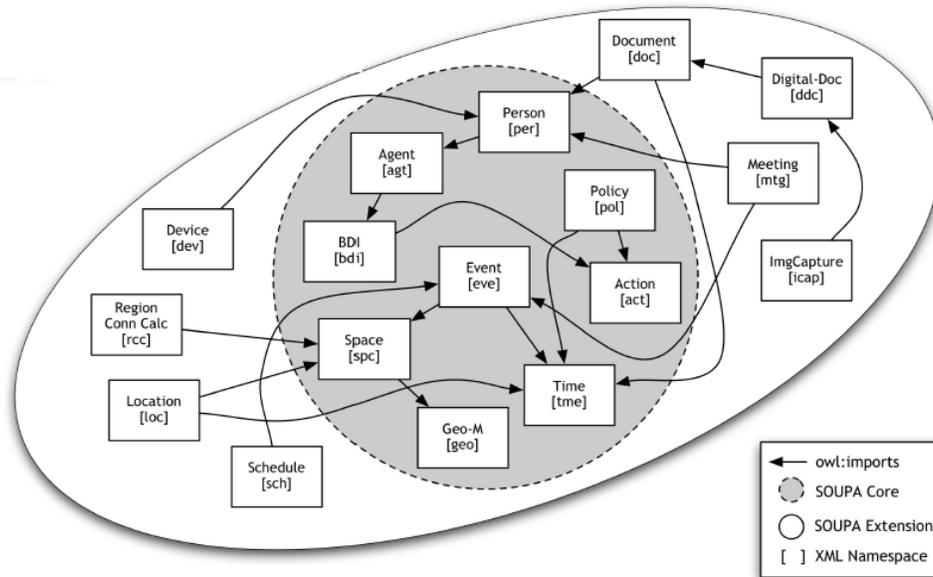
**Ontología sobre el contexto de la actividad de los agentes.** El contexto de la actividad describe los eventos en los que los agentes participan. La clase *PresentationSchedule* representa el calendario de un evento. Esta clase tiene asociadas propiedades para la fecha y hora de inicio *StartTime*, fecha y hora de fin *EndTime*, título de la presentación *PresentationTitle*, resumen de la presentación *PresentationAbstract* y localización del evento *Location*. A su vez, las propiedades *ExpectedAudience* y *InvitedSpeaker* describen a la audiencia y los conferencistas del evento, tienen como dominio la clase *PresentationSchedule* y como rango la clase *Person*.

Para describir un evento que actualmente se está realizando, se define la clase *PresentationEventHappeningNow*, asimismo se han definido las clases *RoomHasPresentationEventHappeningNow*, *SpeakerOfPresentationHappeningNow*, *AudienceOfPresentationHappeningNow* para representar el lugar en el que se realiza el evento, el conferencista y la audiencia que participan actualmente.

### 3.2 Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA)

La propuesta de Chen, Perich, Finin y Joshi [3] es una ontología construida en OWL, donde parte de su vocabulario se deriva de varias ontologías consensuadas, tales como: FOAF (Friend-Of-A-Friend Ontology) [17], DAML Time [18], OpenCyc Spatial Ontologies [19], RCC (Regional Connection Calculus), COBRA-ONT [20], MoGATU BDI Ontology [21] y Rei Policy Ontology [22].

SOUPA está conformado por dos subconjuntos interrelacionados de ontologías: SOUPA CORE y SOUPA EXTENSION, como se puede apreciar en la Figura 2.



**Fig. 2.** Lista de ontologías contenidas en SOUPA [3]

**SOUPA CORE.** Define un vocabulario genérico para modelar el contexto relacionado con los conceptos: personas, políticas de seguridad y privacidad, acciones, agentes, creencias-deseos-intenciones, tiempo, espacio y eventos. A continuación se analizan cada uno de estos conceptos:

**Person.** Esta ontología describe la información de contacto y el perfil de una persona. La clase principal es *per:Person* y es equivalente a la clase *foaf:Person*. Esta clase incluye propiedades que describen: i) el perfil de la persona (nombre, género, edad, fecha nacimiento, etc.); ii) información de contacto (correo electrónico, dirección, página Web del trabajo, teléfono, chat ID, etc.); y iii) perfil social y profesional (personas que conoce, grupos en los que es miembro, organización a la que pertenece, etc.). En definitiva todas las propiedades asociadas a la clase *Person* en la ontología *FOAF* pueden ser usadas.

**Policy & Action.** Provee un vocabulario para la representación de políticas de seguridad y privacidad sobre las acciones que se realicen. El vocabulario se basa en la ontología *Rei Policy Language*.

La clase *act:Action* representa la colección de las acciones. Una instancia de esta clase puede definir las propiedades: i) *act:actor* la entidad que ejecuta la acción; ii) *act:recipient*

la entidad que recibe los efectos después de ejecutar la acción; iii) *act:target* el objetivo de la acción (ej. compartir información de localización de una persona); iv) *act:location* el lugar donde se realiza la acción; v) *act.time* la fecha y hora en la que se ejecuta la acción; y, vi) *act.instrument* las cosas que el actor utiliza para ejecutar la acción.

En SOUPA una política de seguridad y privacidad es un conjunto de reglas para permitir o restringir la ejecución de las acciones. La clase *pol:policy* representa el conjunto de todas las políticas. Un individuo de la clase *pol:policy* puede ser asociado con una o más propiedades *pol:permits* o *pol:forbids*. El rango de estas propiedades son las clases *pol:PermittedAction* y *pol:ForbiddenAction* respectivamente.

Adicionalmente, la ontología incluye vocabulario para describir el autor de la política *pol:creator*, la entidad que aplica la política *pol:enforcer*, la fecha y hora de creación de la política *pol.createdOn*, y el modo por defecto de razonamiento de la política *pol.defaultPolicyMode*.

**Agent & BDI.** Incluye un vocabulario para representar a los usuarios humanos y entidades computacionales como agentes. En SOUPA los agentes se caracterizan por un conjunto de nociones mentales como: conocimiento, creencias, deseos e intenciones. La representación de los estados mentales puede ayudar a razonar a las entidades computacionales acerca de las necesidades de los usuarios dentro de un contexto. La clase *Agent* representa a todos los agentes de un dominio y está asociada a tres propiedades, que son usadas para caracterizar el estado mental de un agente: *agt:believes*, *agt:desires* y *agt:intends*. Los rangos de estas propiedades son: *bdi:Fact* que representa los hechos que conoce el agente, *bdi:Desire* que describe los objetivos del agente, y *bdi:Intention* que define los planes del agente para alcanzar sus objetivos.

**Time.** Está desarrollada en base a la ontología DAML Time. Provee un vocabulario para representar tiempo y relaciones temporales de los diferentes eventos que ocurren en un contexto. Las clases principales son: *tme:TemporalThing* que representa todas las cosas que son temporales, *tme:TimeInstant* que representa un instante de tiempo, y *tme:TimeInterval* que representa un periodo de tiempo.

La clase *tme:TemporalThing* está compuesta por la unión de dos clases: *tme:InstantThing* que representa todas las cosas temporales del tipo instante de tiempo y *tme:IntervalThing* que representa todas las cosas temporales del tipo periodo de tiempo.

La propiedad *tme:at* se utiliza para asociar una instancia de *tme:InstantThing* con un tipo de valor XML `xsd:dateTime` y las propiedades *tme:to* y *tme:from* para asociar una instancia de *tme:IntervalThing* con dos instancias individuales de *tme:TimeInstant*.

Para comparar dos diferentes instancias de tiempo existen propiedades como: *tme:before*, *tme:after*, *tme:beforeOrAt*, *tme:afterOrAt*, y *tme:sameTimeAs*. Asimismo para comparar intervalos de tiempo se tienen las propiedades *tme:startsSoonerThan* y *tme:startsLaterThan*.

**Space & Geo Measurement.** Construida en base a las ontologías OpenCyc y OpenGIS está diseñada para representar la relación espacial entre regiones geográficas y el mapeo de coordenadas geoespaciales a representación simbólica del espacio y viceversa.

La clase principal es *spc:SpatialThing* usada para representar todas las cosas que tienen propiedades espaciales. La propiedad *spc:hasCoordinates* permite describir las coordenadas GPS de una clase *spc:SpatialThing*. El rango de esta propiedad es la clase *geo:LocationCoordinates*.

La clase *spec:GeographicalSpace* representa una región geográfica que es controlada por un ente político, el mismo que es identificado por la clase *spc:GeopoliticalEntity*. Las instancias de la clase *spec:GeographicalSpace* pueden contenerse jerárquicamente entre sí a través de las propiedades *spc:spatiallySubsumes* y *spc:spatiallySubsumedBy*.

**Event.** Describe los eventos que tienen propiedades espaciales y temporales. La clase *eve:Event* representa a todos los eventos en el dominio y la clase *eve:SpatialTemporalThing* representa al conjunto de cosas que tienen extensión espacial y temporal, y se define como la intersección entre las clases *tme:TemporalThing* y *spc:SpatialThing*.

Para describir los eventos que tienen extensión espacial y temporal se define la clase *eve:SpatialTemporalEvent* que es la intersección entre las clases *eve:SpatialTemporalThing* y *eve:Event*.

**SOUPA EXTENSION.** Define un vocabulario extendido para dar soporte a dominios particulares de aplicaciones ubicuas, demostrando cómo nuevas ontologías pueden extender las ontologías SOUPA CORE.

Soupa Extension consiste de un grupo de ontologías experimentales para ambientes ubicuos tales como: i) *Meeting & Schedule* para describir información relacionada con reuniones, calendario de eventos y sus participantes; ii) *Document & Digital Document* para describir información de documentos digitalizados (fecha de creación, autor, tamaño, tipo y dirección URL del documento); iii) *Region Connection Calculus* define vocabulario para expresar relaciones espaciales.

### 3.3 Ontology based Context Modeling and Reasoning using OWL (CONON)

CONON [4] está conformada por una ontología de contexto superior que captura los conceptos generales de un contexto básico, a partir del cual en forma jerárquica se pueden extender nuevas ontologías para añadir los conceptos propios de un dominio específico de computación ubicua (oficina, comunidad universitaria, hogar, vehículo, etc.). CONON define un vocabulario basado en cuatro entidades contextuales: personas, localización, entidades computacionales y actividades, como se puede apreciar en la Figura 3.

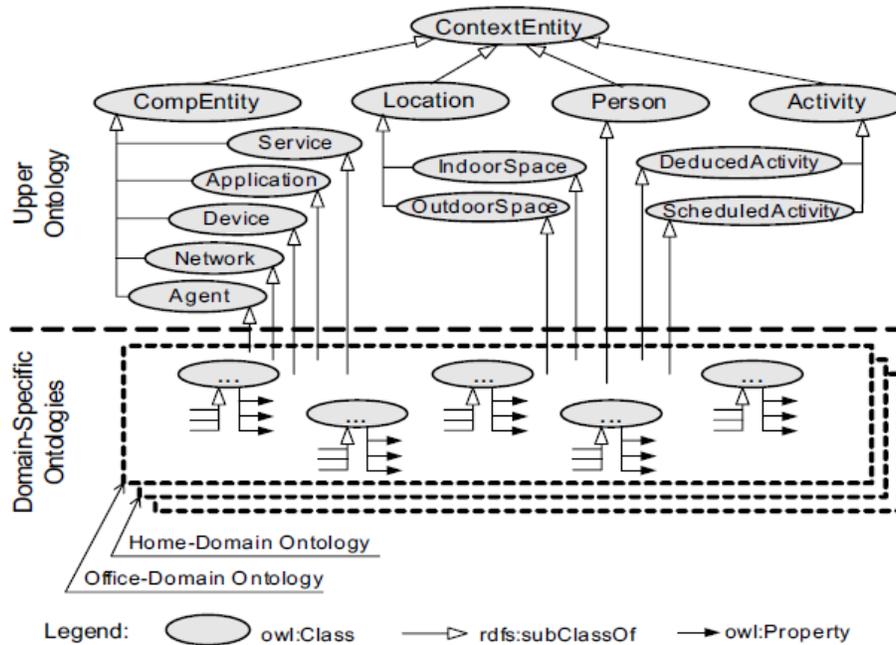


Fig. 3. Definición parcial de la ontología superior de CONON [4]

La clase más general es *ContextEntity* de la cual se derivan las subclases:

- *CompEntity*: representa las entidades computacionales, la cual a su vez introduce las subclases *Service*, *Application*, *Device*, *Network* y *Agent*.
- *Location*: describe los lugares físicos, subdividiéndose en *IndoorSpace* (*building*, *room*, *corridor*, *entry*) y *OutdoorSpace* (*garden*, *dooryard*). Asocia propiedades, tales como: *longitude*, *latitude*, *temperatura*, *altitude*, *noiselevel*, etc.
- *Person*: describe a los usuarios humanos. Asocia propiedades tales como: *name*, *situation*, *homeAddress*, *age*, *nearby*, etc.
- *Activity*: describe las actividades que se ejecutan en un contexto, se subdivide en las clases *ScheduledActivity* y *DeducedActivity* y se asocia con las propiedades *startTime* y *endTime*.

### 3.4 Análisis de las ontologías estudiadas

En las subsecciones previas se han identificado las principales características de tres ontologías propuestas en la literatura para modelar el contexto en diferentes escenarios. Todos estos modelos semánticos hacen uso del lenguaje OWL para representar el conocimiento. La ontología SOUPA ha sido desarrollada utilizando la experiencia de

COBRA-ONT, siendo una ontología más completa y comprensiva, que apoya a la interoperabilidad mediante la reutilización de otras ontologías comunes.

En el caso de la ontología CONON, pese a que la naturaleza del contexto en aplicaciones ubicuas es sumamente amplia y compleja, su contribución radica en la definición de un modelo de contexto común, conformado por las entidades: persona, localización, actividad y entidad computacional. A su vez, estas entidades a más de formar el esqueleto fundamental del contexto actúan como índices para la asociación de nueva información de contexto en dominios particulares.

CONON también demuestra en sus estudios que el razonamiento de contexto es una tarea intensa, por lo que es necesario reducir el número de conceptos o clases que se utilizan o dividir la conceptualización del contexto en varias sub-ontologías.

Finalmente, en base a lo resaltado sobre SOUPA y CONON se puede concluir que la tendencia es crear modelos ontológicos con un alto nivel de abstracción, que se construyan en base a un conjunto de conceptos de contexto comunes a todos los entornos (reutilizando ontologías consensuadas) y que garanticen la extensibilidad mediante la inclusión de conceptos propios de cada dominio.

## **4 Proceso de construcción de la ontología**

En esta sección se describe primero la metodología usada para la construcción de la ontología de contexto. En segundo lugar se aplica la guía propuesta en la metodología para cubrir las actividades de especificación de requisitos de la ontología y la creación del modelo semántico. Finalmente se describe el modelo ontológico resultante de este proceso.

### **4.1 Metodología NeOn**

Al contrario de las metodologías tradicionales [23, 24, 25] que proporcionan básicamente orientación metodológica para la construcción de ontologías, la metodología NeOn [26] identifica un conjunto de escenarios flexibles que soporta diferentes aspectos del proceso de desarrollo de una ontología, así como la reutilización y evolución dinámica de ontologías en red en entornos distribuidos, donde se introduce el conocimiento por diferentes personas y en diferentes etapas del proceso de desarrollo de la ontología.

Los nueve escenarios propuestos en la metodología cubren situaciones que ocurren en el proceso de desarrollo de una ontología, los mismos que se pueden combinar de acuerdo con los requisitos de la ontología y los recursos existentes en el dominio.

A continuación se detalla el proceso utilizado para la construcción de la ontología de contexto, aplicando las guías propuestas en la metodología NeOn.

## 4.2 Especificación de requerimientos de la ontología

El propósito es determinar los requisitos de la ontología que permitan modelar la información de contexto de los elementos involucrados en el ámbito académico de una comunidad universitaria. A continuación se irá describiendo los resultados de cada una de las tareas de la actividad de elicitación de requisitos, usando para ello la plantilla propuesta en la metodología. La Tabla 1 muestra las cinco primeras secciones del documento para el caso de la ontología.

**Tabla 1.** Documento de especificación de requisitos: Secciones 1 a 5.

<b>1</b>	<b>Propósito</b>
El propósito de la construcción de la ontología de contexto es proveer un modelo consensuado que represente los elementos sensibles al contexto dentro del dominio académico universitario de una manera formal y que puedan ser usados en diferentes sistemas de computación.	
<b>2</b>	<b>Alcance</b>
La ontología tiene como alcance determinar las características contextuales de los elementos involucrados en un ambiente académico universitario, tales como: personas, lugares de interés, actividades, dispositivos, aplicaciones informáticas, recursos didácticos, entre otros.	
<b>3</b>	<b>Lenguaje de implementación</b>
La ontología será implementada en OWL [10] y almacenada en un repositorio abierto.	
<b>4</b>	<b>Usuarios finales previstos</b>
Estudiantes y profesores.	
<b>5</b>	<b>Usos previstos</b>
El uso de ontologías para representar información contextual es muy amplio dentro de los sistemas de computación académicos de una universidad, ya que permite dotarles de conciencia del entorno y de esta manera enriquecer la experiencia del usuario, mediante la provisión de información y/o servicios relevantes con su situación actual. A continuación se presenta un listado de aplicaciones que de ninguna manera es exhaustivo, son simplemente ejemplos representativos en los que se puede utilizar la ontología:	
<ul style="list-style-type: none"> <li>• Sistemas colaborativos para apoyar los procesos de enseñanza – aprendizaje dentro y fuera del aula.</li> <li>• Entornos de aprendizaje virtuales.</li> <li>• Agendas electrónicas académicas.</li> <li>• Sistemas de gestión de bibliotecas.</li> <li>• Sistemas de gestión académica (distributivos, horarios, aulas, notas, faltas, etc.).</li> <li>• Sistemas de localización de personas y/o recursos.</li> <li>• Portal web universitario.</li> </ul>	

El propósito, alcance y lenguaje de implementación de la ontología fueron determinados luego de una lluvia de ideas con los miembros del equipo de trabajo. Para determinar los posibles usuarios y usos previstos de la ontología (secciones cuatro y cinco en la Tabla 1),

se tomó como base las posibles aplicaciones que pueden ser usadas en el ámbito académico universitario, y que tienen la necesidad de información de contexto.

Para identificar los requerimientos funcionales se usó la técnica de preguntas de competencia [27] que consiste en escribir preguntas en lenguaje natural que la ontología a desarrollar debe ser capaz de responder. La Tabla 2 muestra una lista ordenada de los requisitos funcionales y no funcionales.

**Tabla 2.** Documento de especificación de requisitos: Sección 6.

<b>6</b>	<b>Requerimientos de la ontología</b>
<b>a)</b>	<b>Requerimientos no funcionales</b>
	La ontología debe estar basada en estándares o modelos reconocidos en existencia o en desarrollo.
<b>b)</b>	<b>Requerimientos funcionales</b>
	PC1. ¿En qué horario el estudiante tiene clases del curso Ingeniería de Software esta semana? Lunes 07h00 – 09h00, Miércoles 07h00 – 09h00.
	PC2. ¿En qué lugar el profesor dictará el curso de Ingeniería de Software el día de hoy? Edificio de la Facultad de Ingeniería, tercer piso, Aula 305.
	PC3. ¿A qué hora el profesor dictará el curso de Ingeniería de Software el día hoy? 07h00.
	PC4. ¿Qué dispositivos, redes o recursos de TI están disponibles para el profesor en este momento?
	<ul style="list-style-type: none"> <li>• Computador Personal del laboratorio 5.</li> <li>• Tablet del profesor.</li> <li>• Smartphone del profesor.</li> <li>• Proyector del laboratorio 5.</li> <li>• Pizarra digital del laboratorio 5.</li> <li>• Aire acondicionado ubicado en el laboratorio 5.</li> <li>• Wifi de Ingeniería.</li> <li>• Bluetooth activado en smartphone, proyector y aire acondicionado.</li> </ul>
	PC5. ¿Qué personas están cerca del profesor en este momento? Estudiantes del curso de Ingeniería de Software.
	PC6. ¿Qué actividades se han planificado para la clase de Ingeniería de Software del día de hoy? Taller sobre estudio de caso de Gestión de Configuración.
	PC7. ¿Qué estado deben presentar los dispositivos cuando el profesor ingresa al laboratorio para empezar la clase de Ingeniería de software? Proyector encendido y mostrando las diapositivas del capítulo de Gestión de Configuración desde la Tablet del profesor. Aire acondicionado encendido.
	PC8. ¿Qué recursos didácticos deben ser enviados al estudiante en este momento? Diapositivas del capítulo de Gestión de Configuración de la asignatura de Ingeniería de Software. Libros y artículos digitales afines a la temática. Documento PDF con la descripción de la tarea de Gestión de Configuración. Sílabo de la asignatura con el detalle de la bibliografía de apoyo, los criterios de evaluación y plazos de entrega de la tarea.
	PC9. ¿Qué aplicaciones o servicios académicos deben ser activados o recordados al profesor cuando finaliza su clase? En los dispositivos móviles del profesor, mostrar accesos directos de las aplicaciones de registro de faltas de estudiantes y de seguimiento académico.
	PC10. ¿Qué eventos afines a la carrera del estudiante se han planificado el día de hoy? y ¿Dónde? Casa abierta de la Facultad de Ingeniería. Parqueadero de la Facultad.

PC11. ¿Qué eventos cercanos a la localización del profesor se están desarrollando? y ¿Dónde? Congreso de Ingeniería de Software. Auditorio de la Facultad de Ingeniería.

PC12. ¿Qué opción debe habilitarse en los dispositivos móviles de los estudiantes cuando se encuentran en la biblioteca? Mostrar una opción con el acceso al sistema de bibliotecas digitales de la Universidad.

PC13. ¿Qué opción debe habilitarse o recordarse en los dispositivos móviles de los estudiantes al finalizar el periodo de exámenes? Mostrar una opción con el acceso al sistema de consulta de calificaciones.

La última sección del documento de especificación de requisitos contiene el pre-glosario de términos dividido en tres partes: términos extraídos de las preguntas de competencia, términos provenientes de las respuestas a las preguntas de competencia, y términos identificados como nombres de entidades (objetos). Para la extracción del glosario de términos se adoptaron sencillas técnicas heurísticas de extracción de terminología. Así por ejemplo, desde los diferentes requisitos se extrajeron nombres, adjetivos y verbos que serán representados en la ontología como conceptos, atributos, relaciones o instancias (en el caso de nombres de entidades). La Tabla 3 muestra las tres partes del glosario de términos. En las secciones *a* y *b* de la tabla, los números representan la frecuencia de aparición de los términos.

**Tabla 3.** Documento de especificación de requisitos: Sección 7.

<b>7</b>	<b>Pre-glosario de términos</b>	
<b>a)</b>	<b>Términos de las preguntas de competencia</b>	
	Estudiante (6)	Lugar (1)
	Horario (1)	Profesor (5)
	Clase, curso (5)	Dispositivo (5)
	Materia (1)	Red (1)
	Semana (2)	Recurso (1)
	Momento (3)	Persona (1)
	Actividad (1)	Día (2)
	Laboratorio (1)	Estado (1)
	Recurso didáctico (2)	Aplicación (2)
	Carrera (1)	Evento (2)
	Biblioteca (2)	Examen (1)
<b>b)</b>	<b>Términos de las respuestas a las preguntas de competencia</b>	
	Edificio (1)	Pizarra digital (1)
	Piso (1)	Aire acondicionado (2)
	Computador (2)	Wifi (1)
	Tablet (2)	Bluetooth (1)
	Smartphone (2)	Estudiante (2)
	Taller (1)	Curso (1)
	Diapositiva (2)	Proyector (1)

Profesor (5)	Capítulo (2)
Documento (1)	Libro (1)
Tarea (2)	Artículos digitales (1)
Sílabo (1)	Asignatura (1)
Plazo (1)	Evaluación (1)
Bibliografía (1)	Congreso (1)
Sistema (3)	Universidad (2)
<b>c) Objetos</b>	
Lunes	Aula 305
Miércoles	Laboratorio 5
07h00 – 09h00	Gestión de Configuración
Ingeniería de Software	Sistema de bibliotecas digitales
Facultad de Ingeniería	Sistema de consulta de calificaciones
Auditorio de la Facultad de Ingeniería	...

Aunque la metodología recomienda usar los términos con la frecuencia más alta para dirigir la búsqueda de recursos existentes que ayuden en el desarrollo de la ontología, en este caso se procedió a categorizar la terminología con el fin de focalizar la búsqueda en dominios más específicos. Las categorías identificadas son: *persona*, *organización*, *localización*, *recurso TI* y *actividad*. A continuación se describen brevemente estas categorías:

- *Persona*: representa a las personas que realizan actividades académicas dentro de la universidad. Incluye las subclases: *Profesor* y *Estudiante*.
- *Organización*: representa la estructura organizacional y curricular de una universidad. Se subdivide en: *Facultad*, *Carrera*, *Asignatura* y *Curso*.
- *Localización*: describe los lugares físicos y su representación simbólica. Permite especificar las coordenadas GPS de una entidad con propiedades espaciales.
- *Recurso TI*: incluye un vocabulario para representar todos los recursos de tecnologías de información que se pueden utilizar dentro de la actividad académica universitaria. Algunos ejemplos de estas clases son: *Dispositivo*, *AplicacionServicio*, *RecursoDidacticoDigital* y *Red*.
- *Actividad*: define un vocabulario para gestionar las actividades de índole académica de una universidad. Involucra las subclases: *CalendarioAcademico*, *HorarioClase* y *ActividadCurso*. Además provee un vocabulario para representar el tiempo y las relaciones de temporalidad de las diferentes actividades que ocurren en un contexto.

Estas categorías de datos proveen la información necesaria para concentrar la búsqueda de modelos ontológicos que cubran estos dominios de interés.

### 4.3 Modelamiento de la ontología de contexto

En esta subsección se explica el proceso seguido para crear una ontología de contexto para un ambiente académico universitario. Como se mencionó en la sección 3, el uso de

ontologías para modelar el contexto ha sido ampliamente estudiado (ver por ejemplo [2, 3, 4, 5, 6, 7, 8]). A pesar de estos esfuerzos no se puede hablar todavía de una ontología estándar para este propósito; la mayoría de las ontologías creadas son específicas a la aplicación, cada una de ellas elaborada para un dominio particular.

El objetivo es reutilizar el trabajo existente en la literatura para crear una red de ontologías que permita representar de manera adecuada el contexto de los elementos que conforman el ambiente académico universitario. Se requiere que el modelo abarque tanto conceptualizaciones de las características de los usuarios (profesores y estudiantes), como de sus actividades académicas, los mismos que pueden cubrir entidades de diferentes dominios.

Para la creación de la ontología se ha adoptado una propuesta top-down, basada en uno de los escenarios para la creación de ontologías propuesto en la Metodología NeOn. Para este caso particular se ha adoptado el escenario seis: *reusando, fusionando y ejecutando reingeniería de recursos ontológicos*. Este escenario parte de la premisa de que los desarrolladores tienen a su disposición recursos ontológicos útiles para modelar el dominio de su problema. Los recursos son fusionados para crear un nuevo modelo ontológico, sin embargo la fusión no es suficiente para cubrir las necesidades de la nueva ontología, por tanto un proceso de reingeniería es requerido. A continuación se describen las actividades que se han llevado a cabo:

**i) Búsqueda de ontologías.** Durante esta etapa se buscaron ontologías que reúnan los requisitos identificados en el *Documento de especificación de requerimientos de la ontología*, introducido en la subsección 4.2. Para la búsqueda de ontologías se utilizó Watson<sup>2</sup> y otros motores de búsqueda semántica como Swoogle<sup>3</sup> y Síndice<sup>4</sup>. Se consultó en forma manual algunas librerías de ontologías existentes como Protége Ontology Library<sup>5</sup>. La tabla 4 muestra las ontologías encontradas por cada categoría, incluyendo el proyecto o institución creador(a) del modelo.

**Tabla 4.** Listado de ontologías encontradas por cada categoría.

Ontología	Proyecto o institución
Categoría 1: Persona	
FOAF	FOAF-Project
GUMO	DFKI GmbH
eBiquity Person	UMBC ebiquity
Person	W3C Schema.org
...	...

<sup>2</sup> <http://watson.kmi.open.ac.uk/>

<sup>3</sup> <http://swoogle.umbc.edu/>

<sup>4</sup> <http://sindice.com/>

<sup>5</sup> [http://protegewiki.stanford.edu/wiki/Protege\\\_\\\_Ontology\\\_\\\_Library](http://protegewiki.stanford.edu/wiki/Protege\_\_Ontology\_\_Library)

Categoría 2: Organización	
SOUPA Organization Ontology LUBM University Ontology AIISO ...	H. Chen et at. W3C Lehigh University J. Heflin Open University ...
Categoría 3: Localización	
CONON SOUPA COBRA-ONT Geo OWL ...	X.H. Wang et at. H. Chen et at. H. Chen et at. W3C ...
Categoría 4: Recurso TI	
CONON SOUPA ...	X.H. Wang et at. H. Chen et at. ...
Categoría 5: Actividad	
AcademIS CONON SOUPA COBRA-ONT DAML Time ...	No especificada X.H. Wang et at. H. Chen et at. H. Chen et at. DALM-Project ...

**ii) Comparación de las ontologías.** Durante esta tarea se compararon las ontologías identificadas en el paso anterior. Para determinar el conjunto de ontologías que no serán reutilizadas, las siguientes acciones fueron ejecutadas:

- Comprobar si el alcance y propósito establecido en el *Documento de Especificación de Requisitos de la Ontología* (DERO) es similar al de la ontología candidata.
- Comprobar si los requisitos no funcionales establecidos en el DERO están cubiertos por el dominio de la ontología candidata.
- Comprobar si los requisitos funcionales en forma de preguntas de competencia incluidos en el DERO son cubiertos (total o parcialmente) por el dominio de la ontología candidata. Esta comprobación fue ejecutada calculando la *precisión* y *cobertura* de la terminología de las ontologías candidatas con respecto a la terminología incluida en el DERO.

Se ha definido la *precisión* como la proporción de los términos en la ontología candidata incluidos en los términos identificados en el DERO sobre los términos en la ontología candidata. Esto se expresa de la siguiente manera:

$$\text{Precisión} = \frac{(\text{TermOntCandidata} \cap \text{TermDERO})}{\text{TermOntCandidata}}. \quad (1)$$

La medida de *cobertura* está basada en la medida *recall* usada en la recuperación de información. En este trabajo, la *cobertura* es la proporción de los términos identificados en el DERO que están incluidos en los términos recogidos en la ontología candidata sobre los términos identificados en el DERO. Esto se expresa de la siguiente manera:

$$\text{Cobertura} = \frac{(\text{TermOntCandidata} \cap \text{TermDERO})}{\text{TermDERO}}. \quad (2)$$

Con el fin de ilustrar el proceso ejecutado para analizar las ontologías candidatas bajo los criterios antes mencionados, se ha seleccionado como ejemplo la categoría de ontologías *Persona* (ver categoría 1 en la tabla 4). En la tabla 5 se muestra el análisis efectuado por el equipo de desarrollo de ontologías al grupo de recursos ontológicos seleccionado.

**Tabla 5.** Ejemplo de comparación de ontologías de la categoría *Persona*.

criterio	Rango de valores	FOAF	eBiquity Person	GUMO	Person
Alcance similar	[Sí, No, Desconocido]	Sí	Sí	Sí	Sí
Propósito similar	[Sí, No, Desconocido]	Sí	No	No	Sí
Requerimientos no funcionales cubiertos	[Sí-Totalmente, Sí-Parcialmente, No, Desconocido]	No	No	No	Sí-Parcialmente
Requerimientos funcionales cubiertos	[Sí-Totalmente, Sí-Parcialmente, No, Desconocido]	Sí-Parcialmente	No	No	Sí-Parcialmente

Para decidir si una ontología candidata no es considerada útil en el proceso de reutilización, la siguiente heurística fue aplicada: Si el desarrollador contestó *No* a los criterios “*Alcance similar*” y/o “*Propósito similar*” y/o “*Requerimientos funcionales cubiertos*”, entonces la ontología fue considerada no útil, y por tanto fue eliminada del conjunto de ontologías candidatas.

Luego de la aplicación de la heurística, las ontologías *eBiquity Person* y *GUMO* fueron eliminadas del conjunto de ontologías candidatas. El mismo proceso fue aplicado para el resto de ontologías identificadas en las otras categorías.

iii) **Selección de las ontologías.** En esta tarea se determinó cuál de las ontologías identificadas en la tarea previa es la más apropiada para ser reutilizada en la ontología a desarrollar. Para determinar tal ontología se analizaron cuatro dimensiones de las características no-funcionales de las ontologías. Las dimensiones consideradas se describen brevemente a continuación:

- *Costo de reutilización:* se refiere a la estimación del costo (económico y temporal) que se necesita para la reutilización de la ontología candidata.
- *Esfuerzo de comprensión:* se refiere a la estimación del esfuerzo necesario para comprender la ontología candidata.
- *Esfuerzo de integración:* se refiere a la estimación del esfuerzo necesario para la integración de la ontología candidata con la ontología que está siendo desarrollada.
- *Confiabilidad:* se refiere al análisis de si se puede confiar en la ontología candidata a reutilizar.

Es necesario aclarar que cada una de las dimensiones arriba descritas posee un conjunto de criterios que permiten obtener un puntaje para cada ontología candidata. Por cada criterio se tiene un rango de valores lingüísticos<sup>6</sup> y un peso numérico.

La tabla 6 muestra los valores obtenidos para cada criterio y por cada ontología candidata de la categoría *Persona*. De las cuatro ontologías sólo dos fueron catalogadas como aceptables para ser reusadas. En la tabla los pesos numéricos fueron definidos por el equipo de trabajo, considerando la importancia de cada uno de los criterios. Los símbolos (+) y (-) en los pesos se establecieron para observar si el criterio se cuenta de manera positiva o negativa, respectivamente.

**Tabla 6.** Ejemplo de selección de ontologías de la categoría *Persona*.

Criterio	Peso	Valores*	
		FOAF	Person
Costo de reutilización			
Costo económico del reúso	(-) 9	1	1
Tiempo requerido para reúso	(-) 7	1	1
Esfuerzo de comprensión			
Calidad documentación	(+) 8	3	2
Disponibilidad de conocimiento externo	(+) 7	3	2
Claridad del código	(+) 8	3	3
Esfuerzo de integración			
Adecuación de extracción de conocimiento	(+) 9	2	2

<sup>6</sup> Estos valores lingüísticos toman valor en una escala de [0 a 3] de la siguiente manera: desconocido=0, bajo=1, medio=2, y alto=3

Adecuación de convenciones de nombrado	(+) 5	2	3
Adecuación del lenguaje de implementación	(+) 8	3	3
Conflictos de conocimiento	(-) 8	2	2
Adaptación al razonador	(+) 5	3	1
Necesidad de términos puente	(-) 7	1	1
<b>Confiabilidad</b>			
Disponibilidad de pruebas	(+) 8	3	1
Soporte teórico	(+) 8	3	3
Reputación del equipo de desarrollo	(+) 8	3	3
Fiabilidad documentación	(+) 3	3	3
Apoyo práctico	(+) 6	3	3
<b>Total:</b>		<b>1.48</b>	<b>1.26</b>

\*desconocido=0, bajo=1, medio=2, y alto=3

Para calcular la puntuación de las diferentes ontologías candidatas se usaron las siguientes fórmulas:

$$\text{Puntaje}_{i(+)} = \sum_{j(+)} \text{Valor}_{T_{i,j}} \times \frac{\text{Peso}_j}{\sum_j \text{Peso}_j} \quad (3)$$

$$\text{Puntaje}_{i(-)} = \sum_{j(-)} \text{Valor}_{T_{i,j}} \times \frac{\text{Peso}_j}{\sum_j \text{Peso}_j} \quad (4)$$

donde:

- $i$  es una ontología candidata particular.
- $\text{Puntaje}_{i(+)}$  es el puntaje para la ontología candidata  $i$  para el conjunto de criterios ponderados con (+).
- $\text{Puntaje}_{i(-)}$  es el puntaje para la ontología candidata  $i$  para el conjunto de criterios ponderados con (-).
- $j$  es un criterio particular de las incluidas en la Tabla 6,  $j(+)$  representa los criterios con peso positivo y  $j(-)$  los criterios con peso negativo.
- $\text{Valor}_{T_{i,j}}$  es el valor para el criterio de  $j$  en la ontología  $i$ .
- $\text{Peso}_j$  es el peso numérico asociado al criterio  $j$ .

La puntuación final para cada ontología candidata se obtuvo de la siguiente manera:

$$\text{Puntaje}_i = \text{Puntaje}_{i(+)} - \text{Puntaje}_{i(-)} \quad (5)$$

Las ontologías con el puntaje más alto en cada una de las categorías fueron seleccionadas para la construcción de la ontología de contexto. En este sentido se ha seleccionado para la categoría *Persona* la ontología FOAF, para las categorías *Localización* y *RecursoTI* la ontología CONON y para la categoría *Actividad* las ontologías CONON y DAML Time. Para la categoría *Organización* ninguna de las ontologías encontradas fue seleccionada como útil para el proceso de reutilización.

**iv) Personalización e integración de las ontologías seleccionadas.** Para personalizar e integrar las ontologías seleccionadas se ejecutó un proceso de alineamiento manual. El objetivo fue crear un modelo ligero y manejable que pueda representar el contexto en un dominio académico universitario bajo un único vocabulario.

El proceso de alineamiento no tiene sentido sin un proceso de re-estructuración de los recursos ontológicos a ser alineados. Algunas situaciones pueden darse:

- Los recursos importados contienen información irrelevante para modelar el perfil.
- Algunos conceptos relevantes para modelar el perfil no están contemplados en los recursos importados.

Con el fin de cumplir los principios antes descritos, algunas de estas sub-actividades fueron ejecutadas en los recursos ontológicos seleccionados para reutilización:

- *Podar las ontologías de acuerdo a las necesidades.* En algunos casos fue necesario eliminar ciertos conceptos que no aportaban nada a la ontología construida.
- *Enriquecer la ontología a reusar.* En ciertas ontologías seleccionadas fue necesario ejecutar un proceso de enriquecimiento con nuevas estructuras conceptuales que permitan cubrir las necesidades puntuales de la ontología a desarrollar.
- *Adaptar las ontologías seleccionadas a los criterios de diseño de la ontología destino.* Algunas convenciones de nombrado de términos fueron aplicadas para homogenizar la ontología final.
- *Incluir conexiones a través de relaciones entre clases que no fueron inicialmente conectadas.* Por ejemplo, para conectar la ontología sobre personas con la ontología que describe las actividades que realiza esa persona se usó la propiedad *cumple*.

Las ontologías FOAF y DAML Time estaban disponibles en el lenguaje OWL por lo que no fue necesario realizar una traducción, mientras que la ontología CONON estaba definida únicamente de manera conceptual, sin embargo se reutilizó sus conceptos y estructura, obteniendo como resultado una ontología con dos capas de abstracción: la capa superior que incluye los conceptos comunes a todos los entornos, y la capa inferior que contiene los conceptos propios del dominio académico universitario.

La figura 4 muestra el metamodelo de la ontología que representa el contexto en un dominio académico universitario. En la ontología las clases se relacionan entre sí y con sus atributos mediante las sentencias OWL ObjectProperty y OWL DatatypeProperty respectivamente. Las tablas 7 y 8 muestran las propiedades tipo objeto y datos, respectivamente.

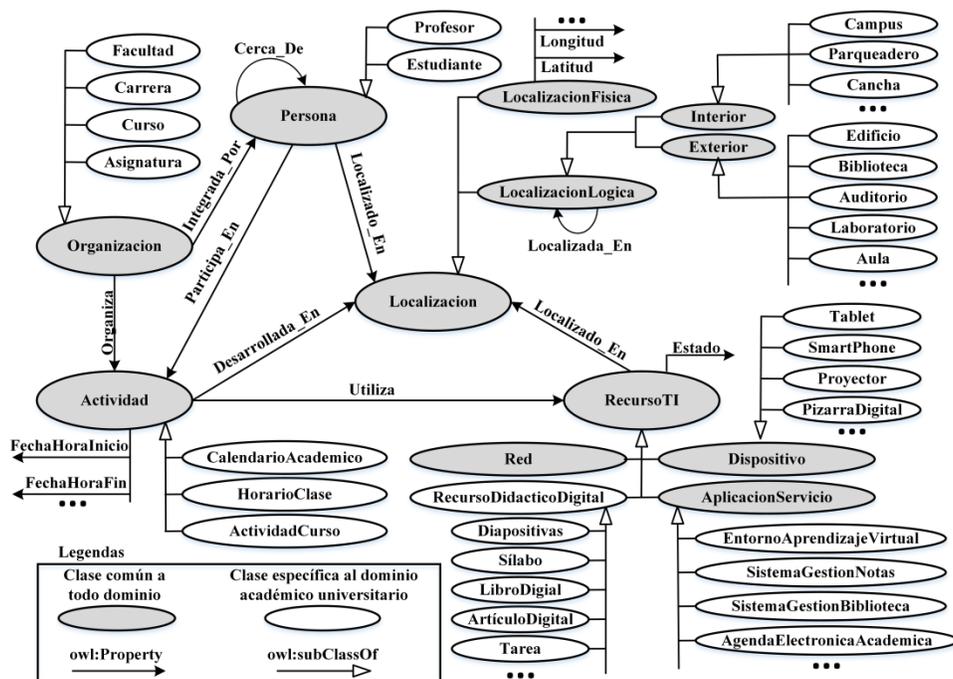


Fig. 4. Metamodelo de la ontología de contexto para un dominio académico universitario

Tabla 7. Listado de propiedades objeto (ObjectProperty)

Domain	ObjectProperty	Range
Persona	<i>Participa_En</i> : Asocia una instancia de la clase <i>Persona</i> con una actividad académica. Propiedad inversa: <i>Es_Realizada_Por</i> .	<i>Actividad</i>
Persona	<i>Cerca_De</i> : Describe las personas o recursos que están cerca de un individuo. Propiedad simétrica.	<i>Persona</i> <i>RecursoTI</i>
Persona RecursoTI	<i>Localizado_En</i> : Describe la ubicación física (coordenadas GPS) o lógica (aula, laboratorio, biblioteca, etc.) de un individuo de la clase <i>Persona</i> o <i>RecursoTI</i> . Propiedad inversa: <i>Ubica</i> .	<i>Localización</i>
Organización	<i>Organiza</i> : Describe la entidad que organiza una actividad. Propiedad inversa: <i>Es_Organizada_Por</i> .	<i>Actividad</i>

<i>Organización</i>	<i>Integrada_Por</i> : Describe la organización a la que pertenece una persona. Propiedad inversa: <i>Es_Parte_De</i> .	<i>Persona</i>
<i>Profesor</i> <i>Estudiante</i>	<i>Cumple</i> : Asocia una instancia de la clase <i>Profesor</i> o <i>Estudiante</i> con las actividades del calendario académico y del horario de clases que debe cumplir. Propiedad inversa: <i>Es_Cumplido_Por</i> .	<i>CalendarioAcademico</i> <i>HorarioClase</i>
<i>Profesor</i>	<i>Gestiona</i> : Asocia una instancia de la clase <i>Profesor</i> con las actividades que gestiona en un curso. Propiedad inversa: <i>Es_Gestionada_Por</i> .	<i>ActividadCurso</i>
<i>Profesor</i>	<i>Dicta</i> : Asocia una instancia de la clase <i>Profesor</i> con los cursos que dicta. Propiedad inversa: <i>Es_Dictado_Por</i> .	<i>Curso</i>
<i>Estudiante</i>	<i>Desarrolla</i> : Asocia una instancia de la clase <i>Estudiante</i> con las actividades que desarrolla. Propiedad inversa: <i>Es_Desarrollada_Por</i> .	<i>ActividadCurso</i>
<i>Estudiante</i>	<i>Se_Matricula</i> : Describe las carreras y cursos en los que el estudiante se encuentra matriculado. Propiedad inversa: <i>Tiene_Matriculado</i> .	<i>Carrera</i> <i>Curso</i>
<i>Facultad</i>	<i>Oferta</i> : Describe las carreras que se ofertan en una facultad. Propiedad inversa: <i>Es_Ofertada_Por</i> .	<i>Carrera</i>
<i>Carrera</i>	<i>Consiste_De</i> : Describe las asignaturas que conforman una carrera. Propiedad inversa: <i>Pertenece_A</i> .	<i>Asignatura</i>
<i>Curso</i>	<i>Estudia</i> : Asocia una instancia de la clase <i>Curso</i> con la asignatura que será objeto de estudio. Propiedad inversa: <i>Es_Estudiada_Por</i> .	<i>Asignatura</i>
<i>Curso</i>	<i>Ejecuta</i> : Asocia una instancia de la clase <i>Curso</i> con la planificación del sílabo. Propiedad inversa: <i>Es_Ejecutado_Por</i> .	<i>Silabo</i>
<i>Curso</i>	<i>Se_Dicta_En</i> : Asocia una instancia de la clase <i>Curso</i> con su horario de clases.	<i>HorarioClase</i>
<i>Sílabo</i>	<i>Contiene</i> : Describe las referencias bibliográficas y actividades del curso que están contenidas en el sílabo. Propiedad inversa: <i>Es_Contenido_En</i> .	<i>Bibliografía</i> <i>ActividadCurso</i>
<i>Actividad</i>	<i>Utiliza</i> : Describe los recursos de tecnologías de información que se utilizan en una actividad. Propiedad inversa: <i>Es_Utilizado_En</i> .	<i>RecursoTI</i>
<i>Actividad</i>	<i>Desarrollada_En</i> : Asocia una instancia de la clase <i>Actividad</i> con el lugar en el que se desarrolla. Propiedad inversa: <i>Se_Desarrolla</i> .	<i>Localizacion</i>

<i>Bibliografia</i>	<i>Referencia_A</i> : Asocia la descripción bibliográfica con los artículos y libros digitales. <i>Propiedad inversa</i> : <i>Es_Referenciado_Por</i> .	<i>LibroDigital</i> <i>ArticuloDigital</i>
<i>Sensor</i>	<i>Obtiene</i> : Describe el dispositivo que obtiene las coordenadas GPS de una persona. <i>Propiedad inversa</i> : <i>Es_Obtendida_Por</i> .	<i>LocalizacionFisica</i>
<i>Actividad</i>	<i>FechaHoraInicio</i> : Asocia una instancia de la clase <i>Actividad</i> con la fecha y hora de inicio de la actividad.	<i>DAMLTime:Instant</i>
<i>Actividad</i>	<i>FechaHoraFin</i> : Asocia una instancia de la clase <i>Actividad</i> con la fecha y hora de fin de la actividad.	<i>DAMLTime:Instant</i>
<i>LocalizacionFisica</i>	<i>FechaHoraLocalizacionFisica</i> : Asocia una instancia de la clase <i>LocalizacionFisica</i> con la fecha y hora de censado de la localización.	<i>DAMLTime:Instant</i>

**Tabla 8:** Listado de propiedades tipo atributo (DatatypeProperty)

<b>Clase</b>	<b>DatatypeProperty</b>
<i>EntidadContexto</i>	<i>ID</i> : Identificador único de una entidad de contexto. <i>Nombre</i> : Nombre de una entidad de contexto. <i>Descripción</i> : Breve descripción de una entidad de contexto.
<i>Persona</i>	Es equivalente con la clase <i>foaf:Person</i> , por lo que utiliza sus propiedades.
<i>Carrera</i>	<i>AñosEstudioCarrera</i> : Número de años de estudio de una carrera.
<i>Asignatura</i>	<i>CreditosAsignatura</i> : Número de créditos asignados a una asignatura. <i>NivelAsignatura</i> : Ciclo o nivel en el que se estudia la asignatura. <i>ContenidoAsignatura</i> : Listado de temas que se tratarán en la asignatura. <i>MetodologiaAsignatura</i> : Descripción de la metodología que se utilizará en la asignatura. <i>ResultadoAprendizajeAsignatura</i> : Descripción de los resultados de aprendizaje de la asignatura.
<i>Curso</i>	<i>ParaleloCurso</i> : Identificación del curso. <i>PeriodoCurso</i> : Periodo lectivo en el que se dicta el curso.
<i>RecursoTI</i>	<i>EstadoRecursoTI</i> : Disponibilidad del recurso de TI.
<i>Actividad</i>	<i>TipoActividad</i> : Clasificador de actividades.
<i>LocalizaciónFisica</i>	<i>Latitud y Longitud</i> : Coordenadas GPS. <i>Precisión</i> : Nivel de precisión de la ubicación física proporcionada.

Para mayor detalle de la estructura de la ontología, en cuanto a sus clases, propiedades y restricciones se puede consultar el archivo “*ContextAware.owl*” disponible en la url: “<https://www.dropbox.com/s/c0mtv5ksvw8e2eq/ContextAware.owl?dl=0>”.

**v) Evaluación de la ontología resultante.** Esta actividad tiene como meta comprobar la calidad técnica de la ontología construida. Hay que indicar que al momento de escribir este artículo, únicamente se ha comprobado que la ontología no contenga errores de sintaxis. Es trabajo en ejecución la evaluación de posibles errores de diseño, estructura y calidad semántica de la ontología.

## 5 Conclusiones y trabajo futuro

En este trabajo se analizaron las diferentes formas de clasificar el contexto y el alcance de las entidades que pueden ser usadas para representarlo. Como una alternativa para representar formalmente los términos que definen el contexto se describió el concepto de ontología y la terminología usada para modelarlos.

Previo al proceso de construcción de una ontología que represente el contexto en un ambiente académico universitario, se analizaron diferentes modelos de contexto ontológicos basados en OWL, lo cual permitió identificar que aunque existen varios esfuerzos en la literatura, no se puede hablar aún de un modelo semántico estándar reconocido por la comunidad. De todas las propuestas analizadas se debe destacar la ontología CONON. La ventaja de esta ontología frente a otras propuestas es que fue diseñada con el objetivo de ofrecer un modelo semántico del más alto nivel de abstracción, de forma que puede ser utilizado en múltiples dominios de aplicación.

Para el proceso de construcción de la ontología se usó la metodología NeOn, aplicando uno de sus escenarios de uso. En la fase de elicitación de requisitos se determinó que las entidades más comunes para representar el contexto en un ambiente académico universitario pueden ser clasificadas en: persona, organización, localización, recursos TI y actividad. Estas categorías fueron utilizadas para localizar recursos ontológicos que permitan representar esta información. Para la actividad de construcción de la ontología se ejecutaron las cinco fases que propone la metodología. Se reutilizaron diferentes modelos semánticos y el resultado fue estructurado en dos capas: la capa superior, que incluye las clases y propiedades que son genéricas para cualquier dominio (aquí el aporte fundamental fue de la ontología CONON), y la capa inferior, que incluye las clases y propiedades específicas del dominio académico universitario.

Como trabajo futuro, previo a la publicación y uso de la ontología, será necesario ejecutar diferentes pruebas al modelo propuesto, tales como medir la usabilidad y la calidad de la modelación. Para medir el primer factor se pretende usar la ontología en diferentes escenarios reales de modelación de contexto en el ambiente académico universitario, validando que los términos de la ontología cubran los conceptos que se quieren representar. Mientras que para medir la calidad se pretende ejecutar diferentes preguntas relacionadas

con el contexto del dominio académico universitario (usando SPARQL), las cuales la ontología debe ser capaz de contestar e inferir.

## Referencias

1. Weiser, M., Brown, J. S.: The coming age of calm technology. In *Beyond Calculation*, pp. 75 a 85. Springer, New York (1997).
2. Chen, H., Finin, T., Joshi, A.: An Ontology for Context Aware Pervasive Computing Environments. *The Knowledge Engineering Review*, vol. 18, no. 03, pp. 197 a 207 (2003).
3. Chen, H., Perich, F., Finin, T., Joshi, A.: SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In *Mobile and Ubiquitous Systems: Networking and Services, MOBIQUITOUS 2004*, pp. 258 a 276. IEEE (2004).
4. Wang, X.H., Zhang, D.Q., Gu, T., Pung, H.K.: Ontology Based Context Modeling and Reasoning using OWL. In *Pervasive Computing and Communications Workshops, 2004*, pp. 18 a 22. IEEE (2004).
5. Strang, T., Linnhoff-Popien, C., Frank, K.: CoOL: A Context Ontology Language to enable Contextual Interoperability. In *4th Int. Conf. on Distributed Applications and Interoperable Systems*, pp. 236 a 247. Springer, Berlin Heidelberg (Noviembre 2003).
6. Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M.H., Tran, L.: Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation. <http://www.w3.org/TR/CCPP-struct-vocab/> (Enero 2004).
7. Gu, T., Wang, X.H., Pung, H.K., Zhang, D.Q.: (2004, January). An ontology-based context model in intelligent environments. In *Proceedings of communication networks and distributed systems modeling and simulation conference*, pp. 270 a 275 (2004).
8. Xu, N., Zhang, W.S., Yang, H.D., Zhang, X.G., Xing, X.: CACOnt: a ontology-based model for context modeling and reasoning. In *Applied Mechanics and Materials*, Vol. 347, pp. 2304 a 2310 (2013).
9. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: principles and methods. *Data and Knowledge Engineering*, vol. 25, no. 1-2, pp. 161 a 197 (1998).
10. Bechhofer, S., Van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL Web Ontology Language Reference. W3C, Tech. Rep. (Febrero 2004).
11. W3C: OWL Web Ontology Language, <https://www.w3.org/2001/sw/wiki/OWL>.
12. Schilit, B., Theimer, M.: Disseminating Active Map Information to Mobile Hosts. *Network*, vol. 8, no. 5, pp. 22 a 32. IEEE (1994).
13. Krumm, J.: *Ubiquitous Computing Fundamentals*. CRC Press. Florida (2010).
14. Dey, A.K.: Providing architectural support for building context-aware applications. Tesis Doctoral. Georgia Institute of Technology (2000).
15. Poslad S., *Ubiquitous Computing: Smart Devices, Environments and Interactions*. Jhon Wiley & Sons, Londres (2009).
16. Dey, A.K., Abowd, G.: Towards a better understanding of context and context-awareness. Workshop on the What, Who, Where, When and How of Context-Awareness, affiliated with the 2000 ACM Conference on Human Factors in Computer Systems (CHI 2000).
17. FOAF Vocabulary Specification 0.99, <http://xmlns.com/foaf/spec/>.
18. DALM Time, <http://www.isi.edu/~hobbs/damlttime/time-entry.owl>.
19. OPENCYC, <http://www.opencyc.org/>.

20. COBRA-ONT v0.4, <http://daml.umbc.edu/ontologies/cobra/0.4/>.
21. MoGATU BDI Ontology , <http://ebiquity.umbc.edu/resource/html/id/2/MoGATU>.
22. Rei Policy Ontology, <http://ebiquity.umbc.edu/resource/html/id/33/Rei-Ontologies>.
23. Fernández-López, M., Gómez-Pérez, A., Pazos-Sierra, J., and Pazos-Sierra, A.: Building a chemical ontology using methontology and the ontology design environment. In *IEEE Intelligent Systems*, vol. 8, no. 1, pp. 37 a 46 (1999).
24. Staab, S., Studer, R., Schnurr, H.P., Sure., Y.: Knowledge processes and ontologies. In *IEEE Intelligent Systems*, vol. 16, no. 1, pp. 26 a 34 (2001).
25. Pinto, S., Staab, S., Sure, Y., Tempich, C.: Ontoedit empowering swap: a case study in supporting distributed, loosely-controlled and evolving engineering of ontologies (diligent) . In *Proceedings of the First European Semantic Web Symposium, ESWS 2004*, pp. 16 a 30. Heraklion, Crete, Greece (2004).
26. Suarez-Figueroa, M.: *NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse*. *Dissertations in Artificial Intelligence / Dissertationen Zur Kunstlichen Intelligenz*. IOS Press (2013).
27. Grüninger, M., Fox, M.S.: Methodology for the design and evaluation of ontologies. In *International Joint Conference on Artificial Inteligence (IJCAI95), Workshop on Basic Ontological Issues in Knowledge Sharing* (1995).