
Web Semántica

Asignatura Doctorado

Marzo-Abril 2005

Índice

1. Introducción
2. Ontologías
3. Lenguajes para formalizar Ontologías
4. Problemas abiertos a la investigación

Introducción

¿A qué nos referimos con el término Web Semántica?

1. ¿Qué interpretaremos por Web?.

>> ... *Web is the largest available environment for the deployment of agents, and much work in agent research is driven by Web-based applications.*

2. ¿Qué interpretaremos por “Semántica” ?

>>> ... *the Web is not geared towards agent use, but is rather designed for human use. The advent of the Semantic Web is rapidly making available a semantically much richer layer of machine accessible contents on top of the existing infrastructure.*

Introducción

¿Los estudios previos en Representación del Conocimiento (KR) pueden ser aplicados a este nuevo contexto?

Sí, pero ...

... many of the assumptions that underly current Knowledge Representation (KR) techniques as used in agent technology are no longer valid when KR techniques are deployed in a large scale and open environment.

How the Semantic Web will change KR: challenges and opportunities for a new research agenda, Frank van Harmelen. *The Knowledge Engineering Review*.

¿Por qué?

1. Tamaño.

Turner, J. and Jennings, N. (2000). Improving the scalability of multi-agent systems. *Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems*.

Introducción

2. Granularidad y frecuencia de cambios alta.

Schut, M. And Wooldridge, M. (2000). Intention reconsideration in complex environments. In *Proceedings of the 4th International Conference on Autonomous Agents*.

3. Falta de integridad referencial.

Sichman, J. S. and Demazeau, Y. (1995). Exploiting social reasoning to deal with agency level inconsistency. In *Proceedings of the First International Conference on MultiAgent Systems (ICMAS-95)*.

4. Desarrollo distribuido \Leftrightarrow Confianza distribuida en el conocimiento representado.

Jonker, C. M. and Treur, J. (1999). Formal analysis of models for the dynamics of trust based on experiences. *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99)*,

Introducción

5. Calidad variable del conocimiento representado. Posibilidad de inconsistencias locales

6. Uso impredecible del conocimiento \Leftrightarrow ¿Cuál es el propósito?.

Musen, M. (1992). Dimensions of knowledge sharing and reuse. *Computers and Biomedical Research*

7. Múltiples fuentes de conocimiento \Leftrightarrow Conceptualizaciones distintas para un mismo dominio/problema.

Jonker, C. M. and Treur, J. (1999). Formal analysis of models for the dynamics of trust based on experiences. *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99)*.

8. Uso de diferentes Vocabularios.

Introducción

9. Imposibilidad de copia. No efectivo el acceso a la información.

10. Importancia de las justificaciones ¿Cuál es la calidad de la inferencia realizada?

Kuhnel, R. (1999). Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1):1–69.

11. Inferencia robusta. Razonadores incompletos e incluso no válidos en determinados grados.

Lesser, V., Hornling, B., Klassner, F., Raja, A., Wagner, T., and Zhang, S. (2000). Big: An agent for resource-bounded information gathering and decision making. *Artificial Intelligence*, 118(1-2):197–244.

Introducción

Investigaciones futuras a considerar:

1. Razonamiento aproximado.
2. Formulación neutra del conocimiento (independiente del propósito).
3. Reconciliar múltiples vocabularios.

Ontologías

¿Qué se entiende por Ontología en la Web Semántica?

Una especificación de una conceptualización con el propósito de ser compartida y reutilizada.

Tom Gruber. *What is an Ontology? Toward principles for the design of ontologies used for knowledge sharing. Workshop on Formal Ontology, Padua, March 1993.*

¿Qué se entiende por Conceptualización?

Modelo simplificado del mundo que nos interesa representar con algún propósito.

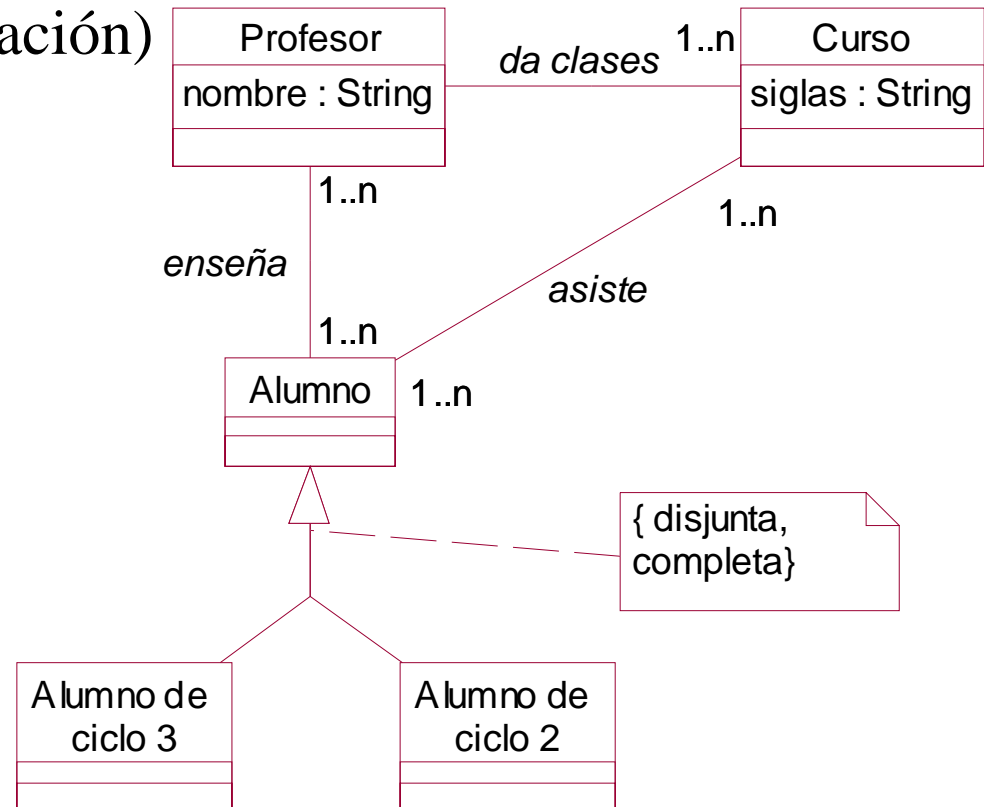
Modelo \Leftrightarrow mundos válidos (Restricción)

Ej. Diagramas E/R yUML .

Ontologías

Elementos lenguaje de modelado

1. Concepto (Clase o Entidad)
2. Relación (Rol o Asociación)
3. Propiedad (Atributo)



Ontologías

Interpretación mediante conjuntos de instancias

Profesor = {P1, P2}

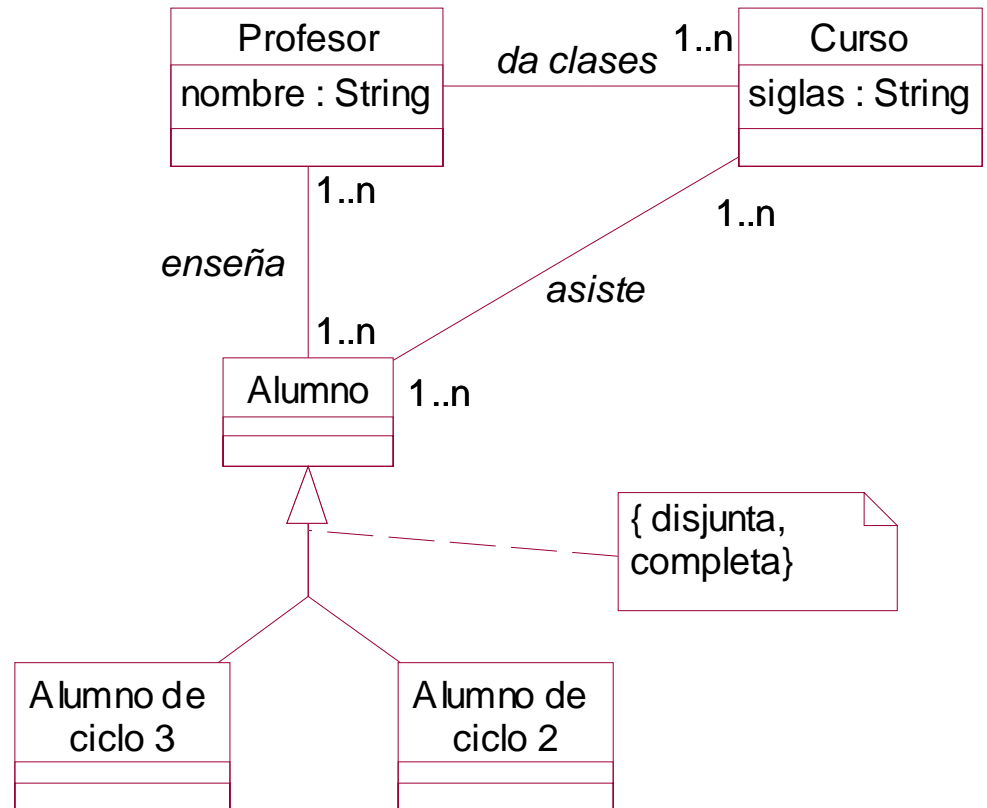
Curso = {C1, C2}

Alumnos = {A1, A2, A3, A4, A5}

String = { "jose", "PL2", ... }

da_clase = { (P1, C1), (P2, C2) }

...



Ontologías

Interpretación mediante conjuntos de instancias

Alumno de ciclo 2 \cap Alumno de ciclo 3 = \emptyset

Alumno de ciclo 2 \cup Alumno de ciclo 3 = Alumno

Alumno de ciclo 2 \subseteq Alumno

Alumno de ciclo 3 \subseteq Alumno

da_clases \subseteq Profesor \times Curso

$\forall c: \text{Curso} \exists_1 p: \text{Profesor} \mid \text{da_clase}(p,c)$

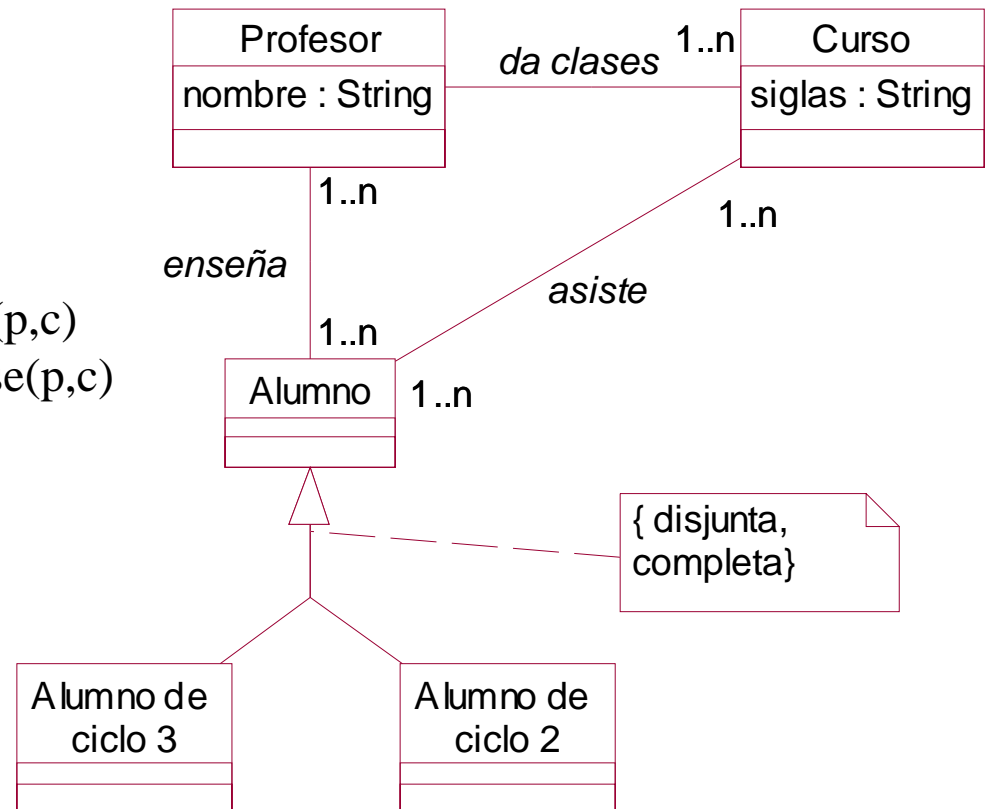
$\forall p: \text{Profesor} \exists_{>=1} c: \text{Curso} \mid \text{da_clase}(p,c)$

enseña \subseteq Profesor \times Alumno

...

asiste \subseteq Alumno \times Curso

...



Lenguajes para formalizar Ontologías

¿Qué formalizaciones se han propuesto?

Ontología \Leftrightarrow Teoría Lógica

¿Qué clases de lógicas?

1. Lógicas de Alto Orden. (Expresividad 😊😊😊, Programación 😞😞😞)
2. Lógicas de Primer Orden. (Expresividad 😊😊, Programación 😞)
3. **Lógicas de Descripciones.** (Expresividad 😊, Programación 😊)
4. Datalog & Programación Lógica. (Expresividad 😊, Programación 😊😊)

Lenguajes para formalizar Ontologías

Lógica de Descripciones ALCUE

(“expresividad reducida, tratamiento automático efectivo”)

Δ dominio de interpretación (conjunto de instancias)

1. A (se interpreta como subconjunto de Δ)
2. R (se interpreta como subconjunto de $\Delta \times \Delta$)
3. F (se interpreta como el conjunto vacío)
4. T (se interpreta como Δ)
5. $\neg C$ (se interpreta como Δ menos las instancias en C)
6. $C \cap D$ (se interpreta como individuos de C y D)
7. $C \cup D$ (se interpreta como individuos en C o D)
8. $\forall R.C$ (se interpreta como $\{x \mid \forall y R(x,y) \Rightarrow C(y)\}$)
9. $\exists R.C$ (se interpreta como $\{x \mid \exists y R(x,y) \wedge C(y)\}$)

Lenguajes para formalizar Ontologías

Base de Conocimiento:

$$\Sigma = (\text{Tbox}, \text{Abox})$$

Axiomas terminológicos (Tbox):

$C \subseteq D$ (se interpreta como que el conjunto de instancias C están incluidas en D),

$C \equiv D$ (se interpreta como que el conjunto de instancias en C y D coinciden)

$\text{Estudiante} \equiv \text{Person} \cap \exists \text{Nombre.String} \cap \exists \text{Direccion.String} \cap \exists \text{Matricula.Curso}$

$\text{Estudiante} \subseteq \exists \text{Matricula.Curso}$

Axiomas de pertenencia (Abox): $C(a)$, $R(a,b)$ (notación equivalente $a:C$ y aRb)

$\text{Estudiante}(\text{jose})$

$\text{Matricula}(\text{jose}, \text{pl2})$

$(\text{Estudiante} \cap \text{Profesor})(\text{francisco})$

Lenguajes para formalizar Ontologías

Modelo para Σ : interpretación que satisface los axiomas en Σ

Implicación Lógica: $\Sigma \models \rho$ si todo modelo para Σ es modelo para ρ

Tbox: $\exists \text{Enseña.Curso} \subseteq \neg \text{Ingeniero} \cup \text{Profesor}$

Abox: $\text{Enseña}(\text{jose}, \text{PL2}), \text{Curso}(\text{PL2}), \text{Ingeniero}(\text{jose})$

$\Sigma \models \text{Profesor}(\text{jose})$

Tbox: $\exists \text{Enseña.Curso} \subseteq \text{Ingeniero} \cup \text{Profesor}$

Abox: $\text{Enseña}(\text{jose}, \text{PL2}), \text{Curso}(\text{PL2}), \text{Ingeniero}(\text{jose})$

$\text{¿}\Sigma \models \text{Profesor}(\text{jose})\text{?}$ $\text{¿}\Sigma \models \neg \text{Profesor}(\text{jose})\text{?}$

Lenguajes para formalizar Ontologías

Problemas interesantes:

1. Satisfactibilidad de una base de conocimiento:

$\Sigma \models \perp$ (¿ Σ tiene modelo?)

2. Satisfactibilidad de un concepto:

$\Sigma \models C \equiv \perp$ (¿existe un modelo de Σ tal que C tiene instancias?)

3. Subsunción :

$\Sigma \models C \subseteq D$ (todo modelo de Σ , ¿las instancias de C están incluidas en D ?)

4. Chequeo de instancia :

$\Sigma \models C(a)$ (en todo modelo de Σ , ¿ a es instancia de C ?)

5. Instanciación: $\{a \mid \Sigma \models C(a)\}$ (¿instancias de un concepto C ?)

6. Realización: $\{C \mid \Sigma \models C(a)\}$ (¿a qué conceptos pertenece la instancia a ?)

Lenguajes para formalizar Ontologías

Reducción a problemas de satisfacibilidad:

Satisfactibilidad de un concepto:

$\Sigma \models C \equiv \perp$ equivale a decidir si existe x tal que $\Sigma \cup \{ C(x) \}$ tiene modelo

Subsunción :

$\Sigma \models C \subseteq D$ equivale a decidir si $\Sigma \cup \{ (C \cap \neg D)(x) \}$ no tiene modelo

Chequeo de instancia :

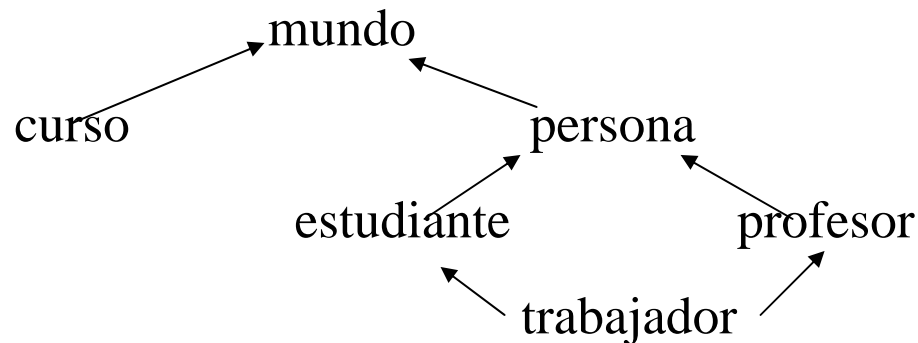
$\Sigma \models C(a)$ equivale a decidir si $\Sigma \cup \{ \neg C(a) \}$ no tiene modelo

Lenguajes para formalizar Ontologías

Taxonomía :

Dado que la subsunción es una relación de orden parcial en el espacio de conceptos.

La taxonomía es la mínima relación en el espacio de conceptos tal que su clausura reflexivo-transitiva es la relación de subsunción.



Clasificación :

Es la tarea de localizar nuevos conceptos en una taxonomía.

Lenguajes para formalizar Ontologías

¿Se puede decidir la satisfactibilidad de forma automática?

Sí, mediante procedimientos basados en cálculos sobre tableros:

Si una fórmula es satisfactible, el procedimiento construye un modelo de la fórmula.

Dicho modelo se construye incrementalmente, descomponiendo y reescribiendo la fórmula y comprobando todas las posibilidades de existencia de modelos.

Lenguajes para formalizar Ontologías

¿Cómo funciona el Tableaux Calculus?

1. Normalización:

Transformar Σ en un sistema de restricciones S .

2. Descomposición + Reescritura:

Descomponer S aplicando las denominadas reglas de terminación (completion rules).

3. Conclusión:

Las distintas ramas producidas en paso 2. Pueden acabar en contradicción o en una rama a la que no se puede aplicar más reglas. Si al menos una rama se ha completado entonces Σ tiene modelo.

Lenguajes para formalizar Ontologías

Ejemplo

$$S = \exists y (p(y) \wedge \neg q(y)) \wedge \forall z (p(z) \vee q(z))$$

$$1. \exists y (p(y) \wedge \neg q(y)) \wedge \forall z (p(z) \vee q(z))$$

$$1.1 \exists y (p(y) \wedge \neg q(y))$$

$$1.2 \forall z (p(z) \vee q(z))$$

$$1.1.1 p(y) \wedge \neg q(y)$$

$$1.1.1.1 p(y)$$

$$1.1.1.2 \neg q(y)$$

$$1.2.1 p(y) \vee q(y)$$

$$1.2.1.1 p(y)$$

$$1.2.1.2 q(y)$$

<Completada>

<Contradicción>

S es satisfactible y el modelo es $\Delta = \{y\}$, $p(y) = \text{cierto}$, $q(y) = \text{falso}$

Lenguajes para formalizar Ontologías

¿Cómo funciona el Tableaux Calculus (II)?

1. Normalización mediante Forma Normal Negativa

Se trata de obtener una fórmula equivalente en el que las negaciones sólo pueden aparecer aplicadas a conceptos.

$$\neg(C \cup D) \Leftrightarrow \neg C \cap \neg D$$

$$\neg(C \cap D) \Leftrightarrow \neg C \cup \neg D$$

$$\neg\forall R.C \Leftrightarrow \exists R.\neg C$$

$$\neg\exists R.C \Leftrightarrow \forall R.\neg C$$

Lenguajes para formalizar Ontologías

¿Cómo funciona el Tableaux Calculus (II)?

2. Descomposición.

Regla AND: $S \rightarrow \{x:C, x:D\} \cup S$ si $x:(C \cap D)$ está en S

Regla OR: $S \rightarrow \{x:C\} \cup S$ si $x:(C \cup D)$ está en S

$S \rightarrow \{x:D\} \cup S$ si $x:(C \cup D)$ está en S

Regla PARA-TODO: $S \rightarrow \{y:C\} \cup S$

si (a) $x:\forall R.C$ está en S

(b) xRy está en S

Regla EXISTE: $S \rightarrow \{xRy\} \cup \{y:C\} \cup S$

si (a) $x:\exists R.C$ está en S ,

(b) y es una variable nueva,

(c) no hay z tal que xRz e $z:C$ estén en S

Lenguajes para formalizar Ontologías

¿Cómo funciona el Tableaux Calculus (II)?

3. Conclusión.

Se dice que un sistema S es insatisfactible (no tiene modelo) si contiene un par de conceptos de la forma: $\{x:C, x:\neg C\}$

“profesor x que da clase en un conjunto de cursos y no da clase en alguno de éstos”

$x:(\forall da_clases.Curso) \cap (\exists da_clases. \neg Curso)$

$(\forall da_clases.Curso)(x)$ (Regla AND)

$(\exists da_clases. \neg Curso)(x)$ (Regla AND)

$x da_clases z$ (Regla EXISTE)

$\neg Curso(z)$ (Regla EXISTE)

$Curso(z)$ (Regla PARA-TODO)

< insatisfactibilidad >

Lenguajes para formalizar Ontologías

¿Cómo funciona el Tableau Calculus (II)?

3. Conclusión.

jose:(Profesor \cap \forall da_clases.Curso)

jose da_clases PL1

jose: \forall da_clases.Curso (Regla AND)

jose:Profesor (Regla AND)

PL1:Curso (Regla PARA-TODO)

< satisfactibilidad >

Lenguajes para formalizar Ontologías

Algunas extensiones a ALCUE

10. Cardinalidad (N) $\geq nR \{x \mid \#\{y \mid xRy\} \geq n\}$

$\leq nR \{x \mid \#\{y \mid xRy\} \leq n\}$

11. Cardinalidad Cualificada (Q) $\geq nR.C \{x \mid \#\{y \mid xRy \wedge C(y)\} \geq n\}$

$\leq nR.C \{x \mid \#\{y \mid xRy \wedge C(y)\} \leq n\}$

12. Enumeración (O) $\{a,b,c,d\}$

13. Rol funcional (F) $f:C$

Lenguajes para formalizar Ontologías

Algunas extensiones a ALCUE

14. Conjunción de roles (composición de roles): $R \cap S \quad \{(x,y) \mid xRy \text{ e } xSy\}$

15. Disyunción de roles (composición de roles): $R \cup S \quad \{(x,y) \mid xRy \text{ o } xSy\}$

16. Negación de rol: $\neg R \quad \{(x,y) \mid \neg xRy\}$

17. Inverso de un rol: $-R \quad \{(x,y) \mid yRx\}$

18. Composición de roles: $R \circ S \quad \{(x,y) \mid xRz \text{ e } zSy\}$

18. Rango condicionado: $R|C \quad \{(x,y) \mid xRy \text{ e } y:C\}$

Problemas de Investigación

1. Ontologías y Contextos

Paolo Bouquet, F. Chiunchiglia, F. Van Harmelen, L. Serafini, H. Stuckenschmidt.
Contextualizing Ontologies.

Journal of Web Semantics. 1 (4) pp. 1-19, 2004

2. Lógicas de Descripción Distribuida

Alex Borgida and L. Serafini.

Distributed Description Logic: Assimilating Information from Peer Sources.

Journal of Data Semantics. LNCS 2800 Springer-Verlag pp. 153-184

3. Modelado Conceptual con Lógicas de Descripción

Alex Borgida and R. Brachman

Distributed Description Logic: Assimilating Information from Peer Sources.

Description Logic Book. Cambridge Press 2002 pp. 359-381 o también en

<http://www.inf.unibz.it/~franconi/dl/course/dlhb/dlhb-10.pdf>

Problemas de Investigación

4. OWL y su relación con las Lógicas de Descripción

I. Horrocks, P. F. Patel-Schneider and F. Van Harmelen.

From SHIQ and RDF to OWL.

Journal of Web Semantics, 1 (1) 2003

5. El Problema de la Coordinación Semántica

P. Bouquet, L. Serafini and S. Zonobini.

Peer-to-Peer Semantic Coordination.

Journal of Web Semantics, (To be published in 2005)

6. La Estandarización a través de OWL

Recomendación W3C Feb 2004

OWL Web Ontology Language.

<http://www.w3.org/TR/2004/REC-owl-guide-20040210/>

Problemas de Investigación

7. La necesidad de Inferencia Aproximada

H. Stuckenschmidt and F. Van Harmelen.

Aproximating Terminological Queries

Journal of Web Semantics, 1 (1) 2003

8. Herramientas: Racer y Protegé

V. Haarslev and R. Möller.

RACER User's Guide and Reference Manual

<http://www.fh-wedel.de/~mo/racer/>

The Protege Ontology Editor and Knowledge Acquisition

<http://protege.stanford.edu/>

Web Semántica

Asignatura Doctorado

Marzo-Abril 2005
