

Propuesta de una ontología para la especificación de procesos

Jose Alberto Arauzo¹, Juan José de Benito¹, Ricardo del Olmo²

¹ Dpto. de Organización de Empresas y CIM. ETSII. Universidad de Valladolid. Pº del Cauce s/n, 47011 Valladolid. {arauzo, debenito}eis.uva.es

² Dpto. de Ingeniería Civil. ETSII. Universidad de Valladolid. Escuela Politécnica Superior, 47011 Valladolid. rdelolmo@ubu.es

Resumen

Con la elaboración de ontologías se pretende capturar conocimiento para que pueda ser utilizado en diferentes sistemas, independientemente de la tarea que pretendan resolver. Para ello se concretará con algún lenguaje formal, un vocabulario (conjunto de términos) común a utilizar por todos los sistemas que deban manejar conocimiento sobre un dominio determinado. En este artículo se propone una ontología básica, concebida para describir procesos, y que se ha desarrollado para facilitar el intercambio de información en sistemas de fabricación basados en agentes. La ontología se ha implementado utilizando las herramientas proporcionadas por JADE, una plataforma basada en la tecnología JAVA utilizada para desarrollar el sistema multiagente al que va destinada la ontología.

Palabras clave: ontología, especificación de procesos, fabricación, sistemas multiagente

1. Introducción

Según la Real Academia Española, el término ontología se define como “*la parte de la metafísica que trata del ser en general y de sus propiedades trascendentales*”. En Inteligencia Artificial, en cambio, tiene otras connotaciones. Algunas de ellas son: “*especificación explícita y formal sobre una conceptualización consensuada*” (Studer 1998); “*conjunto de términos estructurados jerárquicamente que describen un dominio, y que servirán de esqueleto sobre el cual se construye la base de conocimientos*” (Swartout 1997); “*significados que describen explícitamente la conceptualización del conocimiento representado en una base de conocimientos*” (Bernaras 1996).

Según (Noy 2001) una ontología va a permitir: (1) compartir conocimiento común sobre la estructura de las cosas, (2) reutilizar el conocimiento del dominio, (3) explicitar suposiciones sobre el dominio, (4) separar el conocimiento del dominio del conocimiento operacional, y (5) posibilitar el análisis del conocimiento del dominio.

Las ontologías, además de ser la base para los sistemas basados en conocimiento, actualmente el ámbito de aplicación de las ontologías abarca también otras áreas: procesamiento de lenguaje natural, Web semántica, comercio electrónico, gestión de conocimientos, etc.

En nuestro caso la ontología ha sido desarrollada para facilitar la comunicación de agentes en un sistema de fabricación holónico (Arauzo 2008), así como para garantizar la interoperatividad de estos agentes con otros pertenecientes a otras plataformas. Es importante que las ontologías con las que los agentes conceptúan un determinado dominio sean similares. Esto garantizará la misma interpretación del contenido del mensaje por los agentes implicados

en la comunicación. Una vez que los agentes que interactúan se han puesto de acuerdo en una ontología común, se espera que la usen para interpretar las comunicaciones.

Para un agente, una ontología definirá su entorno, y aunque no sea estrictamente necesario la elaboración de ontologías para el desarrollo de sistemas multiagente será recomendable. La ontología facilitará compartir y reutilizar el conocimiento, además de permitir separarlo de los aspectos operacionales. En nuestro caso, los agentes deben almacenar y transmitir conocimiento sobre procesos de fabricación, por ello se desea definir una ontología con ese fin.

2. Ontologías: taxonomía

Dependiendo del punto de vista que se desee considerar se pueden establecer diferentes clasificaciones. En [Lozano 02] encontramos algunas de las propuestas por diferentes autores. Nosotros vamos a citar las dos clasificaciones que nos parecen más interesantes: en función del formalismo utilizado y en función del grado de reusabilidad.

Atendiendo al formalismo se puede diferenciar entre: ontologías ligeras (*Lightweight Ontologies*) y ontologías pesadas (*Heavyweight Ontologies*). Las pesadas, a diferencia de las ligeras, contienen axiomas (a más axiomas más pesadas) (Gómez 1999).

En función del grado de reutilización, en Van Heist (1997) se propone cuatro tipos básicos de ontologías. Ordenadas de mayor a menor reusabilidad son: (1) ontologías de representación, que proporcionan el vocabulario necesario para modelar otras ontologías; (2) ontologías genéricas o meta-ontologías, que proporcionan términos genéricos reutilizables en diferentes dominios (estado, evento, acción, componente, etc.); (3) ontologías de dominio, que expresan conceptualizaciones específicas para dominios particulares, definidas generalmente como especializaciones de conceptos existentes en ontologías genéricas; y (4) ontologías de aplicación, que contienen las definiciones necesarias para modelar los conocimientos requeridos por una aplicación particular (incluyen conceptos tomados de ontologías de dominio y genéricas).

Desde hace unos años se está produciendo un cierto esfuerzo en el desarrollo tanto de lenguajes de especificación (formalismos) como de ontologías. Dentro de los primeros se pueden citar: Ontolingua (Ontolingua Web 2010), OKBC Protocol (*Open Knowledge Base Connectivity*) (OKBC Web 2010), OCML (*Operational Conceptual Modeling Language*) (OCML Web 2010), XML (*eXtended Markup Language*), etc. (Corcho 2000). Entre las segundas se pueden mencionar 'Frame Ontology' del 'ontolingua server' como ontología de representación; y *Meteorology* como ontología genérica. Como ontologías de dominio se puede mencionar EngMath en el dominio matemático, PhysSys en el dominio físico (Borst 1995). La ontología de dominio que más se acerca al propósito de un lenguaje común para el campo de fabricación es PSL (*Process Specification Language*) (PSL Web 2010).

3. Un ejemplo de ontología para la especificación de procesos: PSL

Originalmente el proyecto *Process Specification Language* (PSL) (Schlenoff 1996) se centró en el desarrollo de una representación neutral de los procesos de fabricación. Proporciona definiciones rigurosas e inequívocas de los conceptos necesarios para especificar procesos de fabricación, con el fin de permitir el intercambio de información entre diferentes aplicaciones. La representación de la ontología se realiza usando las especificaciones de KIF (Knutilla 1997).

Con PSL se pretende especificar un conjunto de términos de significado explícito, eliminando cualquier intuición implícita que pueda ser fuente de ambigüedad y confusión. Por ello, PSL proporciona expresiones precisas (axiomas y definiciones) de las propiedades básicas de la información contenida en el lenguaje.

PSL posee esencialmente dos niveles: el núcleo (primer nivel) y las extensiones (segundo nivel). El núcleo de PSL consiste en un conjunto de conceptos comunes a todas las aplicaciones de la fabricación (Schlenoff 1996). Incluye conceptos tales como recursos, actividades e intervalos del tiempo, que permitirán el intercambio de especificaciones de procesos en entornos genéricos y sencillos. Para ampliar la capacidad de la ontología se permite la posibilidad de realizar extensiones de estos conceptos base (segundo nivel). Inicialmente, estas extensiones derivan de una serie de puestas en práctica (en diferentes tipos de proceso) y permiten adaptar la ontología a diferentes entornos de fabricación.

El núcleo de PSL se basa en un sistema primitivo de conceptos (actividades, objetos, instantes de tiempo) y sus interrelaciones. Al ser estos conceptos primitivos, no se podrán definir a partir de otros y sólo se podrán indicar sobre ellos verdades o axiomas. Estas verdades proporcionan al usuario el nivel de entendimiento necesario para poder utilizar los conceptos primitivos con eficacia. Ejemplos de las verdades proporcionadas para los conceptos primitivos (expresadas en KIF) son:

Axioma 1: Todo es una actividad, objeto, o instante de tiempo.

(or (activity ?x) (object ?x) (timepoint ?x))

Axioma 2: El punto de inicio de cada actividad u objeto es anterior o igual a su punto final.

(=> (or (Activity ?x) (Object ?x)) (BeforeEq (Beginof ?x) (Endof ?x)))

Con ellos se pueden definir el resto de conceptos de la ontología PSL. Por ejemplo, el concepto de ‘tareas en serie’ se puede definir a partir de objeto, actividad e instante de tiempo.

4. Desarrollo de ontologías

Aunque para garantizar la máxima utilidad de las ontologías, lo deseable es que sean desarrolladas con suficiente rigor; en la realidad, y dependiendo del propósito para el que sean creadas, serán elaboradas con distintos grados de formalidad. Así se pueden encontrar desde las más informales expresadas en lenguaje natural, hasta las rigurosamente formales expresadas en lenguajes con lógica de primer orden (Lozano 2002). En los casos más formales las ontologías se concretan identificando términos que se suelen organizar según la siguiente clasificación.

- Clase: conceptos del dominio o categorías que representan a colecciones de objetos que se identifican como del mismo tipo.
- Subclase: clase que indica un subtipo dentro de un tipo (clase padre o superclase). Representan conceptos más específicos que la superclase. La relación entre una subclase y su clase padre se le denomina herencia (la subclase hereda características de la superclase).
- Instancias: constituyen elementos determinados de una clase o concepto.

- Propiedades (*Slots*): características o atributos del concepto (clase). Se especifican indicando los siguientes aspectos: (1) tipo de característica, (2) valor de la característica (3) restricciones (*facet*) o limitaciones de los valores que puede tomar un *slot*, (4) valor por defecto que toma el *slot*, (5) cardinalidad o número de elementos que puede contener un slot individual (máximo y mínimo).
- Variables: términos que representan un elemento genérico no conocido a priori.
- Relaciones (en muchos casos definidas como predicados): representan interacciones entre conceptos del dominio. Suelen aparecer las relaciones del tipo: subclase-de, parte-de, conectado-a, etc. Pueden ser binarias o n-arias dependiendo de los argumentos que tomen.
- Funciones: son un tipo especial de relación, en la que para un conjunto de argumentos de entrada se obtiene un único argumento de salida.
- Axiomas: son fórmulas, en la mayoría de las ocasiones expresadas en lógicas de primer orden, que se usan para modelar sentencias que son siempre verdad.

Para definir ontologías (establecer clases, subclases, slots, etc.), se pueden encontrar diferentes metodologías (Fernández 1999, Beck 2002), que consisten en procedimientos para identificar clases, subclases, slots, etc., de un dominio de aplicación. Noy (2001) propone un método muy sencillo para definir ontologías, que además está soportado por la herramienta *Protege* (Protege WEB 2010). Esta metodología pretende llegar a una representación de los conceptos y sus relaciones del dominio de aplicación y para ello propone siete pasos: (1) determinar el dominio de la ontología y su alcance, (2) considerar la posibilidad de reutilización de otras ontologías, (3) enumerar los términos importantes del dominio, (4) definir las clases y sus relaciones, (5) definir las propiedades de las clases (*slots*), (6) concretar los aspectos de los *slots*, y (7) crear las instancias de las clases.

5. Modelo conceptual propuesto para una ontología básica de especificación de procesos

Tal y como se propone en (Noy 2001), antes de identificar los tipos de objetos (clases), subtipos, *slots*, etc., de la ontología que se pretende definir, conviene, en primer lugar, determinar el dominio de aplicación, y tras ello decidir si se reutiliza o extiende alguna ontología ya existente. El dominio lo tenemos bastante detallado; la ontología está destinada a representar la información que los agentes del sistema de gestión de la producción necesitan intercambiar para transmitir información sobre procesos.

Con el dominio descrito, y antes de continuar, conviene buscar ontologías de dominio que, bien directamente o bien mediante extensión, se adapten a las necesidades de representación. Una de ellas es PSL, bastante adaptable al sistema que se pretende desarrollar. Aun así no la utilizaremos directamente, ya que en algunos puntos excede las exigencias del sistema propuesto. Además, aunque está definida, para sacarla el máximo partido sería necesario implementarla mediante las utilidades JADE (plataforma basada en la tecnología JAVA, utilizada para desarrollar el sistema multiagente al que va destinada la ontología), lo que podría suponer más esfuerzo que hacerlo para una más ajustada a nuestro problema.

Una vez tomada esta decisión, la siguiente tarea consistirá en enumerar los términos más importantes del dominio y con éstos identificar las clases, sus características y sus relaciones

Un tipo de ítem ('Producto' o 'Componente') tiene definidos uno o varios procesos de producción (Proceso) con los que puede ser fabricado. Cada proceso se especifica indicando una serie de operaciones (que pueden ser compartidas por varios procesos) y un conjunto de materias primas sobre las que se aplican las operaciones. En cada proceso, una operación tiene unas restricciones de precedencia que determinarán su orden de ejecución dentro del proceso. Las operaciones y micro-operaciones (Tarea) pueden ser ejecutadas en varias máquinas, aunque con diferente tiempo de procesamiento (tiempoProceso).

Una operación constará de un conjunto de 'microoperaciones' (operaciones elementales) que pueden ser de dos tipos, de transformación o de montaje. Una 'Transformación' será una instancia de un 'TipoTransformación' donde se habrán asignado valores a sus atributos 'AtributoTransformación'. Un 'Montaje' será una instancia de un 'TipoMontaje' donde se indiquen los tipos de componentes que van a ser montados.

6. Ontología básica para la especificación de procesos

Como muchas de las plataformas de desarrollo de sistemas multiagente, JADE permite la definición de ontologías, facilitando así el intercambio de conocimiento entre agentes y la solicitud de servicios. El modelo de JADE para la elaboración de ontologías, acorde con la especificación de lenguajes de contenido FIPA-SL, permite definir tres tipos de elementos (Figura 2): conceptos (*Concept*), predicados (*Predicate*) y acciones (*AgentAction*). De éstos, sólo los dos últimos pueden introducirse directamente como contenidos de los mensajes. las acciones en los mensajes que solicitan la realización de alguna tarea y los predicados (Proposition en FIPA-SL) en los mensajes que informan sobre la veracidad o falsedad de algo. Los conceptos podrán incluirse como *slot* (campo, descriptor o atributo) de otros elementos (Caire 2003).

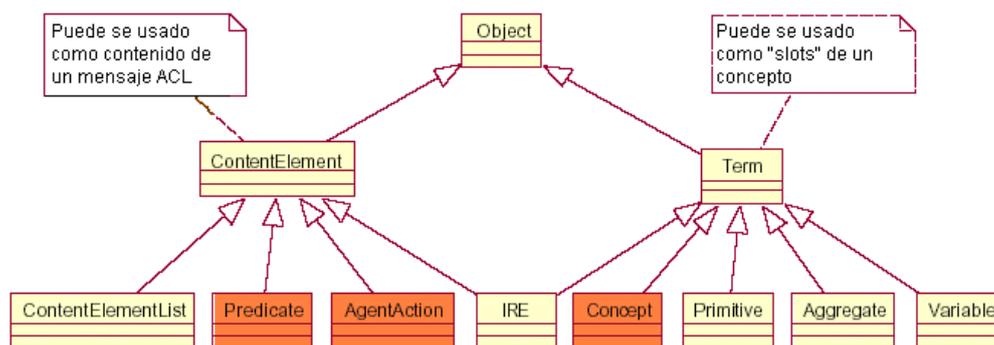


Figura 2. Modelo de contenidos JADE (Caire 2003)

La definición de una ontología consistirá, por lo tanto, en la especificación de predicados, acciones y conceptos, lo que se realizará definiendo para cada uno de ellos: nombre del tipo de elemento o *schema*, *superschema* (tipo del que extienden o heredan características) y los *slots*. Además, cada *slot* se deberá concretar indicando: nombre, tipo, cardinalidad (número de elementos del tipo especificado que pueden ir asociados al *slot*), y restricciones (limitación de los valores que puede tomar el *slot*). Para representar los diferentes conceptos, predicados y acciones de la ontología se utilizará la notación no normalizada de la Figura 3.

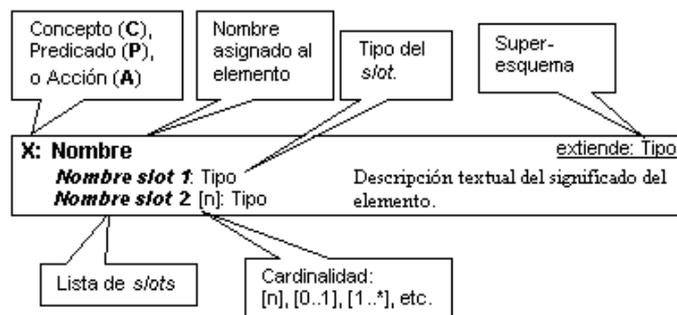


Figura 3. Representación de conceptos predicados y acciones

Desde el punto de vista de la implementación, la definición de ontologías se realizará elaborando una clase JAVA, que extiende la clase de JADE ‘jade.content.onto.Ontology’ que contiene los métodos que permiten ir definiendo los predicados, conceptos y acciones. Además, esta clase permite asignar clases JAVA a cada uno de los elementos definidos, lo que facilitará notablemente la posterior elaboración e interpretación de los contenidos de los mensajes. Los agentes JADE poseen un gestor de contenidos (*ContentManager*) que, una vez configurado, podrán transformar los objetos que representen predicados o acciones (objeto de una clase asociada a un elemento de la ontología) en contenidos de mensajes ACL. Además, el *ContentManager* también permitirá la transformación inversa.

Dado que la representación de la ontología, según las especificaciones de JADE, no coincide exactamente con la representación facilitada por los modelos conceptuales elaborados para el dominio de la aplicación, será necesario realizar una transposición de los elementos del modelado conceptual a los elementos de la ontología. Si la identificación de conceptos resulta sencilla (por lo general cada clase del diagrama conceptual se corresponderá con un concepto), con los predicados, acciones y *slots*, no sucederá lo mismo.

Los *slots* de cada concepto se podrán determinar a partir de los atributos de la clase del diagrama asociada al concepto, o de las relaciones que tienen estas clases con otras. Pero las relaciones no sólo se podrán especificar mediante la asignación de *slots* a los conceptos relacionados, también se podrán traducir en predicados donde sus *slots* representarán las clases relacionadas. Así, por ejemplo, los conceptos de ‘Operacion’ y ‘Microoperacion’ y su relación (Figura 4) se podrán representar de dos formas: (1) definiendo un slot ‘*microoperaciones*’ para el concepto ‘Operacion’ con cardinalidad [1..*] donde se indiquen las micro-operaciones que forman la operación (Figura 5), o (2) definiendo un predicado que indique que una micro-operación pertenece a una operación, representando así la relación entre ambos conceptos (Figura 6). La selección de una u otra alternativa dependerá de las necesidades de comunicación entre agentes.

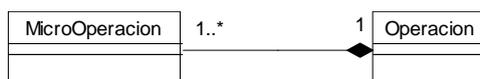


Figura 4. Ejemplo de relación entre conceptos



Figura 5. Posible ontología para el modelo conceptual de la Figura 4

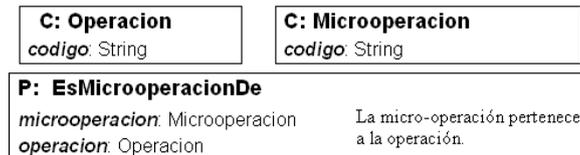


Figura 6. Posible ontología para el modelo conceptual de la Figura 4

Siguiendo este procedimiento se han definido los siguientes conceptos (Figura 7) y predicados (Figura 8). Las acciones que se utilizan básicamente como contenido de los mensajes donde se solicita una actividad al agente participante se definen en función de las actividades que pueden realizar los agentes.

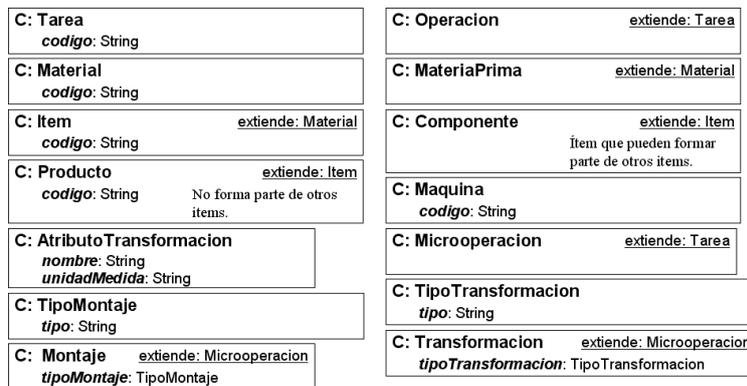


Figura 7. Conceptos de la Ontología

P: EsOperacionItemProceso <i>item</i> : Item <i>proceso</i> : int <i>operacion</i> : Operacion	La operación pertenece al proceso de fabricación del ítem referenciado.
P: EsMateriaPrimalItemProceso <i>item</i> : Item <i>proceso</i> : int <i>materiaPrima</i> : MateriaPrima <i>cantidad</i> : int	La materia prima indicada pertenece al proceso de fabricación del ítem referenciado.
P: PuedeEjecutar <i>item</i> : Item <i>tarea</i> : Tarea <i>maquina</i> : Maquina <i>tiempo</i> : long	La máquina puede ejecutar la tarea asociada al ítem en el tiempo indicado.
P: EsAtributoTransformacion <i>atributoTransformacion</i> : AtributoTransformacion <i>tipoTransformacion</i> : TipoTransformacion	El atributo de transformación describe al tipo de transformación.
P: PuedeTransformar <i>tipoTransformacion</i> : TipoTransformacion <i>maquina</i> : Maquina	La máquina puede realizar el tipo de transformación indicado.
P: PuedeMontar <i>tipoMontaje</i> : TipoMontaje <i>maquina</i> : Maquina	La máquina puede realizar el tipo de montaje indicado.
P: EsMicrooperacion <i>item</i> : Item <i>operacion</i> : Operacion <i>microOperacion</i> : MicroOperacion	Micro-operación asociada a una operación de un ítem.
P: EsValorAtributo <i>item</i> : Item <i>operacion</i> : Operacion <i>transformacion</i> : Transformacion <i>AtributoTransformacion</i> : AtributoTransformacion <i>valor</i> : float	Asocia un valor a un atributo de transformación, para una transformación que forma parte de una operación de un ítem.
P: EsComponenteMontaje <i>item</i> : Item <i>operacion</i> : Operacion <i>montaje</i> : Montaje <i>componente</i> : Componente <i>cantidad</i> : int	Indica que una cantidad de un determinado componente, se monta en un montaje que forma parte de una operación de un ítem.

Figura 8. Predicados de la Ontología

7. Conclusiones

Las ontologías permiten describir y relacionar los objetos con los que se trabaja en un determinado dominio mediante la definición de un lenguaje formal, y facilita, entre otras cosas, la comunicación entre diferentes partes de un sistema o entre diferentes sistemas, y de esta forma poder compartir conocimiento.

Nosotros hemos definido una ontología para la especificación de los procesos de fabricación. La ontología está destinada a soportar la comunicación y la transmisión de conocimiento entre los agentes de un sistema de holónico (basado en agentes) para la gestión de un sistema de fabricación.

El desarrollo de la ontología ha sido muy útil, ya que nos ha permitido separar perfectamente el modelo de conocimiento de la operativa de los agentes. Una vez definida la ontología ya no es necesario pensar en cómo plasmar el conocimiento, solo en cómo interaccionan los agentes y cómo reaccionan ante los mensajes. Esta ontología ha demostrado ser muy expresiva, ha permitido representar todo el conocimiento necesario de nuestra aplicación, bien directamente o bien mediante extensiones.

Referencias

- Arauzo J.A. (2008). Control Distribuido de Sistemas de Fabricación Flexible. Un Enfoque Basado en Agentes. Tesis Doctoral. Universidad de Valladolid.
- Baclawski, K.; Kokar, M.; Kogut, P.; Hart, L.; Smith, J.; Holmes, W.; Letkowski, J.; Aronson M. (2001). Extending UML to Support Ontology Engineering for the Semantic Web. Proc. of the Fourth International Conference on UML (UML2001), Toronto, October.
- Beck, H.; Pinto H.S. (2002). Overview of Approach, Methodologies, Standards, and Tools for Ontologies. Third Agricultural Ontology Service Workshop, Florida, USA, pp 58-62.
- Bellifemine, F.; Caire, G.; Trucco, T.; Rimassa, G.; (2003). *JADE Programmer's Guide*, JADE Tutorial, TILab S.p.A.
- Bernaras, A.; Laresgoti, L.; Corera J.(1996). Building and Reusing Ontologies for Electrical Network Applications. European Conference of Artificial Intelligence (ECAI'96), pp. 298-302. Budapest, Hungary
- Borst, P.; Akkermans, J.; Pos, A.; Top J.(1995). The PhysSys ontology for physical systems. Ninth International Workshop on Qualitative Reasoning QR'95, pp. 11-21. University of Amsterdam.
- Caire G. (2003). Application-Defined Content Languages and Ontologies, JADE Tutorial, TILab S.p.A.
- Corcho, O.; Gómez A.; (2000). A Roadmap to Ontology Specification Languages. Proceedings of 11th European Workshop on Knowledge Acquisition. Modelling and Management (EKAW'00), Jean-Les-Pins. Francia, pp. 80-96.
- Fernández, M. (1999). Overview of Methodologies for Building Ontologies. Proceedings of the IJCAI-99 workshop on Ontologies and Problem Solving Methods (KRR5). Stockholm, Sweden, August 2.
- Gómez, A.; Benjamins A.R. (1999). Applications of Ontologies and Problem-solving Methods. AI Magazine, vol. 20, N° 1, pp. 119-122.
- Knutilla A. (1997). Process Specification Language: Analysis of Existing Representations. Working Paper NISTIR 6133, National Institute of Standards and Technology, Gaithersburg MD.
- Larman, C. (2003). UML y Patrones. Una Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado, Pearson Educación, S. A, Madrid.
- Liu, W. Liu, Z.T. Shao K. (2003). UML Based Domain Ontology Modelling for Multiagent System. Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xian, November.
- Lozano, A. (2002). Métrica de Idoneidad de Ontologías, Tesis Doctoral, Escuela Politécnica de Cáceres, Universidad de Extremadura.

Noy, N.F.; Mcguinness L.D. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05.

OCML Web (2010) <http://technologies.kmi.open.ac.uk/ocml//>

OKBC Web (2010) <http://www.ai.sri.com/~okbc/>

Ontolingua Web(2010) <http://ksl.stanford.edu/software/ontolingua/>

Protege Web (2010). The Protege Ontology <http://protege.stanford.edu/>.

PSL Web (2010) <http://www.mel.nist.gov/psl/>

Schlenoff, C.I. (1996). Unified Process Specification Language: Requirements for Modelling Process. Working Paper NISTIR 5910, National Institute of Standards and Technology, Gaithersburg MD

Studer, S; Benjamins, R; Fensel, D. (1998). Knowledge Engineering: Principles and Methods. Data and Knowledge Engineering, Vol. 25, pp. 161-197.

Swartout, B.; Patil, R.; Knight, K.; Russ T. (1997). Towards Distributed Use of Large-Scale Ontologies. AAAI'97 Spring Symposium Series on Ontological Engineering, pp. 138-148.

Van Heist, G.; Schreiber, A.; Wielinga B. (1997). Using Explicit Ontologies in KBS Development. International Journal of Human-Computer Studies, Vol.45, pp. 183-292.