

Reengenharia e Ontologias: Análise e Aplicação

Regina C. Cantele¹, Diana F. Adamatti²,
Maria A. G. V. Ferreira¹, Jaime S. Sichman^{2*}

¹InterLab – Laboratório de Tecnologias Interativas

²LTI – Laboratório de Técnicas Inteligentes

Escola Politécnica - Universidade de São Paulo

{regina.cantele,diana.adamatti,maria.alice.ferreira,jaime.sichman}@poli.usp.br

Abstract

Ontologies are a basic building block of the Semantic Web. An ontology presents a common vocabulary of a certain domain, defining precisely the meaning and relationships between its terms. As a consequence, a great number of researchers are working in method and techniques to build ontologies through automatic or semi-automatic processes, which perform knowledge acquisition from texts, dictionaries and structured and semi-structured knowledge bases. On the other hand, reverse engineering, when applied to software engineering, uses a collection of theories, methodologies and techniques to support information abstraction and extraction from a piece of software. This paper presents a preliminary proposal, which uses reverse engineering techniques applied to ontology engineering, whose goal is to reduce the time consuming task of ontology creation.

1. Introdução

Andrônico de Rodes, por volta de 50 a.C., recolheu e classificou as obras de Aristóteles que, durante muitos séculos, haviam ficado dispersas e perdidas (Chaui 1995). Desde este período vem o desejo de recolher os dados já disponibilizados pelas organizações, comunidades científicas e indivíduos em diferentes fontes de informação, como na *Web* ou em banco de dados legados, e transformá-los em informações úteis e conhecimento através do uso de ontologias.

De acordo com Guarino (1997) e Gruber (1995), uma ontologia representa um vocabulário comum de um domínio. Assim, define o significado dos termos e as relações entre eles, organizados em uma taxonomia e contém primitivas de modelagem como classes, relações, funções, axiomas e instâncias. Existem linguagens tradicionais para sua representação como: CYCL, Ontolingua, F-Logic, CML etc., e linguagens padrões para *Web* como: OIL (*Ontology Inference Layer*), DAML+OIL (*DARPA Agent Markup Language*), RDF(s), XOL (*Ontology Exchange Language*), SHOE (*Simple HTML Ontology Extensions*), XTM (*XML Topic Maps*) e OWL (*Ontology Web Language*) [Corcho e Gómez-Pérez 2003], [Gómez-Pérez e Corcho 2002], [Su e Iiebrekke 2002]. É importante ressaltar que existem diferentes conexões entre os componentes da ontologia, seus paradigmas de representação do conhecimento e suas linguagens de representação.

A engenharia de software utiliza-se dos princípios da engenharia reversa como uma coleção de teorias, metodologias e técnicas capazes de suportar a extração e abstração de informações de um software existente, produzindo documentos consistentes, quer seja a partir do código fonte, ou através da adição de conhecimento e experiência, que não poderiam ser automaticamente reconstruídos a partir do código [Benedusi *et al.* 1992]. Define-se como engenharia progressiva àquela que segue a seqüência de desenvolvimento estabelecida por uma metodologia, visando à obtenção do sistema implementado [Chikofsky e Cross II 1990].

Os engenheiros de ontologias apresentam o enfoque de aprendizado de ontologia - *Ontology Learning* - como um conjunto de métodos e técnicas usado para construir uma ontologia através de processos automáticos ou semi-automáticos de aquisição de conhecimento via textos, dicionários, bases de conhecimento, dados semi-estruturados e esquemas relacionais já existentes [Meersman *et al.* 2002], [Gómez-Pérez e Manzano-Macho 2003]. Outros termos para construção semi-automática de ontologias são geração de ontologias, mineração de ontologias, e extração de ontologias.

* Parcialmente financiado pelo CNPq, proc. 301041/95-4.

As metodologias para desenvolvimento de ontologias apresentam ciclos de vida distintos, mas as fases de aquisição e formalização das ontologias sempre aparecem em todos os ciclos. Desta forma, na aquisição constrói-se um modelo conceitual e na formalização um modelo formal.

O objetivo principal deste artigo descrever uma proposta de como combinar a reengenharia de sistemas com a engenharia de ontologias para reduzir o tempo dispendido na construção de ontologias. Desta forma, a seção 2 apresenta uma pequena revisão sobre ontologias. A seção 3 apresenta uma revisão de reengenharia e algumas metodologias para desenvolvimento de ontologias. Na seção 4, a metodologia para combinação de engenharia reversa e ontologias é apresentada. Na seção 5, estão as conclusões do artigo.

2. Ontologias

Pesquisas em ontologias têm origem na filosofia com a natureza e organização das coisas. Uma ontologia, segundo [Gruber 1995], é uma especificação explícita dos objetos, conceitos e outras entidades que se assume existirem em uma área de interesse, além das relações entre estes conceitos e restrições, expressos através de axiomas. Em Ciência da Computação, o termo ontologia refere-se a um artefato de engenharia, constituído por um vocabulário específico que descreve um modelo particular do mundo, adicionando um conjunto explícito de suposições relacionando os significados das palavras no vocabulário (Maedche *et al.* 2002). Os vocabulários são, usualmente, organizados em taxonomias.

A ontologia deve fazer uma especificação formal de uma área de conhecimento [Benjamins e Gómez-Pérez 1999]. Cinco componentes foram definidos para esta formalização [Gruber 1993]:

- i. Conceitos: podem representar qualquer coisa em um domínio, como uma tarefa, uma função, uma estratégia, etc.;
- ii. Relações: representam um tipo de interação entre os conceitos no domínio, sendo a cardinalidade sempre $n : n$;
- iii. Funções: são um caso especial de relações, sendo a cardinalidade $n : 1$;
- iv. Axiomas: são as sentenças que são sempre verdadeiras;
- v. Instâncias: são utilizadas para representar os elementos do domínio.

Atualmente existem várias linguagens desenvolvidas para a representação formal de ontologias [Gómez-Pérez e Corcho 2002], [Su e Ilebrikke 2002], entre elas:

- i. KIF (*Knowledge Interchange Format*): é uma lógica de primeira ordem monotônica, possuindo sintaxe simples e com algumas extensões para suportar um raciocinador de relações;
- ii. Ontolingua: é uma linguagem de compartilhamento de ontologias, desenvolvida por Gruber (1995) para permitir o projeto e a especificação de ontologias com semânticas lógicas baseadas no KIF;
- iii. OIL (*Ontology Inference Layer*): é uma camada de inferência e representação baseada na Web, que combina a utilização de modelagem de primitivas provenientes das linguagens baseadas em frames com a semântica formal e, ainda com serviços de raciocinador provenientes de lógicas de descrição;
- iv. XOL (*Ontology Exchange Language*): linguagem, baseada em XML, projetada para proporcionar um formato para troca de definições entre ontologias no domínio da biologia molecular;
- v. RDF (*Resource Description Framework*): recomendação do World Wide Web Consortium - W3C. Constitui-se em uma arquitetura genérica de metadados que permite descrever semanticamente recursos no contexto Web;
- vi. DAML+OIL: este padrão é a união de DARPA *Agent Markup Language* (DAML) e OIL e define uma série de construções específicas para representação de ontologias em RDF. O Consórcio W3C está atualmente construindo um padrão para representação de ontologias, o OWL (*Web Ontology Language*), que é amplamente baseado no DAML+OIL e deverá manter grande parte das suas construções.

Alguns pesquisadores uniram o formalismo existente para representação das ontologias com a UML (*Unified Modeling Language*) do OMG (*Object Management Group*) [Cranefield e Purvis 1999], [Cranefield 2001], [Baclawski *et al.* 2001, 2002], [Dutra 2001], [Guizzardi *et al.* 2002]. OMG é um grupo que fornece diretrizes para a indústria de software, através de especificações de padrões, cuja missão é promover a teoria e a prática da tecnologia de objetos para o

desenvolvimento de sistemas distribuídos. UML é uma linguagem gráfica, utilizada em Engenharia de Software para modelar sistemas orientados a objeto, que além de definir uma notação gráfica, conjunto de símbolos padrões, especifica, em detalhes, a semântica destes símbolos de tal forma que o modelo idealizado para o sistema possa ser compreendido por todos, facilitando a comunicação efetiva entre os desenvolvedores que a utilizam [OMG 2003]. Além disso, a UML é extensível; esta característica tem permitido a sua aplicação em ontologias. Outro padrão OMG utilizado é o MOF (*Meta Object Facility*) - um conjunto de serviços projetados para suportar a administração de metadados. Metadado é um dado, que descreve outro dado, com um nível maior de abstração. Exemplos de metadados são: tipos de dados nas linguagens de programação, definições de interfaces dos componentes, diagramas de análise e projeto, esquemas SQL que descrevem a estrutura de banco de dados relacionais [OMG 2001].

OMG criou, em novembro de 2000, o padrão XML *Metadata Interchange* (XMI) para representar um mecanismo padrão de troca de dados usado entre várias ferramentas, repositórios e *middleware*. Na época, isto deveria garantir a facilidade na troca de metadados entre ferramentas de modelagem baseadas no padrão OMG-UML e os repositórios de metadados baseados em OMG-MOF em ambientes heterogêneos distribuídos. Mais recentemente, o XMI tem sido usado para interpretar artefatos UML, artefatos de base de dados e *data warehouse*, definições de interfaces CORBA, interfaces e classes JAVA [OMG 2002], [Suzuki e Yamamoto 1999]. RDF e OWL são padrões para a Web Semântica, que fornecem um *framework* para gerenciamento e integração de recursos das empresas e compartilhamento e reuso de dados na Web.

Especificações destinadas a padronizar os recursos necessários para as ontologias e os padrões já existentes estão sendo realizadas pelo OMG [OMG 2004].

3. Reengenharia

Para Chikofsky e Cross (1990), IEEE CS-TCSE (2004) e GT-REG (2004), a reengenharia, conhecida também como renovação ou reconstrução, é o exame de um sistema de software, a fim de reconstituí-lo em uma nova forma, e a subsequente implementação dessa nova forma. Um processo de reengenharia geralmente inclui alguma forma de engenharia reversa, seguida por uma forma de engenharia progressiva ou reestruturação [Jacobson e Lindstrom 1991].

3.1 Engenharia reversa

Segundo Benedusi *et al.* (1992) e Chikofsky e Cross (1990), pode-se definir engenharia reversa como uma coleção de teorias, metodologias e técnicas capazes de suportar a extração e abstração de informações de um software existente, produzindo documentos consistentes, quer seja a partir do código fonte, ou através da adição de conhecimento e experiência que não podem ser automaticamente reconstruídos a partir do código.

A engenharia reversa de um sistema legado pode ser realizada de diversas maneiras, por exemplo, através dos códigos fontes ou executáveis, através do repositório de dados da ferramenta CASE utilizada ou através dos dicionários dos bancos de dados utilizados. Assim, tenta-se recuperar a semântica da solução presente nos diferentes metadados existentes no sistema. A representação obtida deste processo de engenharia reversa, geralmente segue padrões de mercado, tais como UML, MOF e XMI, de forma a ser compreendida facilmente pelo maior número possível de ferramentas. Um estudo de caso foi o projeto de Vicent Englebert [Hainaut *et al.* 1997].

Neste enfoque, surge também o aprendizado de ontologia (*Ontology Learning*) como um conjunto de métodos e técnicas usado para construir uma ontologia através de processos automáticos ou semi-automáticos de aquisição de conhecimento via textos, dicionários, bases de conhecimento, dados semi-estruturados e esquemas relacionais já existentes [Meersman *et al.* 2002], [Gómez-Pérez e Manzano-Macho 2003].

Algumas ferramentas com essa finalidade foram desenvolvidas no escopo de projetos maiores, tais como REVERSE, que é ferramenta para extração de ontologias do projeto KAON, e XRA, que é uma ferramenta de extração de ontologias a partir de engenharia reversa do projeto DOME. Os trabalhos de Handschuh *et al.* (2001) e Gómez-Pérez e Manzano-Macho (2003) demonstram a real viabilidade e oportunidade de aplicação do conceito de reengenharia em ontologias.

Foram desenvolvidas algumas ferramentas baseadas em técnicas de linguagem natural e aprendizagem por computador para a extração de ontologias a partir de legados:

- i. Ferramentas baseadas em Análise Léxica e Sintática (*Lexical and Syntactic Analysis*), tais como: LTG [Mikheev e Finch 1997] e Terminate [Biébow *et al.* 1999];

- ii. Ferramentas baseadas em Classificação Consensual (*Conceptual clustering*), tais como: ASIUM [Faure e Nédellec 1999], Mo'K [Bisson et al. 2000] e SVETLAN [Chaelandar e Grau 2000];
- iii. Ferramentas baseadas em Abordagem Estatística (*Statiscal Approach*), tais como: Text-To-Onto [Maedche e Staab 2001].

3.2. Metodologias para engenharia progressiva de ontologias

Segundo Marietto *et. al* (2002), o desenvolvimento e o uso de metodologias para a construção de ontologias é fundamental, na medida que retiram o caráter subjetivo desta atividade. A área de Engenharia Ontológica estuda os aspectos relacionados a tal construção, bem como o desenvolvimento de sistemas que utilizem ontologias em sua estrutura. Para López (2001), o que diferencia uma área de estudo que se encontra na “infância” de outra, em fase “adulta”, é esta tem suas metodologias aceitas e utilizadas. A Engenharia de Software pode ser dita como estando na fase adulta, pois suas metodologias são largamente difundidas e utilizadas.

3.2.1 Metodologia do Projeto Enterprise Ontology

Nesta metodologia, a construção da ontologia está dividida em três fases [Uschold e King 1995]:

- i. Captura da ontologia: identificação de conceitos chave e relacionamentos;
- ii. Codificação: representação das informações obtidas na captura da ontologia, através de uma linguagem formal;
- iii. Integração a ontologias existentes: integração da ontologia obtida a ontologias anteriormente criadas.

Os componentes da ontologia são criados na fase de captura da ontologia, com a identificação e a definição dos conceitos-chave e dos relacionamentos do domínio de interesse.

3.2.2 Metodologia do Projeto TOVE

Esta metodologia envolve a construção de um modelo formal do conhecimento, porém este modelo não é construído diretamente [Grüninger e Fox 1994]. Ele é construído a partir das seguintes fases:

- i. Captura de cenários: nesta metodologia, as ontologias são motivadas por cenários específicos para a aplicação;
- ii. Formulação de questões de competência informais: com base nos cenários capturados, a ontologia deve representar as questões de terminologia;
- iii. Especificação da terminologia da ontologia a partir de uma linguagem formal: criar um conjunto de termos, a partir das questões de competências informais, para especificar uma terminologia formal;
- iv. Formulação de questões de competência formais, utilizando a terminologia formal: utilizando as questões de competência informais e a terminologia formal, obter as questões de competência formal;
- v. Especificação de axiomas e definições para os termos da ontologia com a linguagem formal: os axiomas da ontologia especificam as definições dos termos e as suas interpretações;
- vi. Estabelecer condições para caracterizar a completude da ontologia: definir soluções para que as questões de competência fiquem completas.

Os componentes da ontologia são criados, em sua maior parte, na fase de especificação da terminologia da ontologia e especificação de axiomas e definições.

3.2.3 Metodologia do Projeto KACTUS

Esta metodologia esta inserida no projeto KACTUS. Um dos objetivos deste projeto é investigar quão prático é o reuso do conhecimento em sistemas complexos e quais são as regras ontológicas para suportar isso [Bernaras *et. al* 1996] e está condicionada à aplicação. Desta maneira, a aplicação e a ontologia são desenvolvidas ao mesmo tempo. As fases para construção da ontologia são:

- i. Especificação da aplicação: definir uma lista de termos e tarefas;
- ii. Desenvolvimento preliminar baseado nas categorias ontológicas de relevância: utilizar a lista de termos e tarefas como entrada, a fim de obter visões globais do modelo;

- iii. Refinamento e estruturação da ontologia: definição de uma ontologia definitiva; os módulos não devem ser muito dependentes uns dos outros e devem ser o mais coerentes possíveis, a fim de se obter a máxima homogeneidade de cada módulo.

Os componentes da ontologia são criados na fase de especificação da aplicação, pois nesta fase define-se a lista de termos e tarefas.

3.2.4 Methontology

Esta metodologia foi desenvolvida pelo grupo de Inteligência Artificial da Universidade Politécnica de Madri [Gómez-Pérez 2002], [López *et. al* 1997]. Esta metodologia está dividida em três grupos de atividades:

- i. Atividades de Gerenciamento de Projeto: incluindo planejamento, controle e garantia de qualidade. No planejamento são definidas quais tarefas serão desenvolvidas; no controle é garantido que as tarefas planejadas serão executadas e a garantia de qualidade busca desenvolver o produto como estabelecido pelos seus requisitos;
- ii. Atividades de Desenvolvimento Orientado: inclui especificação, conceitualização, formalização e implementação. A especificação define o porquê da ontologia estar sendo construída (utilização e usuários finais). A conceitualização estrutura o domínio de conhecimento em um modelo conceitual. A formalização transforma o modelo conceitual em um modelo formal ou semi-computável. A implementação transforma modelos computáveis em linguagens computacionais.
- iii. Atividades de Suporte: inclui aquisição de conhecimento, avaliação, integração, documentação e gerenciamento de configurações. Na aquisição de conhecimento é estudado o conhecimento de um determinado domínio. Na avaliação são realizadas comparações técnicas das ontologias, relacionadas com software e documentação. A integração de ontologias busca desenvolver uma nova ontologia a partir de outras já disponíveis. A documentação é a explicação completa e exhaustiva de todos os recursos e fases existentes. Gerenciamento de configurações registra todas as alterações (versões) de documentação, software e ontologias.

A criação dos componentes da ontologia é realizada, principalmente, na etapa de atividades de desenvolvimento orientado, especificamente, na conceitualização e formalização do conhecimento.

4.0 Ontologia e reengenharia

Este trabalho propõe as seguintes etapas para a obtenção de uma ontologia:

- i. Engenharia reversa (obtenção de um metadado da ontologia): através da aplicação da reengenharia nos sistemas legados obtém-se uma ontologia inicial;
- ii. Representação em UML estendida para ontologias, segundo Dutra (2001) e OMG (2004);
- iii. Engenharia progressiva (representação em RDF): aplicar uma das metodologias para o desenvolvimento de uma ontologia completa.

O conhecimento do sistema legado (Figura 1-1a) está geralmente presente nos metadados das ferramentas utilizadas na sua concepção e implementação (Figura 1-1b). Por exemplo, no dicionário de dados do gerenciador de banco de dados relacional utilizado (Figura 1-1c), no repositório da ferramenta CASE utilizada (Figura 1-1b) na sua construção ou nos documentos referentes ao sistema (help, planilhas, textos, etc) (Figura 1-1d). Aplicando a engenharia reversa (Figura 1-1e) sobre o dicionário de dados dos gerenciadores de banco de dados poder-se-ia obter o modelo Entidade-Relacionamento com suas entidades, atributos, relacionamentos, restrições e cardinalidade. Da engenharia reversa do repositório de dados da ferramenta CASE utilizada seriam obtidos os modelos (existentes) do sistema, tais como: Diagramas de Sequência, Diagramas de Estados e o Diagrama de Casos de Uso. Aplicando uma das técnicas de aprendizado de ontologias (Figura 1-1f), descritas na seção anterior, sobre os documentos existentes obtém-se uma taxonomia do sistema (Figura 1-1g).

Este processo de extração pode ser realizado por ferramentas implementadas com o padrão MOF (Figura 1-2a). Assim, o resultado final poderá ser apresentado no padrão UML. É a transformação de uma forma de representação em outra, de mesmo nível de abstração relativo, preservando o comportamento externo do sistema (funcionalidade semântica). Isso resulta em uma representação que preservaria as características do sistema, de forma compreensível pelos engenheiros de ontologias. Assim, poderia ser obtida a primeira representação conceitual da

ontologia - o primeiro modelo da ontologia no formato UML (Figura 1-2b). Os resultados obtidos nos processos Figura 1-1g e Figura 1-2a geram o metadados.

Em um segundo momento, aplica-se a Engenharia Progressiva (Figura 1-3), a partir da primeira versão proposta (Figura 1-2b), os engenheiros de ontologias seguem uma das metodologias descritas na seção 3, garantindo a utilização do padrão RDF. A utilização de metodologias garantem a base ontológica (Figura 1-4) será consistente e coerente com o domínio a ser representado.

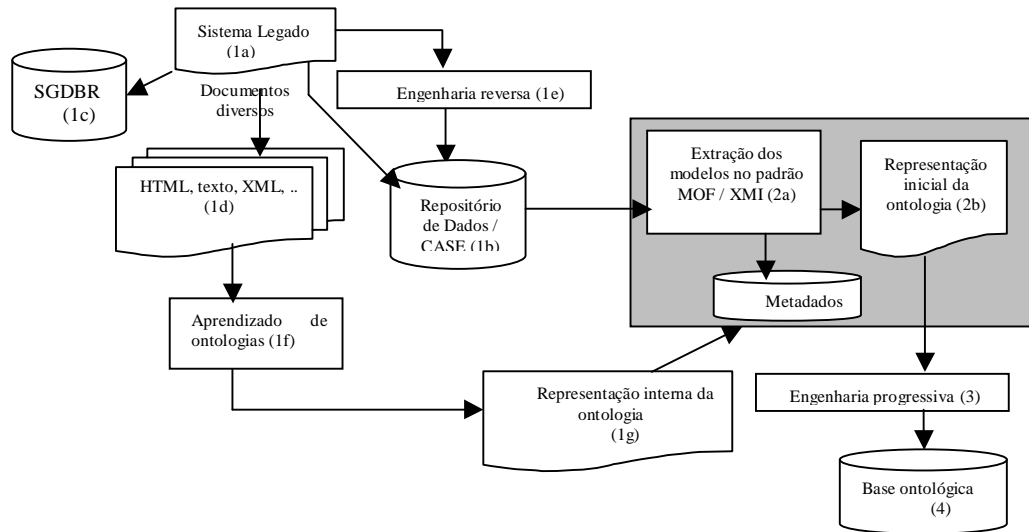


Figura 1. Combinação da reengenharia de sistemas e engenharia de ontologias.

5. Conclusões e Trabalhos Futuros

Este artigo apresentou a viabilidade conceitual de integração dos diversos padrões e técnicas associados aos temas de reengenharia de sistemas e engenharia de ontologias. Tanto os padrões da OMG –(UML, MOF, XMI), quanto os padrões da W3C –(XML e RDF), são reconhecidos, aceitos e utilizados pela comunidade de tecnologia da informação. A engenharia de ontologias, uma área que está em fase inicial de pesquisa, pode se valer desta experiência e maturidade para obter melhores representações de seus artefatos. Obviamente, os especialistas de um dado domínio são as pessoas mais indicadas para definir os conceitos-chave de uma determinada área de conhecimento. Mas, mesmo assim, deve-se considerar que a semântica de um termo pode variar de um contexto para outro, de um lugar para outro e mesmo de uma pessoa para outra. Por causa desta heterogeneidade semântica, a engenharia de ontologias não é uma atividade trivial e requer tempo, disponibilidade e consenso dos especialistas. Logo, a idéia do aproveitamento das fontes existentes, através da reengenharia ou do aprendizado de ontologias, como já realizado por Hainaut *et al.* (1997), Handschuh *et al.* (2001) e Gómez-Pérez e Manzano-Macho (2003), e proposto neste artigo, pode auxiliar muito na construção de um modelo inicial do que já foi representado no passado sobre um determinado domínio específico. Para que possa existir compartilhamento de conhecimento, é necessário que pelo menos os conceitos mais comuns estejam descritos em uma ontologia básica, que possa ser o ponto de convergência dos engenheiros ontológicos.

Como trabalhos futuros pretendemos realizar experimentos práticos conciliando a proposta apresentada anteriormente com o trabalho desenvolvido por Araujo (2003), onde foram realizados manualmente todos os processos anteriormente descritos em nossa proposta (reengenharia, representação em UML e engenharia progressiva utilizando Methontology para construção da ontologia). Desta forma, as etapas 1 e 2, apresentadas na Figura 1, serão feitas de forma automática e poder-se-á confrontar os resultados obtidos com a representação realizada por Araújo (2003) a fim de verificar consistência e completeza da representação obtida.

6. Referências Bibliográficas

- Araujo, M. (2003) "Educação à distância e a Web Semântica: modelagem ontológica de materiais e objetos de aprendizagem para a plataforma COL". São Paulo. 178p. Tese (Doutorado) – Escola Politécnica de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.
- 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". In: Fourth International Conference on UML (UML 2001), Toronto, Canada, October 2-5 (LNCS 2185).
- Baclawski, K.; Kokar, M.; Kogut, P.; Hart, L.; Smith, J.; Holmes, W.; Letkowski, J.; Aroston, M. (2002) "UOL: Unified Ontology Language". Assorted papers discussed at the DC Ontology SIG Meeting. Disponível em: <http://www.omg.org/cgi-bin/doc?ontology/2002-11-02>, 2002 (Set/2004).
- Benedusi, P.; Cimitile, A.; Carlini, U. (1992) "Reverse Engineering Processes, Design Document Production, and Structure Charts". Journal Systems and Software, v. 19, p. 225-245.
- Benjamins, R.; Gómez-Pérez, A. (1999) "Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem solving Methods". In: IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lesson learned and Future Trends, (KRR5), Estocolmo, 1999, Benjamins, V.R., (Ed.), CEUR Publications, vol.18, p. 1.1-1.15.
- Bernaras, A. ; Laresgoiti, I.; Corera, J. (1996) "Building and Reusing Ontologies for Electrical Network. Applications". In: (ECAI'96); 12th European Conference on Artificial Intelligence, Wiley, New York, p. 298-302.
- Biébow, B.; Szulman, S. (1999) "TERMINATE: a linguistic-based tool for the building of a domain ontology". In EKAW'99 – Proceedings of the 11th. European Workshop on Knowledge Acquisition, Modelling and Management. Dagstuhl, Germany, p. 49-66, Berlin. Springer-Verlag.
- Bisson, G.; Nédellec, C.; Cañamero, D. (2000) "Designing Clustering Methods for Ontology Building – The Mo'K Workbench". In Staab, S.; Maedche, A.; Nédellec, C.; Wiemer-Hasting, P. (Eds.). In: 14th. European Conference on Artificial Intelligence, ECAI'00, First Workshop on Ontology Learning, p. 13-18. Berlin, Germany.
- Chaui, M. (1995) "Convite à Filosofia". Editora Ática. São Paulo. 3a. Edição. p. 424.
- Chaelandar, G.; Grau, B. (2000) "SVETLAN - A System to Classify Words in Context". In: Staab, S.; Maedche, A.; Nédellec, C.; Wiemer-Hastings, P. (Eds.) Proceedings of the Workshop on Ontology Learning, 14th. European Conference on Artificial Intelligence ECAI'00, Berlin, Germany.
- CraneField, S.; Purvis, M. (1999) "UML as an Ontology Modeling Language" (1999). In: 16th. Int. Joint Conference on AI (IJCAI-99), Workshop on Intelligent Information Integration, Estocolmo, Suécia CEUR Publications, vol. 23.
- CraneField, S. (2001) "UML and the Semantic Web". In: International Semantic Web Working Symposium, Palo Alto. Disponível em: <http://www.semanticweb.org/SWWS/program/full/paper1.pdf> (Set/2004).
- Corcho, O.; López, M. Fernández; Gómez-Pérez, A. (2003) "Methodologies, tools and languages for building ontologies. Where is their meeting point?" Data & Knowledge Engineering. n. 46, p. 41-64.
- Chikofsky, E. J.; Cross II, J. H. (1990) "Reverse Engineering and Design Recovery: A Taxonomy". IEEE Software, v. 7, n. 1, 1990, p. 13-17.
- Dutra, M. (2001) "UML for Knowledge Representation". In: OMG's 2nd Workshop on UML for Enterprise Applications. Disponível em: http://www.omg.org/news/meetings/workshops/uml_2001.htm (Set/2004).
- Faure, D.; Nédellec, C. (1999) "Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system ASIUM". In: 11th European Workshop (EKAW'99), LNAI 1621, p. 329-334, Fensel, D. and Studer, D. (Eds.), Springer-Verlag.
- Gómez-Pérez, A.; Corcho, O. (2002) "Ontology Languages for the Semantic Web". IEEE Intelligent Systems, vol. 17, n. 1, p. 54-60.
- Gómez-Pérez, A.; Manzano-Macho, D. (2003) "A survey of ontology learning methods and techniques" Disponível em: <http://ontoweb.aifb.uni-karlsruhe.de/Members/ruben/Deliverable%201.5> (Out/ 2003).
- Gruber, T. R. (1993) "A translation approach to portable ontology specifications". Knowledge Acquisition, vol. 5, n. 2, p. 199-220.
- Gruber, T. R. (1995) "Toward principles for the design of ontologies used for knowledge sharing". International Journal of Human-Computer Studies, v. 43, p. 907-928.
- Grüniger, M.; Fox, M. S. (1994) "The Role of Competency Questions in Enterprise Engineering". In: Workshop on Benchmarking, Theory and Practice, Trondheim, Norway. Disponível em: <http://www.ie.utoronto.ca/EIL/public/competency.ps> (Set/2004).
- GT-REG (Georgia Tech's - Reverse Engineering Group) (2004) "Glossary of Reengineering Terms". Disponível em: <http://www.cc.gatech.edu/reverse/glossary.html> (Set/2004).
- Guarino, N. (1997) "Understanding, building, and using ontologies: a commentary to using explicit ontologies". International Journal of Human and Computer Studies, v. 46, p. 293-310.

- Guizzardi, G.; Herre, H.; Wagner, G. (2002) "Towards Ontological Foundations for UML Conceptual Models". In: First International Conference on Ontologies, Databases and Applications of Semantics (ODBASE 2002), LNCS 2519, p. 1118-1131.
- Hainaut, J.L., Englebert, V., Hick, J-M., Henrard, J., Roland, D. (1997) "Knowledge Transfer in Database Reverse Engineering - A Supporting Case Study". In: 4th. IEEE Working Conference on Reverse Engineering, Amsterdam, IEEE Computer Society Press, p. 194-203.
- Handschuh, S. et al. (2001) "KAON — The Karlsruhe Ontology and Semantic Web Infrastructure". Technical Report, Forschungszentrum Informatik Karlsruhe. Disponível em: <http://kaon.semanticweb.org/papers> (Out/2003).
- IEEE CS-TCSE (IEEE Computer Society - Technical Council on Software Engineering) (2004) "Reengineering & Reverse Engineering Terminology". Washington. Disponível em: <http://www.tcse.org/revengr/taxonomy.htm> (Set/2004).
- Jacobson, I.; Lindstrom, F. (1991) "Reengineering of Old Systems to an Object-oriented Architecture". SIGPLAN Notices, v. 26, n. 11, p. 340-350.
- López, M. F.; Gómez-Pérez, A.; Jurino, N. (1997) "Methontology: from Ontological Art towards Ontological Engineering". In: Symposium on Ontological Engineering of AAAI, Stanford, p. 33-40.
- López, M. F. (2001) "Overview of Methodologies for Building Ontologies". Intelligent Systems, vol.16, n. 1, p. 26-34.
- Maedche, A.; Staab, S. (2001) "Ontology Learning for the Semantic Web". IEEE Intelligent Systems, vol. 16, n. 2, p. 72-79.
- Maedche, A. et al. (2002) "SEAL – Tying up information integration and Web site management by ontologies". IEEE-CS Data Engineering Bulletin, vol. 25, n. 1, p. 10-17.
- Marietto, M. G. B.; David, N.; Sichman, J. S.; Coelho, H. (2002) "Infraestrutura para a Construção de Ontologias". Technical Report n. 01-2002. Escola Politécnica da Universidade de São Paulo.
- Meersman, R. et al. (2002) "Data Modelling versus Ontology Engineering". SIGMOD Records vol. 31, n. 4, p. 12-17.
- Mikheev, A.; Finch, S. A. (1997) "Workbench for Finding Structure in Texts". ANLP-97, p. 8.
- OMG: Object Management Group (2001) "Meta-object Facility (MOF)". Disponível em: <http://www.omg.org/cgi-bin/doc?formal/2002-04-03> (Jul/2004).
- OMG: Object Management Group (2002) "OMG XML Metadata Interchange (XMI)". Disponível em: <http://www.omg.org/cgi-bin/doc?formal/2002-01-01> (Jul/2004).
- OMG: Object Management Group (2003) "Unified Modeling Language (UML)". Disponível em: <http://www.omg.org/cgi-bin/doc?formal/03-03-01> (Jul/2004).
- OMG: Object Management Group (2004) "Ontology". Disponível <http://www.omg.org/ontology/> (Jul/2004).
- Su, X.; Ilebrikke, L. (2002) "A Comparative Study of Ontology Languages and Tools". In: 14th International Conference on Advanced Information Systems Engineering, p. 761-765, Springer-Verlag.
- Suzuki, J.; Yamamoto, Y. (1999) "Toward the interoperable software design models: quartet of UML, XML, DOM and CORBA". Software Engineering Standards, 1999. In: 4th. IEEE International Symposium and Forum, p. 163-172.
- Uschold, M.; King, M. (1995) "Towards a Methodology for Building Ontologies.". In: IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Canada. Disponível em: <ftp.aiiai.ed.ac.uk/pub/documents/1995/95-ont-ijcai95-ont-method.ps.gz> (Set/2004).