Problems and Challenges for Ontology Integration in the Semantic Web

Sergio Alejandro Gómez[†] **Carlos Iván Chesñevar**[‡] **Guillermo Ricardo Simari**[†]

[†]Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)[*]
Depto. Cs. e Ing. de la Computación – Universidad Nacional del Sur
Alem 1253 (8000) Bahía Blanca - ARGENTINA –
Tel/Fax: (+54) 291-459 5135/5136 – E-mail: {sag, grs}@cs.uns.edu.ar

[‡]Grupo de Investigación en Inteligencia Artificial
Departament d'Informàtica — Universitat de Lleida
Campus Cappont – C/Jaume II, 69 – E-25001 Lleida, SPAIN
Tel/Fax: (+34) (973) 70 2764 / 2702 – E-mail: cic@eps.udl.es

ABSTRACT

The *Semantic Web* is a future vision of the current web in which the resources have exact meaning. The meaning of resources is given by means of *ontology definitions*. When these ontologies are defined in isolation, the union of two or more ontologies can result in inconsistencies. Resolving such inconsistencies in order to put the ontologies into mutual agreement is known as *ontology integration*. In this paper, we briefly survey the languages for representing information in the web and the Semantic Web. We also review some methodologies for performing ontology integration. Part of our current research is focused into providing alternative representations of current standards for defining ontologies in order to overcome the problems associated with the traditional methods for ontology integration.

Keywords: World Wide Web, Semantic Web, Ontology languages, Ontology integration.

1 INTRODUCTION

The current World Wide Web (WWW) is based primarily on documents written for visual presentation for human users and not for being understood by computer programs. The *Semantic Web* [3] is a future vision of the web where information has exact meaning, thus enabling computers to understand and reason on information on the web.

The Semantic Web addresses the problem of assigning semantics to web resources by means of *ontology definitions*. In the context of knowledge sharing, the term ontology means a specification of a conceptualization. That is, an ontology is a description of the concepts and relationships that can exist for an agent or a community of agents [8]. Ontologies are described in an *ontology description language*, usually based on very-expressive Description Logics (DL) [2]. As ontologies are usually developed independently, their combination could result in incoherences and inconsistencies. The problem of combining two or more different ontologies in order to obtain a unified, consistent ontology is known as *ontology integration*.

In this research line, we are working on developing alternative representations for ontologies in order to solve the problem of integrating successfully two or more inconsistent and incoherent ontologies. With that goal in mind, in this paper we explore the issues concerning the representation of knowledge in the Semantic Web and in particular the problem of ontology integration.

The rest of the paper is structured as follows. Section 2 surveys languages for the representation of documents in the web and Semantic Web. Section 3 presents the general framework of the problem of ontology integration and surveys approaches found in the literature. Finally Section 4 concludes the paper.

2 KNOWLEDGE REPRESENTATION LANGUAGES IN THE WEB

In this section we briefly describe the languages for representing information in the Web and in the Semantic Web.

2.1 Hypertext Markup Language

HyperText Markup Language (HTML) [22] is a markup language designed for the creation of web pages to be displayed in a web browser. HTML is used to structure information (denoting certain text as headings, paragraphs, lists, etc.) and to describe

[*] LIDIA is a member of IICyTI Instituto de Investigación en Ciencia y Tecnología Informática.

the appearance of a document. *Cascading Style Sheets* (CSS) [12] is a stylesheet language used to describe the presentation of an HTML document but enabling the separation of document content (written in HTML) from document presentation (written in CSS). This separation improves content accessibility, providing more flexibility and control in the specification of presentational characteristics, thus reducing complexity and repetition in the structural content.

Besides publishing content in the form of web documents, HTML can be used for building front-ends for web-based applications. In such systems, user input is collected through *HTML forms*. Input is performed by using form elements (*e.g.*, text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.). For extending simple web form capabilities, Javascript scripts can be used to add validation and interactivity without increasing server-side overhead.

2.2 Extensible Markup Language

The *Extensible Markup Language* (XML) [14] is a general-purpose markup language for creating special-purpose markup languages, capable of describing many different kinds of data. Its primary purpose is to facilitate the sharing of data across different systems, particularly systems connected via the Internet. Those systems must agree the common format of the XML documents they interchange.

The purpose of a *Document Type Definition* (DTD) is to define the legal building blocks of an XML document. It defines the document structure with a list of legal elements. DTDs are not defined as XML documents but in a different language. Then *XML Schema* was developed as an XML-based alternative to DTDs.

The *EXtensible HyperText Markup Language* (XHTML) [19] is a stricter and cleaner version of HTML which is aimed to replace HTML. Basically, XHTML is HTML redefined as an XML application. In the framework of XHTML, web forms have also being redefined as an XML application—*XForms* uses XML for data definition and HTML or XHTML for data display. XForms separates the data logic of a form from its presentation. In this way the XForms data can be defined independent of how the end-user will interact with the application.

2.3 Resource Description Framework

The *Resource Description Framework* (RDF) [13] is standard for describing resources on the Web, such as the title, author, modification date, content, and copyright information of a Web page. RDF was designed to provide a common way to describe information so it can be read and understood by computer applications. RDF descriptions are not designed to be displayed on the web. RDF documents are written in an XML language called RDF/XML. By using XML, RDF information can easily be exchanged among different types of computers using different types of operating systems and application languages. While XML provides syntactic support for RDF, graph theory provides semantic support for RDF.

The base element of the *RDF model* is the *triple*: a resource (the *subject*) is linked to another resource (the *object*) through an arc labeled with a third resource (the *predicate*) [5]. We will say that "subject" has a property "predicate" valued by "object". For example, the triple ⟨*http://cs.uns.edu.ar/˜sag/index.htm, http://purl.org/DC/Creator, mailto:sag@cs.uns.edu.ar*⟩ could be read as "Gomez is the creator of index.htm". All the triples result in a directed graph, whose nodes and arcs are all labeled with qualified URIs. This graph describes resources with classes, properties, and values.

In addition, RDF also needs a way to define application-specific classes and properties. Application-specific classes and properties must be defined using extensions such as *RDF Schema* (RDFS). RDFS does not provide actual application-specific classes and properties. Instead RDFS provides the framework to describe application-specific classes and properties. Classes in RDFS are much like classes in object oriented programming languages; in particular, RDFS allows resources to be defined as instances of classes, and subclasses of classes.

2.4 DARPA Agent Markup Language

RDF and provide a basic feature set for information modeling. This simplicity makes it a sort of assembly language on top of which almost every other information-modeling method can be overlaid. However, in response to the need of data types, a consistent expression for enumerations, and other facilities, the *DARPA Agent Markup Language* (DAML) was released as *DAML-ONT* [15], a simple language for expressing more sophisticated RDF class definitions than those permitted by RDFS. The DAML group joined efforts with the *Ontology Inference Layer* (OIL) [7], a project that aimed at providing more sophisticated classification, using constructs from frame-based AI. The result of these efforts is *DAML+OIL* [6] which also adds facilities for data typing based on the type definitions provided in the *XML Schema Definition Language* (XSDL).

DAML+OIL model is based on very expressive *Description Logics*. Description Logics (DL) [2] are a well-known family of knowledge representation formalisms. They are based on the notions of *concepts* (unary predicates, classes) and *roles* (binary relations), and are mainly characterized by constructors that allow complex concepts and roles to be built from atomic ones. The expressive power of a DL system is determined by the constructs available for building concept descriptions, and by the way these descriptions can be used in the terminological (Tbox) and assertional (Abox) components of the system.

2.5 Ontology Web Language

The *Ontology Web Language* (OWL) [16] is built on top of RDF. OWL was designed to be interpreted by computers and not for being read by people. OWL and RDF are much alike, but OWL is a stronger language with greater machine interpretability than RDF (for instance, OWL comes with a larger vocabulary and stronger syntax than RDF). OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full. As in the case of RDF, by using XML, OWL information can easily be exchanged among different types of computers using different types of operating system and application languages.

In addition, OWL makes an open world assumption. That is, descriptions of resources are not confined to a single file or scope. While class C_1 may be defined originally in ontology O_1, it can be extended in other ontologies. The consequences of these additional propositions about C_1 are monotonic. New information cannot retract previous information. New information can be contradictory, but facts and entailments can only be added, never deleted. The possibility of such contradictions is something the designer of an ontology needs to take into consideration. The W3C Recommendation expects that tool support will help detect such cases [16].

3 ONTOLOGY INTEGRATION

Combining two or more ontologies into one single ontology is usually known as integration. However, the terminology regarding the ontology integration field is very disparate and sometimes contradictory as several authors propose different classifications of the terminology (*e.g.*, [10] and [20]). In this section, we briefly review the terminology associated with the subfield of ontology integration and also present a review of some methods for ontology integration.

Combining refers to using two or more different ontologies for a task in which their relation is important [10]; however, other authors see this notion as just *using* ontologies [20]; *i. e.*, the integration of ontologies into applications. *Merging/integration* is the process of creating a new ontology from two or more existing ontologies with overlapping parts [10]. Pinto *et. al.* [20] distinguish between *integration* of ontologies (when building a new ontology reusing other available ontologies) and *merging* of different ontologies about the same subject into a single one that unifies all of them.

Integrated ontologies could not be in agreement. Therefore, *aligning* is the process of bringing two or more ontologies into mutual agreement, making them consistent and coherent [10]. In order to do this a map must be built, thus *mapping* consists of relating similar concepts or relations from different sources to each other using an equivalence relation [10]. *Articulation* is the point of linkage between two aligned ontologies [10]. Articulation points can have the semantics equivalent, subsumes (`is-a`), property (`part-of` and/or `has-knowledge-of`) [18].

Changes to an ontology result in the production of another ontology. *Transforming* consists of changing the semantics of an ontology slightly to make it suitable for a purpose different than the original one. A *version* is the result of a change to an ontology. *Versioning* is a mechanism for keeping track between old and new evolved ontologies.

3.1 Hasse and Motik's Approach

To enable interoperability between applications in distributed information systems based on heterogeneous ontologies, it is necessary to formally define the notion of a mapping between ontologies. In [9], Haase and Motik define a mapping system for OWL-DL ontologies, where mappings are expressed as correspondences between conjunctive queries over ontologies. As query answering within such a general mapping system is undecidable, they identify a decidable fragment of the mapping system, which corresponds to OWL-DL extended with DL-safe rules. They also show how the mapping system can be applied for the task of ontology integration and present a query answering algorithm.

3.2 Pinto and Martins' Approach

In [21], Pinto and Martins describe the activities that compose the process of ontology integration and describe methodology to perform the ontology integration process. Their methodology is composed of the following steps: (1) identifying the integration possibility; (2) identifying modules involved; (3) identifying assumptions and ontological commitments; (4) identifying the knowledge to be represented in each module; (5) identifying candidate ontologies; (6) getting candidate ontologies; (7) studying and analyzing candidate ontologies; (8) choosing source ontologies;

(9) applying integration operations, and (10) analyzing the resulting ontology.

3.3 Wiesman *et. al.* Approach

In [23], Wiesman proposes a domain independent method for handling interoperability problems by learning a mapping between ontologies. The learning method is based on exchanging instances of concepts that are defined in the ontologies. The method starts with identifying pairs of instances of concepts denoting the same entity in the world using information retrieval techniques, followed by proposing and evaluating mappings between the ontologies using the pairs of instances. For each step of this method, the likelihood that a decision is correct is taken into account. Important benefits of the method are that (a) no domain knowledge is required, and (b) the structures of ontologies between which a mapping must be established play no role.

3.4 Li *et. al.* Approach

In [11], a novel agent-based ontology integration framework is developed for agents which consume ontologies in ontology-based applications as well as engage in tasks of ontology integration. The corresponding ontology integration mechanism is discussed. Derived ontologies can be reused in the system. A prototype is built by using the JADE agent platform for evaluation.

3.5 Alasoud *et. al.* Approach

In [1], Alasoud *et. al.* propose a framework for ontology integration which is a hybrid of materialized (data warehouse) and virtual views. They have developed a prototype of the proposed framework. The authors claim that, while much work is still ahead, their experiments so far indicate that the ideas used in this work are promising which may result in significant theoretical as well as practical contributions.

3.6 Mitra *et. al.* Approach

In [17], Mitra presents an *Ontology-Composition Algebra* that consists of a set of basic operators that can be used to manipulate ontologies. The algebraic operators can be used to declaratively specify how to compose new, derived ontologies from the existing source ontologies. A declarative specification allows easy replay of the composition task when source ontologies change and the change needs to be propagated to the derived ontologies. If there does not exist a means to quickly and easily update the derived ontologies when the source ontologies change, the derived ontologies supply stale and often inconsistent information to its clients. Before multiple ontologies can be composed, the semantic heterogeneity among these ontologies must be resolved and a set of articulation rules established that specify the correlation among related concepts across source ontologies. Mitra has decoupled the algebraic machinery that is used to manipulate ontologies from the component that derives the semantic correspondence among ontologies to create two distinct components: (1) *articulation-rule generating functions* generate articulation rules among pairs of ontologies, and (2) *algebraic operators* use the articulation rules to compose the source ontologies. Articulation-rule generation functions can be implemented as semi-automatic subroutines that deploy heuristic algorithms to articulate ontologies. Empirical evidence shows that semi-automatic articulation generating functions can be implemented and form an useful component of information composition tools. The Ontology-Composition Algebra has unary and binary operations that enable an ontology composer to select interesting portions of ontologies and compose them. The properties of the algebraic operators are characterized and the necessary and sufficient conditions for the optimization of composition tasks have are identified. Most of these properties depend upon properties of the articulation generation function employed to resolve the semantic heterogeneity among the ontologies being composed.

3.7 Calvanese *et. al.* Approach

Some authors consider the integration of ontologies with respect to a central ontology. The web is regarded as constituted by a variety of information sources, each expressed over a certain ontology. In order to extract information from such sources, their semantic integration and reconciliation in terms of a global ontology is required. In [4], Calvanese *et. al.* address the fundamental problem of how to specify the mapping between the global ontology and the local ontologies. They argue that for capturing such mapping in an appropriate way, the notion of query is a crucial one, since it is very likely that a concept in one ontology corresponds to a view (*i. e.*, a query) over the other ontologies. As a result query processing in ontology integration systems is strongly related to view-based query answering in data integration.

4 CONCLUSION AND WORK IN PROGRESS

The Semantic Web is a promising research topic that will allow the construction of intelligent applications capable of understanding the contents on the web. The power of such applications will rely on metadata expressed as ontologies. However as ontologies are usually developed in isolation, they can be inconsistent and incoherent respect each other. The task of joining two or more ontologies into one consistent ontology is known as ontology integration.

Although many research has been done in the area of ontology integration, the field still remains open. In this paper, we have surveyed the languages for representation of information in the web and the Semantic Web. We have also reviewed the problems associated with the field of ontology integration. In this research line, we are working in alternative representations for ontologies in order to solve the problem of integrating successfully two or more inconsistent and incoherent ontologies.

ACKNOWLEDGMENTS

This research was funded by Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 No. 13.096, PICT 2003 No. 15.043, PAV 2004 076), by CONICET (Argentina), by projects TIC2003-00950 and TIN2004-07933-C03-03 (MCyT, Spain) and by Ramón y Cajal Program (MCyT, Spain).

References

[1] Alasoud, A., Haarslev, V., and Shiri, N. A Hybrid Approach for Ontology Integration. In *VLDB Workshop on Ontologies-based techniques for DataBases and Information Systems (ODBIS)* (Trondheim, Norway, 2005).

[2] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P., Eds. *The Description Logic Handbook – Theory, Implementation and Applications*. Cambridge University Press, 2003.

[3] Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. *Scientific American* (17 May 2001).

[4] Calvanese, D., Giacomo, G. D., and Lenzerini, M. A Framework for Ontology Integration. In *Proceedings of the 1st Semantic Web Working Symposium (SWWS 2001)* (2001), pp. 303–316.

[5] Champin, P.-A. RDF Tutorial. Tech. rep., University of Lyon, France, 2001.

[6] Connolly, D., van Harmelen, F., Horrocks, I., McGuinness, D. L., and Stein, L. A. DAML+OIL (March 2001) Reference Description, 2001. http://www.w3.org/TR/daml+oil-reference.

[7] Decker, S., Fensel, D., van Harmelen, F., Horrocks, I., Melnik, S., Klein, M., and Broekstra, J. Knowledge representation on the web. *Proc. of the 2000 Description Logic Workshop (DL 2000)* (2000), 89–97.

[8] Gruber, T. R. A translation approach to portable ontologies. *Knowledge Acquisition 5*, 2 (1993), 199–220.

[9] Haase, P., and Motik, B. A mapping system for the integration of owl-dl ontologies. In *IHIS 05: Proceedings of the first international workshop on Interoperability of heterogeneous information systems* (NOV 2005), A. Hahn, S. Abels, and L. Haak, Eds., ACM Press, pp. 9–16.

[10] Klein, M. Combining and relating ontologies: an analysis of problems and solutions. In *Workshop on Ontologies and Information Sharing, IJCAI'01* (Seattle, USA, Aug. 4–5, 2001), A. Gomez-Perez, M. Gruninger, H. Stuckenschmidt, and M. Uschold, Eds.

[11] Li, L., Wu, B., and Yang, Y. Agent-based Ontology Integration for Ontology-based Applications. In *Australasian Ontology Workshop (AOW 2005), Jointly held with the 18th Australian Joint Conference on Artificial Intelligence, Conference in Research and Practice in Information Technology (CRPIT) series* (2005), vol. 58, Australian Computer Society, pp. 53–59.

[12] Lie, H. W., and Bos, B. Cascading Style Sheets, level 1. W3C Recommendation 17 Dec 1996, 1996.

[13] Manola, F., and Miller, E. RDF Primer. W3C Recommendation 10 February 2004, 2004.

[14] McGrath, S. *XML by example. Building e-commerce applications*. Prentice Hall, 1998.

[15] McGuiness, D., Fikes, R., Stein, L. A., and Hendler, J. DAML-ONT: An Ontology Language for the Semantic Web. In *Spinning the Semantic Web*, D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, Eds. The MIT Press, 2003, pp. 65–93.

[16] McGuiness, D. L., and van Harmelen, F. OWL Web Ontology Language Overview, 2004. http://www.w3.org/TR/owl-features/.

[17] Mitra, P. *An Algebraic Framework for the Interoperation of Ontologies*. PhD thesis, Department of Electrical Engineering, 2004.

[18] Mitra, P., Wiederhold, G., and Kersten, M. A Graph-Oriented Model for Articulation of Ontology Interdependencies. *Lecture Notes in Computer Science 1777* (2000), 86+.

[19] Pemberton, S., Austin, D., Axelsson, J., Çelik, T., Dominiak, D., Elenbaas, H., Epperson, B., Ishikawa, M., Matsui, S., McCarron, S., Navarro, A., Peruvemba, S., Relyea, R., Schnitzenbaumer, S., and Stark, P. XHTML 1.0 The Extensible HyperText Markup Language (Second Edition), 2002.

[20] Pinto, H. S., Gómez-Pérez, A., and Martins, J. P. Some issues on ontology integration. In *Proceedings of the IJCAI99's Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends* (1999), pp. 7.1–7.12.

[21] Pinto, H. S., and Martins, J. P. A methodology for ontology integration. In *K-CAP '01: Proceedings of the 1st international conference on Knowledge capture* (New York, NY, USA, 2001), ACM Press, pp. 131–138.

[22] Raggett, D., Hors, A. L., and Jacobs, I. HTML 4.01 Specification. W3C Recommendation 24 December 1999, 199.

[23] Wiesman, F., and Roos, N. Domain independent learning of ontology mappings. In *AAMAS'04* (July 19–23, 2004).