Journal Subline

LNCS 4244

Journal on Data Semantics VII

Stefano Spaccapietra
Editor-in-Chief

Springer

Stefano Spaccapietra (Ed.)

Journal on Data Semantics VII

Springer

Volume Editor

Stefano Spaccapietra
Database Laboratory, EPFL
1015 Lausanne, Switzerland
E-mail: stefano.spaccapietra@epfl.ch

Library of Congress Control Number: 2006935770

CR Subject Classification (1998): H.2, H.3, I.2, H.4, C.2

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 1861-2032
ISBN-10 3-540-46329-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-46329-0 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11890591 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 4244

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

The LNCS Journal on Data Semantics

Computerized information handling has changed its focus from centralized data management systems to decentralized data exchange facilities. Modern distribution channels, such as high-speed Internet networks and wireless communication infrastructure, provide reliable technical support for data distribution and data access, materializing the new, popular idea that data may be available to anybody, anywhere, anytime. However, providing huge amounts of data on request often turns into a counterproductive service, making the data useless because of poor relevance or inappropriate level of detail. Semantic knowledge is the essential missing piece that allows the delivery of information that matches user requirements. Semantic agreement, in particular, is essential to meaningful data exchange.

Semantic issues have long been open issues in data and knowledge management. However, the boom in semantically poor technologies, such as the Web and XML, has boosted renewed interest in semantics. Conferences on the Semantic Web, for instance, attract crowds of participants, while ontologies on their own have become a hot and popular topic in the database and artificial intelligence communities.

Springer's LNCS *Journal on Data Semantics* aims at providing a highly visible dissemination channel for most remarkable work that in one way or another addresses research and development on issues related to the semantics of data. The target domain ranges from theories supporting the formal definition of semantic content to innovative domain-specific application of semantic knowledge. This publication channel should be of the highest interest to researchers and advanced practitioners working on the Semantic Web, interoperability, mobile information services, data warehousing, knowledge representation and reasoning, conceptual database modeling, ontologies, and artificial intelligence.

Topics of relevance to this journal include:

- Semantic interoperability, semantic mediators
- Ontologies
- Ontology, schema and data integration, reconciliation and alignment
- Multiple representations, alternative representations
- Knowledge representation and reasoning
- Conceptualization and representation
- Multi-model and multi-paradigm approaches
- Mappings, transformations, reverse engineering
- Metadata
- Conceptual data modeling
- Integrity description and handling
- Evolution and change
- Web semantics and semi-structured data
- Semantic caching

- Data warehousing and semantic data mining
- Spatial, temporal, multimedia and multimodal semantics
- Semantics in data visualization
- Semantic services for mobile users
- Supporting tools
- Applications of semantic-driven approaches

These topics are to be understood as specifically related to semantic issues. Contributions submitted to the journal and dealing with semantics of data will be considered even if they are not within the topics in the list.

While the physical appearance of the journal issues looks like the books from the well-known Springer LNCS series, the mode of operation is that of a journal. Contributions can be freely submitted by authors and are reviewed by the Editorial Board. Contributions may also be invited, and nevertheless carefully reviewed, as in the case for issues that contain extended versions of best papers from major conferences addressing data semantics issues. Special issues, focusing on a specific topic, are coordinated by guest editors once the proposal for a special issue is accepted by the Editorial Board. Finally, it is also possible that a journal issue be devoted to a single text.

The journal published its first volume in 2003 (LNCS 2800). That initial volume, as well as volumes II (LNCS 3360), III (LNCS 3534), V (LNCS 3870), and coming volume VIII represent the annual occurrence of a special issue devoted to publication of selected extended versions of best conference papers from previous year conferences. Volumes III and VI are annual special issues on a dedicated topic. Volume III, coordinated by guest editor Esteban Zimányi, addressed Semantic-based Geographical Information Systems, while volume VI, coordinated by guest editors Karl Aberer and Philippe Cudre-Mauroux, addressed Emergent Semantics. The fourth volume was the first "normal" volume, built from spontaneous submissions on any of the topics of interest to the Journal. This volume VII is the second of this type.

The Editorial Board comprises one Editor-in-Chief (with overall responsibility), a co-editor-in-chief, and several members. The Editor-in-Chief has a four-year mandate. Members of the board have a three-year mandate. Mandates are renewable. New members may be elected anytime.

We are happy to welcome you to our readership and authorship, and hope we will share this privileged contact for a long time.

Stefano Spaccapietra
Editor-in-Chief
http://lbdwww.epfl.ch/e/Springer/

JoDS Volume VII – Preface

This JoDS volume results from a rigorous selection among 35 abstract/paper submissions received in response to a call for contributions issued July 2005.

After two rounds of reviews, nine papers, spanning a wide variety of topics, were eventually accepted for publication. They are listed in the table of contents herein.

We would like to thank authors of all submitted papers as well as all reviewers who contributed to improving the papers through their detailed comments.

Forthcoming volume VIII will contain extended versions of best papers from 2005 conferences covering semantics aspects. Its publication is expected towards the end of 2006.

We hope you'll enjoy reading this volume.

Stefano Spaccapietra
Editor-in-Chief

Reviewers

We are very grateful to the external reviewers listed below who helped the editorial board in the reviewing task:

Daniele Barone, Università Milano Bicocca, Italy
Stefano Borgo, Laboratory for Applied Ontology (ISTC-CNR), Italy
Federico Cabitza, Università Milano Bicocca, Italy
Flavio De Paoli, Università Milano Bicocca, Italy
Denise de Vries, Flinders University, Australia
Ying Ding, University of Innsbruck, Austria
Gillian Dobbie, University of Auckland, New Zealand
Fabrice Estiévenart, University of Namur, Belgium
Silvia Gabrielli, Università di Roma 1 "La Sapienza," Italy
Birte Glimm, The University of Manchester, UK
Giancarlo Guizzardi, Laboratory for Applied Ontology (ISTC-CNR), Italy
Benjamin Habegger, Università di Roma 1 "La Sapienza," Italy
Markus Kirchberg, Massey University, New Zealand
Jacek Kopecký, DERI Innsbruck, Austria
Changqing Li, National University of Singapore, Singapore
Sebastian Link, Massey University, New Zealand
Andrea Maurino, Università Milano Bicocca, Italy
Jean-Roch Meurisse, University of Namur, Belgium
Diego Milano, Università di Roma 1 "La Sapienza," Italy
Wei Nei, National University of Singapore, Singapore
Fabio Porto, EPFL, Switzerland
Saurav Sahay, Georgia Institute of Technology, USA
Christelle Vangenot, EPFL, Switzerland
Evgeny Zolin, The University of Manchester, UK

JoDS Editorial Board

Lecture Notes in Computer Science

For information about Vols. 1–4185

please contact your bookseller or Springer

Vol. 4272: P. Havinga, M. Lijding, N. Meratnia, M. Wegdam (Eds.), Smart Sensing and Context. XI, 267 pages. 2006.

Vol. 4270: H. Zha, Z. Pan, H. Thwaites, A.C. Addison, M. Forte (Eds.), Interactive Technologies and Sociotechnical Systems. XVI, 547 pages. 2006.

Vol. 4269: R. State, S. van der Meer, D. O'Sullivan, T. Pfeifer (Eds.), Large Scale Management of Distributed Systems. XIII, 282 pages. 2006.

Vol. 4268: G. Parr, D. Malone, M. Ó Foghlú (Eds.), Autonomic Principles of IP Operations and Management. XIII, 237 pages. 2006.

Vol. 4267: A. Helmy, B. Jennings, L. Murphy, T. Pfeifer (Eds.), Autonomic Management of Mobile Multimedia Services. XIII, 257 pages. 2006.

Vol. 4266: H. Yoshiura, K. Sakurai, K. Rannenberg, Y. Murayama, S. Kawamura (Eds.), Advances in Information and Computer Security. XIII, 438 pages. 2006.

Vol. 4265: N. Lavrač, L. Todorovski, K.P. Jantke (Eds.), Discovery Science. XIV, 384 pages. 2006. (Sublibrary LNAI).

Vol. 4264: J.L. Balcázar, P.M. Long, F. Stephan (Eds.), Algorithmic Learning Theory. XIII, 393 pages. 2006. (Sublibrary LNAI).

Vol. 4263: A. Levi, E. Savas, H. Yenigün, S. Balcisoy, Y. Saygin (Eds.), Computer and Information Sciences – ISCIS 2006. XXIII, 1084 pages. 2006.

Vol. 4260: Z. Liu, J. He (Eds.), Formal Methods and Software Engineering. XII, 778 pages. 2006.

Vol. 4257: I. Richardson, P. Runeson, R. Messnarz (Eds.), Software Process Improvement. XI, 219 pages. 2006.

Vol. 4256: L. Feng, G. Wang, C. Zeng, R. Huang (Eds.), Web Information Systems Engineering – WISE 2006 Workshops. XIV, 320 pages. 2006.

Vol. 4255: K. Aberer, Z. Peng, E.A. Rundensteiner, Y. Zhang, X. Li (Eds.), Web Information Systems Engineering – WISE 2006. XIV, 563 pages. 2006.

Vol. 4254: T. Grust, H. Höpfner, A. Illarramendi, S. Jablonski, M. Mesiti, S. Müller, P.-L. Patranjan, K.-U. Sattler, M. Spiliopoulou (Eds.), Current Trends in Database Technology – EDBT 2006. XXXI, 932 pages. 2006.

Vol. 4253: B. Gabrys, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. XXXII, 1301 pages. 2006. (Sublibrary LNAI).

Vol. 4252: B. Gabrys, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. XXXIII, 1335 pages. 2006. (Sublibrary LNAI).

Vol. 4251: B. Gabrys, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LXVI, 1297 pages. 2006. (Sublibrary LNAI).

Vol. 4249: L. Goubin, M. Matsui (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2006. XII, 462 pages. 2006.

Vol. 4248: S. Staab, V. Svátek (Eds.), Engineering Knowledge in the Age of the Semantic Web. XIV, 400 pages. 2006. (Sublibrary LNAI).

Vol. 4247: T.-D. Wang, X. Li, S.-H. Chen, X. Wang, H. Abbass, H. Iba, G. Chen, X. Yao (Eds.), Simulated Evolution and Learning. XXI, 940 pages. 2006.

Vol. 4245: A. Kuba, L.G. Nyúl, K. Palágyi (Eds.), Discrete Geometry for Computer Imagery. XIII, 688 pages. 2006.

Vol. 4244: S. Spaccapietra (Ed.), Journal on Data Semantics VII. XI, 267 pages. 2006.

Vol. 4243: T. Yakhno, E.J. Neuhold (Eds.), Advances in Information Systems. XIII, 420 pages. 2006.

Vol. 4241: R.R. Beichel, M. Sonka (Eds.), Computer Vision Approaches to Medical Image Analysis. XI, 262 pages. 2006.

Vol. 4239: H.Y. Youn, M. Kim, H. Morikawa (Eds.), Ubiquitous Computing Systems. XVI, 548 pages. 2006.

Vol. 4238: Y.-T. Kim, M. Takano (Eds.), Management of Convergence Networks and Services. XVIII, 605 pages. 2006.

Vol. 4237: H. Leitold, E. Markatos (Eds.), Communications and Multimedia Security. XII, 253 pages. 2006.

Vol. 4236: L. Breveglieri, I. Koren, D. Naccache, J.-P. Seifert (Eds.), Fault Diagnosis and Tolerance in Cryptography. XIII, 253 pages. 2006.

Vol. 4234: I. King, J. Wang, L. Chan, D. Wang (Eds.), Neural Information Processing, Part III. XXII, 1227 pages. 2006.

Vol. 4233: I. King, J. Wang, L. Chan, D. Wang (Eds.), Neural Information Processing, Part II. XXII, 1203 pages. 2006.

Vol. 4232: I. King, J. Wang, L. Chan, D. Wang (Eds.), Neural Information Processing, Part I. XLVI, 1153 pages. 2006.

Vol. 4229: E. Najm, J.F. Pradat-Peyre, V.V. Donzeau-Gouge (Eds.), Formal Techniques for Networked and Distributed Systems - FORTE 2006. X, 486 pages. 2006.

Vol. 4228: D.E. Lightfoot, C.A. Szyperski (Eds.), Modular Programming Languages. X, 415 pages. 2006.

Vol. 4227: W. Nejdl, K. Tochtermann (Eds.), Innovative Approaches for Learning and Knowledge Sharing. XVII, 721 pages. 2006.

Vol. 4225: J.F. Martínez-Trinidad, J.A. Carrasco Ochoa, J. Kittler (Eds.), Progress in Pattern Recognition, Image Analysis and Applications. XIX, 995 pages. 2006.

Vol. 4224: E. Corchado, H. Yin, V. Botti, C. Fyfe (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2006. XXVII, 1447 pages. 2006.

Vol. 4223: L. Wang, L. Jiao, G. Shi, X. Li, J. Liu (Eds.), Fuzzy Systems and Knowledge Discovery. XXVIII, 1335 pages. 2006. (Sublibrary LNAI).

Vol. 4222: L. Jiao, L. Wang, X. Gao, J. Liu, F. Wu (Eds.), Advances in Natural Computation, Part II. XLII, 998 pages. 2006.

Vol. 4221: L. Jiao, L. Wang, X. Gao, J. Liu, F. Wu (Eds.), Advances in Natural Computation, Part I. XLI, 992 pages. 2006.

Vol. 4219: D. Zamboni, C. Kruegel (Eds.), Recent Advances in Intrusion Detection. XII, 331 pages. 2006.

Vol. 4218: S. Graf, W. Zhang (Eds.), Automated Technology for Verification and Analysis. XIV, 540 pages. 2006.

Vol. 4217: P. Cuenca, L. Orozco-Barbosa (Eds.), Personal Wireless Communications. XV, 532 pages. 2006.

Vol. 4216: M.R. Berthold, R. Glen, I. Fischer (Eds.), Computational Life Sciences II. XIII, 269 pages. 2006. (Sublibrary LNBI).

Vol. 4215: D.W. Embley, A. Olivé, S. Ram (Eds.), Conceptual Modeling - ER 2006. XVI, 590 pages. 2006.

Vol. 4213: J. Fürnkranz, T. Scheffer, M. Spiliopoulou (Eds.), Knowledge Discovery in Databases: PKDD 2006. XXII, 660 pages. 2006. (Sublibrary LNAI).

Vol. 4212: J. Fürnkranz, T. Scheffer, M. Spiliopoulou (Eds.), Machine Learning: ECML 2006. XXIII, 851 pages. 2006. (Sublibrary LNAI).

Vol. 4211: P. Vogt, Y. Sugita, E. Tuci, C. Nehaniv (Eds.), Symbol Grounding and Beyond. VIII, 237 pages. 2006. (Sublibrary LNAI).

Vol. 4210: C. Priami (Ed.), Computational Methods in Systems Biology. X, 323 pages. 2006. (Sublibrary LNBI).

Vol. 4209: F. Crestani, P. Ferragina, M. Sanderson (Eds.), String Processing and Information Retrieval. XIV, 367 pages. 2006.

Vol. 4208: M. Gerndt, D. Kranzlmüller (Eds.), High Performance Computing and Communications. XXII, 938 pages. 2006.

Vol. 4207: Z. Ésik (Ed.), Computer Science Logic. XII, 627 pages. 2006.

Vol. 4206: P. Dourish, A. Friday (Eds.), UbiComp 2006: Ubiquitous Computing. XIX, 526 pages. 2006.

Vol. 4205: G. Bourque, N. El-Mabrouk (Eds.), Comparative Genomics. X, 231 pages. 2006. (Sublibrary LNBI).

Vol. 4204: F. Benhamou (Ed.), Principles and Practice of Constraint Programming - CP 2006. XVIII, 774 pages. 2006.

Vol. 4203: F. Esposito, Z.W. Raś, D. Malerba, G. Semeraro (Eds.), Foundations of Intelligent Systems. XVIII, 767 pages. 2006. (Sublibrary LNAI).

Vol. 4202: E. Asarin, P. Bouyer (Eds.), Formal Modeling and Analysis of Timed Systems. XI, 369 pages. 2006.

Vol. 4201: Y. Sakakibara, S. Kobayashi, K. Sato, T. Nishino, E. Tomita (Eds.), Grammatical Inference: Algorithms and Applications. XII, 359 pages. 2006. (Sublibrary LNAI).

Vol. 4200: I.F.C. Smith (Ed.), Intelligent Computing in Engineering and Architecture. XIII, 692 pages. 2006. (Sublibrary LNAI).

Vol. 4199: O. Nierstrasz, J. Whittle, D. Harel, G. Reggio (Eds.), Model Driven Engineering Languages and Systems. XVI, 798 pages. 2006.

Vol. 4198: O. Nasraoui, O. Zaiane, M. Spiliopoulou, B. Mobasher, B. Masand, P. Yu (Eds.), Advances in Web Minding and Web Usage Analysis. IX, 177 pages. 2006. (Sublibrary LNAI).

Vol. 4197: M. Raubal, H.J. Miller, A.U. Frank, M.F. Goodchild (Eds.), Geographic, Information Science. XIII, 419 pages. 2006.

Vol. 4196: K. Fischer, I.J. Timm, E. André, N. Zhong (Eds.), Multiagent System Technologies. X, 185 pages. 2006. (Sublibrary LNAI).

Vol. 4195: D. Gaiti, G. Pujolle, E. Al-Shaer, K. Calvert, S. Dobson, G. Leduc, O. Martikainen (Eds.), Autonomic Networking. IX, 316 pages. 2006.

Vol. 4194: V.G. Ganzha, E.W. Mayr, E.V. Vorozhtsov (Eds.), Computer Algebra in Scientific Computing. XI, 313 pages. 2006.

Vol. 4193: T.P. Runarsson, H.-G. Beyer, E. Burke, J.J. Merelo-Guervós, L.D. Whitley, X. Yao (Eds.), Parallel Problem Solving from Nature - PPSN IX. XIX, 1061 pages. 2006.

Vol. 4192: B. Mohr, J.L. Träff, J. Worringen, J. Dongarra (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface. XVI, 414 pages. 2006.

Vol. 4191: R. Larsen, M. Nielsen, J. Sporring (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006, Part II. XXXVIII, 981 pages. 2006.

Vol. 4190: R. Larsen, M. Nielsen, J. Sporring (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006, Part I. XXXVVIII, 949 pages. 2006.

Vol. 4189: D. Gollmann, J. Meier, A. Sabelfeld (Eds.), Computer Security – ESORICS 2006. XI, 548 pages. 2006.

Vol. 4188: P. Sojka, I. Kopeček, K. Pala (Eds.), Text, Speech and Dialogue. XV, 721 pages. 2006. (Sublibrary LNAI).

Vol. 4187: J.J. Alferes, J. Bailey, W. May, U. Schwertel (Eds.), Principles and Practice of Semantic Web Reasoning. XI, 277 pages. 2006.

Vol. 4186: C. Jesshope, C. Egan (Eds.), Advances in Computer Systems Architecture. XIV, 605 pages. 2006.

¥385.00元

Table of Contents

Discovering the Semantics of Relational Tables Through Mappings *

Yuan An[1], Alex Borgida[2], and John Mylopoulos[1]

[1] Department of Computer Science, University of Toronto, Canada
{yuana, jm}@cs.toronto.edu
[2] Department of Computer Science, Rutgers University, USA
borgida@cs.rutgers.edu

Abstract. Many problems in Information and Data Management require a semantic account of a database schema. At its best, such an account consists of formulas expressing the relationship ("mapping") between the schema and a formal conceptual model or ontology (CM) of the domain. In this paper we describe the underlying principles, algorithms, and a prototype tool that finds such semantic mappings from relational tables to ontologies, when given as input *simple correspondences* from columns of the tables to datatype properties of classes in an ontology. Although the algorithm presented is necessarily heuristic, we offer formal results showing that the answers returned by the tool are "correct" for relational schemas designed according to standard Entity-Relationship techniques. To evaluate its usefulness and effectiveness, we have applied the tool to a number of public domain schemas and ontologies. Our experience shows that significant effort is saved when using it to build semantic mappings from relational tables to ontologies.

Keywords: Semantics, ontologies, mappings, semantic interoperability.

1 Introduction and Motivation

A number of important database problems have been shown to have improved solutions by using a conceptual model or an ontology (CM) to provide *precise semantics* for a database schema. These[1] include federated databases, data warehousing [2], and information integration through mediated schemas [13,8]. Since much information on the web is generated from databases (the "deep web"), the recent call for a Semantic Web, which requires a connection between web content and ontologies, provides additional motivation for the problem of associating semantics with database-resident data (e.g., [10]). In almost all of these cases, semantics of the data is captured by some kind of *semantic mapping* between the database schema and the CM. Although sometimes the mapping is just a *simple* association from terms to terms, in other cases what is required is a *complex* formula, often expressed in logic or a query language [14].

For example, in both the Information Manifold data integration system presented in [13] and the DWQ data warehousing system [2], formulas of the form $T(\overline{X}) :- \Phi(\overline{X}, \overline{Y})$

* This is an expanded and refined version of a research paper presented at ODBASE'05 [1].
[1] For a survey, see [23].

S. Spaccapietra (Ed.): Journal on Data Semantics VII, LNCS 4244, pp. 1–32, 2006.

are used to connect a relational data source to a CM expressed in terms of a Description Logic, where $T(\overline{X})$ is a single predicate representing a table in the relational data source, and $\Phi(\overline{X}, \overline{Y})$ is a conjunctive formula over the predicates representing the concepts and relationships in the CM. In the literature, such a formalism is called local-as-view (LAV), in contrast to global-as-view (GAV), where atomic ontology concepts and properties are specified by queries over the database [14].

In all previous work it has been assumed that *humans* specify the mapping formulas – a difficult, time-consuming and error-prone task, especially since the specifier must be familiar with both the semantics of the database schema and the contents of the ontology. As the size and complexity of ontologies increase, it becomes desirable to have some kind of computer tool to assist people in the task. Note that the problem of semantic mapping discovery is superficially similar to that of database schema mapping, however the goal of the later is finding queries/rules for integrating/translating/exchanging the underlying data. Mapping schemas to ontologies, on the other hand, is aimed at understanding the semantics of a schema expressed in terms of a given semantic model. This requires paying special attentions to various semantic constructs in both schema and ontology languages.

We have proposed in [1] a tool that assists users in discovering mapping formulas between relational database schemas and ontologies, and presented the algorithms and the formal results. In this paper, we provide, in addition to what appears in [1], more detailed examples for explaining the algorithms, and we also present proofs to the formal results. Moreover, we show how to handle GAV formulas that are often useful for many practical data integration systems. The heuristics that underlie the discovery process are based on a careful study of standard design process relating the constructs of the relational model with those of conceptual modeling languages. In order to improve the effectiveness of our tool, we assume some user input in addition to the database schema and the ontology. Specifically, inspired by the Clio project [17], we expect the tool user to provide *simple correspondences* between atomic elements used in the database schema (e.g., column names of tables) and those in the ontology (e.g., attribute/"data type property" names of concepts). Given the set of correspondences, the tool is expected to reason about the database schema and the ontology, and to generate a list of candidate formulas for each table in the relational database. Ideally, one of the formulas is the correct one — capturing user intention underlying given correspondences. The claim is that, compared to composing logical formulas representing semantic mappings, it is much easier for users to (i) draw simple correspondences/arrows from column names of tables in the database to datatype properties of classes in the ontology[2] and then (ii) evaluate proposed formulas returned by the tool. The following example illustrates the input/output behavior of the tool proposed.

Example 1.1. An ontology contains concepts (classes), attributes of concepts (datatype properties of classes), relationships between concepts (associations), and cardinality constraints on occurrences of the participating concepts in a relationship. Graphically, we use the UML notations to represent the above information. Figure 1 is an enterprise ontology containing some basic concepts and relationships. (Recall that cardinality

[2] In fact, there exist already tools used in schema matching which help perform such tasks using linguistic, structural, and statistical information (e.g., [4,21]).

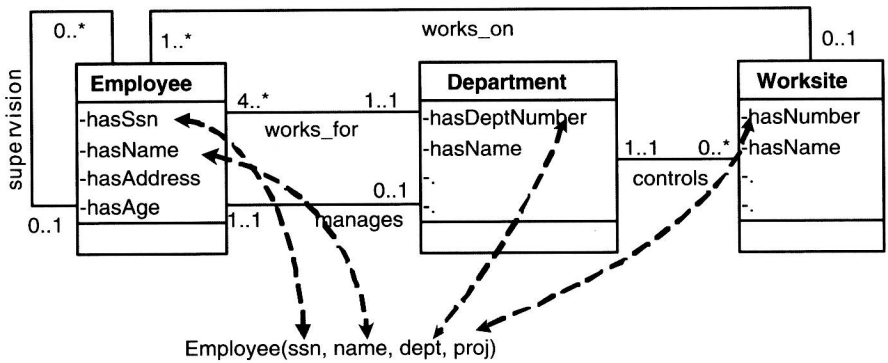

Fig. 1. Relational table, Ontology, and Correspondences

constraints in UML are written at the opposite end of the association: a Department has at least 4 Employees working for it, and an Employee works in one Department.) Suppose we wish to discover the semantics of a relational table $Employee(\underline{ssn}, name, dept, proj)$ with key ssn in terms of the enterprise ontology. Suppose that by looking at column names of the table and the ontology graph, the user draws the simple correspondences shown as dashed arrows in Figure 1. This indicates, for example, that the ssn column corresponds to the $hasSsn$ property of the $Employee$ concept. Using prefixes $\mathcal{T}$ and $\mathcal{O}$ to distinguish tables in the relational schema and concepts in the ontology (both of which will eventually be thought of as predicates), we represent the correspondences as follows:

$\mathcal{T} : Employee.ssn \rightsquigarrow \mathcal{O} : Employee.hasSsn$

$\mathcal{T} : Employee.name \rightsquigarrow \mathcal{O} : Employee.hasName$

$\mathcal{T} : Employee.dept \rightsquigarrow \mathcal{O} : Department.hasDeptNumber$

$\mathcal{T} : Employee.proj \rightsquigarrow \mathcal{O} : Worksite.hasNumber$

Given the above inputs, the tool is expected to produce a list of plausible mapping formulas, which would hopefully include the following formula, expressing a possible semantics for the table:

$\mathcal{T}$:Employee($ssn, name, dept, proj$) :-

$\mathcal{O}$:Employee(x_1), $\mathcal{O}$:hasSsn(x_1,ssn), $\mathcal{O}$:hasName(x_1,$name$), $\mathcal{O}$:Department(x_2),
$\mathcal{O}$:works_for(x_1,x_2), $\mathcal{O}$:hasDeptNumber(x_2,$dept$), $\mathcal{O}$:Worksite(x_3), $\mathcal{O}$:works_on(x_1,x_3),
$\mathcal{O}$:hasNumber(x_3,$proj$).

Note that, as explained in [14], the above, admittedly confusing notation in the literature, should really be interpreted as the First Order Logic formula

$(\forall ssn, name, dept, proj)$ $\mathcal{T}$:Employee($ssn, name, dept, proj$) $\Rightarrow$
$(\exists x_1, x_2, x_3)$ $\mathcal{O}$:Employee(x_1) $\wedge$...

because the ontology *explains* what is in the table (i.e., every tuple corresponds to an employee), rather than guaranteeing that the table satisfies the closed world assumption (i.e., for every employee there is a tuple in the table). ■

An intuitive (but somewhat naive) solution, inspired by early work of Quillian [20], is based on finding the *shortest* connections between concepts. Technically, this involves

(i) finding the minimum spanning tree(s) (actually Steiner trees[3]) connecting the "corresponded concepts" — those that have datatype properties corresponding to table columns, and then (ii) encoding the tree(s) into formulas. However, in some cases the spanning/Steiner tree may not provide the desired semantics for a table because of known relational schema design rules. For example, consider the relational table $Project$ ($\underline{name}$, $supervisor$), where the column $name$ is the key and corresponds to the attribute $\mathcal{O}$:$Worksite.hasName$, and column $supervisor$ corresponds to the attribute $\mathcal{O}$:$Employee.hasSsn$ in Figure 1. The minimum spanning tree consisting of $Worksite$, $Employee$, and the edge $works\_on$ probably does not match the semantics of table $Project$ because there are multiple $Employees$ working on a $Worksite$ according to the ontology cardinality, yet the table allows only one to be recorded, since $supervisor$ is functionally dependent on $name$, the key. Therefore we must seek a functional connection from $Worksite$ to $Employee$, and the connection will be the manager of the department controlling the worksite. In this paper, we use ideas of standard relational schema design from ER diagrams in order to craft heuristics that systematically uncover the connections between the constructs of relational schemas and those of ontologies. We propose a tool to generate "reasonable" trees connecting the set of corresponded concepts in an ontology. In contrast to the graph theoretic results which show that there may be too many minimum spanning/Steiner trees among the ontology nodes (for example, there are already 5 minimum spanning trees connecting $Employee$, $Department$, and $Worksite$ in the very simple graph in Figure 1), we expect the tool to generate only a small number of "reasonable" trees. These expectations are born out by our experimental results, in Section 6.

As mentioned earlier, our approach is directly inspired by the Clio project [17,18], which developed a successful tool that infers mappings from one set of relational tables and/or XML schemas to another, given just a set of correspondences between their respective attributes. Without going into further details at this point, we summarize the contributions of this work:

- We identify a new version of the data mapping problem: that of *inferring* complex formulas expressing the semantic mapping between relational database schemas and ontologies from simple correspondences.
- We propose an algorithm to find "reasonable" tree connection(s) in the ontology graph. The algorithm is enhanced to take into account information about the schema (key and foreign key structure), the ontology (cardinality restrictions), and standard database schema design guidelines.
- To gain theoretical confidence, we give formal results for a limited class of schemas. We show that if the schema was designed from a CM using techniques well-known in the Entity Relationship literature (which provide a natural semantic mapping and correspondences for each table), then the tool will recover essentially all and only the appropriate semantics. This shows that our heuristics are not just shots in the dark: in the case when the ontology has no extraneous material, and when a table's scheme has not been denormalized, the algorithm will produce good results.

[3] A Steiner tree for a set M of nodes in graph G is a minimum spanning tree of M that may contain nodes of G which are not in M.

- To test the effectiveness and usefulness of the algorithm in practice, we implemented the algorithm in a prototype tool and applied it to a variety of database schemas and ontologies drawn from a number of domains. We ensured that the schemas and the ontologies were developed independently; and the schemas might or might not be derived from a CM using the standard techniques. Our experience has shown that the user effort in specifying complex mappings by using the tool is significantly less than that by manually writing formulas from scratch.

The rest of the paper is structured as follows. We contrast our approach with related work in Section 2, and in Section 3 we present the technical background and notation. Section 4 describes an intuitive progression of ideas underlying our approach, while Section 5 provides the mapping inference algorithm. In Section 6 we report on the prototype implementation of these ideas and experiments with the prototype. Section 7 shows how to filter out unsatisfied mapping formulas by ontology reasoning. Section 8 discusses the issues of generating GAV mapping formulas. Finally, Section 9 concludes and discusses future work.

2 Related Work

The Clio tool [17,18] discovers formal queries describing how target schemas can be populated with data from source schemas. To compare with it, we could view the present work as extending Clio to the case when the source schema is a relational database while the target is an ontology. For example, in Example 1.1, if one viewed the ontology as a relational schema made of unary tables (such as $Employee(x_1)$), binary tables (such as $hasSsn(x_1, ssn)$) and the obvious foreign key constraints from binary to unary tables, then one could in fact try to apply directly the Clio algorithm to the problem. The desired mapping formula from Example 1.1 would not be produced for several reasons: (i) Clio [18] works by taking each table and using a chase-like algorithm to repeatedly extend it with columns that appear as foreign keys referencing other tables. Such "logical relations" in the source and target are then connected by queries. In this particular case, this would lead to logical relations such as $works\_for \bowtie Employee \bowtie Department$, but none that join, through some intermediary, $hasSsn(x_1, ssn)$ and $hasDeptNumber(x_2, dept)$, which is part of the desired formula in this case. (ii) The fact that ssn is a key in the table $\mathcal{T}$:$Employee$, leads us to prefer (see Section 4) a many-to-one relationship, such as $works\_for$, over some many-to-many relationship which could have been part of the ontology (e.g., $\mathcal{O}$:$previouslyWorkedFor$); Clio does not differentiate the two. So the work to be presented here analyzes the key structure of the tables and the semantics of relationships (cardinality, IsA) to eliminate/downgrade *unreasonable* options that arise in mappings to ontologies.

Other potentially relevant work includes *data reverse engineering*, which aims to extract a CM, such as an ER diagram, from a database schema. Sophisticated algorithms and approaches to this have appeared in the literature over the years (e.g., [15,9]). The major difference between data reverse engineering and our work is that we are given an existing ontology, and want to interpret a legacy relational schema in terms of it, whereas data reverse engineering aims to construct a new ontology.