On a Proposal to Integrate Web Sources using Semantic-Web Technologies

Hassan A. Sleiman
University of Sevilla, Spain
hassansleiman@us.es

Carlos R. Rivero
University of Sevilla, Spain
carlosrivero@us.es

Rafael Corchuelo
University of Sevilla, Spain
corchu@us.es

***Abstract*—Companies comprise a variety of software applications to carry out their business activities. A recurrent challenge is how to make them interoperate with each other which is usually handcrafted, which is a tedious task that increases integration costs. Enterprise Service Buses range amongst the most popular solution to reduce these costs, and they allow to implement integration solutions by means of one or more layers between software applications and business processes. In this paper, we present a framework for information extraction that allow to wrap information from different web sources and to generate linked data. Furthermore, we survey a number of approaches in the bibliography to build Enterprise Service Buses in the context of semantic-web technologies, which comprise RDF, RDFS, OWL, and SPARQL languages. Finally, we conclude that, thanks to linked data, we may integrate software applications with other applications that generate and/or consume these linked data.**

***Index Terms*—Semantic web services architectures; Semantic web services and linked data.**

I. Introduction

Nowadays, companies comprise a variety of software applications, which are called software ecosystem, to carry out their business activities [40]. One of the most important challenges is to make these software applications interoperate with each other to keep their data synchronised or to create new functionality [22], [27]. This interoperation is usually handcrafted, which is difficult to build and maintain, so integration costs are highly increased [49], [63].

With the motivation of reducing integration costs, companies have invested in solutions to facilitate the task of integrating software applications [5]. Currently, ESB (Enterprise Service Buses) range amongst one of the most popular solutions to reduce integration costs [13], [18]. ESBs allow to implement integration solutions by means of one or more layers between the software ecosystem and the business processes to reduce the coupling between applications and business processes (cf. Figure 1).

Our research focuses on ESBs that use Service and Data Virtualisation Layers to help reduce integration costs. In this context, the Service Layer exposes the applications in the software ecosystem as web services. Note that, to expose them, it is usually needed a number of wrappers to transform messages of the business processes into actions over applications, and vice versa. Furthermore, the Data Virtualisation Layer is responsible for offering a virtual, unified view over a number of web services, i.e., when a query is posed over the virtual view, this layer is responsible for answering the query using the underlying web services only [20], [36], [61].

Supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, TIN2008-04718-E, TIN2010-21744, TIN2010-09809-E, TIN2010-10811-E, and TIN2010-09988-E).

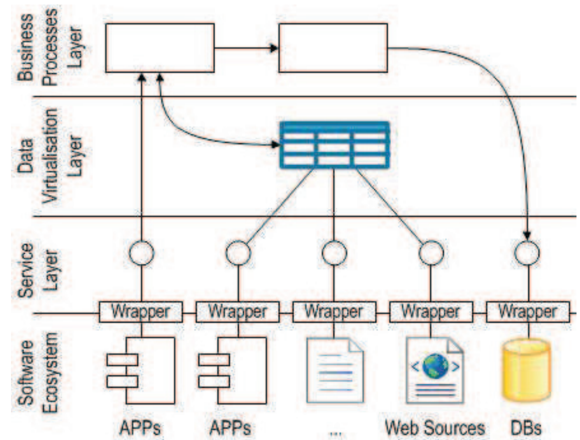

Fig. 1. Layers to integrate software applications

Our research also focuses on ESBs that are based on semantic-web technologies, which comprise RDF, RDF Schema and OWL ontology languages to model structure and data, and their data are queried by means of the SPARQL query language [3], [47]. In this context, the Service Layer provides a number of semantic web services, each of which represents its data by means of a semantic-web ontology that is described using semantic-web technologies. Furthermore, the Data Virtualisation Layer is responsible for integrating different semantic-web ontologies, which are called source ontologies, to offer their data as if they were a unified, single semantic-web ontology, which is called virtual. Information from web sources that only provide a human interface are first wrapped by means of information extractors or web wrappers that extract and structure this information.

In this paper, we focus on wrappers and the Data Virtualisation Layer of Figure 1. We survey briefly web wrapping techniques, also known as information extractors, we then describe our information extraction framework in which several wrapping techniques can be built to wrap several web sources and obtain linked data [7], [8], [9]. Furthermore, we survey a number of approaches in the bibliography that help build wrappers and the Data Virtualisation Layer in the context of semantic-web ontologies.

This paper is organised as follows: Section II surveys the techniques to build Wrappers in the context of semantic-web technologies. Furthermore, in Section III, we describe a number of approaches to build the Data Virtualisation Layer using semantic-web technologies. Finally, Section IV recaps on our conclusions.

II. Exposing Web Sources

The Web is the largest repository of information that has ever existed. This information is presented in a human friendly format using HTML (HyperText Markup Language). Unfortunately, this format complicates accessing and obtaining this information by automatic processes. Solutions to this problem are the Semantic Web and Web Services, but the lack of such services in the majority of web sites has increased the interest on information extraction.

Information extraction is the task in which relevant information from web pages is extracted and structured; note that relevancy depends completely on the context. Information extractors can be classified into two groups, namely: heuristic-based [2], [57], [58] and rule-based. The literature provides many proposals to infer information extraction rules, and they can be classified into non-supervised [17], [65], [62] and supervised [28], [44], [59]. The difference is that the former techniques do not require the user to annotate a set of training pages to indicate which the information of interest is, i.e., (nested) records, (flat) tuples, or attributes; on the contrary, they attempt to extract every piece of information that varies from page to page, so that the user only has to determine which of these data is of interest. Common information extraction rules range from regular expressions to context free grammars, first-order rules, XPath templates, and transducers. We focus on transducers, which have demonstrated their adaptability to data variability such as attributes permutations and missing attributes. Furthermore, several techniques on information extraction can be adapted to transducers.

Despite the high number of proposals on information extraction, there does not exist a universally applicable information extractor [14]. As a consequence, in a data virtualisation process, where more than one web sites are integrated, more than one information extractor are needed. We have developed an information extraction framework in which proposals on information extraction can be developed and integrated. Extracted information from web pages, using the framework, is structured by creating semantic data (OWL).

The framework components are described below:

a) Annotation Tool: The framework is accompanied by an annotation tool with which users can download and annotate web pages according to an OWL ontology in which they describe classes, properties and their relationships. Ontology classes are used to represent records of information, object properties represent nested records, and data properties represent attributes. When a set of web pages are annotated, a Dataset is created.

b) Datasets: This component provides services that allow end users to work with annotations and persist them. Users may use this component during the annotation, learning, and extraction processes. During the annotation process, this component allows users to instantiate ontology classes and properties in addition to their position in the corresponding web page. During the learning process, end users can use a dataset to retrieve and manipulate a text view or a tree view of the pages they have annotated, get the annotations sorted according to their position or to their type, obtain separating texts between annotations or work with DOM trees and annotation nodes. During the extraction process, this component allows to persist the information that is extracted to OWL files.

c) Learner: This component provides end users with services to develop rule learners. There is a service to create the skeleton of a transducer from a dataset, i.e., its states and transitions, but not the transition conditions. This service saves end users from the burden of inferring the structure of a transducer from the annotations in a dataset, since this is common to every learning algorithm. Furthermore, this service determines which annotation corresponds to each state, and it also calculates to set of text fragments, also known as separators, between every two states that are connected by means of a transition. Whilst the other components in our framework are fixed, this component is a point of variability in which software engineers only have to focus on devising their own learning algorithms to learn transition conditions building on separators.

d) Validator: This component provides a tool with which end users can k-cross validate their rule learners. It helps collect precision, recall, specificity, accuracy, sensitivity, and the F1 measure. Thanks to this tool, the results about a given proposal are empirically comparable to other proposals.

e) Utilities: This component offers some utilities to the rest of components, namely: a configurable tokeniser, a web page downloader, web page preprocessors such as an HTML cleaner, a few string alignment algorithms, and a Patricia Tree builder for the rule learners.

To validate our framework, we have developed three techniques from the literature using the framework and assessed them using a collection of Datasets. The developed techniques are NLR [33], FT [31] and SM [28]. Experiments were performed following the guidelines reported in [32]. Each Dataset was obtained by annotating 30 web pages from each web site and then we performed a k-folding cross validation in which $k = 10$. Each test was repeated 10 times. Table I shows the obtained results by the developed techniques on the different datasets. For each technique, we obtained the Precision (P), Recall (R) and the Time (T) in seconds that was necessary to learn extraction rules for each dataset.

Our results show that none of the information extraction techniques performs well for all web site. The existence of a framework is essential since it allows to include several

TABLE I
COMPARING PRECISION AND RECALL OF NLR, SM, AND FT TECHNIQUES

Dataset	NLR			SM			FT		
	P	R	T(s)	P	R	T(s)	P	R	T(s)
Books									
awesomebooks.com	1.000	0.946	15.546	1.000	0.936	7.500	1.000	0.676	4.015
betterworldbooks.com	0.993	0.915	88.375	0.877	0.894	17.859	0.920	0.514	9.406
manybooks.net	0.974	0.824	13.828	0.974	0.824	8.312	0.770	0.536	5.078
www.abebooks.com	0.000	0.000	207.234	0.847	0.700	17.171	0.000	0.000	8.546
www.waterstones.com	1.000	1.000	138.468	1.000	0.940	9.546	0.921	0.549	5.640
Sports									
en.uefa.com	1.000	1.000	39.796	1.000	0.947	8.171	1.000	1.000	4.593
playerprofiles.com	1.000	1.000	20.375	0.872	0.850	6.953	0.872	0.839	3.671
www.atpworldtour.com	0.733	0.667	72.312	0.943	0.855	8.203	1.000	0.400	12.046
www.nfl.com	0.866	0.866	101.218	0.995	0.931	12.203	1.000	0.065	7.734
www.soccerbase.com	0.789	0.778	100.984	0.903	0.851	12.609	0.793	0.537	8.203
Real Estate									
realestate.yahoo.com	0.833	0.000	63.875	1.000	0.900	19.328	1.000	0.300	11.078
www.haart.co.uk	1.000	0.950	65.296	1.000	0.885	8.781	0.950	0.142	9.812
www.homes.com	0.918	0.882	36.531	0.963	0.747	8.953	0.661	0.661	5.000
www.remax.com	0.950	0.461	75.656	1.000	0.409	9.281	1.000	0.967	5.968
www.trulia.com	0.938	0.754	128.421	1.000	0.933	30.093	1.000	0.933	15.031

techniques in our web wrapper and to select the technique that best performs on a certain web site. For example, if we want to integrate the site www.remax.com from the Real Estate category, the framework should use the FT technique since it obtains better Precision and Recall for this web site. Meanwhile, for the site playerprofiles.com in the Sports category, the NLR technique should be selected since it has the best precision and recall.

The result of information extraction from different web sites is a collection of Datasets that contain OWL files as resultsets. These files may have different ontologies, so integrating them requires a mapping phase described in the following section. Note also that Wrappers expose their data using semantic-web services, which try to mitigate the limitations of (non-semantic) web services by enriching them with semantic annotations to improve their discovery and composition [60]. Current ontologies for semantic-web services are OWL-S [38], WSMO [56], or MSM [48].

III. DATA VIRTUALISATION LAYER

In this section, we survey the approaches and techniques to implement the Data Virtualisation Layer, which is responsible for offering a number of virtual views over the ontologies exposed by semantic web services of the Service Layer.

Mediators, which are pieces of software that help software engineers integrate different ontologies, are a well-known solution to the problem of creating virtual views [20], [35], [43], [54]. A mediator relates a number of source ontologies, which contains the data of interest, to a virtual ontology, which usually contains no data. In the following, we describe the process of building and executing a mediator (cf. Figure 2).

A. Mapping generation

To build a mediator, the first step consists of designing mappings, which are the cornerstone components of mediators, and they are relationships amongst source and target ontologies [11], [15], [20], [36]. Building and maintaining mappings automatically is appealing insofar this reduces integration costs and relieves users from the burden of writing them, checking whether they work as expected or not, making changes if necessary, and restarting this cycle [5], [49]. Mappings can be of various types but, in the bibliography, approaches that generate them automatically focus on two, namely: correspondences and executable mappings.

On the one hand, correspondences are hints that specify which elements from the source and target models are related in some unspecified way [6]. They may be automatically generated by means of matching techniques [15], [20], [53]. Correspondences must be interpreted to perform integration tasks. However, there is not a unique interpretation, i.e., different approaches interpret correspondences in different ways [1], [6], [50].

On the other hand, executable mappings, also known as operational mappings, encode an interpretation of correspondences in a given query language [25], [50], [51], [55]. The main benefit of using these mappings is that the composition of the virtual data is simplified, making it more efficient and flexible: thanks to executable mappings, instead of relying on ad-hoc programs that are difficult to create and maintain, the query engine is used as the composition engine [41]. Furthermore, these engines incorporate a vast knowledge on query manipulation, from which it is derived that the executable mappings are automatically optimised for better performance of this composition.

In the bibliography, there are a number of techniques to automatically generate executable mappings [39], [42], [47], [50], [51]. Unfortunately, none of them are suitable in the context of OWL ontologies due to the following reasons, namely:

- They are based on models that do not implement semantic-web technologies, such as Mergen and

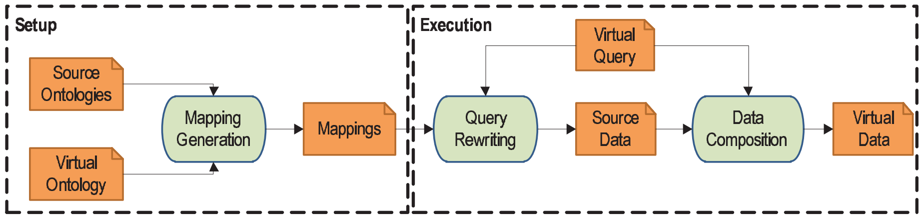

Fig. 2. Workflow of the Data Virtualisation Layer

Heuser [39], or Popa et al. [50].

- They are not based on correspondences, such as Qin et al. [51], which is based on instance examples of the virtual model, so it must be populated, which seems to be quite unusual in practice [1], [5], [21], [24], [50]. Furthermore, Parreiras et al. [47] transforms handcrafted OCL-like executable mappings into SPARQL executable mappings. Note that this approach is usually not appealing since handcrafted executable mappings increase integration costs [5], [49].
- They interpret correspondences in isolation, such as Mocan and Cimpian [42], which transform each correspondence into an executable mapping in isolation; however, correspondences are inherently ambiguous and they need to be interpreted as a whole to perform the data translation task [6], [50].

We have devised a technique to automatically generate SPARQL executable mappings between OWL ontologies that, instead of relying on examples, is based on the restrictions of source and virtual ontologies, and correspondences between these ontologies [55]. We have evaluated our technique in various integration scenarios, such as semantic-web services, evolution in DBpedia, or film reviews with promising results.

B. *Integration of ontologies*

After specifying the mappings, mediators automatically perform the integration between source and virtual ontologies. At run time, a mediator takes a query posed over the virtual ontology as input and it is responsible for answering this query using source ontologies only [26], [36]. There are two tasks regarding the answering of this query: Query Rewriting, and Data Composition.

The Query Rewriting task consists of retrieving data from the source ontologies by means of a query over the virtual ontology. Firstly, this task has to rewrite the virtual query into a single query over the source ontologies. The rewriting of the virtual query depends on the type of mappings previously generated, which may be of the following types: GaV, LaV, or GLaV. GaV mappings refer only to one element of the virtual ontology, and to a number of elements of the source ontologies [36]. In this case, the reformulation is straightforward since it is performed by unfolding the mappings into the virtual query [46]. LaV mappings comprise a number of elements of the virtual ontology and one single element of one source ontology [36]. In this case, the rewriting is performed by applying the techniques of answering queries using views [26]. Finally, GLaV mappings comprise a number of elements of both source and target ontologies [23]. In this case, the rewriting is performed using hybrid techniques from GaV and LaV [64]. Note that these techniques focus on nested relational models (specifically, Datalog and XML [26], [64]); however, there is an increasing interest on SPARQL query rewriting in the semantic-web community [16], [30].

The Query Rewriting task has to divide the source query (obtained after rewriting the virtual query) into single queries that are posed over each source ontology. Furthermore, it generates a plan that specifies how these queries must be executed. In this context, Ives et al. [29] devised a query planner that takes into account the features of the data in the source XML models. Thakkar et al. [61] proposed techniques to reduce the number of requests to source ontologies. Braga et al. [12] presented a framework to answer multi-domain queries, which can be answered by combining the data of one or more sources. Finally, Langegger et al. [34] and Quilitz and Leser [52] proposed techniques to answer SPARQL queries over distributed RDF sources. Thirdly, the Query Rewriting task is responsible for executing the previous queries over source ontologies and retrieving their data. In this context, we have devised a technique that is able to rewrite virtual queries using correspondences in the context of OWL ontologies [45].

The Data Composition task consists of creating virtual data by means of composing the data retrieved from the source ontologies in the previous task. It uses the previously specified mappings to compose this data. It is important to notice that, when using executable mappings, this task consists of executing the mappings by means of a query engine over source data to produce target data [50]. However, when using correspondences, this task is performed by interpreting correspondences using ad-hoc techniques or reasoners [19], [37]. Recall that one of the main issues regarding correspondences is that there are more than one possible interpretation, therefore, it is necessary to check of the final virtual data is coherent with expected results.

IV. Conclusions

In this paper, we present our work regarding Wrappers and the Data Virtualisation Layer in the context of ESB using semantic-web technologies. ESBs range amongst one of the most popular solutions to reduce integration costs when making software applications interoperate with each other to

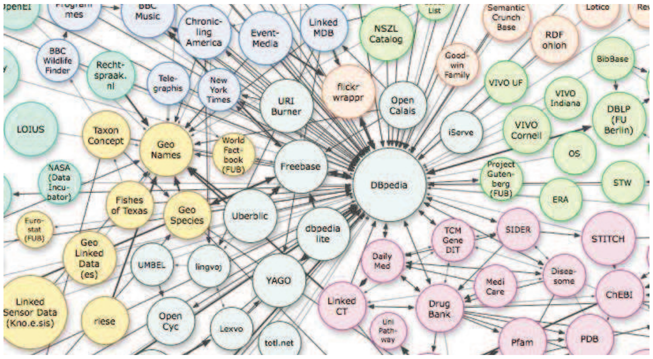

Fig. 3. Examples of web sites that offer linked data

keep their data synchronised or to create new functionality. We present a framework in which wrapping techniques can be built to wrap web sources and compose linked data. Furthermore, we survey a number of approaches to build the Data Virtualisation Layer using semantic-web technologies.

Regarding Wrappers, we have built and validated an information extraction framework in which several techniques were developed and tested. Results of the developed wrapping techniques show that none of these techniques performs well on all web sites, which confirms the need for a framework that includes several wrapping techniques and that should select the technique that best performs on each web site we are interested in.

Regarding the Data Virtualisation Layer, it is used to integrate various ontologies exposed by different semantic-web services. It is important to notice that there are a variety of web sites that offer their data as linked data (cf. Figure 3). Thank to this, we may easily integrate external data sources with software applications, e.g., we may integrate the data of films in DBpedia, which models the data stored at Wikipedia [10], with an application that is used to rent DVDs. Furthermore, it is also possible to integrate the spatial data offered by LinkedGeoData, which is an effort to add a spatial dimension to the Web of Data [4], with a GPS application.

Acknowledgments

We would like to thank to Rafael Z. Frantz for his insightful comments regarding this paper.

References

[1] Bogdan Alexe, Wang Chiew Tan, and Yannis Velegrakis. STBenchmark: towards a benchmark for mapping systems. *PVLDB*, 1(1):230–244, 2008.

[2] Manuel Álvarez, Alberto Pan, Juan Raposo, Fernando Bellas, and Fidel Cacheda. Extracting lists of data records from semi-structured web pages. *Data Knowl. Eng.*, 64(2):491–509, 2008.

[3] Grigoris Antoniou and Frank van Harmelen. *A Semantic Web Primer, 2nd Edition*. The MIT Press, 2008.

[4] Sören Auer, Jens Lehmann, and Sebastian Hellmann. LinkedGeoData: Adding a spatial dimension to the web of data. In *International Semantic Web Conference*, 2009.

[5] Philip A. Bernstein and Laura M. Haas. Information integration in the enterprise. *Commun. ACM*, 51(9):72–79, 2008.

[6] Philip A. Bernstein and Sergey Melnik. Model management 2.0: manipulating richer mappings. In *SIGMOD*, pages 1–12, 2007.

[7] Christian Bizer. The emerging web of linked data. *IEEE Intelligent Systems*, 2009.

[8] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 2009.

[9] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked data on the web (LDOW2008). In *World Wide Web Conference Series*, pages 1265–1266, 2008.

[10] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia - a crystallization point for the web of data. *J. Web Sem.*, 2009.

[11] Christian Bizer and Andreas Schultz. The R2R framework: Publishing and discovering mappings on the web. In *COLD*, 2010.

[12] Daniele Braga, Stefano Ceri, Florian Daniel, and Davide Martinenghi. Optimization of multi-domain queries on the web. *PVLDB*, 1(1):562–573, 2008.

[13] David Chappel. *Enterprise Service Bus: Theory in Practice*. OReilly, 2004.

[14] Laura Chiticariu, Yunyao Li, Sriram Raghavan, and Frederick Reiss. Enterprise information extraction: recent developments and open challenges. In *SIGMOD Conference*, 2010.

[15] Namyoun Choi, Il-Yeol Song, and Hyoil Han. A survey on ontology mapping. *SIGMOD Record*, 35(3):34–41, 2006.

[16] Gianluca Correndo, Manuel Salvadores, Ian Millard, Hugh Glaser, and Nigel Shadbolt. SPARQL query rewriting for implementing data integration over linked data. In *EDBT/ICDT Workshops*, 2010.

[17] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. RoadRunner: Towards automatic data extraction from large web sites. In *Very Large Data Bases*, pages 109–118, 2001.

[18] Jeff Davies, David Schorow, Samrat Ray, and David Rieber. *The Definitive Guide to SOA: Enterprise Service Bus*. Apress, 2008.

[19] Dejing Dou, Drew V. McDermott, and Peishen Qi. Ontology translation on the semantic web. *J. Data Semantics*, 2:35–57, 2005.

[20] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, 2007.

[21] Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.

[22] Rafael Z. Frantz, Antonia M. Reina Quintero, and Rafael Corchuelo. A domain-specific language to design enterprise application integration solutions. *Int. J. Cooperative Inf. Syst.*, 20(2):143–176, 2011.

[23] Marc Friedman, Alon Y. Levy, and Todd D. Millstein. Navigational plans for data integration. In *AAAI*, pages 67–73, 1999.

[24] Ariel Fuxman, Mauricio A. Hernández, C. T. Howard Ho, Renée J. Miller, Paolo Papotti, and Lucian Popa. Nested mappings: Schema mapping reloaded. In *Very Large Data Bases*, pages 67–78, 2006.

[25] Laura M. Haas, Mauricio A. Hernández, Howard Ho, Lucian Popa, and Mary Roth. Clio grows up: from research prototype to industrial tool. In *SIGMOD*, pages 805–810, 2005.

[26] Alon Y. Halevy. Answering queries using views: A survey. *VLDB J.*, 10(4):270–294, 2001.

[27] Gregor Hohpe and Bobbie Woolf. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley, 2003.

[28] Chun-Nan Hsu and Ming-Tzung Dung. Generating finite-state transducers for semi-structured data extraction from the web. *Inf. Syst.*, 23(8):521–538, 1998.

[29] Zachary G. Ives, Alon Y. Halevy, and Daniel S. Weld. Adapting to source properties in processing data integration queries. In *SIGMOD Conference*, pages 395–406, 2004.

[30] Yixin Jing, Dongwon Jeong, and Doo-Kwon Baik. SPARQL graph pattern rewriting for OWL-DL inference queries. *Knowl. Inf. Syst.*, 2009.

[31] Mohammed Kayed and Chia-Hui Chang. FiVaTech: Page-level web data extraction from template pages. *IEEE Trans. Knowl. Data Eng.*, 2010.

[32] Barbara Kitchenham, Shari Lawrence Pfleeger, Lesley Pickard, Peter Jones, David C. Hoaglin, Jarrett Rosenberg, and Khaled El Emam. Preliminary guidelines for empirical research in software engineering. *IEEE Trans. Software Eng.*, 28(8):721–734, 2002.

[33] Nicholas Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artif. Intell.*, 118(1-2):15–68, 2000.

[34] Andreas Langegger, Wolfram Wöß, and Martin Blöchl. A semantic web middleware for virtual data integration on the web. In *ESWC*, pages 493–507, 2008.

[35] Monika Lanzenberger, Jennifer Sampson, Horst Kargl, Manuel Wimmer, Colm Conroy, Axel Polleres, François Scharffe, Jérôme Euzenat, Asunción Gómez-Pérez, Frédéric Fürst, Francky Trichet, Rob Brennan, David Lewis, Declan O'Sullivan, José Ángel Ramos Gargantilla, and Konstantinos Kotis. Making ontologies talk: Knowledge interoperability in the semantic web. *IEEE Int. Sys.*, 23(6):72–85, 2008.

[36] Maurizio Lenzerini. Data integration: A theoretical perspective. In *PODS*, pages 233–246, 2002.

[37] Alexander Maedche, Boris Motik, Nuno Silva, and Raphael Volz. MAFRA - a MApping FRAmework for distributed ontologies. In *Knowledge Acquisition, Modeling and Management*, pages 235–250, 2002.

[38] David L. Martin, Mark H. Burstein, Drew V. McDermott, Sheila A. McIlraith, Massimo Paolucci, Deborah L. McGuinness, Katia P. Sycara, Naveen Srinivasan, and Evren Sirin. Bringing semantics to web services with OWL-S. *World Wide Web*, 10:243–277, 2007.

[39] Sergio L. S. Mergen and Carlos A. Heuser. Data translation between taxonomies. In *Conference on Advanced Information Systems Engineering*, pages 111–124, 2006.

[40] David Messerschmitt and Clemens A. Szyperski. *Software Ecosystem: Understanding an Indispensable Technology and Industry*. The MIT Press, 2003.

[41] Renée J. Miller, Laura M. Haas, and Mauricio A. Hernández. Schema mapping as query discovery. In *VLDB*, pages 77–88, 2000.

[42] Adrian Mocan and Emilia Cimpian. An ontology-based data mediation framework for semantic environments. *Int. J. Semantic Web Inf. Syst.*, 3(2):69–98, 2007.

[43] Michael Mrissa, Chirine Ghedira, Djamal Benslimane, Zakaria Maamar, Florian Rosenberg, and Schahram Dustdar. A context-based mediation approach to compose semantic web services. *ACM Trans. Internet Techn.*, 8(1), 2007.

[44] Ion Muslea, Steven Minton, and Craig A. Knoblock. Hierarchical wrapper induction for semistructured information sources. *Autonomous Agents and Multi-Agent Systems*, 4(1/2):93–114, 2001.

[45] Carlos R. Osuna, David Ruiz, Rafael Corchuelo, and José Luis Arjona. SPARQL query splitter: query translation between different contexts. In *JISBD*, pages 320–323, 2009.

[46] Alberto Pan, Juan Raposo, Manuel Álvarez, Paula Montoto, Vicente Orjales, Ángel Viña, Justo Hidalgo, Anastasio Molano, and Lucía Ardao. The denodo data integration platform. In *Very Large Data Bases*, pages 986–989, 2002.

[47] Fernando Silva Parreiras, Steffen Staab, Simon Schenk, and Andreas Winter. Model driven specification of ontology translations. In *ER*, pages 484–497, 2008.

[48] Carlos Pedrinaci, Dong Liu, Maria Maleshkova, David Lambert, Jacek Kopecký, and John Domingue. iServe: a linked services publishing platform. In *ORES*, 2010.

[49] Michalis Petropoulos, Alin Deutsch, Yannis Papakonstantinou, and Yannis Katsis. Exporting and interactively querying web service-accessed sources: The CLIDE system. *ACM Trans. Database Syst.*, 32(4), 2007.

[50] Lucian Popa, Yannis Velegrakis, Renée J. Miller, Mauricio A. Hernández, and Ronald Fagin. Translating web data. In *VLDB*, pages 598–609, 2002.

[51] Han Qin, Dejing Dou, and Paea LePendu. Discovering executable semantic mappings between ontologies. In *OTM*, pages 832–849, 2007.

[52] Bastian Quilitz and Ulf Leser. Querying distributed RDF data sources with SPARQL. In *ESWC*, pages 524–538, 2008.

[53] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB J.*, 10(4):334–350, 2001.

[54] Carlos R. Rivero, Inma Hernández, David Ruiz, and Rafael Corchuelo. A reference architecture for building semantic-web mediators. In *IWSSA*, 2011.

[55] Carlos R. Rivero, Inma Hernndez, David Ruiz, and Rafael Corchuelo. Generating sparql executable mappings to integrate ontologies. In *ER*, 2011.

[56] Dumitru Roman, Uwe Keller, Holger Lausen, Jos de Bruijn, Rubén Lara, Christoph Bussler, Axel Polleres, Dieter Fensel, Cristina Feier, and Michael Stollberg. Web service modeling ontology. *Applied Ontology*, 1(1):77–106, 2005.

[57] Yuan Kui Shen and David R. Karger. U-REST: an unsupervised record extraction system. In *World Wide Web Conference Series*, pages 1347–1348, 2007.

[58] Kai Simon and Georg Lausen. ViPER: augmenting automatic information extraction with visual perceptions. In *International Conference on Information and Knowledge Management*, pages 381–388, 2005.

[59] Stephen Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272, 1999.

[60] Katia P. Sycara, Massimo Paolucci, Anupriya Ankolekar, and Naveen Srinivasan. Automated discovery, interaction and composition of semantic web services. *J. Web Sem.*, 1(1):27–46, 2003.

[61] Snehal Thakkar, José Luis Ambite, and Craig A. Knoblock. Composing, optimizing, and executing plans for bioinformatics web services. *VLDB J.*, 14(3):330–353, 2005.

[62] Jiying Wang and Frederick H. Lochovsky. Data extraction and label assignment for web databases. In *WWW*, pages 187–196, 2003.

[63] Jeff Weiss. Aligning relationships: Optimizing the value of strategic outsourcing. Technical report, 2005.

[64] Cong Yu and Lucian Popa. Constraint-based XML query rewriting for data integration. In *SIGMOD Conference*, pages 371–382, 2004.

[65] Yanhong Zhai and Bing Liu. Structured data extraction from the web based on partial tree alignment. *IEEE Trans. Knowl. Data Eng.*, 18(12):1614–1628, 2006.