Lifting Tabular Data to RDF: A Survey

Manuel Fiorelli [0000-0001-7079-8941], Armando Stellato [0000-0001-5374-2807]

University of Rome Tor Vergata, Department of Enterprise Engineering,
via del Politecnico 1, 00133 Roma, Italy
fiorelli@info.uniroma2.it, stellato@uniroma2.it

Abstract. Tabular data formats (e.g. CSV and spreadsheets) combine ease of use, versatility and compatibility with information management systems. Despite their numerous advantages, these formats typically rely on column headers and out-of-band agreement to convey semantics. There is clearly a large gap with respect to the Semantic Web, which uses RDF as a graph-based data model, while relying on ontologies for well-defined semantics. Several systems have been developed to close this gap, supporting the conversion of tabular data to RDF. This study is a survey of these systems, which have been analyzed and compared. We identified commonalities and differences among them, discussed different approaches and derived useful insights on the task.

Keywords: CSV, TSV, Excel, Conversion, RDF, Survey

1 Introduction

More and more organizations are going to share their data under open licenses with the aim of fostering their reuse. Usually, this kind of open data includes spreadsheets and, in general, tabular data, which combine wide tooling support, flexibility and conveniency. However, tabular data usually lack explicit semantics backed by a formal ontology, as they rely on column headers and out-of-band information. Therefore, they only enable a low grade in the five-star model of Linked (Open) Data [1].

Tabular formats are still convenient at the beginning of the path toward Linked Data, as almost any data management system can export to CSV or spreadsheets. Subsequent steps include data cleaning, data transformation and, finally, data upload to a new infrastructure based on RDF (Resource Description Framework). Another very frequent scenario involves dataset updates: dataset managers have already migrated to RDF, but they receive many updates from third parties expressed through (possibly heterogeneous) tabular formats (as in the AGRIS Dataset [2]).

Unsurprisingly, several systems have been developed to support this common use case of converting tabular data to RDF. Being interested in data integration and data management, we wanted to make sense of this offering. Accordingly, the contribution of this study is a review of these systems, highlighting their characteristics, similarities and peculiarities. In Section 2, we first described these systems individually. Then, in Section 3 we developed a comparative summary, which allowed us to better understand commonalities and differences among them, discuss different approaches and derive insights on the task. Finally, Section 4 provides the conclusions.

2 Reviewed Systems

2.1 Standalone Converters

Any23. Any23 [3] is a swiss-army knife supporting the conversion of several formats to RDF. It can be used as a library, web service or command line tool. Any23 can convert CSV files under the assumption that each row contains the description of one resource using a vocabulary generated systematically from the header of the table.

Grafter. Grafter [4] is a library that uses an internal DSL based on the functional language Clojure (`http://clojure.org/`). The laziness of Clojure allows Grafter to stream the execution of a processing pipeline, making it scalable to a large amount of data. Furthermore, a pipeline made exclusively of pure functions (i.e. free of side effects) can be executed any number of times without side effects. The functional nature of processing pipelines enables solutions, such as previewing the transformation on a subset of the input data. Grafter aims to separate different concerns in a pipeline, such as loading of input data, cleaning of such data and the actual transformation. Grafter has been integrated into the cloud-based platform DataGraft [5] for the transformation and publication of data as linked data. Grafter is accessed via Grafterizer, a web application that can represent transformation pipelines and RDF templates graphically instead of relying on the Clojure-based DSL. The application supports live preview of the data as these are cleaned and processed, as well as of the final RDF. Furthermore, transformations can be shared and subsequently reused.

Populous. Populous [6] uses OPPL (Ontology Pre-Processing Language) [7] as a mapping language. OPPL supports representation and execution of ontology design patterns [8]. Populous provides a simple tabular user interface for the population of an ontology by domain experts. The table is backed by a template for the given domain ontology, which is instantiated with the values in each row. In line with its goal, Populous supports constraining the columns with respect to an ontology and data catalogs.

RDF123. RDF123 [9] provides an application for writing mapping specifications, as well as a web service for executing them against spreadsheets. It strives to move away from the assumption of rows representing homogenous resources. While every row is transformed using the same graph pattern, the mapping language achieves the desired level of flexibility by having all components of the graph (i.e. vertices and edges) be computed via complex expressions (possibly containing conditionals): when they produce an empty value, the corresponding component is omitted from the generated graph. The mapping is represented as RDF graphs, the components of which may contain expressions interpreted by the RDF123 runtime. **Fig. 1** contains an excerpt of a mapping that uses the value of the second column to set the property `vcard:street-address`.

RMLEditor. RMLEditor [10] provides a graphical user interface for specifying the transformation of raw data to RDF, without knowledge of the underlying mapping language RML[1]. This system allows to lookup column names on LOV (Linked Open Vocabularies) [11] to find a suitable vocabulary property. While RMLEditor mostly deals with CSV/TSV files organized in rows, its mapping language has explicit support for joins, even over different data sources.

SKOS Play!. SKOS Play! converter[2] is an open-source (web) application for converting Excel spreadsheets to SKOS thesauri or RDF datasets in general. The tool assumes that the spreadsheets are structured into rows and relies on information encoded in a heading row to customize the mapping of the cells: e.g. language, datatype, value separator, etc. SKOS Play! supports cross-row references based on a selectable column, and different columns can be associated with subjects generated from other columns.

Spread2RDF. Spread2RDF [12] is a command-line tool that supports Excel/Excelx, Google spreadsheets, OpenOffice, LibreOffice and CSV. It uses an internal DSL (Domain-Specific Language) based on Ruby (`https://www.ruby-lang.org/`) to specify the transformation, which is executed on blocks of columns. Within a block, a cell underneath a column is converted to an RDF node, which is then used as the object of a property (bound to the column) of the subject associated with the row. By defining further column blocks, it is possible to generate resources that are used both as objects in the context of the original column block as well as subjects in the context of the new column block. While the processing is mostly row-oriented, it is also possible to address specific cells in a worksheet, to pinpoint specific pieces of information. The DSL supports templates: named fragments that can be reused within a mapping specification. Spread2RDF can compile mapping specifications into standalone executables.

```
<rdf:RDF
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:foo="http://www.foo.org/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="Ex:$1">
    <rdf:type
      rdf:resource="http://www.w3.org/2006/vcard/ns#Ad-
dress"/>
    <vcard:street-address>Ex:$2</foaf:mbox>
…
</rdf:RDF>
```

Fig. 1. An excerpt of a mapping for RDF123

[1] `https://rml.io/specs/rml/`
[2] `http://labs.sparna.fr/skos-play/convert`

```
<#address>
  a :Resource ;
  :identity [
    :source_column 1 ;
  ] ;
  :type vcard:Address ;
  :attribute[
    :property vcard:street-address;
    :source_column 2
  ]
…
.
```

Fig. 2. An excerpt of a mapping for Vertere

TabLinker. TabLinker [13] is thought for converting statistical data to RDF using the Data Cube [14] vocabulary. While most systems can seamlessly operate on diverse tabular formats, TabLinker depends on the features of spreadsheets, in particular Microsoft Excel, e.g. the use of cell styles as a way to describe the layout of the data: where are the headers (vertical and horizontal), how they are hierarchically decomposed, etc. Furthermore, the vocabulary Open Annotation [15] can be used to represent harmonization rules, e.g. how the various dimensions can be mapped to RDF properties.

Tarql. Tarql [16] is a command-line tool allowing for SPARQL queries over tabular data, interpreted as a table of variable bindings, i.e. a sequence of assignments of terms to variables (usually the result of matching the `WHERE` clause of a SPARQL query).

Vertere. Vertere [17] is a command-line tool that represents mapping rules in RDF and these rules are intended to be executed row by row: this data parallelism can be exploited for parallel execution. **Fig. 2** illustrates how to bind the second column to the property `vcard:street-address`.

XLWrap. XLWrap [18] is an RDF wrapper for spreadsheets, CSV and TSV files, which can be retrieved locally or fetched via HTTP. The transformation is driven by mappings expressed as RDF in Trig syntax using a dedicated mapping vocabulary. A mapping consists of templates, which are applied to portions of the input data depending on user-defined transformations. Transformations, templates and other parts of the mapping can contain expressions referencing the input data, much like what happens in spreadsheets. XLWrap is also available as a server that automates the execution of transformations (e.g. after a change in the input files or the mappings). Furthermore, the sever can optionally start a SPARQL endpoint using Joseki (now deprecated by Apache Fuseki[3]).

[3] https://jena.apache.org/documentation/fuseki2/

2.2 Database to RDF solutions

Sparqlify. Sparqlify [19] is a system for mapping relational databases to RDF that also supports simpler tabular data formats. Its mapping language SML (Sparqlification Mapping Language) combines SQL (Structured Query Language) for querying the source data with the possibility to instantiate graph patterns (like the ones found in SPARQL) out of RDF terms computed based on the information extracted via SQL.

2.3 Knowledge Development Environments

Mapping Master (M2). Mapping Master (M^2) [20] is a plugin for Protégé dealing with data in the financial domain with complex layouts. Its mapping language combines the Manchester Syntax for expressing complex OWL (Web Ontology Language) axioms with (possibly relative) references to cells of the tabular data. The application of a formula to an arbitrary area of the input data allows for handling different data layouts.

PoolParty's Spreadsheet Import. PoolParty [21] can import a thesaurus from a tabular representation in which the indentation of rows represents the hierarchical relationships between concepts.

Sheet2RDF. Sheet2RDF [22] (an early version of which was described in [23]) can be used as a library, a standalone command-line tool and as a tool integrated into the web-based collaborative knowledge development platform VocBench 3 [24]. The latter provides a graphical user interface with suggestions informed by the ontology and data already existing in a target dataset. Sheet2RDF offers different levels of mapping specification, which progressively trade ease of use for flexibility. At the highest level, Sheet2RDF interprets the input rows as descriptions of homogenous resources, whose properties should be found in the header. If the input file conforms to certain conventions[4] inspired by the ones of SKOS Play!, no configuration is requested at all. One interesting feature of Sheet2RDF is its ability to trigger specific customizations for certain modeling patterns requiring more complex property values (e.g. SKOS-XL [25] reified labels). The user can optionally provide hints, e.g. bind a column to a property with a non-matching name, a class, natural language, etc. Sheet2RDF uses an iterative approach, attempting the automatic conversion again considering user-supplied information. A third level consists in allowing users to define new heuristics, patterns and corresponding triggers. Under the hoods, Sheet2RDF repurposes the knowledge acquisition platform CODA [26] using its transformation language PEARL [27]. Indeed, PEARL editing (in a dedicated editor) is the lowest level for specifying transformations.

TopBraid Composer's Tabular Data Import. TopBraid Composer [28] provides wizards for importing tabular data: a table represents a class whose properties are associated with the different columns. It is possible to map that class and properties to an existing ontology, or to import both the new vocabulary items and their instances.

---

[4] `http://art.uniroma2.it/sheet2rdf/documentation/heuristics.jsf`

2.4 Data Wrangling Tools

GraphDB OntoRefine. GraphDB OntoRefine is a data transformation tool that has been available in Ontotext GraphDB [29] since version 8.0. OntoRefine is integrated into its management web application (i.e. workbench) and it is based on OpenRefine. It supports TSV, CSV, *SV, XLS, XLSX, JSON, XML, RDF as XML, and Google sheet: additional formats can be introduced through OpenRefine extensions. OntoRefine relies on a systematic conversion of the input data to RDF triples, which can be accessed through a SPARQL endpoint. SPARQL and SPIN (SPARQL Inference Notation) can be used to transform this raw data according to any target domain model.

Karma. Karma [30] is an ontology-based web application for data integration supporting several input formats (including hierarchical ones). Karma assumes a row-oriented layout for tabular formats. The transformation to RDF is specified visually as a graph, which is really a template for the graph that will be instantiated for each input row. The transformation is internally implemented using an interpretation of the R2RML [31] language called KR2RML. Karma allows for creating new columns executing Python scripts over the content of other columns. This capability can be also found in Grafter. Karma follows the paradigm of programming-by-example, since the system learns from models previously constructed by users how to construct models in the future.

LODRefine. LODRefine extends Open Refine (`http://openrefine.org/`) to provide a web application for the generation of RDF and the interaction with existing RDF datasets, e.g. for the purpose of identity resolution. It inherits from Open Refine a range of facilities for data cleaning and the possibility to load different data formats into a common tabular structure. The mapping of cleaned data to RDF is expressed graphically, with user supporting facilities such as auto-completion of ontology terms. GREL (Google Refine Expression Language) enables complex value transformations.

2.5 Full-fledged ETL solutions

CSV2RDF. CSV2RDF [32] builds on Sparqlify, proposing an extension of Semantic MediaWiki [33] for crowdsourcing the generation of the mappings; these are generated through an iterative process seeded by an initial mapping suggested by the system. Mappings are expressed through a MediaWiki Template supporting a subset of SML. For visualization, CSV2RDF integrates CubeViz [34] (data cubes) and Facete [35] (spatial data).

Csv2rdf4lod-automation. Csv2rdf4lod-automation [36] represents mapping rules in RDF, formalizing the underlying language as an ontology. The idea of Csv2rdf4lod-automation is to start from a systematic conversion of the input data and to refactor this raw RDF by applying several enhancements: e.g. configure the property URIs and associate column groups to different resources. Concerned with provenance, the system updates data monotonically, adding process metadata. TabLinker has a similar concern.

DataLift. DataLift [37] firstly performs a systematic conversion to RDF, which is followed by further processing steps, including discovery of relevant ontologies on LOV [11], refactoring using SPARQL and linking to other datasets.

LinkedPipes ETL. LinkedPipes ETL [38] is an ETL (Extract Transform Load) framework for RDF aimed at overcoming some limitations of UnifiedViews concerning usability and integrability. LinkedPipes ETL has a more sophisticated debugging capability than UnifiedViews. Regarding the development of conversion pipelines, LinkedPipes ETL can suggest new components, based on input/output compatibility, likelihood of usage in a certain context, fulltext search in component metadata, etc. LinkedPipes ETL uses essentially the same approach to triplication of tabular data as UnifiedViews.

UnifiedViews. UnifiedViews [39] is an ETL framework supporting complex data transformation pipelines. These are composed of data processing units (DPUs) arranged into complex data flows, enabling data units (i.e. RDF, files and relational tables) to move from a DPU to another. DPU are classified into extractors (no input, obtain data from outside), transformers (transform input data unit into output ones), loaders (no output, deploy input data unit to some destination), quality assessors (like transformers, but output quality reports). Extensible through the implementation of new DPUs, the framework provides a lot of them out of the box. There is a DPU for turning CSV to RDF, which favors a systematic conversion: further processing is done using a SPARQL DPU. Another DPU can convert Excel spreadsheets to CSV. UnifiedViews has a web-interface with a graphical editor of pipelines. Users can debug a fragment of a pipeline, inspect the input data units, etc. The Semantic Integrator edition of PoolParty uses UnifiedViews to replace its native import mechanism.

3 Comparative Summary

Table 1 contains the systems described in Section 2 and aims at simplifying a comparison between them. We will also provide general considerations on the task.

A very discriminative feature lies in the way the mapping of input data to RDF is specified, allowing for different levels of customization. At the one extreme, Any23 does not support any customization, relying on a systematic mapping that generates property URIs from column headers. At the other extreme, there are systems providing a dedicated mapping language. Vertere and Csv2rdf4lod-automation represent the mapping in RDF, while RDF123 uses more straightforward RDF templates. A comparison between **Fig. 1** and **Fig. 2** (in the sections devoted to these systems) should make the difference clear. XLWrap combines templates like the ones used by RDF123 with transformations of the input data expressed in RDF. Grafter and Spread2RDF use an internal DSL in some general-purpose programming language. Advantages include falling back to the hosting language for coding some aspects of the process and reliance on the existing tools (e.g. IDEs, project managers and interactive consoles).

Users not acquainted with these programming languages might be annoyed by learning a new language and installing its runtime. Spread2RDF addresses the latter problem, by optionally compiling a mapping specification into a standalone executable. The familiarity problem with the mapping language is solved by using SPARQL, which is already well-known to semantic web practitioners. This is used, in particular, by those systems (DataLift, OntoRefine, LinkedPipes ETL and UnifiedViews) that adopt a two-step approach: systematic conversion, followed by RDF transformation. Tarql conflates these two steps by directly allowing the evaluation of SPARQL over tabular data. Sheet2RDF executes the conversion as a single step; however, it offers an incremental and layered approach to mapping specification. Moreover, its PEARL language is based on SPARQL for the specification of graph patterns.

Reuse is a concern in mapping specification (e.g. use the same pattern for SKOS-XL preferred and alternative labels): e.g. Spread2RDF templates, Sheet2RDF patterns and triggers or cascades of SPARQL queries in UnifiedViews and LinkedPipes ETL.
Most systems deal with data laid out in rows representing homogenous records, much like a table in a relational database. Accordingly, systems thought for mapping databases to RDF (e.g. Sparqlify) may optionally process CSV files and the like. The systematic mapping implemented by Any23 is analogous to the Direct Mapping [40] standardized for relational databases. Moreover, the latter is explicitly exploited by DataLift. Beyond Direct Mapping, R2RML [31] standardizes a way to express a custom mapping of relational databases to RDF. Karma's KR2RMLextends it to support other (hierarchical) data sources. RML (used by RMLEditor) has a similar goal, extending R2RML to support heterogeneous data formats, including CSV, XML and JSON. This support consists on a default iteration logic and different reference formulation options (e.g. XPath for XML, JSONpath for Json, etc.).

TabLinker, Mapping Master and XLWrap are specialized in complex data layouts, which can be found in the statistical and financial domains. Following Langegger and Wöß [18], these layouts are just representation models for different information models.

Tabular data formats (e.g. CSV) are in fact bidimensional, and support for different formats is often achieved by merely extracting data along these two dimensions. Spreadsheets, however, allow for additional dimensions. Firstly, they may consist of multiple sheets: these can be considered as multiple tables (e.g. in UnifiedViews and LinkedPipes ETL) or may have a special interpretation (e.g. Sheet2RDF allows for a sheet declaring prefix mappings). Tools such as XLWrap even supports cross-sheet references. Furthermore, even individual sheets allow for additional dimensions: e.g. TabLinker uses cell style and comment to represent additional metadata beyond the two-dimensional grid of cells.

Mapping Master and Populous work at the level of OWL axioms: this approach is particularly suitable when the goal is not to populate an ontology (i.e. increase the A-box) but to extend the ontology vocabulary (i.e. increase the T-box).

Most systems are standalone converters, but we identified other scenarios: database mapping solutions (Sparqlify), knowledge development environments (MappingMaster, Sheet2RDF, PoolParty's and TopBraid Composer's importers), data wrangling tools (Karma, LODRefine and OntoRefine) and full-fledged ETL solutions (CSV2RDF, Csv2rdf4lod-automation, DataLift, LinkedPipes ETL, UnifiedViews).

Table 1. Comparative summary of reviewed systems

System name	Output	Input layout	Mapping	Assistance to Mapping	Integrations
Any23	RDF	rows	—	—	—
Csv2rdf	RDF	rows	MediaWiki template	skeleton mappings	Semantic MediaWiki, Facete, CubeViz, Sparqlify
csv2rdf4lod-automation	RDF	rows	Conversion ontology	—	—
DataLift	RDF	rows	—	—	LOV, Linking services
Grafter	RDF	rows	Clojure DSL	Grafterizer	DataGraft
GraphDB OntoRefine	RDF	rows	SPARQL, SPIN, GREL	SPARQL editor with syntax highlighting and completion	GraphDB Workbench
Karma	RDF	Hierarchical data	Graphical DSL, KR2RML	Learning techniques, Steiner tree optimization, lookup target ontology	Sesame Server and Workbench
LinkedPipes ETL	RDF	rows	SPARQL	debugging, component suggestion	—
LODRefine	RDF	rows	Graphical DSL, GREL	lookup in the target ontology	CrowdFlower, NER services, reconciliation with DBpedia
Mapping Master	OWL	varied layouts	M^2 Language	user interface	Protégé
PoolParty	RDF	indented rows	—	—	—
Populous	OWL	rows	OPPL	—	—
RDF123	RDF	rows	RDF template	—	—
RMLEditor	RDF	rows	Graphical DSL, RML	—	LOV
Sheet2RDF	RDF	rows	PEARL	PEARL editor, heuristics, lookup in the target ontology	VocBench 3
SKOS Play!	RDF	rows	structured header	—	—

System name	Output	Input layout	Mapping	Assistance to Mapping	Integrations
Sparqlify	RDF	rows	SML	—	—
Spread2RDF	RDF	column blocks	Ruby DSL	—	—
TabLinker	Data Cube	data cubes	Annotated Excel file	—	—
Tarql	RDF	rows	SPARQL	—	—
TopBraid Composer	RDF	rows	Wizard	—	—
Unified Views	RDF	rows	SPARQL	debugging	PoolParty
Vertere-RDF	RDF	rows	RDF-based language	—	—
XLWrap	RDF	varied layouts	Conversion ontology + RDF template	—	Joseki

4 Conclusions

We surveyed several systems for the triplification of tabular data. After discussing them in isolation, we highlighted commonalities, differences and interesting insights. We identified different approaches to the conversion, the peculiarities of certain types of inputs and data formats, discussed easiness of use and different deployment scenarios. The summary provided by Table 1 reveals that complex layouts have not been explored by most tools yet; still, 25% of them support a non-row layout, with 2 of them explicitly supporting the most varied layouts. This feature can then be very selective if the input data is not conformant to the usual row series. Otherwise, the choice of a tool can be driven by the integration with other systems and the provided assistance to mapping.

Future work includes improving our survey in two directions: broadening the range of analyzed tools and extending the comparison framework with more detailed features, to facilities a systematic comparison between different mapping specification approaches.

References

1. Berners-Lee, T.: Linked Data. In: Design Issues. (Accessed 2006) Available at: https://www.w3.org/DesignIssues/LinkedData.html
2. Celli, F., Anibaldi, S., Folch, M., Jaques, Y., Keizer, J.: OpenAGRIS: using bibliographical data for linking into the agricultural knowledge web. In : Agricultural Ontology Services (AOS), Bangkok, Thailand (2011)
3. Apache Any23 - CSV Extractor. Available at: https://any23.apache.org/dev-csv-extractor.html

4. Grafter - Linked Data Machine Tools. Available at: http://grafter.org/
5. Roman, D. et al.: DataGraft: One-Stop-Shop for Open Data Management. Semantic Web 9(4), 393-411 (2018)
6. Jupp, S. et al.: Populous: a tool for building OWL ontologies from templates. BMC Bioinformatics 31(1) (2012)
7. Egana, M., Antezana, E., Stevens, R.: Transforming the axiomisation of ontologies: The ontology pre-processor language. In : Proceedings of the Fifth OWLED Workshop on OWL: Experiences and Directions (OWLED-2008) (2008)
8. Gangemi, A., Presutti, V.: Ontology design patterns. In : Handbook on ontologies. Springer Berlin Heidelberg (2009), pp.221-243
9. Han, L., Finin, T., Parr, C., Sachs, J., Joshi, A.: RDF123: From Spreadsheets to RDF. In Sheth, A. et al., eds. : The Semantic Web - ISWC 2008 vol. 5318. Springer Berlin Heidelberg (2008), pp.451-466
10. Heyvaert, P. et al.: RMLEditor: A Graph-Based Mapping Editor for Linked Data Mappings. In : The Semantic Web. Latest Advances and New Domains (Lecture Notes in Computer Science) vol. 9678. Springer, Cham (2016), pp.709-723
11. Vandenbussche, P.-Y., Atemezing, G.A., Poveda-Villalón, M., Vatant, B.: Linked Open Vocabularies (LOV): A gateway to reusable semantic vocabularies on the Web. Semantic Web 8(3), 437-452 (December 2016)
12. A DSL-based converter for spreadsheets to RDF. Available at: https://github.com/marcelotto/spread2rdf
13. TabLinker. Available at: https://github.com/Data2Semantics/TabLinker
14. W3C: The RDF Data Cube Vocabulary. In: World Wide Web Consortium (W3C). (Accessed January 14, 2014) Available at: http://www.w3.org/TR/vocab-data-cube/
15. Ciccarese, P., Soiland-Reyes, S., Clark, T.: Web Annotation as a First-Class Object. IEEE Internet Computing 17(6), 71-75 (2013)
16. Tarql. Available at: https://github.com/cygri/tarql
17. Vertere-RDF. Available at: https://github.com/knudmoeller/Vertere-RDF
18. Langegger, A., Wöß, W.: XLWrap – Querying and Integrating Arbitrary Spreadsheets with SPARQL. In : The Semantic Web - ISWC 2009 (Lecture Notes in Computer Science) vol. 5823. Springer, Berlin, Heidelberg (2009), pp.359-374
19. Sparqlify. Available at: http://aksw.org/Projects/Sparqlify.html
20. O'Connor, M.J., Halaschek-Wiener, C., Musen, M.A.: Mapping Master: A Flexible Approach for Mapping Spreadsheets to OWL. In : The Semantic Web – ISWC 2010. Springer Berlin Heidelberg (2010)
21. In: PoolParty Semantic Suite - Semantic Technology Platform. Available at: https://www.poolparty.biz/
22. ART Group: Sheet2RDF. Available at: http://art.uniroma2.it/sheet2rdf/
23. Fiorelli, M., Lorenzetti, T., Pazienza, M.T., Stellato, A., Turbati, A.: Sheet2RDF: a Flexible and Dynamic Spreadsheet Import&Lifting Framework for RDF. In Ali, M., Kwon, Y.S., Lee, C.-H., Kim, J., Kim, Y., eds. : Current Approaches in Applied Artificial Intelligence (Lecture Notes in Computer Science) vol. 9101. Springer International Publishing (2015), pp.131-140

24. Stellato, A. et al.: VocBench 3: a Collaborative Semantic Web Editor for Ontologies, Thesauri and Lexicons. Semantic Web 11(5), 855-881 (2020)
25. World Wide Web Consortium (W3C): SKOS Simple Knowledge Organization System eXtension for Labels (SKOS-XL). In: World Wide Web Consortium (W3C). (Accessed August 18, 2009) Available at: http://www.w3.org/TR/skos-reference/skos-xl.html
26. Fiorelli, M., Pazienza, M.T., Stellato, A., Turbati, A.: CODA: Computer-aided ontology development architecture. IBM Journal of Research and Development 58(2/3), 14:1 - 14:12 (March-May 2014)
27. Pazienza, M.T., Stellato, A., Turbati, A.: PEARL: ProjEction of Annotations Rule Language, a Language for Projecting (UIMA) Annotations over RDF Knowledge Bases. In : Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey (2012)
28. TopBraid Composer. Available at: http://topbraidcomposer.info/
29. Ontotext: Ontotext GraphDB. Available at: http://graphdb.ontotext.com/
30. Knoblock, C.A. et al.: Semi-automatically Mapping Structured Sources into the Semantic Web. In Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V., eds. : The Semantic Web: Research and Applications (Lecture Notes in Computer Science). Proceedings of the 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31 2012. vol. 7295. Springer International Publishing (2012), pp.375-390
31. Das, S., Sundara, S., Cyganiak, R.: R2RML: RDB to RDF Mapping Language. In: World Wide Web Consortium - Web Standards. (Accessed September 27, 2012) Available at: https://www.w3.org/TR/r2rml/
32. Ermilov, I., Auer, S., Stadler, C.: CSV2RDF: User-Driven CSV to RDF Mass Conversion Framework. In : Proceedings of the ISEM '13, September 04 - 06 2013, Graz, Austria (2013)
33. Krötzsch, M., Vrandečić, D., Völkel, M., Haller, H., Studer, R.: Semantic Wikipedia. Web Semantics: Science, Services and Agents on the World Wide Web 5(4), 251-261 (2007)
34. Martin, M., Abicht, K., Stadler, C., Auer, S., Ngomo, A.-C.N., Soru, T.: CubeViz - Exploration and Visualization of Statistical Linked Data. In : Proceedings of the 24th International Conference on World Wide Web, WWW 2015 (2015)
35. Stadler, C., Martin, M., Auer, S.: Exploring the Web of Spatial Data with Facete. In : Companion proceedings of 23rd International World Wide Web Conference (WWW), pp.175-178 (2014)
36. Lebo, T., Williams, G.T.: Converting Governmental Datasets into Linked Data. In : Proceedings of the 6th International Conference on Semantic Systems, New York, NY, USA, pp.38:1--38:3 (2010)
37. Scharffe, F. et al.: Enabling linked data publication with the Datalift platform. In : AAAI workshop on semantic cities (2012)
38. Klímek, J., Škoda, P., Nečaský, M.: LinkedPipes ETL: Evolved Linked Data Preparation. In : The Semantic Web (Lecture Notes in Computer Science) vol. 9989. (2016), pp.95-100
39. Knap, T. et al.: UnifiedViews: An ETL tool for RDF data management. Semantic Web 9(5), 661-676 (2018)
40. World Wide Web Consortium (W3C): A Direct Mapping of Relational Data to RDF. In: World Wide Web Consortium (W3C). (Accessed September 27, 2012) Available at: http://www.w3.org/TR/rdb-direct-mapping/