Semantically Enhanced Quality Assurance in the JURION Business Use Case

Dimitris Kontokostas[1(✉)], Christian Mader[2], Christian Dirschl[3], Katja Eck[3], Michael Leuthold[3], Jens Lehmann[1], and Sebastian Hellmann[1]

[1] Institut für Informatik, AKSW, Universität Leipzig, Leipzig, Germany
{kontokostas,lehmann,hellmann}@informatik.uni-leipzig.de
[2] Semantic Web Company, Vienna, Austria
c.mader@semantic-web.at
[3] Wolters Kluwer Germany, Munich, Germany
{cdirschl,keck,mleuthold}@wolterskluwer.de
http://aksw.org

Abstract. The publishing industry is undergoing major changes. These changes are mainly based on technical developments and related habits of information consumption. Wolters Kluwer already engaged in new solutions to meet these challenges and to improve all processes of generating good quality content in the backend on the one hand and to deliver information and software in the frontend that facilitates the customer's life on the other hand. JURION is an innovative legal information platform developed by Wolters Kluwer Germany (WKD) that merges and interlinks over one million documents of content and data from diverse sources such as national and European legislation and court judgments, extensive internally authored content and local customer data, as well as social media and web data (e.g. DBpedia). In collecting and managing this data, all stages of the Data Lifecycle are present – extraction, storage, authoring, interlinking, enrichment, quality analysis, repair and publication. Ensuring data quality is a key step in the JURION data lifecycle. In this industry paper we present two use cases for verifying quality: (1) integrating quality tools in the existing software infrastructure and (2) improving the data enrichment step by checking the external sources before importing them in JURION. We open-source part of our extensions and provide a screencast with our prototype in action.

Keywords: RDF quality · Linked Data · Enrichment

1 Introduction

The publishing industry is - like many other industries - undergoing major changes. These changes are mainly based on technical developments and related habits of information consumption[1]. The world of customers has changed dramatically and as an information service provider, Wolters Kluwer wants to meet

[1] For example: http://hmi.ucsd.edu/pdf/HMI_2009_ConsumerReport_Dec9_2009.pdf.

these changes with adequate solutions for customers and their work environment. For a couple of years, Wolters Kluwer has already engaged in new solutions to meet these challenges and to improve processes for generating good quality content in the backend on the one hand and to deliver information and software in the frontend that facilitates the customers life on the other hand.

One of these frontend applications is a platform called JURION.[2] JURION is a legal information platform developed by Wolters Kluwer Germany (WKD) that merges and interlinks over one million documents of content and data from diverse sources, such as national and European legislation and court judgments, extensive internally authored content and local customer data; as well as social media and web data (e.g. from DBpedia). In collecting and managing this data, all stages of the Data Lifecycle are present – extraction, storage, authoring, interlinking, enrichment, quality analysis, repair and publication. On top of this information processing pipeline, the JURION development teams add value through applications for personalization, alerts, analysis and semantic search.

Based on the FP7 LOD2 project[3], parts of the *Linked Data stack*[4] have been deployed in JURION to handle data complexity issues. LOD2 aimed at developing novel, innovative Semantic Web technologies and also at the expansion and integration of openly accessible and interlinked data on the web. More detailed information can be found in [1]. WKD acted as a use case partner for these technologies, supported the development process of semantic technologies and integrated them to support the expansion of linked data in business environments. The software development process and data life cycle at WKD are highly independent from each other and require extensive manual management to coordinate their parallel development, leading to higher costs, quality issues and a slower time-to-market. This is why the JURION use case presented here is situated within both the Software Engineering as well as in the Data Processing area.

Through the ALIGNED project[5], JURION focuses on closing the gap between Software & Data Engineering. This paper describes the JURION results of the first phase of the project. In this phase, we concentrated mainly on the enhancement of data quality and repair processes. We created novel approaches for integrating RDF tools in the existing software engineering tool stack and created bindings to widely used Java libraries. We additionally created a link validation service for cleaning up external metadata residing in our databases. As a proof of concept, we open sourced some of our extensions and provide a screencast of the prototype implementation.

This industry paper is structured as follows: Sect. 2 provides a detailed description of JURION and its architecture. The existing tools that are used and enhanced are detailed in Sect. 3. Section 4 describes the challenges that drove this development. Section 5 provides the detailed approach we took for tackling each challenge. We provide an in-depth evaluation in Sect. 6 and conclude in Sect. 7.

[2] See JURION website https://www.jurion.de/de/home/guest.

[3] http://lod2.eu/Welcome.html.

[4] http://stack.linkeddata.org/.

[5] http://aligned-project.eu.

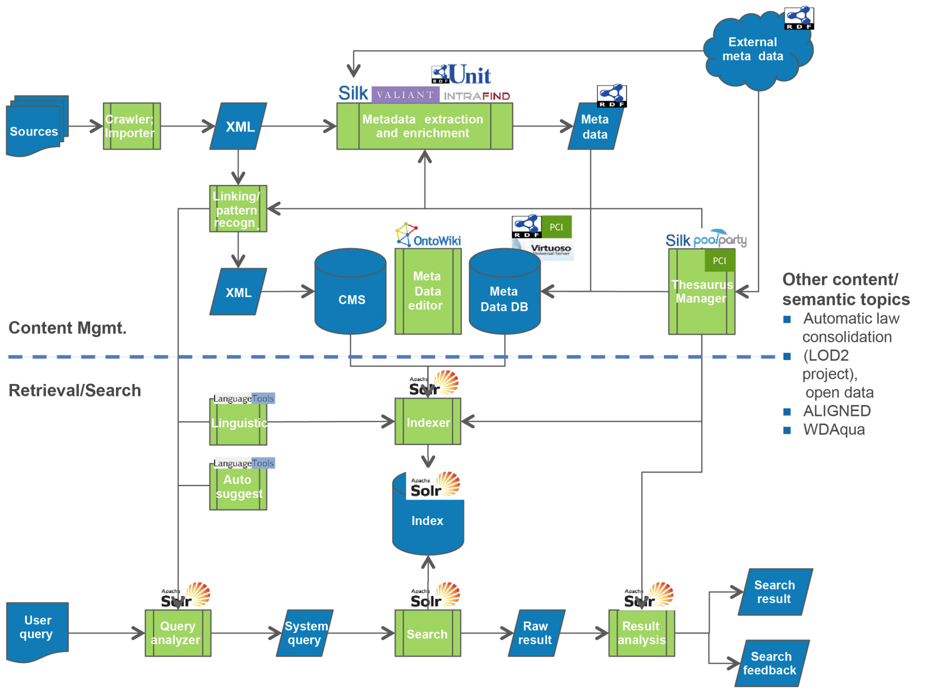

Fig. 1. JURION content pipeline and semantic search

2 The JURION Business Use Case

WKD is a leading knowledge and information service provider in the domains of law, companies and tax and offers high quality business information for professionals. This information is more and more integrated in digital solutions and applications. When using these solutions, customers can make critical decisions more efficiently and they can enhance their productivity in a sustainable way. Wolters Kluwer n.v. is based in more than 40 countries and serves customers in more than 175 countries worldwide.

JURION is the legal knowledge platform developed by WKD. It is not only a legal search platform, but considers search for legal information as an integrated part of the lawyer's daily processes. JURION combines competencies in the areas of legal publishing, software, portal technology and services, which cover all core processes of the lawyer within one single environment by connecting and integrating many different internal and external data sources.

The main goal of JURION is not to be yet another search engine. On the one hand, because Google as a reference application has made major progress in recent times, even in search environments dedicated to professionals. On the other hand, legal research is just one part of the lawyers main and daily tasks. So the higher the coverage of core processes of a digital offering, the more added-value on the customers side is generated and the higher the willingness to pay for that service will be. In addition, the more touchpoints between vendor and customer exist, the lower is the possibility for the service provider to be replaced by others.

Figure 1 describes the overall JURION content processing and retrieval infrastructure. Within the content pipeline, metadata is extracted from the proprietary WKD XML schema and transformed in RDF. In the thesaurus manager, controlled vocabularies and domain models based on SKOS standard are created, maintained and delivered for further usage. The indexing process of a search engine includes more and more additional information on top of the pure text. Queries are analyzed for legal references and keywords, which are matched against existing data in the metadata management systems. Once there are matches, the semantic relations are shown in the results overview by specific references to texts and related knowledge visualizations.

Since most of these new service offerings like workflow support at the lawyers desk are not document and content driven, the current paradigm of using pure XML as the only major data format had to be given up. Data and metadata are driving internal processes and therefore most of the features and functionalities in the JURION application. So, this data must not be locked in DTDs or XML schemas anymore. Conversion of this data in traditional RDBMS would have been possible, but the main benefits of these systems like high performance and robustness were not the major requirements in this setting. Instead, the data needed to be stored and maintained in a much more flexible and interconnectable format, so that new data format requirements like adding new metadata or a new relationship type could be processed in a more or less real-time fashion. Semantic web technologies were chosen to meet that need, since e.g. their triple store technology supports this flexibility and since high performance is as already mentioned not a major requirement in a CMS environment. In addition, due to government initiatives on a national and European level, quite a lot of valuable data in the legal field was available in SKOS and RDF format, so that the integration effort for this data was rather limited, as soon as the basic technical infrastructure was laid. Once the internal and integrated external data was available, new features like personalization based on domain preferences (e.g. boosting labor law documents in the result list, based on current search behavior), context-sensitive disambiguation dialogues to resolve query issues (e.g. contract as labor contract or sales contract) or the sustainable linking to external data sources like EU directives and court decisions with their own legal contexts, e.g. across languages and countries could be established.

2.1 Related Work

In industrial settings, architectural system details are most times kept hidden due to conflicts of interest. However, major companies in the media & publishing sector are using semantic web technologies. In 2013, BBC published their internal knowledge graph[6]. BBC keeps an open position on Linked Data and

[6] http://www.bbc.co.uk/things/.

publishes many of their semantic-web-enabled features[7]. Thomson Reuters provides B2B semantic solutions with OpenCalais[8] and PermID[9]. Guardian uses Linked Data to augment the data they provide behind a paid API[10]. At the end of 2015, Springer announced a semantic wrapper of their data[11]. Nature Publishing Group has been an early adopter on linked data[12]. Finally, Pearson is also observed to use semantic web technologies[13].

3 Use Case Tools

In this paper, we focus on the use and enhancement of two tools: RDFUnit, an RDF Unit Testing suite and PoolParty Thesaurus Server (PPT) which is part of the PoolParty Semantic Suite.[14]

RDFUnit [3] is an RDF validation framework inspired by test-driven software development. In software, every function should be accompanied by a set of unit tests to ensure the correct behaviour of that function through time. Similarly, in RDFUnit, every vocabulary, ontology, dataset or application can be associated by a set of data quality test cases. Assigning test cases (TCs) in ontologies results in tests that can be reused by datasets sharing the same ontology.

The test case definition language of RDFUnit is SPARQL, which is convenient to directly query for identifying violations. For rapid test case instantiation, a pattern-based SPARQL-Template engine is supported where the user can easily bind variables into patterns. RDFUnit has a *Test Auto Generator* (TAG) component. TAG searches for schema information and automatically instantiates new test cases. Schema information can be in the form of RDFS[15] or OWL axioms that RDFUnit translates into SPARQL under Closed World Assumption (CWA) and Unique Name Assumption (UNA). These TCs cover validation against: domain, range, class and property disjointness, (qualified) cardinality, (inverse) functionality, (a)symmetricity, irreflexivity and deprecation. Other schema languages such as *SHACL*, *IBM Resource Shapes* or *Description Set Profiles* are also supported. RDFUnit can check an RDF dataset against multiple schemas but when this occurs, RDFUnit does not perform any reasoning/action to detect inconsistencies between the different schemas.

The PoolParty Semantic Suite[16] is a commercial product developed by Semantic Web Company.[17] For the JURION business case, PPT is of main

[7] http://www.bbc.co.uk/blogs/internet/tags/linked-data.
[8] http://www.opencalais.com/.
[9] https://permid.org/.
[10] http://www.theguardian.com/open-platform/blog/linked-data-open-platform.
[11] http://lod.springer.com/wiki/bin/view/Linked+Open+Data/About.
[12] http://www.nature.com/ontologies/.
[13] https://www.semantic-web.at/pearson.
[14] https://www.poolparty.biz/.
[15] http://www.w3.org/TR/rdf-schema/.
[16] https://www.poolparty.biz/.
[17] https://www.semantic-web.at/.

relevance. With PPT, taxonomists can develop thesauri in a collaborative way, either from scratch or supported by extraction of terms from a document corpus. The created thesauri fully comply to the 5-star Open Data principles[18] by using RDF and SKOS as the underlying technologies for representing the data. Using Linked Data technologies, it is possible to automatically retrieve potential additional concepts for inclusion into the thesauri by querying SPARQL endpoints (e.g. DBpedia). Furthermore, PPT provides interfaces to conveniently identify and link to related resources that are either defined in other PPT projects that reside on the same server or located on the Web. Depending on configuration, taxonomies developed with PPT can be made available as data dumps in various RDF serialization formats or directly queried by using the taxonomy's SPARQL endpoint. When additional semantics are required that exceed that of SKOS, PPT supports the creation of custom schemas. Taxonomies can use them do define their own classes and relation types by using elements of RDFS such as `rdfs:subClassOf` or `rdfs:domain` and `rdfs:range`.

Users of PPT are at any time supported by automated quality assurance mechanisms. Currently, three methodologies are in place: first, conformance with a custom schema or the SKOS model is ensured by the user interface. One for instance, cannot create two preferred labels in the same language. Second, the enforcement level of some quality metrics can be configured by the user so that it is, e.g., possible to get an alert if circular hierarchical relations [4] are introduced. Third, a taxonomy can be checked "as a whole" against a set of potential quality violations, displaying a report about the findings.

4 Challenges

JURION merges and interlinks over one million documents of content and data from diverse sources. Currently, the software development process and data life cycle are highly independent from each other and require extensive manual management to coordinate their parallel development, leading to higher costs, quality issues and a slower time-to-market. The higher flexibility of the data model as well as the shortened time-to-market for data features can only be materialized when most of the testing and QA effort – after data & chema changes are introduced – are tackled in a semi-automatic fashion. Thus, benefits like flexibility and scalability only materialize when it is not necessary to do all the quality checks manually, or involving expensive domain experts.

As depicted in Fig. 1, within the content pipeline, metadata is extracted from the proprietary WKD XML schema and transformed in RDF. Due to regular changes in the XML format, the correct transformation process based on existing XSLT scripts must be secured, so that no inconsistent data is fueled into the metadata database. In the thesaurus manager, controlled vocabularies and domain models based on SKOS standard are created, maintained and delivered for further usage. The integrity of the knowledge management system as a whole needs to be ensured. Therefore, regular local and global quality checks need to

[18] http://5stardata.info/en/.

be executed, so that e.g. inconsistencies across different controlled vocabularies can be detected and resumed.

Through the ALIGNED project, we target to enable JURION to address more complex business requirements that rely on tighter coupling of software and data. In this paper, we focus on improving the metadata extraction process as well as inconsistencies across controlled vocabularies and in particular external links coming from the JURION enrichment phase.

4.1 Metadata RDF Conversion Verification

At the top of Fig. 1 of the content pipeline, metadata is extracted from the proprietary WKD XML schema and transformed to RDF. Due to regular changes in the XML format, the correct transformation process based on existing XSLT scripts must be secured, so that no inconsistent data is fuelled into the metadata database. The main challenge of this task is to reduce the error in data transformation and accelerate the delivery of metadata information to JURION.

Approach & Goals. Based on the schema, test cases should automatically be created, which are run on a regular basis against the data that needs to be transformed. The errors detected will lead to refinements and changes of the XSLT scripts and sometimes also to schema changes, which impose again new automatically created test cases. This approach provides: 1. better control over RDF metadata 2. streamlined transformation process from XML to RDF 3. early detection of errors in RDF metadata, since the resulting RDF metadata are a core ingredient for many subsequent process steps in production and application usage, and 4. more flexibility in RDF metadata creation.

Impact. Continuous high quality triplification of semi-structured data is a common problem in the information industry, since schema changes and enhancements are routine tasks, but ensuring data quality is still very often purely manual effort. So any automation will support a lot of real-life use cases in different domains.

Existing Infrastructure. As part of the core CMS tasks within JURION each WKD XML document is checked-in through internal workflow functionality and gets converted to RDF which is based on the "Platform Content Interface" (PCI) ontology. The PCI ontology is a proprietary schema that describes legal documents and metadata in OWL. Due to change requests and new use cases for the RDF metadata in the ontology, the conversion logic or both the conversion logic and ontology need amendments. In these cases we need to make sure that the RDF data that is generated from the WKD XML documents still complies with the PCI ontology for quality assurance.

Quality Assurance. As a gatekeeper to avoid loading flawed data into the triple store, each result of the conversion from WKD XML into PCI RDF was sent to a dedicated, proprietary validation service that inspects the input and verifies compliance with the ontology. This approach assured that the conversion results are verified but came with major issues which makes it unsuitable for ad-hoc testing and quick feedback. The three most important ones are:

- the current service only processes entire PCI-packages, i.e. several datasets; this makes error detection on single data units quite difficult and service errors block the whole processing pipeline
- the service is a SOAP based web service that operates asynchronously with many independent process steps, which imposes high complexity on its usage
- it depends on other services and requires permanent network access and therefore is potentially unstable.

To improve these issues, we want to implement unit-test scenarios that can be run directly coupled to the development environment of the conversion projects and is therefore seamlessly integrated into the workflow. The tests should be run both automatically on every change in the project, but also be able to be manually triggered. Tests should be easily extendable and expressive enough to effectively spot issues in the conversion process. The feedback loop should be coupled as tightly as possible to the submitted change.

4.2 Quality Control in Thesaurus Management

At the middle right of Fig. 1 of the JURION content pipeline, controlled vocabularies and domain models are created, maintained and delivered for further usage, based on the SKOS standard. The integrity of the knowledge management system as a whole needs to be ensured. Therefore, regular local and global quality checks need to be executed, so that e.g. inconsistencies across different controlled vocabularies can be detected and resumed. The domain models and controlled vocabularies within and beyond the system are partly dependant on each other; sometimes there is even an overlap, e.g. concerning domains and concepts. This dependency must be transparent and consistency must be maintained. In addition, versioning issues must be addressed, since subsequent processes need to be aware of model changes, especially deletions.

Until now we had no effective overview over the validity of linked sources. This for example, causes problems in frontend applications where links do not resolve. The only way to evaluate the quality was to analyze the frontend representations of the linked sources or to follow a link to detect a missing source. There was in general no process in place to control the validity of external sources.

Approach & Goals. WKD is already using the thesaurus management system PoolParty for several vocabularies. With the increasing operational use, amount of content and the extended functionality to define custom schemas, we encounter various pre-existing and new challenges: 1. Transparency of vocabulary dependencies, 2. Consistency of vocabulary dependencies, 3. Versioning issues due to model changes, deletions etc., 4. Tracking of subsequent changes performed by different users, 5. Process definition for the maintenance of vocabularies, 6. Usability related to the understanding of the data models, 7. Ambiguities and doublets. Our goal is to deploy prototypical approaches in the operational system of WKD and to investigate approaches to ensure data quality, enhance the transparency and consistency of dependencies, resolve versioning issues, deploy tracking functionalities, deploy a maintenance process, identify and encounter ambiguities, and deploy a solution for dealing with doublets.

Impact. The creation and maintenance of knowledge models is gaining importance in the Web of Data. These tasks are increasingly being executed by SME's in the domain, not in knowledge modelling and IT as such. Therefore, better automatic support of these processes will directly help achieving quality and efficiency gains.

Existing Infrastructure. As the number of controlled vocabularies and custom schemas we are integrating in the metadata management tool PoolParty is increasing, we need solutions that give an overview of existing relations between projects and external data and schemas. Besides, the number of user roles is growing so that we also need a solution that provides an overview for a number of different users with different purposes. By different queries and enhancements we want to get an impression about the relations between projects and the usage of specific custom schemas. Connections between projects and schemas are not easily traceable. Owners of vocabularies need to provide documentation so that others can also understand the projects and their scope and context. Without this documentation, it is hard to analyse the different projects. Within the tool the user can only analyse the individual concepts for relations to investigate any relations with schemas. For linking to other projects it is possible to get a list of links. This list does not provide the number of links and specific numbers for different kinds of linking. These figures need to be searched manually.

5 Semantically Enhanced Quality Assurance

The JURION challenges described in Sect. 4 are targeted by the following extensions to the JURION workflow and its components: (a) verification of correct metadata conversion, (b) integration of repeatable tests into the development environment and (c) automation of external link clean-up. In the following sections we present our approach for tackling each challenge.

A six-minute screencast video has been developed which showcases the demonstrator described in this paper. The screencast shows some background context on the JURION use case and the prototype implementation in action. It is available on YouTube[19] and is linked through the ALIGNED project website[20]. RDFUnit [3], including the extensions developed for this demonstrator is available as open source code. PoolParty is a commercial closed-source

[19] https://youtu.be/6aLXK7N7wFE.

[20] http://aligned-project.eu/jurion-demonstration-system/.

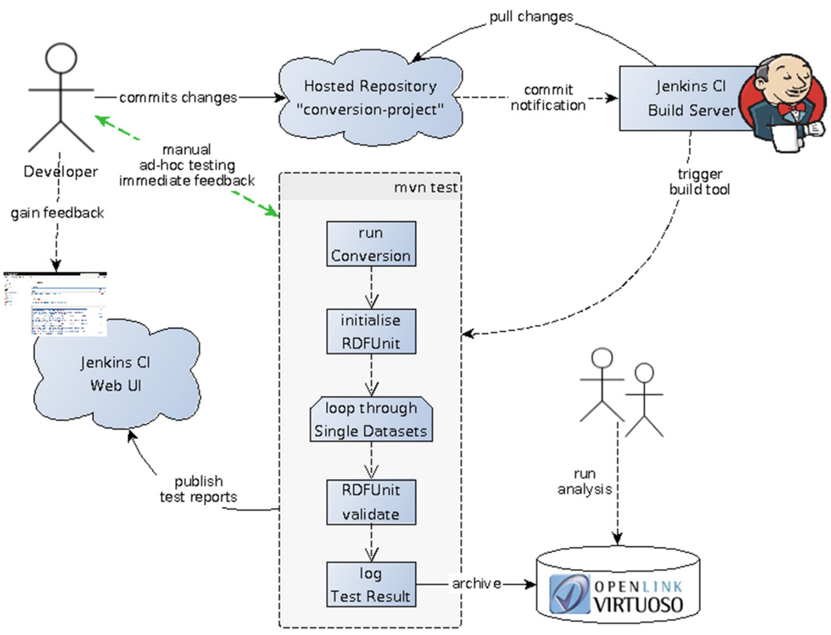

Fig. 2. RDFUnit integration in JURION

product[21] and the extensions developed for this demonstrator will be folded into future releases of the product when this is commercially viable.

XML to RDF Conversion Verification. To allow comparable and reproducible tests with short execution times, a number of WKD XML reference documents have been selected, against which the actual conversion into PCI RDF is executed. Each resulting RDF dataset is then verified individually. The prototyped solution (cf. Figure 2) integrates RDFUnit as the core driver of the tests. It is set upon the JUnit-API[22] so that it integrates seamlessly into the development tool chain and the software build process in general. For instance, a developer can trigger the test chain manually on his local workstation to retrieve direct feedback at any time and any change in the conversion project automatically leads to full test run which is performed by the build system.

All executed tests are based on RDFUnit's auto-generators, which derive test cases at runtime from the latest version of the PCI-ontology. As a proof of concept RDFUnit's test results (the validation model based on the Test Driven Data Validation Ontology [2]) linked to this test is stored into a Virtuoso triplestore to enable future analysis/reviews of historical data.

[21] https://www.poolparty.biz/.
[22] http://junit.org/javadoc/latest/.

Early and quick feedback on changes to the project is very valuable to assure that the project is in good health and existing functionality meets the defined expectations. Good coverage with automated tests prevents bugs from slipping in released functionality which may have side effects on other parts of the system. RDFUnit enables possibilities but still needs a tighter integration as a library with our existing toolchain to improve reporting capabilities and make its feedback even more useful. RDFUnit proves as very useful and will be a fixed component of the operational tech stack within WKD JURION from now on. We will provide further requirements to improve RDFUnit's integration into our development pipeline. At a later point in time, we will utilize RDFUnit to enable monitoring the existing data store to implement quality assurance on operational side.

We additionally integrated RDFUnit with JUnit. JUnit is a unit testing framework for Java, supported by most Java developer IDEs and build tools (e.g. maven, Eclipse). JUnit allows to execute repeatable tests as part of the development workflow in line with the test-driven development paradigm. As part of the described use cases we contributed the integration of the RDFUnit-JUnit-module[23]. We added specific Java annotations on JUnit classes that can define the input dataset and the schema that RDFUnit can test. Each test generated by RDFUnit TAGs is translated in a separate JUnit test and reported by JUnit. This approach facilitates simpler setups and can verify specific input files. The benefit is the immediate integration of RDF dataset testing on existing software development tool stacks.

Verifying Available Instance Data for Linked External Metadata. We currently implemented two different kinds of statistical metrics and integrated them into the PoolParty UI, (i) checking for external links validity and (ii) links to other PoolParty projects on the server. These metrics differ in the methodology they are evaluated. Checking the validity of external links cannot be done using SPARQL and requires external tool support (e.g., Java code, see Sect. 6 on external link validity). Reporting links to PoolParty projects can be achieved in a similar way than checking for data consistency violations [3]. Each statistical property can be formulated as a SPARQL query, which is executed on the relevant project data, i.e., the current project data and metadata as well as all linked project data and metadata.

6 Evaluation

Our methodology for baseline data collection is divided into three categories: productivity, quality and agility. The analysis is based on measured metrics and the qualitative feedback of experts and users. Participants of the evaluation study were selected from WKD staff in the fields of software development and

[23] https://github.com/AKSW/RDFUnit/tree/master/rdfunit-junit.

data development. There were seven participants in total: four involved in the expert evaluation and three content experts involved in the usability/interview evaluation.

6.1 Productivity

Collection Methods & Metrics. We collected content expert evaluations for the metadata extraction verification, a test suite was set up to measure metrics of productivity for all implemented features and finally, interviews were conducted to obtain feedback from prospective users of PoolParty functionalities. For the RDFUnit integration, we measured the total time for quality checks and error detection, as well as the need for manual interaction. For the external link validation we measured the number of checked links, the number of violations and the total time.

RDFUnit enabled us to develop automated tests that provide tight feedback and good integration into the existing toolchain. It enabled error messages, which point exactly to the offending resource, making bug fixing much easier. Depending on the size of the document and size of the ontology, total time to execute a test-suite varies, but can be indicated with 1 ms to 50 ms per single test. With this approach, the feedback is as close to real-time as possible; currently a couple of minutes. Since it is possible to trigger the quality checks manually at any time through the existing developers IDE menus, speedy performance is desirable to avoid developer idle time. Though quality checks can be triggered by manual execution, they are always verified automatically by the build system, which sends a notification if an error occurs. Due to the development process it is guaranteed that with every change the whole set of quality checks is executed and reported automatically. The current setup generates and runs about 44000 tests with a total duration of 11 min which may scale-up easily when parallelized or clustered. The details-section reports each violated RDFUnit test individually with it's corresponding error message and a list of failing resources.

The external link validation is a new feature that evaluates the links to external sources and informs the user in case the sources are not available anymore. Previously, it was only possible to check the links manually in random samples. We evaluated a sample of four WKD projects that were checked for external relations. The results are provided in Table 1. This validation process can considerably improve time spent on error reduction, so that external links can be maintained efficiently and corrected at short notice. The table shows that the validation of the links of the Arbeitsrechtsthesaurus took a relatively long time. The presentation of the results was well understood. In general, the tool was received well by the experts, which was reflected by their feedback in the interviews.

6.2 Quality

Collection Methods & Metrics. We set up a test suite to measure quality metrics for all features, we additionally used content expert dataset evaluations. For the

Table 1. External link validation of several Wolters Kluwer thesauri

External linking	Concepts	Checked links	Violations	Time
Thesaurus	6510	0	0	0,5 min
Gesetzesthesaurus	1307	0	0	0,001 s
Arbeitsrechtsthesaurus	1728	1868	60	10 min
Gerichtsthesaurus	1503	1434	81	3 min 4 s

metadata extraction verification we collected the number of detected error categories, the test coverage and expert evaluation. For the external link validation we measured the correctness of results and usability aspects.

In the metadata extraction verification we had the following questions: "What kind of errors can be detected" and "is categorization possible"? We used the RDFUnit supported axioms to categorize the errors wherever possible. As stated in Sect. 3, RDFUnit supports many RDFS & OWL axioms. Regarding test coverage, RDFUnit provides test coverage metrics. However, we did not yet integrate the test coverage metrics in our operational tool stack. This is a next (crucial) step, as we need to evaluate the relevance of individual tests to the tested dataset. Ideally, we would need to get the percentage of the input dataset that is covered by tests and how many of these tests actually measured features of the input dataset. From our expert evaluation we concluded that it is helpful to spot errors introduced by changes, since issues spotted in this way can be assumed to point to really existing errors; the causes of which can be identified and addressed. In contrast, successful tests are less significant as we are not yet able to evaluate whether and how the measurements taken correspond to target measures and these tests do not point to concrete errors. To resolve this, we will proceed to integrate measures that help evaluating the test cases on the one hand, and the input datasets on the other hand.

To analyse the quality of the external links validation, we evaluated results of two thesauri. Figure 3 depicts the number of concepts that are detected. In the Labor Law Thesaurus there are 40 correctly detected broken image links. The courts thesaurus provides results of better quality. All of the detected violations are indeed broken links and are corrected. The presentation of external link validation results works well in general. The number of checked and incorrect links are shown in the overview. The interface, in its prototype state, still lacks some usability. The interview strengthens this impression of the expert evaluation, and gives suggestions to improve the interface. A next step will be to add the related preferred label of the concept that links to the broken external source to enhance the usability and the quality evaluation, as well as an improvement of the general structure to ease also the reading of the results.

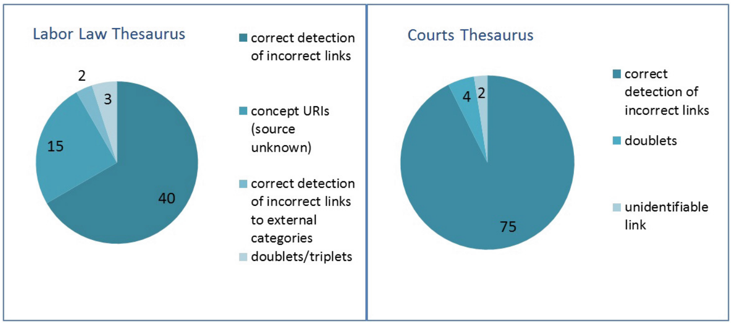

Fig. 3. Incorrect external links of the Labor Law Thesaurus and the Courts Thesaurus

6.3 Agility

Collection Methods & Metrics. We collected evaluations of content experts for the RDFUnit integration and evaluations of technical experts for the PoolParty tool. We collected metrics for time to include new requirements in RDFUnit integration and for the external links validation scope of external link checks, possibility of integration, configuration time and extension.

With respect to the XML-to-PCI conversion verification, including new constraints or adapting existing constraints is a convenient process. The procedure works by adding new reference documents to the input dataset to make the test environment as representative as possible. As the process of generating tests and testing is fully automated, it adapts very easily to changed parameters. However, adding more documents to the input dataset increases the total runtime of the test-suite, which affects the time to feedback. Therefore, one must be careful with the selection of proper reference documents.

Regarding the scope of external link checks, the current solution resolves all links that point to hosts which are not identified as the local host. While this is certainly useful for getting an overview, in many cases it is desired to limit the link lookups and adapt the way links to external datasets are detected. Possibilities are, for instance, to use the current projects base URI or regular expression-based techniques. Internal link checks are not yet implemented. Determining which URIs should be resolved can be done either directly with SPARQL or within the Java resolution algorithm. In each case, the effort for change is low, allowing for agile reaction on changed requirements. However, changes to the current configuration require recompilation and redeployment of PoolParty, which reduces agilty. In order to address performance issues, our future plans are to delegate link checking to an external application. This task would be suitable to be performed by UnifiedViews[24], which we also envision to take a central role in the PoolParty Semantic Suite. This also requires to have a method in place that enables UnifiedViews to report the created statistics back to PoolParty, which is on the roadmap for future work.

[24] https://github.com/UnifiedViews.

6.4 Analysis

The evaluation of the prototype shows clearly that during the first phase of prototype development, we have achieved our aim to improve the productivity and quality of data processes within the data lifecycle. With the presented features, these improvements could be shown. Performance and quality/error rates of the test results were reasonable. In addition to the tests, there will be new scope for data repair processes to correct the detected errors of the dataset, as we gained new insights into data violations (e.g. more categories of violated external links than we expected). Nonetheless, there need to be further improvements, especially with regard to usability, performance, integration of functionalities and required details that are not yet fully working.

In summary, the productivity of data processes is clearly improved by the initial prototype. The statistics and external link validation functionalities can help to save much time by replacing time consuming manual work by efficient data overviews.

Concerning the quality of the prototype functionalities, the results are very satisfying. For notifications and external link validation there are only few issues. For the data transformation with RDFUnit there needs to be further investigations to enable comprehensive and extensive data testing results. Usability issues need to be tackled in all of the features for a better operational implementation. As this is only an initial prototype, usability was less of a focus.

The testers feedback for agility of features is quite positive. The agility of RDFUnit is seen as satisfying, as the automated service allows the implementation of new requirements easily. External link validation has a reasonable agility and is planned to be done by an external application to address performance issues.

7 Conclusions & Future Work

In this paper, we described an industrial use case of RDF technologies with JURION. We managed to weaken the gap between software and data development by integrating quality checks in our existing software tool stack. We provided a screencast of our prototype and contributed bindings between RDFUnit and JUnit as open source.

In future work we plan to improve the RDFUnit integrations in JURION. Further research is needed for test coverage reports as well as the generation test analytics. For example, time to fix a bug, identification of regressions, etc. In regard to PoolParty we will improve the user experience of the extenal metadata checking tool. As far as functionality is concerned, we plan to support internal link checking and provide advanced configuration for the end users.

Acknowledgments. This work was supported by grants from the EU's H2020 Programme ALIGNED (GA 644055). WKD and JURION is a use case partner in the ALIGNED project.

References

1. Auer, S., Bryl, V., Tramp, S. (eds.): Linked Open Data - Creating Knowledge Out of Interlinked Data, vol. 8661, 1st edn. Springer, Heidelberg (2014)
2. Kontokostas, D., Brümmer, M., Hellmann, S., Lehmann, J., Ioannidis, L.: NLP data cleansing based on linguistic ontology constraints. In: Presutti, V., d'Amato, C., Gandon, F., d'Aquin, M., Staab, S., Tordai, A. (eds.) ESWC 2014. LNCS, vol. 8465, pp. 224–239. Springer, Heidelberg (2014)
3. Kontokostas, D., Westphal, P., Auer, S., Hellmann, S., Lehmann, J., Databugger, R.C.: A test-driven framework for debugging the web of data. In: WWW Companion 2014, pp. 115–118 (2014)
4. Mader, C., Haslhofer, B., Isaac, A.: Finding quality issues in SKOS vocabularies. In: Zaphiris, P., Buchanan, G., Rasmussen, E., Loizides, F. (eds.) TPDL 2012. LNCS, vol. 7489, pp. 222–233. Springer, Heidelberg (2012)