I18N OF LINKED DATA TOOLS WITH RESPECT TO WESTERN BALKAN LANGUAGES

Uroš Milošević, Valentina Janev, Sanja Vraneš[1]
[1]*Univerzitet u Beogradu, Institut Mihajlo Pupin*

Abstract – *To be able to cope with all the challenges that are emerging with the rise of the Web of Data, linked data tools must find ways to deal with its heterogeneity with regard to the content being published and the languages that content is being published in. Adequate internationalization support is a must for distributing or consuming multilingual data, but providing it is not straightforward. In this paper, we analyze the problems semantic web tools are faced with when it comes to data being published in Western Balkan languages. We look at resource presentation mechanisms, show how IRIs can solve many of the common problems related to Serbian alphabets, but also that the Semantic Web is still not ready for full transition to internationalized identifiers. We also review different serialization options and explain why most XML-based formats are not suitable for the task of internationalization. To see how popular linked data tools deal with such data, we propose an evaluation methodology, and give an insight into the i18n-readiness of some popular semantic web tools.*

1. INTRODUCTION

For any linked data / semantic web tool, achieving true cross-border acceptance means successfully tackling the inevitable challenges that go along with it, such as providing support for multiple languages and their corresponding alphabets. Having to deal with a plethora of international sources of data, it is of paramount importance for any linked data application to be able to handle such a task.

The process of designing software applications with the possibility of adapting them to various languages and regions (without engineering changes) in mind is known as *internationalization* (i18n). Internationalization in the context of linked data applications corresponds to the task of providing support for importing, modifying and exporting data that contain non-US-ASCII characters, without loss of integrity. That is, any semantic web tool must ensure accuracy and consistency of data, as it is transferred between multiple such applications, throughout its entire lifecycle. However, to make sure the design of each tool adheres to the basic i18n principles, a robust evaluation methodology is needed.

One challenging domain for that task is the region of Western Balkans, as the encodings used to represent all scripts of South-Eastern Europe cover most of European character sets as well.

In this paper, we will first take a look at internationalization in the context of multilingual linked data, and the main obstacle in Section 2. In Section 3, we will analyze the barriers imposed on our task by the Western Balkan languages, with a focus on the alphabets being used and the corresponding encoding support. In Section 4, we will give an overview of possible resource identification mechanisms, and the issues faced by such mechanisms when it comes to representing special characters that are present in some alphabets. The problems and the possible solutions will then be taken to the Linked Data paradigm, in Section 5, where we will discuss RDF serialization challenges. The methodology used to evaluate individual semantic web tools will be presented in Section 6, and the actual evaluation results with respect to some popular linked data applications will be given in Section 7. Finally, we will conclude this paper in Section 8.

1.1. LOD2 STACK

The FP7 LOD2 project[1] is a European initiative to improve coherence and quality of data published on the Web, close the performance gap between relational and RDF data management, establish trust on the Linked Data Web and generally lower the entrance barrier for data publishers and users. The LOD2 Stack is a distribution of integrated software tools that support the entire linked data lifecycle, starting with extraction and authoring/creation, via enrichment, interlinking and fusing, to maintenance and will be used as a test bed for the i18n evaluation methodology.

2. MULTILINGUAL LINKED DATA

Multilingual data is defined as data that appears in a multilingual setting and contains references to human readable textual information. *Multilingual linked data* is multilingual data that follows the linked data principles. In general, almost all linked data can be considered multilingual, when it's purpose is to serve multilingual communities. The nature of the LOD Cloud itself makes this dataset inherently multilingual and publishing or consuming such data poses a difficult task due to a number of i18n challenges, some of which we'll present in detail in sections 4 and 5, and show that without adequate support for internationalization, there is no support for multilingual data.

3. I18N FOR WESTERN BALKAN LANGUAGES

The most wide-spread script used in the region of Western Balkans is an extension of the Latin script,

[1] http://lod2.eu

known as *Gaj's Latin* alphabet (ISO 8859-2; Latin-2), common for Croatia, Serbia, and Bosnia and Herzegovina. A superset with two additional characters (*Ś* and *Ź*) is used in Montenegro, while Slovene uses a subset, leaving out a total of 5 symbols (*Ć*, *Đ*, *Dž*, *Lj*, *Nj*; Slovene doesn't count digraphs as individual characters).

The second most popular script is the Serbian Cyrillic alphabet (ISO 15924), sharing a lot in common with the Macedonian national alphabet. It is worth noting that Serbo-Croatian is the only European language with active digraphia (use of more than one writing system for the same language). This makes any Serbian dataset that combines the two scripts intrinsically multilingual, as far as tool support is concerned. The two different encodings used to represent these scripts provide enough of a challenge for successful internationalization of any software tool.

4. RESOURCE IDENTIFIERS

The first issue the task of i18n raises in the Linked Data paradigm comes from one of its most fundamental building blocks – the resource identifiers.

4.1. UNIFORM RESOURCE IDENTIFIERS

The omnipresent, general resource representation mechanism for Linked Data in most knowledge bases today is the *Uniform Resource Identifiers*, i.e. URIs. Unfortunately, it turns out the default US-ASCII based encoding of URIs is inadequate for the task of internationalization [2].
The W3C recommends using the percent sign '%' immediately followed by two hexadecimal digits (0-9, A-F), i.e. an ISO Latin 1 code, to compensate for any non-US-ASCII character [3]. As there are only 5 special characters in Serbian Latin, the solutions solves the problem of their representation, while maintaining an acceptable level of readability, assuming the identifier doesn't contain too many instances of such characters. However, the readability of any resource identifier made up mostly of Cyrillic characters is immediately lost. Moreover, the URIs have to be encoded and decoded by individual software applications, adding additional overhead.

4.1.1. ROMANIZATION (LATINIZATION)

As every Serbian Cyrillic character has a corresponding representation in the Latin script, one workaround could be to transliterate the Cyrillic identifiers to Latin. However, as mentioned above, this works as long as the Latin version itself isn't made up of too many of non-US-ASCII symbols. Take, for instance, the Serbian word for hardener – *učvršćivač*. For most humans, it's percent-encoded variant, *u%3Fvr%3F%3Fiva%3F*, fails the readability test. One popular solution used by the Serbian online community is to take the transliteration to another level and drop the diacritical marks from Latin characters altogether, i.e. use only *C*, *Dj*, *S* and *Z*, instead of *Č*, *Ć*, *Đ*, *Š* and *Ž*. This, however, might result in ambiguity in certain cases, as shown in the table below.

Word A		Word B		Transliteration final result	
Cyrillic	Latin	Cyrillic	Latin	Word A	Word B
кучe	*kuče*	*кућe*	*kuće*	*kuce*	*kuce*

Table 1. Transliteration issues

Stripping *кучe*/*kuče* ("dog") and *кућe*/*kuće* ("houses") of diacritics in both cases results in *kuce*, a word that has a meaning of its own ("puppies").

Another romanization option could be transcription (spoken word to US-ASCII text). However, as there are no separate representations for Č and Ć using US-ASCII, the aforementioned ambiguity problem would remain. Let's take the example from above.

Word A		Word B		Transcription result	
Latin	IPA value	Latin	IPA value	Word A	Word B
kuče	/kutʃe/	*kuće*	/kutɕe/	*kuche*	*kuche*

Table 2. Transcription problems

Although the International Phonetic Alphabet (IPA)[2] values for Č and Ć are distinct, the closest representation in US-ASCII for both characters would be given by digraph *ch*. Therefore, both *kuče* and *kuće* would be transcribed to *kuche*.

4.2. INTERNATIONALIZED RESOURCE IDENTIFIERS

The only solution to overcoming all of the abovementioned deficiencies of URIs lies in using a different encoding for resource identification altogether. *Internationalized Resource Identifiers* rely on UTF-8 and are, therefore, able to represent all Unicode characters, i.e. a total of 107.000 symbols and 90 different languages.

IRIs, as good of a solution as they may seem, have yet to reach an acceptable level of support by semantic web technology tools. That is to say, most knowledge bases of today are still using URIs, and those that do support IRIs often rely on different interpretations of the standard, thereby, undermining clear and unambiguous transfer of knowledge between tools.

As the character set supported by URIs is merely a subset of the one covered by IRIs, one could assume that

[2] http://www.langsci.ucl.ac.uk/ipa/ipachart.html

conversion of the existing knowledge bases on the Semantic Web to IRIs would be a safe and straight-forward process. One problem that challenges this assumption is to detect whether a percent-encoded sequence comes from UTF-8 or legacy encoding. There is a very high probability, but no guarantee, that percent-encodings that can be interpreted as sequences of UTF-8 octets actually originated from UTF-8 [4].

Finally, similar, or even identical appearance of some characters originating from different encodings raises the old security question that gave birth to internationalized domain name (IDN) homograph attacks. In other words, occurrence of common characters such as *A* or *O*, whose visual representation is the same, or differs very little across different encodings, raises the chance of spoofing when using IRIs.

Internationalized Local Names are IRIs where the domain part is restricted to US-ASCII, while the local name can be in the Unicode range. By limiting the domain part of the IRI to ASCII, the possibility of homograph attacks is reduced. Also, the use of Unicode local names for individual resources improves readability. However, it should be noted that this trade-off between security and readability only partially solves the problem, as the possibility of spoofing for different purposes remains. Moreover, the use of internationalized local names depends heavily on tool support.

5. RDF SERIALIZATION SUPPORT

Serializing multilingual data in RDF is not a straightforward process either. Below, we take a look at the popular RDF serializations with respect to their support for internationalization.

5.1. XML-BASED SERIALIZATIONS

RDF/XML, the normative syntax for writing RDF [5], is the only RDF serialization next to RDF embedding in XHTML (RDFa), that is officially supported by the W3C. Although it is possible to use different languages/alphabets in XML content through specifying the xml:lang attribute, the same doesn't hold for parts of XML markup, such as tag and attribute names, which are limited to a predefined set of allowed characters. Although all characters of both Serbian alphabets fall into this range, both URIs and IRIs can contain a number of characters that fall outside the aforementioned range, such as % (#x0025) and & (#x0026) whose occurrences are not rare. While this doesn't represent a problem for instance data, tag names (classes, properties) in RDF/XML come from ontologies and controlled vocabularies which themselves could be multilingual, which makes serializing internationalized RDF in XML impossible in certain cases.

TriX is another serialization for expressing RDF triples in XML [6]. This experimental format aims to provide a highly normalized, consistent XML representation for RDF graphs, allowing the effective use of generic XML tools such as XSLT, XQuery, etc. However, as with RDF/XML, its expressivity is limited by that of XML.

RDFa (RDF in Attributes) is an extension to XHTML, which in turn is XML based, and, therefore, uses UTF-8 and UTF-16 for encoding [9]. RDF is embedded into XHTML pages in such way that the "%" character can be used for URIs. Moreover, IRIs are usable, too. Embedding RDF in XHTML makes the overhead bigger compared to other serialization technologies and also reduces the readability.

5.2. OTHER SERIALIZATIONS

Other, non-XML-based RDF serialization formats go beyond the RDF data model in their expressivity and flexibility when it comes to data representation.

JSON (JavaScript Object Notation), requires less overhead with respect to parsing and serializing than XML, and encodes text in Unicode, thereby making the use of IRIs possible. The percent character doesn't need special treatment; the only characters that need escaping are quotation marks, reverse solidus and the control character (U+0000 through U+001F). RDF serialization in JSON follows a non-standardized specification , but can be considered a good overall solution for internationalization.

Notation 3 (N3) is much more compact than RDF/XML, but still allows a great deal of expressiveness. It supports UTF-8 and, thus, the use of IRIs does not pose a problem. The "%" character can be used at any place that is allowed in RDF.

N-Triples, a subset of Notation 3, lack shortcuts such as CURIEs, which is why they are less readable and more difficult to create manually. What's more important for our task, is that they support only the 7-bit US-ASCII character encoding instead of UTF-8, meaning there's no support for IRIs either.

Turtle (Terse RDF Triple Language) is a subset of, and compatible with, Notation 3 and a superset of the minimal N-Triples format. It's compact, human-readable, and UTF-8 based and, therefore, makes another great solution for i18n. Turtle is also part of the SPARQL query language for expressing graph patterns.

TriG is a plain text format for serializing Named Graphs and RDF Datasets and a compact and readable alternative to the XML-based TriX syntax [8]. Its syntax is based on Turtle.

All facts considered, it is clear that N3, Turtle and TriG provide the most flexible solution for the task of internationalization with respect to expressivity, Unicode

support, readability and overhead, although JSON can serve as a worthy substitute when needed.

6. METHODOLOGY

To test how popular linked data tools cope with the task of internationalization, we propose a methodology that should make a good enough of a challenge for any such application.

6.1. OPERATIONS

We must ensure the methodology covers the entire data lifecycle within each individual tool. That means testing the data on input, modification and output. We, therefore, propose a set of essential actions that most tools have in common, that should address each of the lifecycle stages.

Importing - Every tool must be able to handle non-US-ASCII data on input. To make sure we cover all aspects of a complex multilingual RDF model, the data is provided in one of the i18n-ready (i.e. "safe") formats described in section 5.

Editing - The data is dealt with as per tool's intended purpose. This means modifying the existing data (e.g. editing a resource by adding Unicode characters, then saving the modified data in the knowledge store), creating new projects, adding new resources or metadata with non-US-ASCII characters, etc.

Exporting/Retrieval - Each application must decode and encode data properly, for on-screen display or serialization to available formats, respectively. The modified data is exported to each of the formats described in section 5 (assuming the tool supports the selected format), including the non-i18n-optimal ones. The tool must be aware of whether the chosen output serialization is unsafe, i.e. if data to be exported contains characters that are not supported by the chosen format. Therefore, we highlight the possibility of incorrect serialization interpretations by individual linked data applications. If a tool provides a SPARQL endpoint or a similar type of interface for data retrieval, we query the backend to get the data.

6.2. EVALUATION CRITERIA

To assess how each tool deals with the task of internationalization, we need to look at three aspects of the resulting data after each of the operations - view correctness, and model and instance data correctness.

View correctness (on screen display/output) - If the tool or the module being tested provides on screen display of knowledge base resources, the resulting data must be presented to the user without loss of integrity.

Model and data correctness - Every linked data application must ensure that both the underlying model and the data will correspond to the original input dataset on import, and to the original knowledge base on export. Such a guarantee is a prerequisite for safe data transfer between different semantic web tools.

Serialization implementation correctness - Every tool must implement offered serialization options according to standard specifications in order to ensure safe data exchange between individual linked data applications.

6.3. EVALUATION

Tool assessment with respect to the aforementioned three aspects is generalized to the following two tests:

- **View evaluation** - We inspect (visually) the correctness of imported/modified data.
- **Model and data evaluation** - We query the underlying data store / SPARQL endpoint after each operation and compare the retrieved triples with the master dataset.

In case an operation fails due to an application's inability to handle the contents of the test dataset, and such failure hampers further testing, we fallback to a simplified test dataset(e.g. by removing any characters that caused the problem) and move on with the evaluation.

6.4. TEST DATA

An excerpt from a sample test dataset in Turtle syntax, combining both Latin and Cyrillic characters, as well as (percent-encoded) URIs and IRIs, is given below. In order to make the task more challenging for the tools, we test classes, properties, instances and literals.

```
res:ĐorđePetrović
 rdf:type cls:Čovek;
 prop:srednjaŠkola
   "Mašinska Tehnička Škola 15. Maj"@sr;
 prop:mestoRođenja res:Niš.

res:ИванаМијовић
 rdf:type cls:Човек;
 prop:средњаШкола
   "Машинска Техничка Школа 15. Maj"@sr;
 prop:местоРођења res:Ниш.

res:NikolaPetrovi%C4%87
 rdf:type cls:%C4%8Covek;
 prop:ro%C4%91ak
   res:%C4%90or%C4%91ePetrovi%C4%87;
 prop:mestoRo%C4%91enja res:Ni%C5%A1.
```

It should be noted, again, that %-encoded sequences are in the character range for IRIs. They are explicitly allowed in local names, and are not decoded during processing. For instance, *res:Ni%C5%A1* in the above example designates the IRI *http://example.org/res#Ni%C5%A1* and not IRI *http://example.org/res#Niš* (corresponding to another resource in the same example).

Moreover, to ensure safe transfer of data between components, we also need to test whether individual serialization formats are interpreted properly by each of

them. Therefore, we also serialize the above RDF snippet using an unsafe format - RDF/XML in order to feed it to our linked data applications. A tool that interprets the XML standard according to its specification should reject such a file on import.

7. EVALUATION RESULTS

The testing environment[3] is based on the latest version of the LOD2 Tool Stack. The respective versions of individual Stack components at the time of testing are given in the table below.

LOD2 Stack component	Version
Virtuoso	06.01.3127
OntoWiki	0.9.7
Silk	2.5.4
PoolParty	n/a
LODRefine	trunk [r0001]

We describe the test for each application and focus mostly only the problems encountered..

7.1. VIRTUOSO

Virtuoso is a knowledge store and virtualization platform that transparently integrates Data, Services, and Business Processes across the enterprise. The open source data integration server and the highly efficient and scalable RDF triple store implementation in Virtuoso are the basis for the knowledge store component in the LOD2 Stack.
In our test, we tried uploading the test data directly to the Virtuoso backend, then checking data integrity by attempting to retrieve it using the provided SPARQL endpoint.

Import - The only issue we've come across was during the upload phase, when the test dump file was rejected as invalid data due to percent-encoded resources not being enclosed in angle brackets.

7.2. ONTOWIKI

OntoWiki is a tool providing support for agile, distributed knowledge engineering scenarios. OntoWiki facilitates the visual presentation of a knowledge base as an information map, with different views on instance data. It enables intuitive authoring of semantic content, with an inline editing mode for editing RDF content, similar to WYSIWYG for text documents.

We uploaded the N3/Turtle serialized test dataset, edited newly created knowledge base with respect to each of the test encodings, and, finally, exported the data using each of the available i18n-ready serialization formats. As described earlier, we also tried creating, as well as exporting a knowledge base using an unsafe format, i.e. RDF/XML.

[3] http://fraunhofer2.imp.bg.ac.rs/lod2demo

Import - OntoWiki rejected the N3 file due to percent-encoded characters in one of the resources, but accepted the same file when Turtle was specified as file format.

Export - During the export phase to JSON, non-US-ASCII data in was serialized as code-points (as in N-Triples), instead of UTF-8. We would also like to note that OntoWiki doesn't warn the user of unsafe export option when RDF/XML is chosen as desired export format, and outputs an empty knowledge base.

7.3. SILK

The Silk framework is a tool for discovering relationships between data items within different linked data sources. Data publishers can use Silk to set RDF links from their data sources to other data sources on the Web.

We tried creating a new project, selecting a couple of sample files with Unicode data and linking them.

Import - A source file with valid N3 syntax was rejected due to a percent-encoded character in one of the identifiers. Silk did, however, also reject invalid RDF/XML syntax (invalid characters in tag names).

Edit - A project cannot be created if it contains ISO 8859-2 or 15924 characters in its name. Also, IRI prefixes are not encoded, nor decoded properly.

Export - Silk couldn't generate any links, as IRIs were not decoded properly. Therefore, no results could be retrieved with non-ASCII identifiers (standard URIs worked as expected).

7.4. POOLPARTY

PoolParty is a thesaurus management system and a SKOS editor for the Semantic Web including text analysis functionalities and linked data capabilities. The system helps to build and maintain multilingual thesauri providing a simple user interface and a number of semantic services.

In our test, we created a simple SKOS concept schema, and enriched it with a number of Unicode concepts and metadata.

Edit - When creating a project, if the title/subject field uses non-ASCII characters, a project gets created, but not loaded automatically.

Export - Latin and Cyrillic entries were not properly decoded in certain places. It is worth noting that PoolParty can successfully serialize international data in XML-based formats as it uses specially-encoded resource identifiers.

7.5. LODREFINE

LODRefine is a LOD-enabled version of OpenRefine (formerly Google Refine), a power tool for working with

messy data, cleaning it up, transforming it from one format into another, extending it with web services, and linking it to data sources such as Freebase[4] or DBpedia[5].

We created an OpenRefine project using the test data, then tried modifying Cyrillic, Latin and percent-encoded columns and cell data both manually and automatically by reconciling against one of the available reconciliation services.

Import - LODRefine has problems parsing any RDF (N3/Turtle or RDF/XML data; the tool hangs on import), so we fall back to JSON.

Editing - Reconciliation works against "absolute" values of strings, i.e. a Cyrillic string will not be transliterated to Latin, and a percent-encoded characters will not be converted to their Unicode values, thereby resulting in no matches (which can be considered expected behavior).

Export - Exporting to RDF (N3/Turtle or RDF/XML) resulted in an almost empty file. That is, an output file was created every time, but contained only prefixes.

8. CONCLUSION

We have seen the obstacles linked data tools face on their way to providing full support for Western Balkan languages, proposed workarounds and showed a general, yet robust, evaluation methodology. The output of the thorough evaluation gives us an insight into the state of i18n readiness of some of the most popular semantic web tools, with regard to importing, modifying and exporting, while preserving accuracy and consistency of data that contains non-US-ASCII characters, as it is transferred between individual tools.

We have seen that IRIs represent a solution to many common problems with Serbian alphabets and their respective encodings, and that N3 and its derivations are the most flexible solutions for data serialization. The presented evaluation results call for better implementation/interpretation of different RDF serialization standard specifications, more robust fallback mechanisms and improved overall support for i18n. Moreover, they can also serve as internationalization support pointers/requirements for future work on our way to fully i18n-ready semantic web tools that will support the entire lifecycle of multilingual linked data.

ACKNOWLEDGEMENTS

The research presented in this paper is partly financed by the European Union (FP7 LOD2 project, Pr. No: 257943), and in part by the Ministry of Science and Technological Development of Republic of Serbia (SOFIA project, Pr. No: TR-32010).

[4] http://www.freebase.com/
[5] http://dbpedia.org

REFERENCES

[1] Commission of the European Communities.. *Western Balkans: Enhancing the European perspective*, 2008. Retrieved from http://ec.europa.eu/enlargement/pdf/balkans_communication/western_balkans_communication_050308_en.pdf
[2] Sören Auer, Matthias Weidl, Jens Lehmann, Amrapali J. Zaveri, Key-Sun Choi. *I18n of Semantic Web Applications*. Lecture Notes in Computer Science, Volume 6497, 1-16, 2010.
[3] W3C. *Universal Resource Identifiers: Recommendations*. Retrieved from http://www.w3.org/Addressing/URL/4_URI_Recommentations.html
[4] Martin J. Dürst: *The Properties and Promises of UTF-8*. Presented at the 11th International Unicode Conference, San Jose, CA, Sept. 1997. Retrieved from http://www.ifi.unizh.ch/mml/mduerst/papers/PDF/IUC11-UTF-8.pdf
[5] Frank Manola, Eric Miller. *RDF Primer*, 2004. Retrieved from http://www.w3.org/TR/rdf-primer/
[6] Jeremy J. Carroll, Patrick Stickler. *RDF Triples in XML*, 2004. Retrieved from http://www.hpl.hp.com/techreports/2003/HPL-2003-268.pdf
[7] Tim Berners-Lee. *Notation 3 Logic*, 2005. Retrieved from http://www.w3.org/DesignIssues/Notation3.html
[8] Chris Bizer, Richard Cyganiak. *The TriG Syntax*, 2007. Retrieved from http://wifo5-03.informatik.uni-mannheim.de/bizer/trig/
[9] W3C. *RDFa 1.1 Primer*, 2012. Retrieved from http://www.w3.org/TR/rdfa-primer/