Decentralization Rules: Linking Solid Pods in Different Vocabularies using Notation3

Dörthe Arndt[1] and William Van Woensel[2]

[1] Computational Logic Group, TU Dresden, Germany
[2] NICHE Research Group, Dalhousie University, Canada
doerthe.arndt@tu-dresden.de

1 Background

The decentralization of personal data on the Web involves each user having full control and ownership over their own data. In this setting, individuals grant access to their data to Web apps of their own choosing, effectively decoupling Web apps from personal data storage. A Solid pod [2] is a decentralized source of personal individual data; using Web Access Controls (WAC), an individual can choose which parties on the Web can access their pod. In Solid, a WebID IRI is a unique identifier for a person and is de-referenced to their personal data within a Solid pod. By further using Linked Data principles, Solid pods contribute to a decentralized Social Knowledge Graph—using vocabularies such as FOAF, OSLO, and HL7 FHIR, a profile can indicate salient personal relationships between denotations of WebIDs described within other pods.

At the same time, however, decoupling Web apps from the representation of personal data will invariably lead to mismatches. Each pod may utilize its own vocabulary to represent personal data, whereas Web apps may hard-code support for a different vocabulary. We point out that this is not necessarily a problem. Many profile vocabularies will have different contexts of use, and thus focus on other types of personal data: FOAF describes social media profiles (e.g., personal relations, interests), OSLO represents citizenship information (e.g., nationality, marital status), and HL7 FHIR encodes patient profiles (e.g., emergency contacts, contact info). Mapping expressions between these vocabularies, or ontologies, involves ontology matching, mediation or alignment, which has been widely studied [7]. Regarding the representation of alignments, the Simple Standard for Sharing Ontological Mappings (SSSOM) [8] focuses on associating metadata (e.g., confidence) and offers an easy table-based format for ontology mediation; the Expressive Declarative Ontology Alignment Language (EDOAL) [6] offers expressivity to deal with more complex ontology alignments.

We show the utility of Notation3 (N3) [4,3] for representing executable, complex, knowledge-based ontology alignments within a decentralized environment. Ontology alignments in N3 can capture complex patterns within the source and target ontology, and can leverage a rich set of string, math, and logical built-ins. Alignments are written based on people's best knowledge of both ontologies, such as common or best-practice design patterns. Further, as Solid pods rely on Linked Data principles, relevant information will often be found in other, decentralized pods. By leveraging the decentralized reasoning offered by N3 (`log:semantics` builtin), alignments can follow and de-reference found WebIDs to other Solid pods, and thus offer a more complete mapping (as shown in our use case). Finally, N3-based mappings can be easily executed using a standard N3 reasoner such as EYE [5]. Below, we describe a case where N3 Logic is used to represent and execute mappings between OSLO and HL7 FHIR in a decentralized setting. All examples mentioned below can be found in the git repository for this paper at `https://github.com/doerthe/oslo-to-fhir`.

2 Use Case: From OSLO to FHIR

Recently, the Flemish government decided to decentralize their citizen data infrastructure according to the SOLID principles, which will involve providing a Solid pod for each of their citizens [1]. Our git repo provides an example representation of a citizen data pod (`pieter_oslo.ttl`) using the OSLO vocabulary, which is a standard developed by the Flemish government for representing citizen information.

Of course, personal data from this pod can also be relevant for applications aside from government. If the citizen travels abroad and needs medical care—for example in a hospital—he will have to provide personal information that can likely already be found in their Solid pod; such as name, address, birth date and phone number. The hospital will store that data in an Electronic Health Record (EHR), together with medical information. In our example, we assume the EHR system supports HL7 FHIR, a well-known standard for exchanging EHR data. Our git repo provides an example representation of a patient health record in HL7 FHIR (`pieter_fhir.ttl`). In order to re-use data from their citizen Solid pod for entry into the EHR, the individual provides the EHR system with WAC access rights as well as their personal WebID.

To map the citizen data (OSLO) to a patient record (FHIR), the EHR system utilizes a set of ontology alignments, or translation rules, written in N3 (e.g., the OSLO ontology may indicate such alignment sets). Our git repo includes these translation rules, i.e., for profiles from OSLO to FHIR (`oslo-to-fhir.n3`). These N3 rules can be executed locally using EYE [5], or remotely using our N3 online editor (`https://github.com/william-vw/n3-editor-js`). We invite the reader to try out and modify N3 code pre-loaded in the editor at `http://ppr.cs.dal.ca:3002/n3/editor/s/zThaCLVD`, which includes the source OSLO citizen profile and OSLO-FHIR translation rules. Using N3, we implement the execution of ontology alignments as a reasoning problem. N3 allows stating classical logical rules with an antecedent and a consequence that may contain universal variables. If we can unify the antecedent with a set of triples, using a variable substitution, we can produce the consequence applying the same substitution. For the use case, it is crucial that N3 supports existential rules producing blank nodes, which are omnipresent in many standards such as FHIR.

An simple (existential) translation rule maps the gender of the patient (marked by "#gender" in our file). The EHR system declares the citizen as a patient before the translation process, whereas their gender is specified in their OSLO pod. When executing this rule using EYE (pre-loaded at `http://ppr.cs.dal.ca:3002/n3/editor/s/WsVzDGqG`), the corresponding FHIR representation is produced. This example illustrates that writing these kinds of translation rules is straightforward, even for arbitrarily nested structures (see also other translation rules in git repo): we specify the pattern we encounter and the pattern we want to produce, possibly adding conditions for more selective alignments (e.g., math comparisons using N3 built-ins). This intuitive structure also allows for ad-hoc adjustments of the rules in case the input's structure slightly differs from expectations, or in case additional assumptions could be made.

N3 offers a variety of built-ins, e.g., to write conditional alignments or to operate on strings; we use `string:concatenation` to combine street and house number to an address line (pre-loaded at `http://ppr.cs.dal.ca:3002/n3/editor/s/qNiZOC2s`). In particular, the N3 built-in `log:semantics` enables reasoning within a distributed environment as it is able to follow and de-reference links. The builtin accesses the content of the link given as a subject, and provides the retrieved N3 code as an object graph. In our example (pre-loaded at `http://ppr.cs.dal.ca:3002/n3/editor/s/yvfMdm3B`), the translation rule searches the OSLO citizen profile for the patient's spouse, and uses `log:semantics` to de-reference the listed WebID to their online Solid pod. Next, the rule extracts the spouse's contact info from the retrieved OSLO graph, using the `log:includes` built-in which performs pattern matching on N3 graphs.

In this paper, we illustrated how N3 can be used to formulate and execute ontology alignments within a decentralized environment, by writing and reasoning over N3 translation rules with expressive builtins. Our use case illustrates that, to fully realize the opportunities of decentralized Solid pods, there is a need for translation between different vocabularies or ontologies. Future work involves larger-scale experiments and using different vocabularies, as well as studying mechanisms for distributing and sharing translation rules.

References

1. Flanders-Inrupt, `https://inrupt.com/flanders-solid`
2. SOLID, `https://solidproject.org/`
3. Arndt, D., Van Woensel, W., Tomaszuk, D.: Notation3: Draft Community Group Report (2021), `https://w3c.github.io/N3/spec/`
4. Berners-Lee, T., Connolly, D., Kagal, L., Scharf, Y., Hendler, J.: N3Logic: A logical framework for the World Wide Web. Theory and Practice of Logic Programming (3), 249–269 (2008). https://doi.org/10.1017/S1471068407003213
5. De Roo, J.: Euler Yet another proof Engine - EYE, `https://josd.github.io/eye/`
6. Euzenat, J., Zimmermann, A.: Expressive alignment language and implementation (06 2007)
7. Liu, X., Tong, Q., Liu, X., Qin, Z.: Ontology matching: State of the art, future challenges, and thinking based on utilized information. IEEE Access **9**, 91235–91243 (2021)
8. Matentzoglu, N., Balhoff, J.P., Bello, S.M., et al.: A simple standard for sharing ontological mappings (SSSOM). CoRR **abs/2112.07051** (2021), `https://arxiv.org/abs/2112.07051`