ActiveRaUL: Automatically generated Web Interfaces for creating RDF data

Anila Sahar Butt Armin Haller Shepherd Liu Lexing Xie
Australian National University, Canberra, Australia
CSIRO Computational Informatics, Canberra, Australia

Introduction

The amount of semantic Web data is still many orders of magnitude smaller than the World-Wide-Web. One of the barriers for semantic Web novices to create machine-readable data is the lack of easy-to-use Web publishing tools that separate the schema modelling from the data creation. We present **ActiveRaUL**, a Web form-based user interface that particularly supports users inexperienced in semantic Web technologies in creating RDF data. These Web form-based user interfaces in ActiveRaUL can be automatically generated from any arbitrary input ontology.

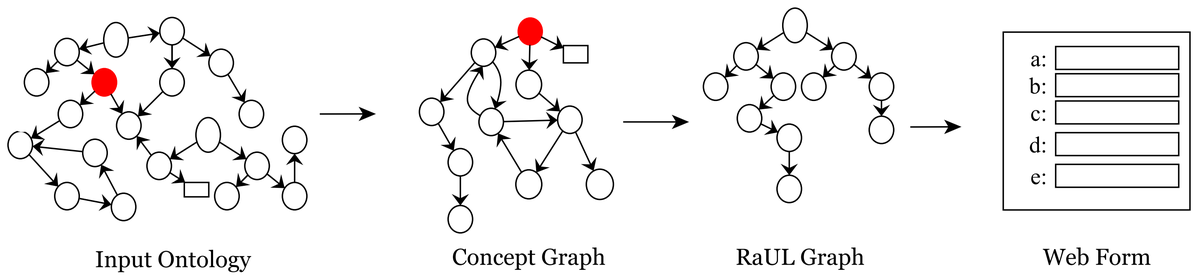
Challenges:

- Mismatch between **graph based data model** of an ontology and the **tree based data model** of Web forms.
- Incompatibility of the **triple model** of RDF to a **key-value pair** of Web form.
- Creating **objects** for object properties from Web form fields rather than **standard datatypes**.
- **Reuse of individuals** across different Web forms.

Solutions:

- **Extract relationships** among the concepts form arbitrary input ontology.
- **Avoid cycles** of an RDF graph by removing *inferable properties* of the ontology.
- **Use RaUL Ontology** - an `RDF representation of Web form elements-to:`
  - o map `acyclic RDF graph` to `tree based data model` of Web forms
  - o map `RDF triple` to `key-value pair` of Web forms
  - o encode information about `objects` creation from Web form fields

Method



Input Ontology Concept Graph RaUL Graph Web Form

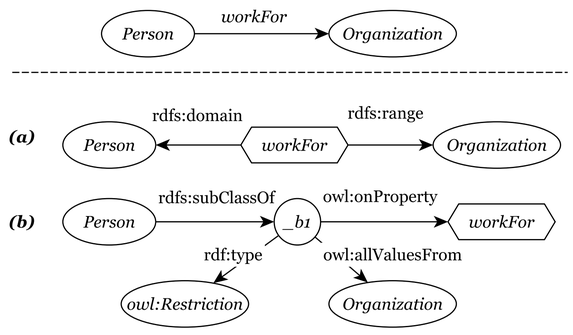
- A user provides input ontology through ActiveRaUL web interface or API and selects a concept - **the concept node**.
- ActiveRaUL constructs a **concept graph** which is composed of relationships of the concept node.
- ActiveRaUL creates a **RaUL graph** for the **concept graph**. The **RaUL graph** is mapped to a Web form.

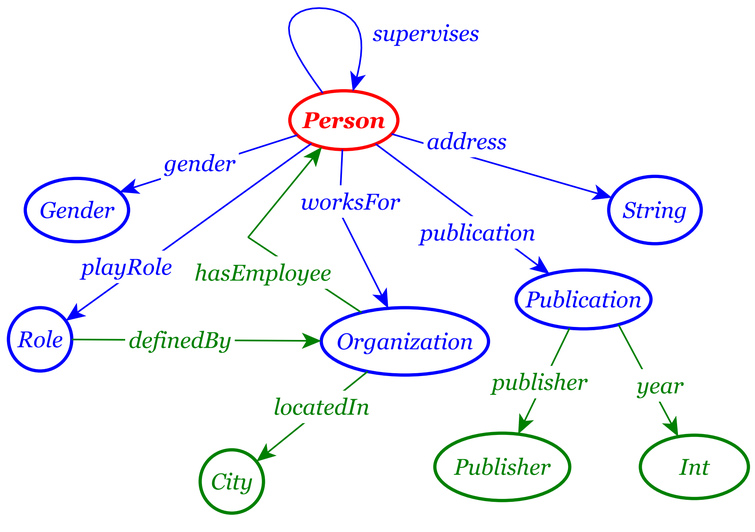
Concept Graph

Concept Graph CG(ν): A directed graph $CG(\nu) = < U, E, \nu >$ where ν is the `concept node` and $\forall\ u \in U$, there exists a path $\overset{\nu \rightsquigarrow u}{\pi}$ of length l between ν and u.

Constructing a Concept Graph CG(ν): In the **ontology** relationships amongs concepts are model through different **physical models** using RDFS and/or OWL modeling languages. The example physical models for "`Person worksFor Organization`" relation are shown.

- We adopt a **breadth first** approach to incrementally extract **associations** (relationships) of different lengths for the `concept node` modeled through different physical models.
- For the **Person** class of an example ontology, we extract associations of length $l = 1,2,\ldots,n$. Where 'n' is maximum length of any association for Person.

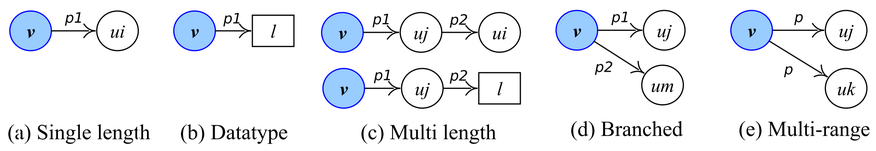




RaUL Web Form

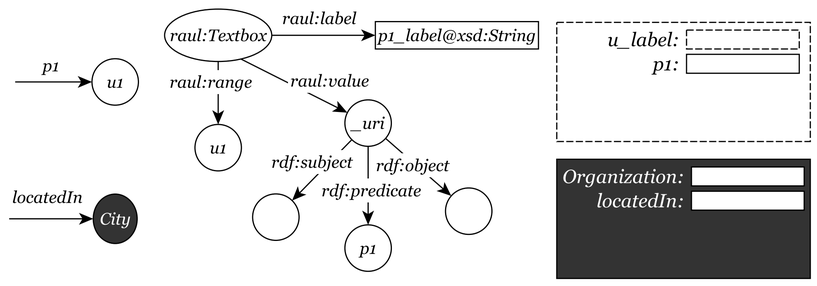
RaUL Graph μ(ν): is RaUL based mappings for CG(ν), where for all unique $\overset{\nu \rightsquigarrow u}{\pi} \in$ CG(ν) a mapping $\mu : \overset{\nu \rightsquigarrow u}{\pi} \rightarrow$ RaUL is one or more valid RaUL widget elements for $\overset{\nu \rightsquigarrow u}{\pi}$.

Mapping the Concept Graph CG(ν) to a RaUL Graph: Befor mapping to a RaUL graph, some **cycles** and **redundant associations** are removed from the concept graph. **Base property paths** (shown below) are identified that alone or together compose an association in any concept graph. A **RaUL mapping** is defined for each of the base property path.

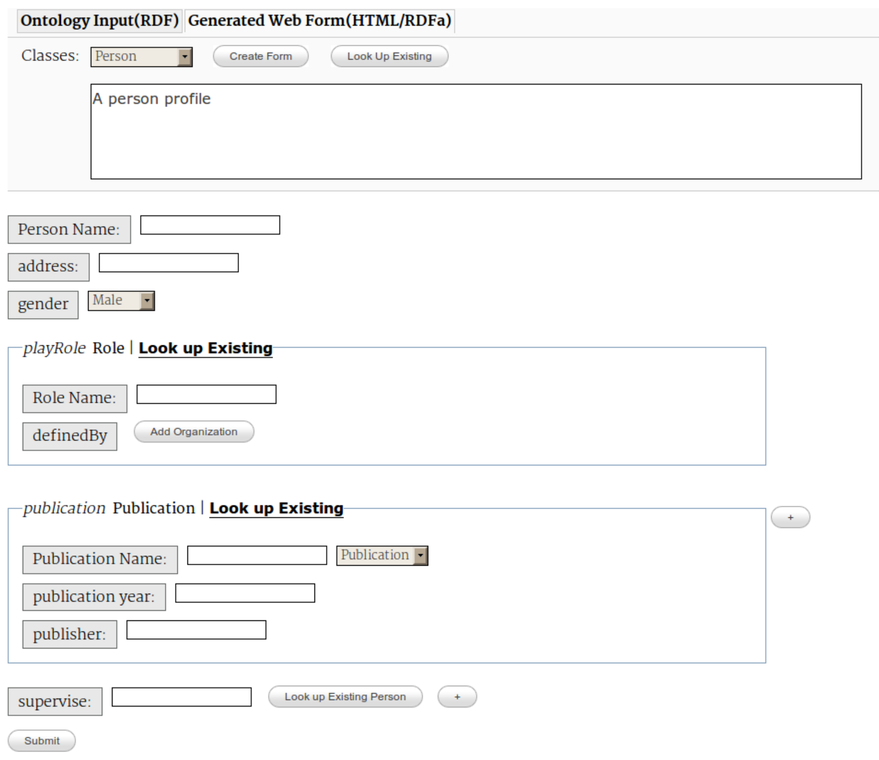


(a) Single length (b) Datatype (c) Multi length (d) Branched (e) Multi-range

Each association is mapped to the corresponding RaUL mapping/s based on its type of property paths. Each RaUL mapping is then mapped to a Web form element.



Example Web Form



Evaluation

We validate our approach in a user study based on use cases developed by the W3C Semantic Sensor Network (SSN) Incubator group. The users created RDF data based on the SSN ontology with **ActiveRaUL** generated user interfaces and **WebProtege**. They then rated **ActiveRaUL** compared to **WebProtege** for the **creation of RDF data** on a System Usability Scale (SUS).
An Average rating for both the systems on **SUS** is shown here:

Summary of results:

- **Correctness:** ActiveRaUL 91% vs WebProtege 82% .
- **Efficiency:** ActiveRaUL between 27% to 56% faster than WebProtege.
- **Satisfaction:** ActiveRaUL 72.1/100 compared to WebProtege 32.5/100 SUS rating