CS549 Distributed Information Systems.

Classwork: Building and processing an ontology in OWL.

Deadline: 12.00noon on 04/04/2022.

THIS IS AN INDIVIDUAL TASK AND MUST BE ACCOMPLISHED WITHOUT COLLABORATION, COLLUSION OR THE SHARING OF SOLUTIONS. FAILURE TO FOLLOW THIS INSTRUCTION WILL RESULT IN DISCIPLINARY ACTION BEING TAKEN.

1) Aim Of The Assignment:

The aim of this exercise is to build an ontology to represent part of the business of the University of Strathclyde's Students Union. (https://www.strathunion.com/) using RDF/RDFS/OWL.

2) Learning Outcomes:

Successful completion of this assignment will indicate your understanding of the design of a distributed global application and your appreciation the range of issues involved in the development and implementation of such systems.

3) Task:

The exercise should be carried out by examining the Students Union Web pages and identifying elements that are suitable for representation in an ontology. Your solution does not need to incorporate all the elements that are included on the Students Union Web pages but you do need to include enough to satisfy the criteria listed "Deliverables (a)" below. You should restrict your work to the Students Union domain but if necessary, it would be permissible to add content to the domain to carry out this task providing that an appropriate argument is presented in the critique.

It is important that the ontology contains multiple instances of classes, modifiers, relations and definables. For relations, you should show which characteristics are necessary and give the definition of definables.

As a guide, you should be aiming for three or four main classes with subclasses, some properties (both object and datatype), instances and definables. You should incorporate at least one example of an element imported from an external ontology.

You also need to formulate at least **five** competency questions that can be answered by the ontology.

You should use Protégé (http://protege.stanford.edu/) to build the OWL representation of the ontology.

The following SPARQL queries should be executed against your ontology:

(a) traverse at least three predicates in the graph and involve at least one predicate from an imported ontology.
(b) use a FILTER
(c) carry out an aggregate operation using a 'group by' clause.
(d) check for the existence of a fact in your collection.
(e) generate RDF output from a query result.

To test the SPARQL queries, you can use Protégé, an endpoint such as Rasql (http://librdf.org/query//) or a stand-alone tool such as Twinkle (http://www.ldodds.com/projects/twinkle/).

You can use Twinkle by placing the file containing the data to be queried in your WWW directory. Make sure that the file has the following permissions:
- r w x r - - r - -
Use a FROM clause in the SPARQL query that makes it use data from
FROM
https://personal.cis.strath.ac.uk/first_part_of_your_email_address/index.rdf

(assuming that the data file is called index,rdf).

You also need to be able to demonstrate a description logic (DL) query over the Protégé representation of your ontology. To do this, you need to:

- use the 'DL Query' tab in Protégé,
- formulate an appropriate DL query in the Query (class expression) box
- select the appropriate element(s) from the 'Query for' panel on the right,
- start a reasoner from the Reasoner menu,
- take a screenshot of the query results panel to include in your submitted report.
- The DL query needs a least TWO 'ands' and TWO qualifiers ('some' etc) and must produce a result.

4) Deliverables:

The deliverables are:

a) A single .pdf file containing:

i) A description of the purpose of the ontology – what kinds of questions should the ontology be able to answer?

ii) A table listing the classes, modifiers, relations and definables and a diagrammatic representation of the ontology. Use a diagrammatic format similar to that introduced in the lectures during Weeks 3 and 4. Do not export the diagram from Protégé. Give a definition of each definable. For relations, list the range, domain and characteristics.

iii) At least **three** extracts of the RDF/RDFS/OWL code that illustrate features of the ontology together with a description of why these are significant.

iv) The narrative question that is answered, text and output of SPARQL queries that process your data structure and produce results that match those requested in Section 3 (a)-(e).

v) A DL query that can be run on your ontology by Protégé, a screenshot of the output as described above and an explanation of the meaning of the output and the reasoning process that produces it.

vi) A critique of your design including a consideration of the extent to which the ontology matches the problem domain and the decisions that you have made in producing the ontology. Also describe alternative approaches and the reasons why a particular implementation was chosen (700-900 words).

b) An .owl file containing the entire RDF/RDFS/OWL representation of this ontology serialised in RDF/XML syntax.

5) Submission:

Your assignment should be submitted on the MyPlace link by the deadline shown at the beginning of this document. All work is evaluated for originality by Turnitin.

6) Marking Criteria:

Marking criteria	Weighting
(i) Description of the purpose of the ontology.	15%
(ii) Description & diagram of the ontology.	20%
(iii) RDF/RDFS/OWL.	10%
(iv) SPARQL queries.	25%
(v) Description Logic Query	5%
(vi) Critique.	20%
(vii) **Strict** adherence to the Formatting requirements and page limitation.	5%

7) Formatting requirements and page limitation:

Two files only must be submitted:
a) A file containing the RDF/XML serialisation of your ontology
b) A single .pdf file containing only the items listed under 'Marking Criteria'. The .pdf file must have a maximum of five pages (A4, single column, portrait orientation, 11pt font, no coversheets etc, all black text on a white background, all pages numbered, minimum 2cm margins). All pages must be numbered. The solution must be typed (not hand written)[1]. Items in paragraph 'Marking Criteria' above must be combined into this document in the order given in the paragraph. The document must not be zipped. The ontology diagram must be in portrait format and not hand drawn. The ontology diagram (part of Deliverable (ii)) and the DL query output (part of Deliverable (v)) and may be presented as a screenshot but no other screenshots or other elements that cannot be processed by Turnitin must be included. If a Turnitin error is returned when you upload the document (eg "Turnitin has returned an error with your submission: Your submission must contain 20 words or more"), it means that the .docx file contains text as images. You must OCR (optical character recognition) the document to produce text and resubmit it so that it can be processed by Turnitin. SPARQL queries must be formatted in the style shown in Lecture 14.

8) Proportion of overall marks for CS549:

This exercise is worth 70% of the overall marks for CS549.

John Wilson
February 2022.

[Ends.]

[1] If you have a requirement that would make this format difficult for you to read, mail John.N.Wilson@strath.ac.uk for further instructions.