Question 3

Web Ontology Language (OWL) was created as a standard by the Web Ontology Working group, which in turn was formed by the World Wide Web Consortium (W3C). Grau et al. (2008) OWL was created because of a number of limitations in expressiveness was found in RDF and RDFS. The reason RDFS was not simply extended, was because of "... the trade-off between expressive power and efficient reasoning" Antoniou and Van Harmelen (2009). In other words, if the RDFS was made more expressive it would have had to be at the expense of being able to write efficient reasoners for it.

There are 3 language variants of OWL. Owl Lite, OWL DL and OWL Full. Where OWL DL has more expressive power than OWL Lite and OWL Full in turn has more expressive power than OWL DL. (Consortium, 2004)

OWL uses various commands from RDFS. For instance *rdfs::comment* is used to store comments on ontology elements, where as *rdfs::label* is used to add human readable names to elements in the ontology. (Horridge et al., 2011) Additionally *rdfs::subClassOf* is used to define whether one class is a subclass of another class. (Antoniou & Van Harmelen, 2009)

There are a number of useful features that cannot be expressed using RDF and RDFS. For instance:

- Whether 2 classes are disjoint or mutually exclusive. This can be accomplished using *owl:disjointWith* in OWL.

- Cardinality cannot be restricted. For example we cannot say that a store has only one owner. OWL uses *owl::cardinality* to express this concept. OWL also has *owl::minCardinality* and *owl::maxCardinality* to express a minimum or a maximum cardinality when the cardinality encompasses a range. For example lecturer must have a minimum of 5 students and a maximum of 400.

- New classes cannot be expressed as the union, intersection and/or complement of other classes. *owl::complementOf* can be used to declare a class to be the compliment of another class. *owl::unionOf* is used to declare a class the uniion of a set of classes and *owl:intersectionOf* is used to declare a class to be the intersection of a set of classes.

- The scope of properties cannot be made local. The operation rdfs:range defines the scope of a property for all classes. In OWL, *owl::Restriction* in conjunction with *owl::allValuesFromm* *owl::someValuesFrom* and *owl::hasValue* can be used to place various restrictions on classes.

- The RDFS cannot express whether a property is transitive or symmetric. This is solved in OWL using *owl::TransitiveProperty* and *owl::SymmetricProperty.*
- RDFS cannot express when one property is the inverse of another property. Owl uses *owl::inverseOf* for this functionality.

(Antoniou & Van Harmelen, 2009)

In conclusion, we can see that OWL was created to provide features that RDFS could not or it would not have been feasible to provide.

References

Antoniou, G., & Van Harmelen, F. (2009). Handbook on Ontologies. In S. Staab & R. Struder (Eds.), *Handbook on ontologies* (Second Edition ed., pp. 91–110). Springer. Retrieved from `http://link.springer.com/10.1007/978-3-540-92673-3` doi: 10.1007/978-3-540-92673-3

Consortium, W. W. W. (2004). *Owl web ontology language semantics and abstract syntax* (Vol. 10).

Grau, B. C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P. F., & Sattler, U. (2008). OWL2: The next step for OWL. *J. Web Semant.*, *6*(4), 309–322.

Horridge, M., Knublauch, H., Rector, A., Stevens, R., Wroe, C., Jupp, S., . . . Brandt, S. (2011). A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3.