

# Σημασιολογικός Ιστός και Ευφυείς Εφαρμογές

7ο Εξάμηνο, 2023 – 2024

## Εργασία B: OWL

### 1 Στόχος

Στόχος της εργασίας είναι να επεκτείνετε το RDFS μοντέλο της πρώτης εργασίας χρησιμοποιώντας τις δομές και την σημασιολογία της **OWL**. Προτείνεται η χρήση του [Protégé](#).

### 2 Απαιτήσεις

Η μοντελοποίηση θα πρέπει να κάνει όσο το δυνατόν περισσότερη (και σωστή) χρήση των δομών και της σημασιολογίας της OWL. Για παράδειγμα, ισοδυναμία κλάσεων, ιδιοτήτων και αντικειμένων, αντίστροφες ιδιότητες, συναρτησιακές και αντίστροφες συναρτησιακές ιδιότητες, συμμετρικές και μεταβατικές ιδιότητες, περιορισμοί κλάσεων (αναγκαίοι και ικανοί), τομή, ένωση, κτλ. Μπορείτε να χρησιμοποιήσετε για reasoner είτε τον Hermit είναι τον Pellet (μπορείτε να περάσετε τον Pellet από το μενού του Protégé File-> Check for plugins, επιλέγετε τον Pellet από την λίστα και επιλέγεται install. Και οι δύο reasoners είναι καλοί, ο Pellet διαχειρίζεται καλύτερα τα hasValue restrictions που αφορούν strings.

### 3 Υλικό προς παράδοση

- Το OWL μοντέλο (κλάσεις, ιδιότητες και αντικείμενα).
- Το [έγγραφο με τις προδιαγραφές](#).

### 4 Οδηγίες υποβολής

- Η εργασία θα πρέπει να γίνει σε **ομάδες των δύο ατόμων (ίδιες με αυτές της πρώτης εργασίας)**. Αν για οποιοδήποτε λόγο αυτό δεν είναι εφικτό, παρακαλώ να επικοινωνήσετε με τον διδάσκοντα.
- Η υποβολή των εργασιών να γίνει στο elearning από ένα μέλος της ομάδας. Τα αρχεία που θα υποβληθούν να είναι rar/zip με όνομα: **<AEM1>-<AEM2>-OWL.zip**
- Προθεσμία υποβολής: **18/01/2024**

### 5 Αξιολόγηση

Η βαθμολογία της εργασίας θα βασιστεί στον αριθμό, την ορθότητα και πληρότητα των δομών της OWL που έχουν χρησιμοποιηθεί, στην εκφραστικότητα του μοντέλου καθώς και στην ποιότητα/πληρότητα της αναφοράς / documentation.

- Χρησιμοποιήστε τα rdfs:comment, rdfs:label, κτλ. για να εισάγετε περιγραφές, σχόλια, κτλ. στο μοντέλο.
- Το μοντέλο θα ελεγχθεί για σφάλματα μοντελοποίησης (λάθη όπως is-a vs. part-of, class vs. instance, σημασιολογικά λανθασμένα domain και ranges, κτλ.), καθώς και για σημασιολογικές ασυνέπειες (inconsistencies) που τυχόν υπάρχουν, όπως αυτές ορίζονται στην OWL και θα παρουσιαστούν στις διαλέξεις.

## Κριτήρια Αξιολόγησης

Στόχος είναι να βασιστείτε στην οντολογία RDFS που έχετε δημιουργήσει (όχι να ξεκινήσετε από την αρχή, δείτε τις σημειώσεις στο τέλος αυτού του αρχείου) και να την επεκτείνετε με OWL, ορίζοντας όπου χρειάζεται **νέες κλάσεις, ιδιότητες και αντικείμενα** για να χρησιμοποιήσετε και να αναδείξετε την χρήση των νέων χαρακτηριστικών της OWL που έχετε προσθέσει.

### **Κ1 Κλάσεις**

a. owl:Restrictions.

- someValuesFrom ( **$\geq 10$** )
- allValuesFrom ( **$\geq 10$** )
- hasValue ( **$\geq 5$** ) → τα hasValue που αναφέρονται σε strings τα διαχειρίζεται καλύτερα ο Pellet reasoner
- minCardinality ( **$\geq 5$** )
- maxCardinality ( **$\geq 5$** )
- cardinality ( **$\geq 2$** )

Τουλάχιστον 10 ( **$\geq 10$** ) από τους παραπάνω περιορισμούς θα πρέπει να δηλωθούν με χρήση owl:equivalentClass σε μια ονοματισμένη κλάση (να είναι δηλαδή ικανές και αναγκαίες συνθήκες). Επίσης, θα πρέπει ο καθένας από αυτούς τους αναγκαίους και ικανούς περιορισμούς να οδηγεί σε αυτόματο classification (μέσω reasoning) τουλάχιστον ενός αντικειμένου στην ονοματισμένη κλάση. Αν, μετά από reasoning, δεν υπάρξει κάποιο αντικείμενο που να έχει κατηγοριοποιηθεί στην ονοματισμένη κλάση λόγω του περιορισμού, **ο περιορισμός δεν θα μετρήσει.**

b. owl:unionOf ( **$\geq 2$** )

c. owl:intersectionOf ( **$\geq 4$** )

d. owl:disjointWith ( **$\geq 10$** )

e. Συνέπεια οντολογίας - Ορθότητα ιεραρχίας. **Θα πρέπει η οντολογία που θα στείλετε να ΜΗΝ έχει ασυνέπειες και μετά την διαδικασία του reasoning**

### **Κ2 Ιδιότητες**

a. owl:SymmetricProperty ( **$\geq 5$** )

b. owl:TransitiveProperty ( **$\geq 5$** )

c. owl:inverseOf ( **$\geq 4$** )

d. owl:FunctionalProperty ( **$\geq 2$** )

e. owl:InverseFunctionalProperty ( **$\geq 2$** )

### **Κ3 Αντικείμενα**

a. Ποικιλία κάλυψης των δυνατών περιπτώσεων που δίνει η οντολογία σε OWL

### **Κ4 Εξαγωγή συμπερασμάτων**

a. Το σχήμα και τα στιγμιότυπα θα πρέπει να είναι συμβατά με τους περιορισμούς της οντολογίας OWL.

b. Είναι σημαντικό να χρησιμοποιηθεί ο reasoner για την κατηγοριοποίηση στιγμιότυπων σε κλάσεις που ορίζονται με την βοήθεια ικανών και αναγκαίων συνθηκών, π.χ. να υπάρχουν

αντικείμενα γενικής κλάσης και μέσα από τους περιορισμούς να κατηγοριοποιούνται σε επιπρόσθετες κλάσεις.

#### **K5 Αναφορά**

- a. Ύπαρξη και πληρότητα του εγγράφου προδιαγραφών (είναι απαραίτητη η υποβολή του)

## **6 Σημειώσεις**

Υπενθυμίζεται ότι για να χρησιμοποιήσετε την οντολογία σε RDFS που κάνατε στην πρώτη εργασία δεν χρειάζεται να την σχεδιάσετε από την αρχή στο Protégé. Μπορείτε να φορτώσετε το αρχείο που δημιουργήσατε με το TopBraid (.ttl) στο Protégé και να το σώσετε. Με αυτόν τον τρόπο γίνονται όλες οι μετατροπές που είναι απαραίτητες, δηλαδή: όλες οι `rdfs:Class` γίνονται `owl:Class` και όλα τα `rdf:property` γίνονται `owl:ObjectProperty` ή `owl:DatatypeProperty`, ανάλογα με το `range`. Υπάρχει όμως περίπτωση η μετατροπή να μην είναι πλήρης, ιδιαίτερα στις ιδιότητες, π.χ. να μην μπορεί να καταλάβει αν μια ιδιότητα είναι `object` ή `datatype property` γιατί δεν έχετε ορίσει `range`. Έτσι μπορεί η ιδιότητα να μην υπάρχει στο Protégé ή να έχει οριστεί ως `AnnotationProperty`. Οπότε, συστήνεται ο έλεγχος της παραγόμενης οντολογίας για ελλείψεις, ώστε να γίνουν οι απαραίτητες διορθώσεις με το χέρι.