**DATA & WEB SCIENCE**

Σημασιολογικός Ιστός

Εργασία 3 OWL

Ιπποκράτης Κοτσάνης ΑΜ: 131

To schema της εργασίας βασίζεται ακριβώς στην εργασία 2. Πρόκειται για μια εταιρεία πληροφορικής, η οποία έχει ομάδες CompanyTeam, εργαλεία τα CompayTool, τους εργαζομένους Employee και τον εξοπλισμό της Equipment.

Πιο συγκεκριμένα:

Όπως είναι φανερό και από την παρακάτω εικόνα η εταιρεία έχει τις ομάδες (CompanyTeam) οι οποίες διακρίνονται σε HR, Programming team στην οποία διακρίνονται σε BackEnd, FrontEnd και Fullstack, την ομάδα πωλήσεων (Sales) η οποία διακρίνεται σε ομάδα που αφορά πωλήσεις στο εξωτερικό (Abroad-Sales) και ομάδα που αφορά πωλήσεις στο εσωτερικό της χώρας (Local-sales).

Ακόμα τα εργαλεία της εταιρείας που χρησιμοποιούνται (CompanyTool) χωρίζονται σε Server, όπου διακρίνονται σε Application-server, Database-server, Mail-server και VM-server, και σε Software που περιλαμβάνει τις γλώσσες (Language) και χωρίζονται σε προγραμματιστικές (Programming) και σε γλώσσες κατάλληλες για testing (Testing). Οι προγραμματιστικές διακρίνονται σε Csharp, Html, Javascript και sql, ενώ οι testing διακρίνονται σε Leap-work-tests και το selenium.

Οι εργαζόμενοι της εταιρείας (Employee) διακρίνονται σε Business Manager, Developer, Economist. QA, και τους team leaders (Team-leader). Οι Developer διακρίνονται σε αυτούς που αναπτύσσουν παιχνίδια Game, οι οποίοι διακρίνονται σε Game2d και Game3d, σε αυτούς που αναπτύσσουν εφαρμογές στα κινητά Mobile και διακρίνονται σε Android, IOS και Windows και τέλος υπάρχουν και οι Web. Επίσης οι QA διακρίνονται σε Component testing, Integration testing, Performance testing και Unit testing. Τέλος οι team leaders χωρίζονται στους Managers οι οποίοι αναλαμβάνουν ευθύνες διαχείρισης της ομάδας και στους Tech οι οποίοι είναι υπεύθυνοι να καθοδηγήσουν την ομάδα τεχνικά και μοιράζουν τα tasks.

O εξοπλισμός (Equipment) της εταιρείας διακρίνεται σε γραφείο Desk, σε Laptop και σε Printer. To Desk διακρίνεται σε καρέκλες εργασίας οι οποίες είναι Anatomic, Comfort, Ergonomic, Gaming και σε οθόνες υπολογιστή Monitor που διακρίνονται σε κυρτά (curve) και τα κλασικά-επίπεδα (Standard) και τέλος τα Laptop, ενώ οι εκτυπωτές διακρίνονται σε Laser και σε ThreeD.

Graphical user interface

Description automatically generated with low confidence

Λόγω αξιολόγησης της 2ης εργασίας έγιναν οι παρακάτω αλλαγές για βελτίωση.

1. Η κλάση laptop, η οποία είχε τις subclasses Linux, MacOs και Windows-laptop, διαγράφτηκαν και έγινε εισαγωγή του domain property hasOs στην κλάση laptop προκειμένου να πάρει τιμές linux, MacOs, windows κτλ.

Graphical user interface, application

Description automatically generated

1. Στην κλάση CompanyTool > Software> Language > Programming/Testing τα πολλά instances εννοούν τις διαφορετικές εκδόσεις των γλωσσών.

Στην συνέχεια για λόγους ευελιξίας στην 3η εργασία προστέθηκαν τα παρακάτω:

1. Η Subclass LargeTeam στην κλάση CompanyTeam.
2. To property isLeadBy στην κλάση Employee, το οποίο φανερώνει ο κάθε εργαζόμενος από ποιον leader καθοδηγείται στην ομάδα που δουλεύει.

Graphical user interface, application

Description automatically generated with medium confidence Graphical user interface, application

Description automatically generated

Αποθηκεύουμε τις αλλαγές στο rdf αρχείο και στην συνέχεια με την βοήθεια του protege 5.5 το μετατρέπουμε σε owl αρχείο.

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Αφού το μετατρέψουμε σε owl αρχείο μπορούμε να δούμε κάποιες διαφορές με το rdf, όπως για παράδειγμα την μετατροπή σε owl:class, τα properties που χωρίζονται σε object properties και datatype properties κτλ.

Table

Description automatically generated Table

Description automatically generated

**Χαρακτηρισμός των Properties**

Τα object properties χαρακτηρίζονται ως functional, InverseFunctional, Symmetric και Transitive ενώ τα datatype properties χαρακτηρίζονται μόνο ως functional.

Ακόμα στα object properties ορίζονται τα παρακάτω inverse:

1. Δημιουργία νέου property στην κλάση Employee isSitBy που αποτελεί inverse του SitsOn. (Employee <-> Chair).
2. Δημιουργία νέου property στην κλάση Printer isUsedForPrintBy που αποτελεί inverse του printsIn. (Employee <->Printer).

Δήλωση functional σε datatype properties (δηλαδή παίρνουν μοναδική τιμή):

* ChairID
* firstname
* lastname
* watts

Σε object properties:

* worksAt

Δήλωση transitive σε object properties:

* QA -uses-> Testing Language -is a-> CompanyTool

QA ----------------uses--------------------> CompanyTool

**Δημιουργία περιορισμών (αναγκαίων & αναγκαίων – ικανών συνθηκών)**

**Necessary:**

1. Ένας QA (Employee) worksAt μόνο στις Programming CompanyTeam.

Graphical user interface, application

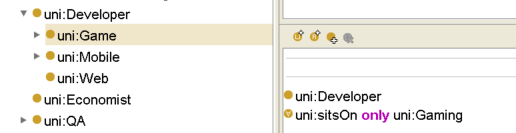
Description automatically generated

1. Οι Manager (Team-leader) printsIn μόνο στους Laser printers.

Graphical user interface, application

Description automatically generated

1. Οι Game (Developer) sitsOn μόνο σε Gaming Chair.



1. Οι Team-leader (Manager & Tech) leads μόνο Programming CompanyTeam.

Graphical user interface, text, application

Description automatically generated

**Necessary & Sufficient:**

1. O business-manager worksAt HR CompanyTeam (SomeValuesFrom), δηλαδή δουλεύει σίγουρα σε HR Team αλλά μπορεί να δουλεύει και σε κάποια άλλη CompanyTeam.

Αυτός ο περιορισμός κάνει δηλαδή **classification**. Αν δουλεύεις σε HR Team CompanyTeam τότε σίγουρα ανήκεις στην κλάση business-manager.

Φτιάχνουμε ένα νέο στιγμιότυπο στην κλάση Employee το Employee\_1\_General και συμπληρώνουμε στο property worksAt την ομάδα HR.

Graphical user interface, text, application

Description automatically generated

Με την βοήθεια του Reasoning κάνοντας κλικ στο κουμπί compute inferred type θα παρατηρήσουμε ότι o Employee\_1\_General γίνεται classified στην κλάση business-manager.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated with medium confidence

Before clicking. After clicking.

Αυτό το παράδειγμα θα μπορούσε να χρησιμοποιηθεί για παράδειγμα στην ένταξη σε ομάδα ενός νέου υπαλλήλου, δηλαδή αν ξεκινήσει στην HR team τότε χαρακτηρίζεται business-manager.

1. Database-server isWrittenBy Sql (programming language)(SomeValuesFrom), δηλαδή είναι γραμμένος σίγουρα σε Sql γλώσσα αλλά μπορεί να είναι σε οποιαδήποτε έκδοση της (range:Sql). Αυτός ο περιορισμός κάνει δηλαδή **classification**. Αν ένας server είναι γραμμένος σε Sql τότε σίγουρα ανήκει στην κλάση Database-server.

Φτιάχνουμε ένα νέο στιγμιότυπο στην κλάση Server το Server\_1\_General και συμπληρώνουμε στο property isWrittenBy την γλώσσα HR.

Για λόγους παραδείγματος το property isWrittenBy θα πάρει domain τον Server και όχι συγκεκριμένα τον Database-server.

A picture containing text

Description automatically generated

Παρακάτω βλέπουμε το νέο στιγμιότυπο που φτιάξαμε Server\_1\_General.

Graphical user interface, text, application

Description automatically generated

Με την βοήθεια του Reasoning κάνοντας κλικ στο κουμπί compute inferred type θα παρατηρήσουμε ότι o Server\_1\_General γίνεται classified στην κλάση Database-server.

Graphical user interface, application

Description automatically generated

**Τρόποι να δείξω μια ασυνέπεια με την βοήθεια του reasoner**

**1ο παράδειγμα:**

Έστω ότι θέλουμε οι Component-testing QA να κάνουν use την γλώσσα testing Leap-work-tests.

Γενικά όταν χρησιμοποιούμε το AllValuesFrom έχει την ιδιότητα του τοπικού range. Εδώ θα χρησιμοποιήσουμε το SomeValuesFrom για να δούμε τα αποτελέσματα από τον reasoner, όπως δόθηκε και σαν οδηγία στο μάθημα (Don’t use All values from for Necessary and Sufficient)

**Graphical user interface, text, application

Description automatically generated**

Και επίσης θέλουμε και οι Integration-testing QA να κάνουν use την γλώσσα testing Leap-work-tests, γιατί ας πούμε αυτό λένε οι προδιαγραφές της εταιρείας.

Graphical user interface, text, application

Description automatically generated

Στην συνέχεια θα δημιουργήσουμε ένα νέο instance που θα λέγεται QA\_1\_General και θα κάνει use την γλώσσα Leap-work-tests.

Στο σημείο αυτό να σημειωθεί ότι ορίζουμε τις κλάσεις Component-testing και Integration-testing disjoint, δηλαδή ένας QA **δεν** μπορεί να ανήκει ταυτόχρονα και στις 2 ομάδες, θα πρέπει να ενταχθεί σε μία.

Graphical user interface

Description automatically generated with low confidence

Περιμένουμε ο QA\_1\_General με τον παραπάνω αναγκαίο και ικανό περιορισμό να ενταχθεί να γίνει δηλαδή classified και στις 2 ομάδες αφού χρησιμοποιεί Leap-work-tests γλώσσα.

**Graphical user interface, application

Description automatically generated**

**Αυτό δεν μπορεί να συμβεί!**

**Οι 2 κλάσεις είναι disjoint, επομένως ο reasoner θα βγάλει error και δεν θα μπορέσει να τον εισάγει σε κάποια ομάδα.**

Παρακάτω μπορούμε να δούμε το error του reasoner όταν κάνουμε κλικ στο κουμπί του reasoner για να κάνει inferred το στιγμιότυπο QA\_1\_General.

**Graphical user interface, text, application

Description automatically generated**

**Θα διαγράψουμε το disjoint των 2 κλάσεων, προκειμένου ο reasoner να λειτουργεί σωστά. Σε περίπτωση που θέλουμε να αναπαράγουμε το error, κάνουμε τις 2 κλάσεις Component-testing και Integration-testing disjoint.**

**2ο παράδειγμα:**

Ο Component-testing QA uses μόνο Leap-work-tests γλώσσα. Ο περιορισμός αυτός δηλώθηκε από το παραπάνω παράδειγμα. Έστω ότι ο Component testing\_1 uses μία νέα γλώσσα την Leap-work-test-Csharp, η οποία δημιουργείται ως ένα νέο στιγμιότυπο της κλάσης Leap-work-tests.

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

Στην συνέχεια ορίζουμε τις κλάσεις Programming και Testing στο CompanyTool > Software > Language disjoint.

Και μεταφέρουμε την γλώσσα (στιγμιότυπο: Leap-work-test-Csharp στην κλάση Programming > Csharp γιατί τα test είναι γραμμένα στην Csharp γλώσσα.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generated

**Προφανώς και σε αυτό το παράδειγμα ο reasoner θα βγάλει error** γιατί ο Component-testing QA χρησιμοποιεί Leap-work-tests γλώσσα και αυτή που έχει ο Component-testing-1 βρίσκεται στην κλάση Programming η οποία κιόλας είναι disjoint με την κλάση Testing.

Graphical user interface, text, application

Description automatically generated

**Επαναφέρουμε το στιγμιότυπο στην κλάση Leap-work-tests της κλάσης Testing, προκείμενου ο reasoner να λειτουργεί σωστά. Σε περίπτωση που θέλουμε να αναπαράγουμε το error, αρκεί να μεταφέρουμε το στιγμιότυπο ξανά πίσω στην κλάση Csharp της κλάσης Programming.**

**Visualization**

**Diagram

Description automatically generated**

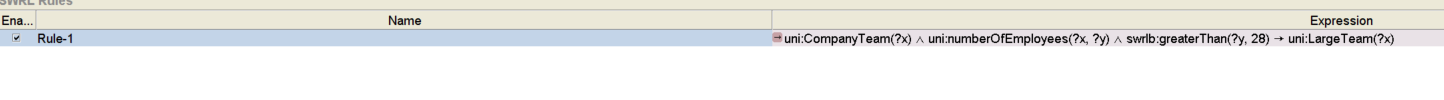
**A picture containing chart

Description automatically generated**

**Κανόνες SWRL**

**1ος κανόνας:**

Αν υπάρχει CompanyTeam που έχει περισσότερους από 28 εργαζομένους (numberOfEmployees) τότε ανήκει στην κλάση LargeTeam.



Before:

Graphical user interface, text, application

Description automatically generated

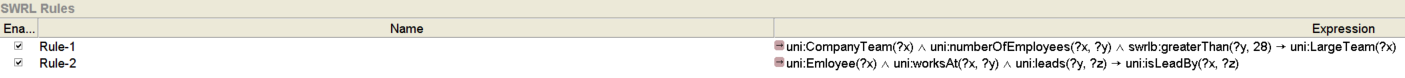
After Rule-1:

Graphical user interface, text, application

Description automatically generated

**2ος κανόνας:**

Αν ο Employee worksAt στην ομάδα CompanyTeam y και την ομάδα αυτή ο z την leads, τότε ο Employee x έχει leader (isLeadBy) τον z. (Chain property)



Before:

Graphical user interface, text, application, email

Description automatically generated

After:

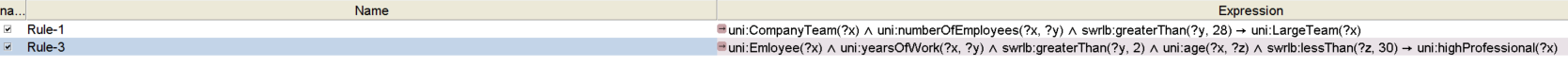
Όταν τρέχουμε τον Rule-2, o reasoner «γονατίζει», διότι τα στιγμιότυπα της κλάσης Employee είναι 64, επομένως θα πρέπει να γίνει assign το z σε 64 στιγμιότυπα όπου δημιουργεί προβλήματα μνήμης. Επομένως ο κανόνας δεν είναι λειτουργικός.

Graphical user interface, text, application, Word

Description automatically generated

**3ος κανόνας:**

Δημιουργούμε την κλάση highProfessional στην κλάση Employee, όπου θέλουμε να εισάγουμε τους εργαζομένους που έχουν πάνω από 2 χρόνια προϋπηρεσίας και είναι κάτω από 30 χρονών.



Before:

A picture containing table

Description automatically generated

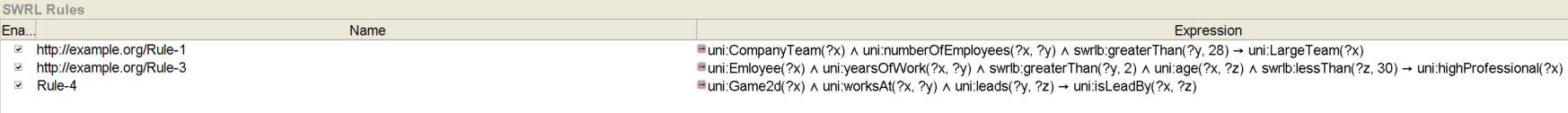
After:

A picture containing table

Description automatically generated

**4ος κανόνας:**

Θα μπορούσαμε να μεταφέρουμε το property isLeadBy στην κλάση Game2d, και να εφαρμόσουμε τον κανόνα 2. Δηλαδή, αν ο Game2d worksAt στην ομάδα CompanyTeam y και την ομάδα αυτή ο z την leads, τότε ο Game2d x έχει leader (isLeadBy) τον z. (Chain property)



Δυστυχώς και σε αυτή την περίπτωση ο Reasoner βγάζει error out of memory και ο κανόνας δεν τρέχει. Δυστυχώς δεν γνωρίζω τον λόγο!

Graphical user interface, application, Word

Description automatically generated

**Σημείωση: Οι κανόνες 2 και 4 δεν έχουν αποθηκευτεί στο project, διότι δημιουργούν θέμα στον Reasoner.**