



**ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ**  
**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ**

Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη Μοντέλου και Πρωτότυπου Συστήματος**  
**Context-Aware E-learning**

Βασίλειος Κ. Στεφανίδης

**Επιβλέπων :** Ευστάθιος Χατζηευθυμιάδης

ΠΑΤΡΑ  
ΜΑΙΟΣ, 2009



Στεφανίδης Βασίλειος, 'Ανάπτυξη Μοντέλου και Πρωτότυπου Συστήματος Context-Aware E-Learning'



ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

Διπλωματική Εργασία

**Ανάπτυξη Μοντέλου και Πρωτότυπου Συστήματος  
Context-Aware E-learning**

Στεφανίδης Βασίλειος

17 Μαΐου 2009



© ΕΑΠ, 2009

Η παρούσα διατριβή, η οποία εκπονήθηκε στα πλαίσια της ΘΕ «Διπλωματική Εργασία» του προγράμματος «Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα» (ΠΛΗΣ), και τα λοιπά αποτελέσματα της αντίστοιχης Διπλωματικής Εργασίας (ΠΕ) αποτελούν συνιδιοκτησία του ΕΑΠ και του φοιτητή, ο καθένας από τους οποίους έχει το δικαίωμα ανεξάρτητης χρήσης και αναπαραγωγής τους (στο σύνολο ή τμηματικά) για διδακτικούς και ερευνητικούς σκοπούς, σε κάθε περίπτωση αναφέροντας τον τίτλο και το συγγραφέα και το ΕΑΠ, όπου εκπονήθηκε η Διπλωματική Εργασία, καθώς και τον επιβλέποντα και την επιτροπή κρίσης.



## Ανάπτυξη Μοντέλου και Πρωτότυπου Συστήματος Context-Aware E-learning

Στεφανίδης Βασίλειος

**Όνοματεπώνυμο  
Επιβλέποντα**

ΧΑΤΖΗΕΥΘΥΜΙΑΔΗΣ  
ΕΥΣΤΑΘΙΟΣ

**Όνοματεπώνυμο  
Μέλους 1**

ΞΕΝΟΣ ΜΙΧΑΗΛ

**Όνοματεπώνυμο  
Μέλους 2**

ΚΙΤΣΟΣ ΠΑΡΑΣΚΕΥΑΣ

### Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος ηλεκτρονικής μάθησης με δυνατότητα επίγνωσης πλαισίου, δηλαδή με δυνατότητα αναγνώρισης των συνθηκών περιβάλλοντος μιας εκπαιδευτικής διαδικασίας με σκοπό την προσαρμογή της στις συνθήκες αυτές. Γενικά οι εφαρμογές που εξαρτώνται από το πλαίσιο αναπτύσσονται και βελτιώνονται διαρκώς (εφαρμογές κινητού υπολογισμού, εφαρμογές που εξαρτώνται από τη θέση κλπ). Συχνά, για την κάλυψη των απαιτήσεων των εφαρμογών αυτών απαιτείται μία μοντελοποίηση του πεδίου εφαρμογής (π.χ. μοντελοποίηση του κτιρίου στο οποίο λειτουργεί η εφαρμογή) η οποία μπορεί να πραγματοποιηθεί μέσω οντολογιών. Μέσω οντολογιών μπορεί να μοντελοποιηθεί/προδιαγραφεί και η συμπεριφορά του συστήματος ανάλογα με την προσδιορισμένη τιμή των περιβαλλοντικών παραμέτρων.

Η ανερχόμενες τεχνολογίες διαχείρισης γνώσης (knowledge management) όπως οι τεχνολογίες του Σημασιολογικού Ιστού μπορούν να χρησιμοποιηθούν όχι μόνο για την περιγραφή μίας εφαρμογής επίγνωσης πλαισίου (context aware application) αλλά και την περιγραφή της διαδικασίας μάθησης και τους μηχανισμούς μάθησης (π.χ., διαγωνίσματα, αξιολογήσεις, ερωτήσεις, παραδείγματα). Αυτή η πληροφορία μπορεί να συγκεκριμενοποιηθεί σε συγκεκριμένα πεδία και να συντελέσει στην ανάπτυξη ευφυούς εκπαιδευτικού λογισμικού. Η συγκεκριμένη εργασία αποσκοπεί :



(Α) στην διερεύνηση άλλων προτάσεων που έχουν διατυπωθεί σχετικά με την οντολογική αναπαράσταση συστημάτων e-learning με context-awareness.

(Β) Συνδυασμός των καταλληλότερων πρακτικών όπως αυτές έχουν προσδιοριστεί στο βήμα Α.

(Γ) Υλοποίηση πρωτότυπου συστήματος ubiquitous e-learning (πραγματοποίηση εκπαιδευτικής δραστηριότητας οπουδήποτε και οποτεδήποτε). Εφαρμογή με πληροφορία πλαισίου σε ότι αφορά το χώρο, με έμφαση στα ηχητικά δεδομένα και τις συντεταγμένες. Προγραμματισμός του μοντέλου αναπαράστασης πληροφορίας (χωρικής, ηχητικής, εκπαιδευτικής) και ανάπτυξη της σχετικής εφαρμογής (υλοποίηση test, διαγωνίσματος).

Πιο συγκεκριμένα μέσα από την εργασία μας αναπτύξαμε και τελικά υλοποιήσαμε ένα σύστημα ηλεκτρονικής μάθησης το οποίο διαδραστικά με τον μαθητή και λαμβάνοντας υπόψη περιβαλλοντικές συνθήκες μπορεί να προσαρμόζει τη διαδικασία της μάθησης.

Για να το πετύχουμε αυτό δημιουργήσαμε μια οντολογία και χρησιμοποιήσαμε και εμπλουτίσαμε μια ακόμα. Οι οντολογίες αυτές αποτέλεσαν τον ενδιάμεσο ανάμεσα στα συστήματα που φέραμε σε συνεργασία. Η μια από τις δύο, η εκπαιδευτική, είναι μια απλή οντολογία για εκπαιδευτικά θέματα, με τις πιο σημαντικές οντότητες της εκπαιδευτικής διαδικασίας, ενώ η δεύτερη είναι μια ήδη γνωστή οντολογία που σχετίζεται με τους εσωτερικούς χώρους ενός κτιρίου. Αυτή τη δεύτερη οντολογία που σχετίζεται με το περιβάλλον, την εμπλουτίσαμε με μερικές ακόμα χρήσιμες για την εργασία μας κλάσεις.

Στη συνέχεια δημιουργήσαμε ένα σύστημα απόφασης το οποίο στην ουσία συνδύασε τα στοιχεία, που συλλέγονται από τους αισθητήρες του συστήματος, και με τη χρησιμοποίηση ενός συνόλου από κανόνες (χρήση fuzzy logic) αποφασίζει για τις επόμενες ενέργειες.

Τέλος στόχος μας ήταν η υλοποίηση του συστήματος σε περιβάλλον Java.

**Λέξεις-κλειδιά:** Επίγνωση Πλαισίου, Ηλεκτρονική Μάθηση, Οντολογίες, Σημασιολογικός Ιστός, Java, Bossam.

**Περιεχόμενο:** Κείμενο, πρόγραμμα σε γλώσσα Java κλπ



## Context-Aware E-Learning Model and Prototype System Development

Stefanidis Vassilis

**Όνοματεπώνυμο**  
**Επιβλέποντα στην**  
**Αγγλική**

Hadjiefthymiades Stathes

**Όνοματεπώνυμο**  
**Μέλους 1 στην Αγγλική**

Xenos Michalis

**Όνοματεπώνυμο**  
**Μέλους 2 στην Αγγλική**

Kitsos Paraskeyas

### Summary

The subject of this Postgraduate Thesis is the development of a context aware e-learning system. More analytical we can describe this system as a fuzzy system, that it can recognize the environment circumstances in order to apply one educational procedure properly. Generally applications that they depends from context, improved continuously (mobile computing applications, position depended applications ect). Often, for the coverage of applications demands it is requested a domain field modeling (for instance a building modeling). This modeling can implemented through the ontologies technology. Furthermore through ontologies technology we can modeling the system behavior, in regard with the context parameters.

Increased technologies, of knowledge management as the Semantic Web, can be used not only for the description of a context aware application but also the description of learning procedures and the mechanisms of learning. For example tests, evaluations, questions, examples, paradigms. This information can be more adaptive in particularly scientist fields, and contribute in the growth of fuzzy educational software. The particular work aims:

A) to survey other proposals that has been formulated with regard to e-learning systems ontological representation with context-awareness.

(B) to do a combination of more suitable practices as these have been determined in the step A.



(Γ) Prototype ubiquitous e-learning system development. Implementation with context information regarding the geographical information, with accent in the context sound data and the coordinates. Programming the information representation model (information relative to space, sound or education) and finally application development.

More specifically, we are obligated to developing and to implement an e-learning system that interactively with the student and with respect to the context awareness it can be adaptive to the educational procedure. To achieve that, we will create ontologies that will combine the systems that will be in collaboration. An additional object of this thesis is to implement and obviously to use two ontologies. An educational and a context ontology.

Continuously, we want to create a system of decision which in the substance will combine the elements, that will be collected by the sensors of the system, and with the utilisation of a set of rules (fuzzy logic usage) will decide for next actions.

Finally our goal will be the implementation of this system in Java environment.

**Keywords:** Context-Aware E-Learning, Ontologies, Semantic Web, Java, Bossam,

**Content:** Text, Java Program



## Εικόνες

- Εικόνα 1:* Στιγμιότυπο από τη δημιουργία νέας οντολογίας με πρότυπο  
*Εικόνα 2 :* Η Οθόνη όπου φαίνονται τα URI της ονοματολογίας  
*Εικόνα 3:* Οι κλάσεις και οι υποκλάσεις της GEOnt  
*Εικόνα 4:* η οπτικοποίηση της GEOnt (Η ιεραρχία)  
*Εικόνα 5:* η οπτικοποίηση της GEOnt σε πιο μεγάλη κλίμακα (Ιεραρχία)  
*Εικόνα 6:* Οι συσχετίσεις – ιδιότητες που δημιουργήσαμε  
*Εικόνα 7 :* Οι συσχετίσεις – ιδιότητες που δημιουργήσαμε  
*Εικόνα 8:* Τα στιγμιότυπα (instances-individuals) της οντολογίας GEOnt  
*Εικόνα 9:* Η Ιεραρχία της οντολογίας INO  
*Εικόνα 10 :* Εμπλουτισμός της INO με δύο νέες υποκλάσεις  
*Εικόνα 11 :* Η νέα συσχέτιση has με τις κλάσεις που συσχετίζει  
*Εικόνα 12 :* Η νέα συσχέτιση is\_it με τις κλάσεις που συσχετίζει  
*Εικόνα 13:* Στιγμιότυπο επιλογής του reasoner Pellet  
*Εικόνα 14 :* Εικόνα με στιγμιότυπα της οντολογίας  
*Εικόνα 15 :* Στιγμιότυπο από την εκτέλεση ερωτήματος DL  
*Εικόνα 16:* Σύνδεση εκφραστικότητας με περιγραφή πεδίου  
*Εικόνα 17 :* Σημεία στο protégé στα οποία μπορούμε να δούμε το namespace (κόκκινο βέλος)  
*Εικόνα 18 :* Το πρώτο παράθυρο της  
*Εικόνα 19 :* Η Οθόνη της εφαρμογής από το οποίο μπορούμε να εκτελέσουμε σενάρια  
*Εικόνα 20 :* Οι υποτιθέμενες ερωτήσεις των σεναρίων  
*Εικόνα 21 :* Η αρχιτεκτονική του συστήματος  
*Εικόνα 22 :* Αρχικοποίηση του σεναρίου  
*Εικόνα 23 :* Εκτέλεση του σεναρίου A  
*Εικόνα 24 :* Ο εξεταζόμενος είναι σε ήσυχο περιβάλλον  
*Εικόνα 25:* Πληροφορία ανά δευτερόλεπτο πρώτου σεναρίου  
*Εικόνα 26 :* Ο εξεταζόμενος είναι σε πιο θορυβώδες περιβάλλον  
*Εικόνα 27 :* Η πρώτη οθόνη του B σεναρίου  
*Εικόνα 28:* Τα B σενάριο στο 3<sup>ο</sup> δευτερόλεπτο  
*Εικόνα 29:* Τα B σενάριο στο 9<sup>ο</sup> δευτερόλεπτο  
*Εικόνα 30:* Η πρώτη οθόνη του C σεναρίου  
*Εικόνα 31:* Το C σενάριο στο 3<sup>ο</sup> δευτερόλεπτο  
*Εικόνα 32:* Το C σενάριο στο 10<sup>ο</sup> δευτερόλεπτο  
*Εικόνα 33:* Το C σενάριο στο 18<sup>ο</sup> δευτερόλεπτο  
*Εικόνα 34 :* Οι βιβλιοθήκες της μηχανής Bossam  
*Εικόνα 35 :* Οι ιδιότητες της εφαρμογής με τις επιπρόσθετες βιβλιοθήκες





## Πίνακες

*Πίνακας 1 : Μερικοί βασικοί κανόνες σύνταξης της Buchingae*

*Πίνακας 2 : Έτοιμες Συναρτήσεις Buchingae*



## Περιεχόμενα

1.	ΕΙΣΑΓΩΓΗ .....	11
2.	ΤΙ ΕΙΝΑΙ ΤΟ SEMANTIC WEB .....	12
3.	ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΓΝΩΣΗΣ .....	15
4.	ΣΥΣΤΗΜΑΤΑ E-LEARNING ΜΕ CONTEXT AWARENESS.....	17
5.	ΟΝΤΟΛΟΓΙΕΣ .....	18
5.1.	ΚΑΤΗΓΟΡΙΕΣ ΟΝΤΟΛΟΓΙΩΝ .....	20
5.2.	ΕΦΑΡΜΟΓΕΣ ΤΩΝ ΟΝΤΟΛΟΓΙΩΝ .....	21
6.	ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΟΝΤΟΛΟΓΙΩΝ ΜΕ ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ.....	22
6.1.	Το PROTEGE.....	22
6.2.	ΔΗΜΙΟΥΡΓΙΑ ΜΙΑΣ ΟΝΤΟΛΟΓΙΑΣ ΣΤΟ PROTEGE .....	24
7.	Η ΓΛΩΣΣΑ OWL.....	25
8.	ΑΝΑΠΤΥΞΗ ΚΑΙ ΧΡΗΣΗ ΟΝΤΟΛΟΓΙΩΝ .....	33
8.1.	ΕΚΠΑΙΔΕΥΤΙΚΗ ΟΝΤΟΛΟΓΙΑ GEONT.....	33
8.1.1.	Η δομή και οι συσχετίσεις της GEOnt .....	37
8.1.2.	Τα Στιγμιότυπα της GEOnt .....	39
8.2.	ΟΝΤΟΛΟΓΙΑ ΠΛΑΙΣΙΟΥ ΙΝΟ (ΠΕΡΙΒΑΛΛΟΝΤΟΣ).....	40
8.3.	ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ .....	43
9.	ΤΕΧΝΟΛΟΓΙΑ ΚΡΙΣΗΣ (REASONERS-CLASSIFIERS ).....	44
10.	ΤΕΧΝΟΛΟΓΙΑ ΣΥΜΠΕΡΑΣΜΟΥ (INFERENCE-REASONING).....	46
10.1.	ΔΗΜΙΟΥΡΓΙΑ INDIVIDUALS .....	47
10.2.	ΥΠΟΒΟΛΗ ΕΡΩΤΗΜΑΤΩΝ .....	48
10.3.	ΑΛΗΘΕΙΕΣ, ΕΡΩΤΗΜΑΤΑ ΚΑΙ ΚΑΝΟΝΕΣ.....	49
11.	Η ΜΗΧΑΝΗ BOSSAM .....	51
12.	BUCHINGAE – Η ΓΛΩΣΣΑ ΟΡΙΣΜΟΥ ΚΑΝΟΝΩΝ ΤΗΣ BOSSAM .....	52
12.1.	ΑΛΗΘΕΙΕΣ (FACTS).....	53
12.2.	ΚΑΝΟΝΕΣ (RULES).....	54
12.3.	ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΚΑΝΟΝΑ (RULE PRIORITY).....	54
12.4.	CLASS .....	55
12.5.	PROPERTY .....	55
12.6.	ΣΤΙΓΜΙΟΤΥΠΑ (INSTANCES Η INDIVIDUALS) .....	56
12.7.	ΣΥΝΑΡΤΗΣΕΙΣ (FUNCTIONS).....	57
13.	ΔΙΑΔΙΚΑΣΙΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΓΝΩΣΗΣ ΚΑΙ ΕΚΔΟΣΗΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ .....	58
13.1.	ΦΟΡΤΩΣΗ OWL ΑΡΧΕΙΩΝ ΚΑΙ QUERYING .....	58
13.2.	ΕΚΤΕΛΕΣΗ ΣΕ ΠΕΡΙΒΑΛΛΟΝ JAVA.....	61
13.3.	ΠΡΟΣΘΗΚΗ ΝΕΩΝ ΣΥΣΧΕΤΙΣΕΩΝ .....	66
13.4.	ΚΑΤΑΡΓΗΣΗ ΕΙΣΑΧΘΕΙΣΗΣ ΓΝΩΣΗΣ – ΙΣΧΥΡΗ ΑΡΝΗΣΗ .....	67
13.5.	ΣΥΜΠΕΡΑΣΜΟΣ .....	68
13.6.	ΕΡΩΤΗΜΑΤΑ ΣΕ ΔΥΟ ΟΝΤΟΛΟΓΙΕΣ ΚΑΙ ΚΑΝΟΝΕΣ ΜΕ ΣΥΝΘΗΚΕΣ.....	69
14.	ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ – ΠΕΡΙΒΑΛΛΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ.....	70
15.	ELSS (E-LEARNING SMART SYSTEM) .....	71
15.1.	ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΙΜΟΤΗΤΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	71



<b>16.</b>	<b>ΟΙ ΘΘΟΝΕΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ</b> .....	<b>72</b>
<b>17.</b>	<b>Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΚΑΙ ΤΑ ΤΜΗΜΑΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ</b> .....	<b>75</b>
17.1.	Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	76
17.2.	ΤΜΗΜΑ ΟΝΤΟΤΗΤΩΝ.....	77
17.3.	ΤΜΗΜΑ ΚΑΝΟΝΩΝ ΚΑΙ ΑΛΗΘΕΙΩΝ.....	77
17.4.	ΤΜΗΜΑ ΕΙΣΑΓΩΓΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΑΠΟ ΤΟ ΠΕΡΙΒΑΛΛΟΝ.....	78
17.4.1.	Τμήμα εισαγωγής πληροφορίας από αισθητήρες.....	81
17.4.2.	Τμήμα εισαγωγής πληροφορίας από GPS.....	81
17.5.	ΤΜΗΜΑ ΣΥΜΠΕΡΑΣΜΟΥ.....	82
17.6.	ΤΜΗΜΑ ΑΠΟΤΕΛΕΣΜΑΤΩΝ.....	83
17.7.	ΑΠΟΚΟΠΗ ΤΟΥ ΑΠΟΤΕΛΕΣΜΑΤΟΣ ΑΠΟ ΤΗΝ URI.....	83
<b>18.</b>	<b>ΕΚΠΑΙΔΕΥΤΙΚΑ ΣΕΝΑΡΙΑ</b> .....	<b>83</b>
18.1.	ΕΚΠΑΙΔΕΥΤΙΚΟ ΣΕΝΑΡΙΟ Α.....	84
18.2.	ΕΚΠΑΙΔΕΥΤΙΚΟ ΣΕΝΑΡΙΟ Β.....	89
18.3.	ΕΚΠΑΙΔΕΥΤΙΚΟ ΣΕΝΑΡΙΟ C.....	93
18.4.	ΠΑΡΑΛΛΑΓΕΣ ΤΩΝ ΣΕΝΑΡΙΩΝ.....	97
<b>19.</b>	<b>ΜΕΤΑΓΛΩΤΤΙΣΗ ΚΑΙ ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ</b> .....	<b>99</b>
	<b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b> .....	<b>102</b>
	<b>ΠΑΡΑΡΤΗΜΑ Α</b> .....	<b>106</b>
	ΚΩΔΙΚΑΣ 1.....	106
	ΚΩΔΙΚΑΣ 2.....	108
	ΚΩΔΙΚΑΣ 3.....	110
	<b>ΠΑΡΑΡΤΗΜΑ Β</b> .....	<b>114</b>
	ΛΙΓΑ ΛΟΓΙΑ ΓΙΑ ΤΙΣ ΠΛΑΤΦΟΡΜΕΣ ΑΝΑΠΤΥΞΗΣ ECLIPSE ΚΑΙ NETBEANS.....	114
	ΚΩΔΙΚΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ.....	115



## 1. Εισαγωγή

Στη σημερινή εποχή υπάρχει ολοένα και περισσότερο η ανάγκη για οργάνωση της πληροφορίας κάτω από ομαδοποιημένες ενότητες γνώσης. Η γνώση ομαδοποιείται σε συγγενείς ομάδες οι οποίες με τη σειρά τους οργανώνονται κάτω από γενικότερα και πιο μοντέρνα αντικείμενα τα οποία ονομάζονται οντολογίες. Η οντολογία συμβάλλει στην καλύτερη και αποτελεσματικότερη οργάνωση και απεικόνιση ενός πεδίου ορισμού. Η οργανωτική αυτή δομή συμβάλλει στην ανάπτυξη του Σημασιολογικού Ιστού. Αυτό το τελευταίο μπορεί να βάλει μια τάξη στον αχανή παγκόσμιο ιστό και να εξάγει ακριβείς πληροφορίες στους χρήστες του.

Ο παγκόσμιος ιστός είναι μια μεγάλη χοάνη με κείμενα, ιδέες, μηνύματα, εικόνες τα οποία προς το παρόν στο μεγαλύτερο μέρος τους παραμένουν αχρησιμοποίητα. Παράλληλα όλα αυτά βρίσκονται κάτω από ετερογενή συστήματα τα οποία είναι απρόθυμα να συνεργαστούν μεταξύ τους. Ο σημασιολογικός ιστός με τη βοήθεια των οντολογιών υπόσχεται στο μέλλον να βάλει μια τάξη σε όλη αυτή την κατάσταση και να καταστήσει χρήσιμη την αχρησιμοποίητη σήμερα πληροφορία.

Οι υπηρεσίες του ιστού είναι κατανεμημένες στο διαδίκτυο. Οι χρήστες πολλές φορές θέλουν τα οφέλη από αντίστοιχες υπηρεσίες και δεν μπορούν να τις εντοπίσουν και να τις χρησιμοποιήσουν. Σε άλλες περιπτώσεις οι υπηρεσίες δεν μπορούν να είναι διαθέσιμες εξαιτίας περιβαλλοντικών συνθηκών, ενώ σε άλλες περιπτώσεις η αδυναμία προέρχεται από την κατάσταση του ίδιου του χρήστη. Τέτοιες αδυναμίες έρχονται να διορθώσουν τα συστήματα που έχουν επίγνωση πλαισίου και που έχουν τη δυνατότητα να προσαρμόζουν τη συμπεριφορά τους.

Επιπρόσθετα ο εκπαιδευτικός χώρος χρειάζεται νέες ιδέες και νέες μεθόδους που μπορούν να βοηθήσουν την εκπαιδευτική διαδικασία. Ιδέες που έχουν να κάνουν με την μάθηση οπουδήποτε και οποτεδήποτε είναι μοντέρνες, καινοτόμες και μπορούν να εφαρμοσθούν και σε διαφορετικά σημεία, πέρα από τα στενά περιθώρια μιας τάξης. Στην αποτελεσματικότητα της καινοτομίας, μπορούν να βοηθήσουν οι τεχνολογίες που αναφέραμε πιο πάνω, όπως είναι η τεχνολογία των οντολογιών, τα δίκτυα, οι κινητοί υπολογισμοί και ο σημασιολογικός ιστός.

Στις επόμενα κεφάλαια αυτής της εργασίας θα γνωρίσουμε αναλυτικά τις προαναφερθείσες τεχνολογίες, και θα αναπτύξουμε δικά μας εργαλεία για να τις χρησιμοποιήσουμε. Τέλος θα συνδέσουμε όλες αυτές τις τεχνολογίες και θα τις εφαρμόσουμε στην εκπαιδευτική διαδικασία. Αναπτύσσουμε τρία ενδεικτικά εκπαιδευτικά σενάρια και παρακολουθούμε την



εφαρμογή της διαδικασίας ηλεκτρονικής μάθησης με επίγνωση πλαισίου, δηλαδή με επίγνωση των συνθηκών περιβάλλοντος ή με επίγνωση της κατάστασης του χρήστη.

Τέλος αναπτύσσουμε μια εφαρμογή σε περιβάλλον Java, η οποία μπορεί να καταδείξει τη δύναμη των τεχνολογιών αυτών, και παράλληλα προσφέρει μια πλατφόρμα για περαιτέρω έρευνα. Η πλατφόρμα μας αναπτύχθηκε σε περιβάλλον windows αλλά είναι μεταφέρσιμη σε λειτουργικά συστήματα ανοικτού λογισμικού αφού το περιβάλλον της Java είναι διαθέσιμο και για τέτοια λειτουργικά συστήματα.

## 2. Τι είναι το semantic web

Ο Σημασιολογικός Ιστός (Semantic Web), αποτελεί [17] [29] [27] [28] μια πρωτοβουλία της Κοινοπραξίας του Παγκοσμίου Ιστού (World Wide Web Consortium – W3C [43]) και παρέχει μια διεθνώς προσβάσιμη πλατφόρμα που επιτρέπει σε υπολογιστικά συστήματα, διαδικτυακές μηχανές, αυτοματοποιημένες υπηρεσίες αλλά και σε ανθρώπους να επεξεργάζονται και να διαμοιράζονται δεδομένα. Είναι κατ' ουσίαν μια πρόταση για την μετεξέλιξη του διαδικτύου. Ο στόχος του Σημασιολογικού Ιστού είναι να οδηγήσει και να μετεξελίξει τη σημερινή μορφή του διαδικτύου, έτσι ώστε οι πληροφορίες που υπάρχουν και διακινούνται σε αυτό να είναι επεξεργάσιμες.

Αντίθετα με τη σημερινή μορφή του διαδικτύου όπου οι υπολογιστές χρησιμοποιούνται απλώς για την αποθήκευση, ο Σημασιολογικός Ιστός είναι ένα σύνολο πληροφοριών, διασυνδεδεμένων με κατάλληλο τρόπο, ώστε να είναι εύκολα και με αποδοτικό τρόπο επεξεργάσιμες, σε παγκόσμια κλίμακα. Θα πρέπει να τον θεωρούμε ως μία βάση δεδομένων με παγκόσμιο χαρακτήρα, της οποίας η δομή και οργάνωση επιτρέπει αφενός στους χρήστες (ανθρώπους), αφετέρου δε στις μηχανές να χρησιμοποιήσουν την πολύτιμη αποθηκευμένη πληροφορία. Ο Σημασιολογικός Ιστός αποτελεί ουσιαστικά μία επέκταση του Παγκόσμιου Ιστού (World Wide Web) η οποία επιτρέπει την αποτελεσματικότερη συνεργασία ανθρώπων και υπολογιστών, σύμφωνα με τον εμπνευστή του (και εμπνευστή των WWW, URIs, HTTP και HTML) Tim Berners-Lee.[43]

Τα τρέχοντα συστήματα διαχείρισης πληροφορίας έχουν σημαντικές αδυναμίες [6] [28]:

- Αναζήτηση πληροφορίας: Τα υπάρχοντα συστήματα αναζήτησης που βασίζονται στην πληκτρολόγηση λέξεων-κλειδιών μπορούν να συμπεριλάβουν στα



αποτελέσματα και άσχετες προς το θέμα πληροφορίες όταν οι λέξεις έχουν πολλαπλή έννοια. Επίσης μπορεί να χάσουν πληροφορίες αν χρησιμοποιηθεί διαφορετική ορολογία για το ίδιο θέμα.

- Η λήψη της πληροφορίας: Σήμερα η επισκόπηση (browsing) και η ανάγνωση της πληροφορίας είναι απαραίτητη ώστε κάποιος να εντοπίσει τη σωστή πληροφορία αφού οι μηχανές αναζήτησης δεν μπορούν να ξεχωρίσουν το ιδιαίτερο νόημα κάθε λέξης και να το υποδείξουν σε διαφορετικά κείμενα.
- Η Διατήρηση πληροφορίας: Η οργάνωση και διάρθρωση μιας βάσης δεδομένων είναι μια δύσκολη και χρονοβόρα εργασία, ιδιαίτερα όταν οι πηγές πληροφόρησης είναι μεγάλες. Το να κρατά κανείς τέτοια συλλογή, να τη διορθώνει και να την ανανεώνει απαιτεί χρόνο και καλή μέθοδο ταξινόμησης.
- Αυτόματη παραγωγή document: Η μεταπήδηση μελλοντικά από το κείμενο που είναι κατανοητό μόνο από τον άνθρωπο σε ημι-δομημένη ή και δομημένη πληροφορία που μπορεί να γίνει αυτόματα κατανοητή από διαδικτυακές εφαρμογές (π.χ διαδικτυακές εφαρμογές, ευφυείς πράκτορες).

Η κυριότερη λειτουργία του Σημασιολογικού Ιστού είναι ότι περιλαμβάνει τη σαφή αναπαράσταση του νοήματος των πληροφοριών και των εγγράφων, επιτρέποντας την αυτόματη επεξεργασία και ενοποίηση διαδικτυακών πόρων από "έξυπνα" προγράμματα-πράκτορες. Έτσι, επιδιώκει να καταστήσει την πληροφορία πιο κατανοητή για τους υπολογιστές με την εισαγωγή μιας αυστηρότερης δομής βασισμένης στις οντολογίες. Με τον όρο οντολογία εννοούμε την ακριβή περιγραφή εννοιών καθώς και των σχέσεων που υπάρχουν ανάμεσά τους, και γύρω από ένα πεδίο ενδιαφέροντος. [6], [42]. Η έννοια της οντολογίας θα μας απασχολήσει στη συνέχεια εκτενώς.

Ο Σημασιολογικός Ιστός βασίζεται στα μετα-δεδομένα (metadata) ή μετα-πληροφορία. Τα μετα-δεδομένα είναι δεδομένα που αναφέρονται σε άλλα δεδομένα (data about data). Για να λειτουργήσει απαιτείται εμπλουτισμός των δεδομένων του Ιστού με σημασιολογία, έτσι ώστε να είναι κατανοητά από τους υπολογιστές επιτρέποντας έτσι την εξαγωγή υπονοούμενης γνώσης. Στην κατεύθυνση αυτή τα σημερινά κείμενα στις σελίδες του Web θα αντικατασταθούν με δομημένα κείμενα και δεδομένα σε μορφή XML και RDF. Η σημερινή αναπαράσταση που προορίζεται για χρήση από ανθρώπους θα αντικατασταθεί από αναπαράσταση κατανοητή στους υπολογιστές.



Με τη σημερινή υποδομή που έχει δημιουργηθεί τα τελευταία 20 χρόνια, υπάρχει έλλειψη πραγματικών σημασιολογικών δεδομένων. Τα δεδομένα που υπάρχουν στον παγκόσμιο ιστό είναι κείμενα, εικόνες, ήχος, βίντεο καθώς και δεδομένα από βάσεις που στην πλειοψηφία τους είναι σχεσιακά δεδομένα. Καθίσταται λοιπόν αναγκαία η εύρεση ενός τρόπου δημιουργίας σημασιολογικών δεδομένων από ήδη υπάρχοντα δεδομένα. Είναι γνωστό ότι μεγάλη ποσότητα δεδομένων στον ιστό είναι αποθηκευμένη σε σχεσιακές βάσεις δεδομένων. [5], [6]. Επίσης το μεγαλύτερο μέρος της διαθέσιμης πληροφορίας οργανώνεται με τρόπο που καθιστά δύσκολη την επεξεργασία της από μία μηχανή. Η πληροφορία αποθηκεύεται συνήθως στη μορφή HTML αρχείων, τα οποία προσφέρουν μόνο οπτική απεικόνιση και όχι σημασιολογική ταξινόμηση της πληροφορίας. Αυτό έχει ως αποτέλεσμα, ένας άνθρωπος, διαβάζοντας μία σελίδα HTML, να μπορεί να διαχωρίσει σημασιολογικά την πληροφορία που περιέχει, αλλά αυτό να είναι αδύνατο για μία εφαρμογή.

Αυτό που επιδιώκεται τελικά με το Σημασιολογικό Ιστό είναι η ρητή και με σαφήνεια σημασιολογική επισημείωση της πληροφορίας, έτσι ώστε να διευκολύνεται η αυτοματοποιημένη επεξεργασία και ολοκλήρωσή της από μία μηχανή. Οι πιο γνωστές γλώσσες που καλούνται σήμερα να περιγράψουν αυτήν την εκτεταμένη πληροφορία είναι οι XML, RDF, RDFS και η OWL. Θέλοντας να περιγράψουμε το τι κάνει κάθε γλώσσα θα μπορούσαμε να πούμε ότι η XML (Extensible Markup Language) παρέχει την δυνατότητα ορισμού ετικετών στα διάφορα σχήματα που δημιουργούνται, η RDF (Resource Description Framework) έχει τη δυνατότητα ευέλικτης παρουσίασης δεδομένων της πληροφορίας και η OWL (Web Ontology Language) η οποία παρέχει τη δυνατότητα για τυπική περιγραφή της σημασιολογίας και ορολογίας ενός εγγράφου.

Τα παραπάνω θέματα τα οποία θέλουμε να περιγράψουμε με το Σημασιολογικό Ιστό δεν αφορούν μόνο το διαδίκτυο, αλλά αφορά και οποιονδήποτε άλλο τομέα στον οποίο προκύπτει η ανάγκη για αποθήκευση και ανάσυρση πληροφορίας. Εν προκειμένω, όσον αφορά την αναζήτηση εγγράφων στο σκληρό δίσκο ενός υπολογιστή, η μέχρι στιγμής υπάρχουσα κλασσική αναζήτηση με λέξεις κλειδιά εμφανίζει διαφόρων ειδών ατέλειες. Για παράδειγμα, ένα έγγραφο μπορεί να αναφέρεται σε μία έννοια, αλλά να μην περιέχει (αρκετές) λέξεις κλειδιά που να περιγράφουν τη συγκεκριμένη έννοια ή και το αντίστροφο, να περιέχει πολλές λέξεις κλειδιά μίας έννοιας στην οποία όμως δεν αναφέρεται. Επιπλέον, μπορεί διαφορετικά σημεία του εγγράφου να αναφέρονται σε διαφορετικές έννοιες. Ο Σημασιολογικός Ιστός



ευελπιστεί να λύσει και θέματα οργάνωσης και περιγραφής της πληροφορίας, τοπικού χαρακτήρα.

Οι εφαρμογές του Σημασιολογικού Ιστού βασίζονται σε ετερογενείς πλατφόρμες υλικού, γλώσσες προγραμματισμού και πρωτόκολλα επικοινωνίας. Ωστόσο αυτές οι εφαρμογές δουλεύουν και επικοινωνούν χρησιμοποιώντας δηλώσεις, θέτουν ερωτήσεις, ζητούν και δίνουν απαντήσεις και παίρνουν την προηγούμενη γνώση σαν εισαγωγή δεδομένων (in put), λειτουργούν σαν ατζέντες (προγράμματα-πράκτορες) και διαχειρίζονται και ανταλλάσσουν πληροφορίες - γνώση. Τέλος, βασικό συστατικό του Semantic Web πέρα από μέθοδο αναπαράστασης της γνώσης είναι και ένας μηχανισμός που μας επιτρέπει να επεξεργαστούμε την γνώση αυτή. Αυτός ο μηχανισμός θα πρέπει να υποστηρίζει την δυνατότητα λογικής επεξεργασίας των πληροφοριών με σκοπό την εξαγωγή συμπερασμάτων, την δημιουργία νέας γνώσης, την υποστήριξη στην λήψη αποφάσεων και την αυτόματη εκτέλεση ενεργειών.

### 3. Συστήματα Διαχείρισης Γνώσης

Τα συστήματα διαχείρισης γνώσης αποτελούν σήμερα έναν ταχέως αναπτυσσόμενο επιστημονικό τομέα με αιχμή του δόρατος την οργάνωση των διαδικτυακών δεδομένων όπως αναφέραμε ήδη στην παράγραφο του Σημασιολογικού Ιστού. Η διαχείριση γνώσης έχει προσδιορισθεί ως μια στρατηγικά σημαντική ικανότητα και μπορεί να έχει εφαρμογή στη διακυβέρνηση, τις επιχειρήσεις, την εκπαίδευση, κλπ.

Στο παρελθόν η τεχνολογία των πληροφοριών για διαχείριση γνώσης είχε εστιασθεί στη διαχείριση απλών βάσεων δεδομένων, χρησιμοποιώντας απλά κείμενα ως βασική αποθήκη και πηγή γνώσης. Τα τελευταία χρόνια όμως υπάρχει μια σαφής εξειδίκευση με αποτέλεσμα την ανάπτυξη των τεχνολογιών και των συστημάτων τα οποία διαχειρίζονται τη γνώση. Επιπλέον έχουν αναπτυχθεί πολλές αξιόλογες και επιτυχημένες μέθοδοι και τεχνικές αναπαράστασης γνώσης (ηλεκτρονικές βιβλιοθήκες, διαχείριση γνώσης με στόχευση τις επιχειρήσεις, δίκτυα διαχείρισης δεδομένων κλπ). Ως γενική αρχή μπορούμε να πούμε ότι τα συστήματα αυτά είναι κλειστά (ως προς τη χρήση τους) και έχουν σαφώς καθορισμένα πεδία εφαρμογής. Σε αντίθεση, το διαδίκτυο είναι ένα καθολικό και αποκεντρωμένο σύστημα πληροφοριών. Τα χαρακτηριστικά αυτά προσδίδουν πλεονεκτήματα και μειονεκτήματα. Στα πλεονεκτήματα αναφέρουμε την ταχεία εξάπλωση, τον τεράστιο όγκο πληροφοριών την ευελιξία, την ευκολία χρήσης και κυρίως τον ανοιχτό χαρακτήρα: Βασική αρχή του Web





είναι ότι “οποιοσδήποτε μπορεί να πει οτιδήποτε για οποιοδήποτε θέμα”. Στην αρχή αυτή βέβαια τα μειονεκτήματα είναι η αναξιοπιστία, η έλλειψη ορθότητας και ακεραιότητας των πληροφοριών. Αυτό το κενό καλείται να καλύψει ο Σημασιολογικός Ιστός μέσα από τα συστήματα διαχείρισης γνώσης. Πρωτεύοντα ρόλο στην ανάπτυξη αυτών των συστημάτων έχει παίξει η ανάπτυξη των οντολογιών.

Οι οντολογίες είναι βασικό στοιχείο της παρούσας εργασίας και για το λόγο αυτό, γίνεται εκτενής αναφορά στη συνέχεια. Εν συντομία μπορούμε να πούμε ότι η οντολογία είναι ένας απαραίτητος παράγοντας για την μοντελοποίηση και την διαχείριση της γνώσης. Στην περίπτωση των πληροφοριακών συστημάτων, ως οντολογία χαρακτηρίζεται η ενιαία και γενικά αποδεκτή σημασιολογική κωδικοποίηση της πληροφορίας ενός θεματικού χώρου. Η διαδικασία της μετατροπής της πληροφορίας σε σημασιολογική είναι δυνατή μέσω μιας γλώσσας περιγραφής. Αναφέραμε ήδη ότι οι πιο γνωστές γλώσσες είναι οι RDF, XML και η OWL. Οι γλώσσες αυτές επιτρέπουν τη διασύνδεση του περιεχομένου με ειδικά λεξιλόγια τα οποία ερμηνεύουν την σημασία του. Αυτά τα λεξιλόγια αποτελούν τις οντολογίες. Ο συνδυασμός του περιεχομένου με τις οντολογίες (που αναπαριστούν τις σημασιολογικές σχέσεις) επιτρέπουν την καλύτερη οργάνωση, ανάκτηση, αναζήτηση και ολοκλήρωση του περιεχομένου και συνεπώς την βελτιωμένη απόδοση των αντίστοιχων υπηρεσιών ή και την δυνατότητα κατασκευής νέων. [42]

Η τάση σήμερα είναι η αξιοποίηση του διαδικτύου για τη χρήση του από τα συστήματα διαχείρισης γνώσεων. Για παράδειγμα όπως θα δούμε στη συνέχεια, οι οντολογίες έχουν διαδικτυακά πρότυπα τα οποία χρησιμεύουν ως χώροι ονοματολογίας (namespaces) για την αναφορά κλάσεων και ιδιοτήτων των οντοτήτων ενός γνωσιακού μοντέλου. Έχοντας λοιπόν ως αναφορά μια ονοματολογία που είναι διαμοιραζόμενη στο διαδίκτυο μπορούμε να γενικεύουμε τη γνώση.

Ας δούμε μερικούς λόγους για τους οποίους πρέπει να γίνει ομαδοποίηση και γενίκευση της γνώσης [28] [42] :

- Άνθρωποι, οργανισμοί και προγράμματα πρέπει να επικοινωνούν μεταξύ τους. Υπάρχουν διαφορετικές ανάγκες και διαφορετικό υπόβαθρο τα οποία οδηγούν σε αποκλίνουσες οπτικές γωνίες και παραδοχές για πράγματα που στην ουσία είναι ίδια



μεταξύ τους. Έτσι κρίνεται απαραίτητο να γίνει κατηγοριοποίηση και ομαδοποίηση γνώσης.

- Πολλές φορές προβλήματα ομαδοποίησης προκύπτουν από έλλειψη κοινής αντίληψης. Όπως για παράδειγμα προβλήματα στην επικοινωνία μεταξύ ανθρώπων και οργανισμών ή δυσκολίες στον προσδιορισμό των απαιτήσεων και κατά συνέπεια στην ανάπτυξη των προδιαγραφών των συστημάτων.
- Άλλες φορές υπάρχουν ανομοιόμορφες μέθοδοι μοντελοποίησης, οι γλώσσες και τα εργαλεία λογισμικού περιορίζουν σοβαρά τη δια-λειτουργικότητα, την επαναχρησιμοποίηση και το διαμοιρασμό εφαρμογών, και τελικά όλα τα παραπάνω οδηγούν στο να ξαναανακαλύπτουμε τον τροχό.

Οι μέθοδοι για να καλυφθούν αυτά τα κενά είναι :

- Η εξάλειψη ή η μείωση της σύγχυσης σχετικά με τις έννοιες και τους όρους και τελικά η απόκτηση κοινής αντίληψης.
- Αυτή η κοινή αντίληψη μπορεί να αποτελέσει το ενοποιητικό πλαίσιο ανάμεσα στις διαφορετικές οπτικές γωνίες και να συμβάλει στην βελτίωση της επικοινωνίας, της δια-λειτουργικότητας, και να οδηγήσει σε πλεονεκτήματα αναφορικά με την μηχανική των συστημάτων παρέχοντας δυνατότητες επαναχρησιμοποίησης, βελτιώνοντας την αξιοπιστία, και διευκολύνοντας την ανάπτυξη προδιαγραφών.

Μια επιστημονική περιοχή στην οποία τα τελευταία χρόνια βρίσκουν εφαρμογή οι αρχές του Σημαιολογικού Ιστού και των Συστημάτων Διαχείρισης Γνώσης είναι η περιοχή της μάθησης εν γένει. Και όταν λέμε εν γένει εννοούμε τόσο τα συστήματα βασικών γνώσεων, όσο και τα συστήματα εκπαίδευσης σε πιο εξειδικευμένη γνώση. Η παρούσα εργασία ασχολείται με την μάθηση και από εδώ και πέρα θα επικεντρώσουμε το ενδιαφέρον μας σε αυτήν και πιο ειδικά στην ηλεκτρονική μάθηση.

#### **4. Συστήματα e-learning με context awareness**

Δεν υπάρχουν πολλά έργα-συστήματα τα οποία προσπαθούν να λειτουργήσουν στα πλαίσια της διαχείρισης γνώσης με στόχευση την εκπαιδευτική διαδικασία. Τα σημαντικότερα συστήματα που βρήκαμε κατά τη διάρκεια της έρευνάς μας και τα οποία είναι κοντά στο αντικείμενο της ηλεκτρονικής μάθησης με γνώση των συνθηκών πλαισίου είναι τα [16], [27], [30], [8], [12], [13]. Σε κάποιες περιπτώσεις έχουμε σαφές γνωσιακό αντικείμενο το οποίο



προσαρμόζεται ανάλογα με την τοποθεσία [16]. Επίσης υπάρχουν περιπτώσεις όπου η διαχείριση γνώσης έχει να κάνει με την σωστή καθοδήγηση μέσω πολυμεσικών αντικειμένων [30]. Η τάση που βλέπουμε να επικρατεί σε όλες αυτές τις προσπάθειες είναι η οργάνωση διαφόρων αντικειμένων μάθησης (LOs - Learning Objects) [8], [12], [13], για τα οποία αντικείμενα, η τάση είναι να βρίσκονται αποθηκευμένα σε κεντρικούς εξυπηρετές. Αυτά τα αντικείμενα μάθησης θα πρέπει να είναι διαθέσιμα οπουδήποτε και οποτεδήποτε. Αυτό φυσικά προϋποθέτει την ύπαρξη κινητών δικτύων τα οποία θα έχουν σύνδεση με τον εξυπηρετή και θα τραβούν δεδομένα ανάλογα με κάποιες συνθήκες. Τέτοιες συνθήκες μπορεί να είναι οι γεωγραφικές συντεταγμένες που μπορεί να βρίσκεται ο χρήστης στα πλαίσια της κίνησής του σε έναν εσωτερικό χώρο [30] ή στα πλαίσια της κίνησής του μέσα στις συνοικίες μιας πόλης [16]. Επίσης πρέπει να αναφέρουμε ότι τα αντικείμενα μάθησης μπορεί να είναι κάποιες πολυμεσικές αναπαραστάσεις (ήχοι, videos, εικόνες [30]), ή κάποιες πληροφορίες με τοπικιστικό χαρακτήρα [16], ή κάποια αόριστα αντικείμενα που μπορεί να έχουν οποιοδήποτε χαρακτήρα εκπαιδευτικής διαδικασίας [12]. Πράγματι παρατηρήσαμε ότι εν γένει ([12], [13]) δεν ενδιαφέρει αυτό κάθε αυτό το αντικείμενο της μαθησιακής διαδικασίας αλλά η αρχιτεκτονική υλοποίησης του έργου. Ούτε ενδιαφέρει σε όλες τις περιπτώσεις η ενσωμάτωση διαδικασιών συμπερασμού.

Στη δική μας περίπτωση, δίνουμε έμφαση στην εκπαιδευτική διαδικασία και τον τρόπο με το οποίο μπορεί να επηρεαστεί από εξωτερικούς παράγοντες. Δημιουργούμε ένα σύστημα με πραγματικά εκπαιδευτικά αντικείμενα (με έμφαση την εξέταση-αξιολόγηση), τα οποία σε πραγματικές συνθήκες (με έμφαση τον θόρυβο και την τοποθεσία), είναι προσαρμοστικά. Για να συμβεί αυτό διασυνδέουμε τεχνολογίες αιχμής (σημασιολογικός ιστός, οντολογίες, συστήματα κρίσης και συμπερασμού) με ερωτήματα και κανόνες κάτω από μια πλατφόρμα ελεύθερου λογισμικού (Java).

## 5. Οντολογίες

Μια οντολογία είναι [44], [2], [7], [28], [27] μια τυπική (formal), κατηγορηματική (explicit) προδιαγραφή μιας διαμοιρασμένης (shared) εννοιολογικής αντίληψης (conceptualization). Με τον όρο «εννοιολογική αντίληψη» (conceptualization) αναφερόμαστε σε ένα αφηρημένο μοντέλο φαινομένων του κόσμου στο οποίο έχουν προσδιοριστεί οι έννοιες που σχετίζονται με τα φαινόμενα αυτά. Με τον όρο «κατηγορηματική» (explicit) αναφερόμαστε στο είδος των εννοιών που χρησιμοποιούνται, και οι περιορισμοί που αφορούν την χρήση αυτών των εννοιών είναι προσδιορισμένα με σαφήνεια. Με τον όρο «αυστηρή» (formal) αναφερόμαστε



στο γεγονός ότι η οντολογία πρέπει να είναι μηχανικά αναγνώσιμη. Τέλος με τον όρο «διαμοιρασμένη» (shared) αναφερόμαστε στο γεγονός ότι η οντολογία πρέπει να αποτυπώνει γνώση κοινής αποδοχής στα πλαίσια μιας κοινότητας.

Η λέξη οντολογία είναι προφανώς Ελληνική, και στην Ελληνική φιλοσοφία αναφέρεται στην έννοια της ύπαρξης. Στον επιστημονικό χώρο του διαμοιρασμού της γνώσης ο όρος σημαίνει τον προσδιορισμό μιας αντίληψης. Και αυτό είναι κάτι διαφορετικό από τη λέξη, έτσι όπως χρησιμοποιείται στην Ελληνική φιλοσοφία. «Προσδιορισμός μιας αντίληψης» σημαίνει την περιγραφή των εννοιών και των σχέσεων οι οποίες υπάρχουν-ισχύουν για κάποιο παράγοντα ή πλήθος παραγόντων (αναφέρεται ως agent στην πηγή). Ο προσδιορισμός είναι συνεπής με τη χρήση της οντολογίας ως ένα σύνολο εννοιών-προσδιορισμών, αλλά σε πιο γενικό ύφος.

Οι Οντολογίες χρησιμοποιούνται όχι μόνο για να αναπαραστήσουν-περιγράψουν ένα πεδίο ενδιαφέροντος, αλλά και για να καθορίσουν-ορίσουν έννοιες, να περιγράψουν σχέσεις ανάμεσά τους και επίσης να εισάγουν στιγμιότυπα (individuals-instances).

Μια οντολογία περιλαμβάνει ένα λεξιλόγιο όρων και κάποιας μορφής προδιαγραφές για τη σημασία τους. Σχετικά με τον βαθμό της τυπικότητας της αναπαράστασης μιας οντολογίας αυτή μπορεί να είναι [30][27]:

- Άτυπη (informal), εκφρασμένη σε μια φυσική γλώσσα.
- Ημι-άτυπη (semi-informal): για παράδειγμα διατυπωμένη σε ένα περιορισμένο και δομημένο υποσύνολο κάποιας φυσικής γλώσσας.
- Ημι-τυπική (semi-formal): διατυπωμένη σε μια τεχνητή και αυστηρά ορισμένη γλώσσα.
- Αυστηρά τυπική (rigorously formal): ορισμοί όρων με αυστηρή σημασιολογία, θεωρήματα και αποδείξεις ιδιοτήτων όπως η ορθότητα (soundness) και η πληρότητα (completeness).

Τα βασικά συστατικά μιας Οντολογίας είναι τα εξής: (χρησιμοποιούμε την ονοματολογία που βρίσκουμε σε ένα διάσημο εργαλείο διαχείρισης οντολογιών το protégé [31], το οποίο χρησιμοποιούμε και εμείς στην παρούσα εργασία)



- Οι κλάσεις (classes) οι οποίες είναι έννοιες που σχετίζονται με ένα πεδίο ή κάποιες εργασίες, οι οποίες είναι συνήθως οργανωμένες σε κάποιο ταξινομικό σύστημα. Για παράδειγμα σε μια οντολογία που αφορά το πανεπιστήμιο: ο «μαθητής» και ο «καθηγητής» αποτελούν δύο κλάσεις.
- Οι Ιδιότητες (properties) οι οποίες είναι ένας τύπος αλληλεπίδρασης μεταξύ εννοιών ενός πεδίου. Οι ιδιότητες στη διεθνή βιβλιογραφία και ορολογία των οντολογιών συναντώνται και ως σχέσεις (relations).
- Συναρτήσεις (functions) οι οποίες είναι μια ειδική περίπτωση σχέσης στην οποία το  $n$ -οστό στοιχείο της σχέσης προσδιορίζεται μοναδικά από τα  $n-1$  προηγούμενα στοιχεία. Για παράδειγμα η τιμή-μεταχειρισμένου-αυτοκινήτου μπορεί να προσδιορίζεται σαν συνάρτηση της αρχικής τιμής του καινούριου αυτοκινήτου, του μοντέλου του αυτοκινήτου, των χαρακτηριστικών του αυτοκινήτου και των χιλιομέτρων που έχει διανύσει.
- Αξιώματα (axioms) τα οποία αναπαριστούν προτάσεις που είναι πάντα αληθείς. Για παράδειγμα αν ο  $\Phi$  είναι δευτεροετής φοιτητής τότε μπορεί να εγγραφεί στο επιλεγόμενο μάθημα  $M$ .
- Στιγμιότυπα (instances ή individuals), τα οποία αναπαριστούν συγκεκριμένα στοιχεία. Για παράδειγμα ο φοιτητής με το όνομα Νίκος είναι ένα στιγμιότυπο της κλάσης «φοιτητής».

Εν κατακλείδι θα μπορούσαμε να πούμε ότι η μια Οντολογία παρουσιάζει ένα πεδίο ενδιαφέροντος, προσδιορίζει έννοιες, περιγράφει σχέσεις ανάμεσα σε αυτές, και τέλος εισάγει στιγμιότυπα.

### 5.1. Κατηγορίες Οντολογιών

Μερικές χαρακτηριστικές κατηγορίες οντολογιών είναι οι ακόλουθες [44]:

- Οντολογίες πεδίου ορισμού (domain ontologies): αναπαριστούν γνώση γύρω από ένα συγκεκριμένο πεδίο (π.χ. ιατρική, ηλεκτρονικά κλπ).
- Οντολογίες μεταδεδομένων (metadata ontologies): παρέχουν ένα λεξιλόγιο για την περιγραφή του περιεχομένου ηλεκτρονικά διαθέσιμης πληροφορίας.



- Γενικές ή κοινές οντολογίες (generic or common sense ontologies): στοχεύουν στο να αποτυπώσουν γενική γνώση γύρω από τον κόσμο, παρέχοντας βασικές έννοιες όπως ο χρόνος, ο χώρος, τα συμβάντα κλπ.
- Οντολογίες αναπαράστασης (representational ontologies): παρέχουν οντότητες αναπαράστασης χωρίς να προσδιορίζουν τη συγκεκριμένο αναπαριστούν
  - ο π.χ. Frame Ontology (Gruber 1993): ορίζει έννοιες όπως frames, slots, slot constraints κ.λ.π.
- Οντολογίες μεθοδολογίας ή εργασιών (method or task ontologies): παρέχουν όρους που αναφέρονται σε συγκεκριμένες εργασίες (π.χ. διάγνωση κ.λ.π.)

Σε αυτό το σημείο να αναφερθούμε στην έννοια της ταξινόμιας, η οποία είναι σημαντική στο χώρο των οντολογιών. Ταξινόμια είναι η ανάγκη περιγραφής για οτιδήποτε, στη βάση ενός ιεραρχικού μοντέλου. Είναι η πρακτική της δημιουργίας ιεραρχίας σε οτιδήποτε θέλουμε να περιγράψουμε αναλυτικά. Αυτό το μοντέλο περιγραφής έχει εφαρμογή σε όλες σχεδόν τις εκφάνσεις της ζωής. Στις επιχειρήσεις, την κυβέρνηση, το στρατό, την εκπαίδευση, τη φύση, παντού. Έτσι οτιδήποτε θέλουμε να περιγράψουμε με μορφή οντολογίας θα πρέπει να είναι καλά ιεραρχημένο. Να έχει όπως λέμε σωστή ταξινόμια. Τα ανώτατα κλιμάκια ιεράρχησης είναι οι κλάσεις, ενώ όσο βαθύτερα αναλύεται μια αρχική οντότητα, προχωράμε σε υποκλάσεις.

## 5.2. Εφαρμογές των Οντολογιών

Όπως είδαμε οι οντολογίες είναι πλέον ένα ισχυρότατο εργαλείο για τις εφαρμογές που απαιτούν την οργάνωση δεδομένων και την εξαγωγή συμπερασμάτων με βάση κάποιους κανόνες. Ας δούμε κάποιες κατηγορίες εφαρμογών που χρειάζονται τη χρήση οντολογιών [28]:

- Επικοινωνία μεταξύ ανθρώπων και οργανισμών.
  - ο Παρέχουν ολοκληρωμένο πλαίσιο εννοιών και ορολογίας μεταξύ ανθρώπων με διαφορετικές ανάγκες και οπτικές γωνίες στα πλαίσια ενός οργανισμού. Διευκολύνουν την επικοινωνία των ανθρώπων στα πλαίσια του οργανισμού.
- Δια-λειτουργικότητα (inter-operability) μεταξύ συστημάτων
  - ο Διάφοροι χρήστες χρειάζεται να ανταλλάσσουν δεδομένα ή χρησιμοποιούν διαφορετικά πακέτα λογισμικού.



- Χρήση οντολογιών για την υποστήριξη μετάφρασης μεταξύ διαφορετικών γλωσσών και αναπαραστάσεων.
- Μηχανική Συστημάτων (systems engineering)
  - Προδιαγραφές
  - Επαναχρησιμοποίηση τμημάτων
  - Αξιοπιστία

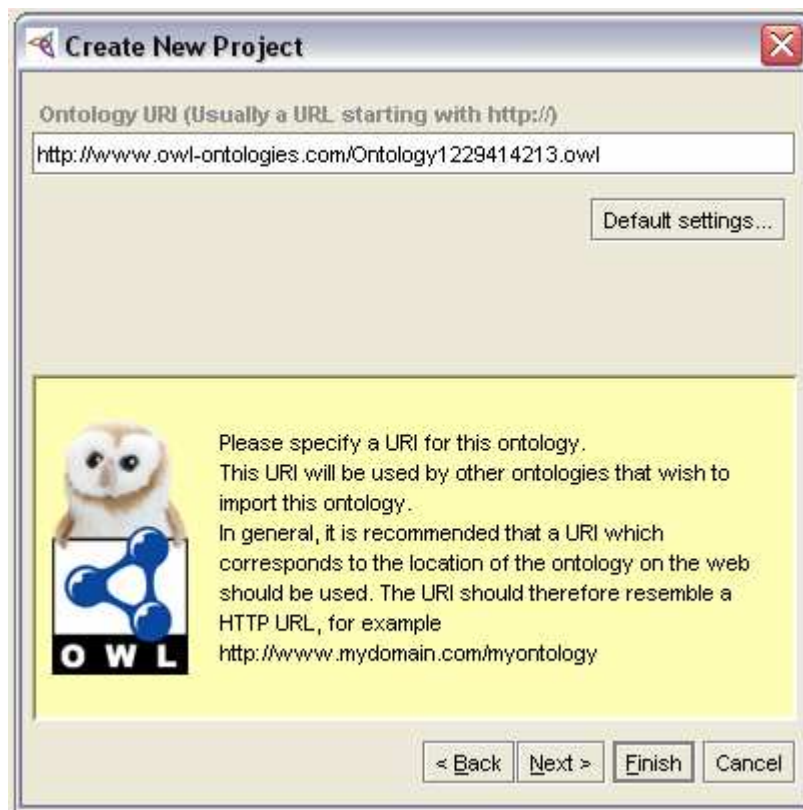
## 6. Εργαλεία ανάπτυξης οντολογιών με γραφικό περιβάλλον

Κατά τη διάρκεια της έρευνάς μας ανακαλύψαμε αρκετά εργαλεία τα οποία χρησιμεύουν στην ανάπτυξη και διαχείριση οντολογιών. Υπάρχουν καθαρές γλώσσες ορισμού οντολογιών σε γραμμή εντολών, αλλά υπάρχουν και πιο σύγχρονα εργαλεία (κυρίως Java based) τα οποία διαθέτουν παραθυρικό περιβάλλον. Με τα τελευταία να υπερτερούν σαφώς των πρώτων λόγω της ευκολίας και της εποπτείας με εικόνα της διαχείρισης. Μερικά τέτοια εργαλεία είναι τα Methontology, On-To-Knowledge methodology, AFM:Activity-First Method κλπ. Μια πολύ καλή μελέτη για τα εργαλεία αυτά μπορεί κανείς να βρει στο [14]. Τα σημαντικότερα εργαλεία ανάπτυξης οντολογιών με γραφικό περιβάλλον είναι : OntoEdit, WebODE, Protégé και Hozo. Το protégé δείχνει να έχει «τα σκήπτρα» στα περιβάλλοντα ανάπτυξης, και για το λόγο αυτό στα επόμενα θα ασχοληθούμε με αυτό. Θα δούμε κάποια αναλυτικά στοιχεία και θα το περιγράψουμε σε μεγάλο βαθμό.

### 6.1. Το Protégé

Το Protégé, το οποίο έχει φτάσει στην έκδοση 4, δείχνει να έχει τα σκήπτρα σε αυτού του είδους τα λογισμικά ανάπτυξης και διαχείρισης οντολογιών. Είναι μία ολοκληρωμένη εφαρμογή διαχείρισης, υλοποιημένη σε Java, που προσφέρει επιπλέον και API βιβλιοθήκες (protégé και protégé-owl) για να μπορεί να εισάγει και να συνεργάζεται με άλλα προγράμματα, μέσω Java. Για παράδειγμα μέσα από API συναρτήσεις μπορεί να συνεργασθεί με μηχανές συμπερασμού, όπως θα δούμε και στη συνέχεια. Επιπρόσθετα με το απλό Protégé μπορεί κανείς να βρει και το περιβάλλον OWL-Protégé, με τη χρήση του οποίου καθίσταται δυνατή η δημιουργία μίας OWL οντολογίας, η επεξεργασία των στοιχείων (κλάσεων, στιγμιότυπων, ιδιοτήτων) της οντολογίας και η εκτέλεση ερωτημάτων πάνω στην οντολογία. Αρκετές από αυτές τις λειτουργίες συνοδεύονται από ολοκληρωμένες γραφικές διεπαφές (GUIs), τις οποίες ο χρήστης μπορεί να χρησιμοποιήσει χωρίς αλλαγές.

Στην *Εικόνα 1*, βλέπουμε ένα στιγμιότυπο του μάγου (wizard) του protégé μέσω του οποίου συσχετίζουμε την υπό κατασκευή οντολογία μας, με μια πρότυπη οντολογία OWL.



Εικόνα 1: Στιγμιότυπο από τη δημιουργία νέας οντολογίας με πρότυπο

Στη συγκεκριμένη εργασία αποφασίσαμε τελικά να χρησιμοποιήσουμε το Protégé, για το σύνολο των λόγων που αναπτύξαμε στην προηγούμενη παράγραφο. Συνοψίζοντας να αναφέρουμε ότι διαθέτει μια τυποποιημένη μεθοδολογία δημιουργίας οντολογιών περιγραφής της οποίας μπορεί κανείς να βρει στην επίσημη ιστοσελίδα του έργου [38], αλλά και σε ένα πλήθος άλλων ερευνητικών έργων. Το γραφικό περιβάλλον του Protégé είναι πολύ φιλικό και επεξηγηματικό. Επιπλέον μπορεί να εξάγει ολόκληρη την οντολογία μέσω ενός μηχανισμού σε διάφορες μορφές και αναπαραστάσεις. Μπορεί να εξάγει την οντολογία σε μορφή XML, RDF, σε εικόνα jpg μέσω visualization, αλλά και σε έναν πολύ λειτουργικό ιστότοπο.

Επίσης έχουν αναπτυχθεί εργαλεία που εξετάζουν τη συνοχή-σταθερότητα (consistency) της οντολογίας και στην έκδοση 4 ενσωματώνει 2 τέτοια εργαλεία. Τα FaCT++ [34] [22], και Pellet 1.5 [37]. Το consistency είναι μια διαδικασία μέσα από την οποία βλέπουμε αν οι κλάσεις και οι ιδιότητες είναι καλά ορισμένες και δεν παρουσιάζουν αλληλεξαρτήσεις που κάνουν ασθενή την υπόστασή της. Ένα από τα βασικά εργαλεία του protégé είναι το OWL plug in που έχει και με το οποίο το περιβάλλον του protégé μεταμορφώνεται κατάλληλα,

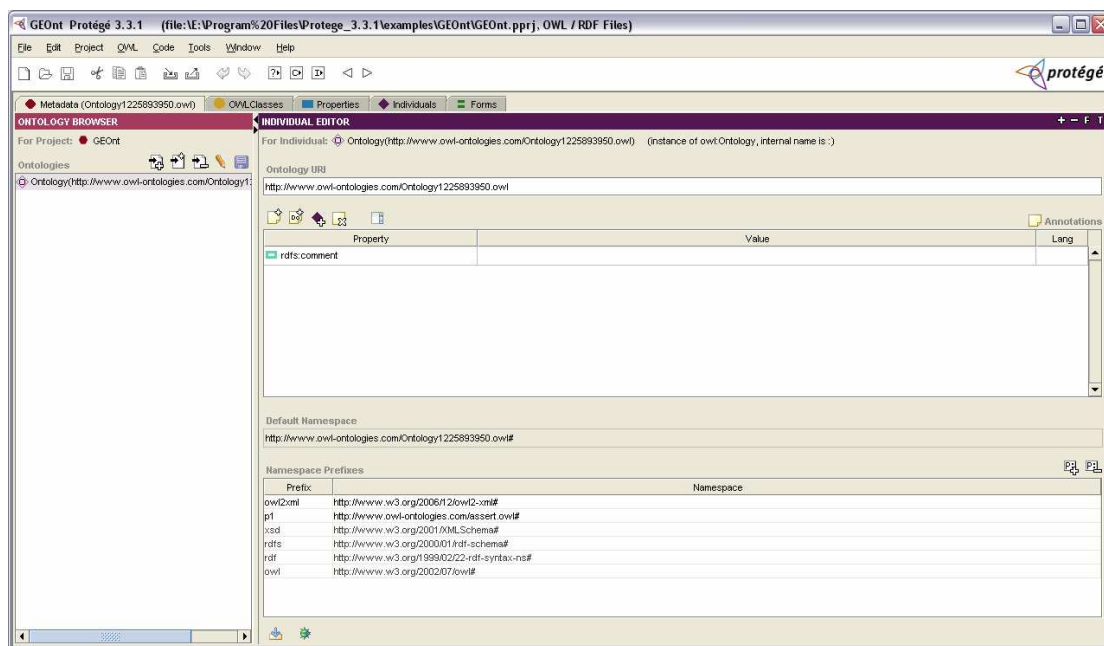




ώστε να έχει online υποστήριξη από την γλώσσα OWL, την οποία περιγράφουμε αμέσως πιο κάτω. Επίσης έχουν αναπτυχθεί μηχανές συμπερασμού οι οποίες συνεργάζονται με την τεχνολογία Protégé-OWL ώστε να μπορούν να θέτουν αλήθειες (Facts) και κανόνες (Rules) και να εξάγουν συμπεράσματα (Bossam [39]). Τέλος στο Protégé μπορεί να εισαχθούν (import) οντολογίες γραμμένες σε μια άλλη γλώσσα και σε άλλη μορφή (πχ RDF και XML), χωρίς να χαθεί κάποιο μέρος της πληροφορίας. Σε τέτοιες περιπτώσεις γίνεται εισαγωγή ακόμα και σχολίων και επεξηγήσεων των κλάσεων.

## 6.2. Δημιουργία μιας οντολογίας στο Protégé

Για τη δημιουργία μιας οντολογίας δεν υπάρχουν κανόνες. Πιο σημαντική είναι η εμπειρία. Αρχικά πρέπει να προσδιορίσουμε τις βασικές έννοιες. Επίσης πρέπει να συνοδεύουμε τον ορισμό των εννοιών με κείμενο που τις περιγράφει. Περιγράφουμε τις προαναφερθείσες έννοιες με μια κατάλληλη γλώσσα.



Εικόνα 2 : Η Οθόνη όπου φαίνονται τα URI της ονοματολογίας

Οι βασικές έννοιες ορίζονται μέσα από τον ορισμό κλάσεων στην κατάλληλη καρτέλα OWL classes του protégé (Εικόνα 2). Υπάρχει η δυνατότητα να ορίσουμε κλάσεις και υποκλάσεις ανάλογα αν οι έννοιες που θέλουμε να περιγράψουμε είναι κύριες ή μπορούν να ενταχθούν κάτω από μια πιο κύρια έννοια. Για παράδειγμα αν μια κλάση είναι ένα κτίριο, τότε



υποκλάσεις μπορεί να είναι οι όροφοι και ακόμη ένα επίπεδο υποκλάσης, που θα βρίσκεται κάτω από την υποκλάση όροφος θα μπορούσε να είναι το διαμέρισμα.

Στη συνέχεια στην καρτέλα properties μπορούμε να ορίσουμε τις σχέσεις που συνδέουν τις κλάσεις. Για παράδειγμα μια σχέση που μπορεί να συνδέει μια κλάση **μάθημα** με μια άλλη κλάση **μαθητής**, μπορεί να είναι η σχέση **δήλωσε**. Και έτσι η πλήρης έκφραση που συνδέει τις δύο έννοιες θα είναι : ο μαθητής δήλωσε το μάθημα.

Τέλος μπορούμε να δηλώσουμε στιγμιότυπα της οντολογίας στην καρτέλα individuals. Τα στιγμιότυπα είναι πραγματικές οντότητες, αντικείμενα, έννοιες που αντιστοιχούν στις δημιουργηθείσες κλάσεις. Για παράδειγμα ένα στιγμιότυπο της κλάσης **μαθητής** είναι ο Πέτρος. Επίσης ένα στιγμιότυπο της κλάσης **μάθημα** είναι η Φυσική.

Στην παράγραφο 8 περιγράφουμε αναλυτικά τη δημιουργία της οντολογίας GEOnt την οποία και χρησιμοποιούμε για τις ανάγκες του συστήματός μας. Πρόκειται για μια πολύ απλή οντολογία η οποία δεν ευελπιστεί να περιγράψει επακριβώς κάποιες συγκεκριμένες εκπαιδευτικές δραστηριότητες, παρά μόνο να χρησιμοποιηθεί για τις ανάγκες της εφαρμογής μας. Στην ίδια παράγραφο, παραθέτουμε μια σειρά εικόνων οι οποίες κάνουν πιο σαφή τη διαδικασία δημιουργίας μιας οντολογίας.

## 7. Η γλώσσα OWL

Η OWL (Web Ontology Language) έχει σχεδιαστεί για να χρησιμοποιείται από εφαρμογές οι οποίες έχουν ως σκοπό να επεξεργάζονται πληροφορίες [35]. Διευκολύνει τη δυνατότητα διερμηνεύσης των περιεχομένων του παγκόσμιου ιστού από κατάλληλες μηχανές που μπορούν να τη χρησιμοποιούν. Η δυνατότητα αποτύπωσης των περιεχομένων είναι δε μεγαλύτερη από αυτή που προσφέρουν οι XML, RDF και RDFS (RDF Schema), οι οποίες θεωρούνται μαζί με την OWL οι κύριες γλώσσες περιγραφής περιεχομένου για Σημασιολογικό Ιστό [27],[34],[35]. Όπως ισχυρίζονται οι δημιουργοί της [34] [35], περιλαμβάνει επιπλέον γλωσσάρι από τις δύο άλλες προαναφερόμενες γλώσσες, και για το λόγο αυτό έχει περισσότερη δύναμη στην περιγραφή περιεχομένου. Η OWL έχει τρεις διαφορετικές μορφές (ή υπογλώσσες-sublanguages όπως αναφέρονται). Τις OWL Lite, OWL DL, και OWL Full. Πιο συγκεκριμένα [27], [24]:

- Η OWL Lite προορίζεται για χρήστες που χρειάζονται πρωτίστως μία ιεραρχία ταξινόμησης και απλούς περιορισμούς. Για παράδειγμα, ενώ υποστηρίζει περιορισμούς πληθικότητας, επιτρέπει τιμές πληθικότητας μόνο 0 και 1. Η



δημιουργία και ο χειρισμός μίας οντολογίας με αυτή τη γλώσσα είναι σαφώς πιο εύκολα, σε σχέση με τις άλλες δύο γλώσσες, αφού διαθέτει λιγότερο πολύπλοκο λεξιλόγιο και σύνταξη. Παράλληλα, όμως, προσφέρει μειωμένη εκφραστικότητα σε σχέση με τις άλλες δύο γλώσσες.

- Η OWL DL προορίζεται για χρήστες που χρειάζονται μέγιστη εκφραστικότητα, διατηρώντας παράλληλα υπολογιστική πληρότητα (όλα τα συμπεράσματα είναι υπολογίσιμα) και αποκρισιμότητα (όλοι οι υπολογισμοί τελειώνουν σε πεπερασμένο χρονικό διάστημα). Περιλαμβάνει όλα τα στοιχεία της γλώσσας OWL, τα οποία, όμως, μπορούν να χρησιμοποιηθούν κάτω από ορισμένους περιορισμούς (για παράδειγμα, μία κλάση μπορεί να είναι υποκλάση πολλών κλάσεων, όχι όμως στιγμιότυπο μίας άλλης κλάσης).
- Η OWL Full προορίζεται για χρήστες που χρειάζονται μέγιστη εκφραστικότητα και τη συντακτική ελευθερία της RDF, χωρίς να ενδιαφέρονται για την διατήρηση της υπολογιστικής πληρότητας και αποκρισιμότητας. Για παράδειγμα, μία κλάση μπορεί υπολογίζεται ως συλλογή στιγμιότυπων, αλλά ταυτόχρονα και η ίδια ως στιγμιότυπο. Η OWL Full επιτρέπει σε μία οντολογία να επεκτείνει τη σημασία προκαθορισμένου RDF ή OWL λεξιλογίου. Λόγω της μεγάλης εκφραστικότητας, η γλώσσα αυτή αφήνει λίγα περιθώρια για πλήρη συλλογιστική υποστήριξη.

Για τις τρεις αυτές υπογλώσσες της OWL ισχύουν οι παρακάτω προτάσεις:

- Κάθε νόμιμη OWL Lite οντολογία είναι και νόμιμη OWL DL οντολογία.
- Κάθε νόμιμη OWL DL οντολογία είναι και νόμιμη OWL Full οντολογία.
- Κάθε έγκυρο OWL Lite συμπέρασμα είναι και έγκυρο OWL DL συμπέρασμα.
- Κάθε έγκυρο OWL DL συμπέρασμα είναι και έγκυρο OWL Full συμπέρασμα.

Η OWL Full μπορούμε να πούμε ότι είναι μία επέκταση της RDF, ενώ οι OWL Lite και OWL DL επεκτάσεις μίας περιορισμένης όψης της RDF. Για αυτό, ισχύουν επιπλέον τα παρακάτω:

- Κάθε OWL (Lite, DL, Full) έγγραφο είναι και RDF έγγραφο.
- Κάθε RDF έγγραφο είναι και OWL Full έγγραφο.
- Μόνο ορισμένα RDF έγγραφα μπορούν να είναι και OWL Lite OWL ή DL έγγραφα.



Η επιλογή ανάμεσα στις τρεις εκδοχές της OWL καθορίζεται από τις απαιτήσεις που έχει ο χρήστης από τη γλώσσα. Η επιλογή ανάμεσα στην OWL Lite και στην OWL DL εξαρτάται από το πόσο αναγκαίες είναι οι πιο εκφραστικές δομές που χρησιμοποιεί η OWL DL. Η επιλογή ανάμεσα στην OWL DL και στην OWL Full εξαρτάται από την ανάγκη για χρήση των διευκολύνσεων όσον αφορά τη μοντελοποίηση που προσφέρει η OWL Full (για παράδειγμα δημιουργία κλάσεων από κλάσεις, ενσωμάτωση ιδιοτήτων σε κλάσεις).

Ακολουθούν τα βασικά δομικά στοιχεία της OWL. Αρχικά, δίνονται εκείνα τα στοιχεία της OWL Lite που προέρχονται από την RDFS [27], [24]:

- **Class:** Μία κλάση (Class) ορίζει μία ομάδα στιγμιότυπων (αντικειμένων) που μοιράζονται κάποιες κοινές ιδιότητες. Οι κλάσεις μπορούν να οργανωθούν σε μία ιεραρχία χρησιμοποιώντας τη `subClassOf`. Υπάρχει μία προκαθορισμένη κλάση, η `Thing`, που περιέχει όλα τα στιγμιότυπα και είναι υπερκλάση όλων των κλάσεων και η κλάση `Nothing`, που δεν περιέχει κανένα στιγμιότυπο και είναι υποκλάση όλων των κλάσεων.
- **rdfs:subClassOf:** Ιεραρχίες κλάσεων μπορούν να δημιουργηθούν με τη δήλωση ότι μία κλάση είναι υποκλάση (`subClassOf`) μίας άλλης κλάσης.
- **rdfs:Property:** Μία ιδιότητα (Property) δηλώνει μία σχέση μεταξύ στιγμιότυπων ή μία σχέση ανάμεσα σε στιγμιότυπο και μία τιμή δεδομένου. Οι ιδιότητες της πρώτης κατηγορίας ονομάζονται ιδιότητες αντικειμένων (`ObjectProperty`), ενώ της δεύτερης κατηγορίας ιδιότητες τύπων δεδομένων (`DatatypeProperty`). Και οι δύο είναι υποκλάσεις της `Property`.
- **rdfs:subPropertyOf:** Ιεραρχίες ιδιοτήτων μπορούν να δημιουργηθούν με τη δήλωση ότι μία ιδιότητα είναι υπό-ιδιότητα (`subPropertyOf`) μίας άλλης ιδιότητας.
- **rdfs:domain:** Το πεδίο ορισμού (`domain`) μίας ιδιότητας περιορίζει τα αντικείμενα στα οποία μπορεί να εφαρμοστεί η ιδιότητα. Αν μία ιδιότητα δηλώνει μία σχέση ενός στιγμιότυπου με ένα άλλο κα έχει ως πεδίο ορισμού μία κλάση, τότε το πρώτο στιγμιότυπο πρέπει να ανήκει σε αυτήν την κλάση. Το πεδίο ορισμού είναι ένας γενικευμένος περιορισμός, αφού χαρακτηρίζει μία ιδιότητα γενικά και όχι μία ιδιότητα που συνδέεται με μία συγκεκριμένη κλάση.
- **rdfs:range:** Το πεδίο τιμών (`range`) μίας ιδιότητας περιορίζει τα αντικείμενα τα οποία μπορούν να αποτελούν τιμή της ιδιότητας. Αν μία ιδιότητα δηλώνει μία σχέση ενός στιγμιότυπου με ένα άλλο κα έχει ως πεδίο τιμών μία κλάση, τότε το δεύτερο



στιγμιότυπο πρέπει να ανήκει σε αυτήν την κλάση. Το πεδίο τιμών είναι επίσης γενικευμένος περιορισμός.

- Individual: Το άτομο (Individual) είναι ένα στιγμιότυπο μίας κλάσης. Οι ιδιότητες μπορούν να χρησιμοποιηθούν για να συσχετίσουν ένα άτομο με κάποιο άλλο.

Στη συνέχεια, δίνονται εκείνα τα στοιχεία της OWL Lite που σχετίζονται με ισότητα και ανισότητα [27], [24]:

- `equivalentClass`: Ισοδύναμες (`equivalentClass`) μπορούν να δηλωθούν δύο κλάσεις που έχουν τα ίδια στιγμιότυπα. Με αυτόν τον τρόπο δημιουργούνται συνώνυμες κλάσεις.
- `equivalentProperty`: Ισοδύναμες (`equivalentProperty`) μπορούν να δηλωθούν δύο ιδιότητες που συνδέουν ένα στιγμιότυπο με την ίδια ομάδα στιγμιότυπων. Με αυτόν τον τρόπο δημιουργούνται συνώνυμες ιδιότητες.
- `sameAs`: Ίδια (`sameAs`) μπορούν να δηλωθούν δύο στιγμιότυπα με σκοπό να δημιουργηθούν περισσότερα από ένα ονόματα αναφοράς στο ίδιο στιγμιότυπο.
- `differentFrom`: Ένα στιγμιότυπο μπορεί να χαρακτηριστεί διαφορετικό (`differentFrom`) από ένα άλλο στιγμιότυπο για να δηλωθεί ρητά ότι τα δύο στιγμιότυπα δεν είναι ίδια. Αυτό είναι πολύ σημαντικό σε γλώσσες όπως η OWL, οι οποίες δεν υποθέτουν ότι κάθε στιγμιότυπο έχει ένα και μόνο όνομα.
- `AllDifferent`: Ένα σύνολο από στιγμιότυπα μπορεί να δηλωθεί `AllDifferent`, δηλαδή ότι κάθε στιγμιότυπο είναι διαφορετικό από τα υπόλοιπα. Αυτό είναι πολύ σημαντικό όταν απαιτείται από ένα σύνολο στιγμιότυπων να έχει το καθένα ένα και μόνο όνομα.

Ακολουθούν εκείνα τα στοιχεία της OWL Lite που σχετίζονται με τις ιδιότητες και τις τιμές τους [27], [24]:

- `inverseOf`: Μία ιδιότητα μπορεί δηλωθεί ως η αντίστροφη (`inverseOf`) μίας άλλης ιδιότητας. Σε αυτήν την περίπτωση αν το στιγμιότυπο X συνδέεται μέσω μίας ιδιότητας με το στιγμιότυπο Y, τότε το Y συνδέεται με το X μέσω της αντίστροφης ιδιότητας.
- `TransitiveProperty`: Μία ιδιότητα μπορεί δηλωθεί ως η μεταβατική (`TransitiveProperty`). Σε αυτήν την περίπτωση αν τα ζεύγη (X, Y) και (Y, Z) είναι



στιγμιότυπα μίας μεταβατικής ιδιότητας, τότε και το ζεύγος  $(X, Z)$  είναι στιγμιότυπο της ίδιας ιδιότητας. Οι OWL Lite και OWL DL επιβάλλουν τον περιορισμό ότι οι μεταβατικές ιδιότητες (και οι υπερ-ιδιότητες τους) δεν μπορούν να έχουν περιορισμό μέγιστης πληθικότητας 1. Σε διαφορετική περίπτωση δεν θα ίσχυε η αποκρισιμότητα της γλώσσας.

- **SymmetricProperty:** Μία ιδιότητα μπορεί δηλωθεί ως η συμμετρική (SymmetricProperty). Σε αυτήν την περίπτωση αν το ζεύγος  $(X, Y)$  είναι στιγμιότυπο μίας συμμετρικής ιδιότητας, τότε και το ζεύγος  $(Z, X)$  είναι στιγμιότυπο της ίδιας ιδιότητας.
- **FunctionalProperty:** Μία ιδιότητα μπορεί δηλωθεί ως η μονότιμη (FunctionalProperty) στην περίπτωση που δεν έχει παραπάνω από μία τιμή για κάθε στιγμιότυπο.
- **InverseFunctionalProperty:** Μία ιδιότητα μπορεί δηλωθεί ως η αντιστρόφως μονότιμη (InverseFunctionalProperty) στην περίπτωση που η αντίστροφη της είναι μονότιμη.

Στη συνέχεια, δίνονται εκείνα τα στοιχεία της OWL Lite που σχετίζονται περιορισμούς που καθορίζουν πώς μία ιδιότητα μπορεί να χρησιμοποιηθεί από τα στιγμιότυπα μίας κλάσης [27], [24]:

- **allValuesFrom:** Ο περιορισμός allValuesFrom (όλες οι τιμές από) τίθεται σε μία ιδιότητα, αναφερόμενος σε μία κλάση. Αυτό σημαίνει ότι η ιδιότητα, για τη συγκεκριμένη κλάση, έχει ένα περιορισμένο πεδίο τιμών. Έτσι, αν ένα στιγμιότυπο  $X$  της κλάσης συνδέεται με τη συγκεκριμένη ιδιότητα με ένα στιγμιότυπο  $Y$ , τότε το  $Y$  ανήκει στην κλάση που προσδιορίζεται από τον αντίστοιχο περιορισμό του πεδίου τιμών.
- **someValuesFrom:** Ο περιορισμός someValuesFrom (όλες οι τιμές από) τίθεται σε μία ιδιότητα, αναφερόμενος σε μία κλάση. Αυτό σημαίνει ότι η ιδιότητα, για τη συγκεκριμένη κλάση, για μία τουλάχιστον από τις τιμές της, έχει ένα περιορισμένο πεδίο τιμών.
- **minCardinality:** Ο περιορισμός ελάχιστης πληθικότητας (minCardinality) τίθεται σε μία ιδιότητα, αναφερόμενος σε μία κλάση. Αν για μία ιδιότητα έχουμε minCardinality 1, τότε κάθε στιγμιότυπο της κλάσης θα πρέπει να συνδέεται



τουλάχιστον με ένα άλλο στιγμιότυπο, μέσω αυτής της ιδιότητας. Με άλλα λόγια, η ιδιότητα θα πρέπει να έχει τουλάχιστον μία τιμή για κάθε στιγμιότυπο της κλάσης. Στην OWL Lite οι τιμές ελάχιστης πληθικότητας που επιτρέπονται είναι 0 και 1. Τιμή 0 σημαίνει ότι η ιδιότητα είναι προαιρετική για τη συγκεκριμένη κλάση.

- **maxCardinality**: Ο περιορισμός μέγιστης πληθικότητας (**maxCardinality**) τίθεται σε μία ιδιότητα, αναφερόμενος σε μία κλάση. Αν για μία ιδιότητα έχουμε **maxCardinality** 1, τότε κάθε στιγμιότυπο της κλάσης θα πρέπει να συνδέεται το πολύ με ένα άλλο στιγμιότυπο, μέσω αυτής της ιδιότητας. Αν έχουμε **maxCardinality** 0, τότε υποδηλώνουμε ότι κάποια κλάση δεν έχει τιμές για τη συγκεκριμένη ιδιότητα.
- **cardinality**: Ο περιορισμός πληθικότητας (**maxCardinality**) τίθεται σε μία ιδιότητα, αναφερόμενος σε μία κλάση και καθορίζει την ακριβή πληθικότητα της ιδιότητας για αυτήν την κλάση

Επίσης, στοιχείο της OWL Lite είναι η τομή (**intersectionOf**) της οποίας η χρήση περιορίζεται σε επώνυμες κλάσεις και περιορισμούς. Ακολουθούν εκείνα τα στοιχεία που επεκτείνουν την OWL Lite σε OWL DL και OWL Full.

- **oneOf**: Οι κλάσεις μπορούν να περιγραφούν ως απαρίθμηση των στιγμιότυπων από τα οποία αποτελούνται. Τα μέλη της κλάσης είναι ακριβώς τα απαριθμημένα στιγμιότυπα και μόνο αυτά.
- **hasValue**: Μία ιδιότητα ενδέχεται να απαιτείται να έχει ως τιμή ένα συγκεκριμένο στιγμιότυπο.
- **disjointWith**: Κάποιες κλάσεις μπορούν να δηλωθούν ότι είναι ξένες μεταξύ τους (**disjointWith**), δηλαδή ότι δεν έχουν ούτε ένα κοινό στοιχείο. Έτσι, αν κάποιο στιγμιότυπο ανήκει σε μία κλάση, αποκλείεται να ανήκει και σε μία ξένη προς αυτήν κλάση.
- **unionOf**, **complementOf**, **intersectionOf** : Σε OWL DL και OWL Full η ένωση, το συμπλήρωμα και η τομή αντίστοιχα επιτρέπουν διάφορους συνδυασμούς κλάσεων και περιορισμών.
- **minCardinality**, **maxCardinality**, **cardinality** : Οι περιορισμοί πληθικότητας επιτρέπεται να παίρνουν τιμές, όχι μόνο 0 και 1, αλλά οποιονδήποτε φυσικό αριθμό.
- **complex classes** : Σε πολλές δομές, η OWL Lite περιορίζει τη σύνταξη σε μοναδικά ονόματα κλάσεων. Η OWL Full επεκτείνει αυτόν τον περιορισμό στο να επιτρέπει



σύνθετες περιγραφές κλάσεων, όπως απαριθμημένες κλάσεις, περιορισμούς ιδιοτήτων και συνδυασμούς λογικών εκφράσεων. Επίσης, η OWL Full επιτρέπει σε κλάσεις να χρησιμοποιούνται ως στιγμιότυπα.

Το σπουδαίο είναι ότι η OWL είναι και η ίδια οντολογία. Αυτό έχει ως αποτέλεσμα να υπάρχουν πρότυπα οντολογιών έτοιμα στο διαδίκτυο με πεδία ονοματολογίας (namespaces) έτοιμα προς χρήση. Στην περίπτωση ενσωμάτωσης του αντίστοιχου plugin στο protégée, κάθε φορά που δημιουργούμε καινούρια οντολογία, αυτή βασίζεται σε κάποιο πρότυπο το οποίο είναι «κρεμασμένο» σε κάποια URL, και το οποίο μπορούμε να χρησιμοποιήσουμε ως πεδίο ονοματολογίας (namespace) διαδικτυακά [36]. Αυτό καθιστά την εφαρμογή μας παντοδύναμη, αφού μπορεί μια συσκευή να λειτουργεί ανεξάρτητα ως προς το σημείο που βρίσκεται αρκεί να έχει πρόσβαση στο διαδίκτυο. Οι δικτυακές αυτές αναφορές ονοματολογίας ονομάζονται URI's (Uniform Resource Identifier). Η URI είναι μια συνεκτική μορφή η οποία προσδιορίζει μια πηγή πληροφορίας στο διαδίκτυο. Ο κύριος λόγος χρήσης μιας τέτοιας μορφής είναι το γεγονός ότι μπορούμε να έχουμε πρόσβαση στα δεδομένα, μέσω δικτύου (και κατ' επέκταση διαδικτύου). Και μάλιστα η πρόσβαση αυτή είναι εφικτή και δυνατή μέσω του πρωτοκόλλου HTTP. Το W3C δεν αποδέχεται πλέον τον όρο URL για την αναφορά σε γνωσιακό περιεχόμενο και έχει διαλέξει σαν επίσημο τον όρο URI. Για να καταλάβουμε τη διαφορά ανάμεσα σε URLs και URIs θα δώσουμε ένα παράδειγμα. Τεχνικά όπως φαίνεται και στο παράδειγμα ένα URL είναι πάντα URI, ενώ ένα URI δεν είναι πάντα URL.

Κατά τα γνωστά μια URL είναι μια διεύθυνση στο διαδίκτυο. Για παράδειγμα :

URL:<http://vsdesign.homelinux.org/~bstefan/GEOnt.html>

Η URI είναι μια αναφορά σε μια οντότητα του διαδικτύου η οποία ανήκει σε κάποια οντολογία. Για παράδειγμα :

URI: <http://vsdesign.homelinux.org/~bstefan/GEOnt.owl#kathigitis>

Η παραπάνω αναφορά αναφέρεται στην κλάση *kathigitis* της οντολογίας *GEOnt.owl* και μπορεί να ζητήσει στιγμιότυπα τα οποία ανήκουν στην κλάση *kathigitis*





Το σχήμα των URIs όπως θα δούμε στη συνέχεια είναι γνωστικό στοιχείο για συμπερασμό.



## 8. Ανάπτυξη και Χρήση Οντολογιών

Στην παρούσα εργασία αναπτύσσουμε ένα μοντέλο και τελικά υλοποιούμε ένα σύστημα που αποτελείται από πραγματικά αντικείμενα μαθησιακής διαδικασίας και επικεντρωνόμαστε στη διαδικασία αξιολόγησης ενός μαθητή δευτεροβάθμιας εκπαίδευσης. Αξιολογούμε τα δεδομένα από τις περιβαλλοντικές συνθήκες (διαδικασία επίγνωσης πλαισίου), τη μαθησιακή διαδικασία και παρεμβάλουμε κανόνες. Όλα αυτά περνούν μέσα από μια μηχανή αξιολόγησης. Το σύστημα της μηχανής και της τελικής εξαγωγής συμπερασμάτων υλοποιείται σε Java. Το σύστημα είναι πρωτότυπο αφού τόσο στην Ελληνική όσο και στη Διεθνή βιβλιογραφία δε βρήκαμε κάποιο παρόμοιο σύστημα που να δουλεύει με αυτόν τον τρόπο και για αυτόν το σκοπό.

Το σύστημά μας αποτελείται από τις οντολογίες, το σύστημα συλλογής στοιχείων από το περιβάλλον, το σύστημα κανόνων, το σύστημα συμπερασμού και το σύστημα διεπαφής χρήστη.

Πιο συγκεκριμένα, έχουμε δημιουργήσει μια απλή εκπαιδευτική οντολογία στην οποία δώσαμε το όνομα GEOnt, και η οποία περιλαμβάνει τα πιο βασικά στοιχεία της εκπαιδευτικής διαδικασίας. Έχουμε χρησιμοποιήσει την οντολογία INO που δημιουργήθηκε στα πλαίσια του έργου που περιγράφεται στο [30] ως οντολογία περιβάλλοντος η οποία περιγράφει θαυμάσια εσωτερικούς χώρους. Δημιουργήσαμε σε περιβάλλον Java κανόνες για συμπερασμό, χρησιμοποιήσαμε τη μηχανή Bossam [39] [40], για έλεγχο σταθερότητας και συμπερασμό και τέλος όλα αυτά τα λειτουργήσαμε κάτω από ένα γραφικό περιβάλλον διεπαφής χρήστη το οποίο δημιουργήθηκε και αυτό σε Java.

### 8.1. Εκπαιδευτική Οντολογία GEOnt

Οι έννοιες που πρέπει να περιληφθούν [28] σε μια εκπαιδευτική οντολογία, πρέπει να καλύπτουν την ποικιλία και την ιδιομορφία που παρουσιάζει ο τομέας και εμπίπτουν στην κατηγορία οντολογιών που είναι οντολογίες πεδίου (domain ontologies). Αυτές σύμφωνα με τον ορισμό του [9], είναι συχνά εξαρτημένες από το πλαίσιο στο οποίο γίνονται (context-driven). Δηλαδή οι έννοιες-κατηγορίες που υπάρχουν σε αυτές, ακολουθούν την αντίληψη που μια δεδομένη ομάδα χρηστών ή δημιουργών έχουν για το γνωστικό ή επιστημονικό αντικείμενο κατά το οποίο είναι χτισμένη η οντολογία. Έτσι, για παράδειγμα μια οντολογία



για το μάθημα της Φυσικής στο Λύκειο, δε θα περιλαμβάνει έννοιες όπως αστροφυσική ή μικρο-φυσική, οι οποίες έτσι και αλλιώς είναι οντότητες της φυσικής γενικότερα.

Σκοπός μας δεν είναι να καλύψουμε πολύπλευρα μια εκπαιδευτική οντολογία αλλά να δημιουργήσουμε μια υποδομή η οποία μπορεί εύκολα να εμπλουτισθεί. Υλοποιούμε το σύστημα το οποίο μπορεί να κάνει συμπερασμό και να δίνει αποτελέσματα τα οποία θα εξαρτώνται από την εκπαιδευτική διαδικασία, το πλαίσιο λειτουργίας (περιβάλλον κίνησης του ενδιαφερόμενου). Επιπλέον υλοποιούμε ένα σύστημα το οποίο μπορεί να λειτουργεί με διεπαφή αναπτυγμένη σε ανοικτό λογισμικό όπως για παράδειγμα είναι το περιβάλλον της Java.

Για να δημιουργηθεί μια καλή εκπαιδευτική οντολογία είναι προφανής η ανάγκη περιγραφής των βασικών εννοιών της. Για τις ανάγκες της εργασίας μας, και επειδή κατά τη γνώμη μας είναι αρκετό, περιορισθήκαμε ακριβώς στις πού βασικές εκπαιδευτικές έννοιες. Όπως θα δούμε στη συνέχεια δημιουργήσαμε ως βασικές κλάσεις της οντολογίας μας θεμελιώδεις έννοιες της εκπαιδευτικής διαδικασίας..

Τα θεμελιώδη μεγέθη που ορίζουμε στην εκπαιδευτική οντολογία GEOnt (General Educational Ontology) είναι τα εξής :

Μάθημα

Μαθητής

Καθηγητής

Εξέταση

Τάξη

Στο επόμενο επίπεδο της ταξινόμιας διαχωρίζουμε ως ακολούθως :

**Μάθημα**

**Μαθητής**

**Εξέταση**

Εσωτερική\_Εξέταση

Εσωτερική\_Δύσκολη

question

result

Εσωτερική\_Εύκολη

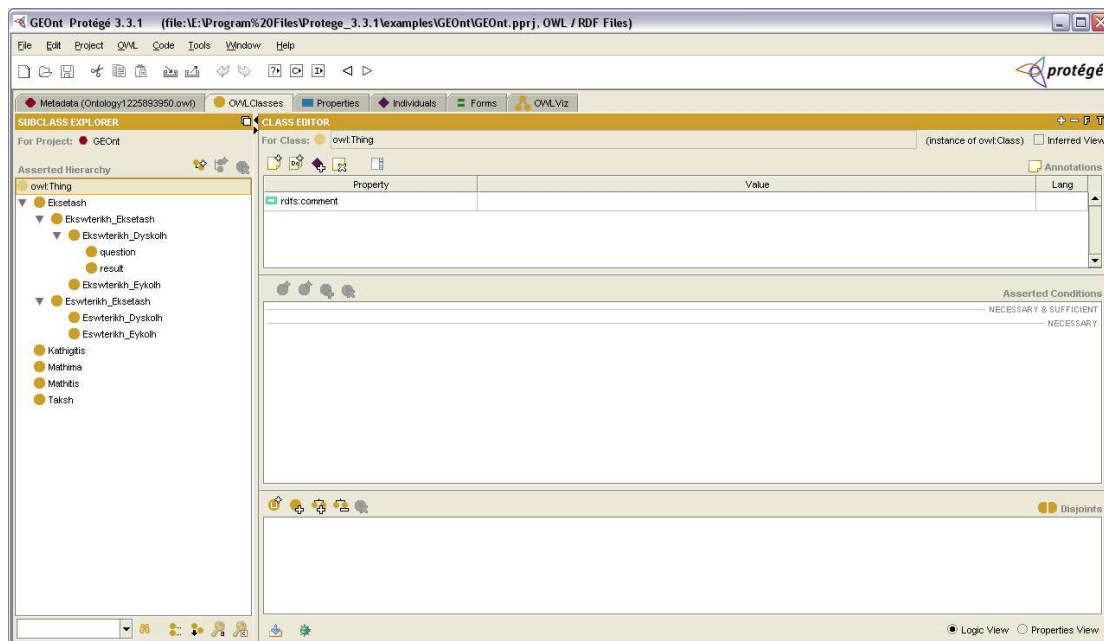


Εξωτερική\_Εξέταση  
Εξωτερική\_Δύσκολη  
Εξωτερική\_Εύκολη

## Καθηγητής

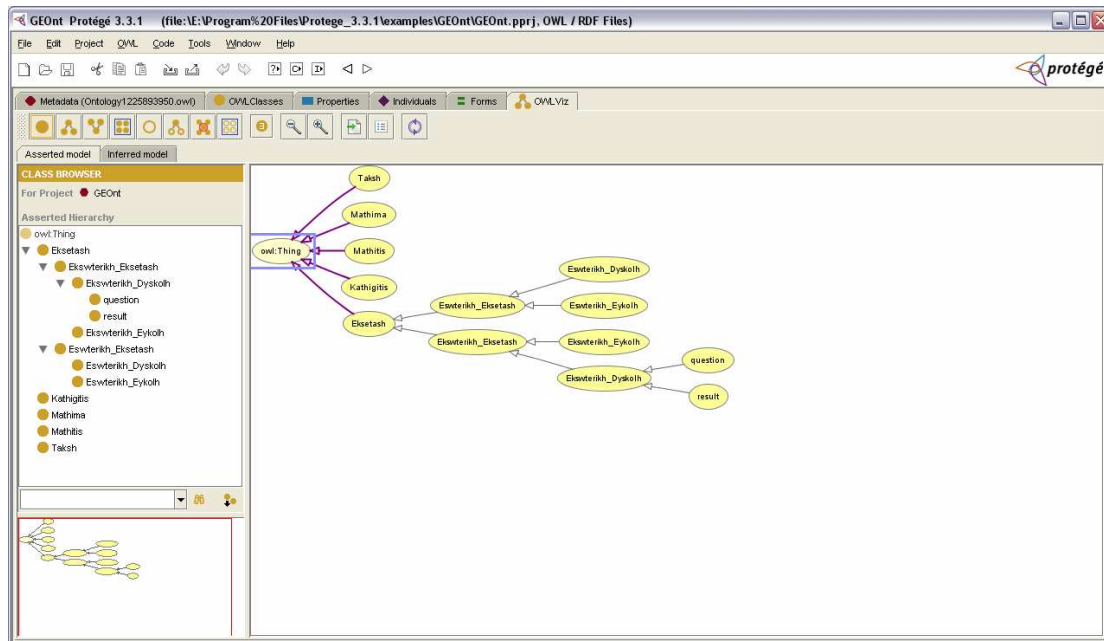
## Τάξη

Στις (Εικόνα 1) και (Εικόνα 2) είδαμε δύο εικόνες από το protégée [31]. Στην (Εικόνα 3) βλέπουμε τη βασική μας οντολογία GEOnt και την οπτικοποίησή της από το OWL Plugin OWL\_Viz (Εικόνα 4) & (Εικόνα 5). Επίσης στην (Εικόνα 3) βλέπουμε και την ταξινόμιά της GEOnt.

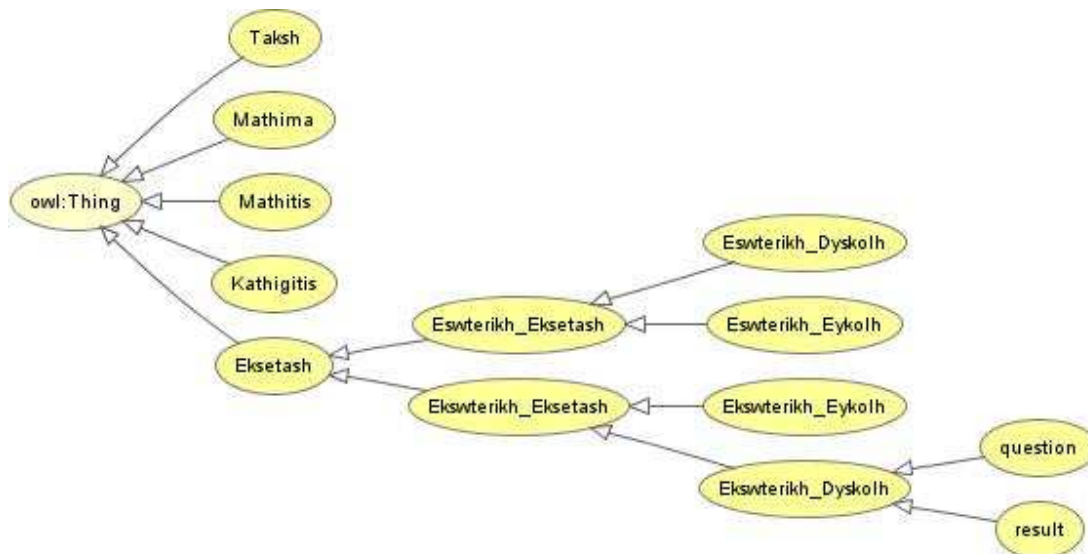


Εικόνα 3: Οι κλάσεις και οι υποκλάσεις της GEOnt

Χρησιμοποιήσαμε τις υποκλάσεις Εσωτερική\_Εξέταση, Εσωτερική\_Δύσκολη και Εξωτερική\_Δύσκολη, Εξωτερική\_Εύκολη και στα τρία εκπαιδευτικά σενάρια που περιγράφονται στο κεφάλαιο 18. Επίσης χρησιμοποιήσαμε τις υποκλάσεις result και question για τις ανάγκες του 2ου εκπαιδευτικού σεναρίου (κεφάλαιο 18.2), όπου έχουμε διάδραση του χρήστη με το σύστημα τη στιγμή του σεναρίου με απαντήσεις των ερωτήσεων.



Εικόνα 4: η οπτικοποίηση της GEOnt (Η ιεραρχία)



Εικόνα 5: η οπτικοποίηση της GEOnt σε πιο μεγάλη κλίμακα (Ιεραρχία)

Όπως έχουμε περιγράψει θέλουμε να αναπτύξουμε και να υλοποιήσουμε ένα σύστημα ηλεκτρονικής μάθησης το οποίο διαδραστικά με τον μαθητή και λαμβάνοντας υπόψη περιβαλλοντικές συνθήκες θα μπορεί να προσαρμόζει τη διαδικασία της μάθησης.

Από τις θεμελιώδεις οντότητες της εκπαιδευτικής οντολογίας, αποφασίσαμε να δώσουμε βάρος και βάθος σε μια κλάση, την οποία και αναλύσαμε πιο πολύ. Η κλάση αυτή είναι η εξέταση. Ο λόγος είναι ότι η εφαρμογή μας θα έχει ως σκοπό να μπορεί να βγάζει



συμπεράσματα κατά τη διαδικασία της αξιολόγησης-εξέτασης ενός μαθητή. Για παράδειγμα θα μπορεί να προσαρμόζει τη διαδικασία της εξέτασης παίρνοντας υπόψη τις συνθήκες περιβάλλοντος. Έτσι λοιπόν μας ενδιαφέρει το είδος της εξέτασης και πώς αυτό μπορεί να προσαρμοσθεί ανάλογα με τις συνθήκες στις οποίες δουλεύει ο μαθητής.

### 8.1.1. Η δομή και οι συσχετίσεις της GEOnt

οι συσχετίσεις – ιδιότητες (properties) που έχουμε δημιουργήσει προς το παρόν είναι: (Εικόνα 6), (Εικόνα 7). Οι συσχετίσεις που δημιουργήσαμε συσχετίζουν τις κλάσεις της οντολογίας. Αναλυτικά αυτές είναι :

- **<Eksetazei>**, η οποία διασυνδέει τον Καθηγητή με το είδος της Εξέτασης : (Εσωτερική\_Δύσκολη, Εσωτερική\_Εύκολη, Εξωτερική\_Δύσκολη, Εξωτερική\_Εύκολη)
- **<has>**, η οποία διασυνδέει την υποκλάση question με την υποκλάση result
- **<Exei\_Mathiti>**, η οποία διασυνδέει τον Καθηγητή με τον Μαθητή και έχει αντίστροφη τη σχέση Είναι\_Μαθητής.
- **<Einai\_Mathitis>**, η οποία διασυνδέει τον Μαθητή με τον Καθηγητή και έχει αντίστροφη τη σχέση Έχει\_Μαθητή. Οι δύο τελευταίες είναι και αντίστροφες.
- **<einai>**, και **<mporei\_na\_einai>** οι οποίες είναι αντίθετες. Αυτές οι σχέσεις δε διασυνδέουν στατικά κάποιες κλάσεις. Τις δημιουργήσαμε για να διασυνδέσουμε δυναμικά μέσα από την εφαρμογή τις κλάσεις μάθημα
- **<didaskei>**, η οποία διασυνδέει τον Καθηγητή με τον Μαθητή, και έχει αντίστροφη της σχέση **<didasketai\_apo>**.
- **<didasketai\_apo>**, η οποία διασυνδέει τον Μαθητή με τον Καθηγητή και έχει αντίστροφη τη σχέση **<didaskei>**. Οι δύο τελευταίες είναι και αντίστροφες.
- **<epilegei>**, η οποία διασυνδέει τον Μαθητή με το Μάθημα, και είναι αντίστροφη της σχέσης **<exei\_epilexthei>**.
- **<exei\_epilexthei>**, η οποία διασυνδέει το Μάθημα με το Μαθητή και έχει αντίστροφη τη σχέση **<epilegei>**. Οι δύο τελευταίες είναι και αντίστροφες.
- **<eksetazetai>**, η οποία χρησιμοποιήθηκε δυναμικά μέσα από την εφαρμογή.

Στα παραπάνω, αναφέρθηκε αρκετές φορές ο όρος **αντίστροφη (inverse)** συσχέτιση. Αντίστροφες είναι δύο συσχετίσεις αν διασυνδέονται με τις εξής σχέσεις :

Αν ισχύουν



A <σχέση\_S> B, και

B <αντίστροφη\_σχέσης\_S> A.

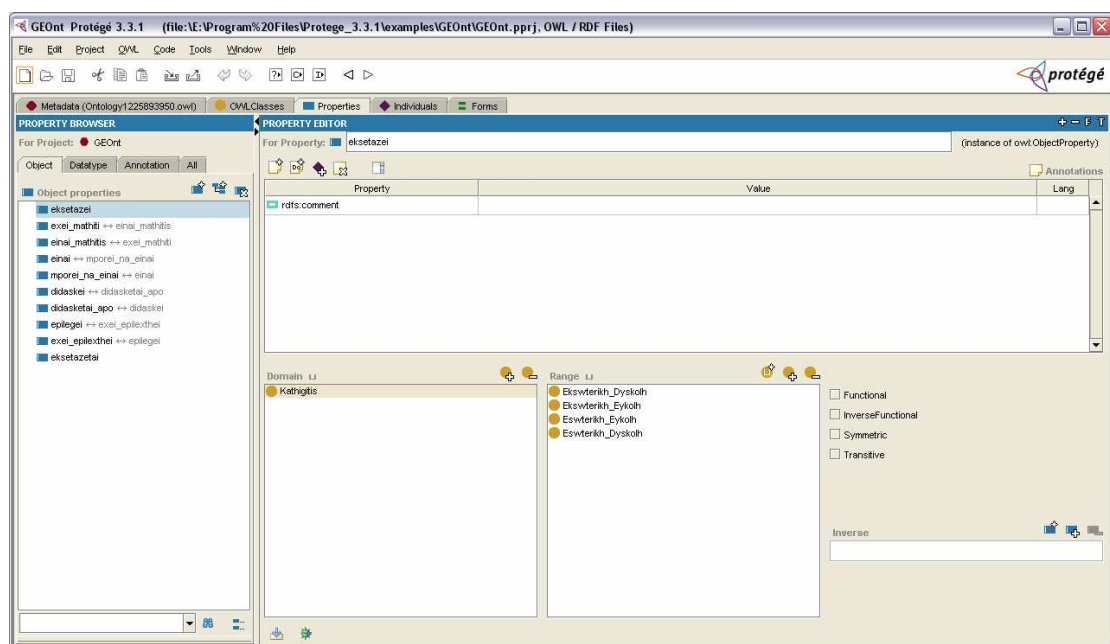
Όπου A και B κλάσεις της οντολογίας, τότε οι σχέσεις <σχέση\_S> και <αντίστροφη\_σχέσης\_S> είναι αντίστροφες συσχετίσεις. Για παράδειγμα :

Αν

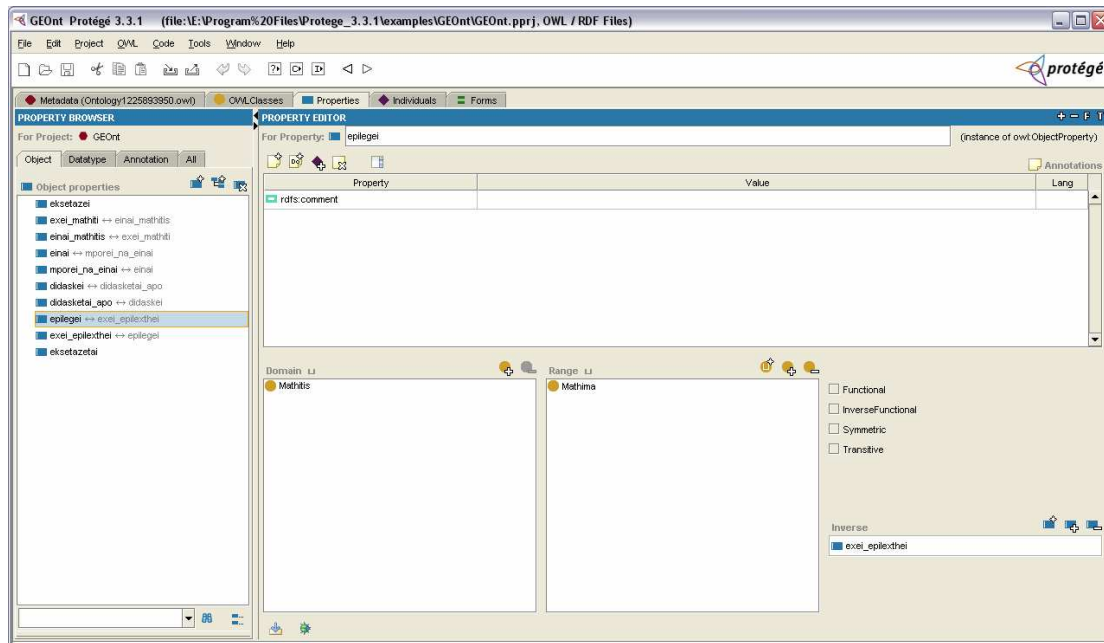
ο Petroy <Είναι\_Μαθητής> του Georgiou και

οι συσχετίσεις <Είναι\_Μαθητής> και <Έχει\_Μαθητή> είναι αντίστροφες, τότε

ο Georgiou <Έχει\_Μαθητή> τον Petrou



Εικόνα 6: Οι συσχετίσεις – ιδιότητες που δημιουργήσαμε



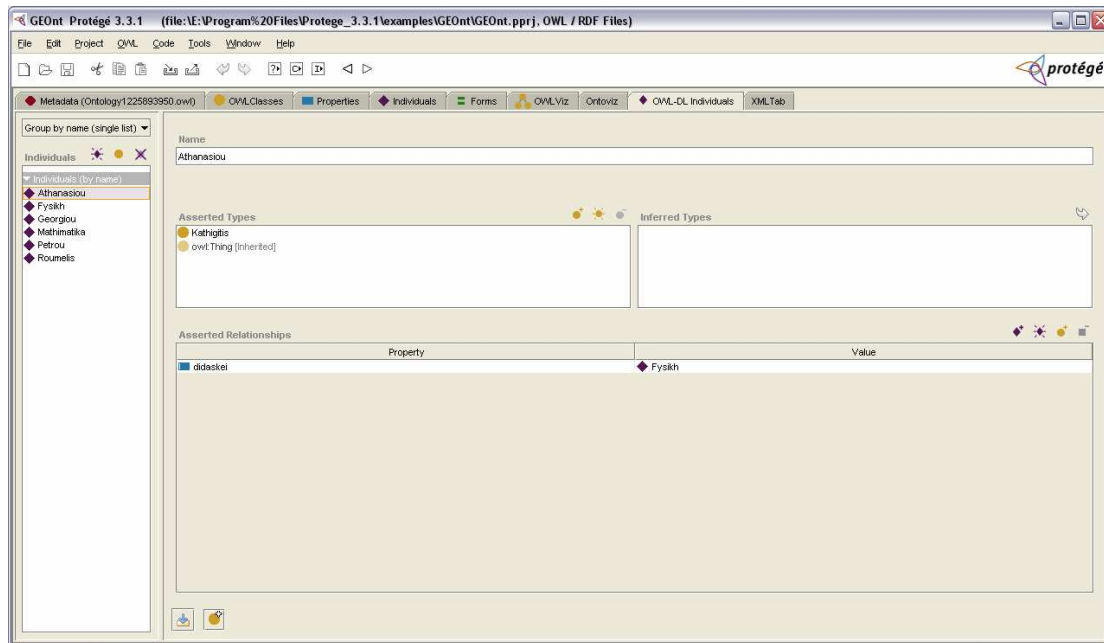
Εικόνα 7 : Οι συσχετίσεις – ιδιότητες που δημιουργήσαμε

### 8.1.2. Τα Στιγμιότυπα της GEOnt

Προκειμένου να χρησιμοποιήσουμε την οντολογία στην πράξη, δημιουργήσαμε στιγμιότυπα (individuals-Instances) τα οποία αντιστοιχούν στις κλάσεις της οντολογίας και διασυνδέονται μεταξύ τους με τις συσχετίσεις που ορίσαμε στην προηγούμενη παράγραφο. Δημιουργήσαμε μαθητές, καθηγητές και μαθήματα, τόσο στατικά μέσα στην οντολογία, όσο και δυναμικά με αλήθειες μέσα από την εφαρμογή.

Μια εικόνα από την καρτέλα όπου δημιουργούνται τα στιγμιότυπα βλέπουμε στην (Εικόνα 8):





Εικόνα 8: Τα στιγμιότυπα (instances-individuals) της οντολογίας GEOnt

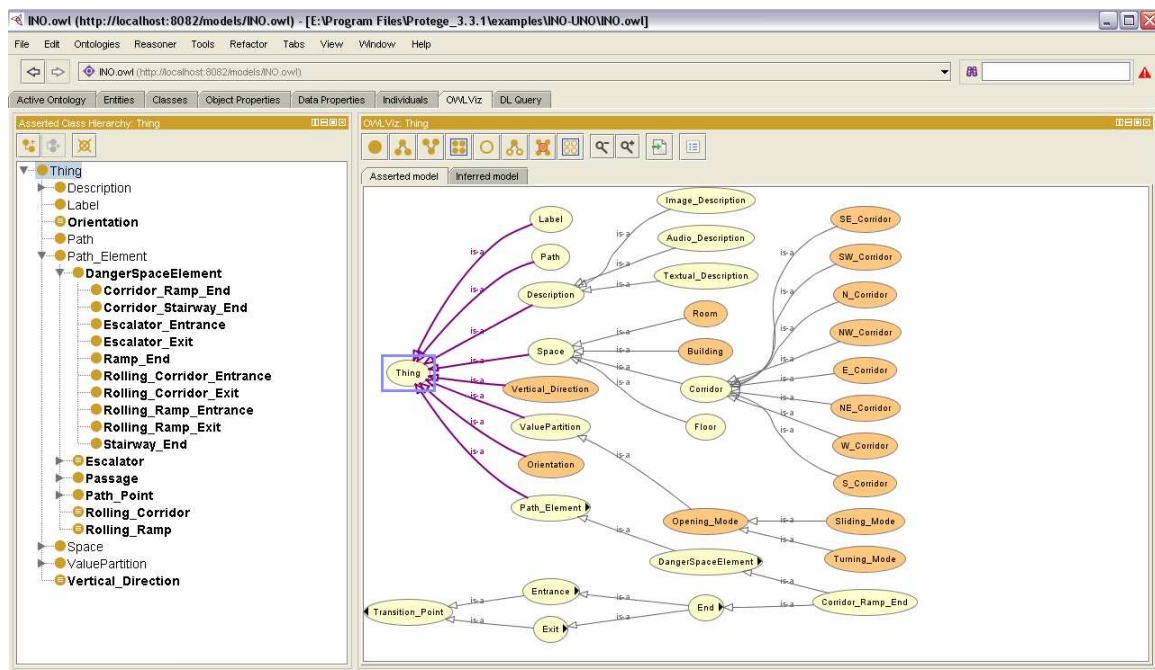
Όπως θα δούμε και στη συνέχεια, η δημιουργία και ο ορισμός στιγμιότυπων δε δεσμεύει ούτε το πρόγραμμα, ούτε το σύστημα που δημιουργούμε, αφού μπορούμε να ορίσουμε τόσο νέα στιγμιότυπα, όσο και νέες συσχετίσεις-συμπεριφορές (behaviors) από τη μηχανή κρίσεως και συμπερασμού που αναλύουμε, περιγράφουμε και χρησιμοποιούμε στη συνέχεια. Αυτό που μπορεί να διακρίνει κανείς ως λεπτομέρεια στην οντολογία μας είναι ότι βάζουμε δύο-τριών ειδών στιγμιότυπα για κάθε κλάση δυναμικά και στατικά, και στα ερωτήματα που θέτουμε παίρνουμε απαντήσεις που σχετίζονται και με τα δύο είδη πληροφορίας.

## 8.2. Οντολογία Πλαισίου INO (περιβάλλοντος)

Όπως ήδη έχουμε αναφέρει, στα πλαίσια του έργου που περιγράφεται στο [30] αναπτύχθηκε μια οντολογία που περιγράφει εσωτερικούς χώρους. Στην προαναφερθείσα αναφορά μπορεί να βρει κανείς μια πλήρη ανάλυση και περιγραφή τόσο της οντολογίας όσο και της δημιουργίας της. Στη εργασία μας θα χρησιμοποιήσουμε την ίδια οντολογία θεωρώντας ότι το πλαίσιο στο οποίο θα διαδραματισθεί η εκπαιδευτική διαδικασία είναι το ίδιο με αυτό του έργου. Η οντολογία ονομάζεται INO και είναι μια πολύπλοκη και πολύ εκφραστική οντολογία. Έχει δημιουργηθεί με GIS από στοιχεία κατόψεων κτιρίων οι οποίες έχουν ψηφιοποιηθεί και έχουν εισαχθεί σε κάποιο σύστημα GIS.



Παράλληλα κατά τη διάρκεια της ψηφιοποίησης των στοιχείων για τη δημιουργία της οντολογίας έγινε σημασιολογικός εμπλουτισμός, των κατόψεων. Για παράδειγμα μια κάτοψη ενός όροφου δημιουργήθηκε σε πολλαπλά επίπεδα. Ένα επίπεδο δημιουργήθηκε με στοιχεία τις εξόδους των δωματίων. Ένα ακόμη επίπεδο δημιουργήθηκε με τις εξόδους των ανελκυστήρων. Ένα άλλο επίπεδο δημιουργήθηκε με τις εξόδους των κλιμακοστασίων κλπ. Κάθε επίπεδο αποθηκεύτηκε διανυσματικά (πχ ESRI shapefiles). Η οντολογία περιέχει ένα πλήθος κλάσεων, υποκλάσεων, ιδιοτήτων, συναρτήσεων κλπ. Μια άποψη για την ιεραρχία-ταξινόμια μπορούμε να πάρουμε από την εικόνα (Εικόνα 9)



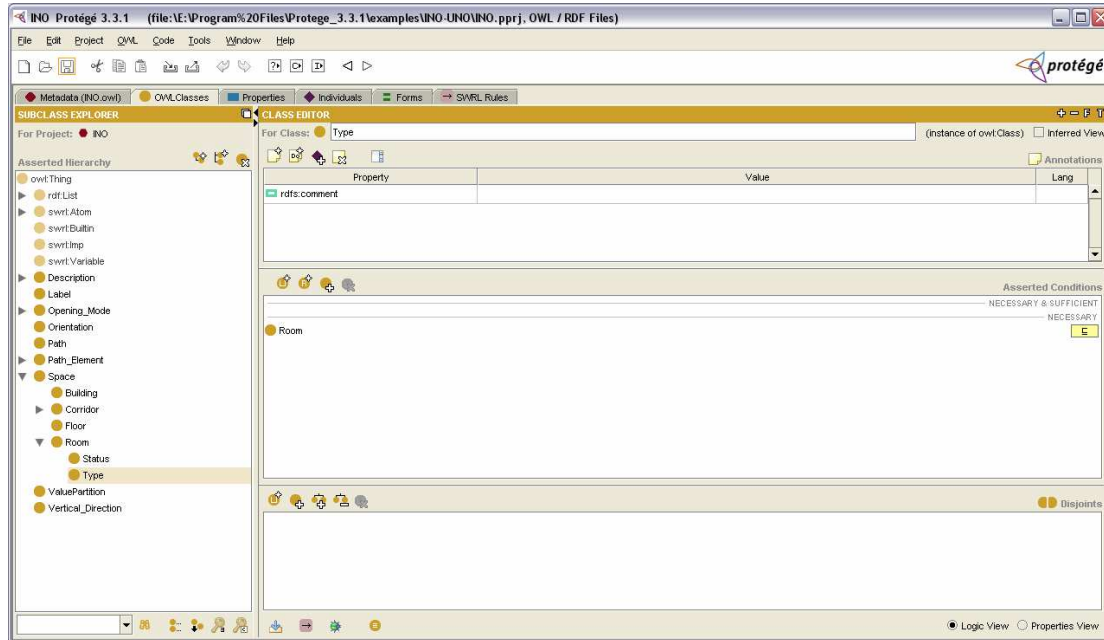
Εικόνα 9: Η Ιεραρχία της οντολογίας INO

Για τη δική μας εργασία, τα περισσότερα από τα στοιχεία της INO θα μείνουν αναξιοποίητα. Τα στοιχεία που θα χρειαστούμε θα είναι αυτά που έχουν να κάνουν με τα δωμάτια και την κίνηση από και προς ένα από αυτά. Φυσικά όπως εξηγούμε και στο τέλος αυτής της εργασίας, οι προεκτάσεις που μπορεί να λάβει η εφαρμογή είναι πάρα πολλές και κάθε φορά μπορούμε να τροποποιούμε και την GEOnt αλλά και την INO έτσι ώστε να καλυφθούν και οι επεκτάσεις αυτές.

Για τις ανάγκες των εκπαιδευτικών μας σεναρίων (περιγράφονται στο 18), εμπλουτίσαμε την INO με δύο νέες υποκλάσεις της ήδη υπάρχουσας κλάσης Room. Τις νέες υποκλάσεις της Room τις ονομάσαμε Type και Status (Εικόνα 10). Η Type δημιουργήθηκε για να καλύψει το

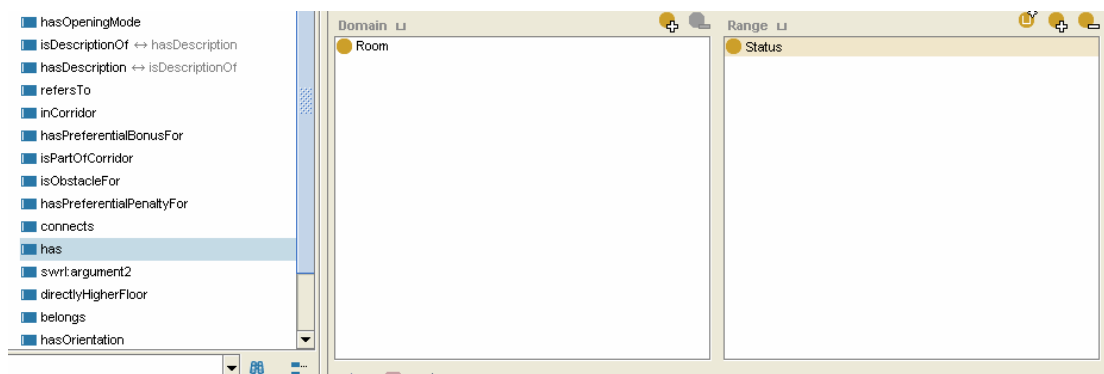


τρίτο εκπαιδευτικό σενάριο (κεφάλαιο 18.3), ενώ η Status δημιουργήθηκε για να καλύψει το πρώτο και το δεύτερο εκπαιδευτικό σενάριο (κεφάλαια 18.1 και 18.2 αντίστοιχα) .



Εικόνα 10 : Εμπλουτισμός της INO με δύο νέες υποκλάσεις

Επιπλέον δημιουργήσαμε και δύο νέες συσχετίσεις, τις has και is\_it (Εικόνα 11 και Εικόνα 12), οι οποίες συνδέουν η μεν πρώτη την Room και την Status, ενώ η δεύτερη την Room και την Type.



Εικόνα 11 : Η νέα συσχέτιση has με τις κλάσεις που συσχετίζει



Εικόνα 12 : Η νέα συσχέτιση is\_it με τις κλάσεις που συσχετίζει

Μέσω των συσχετίσεων και των υποκλάσεων που μόλις περιγράψαμε προσθέτουμε νέα γνώση στην εφαρμογή μας με την υπορουτίνα **facts** την οποία μπορούμε να δούμε στον κώδικα του προγράμματος στο ΠΑΡΑΡΤΗΜΑ Β.

### 8.3. Σχεδιασμός Συστήματος

Το σύστημα που δημιουργήσαμε περιλαμβάνει :

- Σύστημα αξιολόγησης των δεδομένων των αισθητήρων. Το σύστημα αυτό περιλαμβάνει σύνδεση του υποτιθέμενου συστήματος συλλογής των δεδομένων με το σύστημα χειρισμού των οντολογιών. Τα στοιχεία που συλλέγονται διασυνδέονται με στοιχεία από τις οντολογίες. Τα αποτελέσματα προωθούνται στο επόμενο επίπεδο, το οποίο είναι το επίπεδο συμπερασμού.
- Μηχανή κανόνων : Οι κανόνες θα πρέπει να είναι εκφρασμένοι σε κάποια μορφή αναπαράστασης γνώσης σε κάποια γλώσσα αναπαράστασης γνώσης. Τελικά χρησιμοποιήσαμε τη Bossam [39] η οποία έχει τη δυνατότητα με μια γλώσσα που ενσωματώνει (την Buchingae) να μπορεί να ορίζει κανόνες και αλήθειες. Αναπτύσσουμε πιο κάτω αναλυτικά την τεχνολογία και τις μεθόδους. Οι κανόνες θα προωθούνται στη μηχανή συμπερασμού.
- Μηχανή συμπερασμού. Στο σημείο αυτό αναπτύξαμε αλγόριθμο που παράγει συμπέρασμα ανάλογα με τους κανόνες και τα στοιχεία γνώσης περιβάλλοντος και εκπαιδευτικής διαδικασίας. Ο μηχανή συμπερασμού που αναπτύξαμε διασυνδέει όλη την τεχνολογία που περιγράφουμε σε αυτή την εργασία, καθώς και προγραμματιστικές δομές, *switch (c) { case }*.

Πρέπει να διευκρινίσουμε ότι το σύστημά μας δε θα περιλαμβάνει σύστημα συλλογής των δεδομένων από αισθητήρες. Όμως, το σύστημα συλλογής με μηχανικούς ή ηλεκτρονικούς



αισθητήρες είναι ένα σχετικά τετριμμένο θέμα, και η τεχνολογία του μπορεί να ενσωματωθεί εύκολα στο σύστημά μας. Εξάλλου τα στοιχεία που υποτίθεται ότι συλλέγουν οι αισθητήρες, στην ουσία τα έχουμε, είναι πραγματικά και εισέρχονται στο σύστημά μας υπό μορφήν αρχείων ASCII.

## 9. Τεχνολογία Κρίσης (Reasoners-Classifiers )

Η κρίση (reasoning) ή classification είναι μια διαδικασία απαραίτητη για την τεχνολογία των οντοτήτων, γι' αυτό και από την έκδοση 4 το protégé την ενσωματώνει στο περιβάλλον της. Οι διάφορες τεχνολογίες κρίσης έχουν και διαφορετικά μειονεκτήματα και πλεονεκτήματα. Το κρίσιμο σημείο όπου διαφοροποιούνται οι reasoners είναι η ποσότητα των στιγμιότυπων που έχουν να διαχειριστούν. Άλλοι είναι πιο γρήγοροι στη διαχείριση, ενώ άλλοι είναι πιο αργοί με καλύτερη διαχείριση μνήμης.

Στη δική μας βέβαια περίπτωση χρησιμοποιήσαμε μια εξωτερική μηχανή κρίσεως (όχι ενσωματωμένη στο protégé) η οποία μας έδωσε τη δυνατότητα δημιουργίας, ορισμού και σε πολλές περιπτώσεις επαναπρογραμματισμού των αληθειών και των κανόνων όπως επίσης και δυνατότητα προσθήκης νέας γνώσης δυναμικά με συναρτήσεις και βιβλιοθήκες γραμμένες σε Java. Η μηχανή αυτή είναι η Bossam, την οποία περιγράφουμε αναλυτικά πιο κάτω.

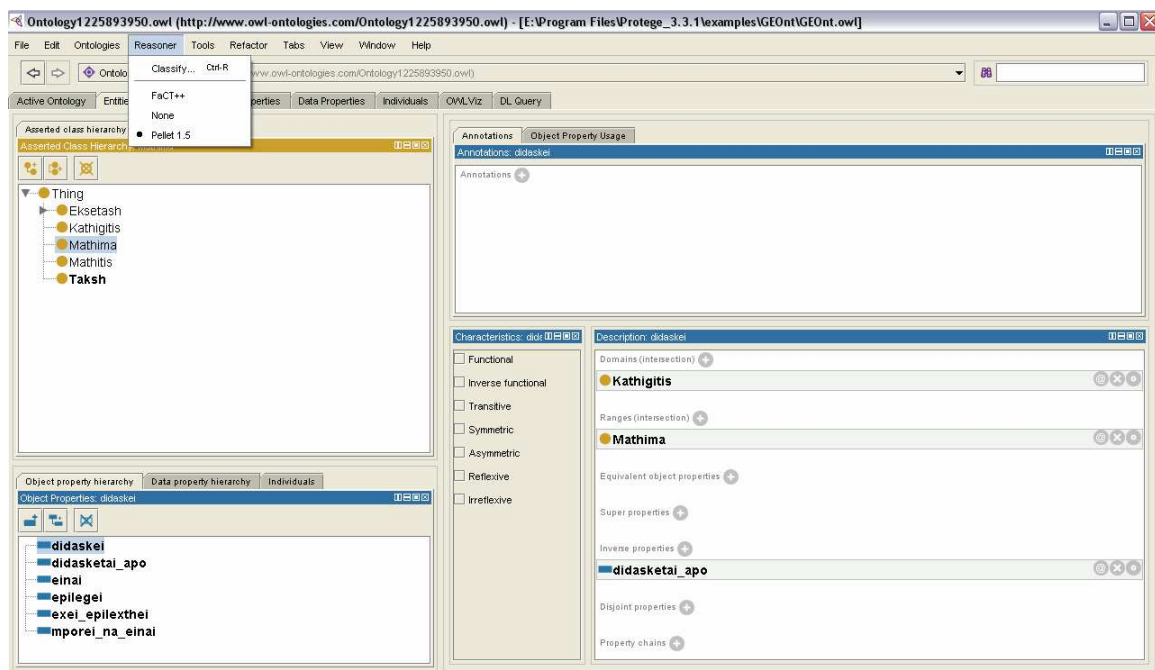
Η διαδικασία Reasoning είναι ένα από τα χαρακτηριστικά κλειδιά του περιβάλλοντος Protégé-OWL. Μέσα από τη διαδικασία αυτή ελέγχεται η οντολογία για τη σταθερότητα-συνέπεια-συνοχή της (consistency). Δηλαδή ελέγχεται για το κατά πόσο οι κλάσεις και οι υποκλάσεις έχουν ορισθεί σωστά και δεν έχουν αλληλεξαρτήσεις. Επίσης ελέγχεται το γεγονός ότι μπορεί από την οντολογία να παραχθεί συμπερασμός. Όπως διαβάζουμε στο [10], μια από τις δουλειές ενός reasoner είναι να κάνει έναν έλεγχο αν μια κλάση είναι υποκλάση μιας άλλης κλάσης. Έτσι μπορεί στη συνέχεια να συμπεράνει αν υπάρχει ιεραρχία-ταξινόμια και ποια είναι αυτή. Ως αποτέλεσμα συμπεραίνει η μηχανή reasoning αν είναι δυνατόν να προστεθούν στιγμιότυπα (instances-individuals). Επίσης όπως διαβάζουμε στα [10], [38], είναι απαραίτητο το reasoning στις οντολογίες, επειδή η υποβολή ερωτημάτων σε μια οντολογία δημιουργείται από εκτελέσιμους αλγόριθμους, και θα πρέπει να είναι σίγουρο ότι κάθε αλγόριθμος θα ολοκληρωθεί μετά από ένα ορισμένο αριθμό βημάτων. Με πιο απλά λόγια το reasoning εξασφαλίζει το κριτήριο της περατότητας και αυτό σημαίνει σαφήνεια στον ορισμό μιας οντολογίας.

Η έκδοση του protégé που ενσωματώνει reasoners είναι η έκδοση 4.0. Οι μηχανές κρίσης (reasoners) που περιέχει είναι οι FaCT++ [34] [22], και Pellet 1.5 [37]. Εδώ να σημειώσουμε



ότι η Pellet δε μας δούλεψε σωστά όταν οι ονομασίες των κλάσεων και υποκλάσεων καθώς και των σχέσεων ήταν γραμμένες στα Ελληνικά. Δηλαδή όταν υποβάλαμε ερωτήματα DL στην ονοματολογία που είχαμε κάνει reasoning με την Pellet δεν παίρναμε αποτέλεσμα, ενώ αν είχαμε κάνει το reasoning με την FaCT++ υποβάλλοντας το ίδιο ερώτημα είχαμε σωστό αποτέλεσμα.

Στην (Εικόνα 13) βλέπουμε ένα στιγμιότυπο του protégé 4 κατά τη στιγμή που έχουμε επιλέξει ως reasoner τον Pellet.



Εικόνα 13: Στιγμιότυπο επιλογής του reasoner Pellet

Για τις ανάγκες της δικής μας εργασίας, και εξαιτίας του γεγονότος ότι χρησιμοποιήσαμε την μηχανή Bossam για συμπερασμό, χρησιμοποιήσαμε ως μηχανή reasoning μια μηχανή που δημιουργείται από τη Bossam κατά τη διάρκεια εκτέλεσης του προγράμματος με τη χρήση βιβλιοθηκών Bossam-Java. Έτσι στο πρόγραμμα όταν εμείς θελήσουμε να υποβάλλουμε ερώτημα στην οντολογία, κάνουμε πρώτα ορισμό του reasoner. Όλα αυτά θα τα δούμε αναλυτικά στη συνέχεια.

Τελικά στην παρούσα εργασία χρησιμοποιούμε την έκδοση 3.3 του Protégé, αφού κρίθηκε πιο πλήρης, κυρίως εξαιτίας των επιπρόσθετων (add-ons), και των αρθρωμάτων (modules) που ενσωματώνει η έκδοση αυτή. Για παράδειγμα το άρθρωμα της οπτικοποίησης (visualization) δεν υπάρχει στην έκδοση 4. Να διευκρινίσουμε βέβαια ότι όλα αυτά



συνέβαιναν μέχρι τον Οκτώβριο περίπου του 2008, όπου και επιλέχθηκε τελικά η έκδοση *protégé* που χρησιμοποιήσαμε. Έτσι λοιπόν η συγκεκριμένη έκδοση 3.3 του *Protégé*, ενσωματώνει χαρακτηριστικά (features) και αρθρώματα (modules) τα οποία δεν έχουν ακόμα ενσωματωθεί στην έκδοση 4. Επίσης η έκδοση 4 δεν συνεργαζόταν σε άριστο βαθμό με την OWL. Ειδικά αυτό το τελευταίο ήταν και η αφορμή να επιστρέψουμε στην έκδοση 3.3, η οποία παρείχε πρόγραμμα εγκατάστασης *Protégé-OWL*.

Να αναφέρουμε ότι ένα μειονέκτημα της έκδοσης 3.3 είναι ότι δεν ενσωματώνει αυτόματους reasoners, αλλά στη δική μας περίπτωση αυτό δεν αποτέλεσε μειονέκτημα, αφού χρησιμοποιήσαμε εξωτερικό reasoner, όπως ήδη αναφέραμε.

Όπως αναφέρεται στη βιβλιογραφία [29], [12], [17] όταν δημιουργούμε μεγάλες οντολογίες είναι πολύ καλό να έχουμε ένα περιβάλλον το οποίο αυτοματοποιημένα να ελέγχει τη ορθότητα-σταθερότητα (consistency) της οντολογίας, ώστε σε κάθε βήμα να την έχουμε εξασφαλισμένη με σαφήνεια. Έτσι αποφεύγουμε τη δυσάρεστη κατάσταση κατά την οποία έχει φτιαχτεί μια μεγάλη οντολογία με αρκετές κλάσεις και υποκλάσεις και ξαφνικά ανακαλύπτουμε ότι δεν έχει λογική συνέπεια, με αποτέλεσμα να μην μπορούμε να υποβάλλουμε λογικά ερωτήματα. Αυτός είναι και ο κυριότερος λόγος, για τον οποίο στην έκδοση 4 ενσωματώθηκαν δύο reasoners.

## 10. Τεχνολογία Συμπερασμού (Inference-Reasoning)

Η διαδικασία συμπερασμού κάνοντας χρήση οντολογιών είναι μια από τις σημαντικότερες διαδικασίες. Μέσα από αυτήν μπορούμε να εξάγουμε χρήσιμα συμπεράσματα και να πάρουμε πληροφορίες για τις διαδικασίες που πρέπει να ακολουθηθούν. Είναι η βάση των συστημάτων τεχνητής νοημοσύνης (artificial intelligence και embedded systems) και των συστημάτων που συνεργάζονται στο σημασιολογικό ιστό. Για να επιτευχθεί συμπερασμός πρέπει να λειτουργήσει μια σειρά από μηχανές οι οποίες κάνουν η κάθε μια συγκεκριμένες δουλειές.

Καταρχήν πρέπει να υπάρξει η οντολογία η οποία όπως είδαμε θα πρέπει να έχει υποστεί τη διαδικασία της κρίσεως (reasoning), ώστε να ελεγχθεί για την ορθότητα-σταθερότητά της (consistency) και να βεβαιωθούμε ότι μπορεί να παράγει συμπερασμό. Στη συνέχεια πρέπει να υπάρξει η μηχανή που θα παράγει τα ερωτήματα. Μια μηχανή συμπερασμού βασίζεται-χρειάζεται μια κατάλληλη γλώσσα υποβολής ερωτημάτων. Υπάρχουν αρκετές τέτοιες γλώσσες. Για παράδειγμα, η γλώσσα SPARQL [20], είναι μια από τις πολλές γλώσσες υποβολής ερωτημάτων (query languages - QLs) τις οποίες μπορούμε να χρησιμοποιήσουμε



ώστε να υποβάλουμε ερωτήματα σε RDF και OWL οντολογίες. Μια άλλη γλώσσα υποβολής ερωτημάτων είναι η DL, η οποία μάλιστα είναι ενσωματωμένη στο προτέγε.

Έπειτα και επικουρικά θα πρέπει να υπάρχει μια μηχανή που να παράγει κανόνες και αλήθειες, ώστε να είναι δυνατή η εισαγωγή δυναμικά νέας γνώσης στις οντολογίες. Μια γλώσσα δημιουργίας κανόνων είναι η Semantic Web Rule Language (SWRL). Η SWRL (Semantic Web Rule Language) είναι μια πρόταση για Semantic Web rules-language, η οποία συνενώνει τις περιγραφικές γλώσσες OWL Web Ontology Language (OWL DL and Lite) και Rule Markup Language (Unary/Binary Datalog).

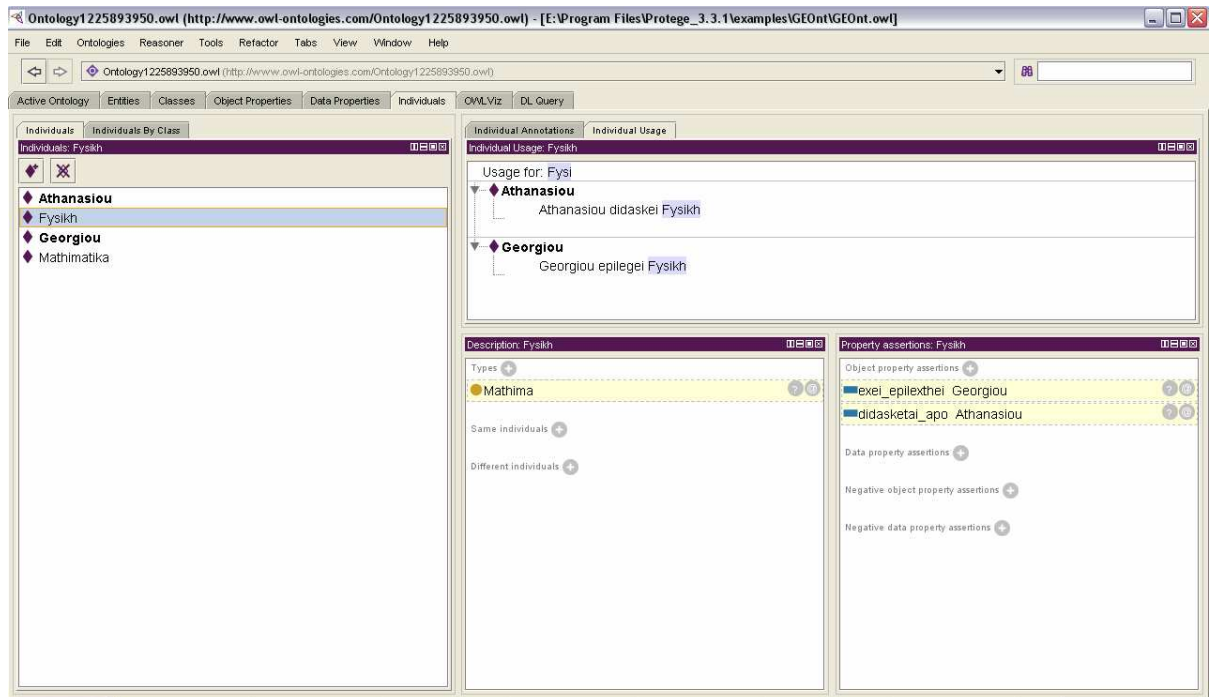
Επίσης υπάρχει η μηχανή Bossam [39] η οποία είναι μια μηχανή για κανόνες (SWRL) και OWL reasoning συμπερασματολογίας (OWL reasoning - an inference engine - a Semantic Reasoner).

Στα επόμενα θα δούμε μια διαδικασία παραγωγής συμπερασμάτων στη δική μας οντολογία, μέσα από την DL, εν είδει παραδείγματος

### 10.1. Δημιουργία individuals

Για να μπορέσουν να υποβληθούν ερωτήματα θα πρέπει στις οντολογίες να έχουν ορισθεί κάποια στιγμιότυπα (individuals-instances). Είδαμε στα προηγούμενα μια εικόνα με τέτοια στιγμιότυπα όπως τα έχουμε ορίσει (Εικόνα 8). Τα στιγμιότυπα είναι στην ουσία τα περιεχόμενα των οντολογιών. Για παράδειγμα αν «το μάθημα» είναι μια κλάση της οντολογίας, «η φυσική» είναι ένα στιγμιότυπό της. Άλλη μια εικόνα με στιγμιότυπα βλέπουμε στην (Εικόνα 14).



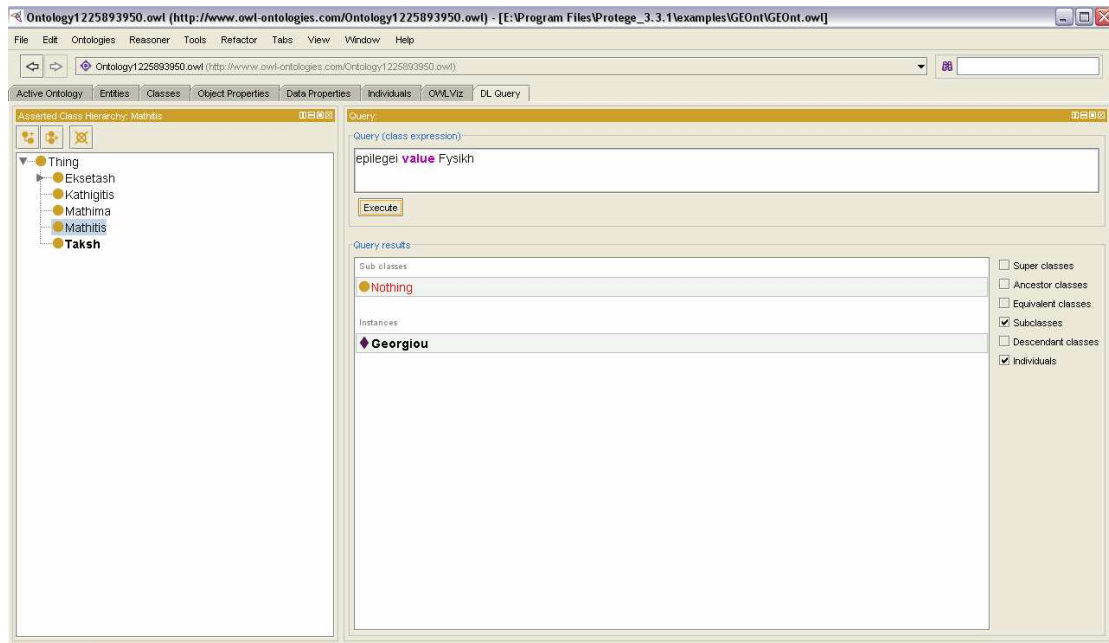


Εικόνα 14 : Εικόνα με στιγμιότυπα της οντολογίας

## 10.2. Υποβολή ερωτημάτων

Η DL (Description Logics) ή OWL-DL όπως αναφέρεται στη βιβλιογραφία είναι η μορφή της OWL η οποία επιτρέπει διαδικασίες υποβολής ερωτημάτων. Έχει συγκεκριμένο συντακτικό και με τη βοήθειά της μπορούμε να υποβάλλουμε ερωτήματα σε οντολογίες, αφού πρώτα τις έχουμε περάσει από τη διαδικασία της κρίσης (reasoning). Στο protégé υπάρχει ειδική καρτέλα (tab) η οποία φαίνεται στη (Εικόνα 15). Είναι και αυτή ένα από τα επιπρόσθετα που διαθέτει η έκδοση 3.3

Στην Εικόνα 15 θέτουμε στην οντολογία το ερώτημα, «ποιος έχει επιλέξει τη Φυσική ως μάθημα» :



Εικόνα 15 : Στιγμιότυπο από την εκτέλεση ερωτήματος DL

Το συντακτικό που χρησιμοποιούμε προκειμένου να υποβάλλουμε το ερώτημά μας, όπως φαίνεται και από την παραπάνω εικόνα είναι :

ερώτημα DL : «epilegei value Fysikh»

Δηλαδή για την υποβολή του ερωτήματος χρησιμοποιούμε μια σχέση (relation) από αυτές που έχουμε ορίσει στην οντολογία μας, τη σχέση «*epilegei*», τη δεσμευμένη λέξη «*value*», και ένα στιγμιότυπο που αναφέρεται σε ένα αντικείμενο της κλάσης «*μάθημα*», τη «*φυσική*». Η απάντηση που μας επιστρέφει η DL είναι :

απάντηση DL : «*Georgiou*»

### 10.3. Αλήθειες, Ερωτήματα και Κανόνες

Οι κανόνες βασίζονται στα υποσύνολα του First Order Logic (FOL) [17][48]. Οι βασικοί φορμαλισμοί κανόνων είναι οι Semantic Web Rule Language (SWRL) και Answer Set Programming (ASP). Είπαμε νωρίτερα ότι μια Οντολογία περιγράφει ένα πεδίο ενδιαφέροντος, προσδιορίζει έννοιες, περιγράφει σχέσεις ανάμεσα σε αυτές, και τέλος εισάγει στιγμιότυπα. Οι οντολογίες (τουλάχιστον στη μορφή OWL-DL) βασίζονται στην

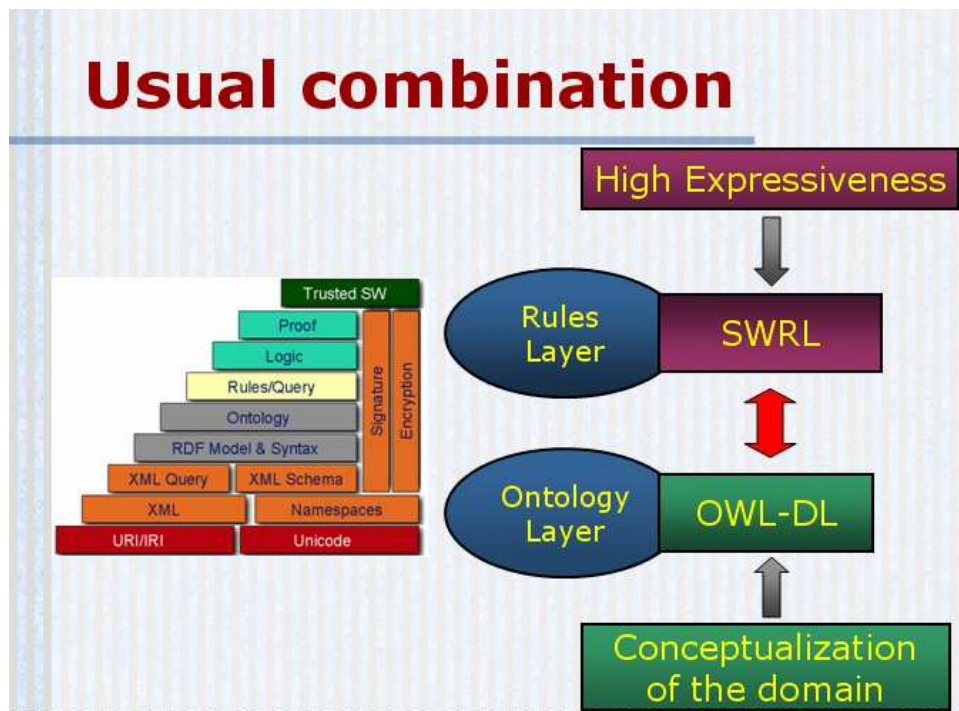


περιγραφική λογική (Description Logic), έτσι δεν μπορούν να έχουν την εκφραστικότητα που θα θέλαμε. Οι κανόνες έρχονται να συμπληρώσουν αυτό το κενό στην εκφραστικότητα. Όπως οι κανόνες, έτσι και οι αλήθειες είναι οι διαδικασίες με τις οποίες κάποιος μπορεί να εισάγει νέα γνώση σε μια οντολογία και στη συνέχεια να εξάγει συμπεράσματα. Ενώ με τα ερωτήματα μπορούμε να παράγουμε συμπερασμό.

Οι αλήθειες και τα ερωτήματα είναι εργαλεία πολύ χρήσιμα στο σημασιολογικό ιστό. Μπορούμε να θέτουμε αλήθειες σε μια οντολογία, και έπειτα να υποβάλλουμε ερωτήματα. Τα αποτελέσματα έπειτα από κατάλληλη επεξεργασία μπορούν να δώσουν τη δυνατότητα ορισμού νέων στιγμιότυπων στην οντολογία (individuals-instances), αλλά επιπρόσθετα και να εξάγουν συμπεράσματα. Οι κανόνες (rules) από την άλλη είναι διαδικασίες που ορίζονται και αυτές μέσα από κατάλληλες γλώσσες. Με τους κανόνες υπάρχει η δυνατότητα κατασκευής πιο σύνθετων ιδιοτήτων και συγχρόνως της αναζήτησης κάποιας απάντησης μέσα από ένα πιο σύνθετο ερώτημα. Η εκτέλεση ενός κανόνα έχει την ιδιότητα ενσωμάτωσης της νέας γνώσης και των συμπερασμάτων της στην οντολογία.

Στην πραγματικότητα, ο συνδυασμός αληθειών, ερωτημάτων και δομών επιλογής, αποφέρουν τα ίδια αποτελέσματα με την υποβολή κανόνων. Ο συνδυασμός είναι ελαφρά πιο ευέλικτος, γιατί έχει πιο μεγάλη ελευθερία στη σύνταξη και έχει πιο ενιαίο σχήμα κατά τη συγγραφή του, χρησιμοποιώντας διαφορετικές γλώσσες.

Στην (Εικόνα 16) [17] μπορούμε να δούμε μια σχηματική αναπαράσταση της διασύνδεσης της απαραίτητης εκφραστικότητας (με κανόνες) με τη διαδικασία περιγραφής κάποιου πεδίου (μέσω κάποιας οντολογίας).



Εικόνα 16: Σύνδεση εκφραστικότητας με περιγραφή πεδίου

## 11. Η μηχανή Bossam

Η μηχανή Bossam είναι ένα περιβάλλον διαχείρισης συμπερασματολογίας για οντολογίες. Είναι ένα αρκετά ισχυρό περιβάλλον, με δυνατότητα λειτουργία τόσο από τη γραμμή εντολών (command line), όσο και μέσα από ένα πρόγραμμα γραμμένο σε Java, αφού διαθέτει ένα αρκετά μεγάλο σύνολο συναρτήσεων για αυτήν. Η Bossam πριν κάνει οποιαδήποτε διαδικασία συμπερασμού, περνά την οντολογία από κρίση και επιβεβαίωση ορθότητας ταξινομίας (reasoning & confirmation-classification) μέσα από ένα δικό της σύστημα κρίσης (reasoning). Η διαδικασία είναι η παρακάτω [40] :

Για να φτιαχτεί ένας reasoner θα πρέπει να φτιαχτεί πρώτα μια reasoner factory. Ο ακόλουθος κώδικας μας δείχνει πώς δημιουργείται ένας reasoner.

```
import bossam.app.IReasonerFactory;
import bossam.app.ReasonerFactory;
import bossam.app.IReasoner;

IReasonerFactory reasonerFactory =
ReasonerFactory.getInstance();
IReasoner reasoner = reasonerFactory.createReasoner();
```



Ένα απαραίτητο στοιχείο της λειτουργίας της μηχανής Bossam είναι το γεγονός ότι πάντα στο classpath της εφαρμογής μας θα πρέπει να περιλαμβάνονται και οι βιβλιοθήκες της μηχανής Bossam. Αυτό είναι ένα σημαντικό στοιχείο το οποίο θα το δούμε και στη συνέχεια για την εφαρμογή που αναπτύσσουμε στα πλαίσια της εργασίας μας.

Παρακάτω παρουσιάζονται οι API specifications του IReasonerFactory.

- IReasoner createReasoner()
- IReasoner createOwlReasoner()
- IReasoner createSwrlReasoner()

Πιο κάτω βλέπουμε έναν οδηγό για το ποια μέθοδο θα ακολουθήσουμε ανάλογα τη γλώσσα, ώστε να φτιάξουμε έναν reasoner.

- Εάν θέλουμε να επεξεργαστούμε Buchingae και RuleML knowledge, χρησιμοποιούμε τη μέθοδο createReasoner().
- Εάν θέλουμε να επεξεργαστούμε OWL όπως επίσης και Buchingae και RuleML knowledge, χρησιμοποιούμε την createOwlReasoner().
- Εάν θέλουμε να επεξεργαστούμε SWRL όπως επίσης και Buchingae, OWL, και RuleML, χρησιμοποιούμε την createSwrlReasoner().

Στη συνέχεια εκτελεί κάποιο query SWRL και αφού πάρει υπόψη της κάποιους κανόνες δίνει τελικά συμπεράσματα.

Η μηχανή Bossam προσφέρει μια απλή διεπαφή Java για τη δημιουργία στιγμιοτύπων της οντολογίας, αληθειών, κανόνων με συναρτήσεις για τη διαχείρισή τους και για τη λειτουργία τους. Η κύρια συνάρτηση η οποία αρχικοποιεί όλες τις λειτουργίες της μηχανής Bossam είναι η etri.bossam.app.IReasonerFactory και η etri.bossam.app.IReasoner.

## 12. Buchingae – Η γλώσσα ορισμού κανόνων της Bossam

Η Buchingae είναι η γλώσσα ορισμού κανόνων της μηχανής Bossam. Είναι μια σχετικά απλή γλώσσα προσανατολισμένη στις εφαρμογές web (web-oriented) η οποία ενσωματώνεται στη μηχανή και της προσδίδει μεγάλες δυνατότητες αφού είναι δυνατή η απευθείας επικοινωνία με τα δεδομένα της URI. Αυτά τα δεδομένα είναι οι οντότητες, είτε σε μορφή OWL είτε σε RDF resources, είτε σε UML. Επίσης τέτοια δεδομένα μπορεί να είναι ακόμα και ερωτήματα



εκφρασμένα σε DL και SWRL τα οποία πιθανώς να έχουν διατυπωθεί από το περιβάλλον του protégé. Επίσης δεδομένα αποτελούν και τα αποτελέσματα τέτοιων ερωτημάτων.

Μερικοί κανόνες σύνταξης που θα χρησιμοποιήσουμε στην Buchingae είναι :

$?x(?p)$        $?p$  είναι  $?x$ .

$?p(?x,?y)$   $?p$  του  $?x$  είναι  $?y$ . (ή " $?x$  έχει το  $?y$  ως  $?p$ ")

$?p$              $?p$ . ( $?p$  will be bound to all the facts which have no terms.)

**Πίνακας 1** : Μερικοί βασικοί κανόνες σύνταξης της Buchingae

Η Buchingae [41] περιλαμβάνει όλους τους γνωστούς μας γνωσιακούς τύπους από τις οντολογίες. Μπορεί να ορίσει αλήθειες (facts), κανόνες (rules), κλάσεις (classes), σχεσιακές ιδιότητες (properties), στιγμιότυπα (instances ή individuals) και τέλος συναρτήσεις (functions).

Στις επόμενες παραγράφους περιγράφουμε τα βασικά γνωσιακά στοιχεία της Buchingae[40], [41].

### 12.1. Αλήθειες (Facts)

Η αλήθεια (fact) είναι η πιο πολυχρησιμοποιούμενη μαζί με τους κανόνες γνωσιακή οντότητα της Buchingae. Θα μπορούσαμε να μεταφράζουμε τον όρο και ως «πραγματικότητα» ή «πραγματικό γεγονός». Στο παρόν κείμενο προτιμήσαμε να μεταφράζουμε παντού τον όρο ως «αλήθεια» επειδή ο όρος είναι μικρός, σαφής και εννοεί ακριβώς αυτό που θέλει να περιγράψει ο όρος «fact». Με τις «αλήθειες» ορίζουμε στην οντολογία μας ποια πράγματα θεωρούμε αλήθειες. Στην πραγματικότητα όμως, αυτό που μπορούμε να κάνουμε με μια αλήθεια είναι να εισάγουμε νέα γνώση στην οντολογία μας. Μια αλήθεια ορίζεται ως ακολούθως :

```
fact factID is predicate-symbol[(constant-term { ,constant-term } )];
```

όπου factID είναι ένα τυπικό όνομα το οποίο δείχνει μια καθορισμένη αλήθεια. Πρέπει να τονίσουμε ότι το όνομα μιας αλήθειας είναι και το χαρακτηριστικό που της δίνει υπόσταση. Η λογική ύπαρξης ορισμού και διάρκειας ζωής μιας αλήθειας θυμίζει τον τρόπο λειτουργίας των access-lists σε έναν δρομολογητή ή σε μια ip-chain στο linux. Το κυρίως σώμα της αλήθειας είναι μια δήλωση (predicate), η οποία δημιουργείται από σταθερά κατηγορήματα



σύμβολα (constant predicate symbols) και σταθερούς όρους (constant terms). Στο τέλος δε της δήλωσης, τοποθετούμε ένα ελληνικό ερωτηματικό (semicolon). Στα παρακάτω παραθέτουμε κώδικα ο οποίος περιέχει αλήθειες και στη συνέχεια παραθέτουμε τα αποτελέσματα από την εκτέλεση ώστε να δούμε και πώς λειτουργούν με παραδείγματα.

## 12.2. Κανόνες (Rules)

Ο κανόνας είναι και αυτός μια διαδικασία που χρησιμοποιείται πολύ συχνά και έχει ως αποτέλεσμα τελικώς την εξαγωγή συμπερασμάτων από την οντολογία. Η λειτουργία του είναι λίγο πιο σύνθετη από αυτήν της αλήθειας, αλλά και ο κανόνας είναι μια δομή για εισαγωγή νέας γνώσης σε μια οντολογία. Όμως ο κανόνας δε μένει μόνο στην εισαγωγή της νέας γνώσης. Με τον κανόνα μπορούμε να προχωρήσουμε και σε συμπερασμό με την βοήθεια λογικών ερωτημάτων. Όπως έχουμε αναφέρει ήδη, ένας κανόνας είναι από μόνος του ένας συνδυασμός, αλήθειας, ερωτήματος και δομής επιλογής εάν (if). Ο κανόνας συντάσσεται δε ως ακολούθως :

```
rule ruleID is if condition-part then conclusion-part;
```

Η πρώτη λέξη (rule) είναι δεσμευμένη. Στο δεύτερο μέρος (ruleID) ορίζουμε το όνομα του κανόνα, ενώ στο επόμενο μέρος ορίζουμε τη συνθήκη (condition-part) του κανόνα. Η συνθήκη είναι μια λογική πρόταση. Τέλος στο τελευταίο μέρος (conclusion-part) ορίζουμε το αποτέλεσμα του κανόνα. Τι θα γίνει δηλαδή, εάν ο το τρίτο μέρος είναι αληθές. Στην ουσία δηλαδή έχουμε όλη τη διαδικασία του συμπερασμού σε μια πρόταση.

## 12.3. Προτεραιότητα Κανόνα (Rule Priority)

Για τον κανόνα μπορούμε να ορίσουμε και καταστάσεις προτεραιότητας. Η προτεραιότητα δηλώνεται με μια τιμή προτεραιότητας, η οποία μπορεί να είναι ένας θετικός ή ένας αρνητικός ακέραιος. Η τιμή αυτή τοποθετείται σε παρένθεση δίπλα στο αναγνωριστικό όνομα (ruleID) του κανόνα. Μεγάλος αριθμός, σημαίνει υψηλή προτεραιότητα εκτέλεσης. Τι σημαίνει όμως υψηλή προτεραιότητα εκτέλεσης; Σημαίνει ότι προηγείται από κάθε άλλον απλό κανόνα ή κάθε άλλο κανόνα ο οποίος πρόκειται να εκτελεστεί και έχει χαμηλότερη προτεραιότητα. Στο επόμενο παράδειγμα έχουμε δώσει στον κανόνα example προτεραιότητα 10.

```
rule example(10) is if <συνθήκη> then <αποτέλεσμα>;
```



#### 12.4. Class

Επίσης στην Buchingae, μπορούμε να ορίσουμε νέες κλάσεις για την οντολογία μας. Ο ορισμός μιας νέας κλάσης γίνεται με την ακόλουθη σύνταξη :

```
class classID [inherits classID { , classID}];
```

όπως και στα προηγούμενα οι δύο πρώτες λέξεις είναι η δεσμευμένη λέξη και το αναγνωριστικό όνομα της νέας κλάσης. Η λέξη inherits στη συνέχεια, κατά κυριολεξία σημαίνει «κληρονομώ» και στην ουσία αυτό θέλει να δηλώσει για την οντολογία. Δηλαδή από ποια κλάση κληρονομεί ιδιότητες, ή αλλιώς ποιας κλάσης είναι υποκλάση. Στην Buchingae δεν μπορούμε να ορίσουμε αρχικές κλάσεις, παρά μόνο υποκλάσεις. Άρα θα πρέπει να έχουν ορισθεί κάποιες κλάσεις στην οντολογία, πριν προσπαθήσουμε να δημιουργήσουμε κάποιες υποκλάσεις, ώστε αυτή να κληρονομήσει όλες τις ιδιότητες από την κλάση γονέα. Οι ιδιότητες ή σχέσεις (properties-relations) δημιουργούνται στη συνέχεια, με ξεχωριστές εντολές. Στην ουσία η διαδικασία αυτή είναι η δημιουργία νέας γνώσης στην οντολογία. Στη συνέχεια βλέπουμε ένα παράδειγμα με τη σύνταξή του.

```
class Human inherits Animal;
```

Με την παραπάνω εντολή δημιουργούμε την κλάση Human η οποία είναι υποκλάση στην Animal. Στην επόμενη παράγραφο περιγράφουμε την εντολή με την οποία δημιουργούμε ιδιότητες (σχέσεις) για τη νέα οντολογία.

#### 12.5. Property

Όπως είπαμε ήδη, μετά τη δημιουργία υποκλάσεων μπορούμε να δηλώσουμε ιδιότητες-σχέσεις για αυτές,. Η σύνταξη της εντολής είναι η ακόλουθη :

```
property propertyID for classID { ,classID} is datatypeID;
```

Σύμφωνα και με τα όσα έχουμε περιγράψει είναι κατανοητό τι κάνει η κάθε λέξη στην παραπάνω εντολή. Η αρχική λέξη property είναι δεσμευμένη ενώ η κάθε ιδιότητα ορίζεται για συγκεκριμένη κλάση. Μια ιδιότητα μπορεί να ορισθεί ως ιδιότητα για πολλές κλάσεις και αυτό προσδιορίζεται από τη λίστα των classID που υπάρχει μέσα στα άγκιστρα. Το datatypeID είναι ο τύπος της ιδιότητας. Δηλαδή η ιδιότητα μπορεί να πάρει τιμές τύπου datatypeID. Για παράδειγμα :





```
property age for Human is xsd:positiveInteger;
```

Δηλαδή στην κλάση Human ορίζουμε μια ιδιότητα age η οποία θα μπορεί να παίρνει τιμές θετικές ακέραιες. Οι δεσμευμένες λέξεις για τους τύπους των ιδιοτήτων είναι οι ίδιες με αυτές που συναντούμε και στο protégé όταν έχουμε οντολογίες είτε RDFS, είτε QWL, είτε XML. Για έναν πλήρη πίνακα με τα xsd data type references μπορούμε να δούμε το [45].

## 12.6. Στιγμιότυπα (Instances ή Individuals)

Η επόμενη οντότητα που πρέπει να μπορούμε να ορίσουμε με τη βοήθεια της Buchingae θα πρέπει να είναι τα στιγμιότυπα, τα οποία όπως είδαμε είναι οι φυσικές οντότητες οι οποίες δίνουν σημασία σε όλες τις προηγούμενες μορφές (κλάσεις και ιδιότητες). Η επόμενη εντολή μας δίνει τη δυνατότητα δημιουργίας ενός στιγμιότυπου με την εξής σύνταξη :

```
individual individualID is classID [and propertyID = value {,  
propertyID = value};
```

Με το παρακάτω παράδειγμα αποσαφηνίζεται πλήρως ο ρόλος των λέξεων κλειδιών της παραπάνω εντολής :

```
individual John is Mathitis and age = 17, name = "John Ioannou",  
birthdate = 1992-02-28;
```

Το Mathitis είναι το αναγνωριστικό της κλάσης για την οποία προσθέτουμε στιγμιότυπο με αυτήν την εντολή. Το όνομα του στιγμιότυπου είναι John, ενώ δίνουμε τιμή και σε κάποιες άλλες ιδιότητες της κλάσης. Το όνομα (John Ioannou), την ηλικία (17) και την ημερομηνία γέννησης (1992-02-28).

Ένα ιδιαίτερο σημείο που πρέπει να τονίσουμε είναι το γεγονός ότι αντί να γράφουμε την ιδιαίτερη σύνταξη της εντολής δημιουργίας στιγμιότυπου, μπορούμε να γράψουμε μια αλήθεια και να επιτύχουμε το ίδιο αποτέλεσμα. Στη συγκεκριμένη εργασία ενεργούμε και εργαζόμαστε με αυτή τη λογική και αναδεικνύουμε τη δύναμη ορισμού στιγμιότυπων με την εντολή δημιουργίας αλήθειας, η οποία μας δίνει ενιαίο τρόπο εργασίας για την εισαγωγή νέας γνώσης.

Έτσι λοιπόν αντί να γράφουμε την παραπάνω ιδιαίτερη εντολή μπορούμε να γράψουμε :

```
fact forJohn1 is Person(John);  
fact forJohn2 is age(John,17);  
fact forJohn3 is name(John,"John Ioannou");
```



Στα παρακάτω θα δείξουμε το αποτέλεσμα από διάφορες εκτελέσεις τέτοιων εντολών στη δική μας οντολογία και θα παραθέσουμε τα αποτελέσματα που παίρνουμε. Η λειτουργία των εντολών της Buchingae γίνεται με τη βοήθεια της Java.

### 12.7. Συναρτήσεις (Functions)

Στην Buchingae, δεν υπάρχει κάποιο συγκεκριμένο συντακτικό για τη δημιουργία συναρτήσεων. Υπάρχει όμως η δυνατότητα χρήσης έτοιμων. Στον Πίνακα 2 υπάρχει μια λίστα με τις έτοιμες συναρτήσεις που μπορεί κανείς να χρησιμοποιήσει. Όπως μπορούμε να δούμε υπάρχουν συναρτήσεις διαφόρων ειδών. Για σύγκριση, αριθμητικές κλπ. Πιο συγκεκριμένα για σύγκριση υπάρχουν 6 έτοιμες συναρτήσεις, οι: lessThan(<), lessThanOrEquals(<=), equals(=), notEquals(!=), greaterThan(>), greaterThanOrEquals(>=). [41]

Builtin Name	Descriptions
func:equals(term1, term2)	Returns true if term1 equals to term2.
func:greaterThan(term1, term2)	Returns true if term1 is greater than term2.
func:greaterThanOrEquals(term1, term2)	Returns true if term1 is greater than or equal to term2.
func:lessThan(term1, term2)	Returns true if term1 is less than term2.
func:lessThanOrEquals(term1, term2)	Returns true if term1 is less than or equal to term2.
func:add(term1, term2)	Returns the sum of term1 and term2.
func:subtract(term1, term2)	Returns the subtraction of term2 from term1.
func:multiply(term1, term2)	Returns the product of term2 by term1.
func:divide(term1, term2)	Returns the division of term1 by term2.
func:mod(term1, term2)	Returns the mod of term1 by term2.
func:after(term1, term2)	Returns true if term1 is after term2. term1 and term2 should designate time, date, or datetime.
func:containedIn(term1, term2, term3)	Returns true if term1 is in the period of time beginning at term2 and ending at term3.



func:before(term1, term2)	Returns true if term1 is before term2. term1 and term2 should designate time, date, or datetime.
func:dateTime(year,month,date,hour,min,sec)	Returns a datetime constant built by the parameters.

**Πίνακας 2 :** Έτοιμες Συναρτήσεις Buchingae

Οι συναρτήσεις αυτές μπορούν να χρησιμοποιηθούν είτε ως απλές συναρτήσεις, είτε από τα αντίστοιχά τους απλά μαθηματικά σύμβολα (με τη βοήθεια μαθηματικών τελεστών). Στα επόμενα παραδείγματα, μπορούμε να δούμε και τις δύο χρήσεις :

```
rule r1 is if some(?x) and some(?y) and test(greaterThan(?x,?y)) then
...
rule r2 is if some(?x) and some(?y) and [?x < ?y] then ...
```

Στα παραπάνω παραδείγματα η λέξη test είναι μια εσωτερική δήλωση της Buchingae, η οποία χρησιμοποιείται ώστε να γίνει κλήση στις λογικές εκφράσεις που περιλαμβάνονται στον Πίνακα 2.

Οι μαθηματικοί τελεστές που μπορούμε να χρησιμοποιήσουμε είναι οι: +, -, /, \*, %. Πρόσθεση, αφαίρεση, ακέραια διαίρεση, πολλαπλασιασμός, ακέραιο ηλίκο. Στα παρακάτω παραδείγματα μπορούμε να δούμε τη χρήση μαθηματικών τελεστών στον ορισμό ενός κανόνα.

```
rule r1 is if some(?x) and some(?y) and checkIt([?x+(?y-1)*2]) then
conclude([?x+?y]);
rule r2 is if some(?x) and some(?y) and [?x-(?y+1)/(?y-1) < 0] then
conclude(?x,[?y+1]);
rule r3 is if some(?x) and some(?y) then conclude([1/2]);
```

Πρέπει να είμαστε προσεκτικοί με τη σύνταξη, ώστε να περικλείουμε τις εκφράσεις σε αγκύλες : [].

## 13. Διαδικασίες διαχείρισης γνώσης και έκδοσης αποτελεσμάτων

### 13.1. Φόρτωση OWL αρχείων και Querying

Όπως έχουμε ήδη δει, μια URI (Uniform Resource Identifier) είναι μια συνεκτική μορφή η οποία προσδιορίζει μια πηγή πληροφορίας στο διαδίκτυο. Είδαμε (στο κεφάλαιο 7) ότι το βασικό πλεονέκτημα της χρήσης της URI είναι η διαδικτυακή πρόσβαση στα δεδομένα και μάλιστα με πρωτόκολλο http.



Η URL ένας πιο γνωστός όρος, και δεν είναι στην πραγματικότητα παρά μια URI η οποία εκτός από τα δεδομένα μπορεί να μας παρέχει και μια αναπαράσταση των δεδομένων. Μια ακόμα κρίσιμη έννοια είναι αυτή του πεδίου ονομάτων (namespace) [36]. Το πεδίο αυτό είναι ένα περιεκτικός κλωβός (abstract container) ο οποίος μας παρέχει ονοματολογία για τα στοιχεία-αντικείμενα του πλαισίου (του περιβάλλοντος). Η ονοματολογία είναι ονόματα, τεχνικοί όροι, ή απλές λέξεις. Ο κυριότερος ρόλος του πεδίου ονομάτων είναι η αποσαφήνιση των στοιχείων-αντικειμένων τα οποία πιθανώς να έχουν και το ίδιο όνομα αλλά σε διαφορετικά πεδία. Για παράδειγμα ένα εργαλείο μπορεί να ονομάζεται διαφορετικά σε κάποιο namespace και ένα άλλο όνομα σε κάποιο άλλο namespace.

Ως κανόνα, πρέπει να έχουμε στο νου μας ότι σε ένα namespace, ένα όνομα δεν μπορεί να έχει διπλή σημασία και επιπλέον ένα αντικείμενο δεν μπορεί να έχει δύο ονόματα. Όμως ένα όνομα μπορεί να αντιστοιχεί σε άλλο αντικείμενο σε ένα διαφορετικό namespace. Τα ονόματα του namespace μπορούν να αναπαριστούν αντικείμενα, έννοιες, σε φυσική ή τεχνική γλώσσα.

Για να κατανοήσουμε πιο καλά τι είναι το namespace μπορούμε να πούμε ότι το namespace μιας γλώσσας προγραμματισμού είναι τα αναγνωριστικά της (identifiers). Σε ένα λειτουργικό σύστημα, ένα παράδειγμα namespace είναι ο κατάλογος (directory). Περιέχει αντικείμενα, τα οποία μπορούν να έχουν μοναδικά ονόματα. Τα ίδια ονόματα μπορούμε να τα συναντήσουμε και σε κάποιο άλλον κατάλογο (directory) άρα σε ένα άλλο namespace.

Η οντολογία γενικά ως αντικείμενο, περιέχει και διαμοιράζει στατικά δεδομένα [13]. Όμως στις σύγχρονες εφαρμογές κάποια πράγματα αλλάζουν πολύ εύκολα, όπως οι συνθήκες περιβάλλοντος. Τέτοια δεδομένα πλέον είναι πολύ εύκολο να συλλεχθούν από αισθητήρες. Αυτή η νέα γνώση μπορεί να δώσει χρήσιμα συμπεράσματα στην εφαρμογή. Η διαδικασία παραγωγής συμπερασμάτων ονομάζεται συμπερασμός. Η μηχανή Bossam με τη βοήθεια της Buchingae, μας δίνει τα απαραίτητα εργαλεία ώστε να καταφέρουμε να εισάγουμε νέα γνώση δυναμικά σε μια οντολογία, χρησιμοποιώντας τα στοιχεία της (κλάσεις, ιδιότητες-συσχετίσεις) ως πεδίο ονοματολογίας (namespace), και στη συνέχεια να προχωρούμε σε συμπερασμό με βάση διάφορους κανόνες που ορίζουμε με τη βοήθεια δομών της γλώσσας προγραμματισμού.

Στα επόμενα δείχνουμε με παράδειγμα, όσα περιγράψαμε στην προηγούμενες ενότητες. Σε πρώτη φάση θα δούμε πώς φορτώνουμε ένα αρχείο OWL, κάνοντας reasoning, και εφαρμόζοντας ερωτήματα. Πρέπει να διευκρινίσουμε ότι η συγκεκριμένη διαδικασία εκτελείται σε περιβάλλον γραμμής εντολών (command line). Το δοκιμάσαμε τόσο σε περιβάλλον κονσόλας DOS (Windows XP) όσο και σε περιβάλλον κονσόλας linux.



```
// Γίνεται ορισμός του reasoner ο οποίος θα πιστοποιήσει το
// consistency της οντολογίας

IRReasonerFactory reasonerFactory = ReasonerFactory.getInstance();
IRReasoner r = reasonerFactory.createOwlDlReasoner();

// Φορτώνουμε την οντολογία wine από συγκεκριμένη URL

r.load(IRReasoner.OWL, "http://www.w3.org/TR/2004/REC-owl-guide-
20040210/wine.rdf");

// θέτουμε τα prefixes για το πεδίο ονοματολογίας

r.setNamespacePrefix("wine", "http://www.w3.org/TR/2003/PR-owl-
guide-20031209/wine#");
r.setNamespacePrefix("owl", "http://www.w3.org/2002/07/owl#");

// πριν θέσουμε οποιοδήποτε ερώτημα κάνουμε το
// προαπαιτούμενο Reasoning

try
{
    r.run();
}
catch (InconsistencyException e)
{
    e.printStackTrace();
    return;
}

// τέλος, θέτουμε ερώτημα

String answer = r.ask("query q is owl:Class(wine:WhiteTableWine);");
```

Ο παραπάνω κώδικας φορτώνει την οντολογία Wine η οποία είναι διαθέσιμη στον ιστοχώρο της W3C [46]. Με την ερώτηση που θέτουμε, ρωτούμε στην ουσία αν η οντολογία είναι άδεια ή όχι. Το αποτέλεσμα είναι αληθές, το οποίο καταδεικνύεται από το γεγονός ότι η απάντηση δεν είναι null και η `answer.isEmpty()` είναι αληθής.

Στο επόμενο παράδειγμα ζητούμε όλα τα στιγμιότυπα (instances- individuals) της wine.

```
// ένα άλλο ερώτημα

String answer = r.ask("query q is wine:Wine(?x);");

// εμφάνιση στην οθόνη του αποτελέσματος

System.out.println(answer);
```



Το αποτέλεσμα που βλέπουμε είναι το ακόλουθο (την παρακάτω έξοδο τη βλέπουμε σε περιβάλλον κονσόλας, στο οποίο εκτελούμε και τον παραπάνω κώδικα) :

```
{[x = http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine#PageMillWineryCabernetSauvignon]}
{[x = http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine#KathrynKennedyLateral]}
{[x = http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine#SevreEtMaineMuscadet]}
{[x = http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine#VentanaCheninBlanc]}
{[x = http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine#RoseDAnjou]}
{[x = http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine#ChateauLafiteRothschildPauillac]}
{[x = http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine#MariettaOldVinesRed]}
{[x = http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine#FormanChardonnay]}
```

...μια μεγάλη λίστα αποτελεσμάτων...

```
{[x = http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine#SaucelitoCanyonZinfandel1998]}
{[x = http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine#SelaksSauvignonBlanc]}
{[x = http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine#FoxenCheninBlanc]}
{[x = http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine#FormanCabernetSauvignon]}
```

Το αποτέλεσμα είναι το εξής : Η μεταβλητή x παίρνει όλες τις τιμές των στιγμιότυπων. Τα στιγμιότυπα δε, παρουσιάζονται με πρόθεμα την URI της οντολογίας. Για παράδειγμα στο τελευταίο αποτέλεσμα, το x έχει τιμή FormanCabernetSauvignon, η οποία είναι στιγμιότυπο της οντολογίας wine, η οποία βρίσκεται στην URL <http://www.w3.org/TR/2003/PR-owl-guide-20031209/>

Στην επόμενη παράγραφο παρουσιάζουμε τις ίδιες διαδικασίες σε περιβάλλον Java.

### 13.2. Εκτέλεση σε περιβάλλον Java

Περίπτωση εκτέλεσης μέσα από το περιβάλλον μιας εφαρμογής Java. Στην περίπτωση αυτή ο κώδικας μοιάζει με το παρακάτω :

```
public class WineQuery01
{
    // Ορίζουμε το prefix της οντολογίας και παράλληλα
    // ορίζουμε το URI της οντολογίας
```



```
// Η οντολογία GEOnt βρίσκεται σε έναν προσωπικό server
```

```
final static String GEOntURI =  
"http://vsdesign.homelinux.org/~bstefan/GEOnt.owl#";
```

Γενικά το όνομα της URI εξαρτάται από το σημείο στο οποίο θα ανεβάσουμε την οντολογία μας. Αντίθετα το namespace εξαρτάται κατά κύριο λόγο από το protégé και σχετίζεται με το πρότυπο που χρησιμοποιούμε κατά τη στιγμή της δημιουργίας της οντολογίας (Εικόνα 1). Μπορούμε να τη δούμε στην περιοχή των facts της εκτέλεσης. Από τη στιγμή που δημιουργηθεί η οντολογία και μετά, μπορούμε να βρούμε αυτό το namespace στο περιβάλλον του protégé (Εικόνα 17).

```
// Εδώ λοιπόν ορίζουμε το prefix για το namespace
```

```
final static String ruleURI = "http://www.owl-  
ontologies.com/Ontology1225893950.owl#";
```

```
public static void main(String[] args)  
{  
    try  
    {
```

```
        // δημιουργούμε τον reasoner
```

```
        IReasonerFactory factory = ReasonerFactory.getInstance();  
        IReasoner r = factory.createOwlDlTrMReasoner();
```

Η πλήρης ονομασία του reasoner που δημιουργείται σε αυτό το σημείο είναι : rule-based OWL DL reasoner

```
        // Φορτώνουμε την οντολογία GEOnt στη μηχανή Bossam
```

```
        r.load(IReasoner.OWL, GEOntURI);
```

```
        // εκτελούμε το reasoning
```

```
        String result = r.run();
```

```
        // Εδώ εκτυπώνεται στην οθόνη το αποτέλεσμα που προέκυψε από την  
        // εκτέλεση του reasoning  
        System.out.println("Conclusions: n" + result);
```

Πρέπει να πούμε ότι όταν εκτελούμε ένα reasoning, η μηχανή τοποθετεί στην μεταβλητή result, όλα τα περιεχόμενα της οντολογίας. Έτσι με την παραπάνω εντολή, αυτό που καταφέρνουμε είναι να τυπώσουμε στην οθόνη όλα τα περιεχόμενα της οντολογίας.

```
        // θέτουμε το namespace prefix της οντολογίας wine ως «w»  
        // από δω και πέρα όταν θέλουμε να αναφερθούμε στην οντολογία  
        // θα αναφερόμαστε σε αυτήν με το «w»
```



```
r.setNamespacePrefix("w", ruleURI);

// εκτελούμε ένα ερώτημα και το αποτέλεσμα αποθηκεύεται
// στην μεταβλητή answer
// Η ερώτηση είναι : βρες τα στιγμιότυπα της οντολογίας
// που είναι mathitis και τοποθέτησέ τα στη μεταβλητή x

Answer answer = r.ask1("query q is w:Mathitis(?x);");

// Τύπωσε το αποτέλεσμα

// Prints out the query result
if (answer == null)
{
    System.out.println("The query returns false!");
}
else
{
    System.out.println("Answer (" + answer.getBindings().size() +
":n"
+ answer);
}

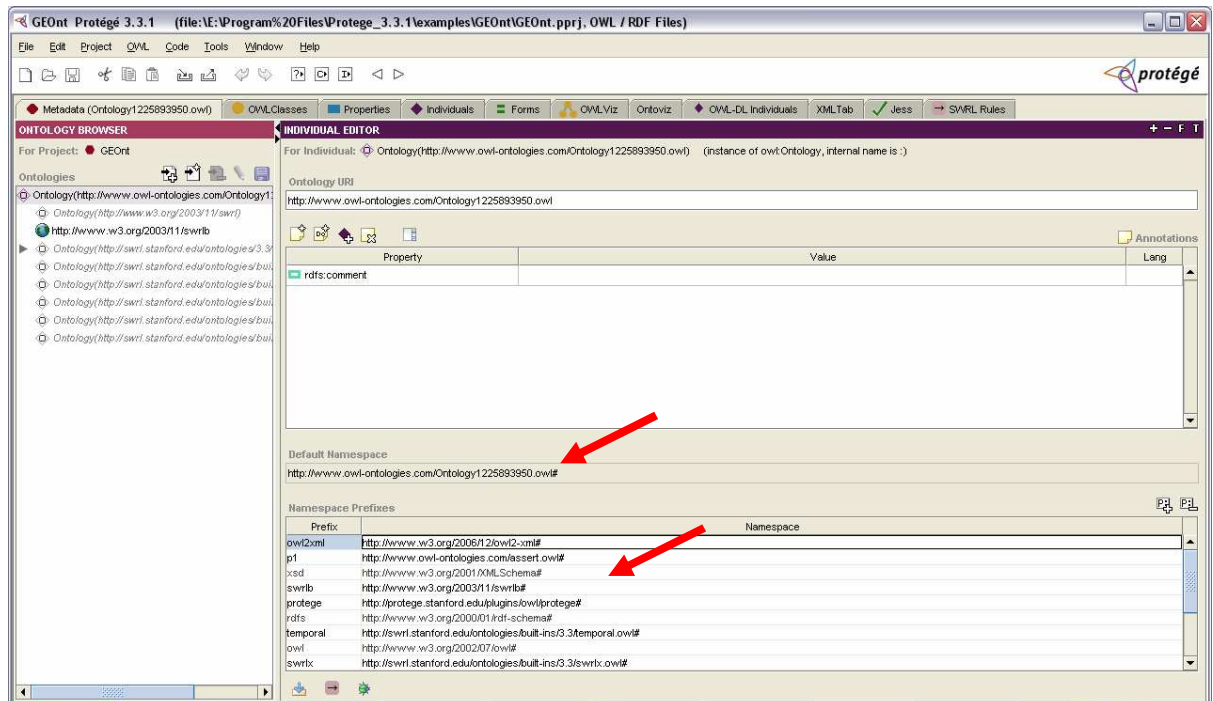
// κατά τα γνωστά για να εκτελεστεί ένα πρόγραμμα Java, πρέπει να
// εξασφαλίσουμε ότι θα πιάσουμε τα exceptions

catch (Exception e)
{
    e.printStackTrace();
}
}
```

Ολόκληρο το πρόγραμμα παρουσιάζεται στο (παράρτημα Α – κώδικας 1)

Στην *Εικόνα 17*, βλέπουμε ένα στιγμιότυπο από το περιβάλλον του protégé .Στην εικόνα αυτή, μπορούμε να διακρίνουμε, πού μπορεί κανείς να βρει το namespace της οντολογίας του.





Εικόνα 17 : Σημεία στο protégé στα οποία μπορούμε να δούμε το namespace (κόκκινο βέλος)

το αποτέλεσμα της εκτέλεσης του (παράρτημα Α – κώδικας 1) είναι το ακόλουθο :

```
.....  
.....  
fact http://bossam.com/default#Fact136 is neg  
http://www.w3.org/2002/07/owl#Nothing(http://www.owl-  
ontologies.com/Ontology1225893950.owl#Georgiou);  
fact http://bossam.com/default#Fact137 is neg  
http://www.w3.org/2002/07/owl#Nothing(http://www.owl-  
ontologies.com/Ontology1225893950.owl#Petrou);  
fact http://bossam.com/default#Fact138 is neg  
http://www.w3.org/2002/07/owl#Nothing(http://www.owl-  
ontologies.com/Ontology1225893950.owl#Mathimatika);  
fact http://bossam.com/default#Fact139 is neg  
http://www.w3.org/2002/07/owl#Nothing(http://www.owl-  
ontologies.com/Ontology1225893950.owl#Fysikh);  
fact http://bossam.com/default#Fact140 is neg  
http://www.w3.org/2002/07/owl#Nothing(http://www.owl-  
ontologies.com/Ontology1225893950.owl#Roumelis);  
fact http://bossam.com/default#Fact141 is neg  
http://www.w3.org/2002/07/owl#Nothing(http://www.owl-  
ontologies.com/Ontology1225893950.owl#Athanasiou);  
fact http://bossam.com/default#Fact142 is neg  
http://www.w3.org/2002/07/owl#Nothing(http://www.owl-  
ontologies.com/Ontology1225893950.owl#x);  
Answer (2):n{[?x=http://www.owl-  
ontologies.com/Ontology1225893950.owl#Georgiou]}  
{[?x=http://www.owl-ontologies.com/Ontology1225893950.owl#Petrou]}
```



Στην εκτέλεση παρατηρούμε ότι πριν προκύψει το αποτέλεσμα (αυτό που φαίνεται με έντονα κόκκινα γράμματα), έχουμε μια παράθεση των αληθειών της οντολογίας. Και για το πρόγραμμα αλήθειες είναι τόσο η επιπρόσθετη γνώση που εισάγουμε στην οντολογία, όσο και η αρχική, όπως την έχουμε εισάγει «με το χέρι» στην οντολογία από το προτέγε (ή οποιοδήποτε άλλο εργαλείο διαχείρισης οντολογιών).

Είναι βέβαια προφανές από την ανάγνωση των προηγούμενων ενοτήτων, αλλά μπορούμε να διευκρινίσουμε το γεγονός ότι η κλάση *mathitis* έχει ορισθεί στην οντολογία από το περιβάλλον του προτέγε.

Παρακάτω βλέπουμε ένα τμήμα κώδικα στον οποίο εφαρμόζουμε κανόνες και αλήθειες :

Στο (παράρτημα Α – κώδικας 2), μπορούμε να δούμε ακόμη ένα παράδειγμα κώδικα σε Java, με το οποίο δημιουργούμε κάποιον *reasoner*, εισάγουμε αλήθειες, και εντέλει θέτουμε ερωτήματα και παίρνουμε αποτελέσματα. Το επιπρόσθετο σε αυτόν τον κώδικα σε σχέση με αυτά που είδαμε στον κώδικα 1, είναι ότι τώρα με τα δύο ερωτήματα που θέτουμε παίρνουμε ως αποτέλεσμα και την αποθηκευμένη «με το χέρι» γνώση, αλλά και τη νέα γνώση που εισάγουμε με αλήθειες.. Ορίζουμε λοιπόν δύο αλήθειες :

```
r.tell("fact a is didaskei(w:John,w:Gymnastikh);");  
r.tell("fact b is exei_epilexthei(w:Gymnastikh,w:nil);");
```

και ζητούμε με υποβολή ερωτήματος την εξαγωγή ήδη αποθηκευμένης αλλά και νέας γνώσης στην οθόνη :

```
String result3 = r.ask("query q is didaskei(?x,?y) and  
exei_epilexthei(?y,?z);");
```

Επίσης ζητούμε και όλα τα δεδομένα της οντολογίας :

```
String result2 = r.run();
```

Η εκτέλεση του κώδικα (παράρτημα Α – κώδικας 2) δίνει :

```
// Edw einai to result2  
// O Petrou einai mathitis toy Roumeli  
// Ogeorgiou einai mathitis kai toy Roumeli kai toy Athanasiou  
  
fact http://bossam.com/default#Fact211 is http://www.owl-  
ontologies.com/Ontology1225893950.owl#einai_mathitis(http://www.owl-  
ontologies.com/Ontology1225893950.owl#Petrou,http://www.owl-  
ontologies.com/Ontology1225893950.owl#Athanasiou);
```



```
// Edw einai to result3
```

```
{[?z = http://www.owl-ontologies.com/Ontology1225893950.owl#Petrou][?y = http://www.owl-ontologies.com/Ontology1225893950.owl#Mathimatika][?x = http://www.owl-ontologies.com/Ontology1225893950.owl#Roumelis]}
{[?z = http://www.owl-ontologies.com/Ontology1225893950.owl#Georgiou][?y = http://www.owl-ontologies.com/Ontology1225893950.owl#Mathimatika][?x = http://www.owl-ontologies.com/Ontology1225893950.owl#Roumelis]}
{[?z = http://www.owl-ontologies.com/Ontology1225893950.owl#Georgiou][?y = http://www.owl-ontologies.com/Ontology1225893950.owl#Fysikh][?x = http://www.owl-ontologies.com/Ontology1225893950.owl#Athnasiou]}
```

### 13.3. Προσθήκη Νέων Συσχετίσεων

Στον παρακάτω κώδικα έχουμε προσθέσει προχωράμε ακόμη ένα βήμα πιο πέρα. Εκτός από την εισαγωγή νέας γνώσης, χρησιμοποιώντας ήδη υπάρχουσες και ορισμένες συσχετίσεις εντός της οντολογίας, μπορούμε να ορίσουμε καινούριες συσχετίσεις και να εισάγουμε επιπλέον στιγμιότυπα (individuals-instances). Αυτό ακριβώς υλοποιούμε στο αμέσως παρά κάτω κομμάτι κώδικα.

```
// ο John διδάσκει Gymnastikh και ο Nil την έχει επιλέξει ως
// μάθημα.
// Eisagwgh newn sysxetisewn kai emfanish palias kai neas me
// query poy exei kai logiko and

r.tell("fact a is didaskei_o(w:John,w:Gymnastikh);");
r.tell("fact b is exei_epilexthei_apo(w:Gymnastikh,w:Nil);");

String result3 = r.ask("query q is didaskei_o(?x,?y) and
exei_epilexthei_apo(?y,?z);");
String result2 = r.run();
```

Εκτελώντας αυτό το κομμάτι κώδικα παίρνουμε στη μεταβλητή result 3 το παρακάτω αποτέλεσμα :

```
{[?z = http://www.owl-ontologies.com/Ontology1225893950.owl#Nil][?y = http://www.owl-ontologies.com/Ontology1225893950.owl#Gymnastikh][?x = http://www.owl-ontologies.com/Ontology1225893950.owl#John]}
{[?z = http://www.owl-ontologies.com/Ontology1225893950.owl#Petrou][?y = http://www.owl-ontologies.com/Ontology1225893950.owl#Mathimatika][?x = http://www.owl-ontologies.com/Ontology1225893950.owl#Roumelis]}
```



```
{[?z = http://www.owl-ontologies.com/Ontology1225893950.owl#Georgiou][?y = http://www.owl-ontologies.com/Ontology1225893950.owl#Mathimatika][?x = http://www.owl-ontologies.com/Ontology1225893950.owl#Roumelis]}
```

```
{[?z = http://www.owl-ontologies.com/Ontology1225893950.owl#Georgiou][?y = http://www.owl-ontologies.com/Ontology1225893950.owl#Fysikh][?x = http://www.owl-ontologies.com/Ontology1225893950.owl#Athanasiou]}
```

Παρατηρούμε ότι η πρώτη γραμμή έχει ως αποτέλεσμα αυτήν ακριβώς την καινούρια γνώση. Ότι δηλαδή ο Neil ο οποίος έχει επιλέξει τη Gymnastikh ως μάθημα, η οποία διδάσκεται από τον john.

#### 13.4. Κατάργηση εισαχθείσης γνώσης – Ισχυρή άρνηση

Στις γλώσσες τύπου DL (Description Logic) όπως είναι η Buchingae δεν υπάρχει η έννοια της κατάργησης εισαχθείσης γνώσης. Η μηχανή Bossam όπως έχουμε αναφέρει ήδη, χειρίζεται οντολογίες (RDF(S), OWL) καθώς και κανόνες SWRL. Και στις δύο περιπτώσεις η γνώση αυτής της μορφής εντάσσεται στις περιγραφικές λογικές (Description Logics - DLs). Οι γλώσσες DLs με τη σειρά τους είναι υποσύνολα της λογικής πρώτης τάξης (First-Order Logic - FOL). Στην λογική αυτή (FOL) η άρνηση είναι ισχυρή (strong negation) και δεν υπάρχει η έννοια Negation-As-Failure (NAF) όπως για παράδειγμα υπάρχει στην Prolog. Οπότε αν δηλώσουμε κάτι με άρνηση σε DL (επομένως εν γένει και στη μηχανή bossam) δεν μπορούμε να ακυρώσουμε την ισχύ του. Αντίθετα επιβεβαιώνουμε την άρνησή του. Για παράδειγμα είναι διαφορετικό να πούμε ότι ισχύει η άρνηση της πρότασης «Ο Βασίλης είναι ψηλός» (strong negation) από το να συμπαιράνουμε ότι ισχύει η πρόταση «Ο Βασίλης δεν είναι ψηλός» επειδή δεν υπάρχει ως αλήθεια (fact) η πρόταση «Ο Βασίλης είναι ψηλός» (NAF).

Τώρα στη bossam, δεν είναι δυνατό να ανατρέξουμε κάποια δήλωση (αλήθεια). Δηλαδή να πούμε ότι μια νέα γνώση που εισήχθηκε κάποια στιγμή δεν ισχύει πλέον. Αυτό λοιπόν δεν είναι επιτρεπτό όταν χειριζόμαστε OWL/SWRL επειδή ακριβώς δεν επιτρέπεται σε DL (λόγω του Open World Assumption). Δηλαδή εάν προσθέσουμε την άρνηση της πρότασης (αυτό για παράδειγμα μπορεί να γίνει μέσω ενός τελεστή «not» της μηχανής bossam) δε θα καταφέρουμε να ακυρώσουμε τη γνώση αυτή. Αντίθετα θα προκαλέσουμε κάποια αντίφαση-ασυνέπεια στην οντολογία (ontology inconsistency). Αυτό θα συμβεί επειδή θα υπάρχει και η ευθεία πρόταση στην οντολογία αλλά και η άρνησή της.



Σε μια τέτοια περίπτωση, όπου πρέπει να αναιρέσουμε γνώση οντολογίας, ο μοναδικός τρόπος είναι με παρέμβαση στην ίδια την οντολογία, με κάποιον συντάκτη οντολογιών (ontology editor) όπως το protégé. Και φυσικά αυτό μπορεί να συμβεί μόνο εάν η γνώση είναι στατική. Σε περίπτωση γνώσης που εισάγεται δυναμικά μέσα από το πρόγραμμα, δεν υπάρχει κανένας τρόπος άρνησης της εισαχθείσας γνώσης. Στην εργασία μας η νέα γνώση προστίθεται με μια επιπλέον ενέργεια του χρήστη, όπως είναι η κατοχύρωση της γνώσης με ειδικό κουμπί όπως περιγράφουμε στο 18.

### 13.5. Συμπερασμός

Αφήσαμε για το τέλος τη συμπερασματολογία, η οποία είναι και η πεμπτουσία όλης αυτής της διαδικασίας. Πώς τελικά μπορούμε μέσα από διάφορες αλήθειες και προγραμματιστικές δομές να φτάσουμε σε τελικά συμπεράσματα ;

Στο παρακάτω η πληροφορία από τις αλήθειες εισάγεται στην οντολογία και ορίζεται και μια καινούρια σχέση από έναν κανόνα.

```
// Eisagwgh neas gnwshs se property pou den yparxei sthn ontologia
// kai emfanish mono twn properties pou exoun zitithei sta facts
// mporoyme dhl na kanoume query mono tis idiothtes didaskei &
// exei_epilexthei

r.tell("fact c is didaskei(w:Vranas,w:Istoria);");
r.tell("fact d is exei_epilexthei(w:Istoria,w:Kolokythas);");

r.tell("rule r3 is if didaskei(?x,?y) and
exei_epilexthei(?y,?z) then einai_mathitis(?x,?z);");

String result5 = r.run();
```

Στην οθόνη εκτέλεση μπορεί κανείς να παρατηρήσει ότι όλη η γνώση έχει φορτωθεί στην οντολογία και υπάρχει συμπερασμός. Αν δηλαδή βρεθεί ότι ο x διδάσκει το μάθημα y και το μάθημα y έχει επιλεχθεί από το μαθητή z, τότε ο μαθητής z einai\_mathitis του καθηγητή x :

Τα ζευγάρια που προκύπτουν παρακάτω είναι ζευγάρια μαθητών-καθηγητών :

```
fact http://bossam.com/default#Fact121 is http://www.owl-
ontologies.com/Ontology1225893950.owl#einai_mathitisk(http://www.owl-
ontologies.com/Ontology1225893950.owl#John,http://www.owl-
ontologies.com/Ontology1225893950.owl#Nil);
fact http://bossam.com/default#Fact122 is http://www.owl-
ontologies.com/Ontology1225893950.owl#einai_mathitisk(http://www.owl-
ontologies.com/Ontology1225893950.owl#Roumelis,http://www.owl-
ontologies.com/Ontology1225893950.owl#Petrou);
fact http://bossam.com/default#Fact123 is http://www.owl-
ontologies.com/Ontology1225893950.owl#einai_mathitisk(http://www.owl-
```



```
ontologies.com/Ontology1225893950.owl#Roumelis,http://www.owl-  
ontologies.com/Ontology1225893950.owl#Georgiou);  
fact http://bossam.com/default#Fact124 is http://www.owl-  
ontologies.com/Ontology1225893950.owl#einai_mathitisk(http://www.owl-  
ontologies.com/Ontology1225893950.owl#Athanasίου,http://www.owl-  
ontologies.com/Ontology1225893950.owl#Petrou);  
fact http://bossam.com/default#Fact125 is http://www.owl-  
ontologies.com/Ontology1225893950.owl#einai_mathitisk(http://www.owl-  
ontologies.com/Ontology1225893950.owl#Athanasίου,http://www.owl-  
ontologies.com/Ontology1225893950.owl#Georgiou);  
fact http://bossam.com/default#Fact126 is http://www.owl-  
ontologies.com/Ontology1225893950.owl#einai_mathitisk(http://www.owl-  
ontologies.com/Ontology1225893950.owl#Vranas,http://www.owl-  
ontologies.com/Ontology1225893950.owl#Kolokythas);
```

### 13.6. Ερωτήματα σε δύο οντολογίες και κανόνες με συνθήκες

Στα παρακάτω προχωρούμε ένα βήμα πιο πέρα, και ρωτούμε δύο οντολογίες για να πάρουμε ένα αποτέλεσμα. Το πλήρες πρόγραμμα βρίσκεται στο ΠΑΡΑΡΤΗΜΑ Α – Κώδικας 3. Επίσης στο ίδιο σημείο, θα βρούμε και το αποτέλεσμα της εκτέλεσης του κώδικα. Αφού ορίσουμε τους reasoners, (r και ir) κατά τα γνωστά, ορίζουμε τη νέα γνώση σε κάθε οντολογία :

```
ir.tell("fact ic .....");  
r.tell("fact ic2 .....");
```

Στη συνέχεια θέτουμε ερωτήματα σε κάθε οντολογία :

```
String result4 = r.ask("query q is didaskei(?x,?y) and  
exei_epilexthei(?y,?z);");  
String result6 = ir.ask("query q is has(?x,?z) and  
(?x=Noisy);");
```

Και εκτελούμε τα ερωτήματα :

```
String resulti5 = ir.run();  
String resulti7 = r.run();
```

Στη συνέχεια διαχωρίζουμε το αποτέλεσμα που υπάρχει σε κάθε μεταβλητή από τη URI του. Η διαδικασία αυτή, μπορεί να παραλληλισθεί «με την αποκόλληση ενός μυδιού από το όστρακό του». Σε όλα τα αποτελέσματα ως τώρα μπορούμε να παρατηρήσουμε ότι ας πούμε το αποτέλεσμα είναι της μορφής

```
?z = http://www.owl-ontologies.com/Ontology1225893950.owl#Petrou
```



Το πραγματικό όμως περιεχόμενο της z είναι ο μαθητής Petrou. Το υπόλοιπο είναι εμφανώς η διεύθυνση της ονοματολογίας της οντολογίας μας. Όταν θέλουμε να προχωρήσουμε σε έλεγχο δεδομένων, θα πρέπει να έχουμε αποκολλήσει το πραγματικό περιεχόμενο από την URI του.

Ο παρακάτω κώδικας κάνει ακριβώς αυτή τη δουλειά. Το δε αποτέλεσμα τελικά και μετά την εκτέλεση, βρίσκεται στη μεταβλητή service

```
ArrayList actions = new ArrayList(result9.getBindings());  
  
for (Iterator it = actions.iterator(); it.hasNext();) {  
    HashMap hm = (HashMap) it.next();  
    String segment = hm.get("?x").toString();  
    segment = segment.substring(segment.indexOf("#") + 1);  
    String service = hm.get("?z").toString();  
    service = service.substring(service.indexOf("#") + 1);  
    System.out.println("\n Mathima: " + segment);  
    System.out.println("\n Mathitis: " + service);  
}
```

Τώρα πλέον μπορούμε να συνδυάσουμε τα αποτελέσματα που πήραμε από τα ερωτήματα στις οντολογίες με απλές προγραμματιστικές δομές ελέγχου ώστε να καταλήξουμε σε συμπεράσματα. Τέτοιες διαδικασίες περιγράφονται στην ανάλυση του συστήματος ELSS που αναπτύξαμε στα πλαίσια της διπλωματικής εργασίας, και το οποίο περιγράφεται στο κεφάλαιο 15.

## 14. Γλώσσα Προγραμματισμού – Περιβάλλον προγραμματισμού

Η γλώσσα προγραμματισμού επιλέχθηκε να είναι η Java η οποία είναι σύγχρονη, αντικειμενοστραφής, ελεύθερη στη χρήση, με δυνατότητα δημιουργίας μιας όμορφης Γραφικής Διεπαφής Χρήστη (GUI – Graphical User Interface). Επίσης οι κλάσεις του OWL protégé είναι γραμμένες σε Java, και αυτός είναι ακόμη ένας λόγος χρήσης της συγκεκριμένης γλώσσας. Το περιβάλλον στο οποίο δουλέψαμε ήταν ανοικτού λογισμικού και ελεύθερου στη χρήση. Πιο συγκεκριμένα δουλέψαμε επάνω στις πλατφόρμες ανάπτυξης Eclipse και Netbeans. Στο παράρτημα Β πριν τον κώδικα δίνουμε κάποια στοιχεία για τη χρήση αυτών των πλατφορμών ανάπτυξης προγραμμάτων χωρίς να εισερχόμαστε σε λεπτομέρειες, κάτι το οποίο άλλωστε δεν αφορά την παρούσα εργασία.



## 15. ELSS (E-Learning Smart System)

Στις παρακάτω ενότητες, παρουσιάζουμε την εφαρμογή που αναπτύξαμε στα πλαίσια της διπλωματικής εργασίας και στην οποία υλοποιούνται και χρησιμοποιούνται όλα αυτά που έχουμε περιγράψει στα προηγούμενα κεφάλαια. Έχουμε ονομάσει την εφαρμογή μας ELSS (E-Learning Smart System). Πράγματι το σύστημα που αναπτύξαμε παρουσιάζει μια έξυπνη προσαρμοστικότητα κατά τη διάρκεια της εκπαιδευτικής διαδικασίας σε σχέση με τις συνθήκες πλαισίου.

### 15.1. Παραμετροποιησιμότητα του συστήματος

Πρέπει να τονίσουμε κάτι πολύ σημαντικό το οποίο υπάρχει στο σχεδιασμό και την υλοποίηση τόσο του συστήματος όσο και της τελικής εφαρμογής. Το γεγονός ότι είναι παραμετροποιήσιμα. Δηλαδή και οι οντολογίες μπορούν να εμπλουτισθούν, αλλά και να αλλάξουν αφού η χρήση που τους κάνουμε έχουν και τους δύο προσανατολισμούς. Οι οντολογίες είναι ανοικτές και διαχειρίσιμες από οποιοδήποτε πρόγραμμα διαχείρισης οντολογιών (όπως είδαμε ήδη αρκετές φορές εμείς χρησιμοποιούμε το protégé). Έτσι μπορούμε να προσθέσουμε νέα στατική γνώση (είτε κλάσεις, είτε συσχετίσεις, είτε στιγμιότυπα). Μπορούμε επίσης να προσθέσουμε ή να αφαιρέσουμε ή να τροποποιήσουμε αλήθειες στην ειδική ρουτίνα-μηχανισμό αληθειών. Μπορούμε να αλλάξουμε τη λογική των εκπαιδευτικών σεναρίων, όπως εμείς θέλουμε με πολύ απλές κινήσεις, για παράδειγμα με ένα απλό editing. Εμείς λόγω χάρη στην εφαρμογή υλοποιούμε το επίπεδο δυσκολίας της εξέτασης με δύο στάθμες. Εύκολη εξέταση και δύσκολη εξέταση. Θα μπορούσε η εύκολη εξέταση να γίνει διακοπή εξέτασης και το εκπαιδευτικό σενάριο να αλλάξει εντελώς προσανατολισμούς. Δηλαδή να υπάρχουν στιγμές οπότε θα υπάρχει εξέταση και άλλες στιγμές όπου δεν θα υπάρχει καθόλου εξέταση. Και αυτό μπορεί να γίνει με ένα απλό editing στην οντολογία ή στο μηχανισμό αληθειών.

Επίσης ο μηχανισμός συμπερασμού είναι και αυτός πολύ ευέλικτος, αφού στηρίζεται σε γνωστές προγραμματιστικές δομές.

Επιπλέον μέσα από τα εκπαιδευτικά σενάρια, έχουμε υλοποιήσει διαδικασίες δυναμικής διαχείρισης των οντολογιών και κανόνων, ενεργοποιώντας και απενεργοποιώντας αλήθειες που προσδίδουν διαδραστικό χαρακτήρα στο σύστημά μας.

Η εφαρμογή μας δεν φιλοδοξεί να δημιουργήσει μηχανισμούς που να καλύπτουν κάποιο συγκεκριμένο επιστημονικό πεδίο. Αυτό που πιστεύουμε ότι δημιουργήσαμε είναι μια πλατφόρμα η οποία καταρχήν λειτουργεί άψογα, συνενώνει πάρα πολλές νέες τεχνολογίες με





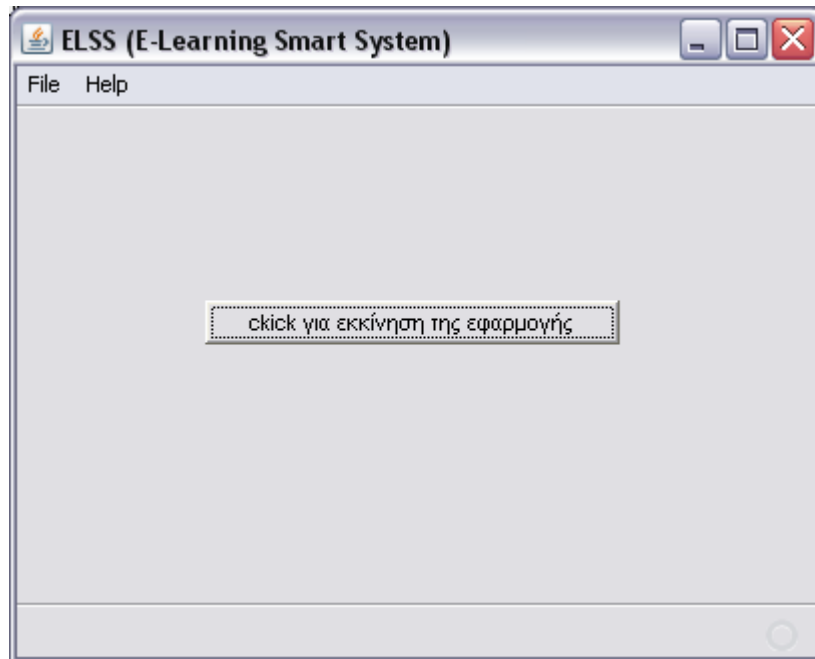
κατεύθυνση σημασιολογικού ιστού και τέλος έχει πολύ μεγάλες δυνατότητες παραμετροποίησης. Οι προσθήκες και οι μετατροπές μπορούν αν δημιουργήσουν πλήθος διαφορετικών καταστάσεων και σεναρίων, που μπορούν να χρησιμοποιηθούν τόσο στον εκπαιδευτικό χώρο, τόσο και σε άλλα πεδία.

Στη συνέχεια περιγράφουμε το περιβάλλον προγραμματισμού που χρησιμοποιήσαμε, τα τμήματα της εφαρμογής, και τέλος παρουσιάζουμε τα πραγματικά εκπαιδευτικά σενάρια που υλοποιήσαμε.

## 16. Οι Οθόνες της εφαρμογής

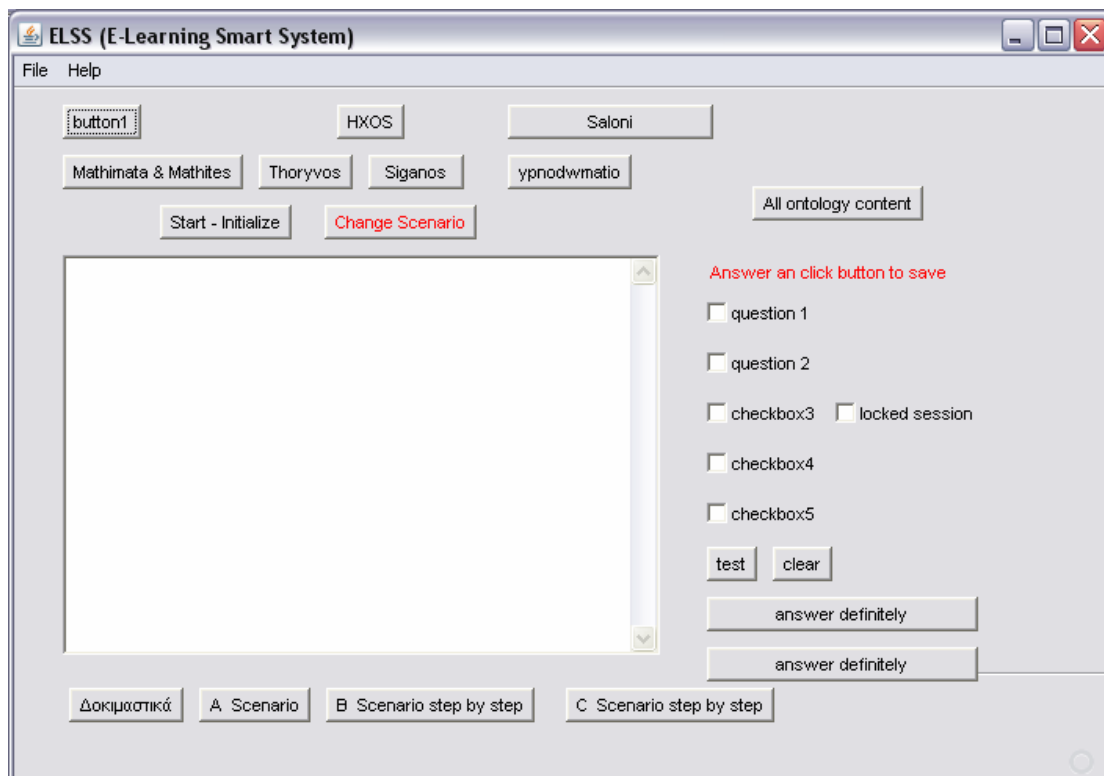
Η εφαρμογή μας για τους λόγους που εξηγήσαμε πιο πάνω, αναπτύχθηκε σε Java. Το γεγονός αυτό την κάνει ισχυρή, αφού μπορεί να εκτελεστεί σε περιβάλλον συσκευής palmtop ή κάποιας άλλης φορητής συσκευής (mobile device). Μπορούμε να φέρουμε στο μυαλό μας τα σύγχρονα κινητά τηλέφωνα των οποίων διάφορες εφαρμογές (παιχνίδια, calculators, organizers κλπ) έχουν αναπτυχθεί σε Java. Γίνεται λοιπόν κατανοητό, ότι η εφαρμογή μας είναι έτοιμη να εκτελεστεί σε μικρές κινητές συσκευές. Οποσδήποτε, κάποιες προσαρμογές θα πρέπει να γίνουν στα paths των βιβλιοθηκών και των αρχείων πλαισίου (context). Άρα η μεταφορά (porting) της εφαρμογής σε κάποια τέτοια συσκευή είναι εφικτή με ελάχιστες παρεμβάσεις. Η αρχική οθόνη της εφαρμογής διαθέτει αρκετά κουμπάκια, ετικέτες, και ένα μεγάλο text box το οποίο παίζει το ρόλο της οθόνης της κινητής συσκευής. Έτσι οτιδήποτε εξάγουμε ως αποτέλεσμα σε αυτό το text box ο αναγνώστης-χρήστης θα πρέπει να το εκλάβει ως έξοδο στην standard output της κινητής συσκευής.

Το πρώτο παράθυρο της εφαρμογής μας εμφανίζεται με την εκτέλεση του rtychiakh\_argikh.jar. Η πρώτη οθόνη της εφαρμογής μας φαίνεται στην *Εικόνα 18*. Από εδώ μπορούμε να καλέσουμε το κυρίως παράθυρο της εφαρμογής μας, το οποίο φαίνεται στην *Εικόνα 19*. Επίσης έπειτα από την ολοκλήρωση κάθε εκπαιδευτικού σεναρίου, η εφαρμογή επιστρέφει στο παράθυρο της *Εικόνα 18* επιλέγοντας το κουμπάκι «Change Scenario».



Εικόνα 18 : Το πρώτο παράθυρο της εφαρμογής

Όταν επιλέξουμε το κουμπάκι «click για εκκίνηση της εφαρμογής» μεταφερόμαστε στο



Εικόνα 19 : Η Οθόνη της εφαρμογής από το οποίο μπορούμε να εκτελέσουμε σενάρια



παράθυρο της *Εικόνα 19*. Από το σημείο αυτό, μπορούμε να επιλέξουμε το εκπαιδευτικό σενάριο που επιθυμούμε. Τα εκπαιδευτικά σενάρια αναλύονται στις αμέσως επόμενες παραγράφους [18. Εκπαιδευτικά Σενάρια].

Τα τμήματα της οθόνης επιλογής σεναρίου είναι τα εξής :

Στο κέντρο του παραθύρου της εφαρμογής βλέπουμε ένα μεγάλο text box, το οποίο παίζει το ρόλο της οθόνης της συσκευής στην οποία θα λειτουργήσει η εφαρμογή μας. Πρόκειται για μια προσομοίωση οθόνης συσκευής με εντολές εξόδου σε αυτήν, πολύ εύκολα μετατρέψιμες σε εντολές εξόδου σε πραγματική οθόνη (standard output) μιας πραγματικής φορητής ή και σταθερής συσκευής. Στην οθόνη αυτή μπορούμε να δούμε όλες τις εκτελέσεις και τα αποτελέσματα που προκύπτουν από τα κουμπάκια της εφαρμογής. Πρέπει να διευκρινίσουμε ότι έχει υλοποιηθεί κατά κάποιον τρόπο και διάδραση με την οθόνη, αφού τα εκπαιδευτικά σενάρια Β και Γ που έχουν αναπτυχθεί, υποθέτουμε ότι περιέχουν και διαδικασίες που χρειάζονται την ενέργεια του χρήστη.

Τα κουμπάκια που χρειάζονται για την εκτέλεση των σεναρίων είναι :

- Το κουμπί «start-initialize» πάνω από την οθόνη το οποίο πρέπει να πατηθεί πρώτα από όλα, ώστε να υπάρξει αρχικοποίηση της μηχανής κανόνων, αληθειών και συμπερασμού.
- Τα τρία κουμπιά των σεναρίων (Α, Β και C), κάτω από την οθόνη, με τα οποία επιλέγουμε το σενάριο που θα εκτελέσουμε.
- Το κουμπί «Change Scenario» με το οποίο μπορούμε να σταματήσουμε την εκτέλεση του σεναρίου. Όταν το πατήσουμε, κλείνει το τρέχον παράθυρο, επιστρέφουμε στην *Εικόνα 18* και ξεκινούμε από την αρχή.
- Το κουμπί «answer definetly» με το οποίο οριστικοποιούμε τις απαντήσεις του χρήστη στα διαδραστικά εκπαιδευτικά σενάρια Β και C.

Πρέπει να τονίσουμε ότι όταν πατηθεί το κουμπί «start-initialize», τότε γίνονται αόρατα τα κουμπιά που δε χρειάζονται, ενώ και με την επιλογή του σεναρίου γίνονται ανενεργά και τα υπόλοιπα κουμπιά που δε μας χρειάζονται πλέον.

Όλα τα υπόλοιπα κουμπιά της εφαρμογής είναι βοηθητικά και χρησιμεύουν μόνο στην περίπτωση που κάποιος θέλει να δει αποτελέσματα εκτέλεσης απλών ερωτημάτων εκτός της διαδικασίας των σεναρίων και στην ουσία εκτός της εφαρμογής μας. Για παράδειγμα με το κουμπί «Mathimata & Mathites», υποβάλλουμε στην οντολογία ένα ερώτημα το οποίο δείχνει στην υποτιθέμενη οθόνη της εφαρμογής το σύνολο των Μαθημάτων που υπάρχουν στην οντολογία ή που έχουν προστεθεί με κανόνες και αλήθειες και των Μαθητών που τα



παρακολουθούν. Η συγκεκριμένη λειτουργία λοιπόν είναι στατική και χρησιμοποιείται για μικρά εκπαιδευτικά παραδείγματα ή για λόγους ελέγχου της εφαρμογής.

Στο δεξί μέρος της οθόνης διακρίνουμε το διαδραστικό μέρος της εφαρμογής. Είπαμε ήδη ότι διαδραστικά είναι τα σενάρια Β και C. Στο σημείο αυτό ο εξεταζόμενος παρεμβαίνει στη διαδικασία και απαντά σε ερωτήσεις. Πρέπει να αναφέρουμε σε αυτό το σημείο, ότι οι ερωτήσεις δεν είναι στατικές. Διαφοροποιούνται ανάλογα με το επίπεδο δυσκολίας της εξέτασης και παρέχουν δυναμική χροιά στο σύστημα.

Answer an click button to save

question 1

question 2

checkbox3

checkbox4

checkbox5

test clear

answer & click before next step

**Εικόνα 20 :** Οι υποτιθέμενες ερωτήσεις των σεναρίων

Στο σημείο αυτό να προσθέσουμε και μια ακόμα πληροφορία για το κουμπάκι «All Ontology Content». Το κουμπάκι αυτό προστέθηκε έτσι ώστε να μπορεί κάποιος να έχει πρόσβαση στην πληροφορία της οντολογίας ανά πάσα στιγμή, και μάλιστα έτσι όπως διαμορφώνεται σε κάθε δευτερόλεπτο και σε κάθε φάση της εκτέλεσης. Όμως το πάτημα αυτού του κουμπιού, προφανώς παράγει έναν μεγάλο όγκο πληροφορίας στην οθόνη της συσκευής, και πρέπει να το πάρουμε υπόψη μας σε περίπτωση που το επιλέξουμε.

## **17. Η αρχιτεκτονική του συστήματος και τα τμήματα της εφαρμογής**

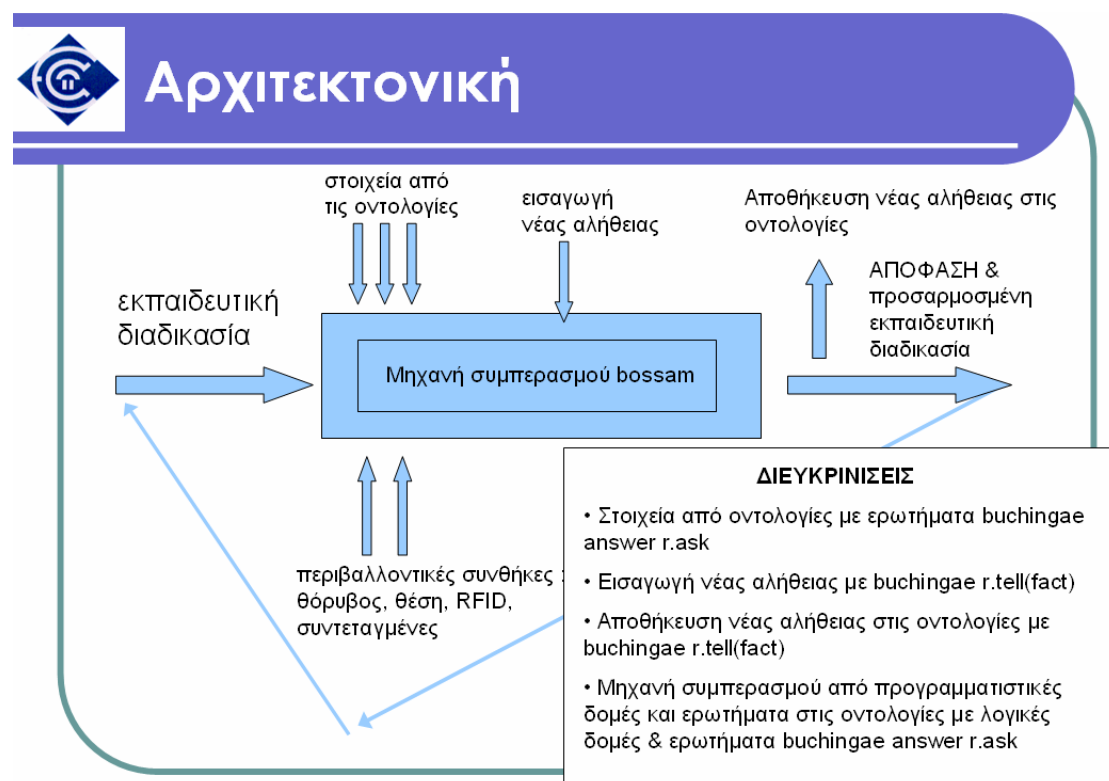
Το σύστημά μας υλοποιεί εκπαιδευτικά σενάρια με κοινό παρονομαστή την αξιολόγηση ενός (ή και περισσότερων) μαθητή. Αποτελείται από πέντε κύρια τμήματα, τα οποία αναλύονται στη συνέχεια, και συνενώνει πολλές διαφορετικές τεχνολογίες. Το τμήμα οντοτήτων, το τμήμα κανόνων και αληθειών, το τμήμα το οποίο συλλέγει τις πληροφορίες από το



περιβάλλον, το τμήμα συμπερασμού και το τμήμα αποτελεσμάτων. Ως τεχνολογίες, όλα αυτά έχουν περιγραφεί στα προηγούμενα κεφάλαια. Όμως παραμένει πάντα ανοικτό το επίπεδο της έρευνας και της περαιτέρω ανάπτυξης των, αφού μπορεί να εμπλουτισθεί με πολλά ακόμα στοιχεία σε ότι αφορά τις οντολογίες, τους κανόνες, και τις πληροφορίες από το περιβάλλον.

### 17.1. Η αρχιτεκτονική του συστήματος

Η αρχιτεκτονική του συστήματος που αναπτύσσουμε στην παρούσα εργασία φαίνεται στην *Εικόνα 21*. Υπάρχει μια εκπαιδευτική διαδικασία η οποία εξελίσσεται χρονικά. Η διαδικασία αυτή έχει καταρχήν πηγές γνώσης αφενός τις οντολογίες αλλά και αφετέρου άλλη εξωτερική γνώση η οποία εισέρχεται στο σύστημα υπό μορφή ενός αλήθειας (fact).



*Εικόνα 21* : Η αρχιτεκτονική του συστήματος

Η διαδικασία επηρεάζεται από εξωτερικούς παράγοντες όπως είναι οι θόρυβοι από το περιβάλλον αλλά και η θέση αυτού που συμμετέχει σε αυτήν. Όλα αυτά εισέρχονται στη μηχανή bossam όπου υφίστανται επεξεργασία και παράγεται συμπερασμός. Ο συμπερασμός



αυτός αφενός αποθηκεύεται ως νέα γνώση στο σύστημα, αλλά και αφετέρου αποτελεί είσοδο για τον επόμενο κύκλο λειτουργίας του συστήματος..

Στα επόμενα κεφάλαια θα δούμε τα διάφορα τμήματα της εφαρμογής που υλοποιεί το παραπάνω σύστημα.

### **17.2. Τμήμα οντοτήτων**

Οι οντότητες που χρησιμοποιούμε στο σύστημά μας είναι δύο όπως ήδη έχουμε αναφέρει και αναλύσει εκτενώς στα 8.1 και 8.2. Η γνώση που υπάρχει σε μια οντολογία είναι γενικά στατική όταν μιλάμε για σύστημα το οποίο λειτουργεί με επίγνωση πλαισίου. Μέσα από την εφαρμογή ο χρήστης μπορεί να διαπιστώσει ότι αν και είναι δυνατόν να εισάγουμε επιπρόσθετη στατική γνώση στο σύστημα, το σπουδαιότερο είναι ότι η γνώση είναι διαχειρίσιμη μέσα από το πρόγραμμα. Η γνώση αυτή είναι δυναμική και μπορεί να εισαχθεί ως καινούρια γνώση με κανόνες και αλήθειες.

Επιπλέον μέσα από τον κώδικα μπορεί ο αναγνώστης να βρει τις συναρτήσεις Java που μπορεί κανείς να χρησιμοποιήσει ώστε να κάνει χρήσιμη την αποθηκευμένη στις οντολογίες γνώση. Με κανέναν τρόπο δεν προσπαθούμε να δημιουργήσουμε οντολογίες που να καλύπτουν σφαιρικά και ολοκληρωμένα είτε την εκπαιδευτική διαδικασία, είτε το περιβάλλον λειτουργίας και κίνησης ενός ανθρώπου. Δημιουργήσαμε δύο πολύ απλές οντολογίες για να χρησιμοποιήσουμε την αποθηκευμένη γνώση, αλλά και για να δείξουμε πώς μπορούμε να τη διαχειριστούμε, να την εμπλουτίσουμε και να τη χρησιμοποιήσουμε. Από κει και πέρα, η έρευνα μπορεί να επεκταθεί και προς αυτήν την κατεύθυνση.

### **17.3. Τμήμα κανόνων και αληθειών**

Οι κανόνες και οι αλήθειες σε μια οντολογία όπως είδαμε μπορεί να είναι καταχωρημένες στην οντολογία με τη μορφή σχέσεων των individuals, αλλά μπορεί να προστεθούν δυναμικά και στη συνέχεια. Στη δική μας εργασία χρησιμοποιήσαμε ένα συνδυασμό αληθειών, ερωτημάτων και δομών επιλογής που μας επέτρεψε να δημιουργήσουμε ένα σύνολο ανεξάρτητων μηχανών. Μια από αυτές τις μηχανές είναι η μηχανή δημιουργίας κανόνων και αληθειών. Η μηχανή αποτελείται από κάποιες γραμμές κώδικα μέσα σε κάθε κουμπάκι, αλλά και με μια ανεξάρτητη ρουτίνα, την facts (παράρτημα Β). Γενικότερα θα μπορούσαν όλες οι αλήθειες να ορισθούν μέσα σε αυτήν τη ρουτίνα. Η τοποθέτησή τους και σε κάποια άλλα σημεία, έγινε για εκπαιδευτικούς λόγους.

Είναι εντελώς διαφορετικό πράγμα να βλέπεις όλες τις αλήθειες μαζεμένες σε ένα σημείο και να μην ξέρεις ποια χρησιμοποιείται και πού, και άλλο πράγμα να βλέπεις ορισμούς αληθειών,



στο σημείο που χρειάζονται. Εμείς χρησιμοποιήσαμε και τις δύο διαδικασίες με καθαρά εκπαιδευτικούς σκοπούς.

Επιπλέον μπορούμε να πούμε ότι θα μπορούσε μια τέτοια μηχανή αληθειών να είναι συνδεδεμένη με μια ευρύτερη οντολογία η οποία να λειτουργεί ανεξάρτητα σε κάποιον μεμονωμένο εξυπηρετή και από τον οποίο με κατάλληλα ερωτήματα θα μπορούσε να εξαχθεί η κατάλληλη ρουτίνα αληθειών της εφαρμογής μας.

#### **17.4. Τμήμα εισαγωγής πληροφορίας από το περιβάλλον**

Στο σύστημά μας θεωρούμε ενδεικτικά δύο τρόπου εισαγωγής πληροφορίας από το περιβάλλον. Με αισθητήρες ήχου, για συλλογή των ήχων του περιβάλλοντος, και με απλό σύστημα GPS για εντοπισμό της γεωγραφικής θέσης του εξεταζόμενου. Η πληροφορία εισάγεται σε μορφή αρχείου κειμένου (text file). Το αρχείο που χρησιμοποιήσαμε για να προσομοιώσουμε τη συλλογή πληροφορίας από τους αισθητήρες, μας το παρείχε η εταιρεία NOKIA, και είναι ένα πραγματικό αρχείο με στοιχεία που συλλέγονται από πραγματικές κινητές συσκευές της. Επίσης το αρχείο που χρησιμοποιήσαμε για την προσομοίωση του πειράματος με το GPS το δημιουργήσαμε μόνοι μας. Αυτό το δεύτερο αρχείο (RFIDcontextdata.txt ) περιγράφεται επακριβώς στο 17.4.2.

Το πρώτο αρχείο (contextdata.txt - από NOKIA) ονομάζεται από την ίδια την εταιρεία : «Sensor Signal Data Set for Exploring Context Recognition of Mobile Devices». Δηλαδή «Σύνολο Δεδομένων Σημάτων Αισθητήρων για Εξερεύνηση Αναγνώρισης Περιβάλλοντος Κινητών Συσκευών». Η εταιρεία παρέχει τα αρχείο αυτό κάτω από την ακόλουθη άδεια :

Terms of usage

The data set and the associated images can be freely used for research under the following conditions:

1. You may not redistribute the data nor the associated images; illustrating part of the data and images in tables or diagrams when reporting the results is, of course, allowed.
2. When reporting any results, the following article has to be cited:

Jani Mentyjervi, Johan Himberg, Petri Kangas, Urpo Tuomela, and Pertti



Huuskonen (2004). Sensor Signal Data Set for Exploring Context Recognition of Mobile Devices. In Proceedings of 2nd Int. Conf. on Pervasive Computing (PERVASIVE 2004), April 18-23, Linz/Vienna, Austria. LNCS 3001. Springer.

3. Please send a reference to the article where you have used the data.

Το αρχείο δεδομένων αποτελείται από ένα σύνολο αριθμών τα οποία είναι ορισμένα σε στήλες. Κάθε γραμμή αποτελείται από ένα σύνολο δεδομένων τα οποία έχουν συλλεγεί από αισθητήρες. Τα δεδομένα είναι σε ASCII μορφή και βρίσκονται στο αρχείο «contextdata.txt» Οι 32 συνολικά στήλες αντιστοιχούν στα εξής δεδομένα :

Scenario number

Repetition number

Time (s)

Device:Position:DisplayDown

Device:Position:DisplayUp

Device:Position:AntennaDown

Device:Position:AntennaUp

Device:Position:SidewaysRight

Device:Position:SidewaysLeft

Device:Stability:Stable

Device:Stability:Unstable

Device:Placement:AtHand

Environment:Light:EU

Environment:Light:USA

Environment:Light:Bright

Environment:Light:Normal

Environment:Light:Dark

Environment:Light:Natural

Environment:Light:TotalDarkness

Environment:Temperature:Hot

Environment:Temperature:Warm





Environment:Temperature:Cool  
Environment:Temperature:Cold  
Environment:Humidity:Humid  
Environment:Humidity:Normal  
Environment:Humidity:Dry  
Environment:SoundPressure:Silent  
Environment:SoundPressure:Modest  
Environment:SoundPressure:Loud  
UserAction:Movement:Walking  
UserAction:Movement:WalkingFast  
UserAction:Movement:Running

Όπως είναι σαφές από μια απλή ανάγνωση, υπάρχουν δεδομένα που συλλέγονται από αισθητήρες κλίσης (ως προς το οριζόντιο) της συσκευής, αισθητήρες κατάστασης της κεραίας, αισθητήρες φωτεινότητας του περιβάλλοντος, αισθητήρες θερμοκρασίας και υγρασίας του περιβάλλοντος. Υπάρχουν επίσης αισθητήρες για την ένταση του ήχου, και τέλος αισθητήρες ταχύτητας κίνησης.

Κάθε γραμμή όπως είπαμε είναι μια ομάδα δεδομένων που λαμβάνονται όλα μαζί κάθε δευτερόλεπτο. Σε όλο το αρχείο υπάρχουν 46045 γραμμές, άρα 46045 δευτερόλεπτα. Αυτά τα δευτερόλεπτα, χωρίζονται σε 5 διαφορετικά σενάρια. Οι τρεις πρώτες στήλες του αρχείου περιλαμβάνουν α) τον αριθμό του σεναρίου, β) τον αριθμό της επανάληψης του ίδιου σεναρίου και γ) το χρόνο σε δευτερόλεπτα από το ξεκίνημα της κάθε επανάληψης. Για κάποιο λόγο άγνωστο προς εμάς, λείπει κάποιος αριθμός επαναλήψεων. Για παράδειγμα, λείπει ο αριθμός των επαναλήψεων 1 και 2, του σεναρίου 1. Αυτό όμως δεν έχει κάποια ιδιαίτερη επίπτωση στα πειράματά μας.

Τα σενάρια περιέχουν 40-50 επαναλήψεις. Κάθε σενάριο υποτίθεται ότι είναι κάτι το οποίο κάνει αυτός που κουβαλάει τη φορητή συσκευή. Για παράδειγμα ο φέρων τη συσκευή περπατάει μέσα σε κλειστό χώρο ή έξω σε ανοικτό, χρησιμοποιεί ασανσέρ ή σκάλες, πηγαίνει σε κάποιο γραφείο ή σε κάποιο café bar.

Τα σενάρια αυτά μας βοήθησαν να υλοποιήσουμε τις δικές μας ιστορίες, οι οποίες αντιστοιχούν περίπου σε τέτοιες κινήσεις υποτιθέμενων εξεταζόμενων, και τα δεδομένα που χρησιμοποιήσαμε ήταν αυτά του ήχου και αυτά της κίνησης. Πολλά από τα δεδομένα, όπως της φωτεινότητας και της θερμοκρασίας δε χρησιμοποιήθηκαν, αλλά σε κάποιες μελλοντικές



εκδόσεις της εφαρμογής θα μπορούσαν να ενσωματωθούν ως πρόσθετη αξιοποιήσιμη πληροφορία περιβάλλοντος.

#### **17.4.1. Τμήμα εισαγωγής πληροφορίας από αισθητήρες**

Στο σημείο αυτό θα εισάγουμε πληροφορίες στο σύστημά μας από αισθητήρες. Οι αισθητήρες που θεωρούμε στο παρόν σύστημα είναι αισθητήρες ήχου από το περιβάλλον. Θεωρούμε την ύπαρξη μιας συσκευής η οποία μπορεί και καταγράφει τους ήχους από το περιβάλλον και τις αποθηκεύει σε ένα αρχείο σε μορφή ASCII. Μπορούμε να θεωρήσουμε ότι πρόκειται για κάποια φορητή συσκευή NOKIA, από την οποία έχουμε προμηθευτεί και το αρχείο με τα πραγματικά δεδομένα. Τα δεδομένα αυτά βρίσκονται στο αρχείο που μόλις περιγράψαμε και τα δεδομένα που μας ενδιαφέρουν βρίσκονται στις στήλες 26, 27 και 28.

#### **17.4.2. Τμήμα εισαγωγής πληροφορίας από GPS**

Στο σημείο αυτό θα εισάγουμε πληροφορίες στο σύστημά μας από κάποιο απλό σύστημα GPS. Το GPS είναι ένα σύστημα εντοπισμού θέσης και υπό αυτήν την έννοια τα στοιχεία που συλλέγουμε είναι είτε στοιχεία που αφορούν συντεταγμένες θέσης, είτε στοιχεία που αφορούν τοποθέτηση στο χώρο. Για το σκοπό αυτό έχουμε στη διάθεσή μας ένα ακόμα υποτιθέμενο αρχείο το οποίο θα δημιουργηθεί από ένα τέτοιο κοινό σύστημα GPS. Ένα από τα στοιχεία που μας παρέχει ένα τέτοιο σύστημα δεδομένων από αισθητήρες είναι το νούμερο του δωματίου στο οποίο βρίσκεται ο υπό εξέταση μαθητής. Το σύστημα περιγράφεται αναλυτικά στο [18], ονομάζεται RFID Technology και προϋποθέτει ότι υπάρχει μια προχαρτογράφηση του χώρου στον οποίο διεξάγεται το πείραμα. Κάθε δωμάτιο αριθμείται με κάποιον ακέραιο, και το σύστημα εντοπισμού, παράγει κάθε φορά έναν αριθμό για να μας δείξει ότι το άτομο βρίσκεται στον συγκεκριμένο χώρο. Πάντως η χρήση μιας τέτοιας τεχνολογίας δεν είναι δεσμευτική για την εφαρμογή μας, αφού τα δεδομένα σε τέτοια αρχεία με δεδομένα αισθητήρων είναι σε ASCII μορφή. Έτσι αντί να διαβάζει κάποιος ένα νούμερο που αντιστοιχεί σε ένα δωμάτιο, μπορεί να διαβάζει δύο νούμερα, τα οποία αντιστοιχούν σε γεωγραφικές συντεταγμένες. Επίσης, ένας χώρος μπορεί να χαρτογραφηθεί και με τρεις συντεταγμένες, έχοντας σημείο (0, 0, 0) κάποια γωνία του κτηρίου. Σε αυτό το σύστημα κατακόρυφα συνήθως έχουμε μοναδιαία μεταβολή (ανάλογα τον όροφο) και στις άλλες διαστάσεις έχουμε αποστάσεις σε μέτρα.

Με όποιον τρόπο και να γίνει η χαρτογράφηση και η αναφορά στην τοποθεσία, αυτό δεν μπορεί να αποτελέσει κάποια δέσμευση για την εφαρμογή γιατί όλα τα δεδομένα του αρχείου RFIDcontextdata.txt θα είναι σε ASCII μορφή.



Στο Παράρτημα Β στο οποίο μπορούμε να δούμε τον πηγαίο κώδικα της εφαρμογής, τα δεδομένα διαβάζονται και τοποθετούνται στον πίνακα `arrayOfIntegers` (ρουτίνα `button15ActionPerformed`), άρα είναι άμεσα επεξεργάσιμα. Εμείς προς χάρη της απλότητας, θα θεωρήσουμε ότι τα δεδομένα του αρχείου `RFIDcontextdata.txt` είναι ήδη επεξεργασμένα από το σύστημα RFID [18], και έτσι το περιεχόμενο του αρχείου είναι μόνο αριθμοί οι οποίοι αντιστοιχούν σε αριθμούς δωματίων. Πιο ειδικά και για τη λειτουργία του εκπαιδευτικού σεναρίου Γ, θα θεωρήσουμε μόνο δύο αριθμούς, το 1 που θα αντιστοιχεί σε κάποιο ήσυχο δωμάτιο (πχ το δωμάτιο του φοιτητή στην εστία) και το 2 σε κάποιο πιο θορυβώδες δωμάτιο (πχ την καφετερία της εστίας ή τους διαδρόμους της εστίας). Επίσης στο `RFIDcontextdata.txt` υπάρχει ακόμη μια στήλη η οποία είναι τα δευτερόλεπτα.

### 17.5. Τμήμα συμπερασμού

Το τμήμα συμπερασμού είναι ένα πολύ σπουδαίο κομμάτι της εφαρμογής μας. Λειτουργεί σε πολλά σημεία της εφαρμογής μας και διαφέρει ανάλογα το εκπαιδευτικό σενάριο. Για παράδειγμα στα σενάρια Α και Β λειτουργεί μια ρουτίνα, η «`private void symperasmos_dwmatiou`», (Παράρτημα Β) η οποία παράγει συμπέρασμα για το δωμάτιο στο οποίο βρίσκεται ο μαθητής. Εξορισμού εδώ θεωρήσαμε ότι αν ανιχνεύεται θόρυβος τότε ο μαθητής βρίσκεται στο σαλόνι, ενώ όταν υπάρχει ησυχία ο μαθητής βρίσκεται στο υπνοδωμάτιο. Όπως μπορεί να δει κανείς στα εκπαιδευτικά σενάρια που αναλύονται στο κεφάλαιο 18, στις οθόνες της εφαρμογής υπάρχει μήνυμα στην οθόνη του χρήστη για το δωμάτιο στο οποίο βρίσκεται εκείνη τη στιγμή (πχ *Εικόνα 24*). Επίσης στα ίδια σενάρια η μηχανή συμπερασμού βρίσκεται στα σημεία όπου η πληροφορία από τους αισθητήρες περνά στην εφαρμογή και συμπεραίνει αυτή αν υπάρχει ησυχία ή θόρυβος.

Στο τρίτο εκπαιδευτικό σενάριο όπως περιγράφεται αναλυτικά στο 18.3 ο μηχανισμός συμπερασμού επεκτείνεται έτσι ώστε να περιλαμβάνει και άλλες πληροφορίες, όπως ας πούμε την πληροφορία για το πόσες ερωτήσεις έχουν απαντηθεί σε μια συγκεκριμένη αίθουσα, κλπ.

Η μηχανή συμπερασμού παράγει συμπεράσματα παίρνοντας υπόψη τις συνθήκες πλαισίου, τις αλήθειες από το τμήμα κανόνων και αληθειών και την αποθηκευμένη στις οντολογίες γνώση. Αποτελείται από προγραμματιστικές δομές ελέγχου, ενώ το συμπέρασμα εκφράζεται είτε με την εξαγωγή στην οθόνη κάποιων συμπερασμάτων είτε με την εισαγωγή καινούριας γνώσης στο σύστημα. Ο συμπερασμός γενικότερα δίνει στο σύστημα χαρακτήρα δυναμικό. Επιτρέπει στο χρήστη να χρησιμοποιεί ένα σύστημα έξυπνα και να το κάνει δημιουργικό για αυτόν.



### 17.6. Τμήμα αποτελεσμάτων

Το τμήμα αποτελεσμάτων είναι αλληλένδετο με το τμήμα συμπερασμού. Μπορεί να βρει κανείς τμήματα εξαγωγής αποτελεσμάτων στα οποία όπου υπάρχουν εντολές του τύπου :

```
textArea1.setText( "....." );
```

στα σημεία αυτά προφανώς γίνεται εξαγωγή αποτελεσμάτων στο textbox της εφαρμογής. Αν θέλαμε να εξάγουμε αποτελέσματα στην οθόνη μιας συσκευής θα αλλάζαμε αυτή τη γραμμή με εντολές του τύπου :

```
System.out.println( "....." );
```

Δηλαδή εντολές εκτύπωσης στη standard output, δηλαδή στην οθόνη μιας υποτιθέμενης φορητής συσκευής.

### 17.7. Αποκοπή του αποτελέσματος από την URI

Στην παράγραφο 13.6 συζητήσαμε εκτενώς το λόγο που είναι επιβεβλημένη αυτή η διαδικασία και παραθέσαμε και τον κώδικα με τον οποίο επιτυγχάνεται αυτό. Όπως αναφέρουμε και εκεί είναι μια κύρια διαδικασία η οποία επιτρέπει την αποκοπή του πραγματικού αποτελέσματος δεδομένου από τη διεύθυνση που φιλοξενεί το αρχείο ονοματολογίας της οντολογίας.

## 18. Εκπαιδευτικά Σενάρια

Στα πλαίσια της λειτουργίας της εφαρμογής σε όσο το δυνατόν πιο αληθινές συνθήκες, αναπτύξαμε κάποια εκπαιδευτικά σενάρια, τα οποία εκτός της παρούσας παρουσίασης με την υπόθεση ότι πρόκειται για πραγματικά σενάρια, έχουν τη δυνατότητα πλήθους παραλλαγών, μερικές από τις οποίες αναφέρουμε στη συνέχεια (παράγραφος 18.4). Το γεγονός αυτό δίνει ακόμη μια προστιθέμενη αξία στην εφαρμογή που αναπτύξαμε στη συγκεκριμένη διπλωματική εργασία. Στο ότι δηλαδή υπάρχει ο κορμός υλοποίησης και λειτουργίας εκπαιδευτικών σεναρίων, τα οποία μπορούν να αναπτυχθούν και να λειτουργήσουν με ελάχιστες παρεμβάσεις στον κώδικα, αλλά επιπλέον δίνεται η δυνατότητα ανάπτυξης άλλων εναλλακτικών σεναρίων τα οποία μπορούν να έχουν οδηγούς τα ήδη υλοποιημένα.



Στις επόμενες παραγράφους περιγράφουμε και σχολιάζουμε αναλυτικά, βήμα προς βήμα όλη τη διαδικασία εκτέλεσης για κάθε σενάριο ξεχωριστά.

### 18.1. Εκπαιδευτικό Σενάριο Α

Το πρώτο εκπαιδευτικό σενάριο δεν είναι δυναμικό. Το σενάριο έχει διάρκεια 48 δευτερολέπτων, και βασίζεται στο αρχείο με τα δεδομένα περιβάλλοντος contextdata.txt, (έχει περιγραφθεί αναλυτικά στο 17.4), σύμφωνα με το οποίο ο μαθητής κινείται στο χώρο έχοντας στα χέρια μια κινητή συσκευή με αισθητήρες, η οποία μπορεί και λαμβάνει τα δεδομένα του αρχείου. Η αξιολόγηση αλλάζει δύο επίπεδα. Από Δύσκολη σε Εύκολη και το αντίστροφο. Η αλλαγή επιπέδου γίνεται όταν το επίπεδο θορύβου περάσει κάποια στάθμη. Πιο συγκεκριμένα αν το πεδίο Environment:SoundPressure:Modest του αρχείου contextdata υπερβεί το 0,3 τότε το επίπεδο δυσκολίας της εξέτασης ανεβαίνει. Η κίνηση γίνεται σύμφωνα με το αρχείο και δεν επεμβαίνουμε σε αυτό. Στο τέλος μπορούμε στην οθόνη να δούμε μια πλήρη καταγραφή του σεναρίου.

Για εκπαιδευτικούς λόγους το σενάριο αυτό τρέχει παράλληλα για τρεις μαθητές. Οι δύο είναι στιγμιότυπα (instances) της οντολογίας (στατική γνώση), ενώ ο τρίτος είναι νέα γνώση που προστίθεται δυναμικά μέσα από το πρόγραμμα

Και οι τρεις έχουν στιγμιότυπο για δύσκολη εξέταση (για τους δύο πρώτους έχει ορισθεί στην οντολογία, για τον τρίτο μέσα από το πρόγραμμα), ενώ στιγμιότυπο για εύκολη εξέταση έχουν μόνο ο δεύτερος και ο τρίτος. Άρα όταν το επίπεδο δυσκολίας πέφτει, ο πρώτος παύει να εξετάζεται.

Υλοποιήσαμε κατ' αυτό τον τρόπο το σενάριο, για να δείξουμε ότι η εφαρμογή μπορεί να λειτουργήσει το ίδιο καλά είτε η γνώση είναι σταθερή και ορισμένη μέσα στην οντολογία, είτε είναι ορισμένη δυναμικά ως νέα γνώση μέσα στο πρόγραμμα.

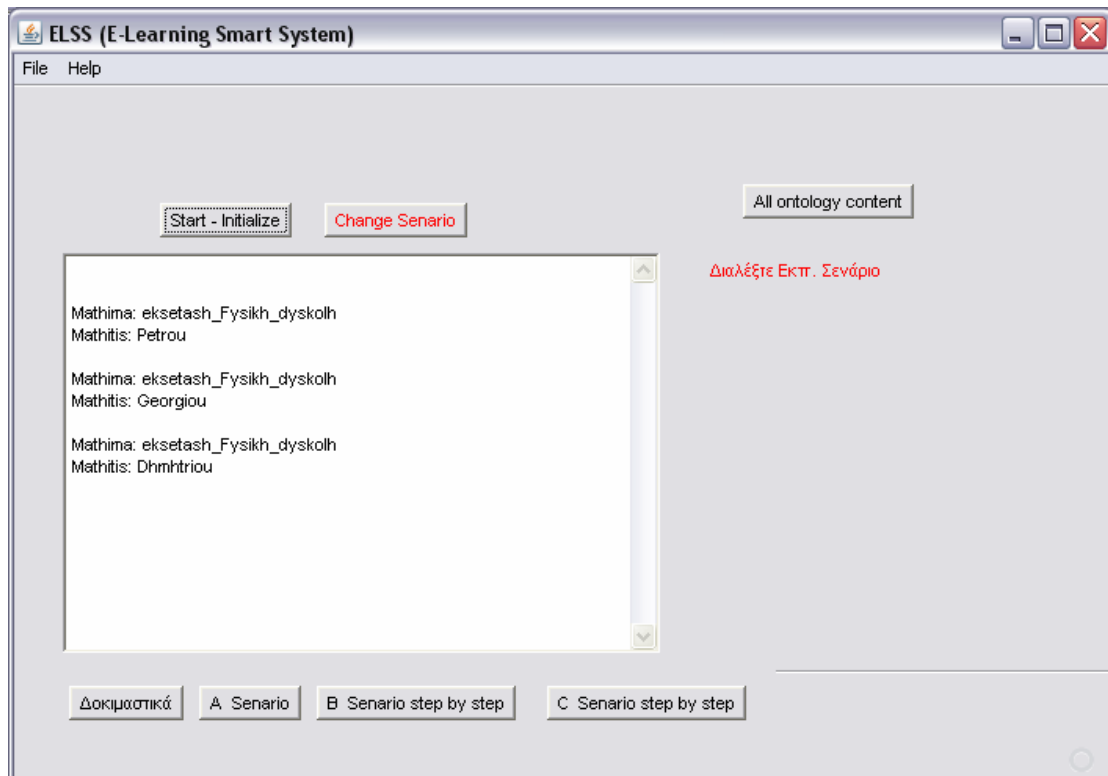
Για να γίνει πιο κατανοητό πώς λειτουργεί το σενάριο, ας το παρακολουθήσουμε βήμα-βήμα στις ακόλουθες εικόνες :

Στην (Εικόνα 19 παράγραφος 16) είδαμε και περιγράψαμε την πρώτη οθόνη της εφαρμογής. Όπως είπαμε η πρώτη ενέργεια είναι η αρχικοποίηση της εφαρμογής, με το κουμπάκι «Start-Initialize». Τότε βλέπουμε την (Εικόνα 22) στην οποία εμείς βλέπουμε το μήνυμα με το όνομα του φοιτητή, το μάθημα στο οποίο εξετάζεται και το επίπεδο δυσκολίας της εξέτασης. Φυσικά στην πραγματικότητα αντί για αυτό το μήνυμα, ο μαθητής θα πάρει το ίδιο το διαγώνισμα με το αντίστοιχο επίπεδο δυσκολίας. Προτιμήσαμε αυτόν τον τρόπο παρουσίασης, για να υπάρχει καλύτερη εποπτεία του αποτελέσματος και επειδή στόχος αυτής της εργασίας δεν είναι να υλοποιηθεί το διαγώνισμα αλλά τη διαδικασία, και ουσιαστικά να



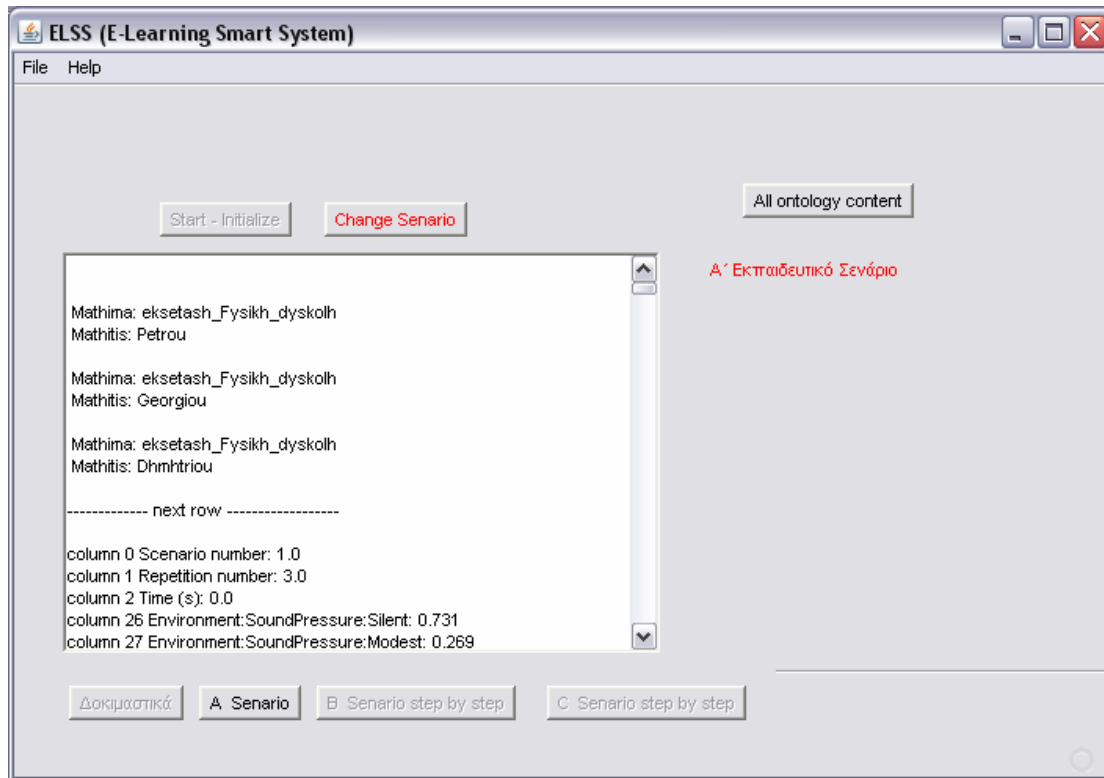
υλοποιήσει την υποδομή πάνω στην οποία μπορούν να αναπτυχθούν πολλές παραλλαγές της διαδικασίας.

Πατώντας το κουμπί της αρχικοποίησης, στην οθόνη βλέπουμε τα ονόματα των τριών μαθητών με το μάθημα στο οποίο έχουν δύσκολη εξέταση, όπως ακριβώς περιγράψαμε πιο πάνω.



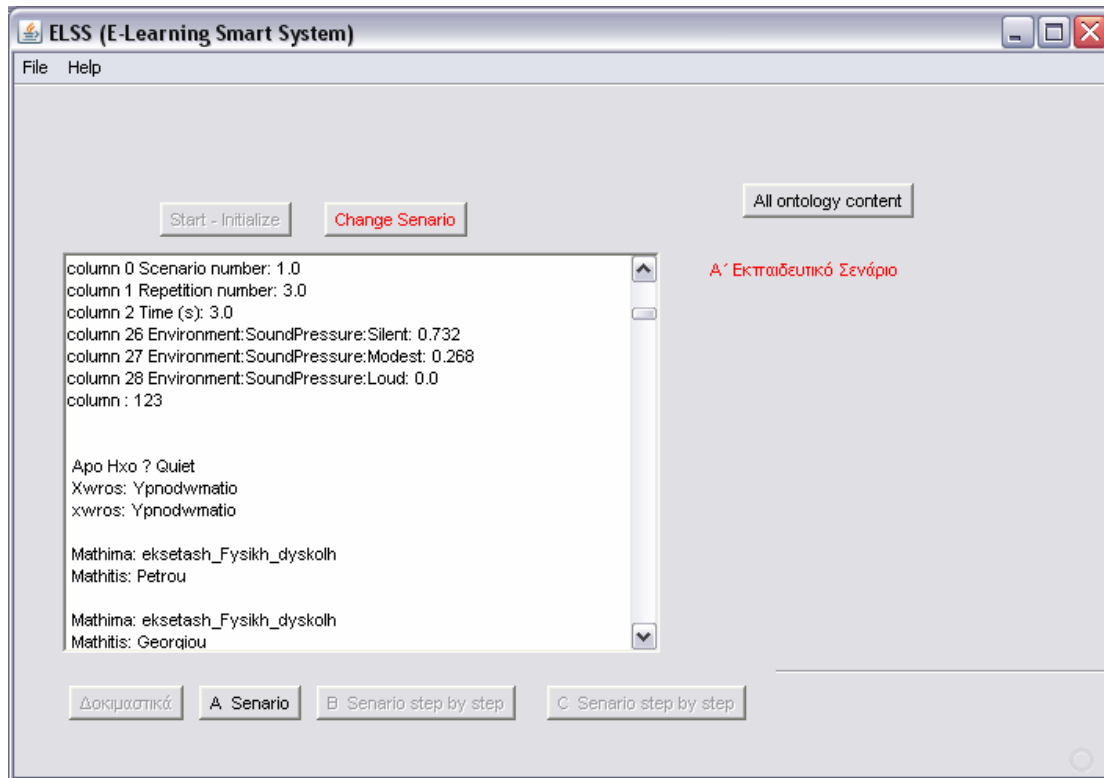
**Εικόνα 22 :** Αρχικοποίηση του σεναρίου

Για να ξεκινήσει το πρώτο εκπαιδευτικό σενάριο, επιλέγουμε το κουμπί «A Scenario» στο κάτω μέρος της οθόνης. Μόλις πατηθεί το κουμπί ξεκινάει η εκτέλεση του πρώτου σεναρίου. Σε αυτήν τη φάση συμβαίνουν τα εξής : Ανοίγει το αρχείο «contextdata.txt» και η εφαρμογή διαβάζει τα δεδομένα των στηλών 26, 27 και 28 που αφορούν τον ήχο και τη στήλη 2 που αφορά τα δευτερόλεπτα. Ο μαθητής υποτίθεται ότι κινείται στο χώρο και ακούει κάποιους ήχους. Η συσκευή του συλλαμβάνει τα δεδομένα περιβάλλοντος και τα αποθηκεύει στο αρχείο «contextdata.txt». Αυτό διαβάζει συνεχώς η εφαρμογή και ανάλογα το ύψος του θορύβου, διαμορφώνει το επίπεδο της εξέτασης. Όλη η εκτέλεση γίνεται χωρίς δική μας παρέμβαση, οπότε βλέπουμε την *Εικόνα 23*.



**Εικόνα 23 :** Εκτέλεση του σεναρίου A

και μετά από λίγο τελειώνει (ανάλογα την ταχύτητα σύνδεσης) και έτσι μπορούμε να κυλίσουμε την οθόνη της εφαρμογής πάνω κάτω και να δούμε το σενάριο ανά δευτερόλεπτο. Τότε βλέπουμε την οθόνη που φαίνεται στην *Εικόνα 24*.



Εικόνα 24 : Ο εξεταζόμενος είναι σε ήσυχο περιβάλλον

Θεωρήσαμε **χαμηλή** τη στάθμη θορύβου όταν η ένδειξη Environment Sound Pressure Modest (στήλη 27 του αρχείου «contextdata.txt») είναι μικρότερη από 0,3. Έτσι όσο παραμένει εκεί η στάθμη του θορύβου, το επίπεδο δυσκολίας της εξέτασης θα είναι υψηλό (δύσκολη εξέταση). Ενώ όταν ξεπεράσει το 0,3 θεωρούμε ότι υπάρχει θόρυβος και άρα το επίπεδο δυσκολίας θα πέσει και η εξέταση θα γίνει «εύκολη».

Η οθόνη έχει ενδείξεις οι οποίες για κάθε δευτερόλεπτο φαίνονται στην Εικόνα 25





```
----- next row -----  
column 0 Scenario number: 1.0  
column 1 Repetition number: 3.0  
column 2 Time (s): 3.0  
column 26 Environment:SoundPressure:Silent: 0.732  
column 27 Environment:SoundPressure:Modest: 0.268  
column 28 Environment:SoundPressure:Loud: 0.0  
column : 123  
  
Apo Hxo ? Quiet  
Xwros: Yrnodwmatio  
xwros: Yrnodwmatio  
  
Mathima: eksetash_Fysikh_dyskolh  
Mathitis: Petrou  
  
Mathima: eksetash_Fysikh_dyskolh  
Mathitis: Georgiou  
  
Mathima: eksetash_Fysikh_dyskolh  
Mathitis: Dhrahtriou
```

**Εικόνα 25:** Πληροφορία ανά δευτερόλεπτο πρώτου σεναρίου

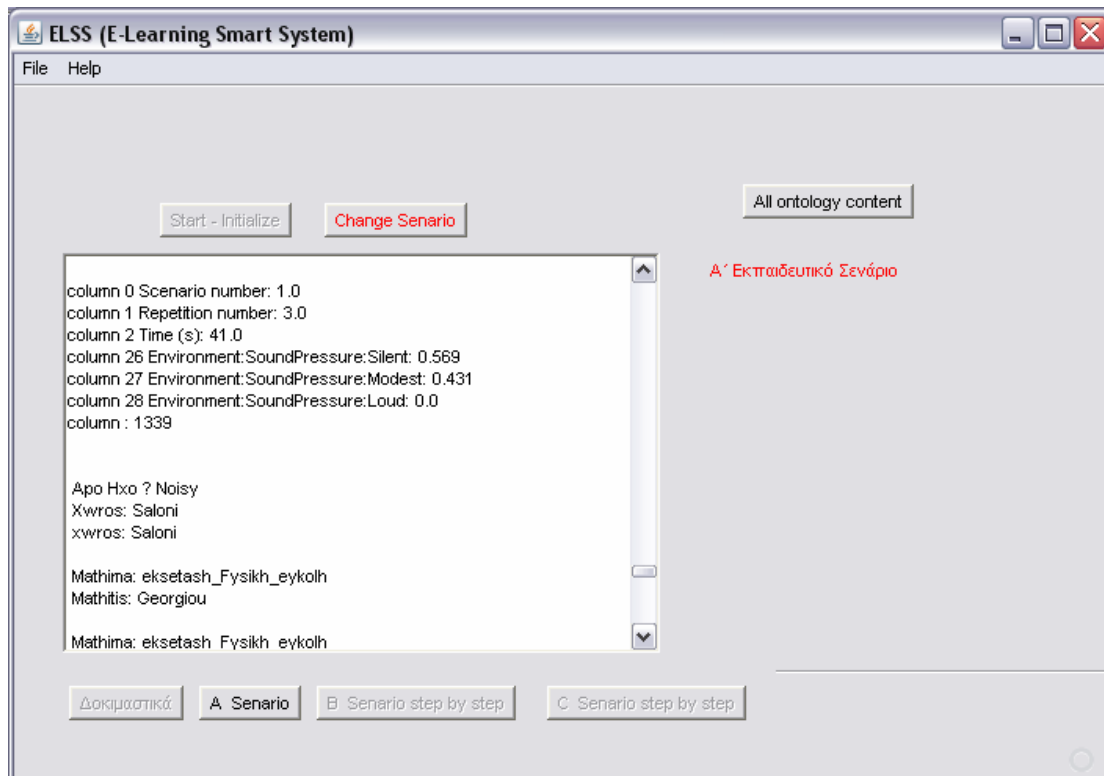
Το next row σημαίνει επόμενο δευτερόλεπτο, στη συνέχεια βλέπουμε το νούμερο του σεναρίου και τον αριθμός της επανάληψης, τα οποία είναι στοιχεία του αρχείου «contextdata.txt», τα οποία εδώ δε θα μας απασχολήσουν. Στη συνέχεια βλέπουμε το δευτερόλεπτο στο οποίο βρισκόμαστε (συνολικά είναι 48 δευτερόλεπτα), και έπειτα τα τρία επίπεδα θορύβου. Πιο κάτω βλέπουμε το αποτέλεσμα του συμπερασμού. Τα δεδομένα του πλαισίου (περιβάλλοντος έχουν εισαχθεί στο πρόγραμμα, έχουν συγκριθεί με τη νέα αλλά και τη σταθερή γνώση της οντολογίας και υπάρχει συμπερασμός και ως προς το επίπεδο του θορύβου (βλέπουμε τη λέξη «Quiet») αλλά και συμπερασμό για το δωμάτιο στο οποίο βρίσκεται («Yrnodwmatio»). Η μηχανή συμπερασμού μας βγάζει και το τελικό αποτέλεσμα στη συνέχεια που είναι η σύνδεση του ονόματος του μαθητή με το μάθημα και το επίπεδο δυσκολίας.

Προφανώς οι χαρακτηρισμοί «Quiet», «Noisy», «Yrnodwmatio», «Saloni» είναι καθαρά ενδεικτικοί και μπορούν να αλλάξουν ανάλογα το σενάριο και τη φαντασία μας κάθε φορά που θέλουμε να δημιουργήσουμε μια εκπαιδευτική διαδικασία.

Στην (Εικόνα 26) βλέπουμε και ένα ακόμα στιγμιότυπο της εκτέλεσης, όπου όπως μπορούμε να παρατηρήσουμε στο 41ο δευτερόλεπτο ο εξεταζόμενος ακούει πιο δυνατό ήχο. Αυτό φαίνεται από την ένδειξη της στάθμης Environment Sound Pressure Modest η οποία βρίσκεται στο 0,431 (> 0.3). Έτσι λοιπόν παράγεται συμπερασμός και το πρόγραμμα λαμβάνει από τη γνώση που έχει τα νέα δεδομένα και αλλάζει το επίπεδο δυσκολίας της



εξέτασης σε πιο εύκολο. Παράλληλα μας δίνει συμπερασμό για το ύψος του θορύβου («Noisy») και επίσης κάνει συμπερασμό για το πιθανό δωμάτιο στο οποίο βρίσκεται ο εξεταζόμενος.

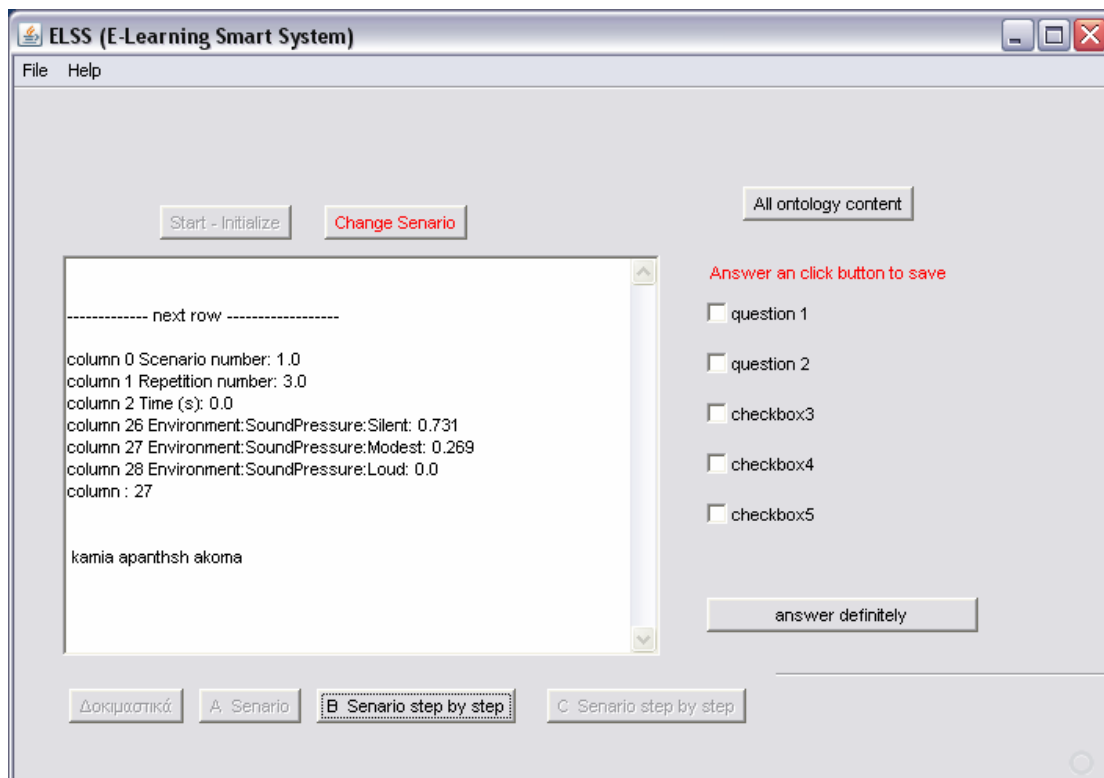


**Εικόνα 26 :** Ο εξεταζόμενος είναι σε πιο θορυβώδες περιβάλλον

Η εξέλιξη της εξέτασης είναι η ίδια. Το σενάριο τερματίζεται μετά από 48 δευτερόλεπτα και αφού ο εξεταζόμενος έχει περάσει από διάφορα σημεία με διάφορες στάθμες θορύβου.

## 18.2. Εκπαιδευτικό Σενάριο Β

Το δεύτερο εκπαιδευτικό σενάριο είναι δυναμικό σε ότι αφορά τη διάδραση των χρηστών με την εξέταση. Οι χρήστες κινούνται όπως ακριβώς περιγράφεται από το αρχείο contextdata.txt, (έχει περιγραφθεί αναλυτικά στο 17.4) με τη διαφορά ότι η κίνηση γίνεται ανά δευτερόλεπτο. Επιστρέφουμε στην *Εικόνα 22*, όπου μόλις έχουμε επιλέξει αρχικοποίηση. Πατάμε **δύο φορές** το κουμπί επιλογής του «B Scenario step by step» και βλέπουμε την *Εικόνα 27*. Στη συνέχεια ξεκινούμε τη διαδικασία, η οποία έχει ως εξής :



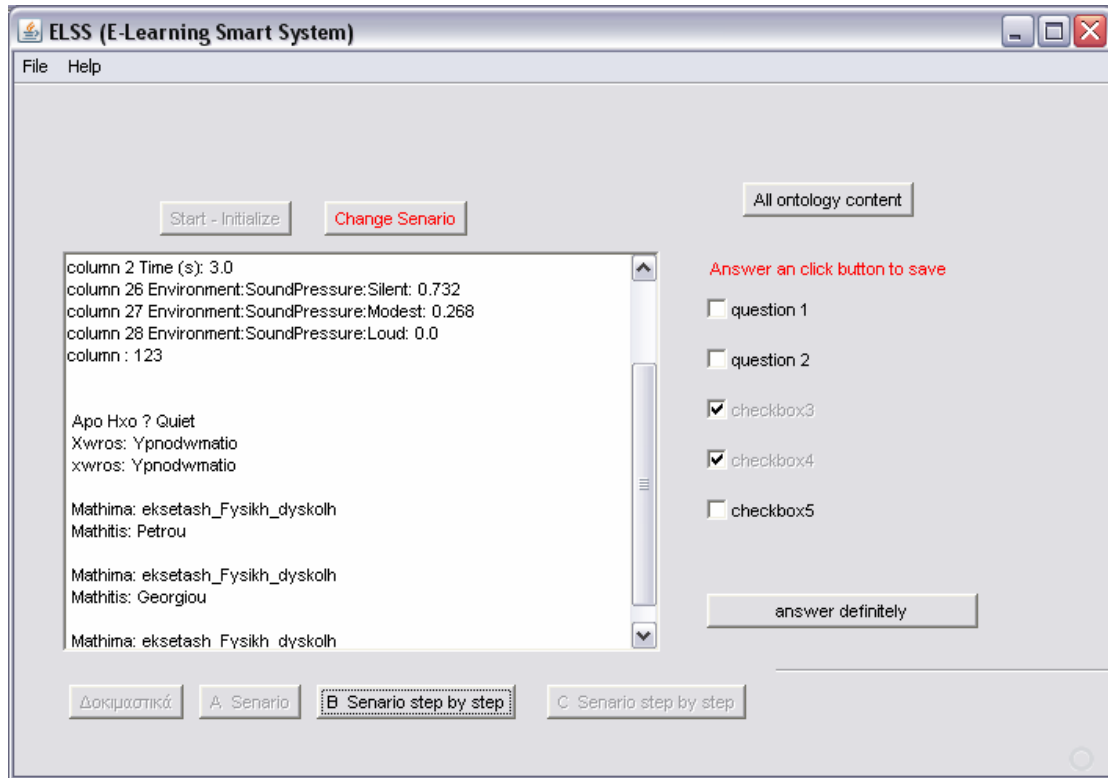
**Εικόνα 27 :** Η πρώτη οθόνη του Β σεναρίου

Κάθε δευτερόλεπτο ελέγχεται από την εκτέλεση από το πάτημα του πλήκτρου «B Scenario step by step». Κάθε φορά που πατάμε το πλήκτρο αυτό, περνά ένα δευτερόλεπτο. Στο μεσοδιάστημα μεταξύ της εκτέλεσης ενός δευτερολέπτου ο χρήστης έχει δικαίωμα να απαντήσει σε κάποια ερώτηση του ερωτηματολογίου που έχει μπροστά του. Το ερωτηματολόγιο δεν είναι πάντα το ίδιο. Αν αλλάξουν οι περιβαλλοντικές συνθήκες (η στάθμη του θορύβου) και βρεθεί σε άλλο σημείο, όπου το επίπεδο πρέπει να αλλάξει, αλλάζει και το ερωτηματολόγιο. Υπάρχουν δύο ομάδες ερωτήσεων. Οι εύκολες και οι δύσκολες. Ξεκινώντας τη δύσκολη εξέταση έχουμε μια 5άδα ερωτήσεων. Όταν το επίπεδο του θορύβου ανέβει και πρέπει να αλλάξει το επίπεδο της εξέτασης σε «εύκολο», τότε αλλάζει και η πεντάδα των ερωτήσεων. Το σύστημα εξασφαλίζει ότι οι απαντημένες ερωτήσεις του ανενεργού πλέον ερωτηματολογίου δε θα χαθούν αλλά θα επανέλθουν όταν οι συνθήκες το επιτρέψουν και επιπλέον με τον ίδιο τρόπο θα αποθηκευτούν ως νέα γνώση στην οντολογία οι απαντημένες ερωτήσεις του δεύτερου ερωτηματολογίου οι οποίες με τη σειρά τους θα επανέλθουν όταν το επιτρέψουν οι νέες συνθήκες. Κάθε φορά οι ερωτήσεις που επανέρχονται προκύπτουν από συμπερασμό στις οντολογίες με βάση τις συνθήκες περιβάλλοντος.

Πρέπει να τονίσουμε ότι κάθε φορά που απαντάει ο ενδιαφερόμενος μια απάντηση (τσεκάροντας το αντίστοιχο κουτάκι) δεν αποθηκεύεται αυτόματα και η απάντηση. Για να

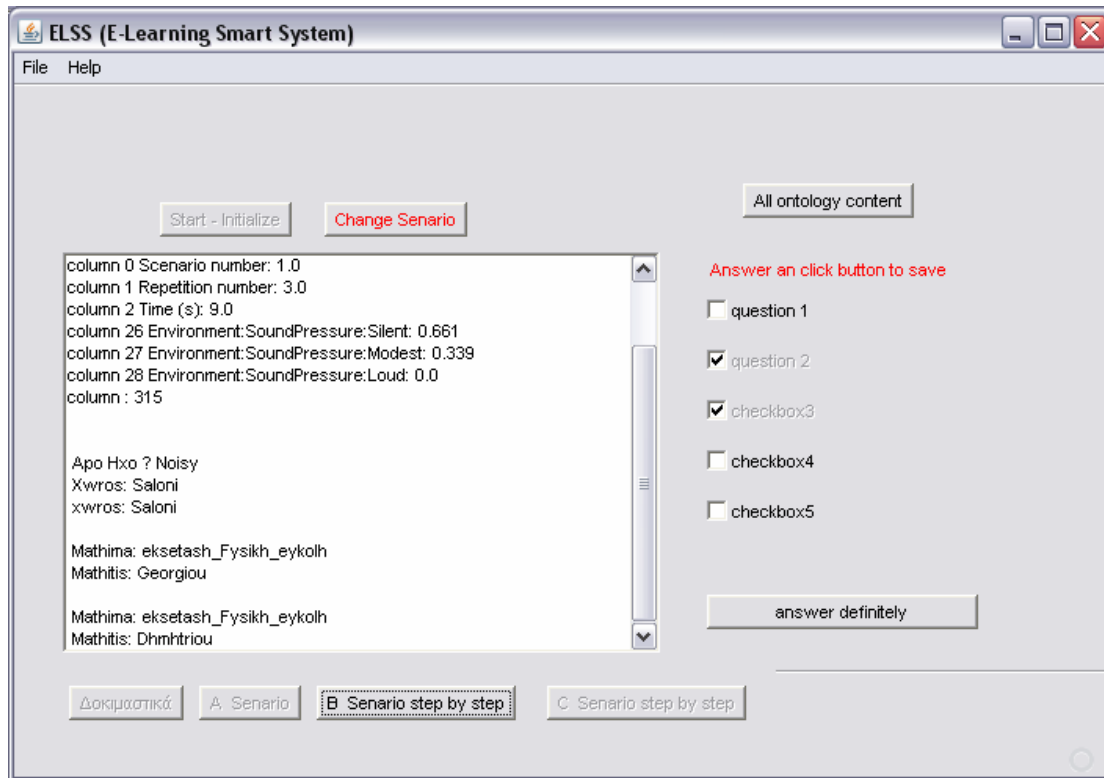


γίνει αποθήκευση της απάντησης, πρέπει να πατηθεί το κουμπάκι «answer definetly», οπότε και δεν γίνεται πλέον να αλλάξει η απάντηση. Στην *Εικόνα 28* το στιγμιότυπο είναι στο 3<sup>ο</sup> δευτερόλεπτο, το επίπεδο θορύβου είναι 0,268 (χαμηλό), άρα η εξέταση είναι «δύσκολη», έχουν απαντηθεί οι ερωτήσεις 3 και 4 οι οποίες είναι και οριστικοποιημένες, χωρίς δυνατότητα αλλαγής.



**Εικόνα 28:** Τα Β σενάριο στο 3<sup>ο</sup> δευτερόλεπτο

Στην επόμενη *Εικόνα 29*, είμαστε στο 9<sup>ο</sup> δευτερόλεπτο και πλέον το επίπεδο θορύβου είναι 0,339 (άρα υψηλό), και επομένως το επίπεδο της εξέτασης είναι «εύκολο», ενώ οι απαντήσεις που έχουν οριστικοποιηθεί είναι οι 2 και 3.



**Εικόνα 29:** Τα Β σενάριο στο 9<sup>ο</sup> δευτερόλεπτο

Εξετάζοντας τη δομή της υλοποίησης του Β σεναρίου, επιπρόσθετα στις αλλαγές που έχουμε κάνει ως προς της διάδραση με το χρήστη, αλλάζουμε και τη λογική με την οποία εισάγουμε τη νέα γνώση στο σύστημα. Πλέον η νέα γνώση δεν εισάγεται ως ομαδικό αρχείο, όπως γίνεται στο Α εκπαιδευτικό σενάριο. Η νέα γνώση σε αυτό το σενάριο σχετίζεται με την αντίδραση του χρήστη. Ως νέα γνώση στο σύστημα θεωρείται αυτή την οποία ο χρήστης εισάγει μέσω κάποιας συσκευής εισόδου. Αυτή θα μπορεί για παράδειγμα να είναι η οθόνη αφής της κινητής συσκευής.

Ο μηχανισμός συμπερασμού ενεργοποιείται

- Σε περίπτωση που ο μαθητής μεταφερθεί σε χώρο με περισσότερη ή λιγότερη φασαρία

Τα δεδομένα του περιβάλλοντος που τροφοδοτούν και επηρεάζουν το μηχανισμό συμπερασμού είναι

- Ο στάθμη του θορύβου που συλλαμβάνεται από τους αισθητήρες



### 18.3. Εκπαιδευτικό Σενάριο C

Στο τρίτο και τελευταίο εκπαιδευτικό σενάριο, υλοποιήσαμε αρκετές διαφορετικές καταστάσεις, εξαναγκάσαμε το μηχανισμό συμπερασμού να ενεργοποιηθεί σε περισσότερες περιπτώσεις, καθώς επίσης το περιβάλλον επηρεάζει τη διαδικασία εξέτασης με περισσότερους τρόπους.

Σύμφωνα με το σενάριο αυτό έχουμε καταγραφή από το σύστημα RFID [18], το οποίο περιγράψαμε στο 17.4.2. Επομένως σε αυτή την περίπτωση έχουμε έλεγχο της θέσης του εξεταζόμενου με σύστημα GPS, και μεταβολή του επιπέδου εξέτασης ανάλογα τη θέση στο δωμάτιο (ή κατ' επέκταση στο χώρο) στον οποίο κινείται. Στο εκπαιδευτικό αυτό σενάριο εκτός από την αλλαγή στο είδος των δεδομένων περιβάλλοντος, έχουμε αναπτύξει και ένα διαφορετικό σενάριο εκπαιδευτικής δραστηριότητας. Σύμφωνα με τη δραστηριότητα αυτή, ο μαθητής κινείται στους χώρους ενός μουσείου, όπου μετακινείται από αίθουσα σε αίθουσα έχοντας μια φορητή συσκευή η οποία του παρέχει μια εξέταση σχετική με τα εκθέματα που βλέπει στη συγκεκριμένη αίθουσα.

Το σενάριο διαρκεί 50 δευτερόλεπτα και η διάρκεια παραμονής σε κάθε αίθουσα είναι 5 δευτερόλεπτα. Οι αίθουσες συνολικά είναι 3. Ο μαθητής στη διάρκεια των 50 δευτερολέπτων μπορεί να επισκεφτεί τις 3 αίθουσες πάνω από μια φορά.

Με την αλλαγή της αίθουσας ο μαθητής και με βάση τα στοιχεία από το σύστημα RFID, λαμβάνει μια 5άδα ερωτήσεων οι οποίες είναι ενός επιπέδου δυσκολίας, και στις οποίες καλείται να απαντήσει.

Στις δύο από τις τρεις αίθουσες μπορεί να συνεχίσει τις απαντήσεις του και τις επόμενες επισκέψεις του πέρα από την πρώτη φορά. Στη μία όμως από τις τρεις, έχει δικαίωμα να απαντήσει μόνο κατά την πρώτη του επίσκεψη. Όταν τελειώσει η πρώτη του επίσκεψη σε αυτήν την συγκεκριμένη αίθουσα, το σύστημα του κλειδώνει τη συγκεκριμένη πεντάδα.

Συνοπτικά η πεντάδα των ερωτήσεων για κάθε αίθουσα κλειδώνει σε δύο περιπτώσεις

- Αν απαντηθούν και οι πέντε ερωτήσεις που αντιστοιχούν στην αίθουσα
- Για την μια και μοναδική αίθουσα όταν τελειώσουν τα 5 δευτερόλεπτα της πρώτης επίσκεψης

Ο μηχανισμός συμπερασμού ενεργοποιείται



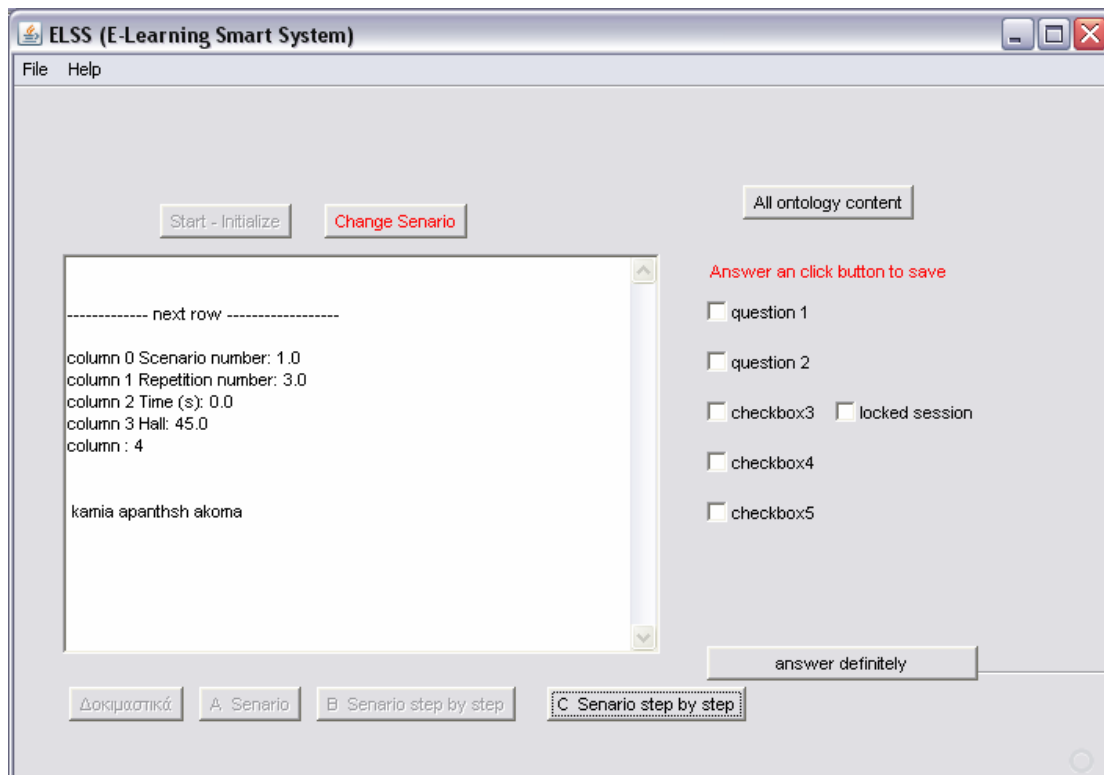
- Σε περίπτωση που ο μαθητής αλλάζει αίθουσα οπότε πρέπει να του παρέχει μια συγκεκριμένη πεντάδα ερωτήσεων
- Σε περίπτωση επανεπίσκεψης στην ίδια αίθουσα, ώστε να τον πληροφορήσει ποιες ερωτήσεις έχει ήδη απαντήσει
- για το αν θα πρέπει να κλειδώσει ή όχι τις ερωτήσεις ανάλογα σε ποια αίθουσα είναι και πόσες απαντήσεις έχει δώσει

Τα δεδομένα του περιβάλλοντος που τροφοδοτούν και επηρεάζουν το μηχανισμό συμπερασμού είναι

- τα στοιχεία από το σύστημα RFID, τα οποία στην ουσία είναι αριθμοί που δηλώνουν την αίθουσα στην οποία βρίσκεται ο εξεταζόμενος.
- Οι απαντήσεις του μαθητή (interaction).
- Ο χρόνος που περνά και μπορεί να κλειδώσει την ομάδα ερωτήσεων

Οι θόνοι που θα δούμε εκτελώντας αυτό το εκπαιδευτικό σενάριο είναι οι ακόλουθες :

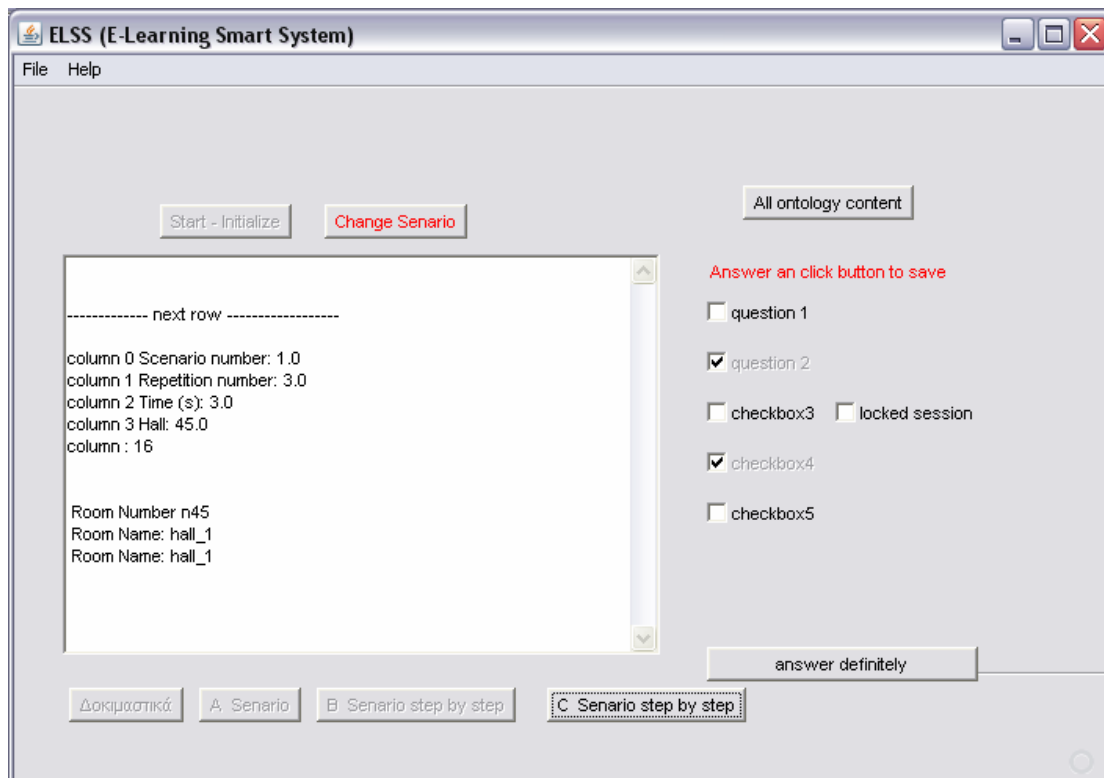
Μετά την αρχικοποίηση του σεναρίου την οποία μπορούμε να δούμε στην *Εικόνα 22*, πατούμε το κουμπάκι C Scenario δύο φορές, οπότε βλέπουμε την *Εικόνα 30*



**Εικόνα 30:** Η πρώτη οθόνη του C σεναρίου



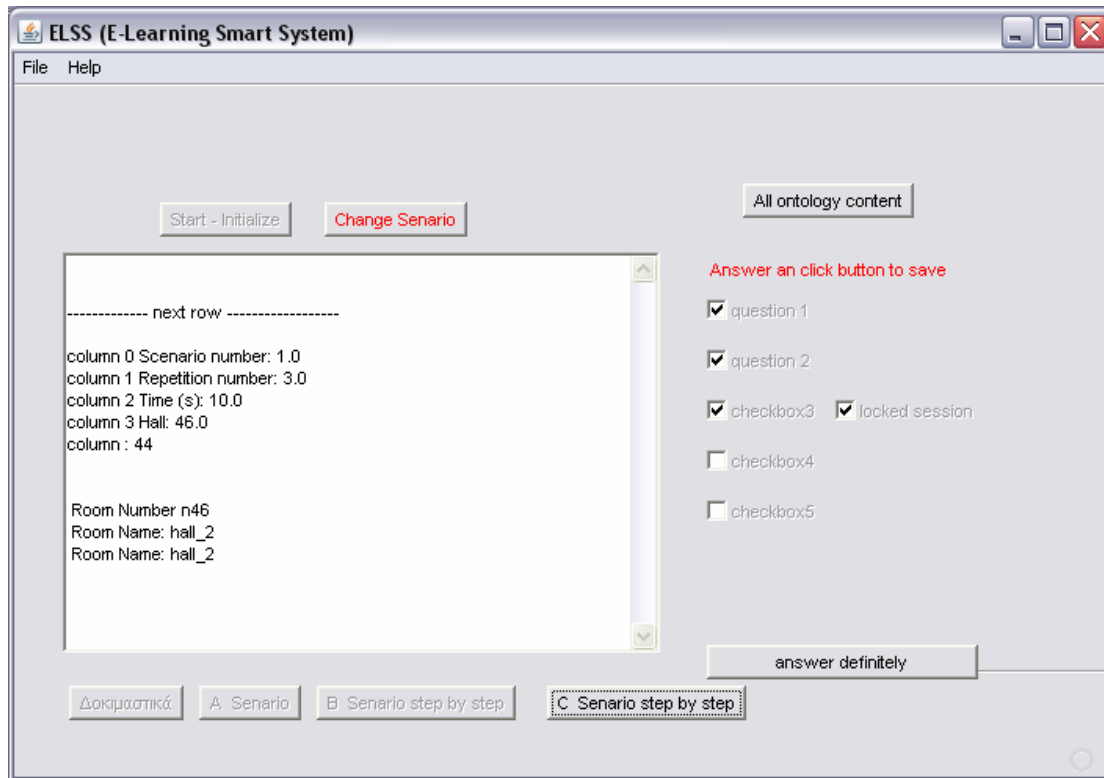
Στη συνέχεια το σενάριο εξελίσσεται ανά δευτερόλεπτο, το οποίο ελέγχουμε εμείς με το πάτημα του κουμπιού «C Scenario step by step». Κάθε φορά που θέλει να κατοχυρώσει μια απάντηση ο μαθητής, πρέπει να πατήσει το κουμπάκι, «answer definitely». Στην *Εικόνα 31* η



**Εικόνα 31:** Το C σενάριο στο 3<sup>ο</sup> δευτερόλεπτο

διαδικασία βρίσκεται στο 3ο δευτερόλεπτο, ο μαθητής είναι στην αίθουσα 45 και έχει απαντήσει στις ερωτήσεις 2 και 4. Στην *Εικόνα 32*,

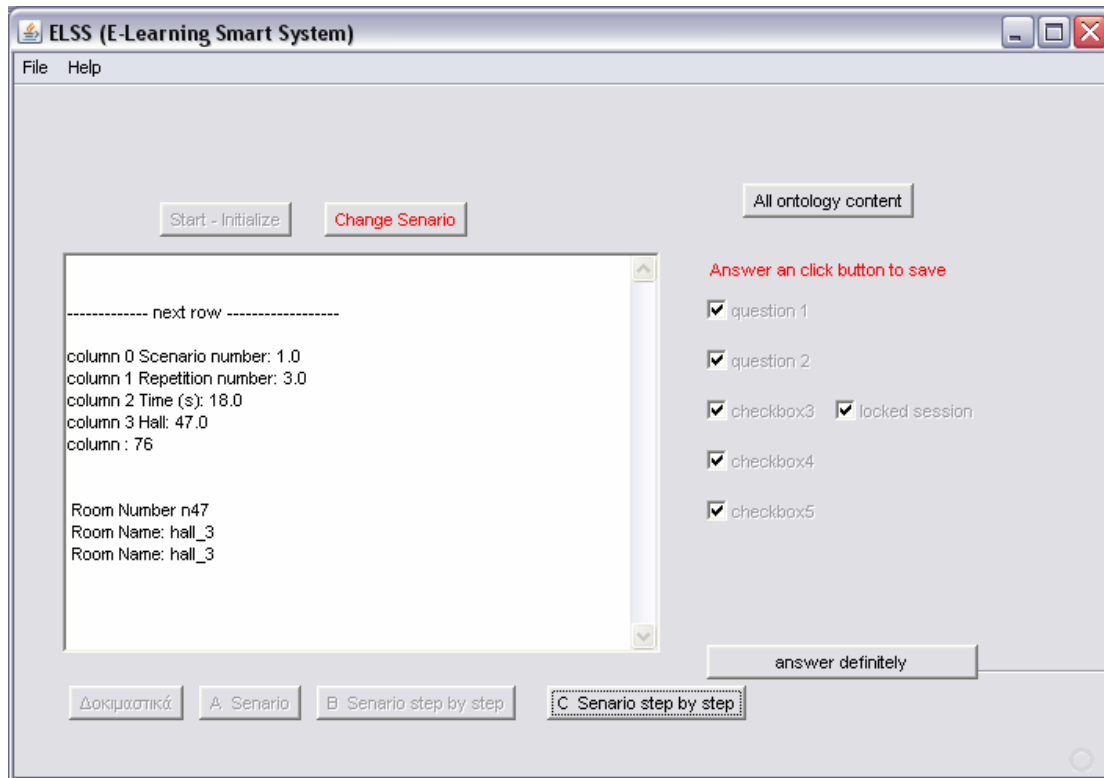




**Εικόνα 32:** Το C σενάριο στο 10<sup>ο</sup> δευτερόλεπτο

η διαδικασία βρίσκεται στο 10<sup>ο</sup> δευτερόλεπτο, ο μαθητής βρίσκεται στην αίθουσα 46 στην οποία η ιδιαιτερότητα είναι ότι έχει δικαίωμα να απαντήσει κατά την πρώτη του επίσκεψη και μόνο, και παρατηρούμε πράγματι ότι το σύστημα του κλείδωσε τη δυνατότητα να απαντήσει εκ νέου στην επόμενη επίσκεψή του στην ίδια αίθουσα, παρότι έχει απαντήσει μόνο στις ερωτήσεις 1, 2 και 3.

Στην επόμενη οθόνη τέλος (Εικόνα 33) βρισκόμαστε στο 18<sup>ο</sup> δευτερόλεπτο



Εικόνα 33: Το C σενάριο στο 18<sup>ο</sup> δευτερόλεπτο

και ο μαθητής βρίσκεται στην αίθουσα 47. Η ιδιαιτερότητα εδώ είναι ότι ο μαθητής έχει απαντήσει και τις 5 ερωτήσεις της αίθουσας, κατοχύρωσε τις απαντήσεις του και επομένως το σύστημα του κλείδωσε τις απαντήσεις, παρά το γεγονός ότι ο μαθητής θα μείνει στην αίθουσα άλλα δύο δευτερόλεπτα. Όπως αναφέραμε ήδη, τα 5 δευτερόλεπτα που διαρκεί η παραμονή σε κάθε αίθουσα είναι καθαρά ενδεικτικά και τα επιλέξαμε έτσι ώστε να μην είναι κουραστική η διαδικασία για αυτόν που εκτελεί το σενάριο για εκπαιδευτικούς λόγους.

#### 18.4. Παραλλαγές των σεναρίων

Τα εκπαιδευτικά σεναρία πάνω στα οποία εργαστήκαμε είναι προφανές ότι είναι ενδεικτικά. Ένα πλήθος άλλων σεναρίων μπορεί να διατυπωθεί και φυσικά μπορεί να βασίζεται σε άλλη εκπαιδευτική διαδικασία. Αυτό είναι και το πολύ σημαντικό κομμάτι της υλοποίησης. Ότι δηλαδή δημιουργήθηκε μια υποδομή η οποία συνέδεσε και συνδύασε πολλές διαφορετικές τεχνολογίες και μπορεί πλέον να εφαρμοσθεί σε διαφορετικές πειραματικές διατάξεις.

Ας δούμε για παράδειγμα κάποιες ενδεικτικές παραλλαγές. Στα δύο πρώτα εκπαιδευτικά σεναρία λαμβάνουν μέρος τρεις υποτιθέμενοι μαθητές οι οποίοι εξετάζονται με αποστολή της εξέτασης σε κάποια φορητή συσκευή. Ταυτόχρονα αυτοί οι μαθητές κινούνται στο χώρο. Η



κίνηση αυτή στο τρίτο σενάριο γίνεται μέρος αυτής της εξέτασης. Δηλαδή στην Τρίτη εξέταση οι ερωτήσεις σχετίζονται με τα αντικείμενα της αίθουσας στην οποία βρίσκεται ο μαθητής. Μια παραλλαγή θα ήταν το σύνολο των ερωτήσεων να αλλάζει εκτός από την αίθουσα αλλά και σε σχέση με το αντικείμενο μπροστά στο οποίο βρίσκεται κάθε φορά ο μαθητής και επιπλέον να σχετίζεται και με το πόσο κόσμο έχει τριγύρω του, και το επίπεδο της δυσκολίας να αυξομειώνεται και με την ένταση θορύβου του περιβάλλοντος.

Παραλλαγές φυσικά μπορούν να προκύψουν σε σχέση και με την εκπαιδευτική διαδικασία. Μπορεί η εκπαιδευτική διαδικασία να μη σχετίζεται με εξέταση αλλά με μια απλή διαδικασία μάθησης. Κάθε φορά, στο σημείο στο οποίο βρίσκεται ο μαθητής να μη φτάνει σύνολο ερωτήσεων αλλά εκπαιδευτικό κείμενο το οποίο σχετίζεται με το σημείο στο οποίο βρίσκεται ο μαθητής. Μπορεί επίσης να μην είναι ούτε καν αίθουσα, αλλά να είναι μια εκδρομή σε ζωολογικό κήπο, σε ένα εκπαιδευτικό πάρκο ή κάτι παρόμοιο.

Επίσης η εκπαιδευτική διαδικασία μπορεί να είναι περίπατος σε κάποιο εργοστάσιο, επιχείρηση, πανεπιστήμιο ή οτιδήποτε άλλο στο οποίο θα μπορούσε να εξελιχθεί μια εκπαιδευτική διαδικασία.

Επιπλέον η εκπαιδευτική διαδικασία μπορεί να μην αφορά ούτε καν μαθητές αλλά εκπαιδευόμενους τεχνίτες, εργάτες και οποιασδήποτε μορφής εργαζόμενους οι οποίοι θα ήθελε μια επιχείρηση να εκπαιδεύσει περνώντας τους από τα διάφορα επίπεδα παραγωγής. Φυσικά παραλλαγές μπορούν να γίνουν όχι μόνο σε σχέση με αυτόν ή αυτούς που δέχονται την υπηρεσία και την ίδια την υπηρεσία αλλά και ως προς το πώς μπορεί να εξελιχθεί η διαδικασία. Είδαμε στα δύο πρώτα σενάρια ότι μπορεί να αυξομειώνεται το επίπεδο δυσκολίας τη εξέτασης. Θα μπορούσαν οι καταστάσεις να μην είναι δύσκολη εύκολη, αλλά υπάρχει ή δεν υπάρχει εξέταση.

Επιπλέον παραλλαγές μπορούμε να σκεφτούμε και σε σχέση με τους μετρήσιμους παράγοντες των περιβαλλοντικών συνθηκών. θεωρούμε ότι οι συνθήκες περιβάλλοντος που μετράμε είναι ο τόπος και ο ήχος.

Άλλες αλλαγές μπορούν να προκύψουν από τη βελτίωση της ποιότητας και της ταξινόμησης των οντολογιών. Όπως περιγράψαμε ήδη οι οντολογίες αποτελούν σύγχρονα και δυναμικά εργαλεία για τη διαχείριση της γνώσης. Υπάρχουν πολλές εργασίες διεθνώς οι οποίες



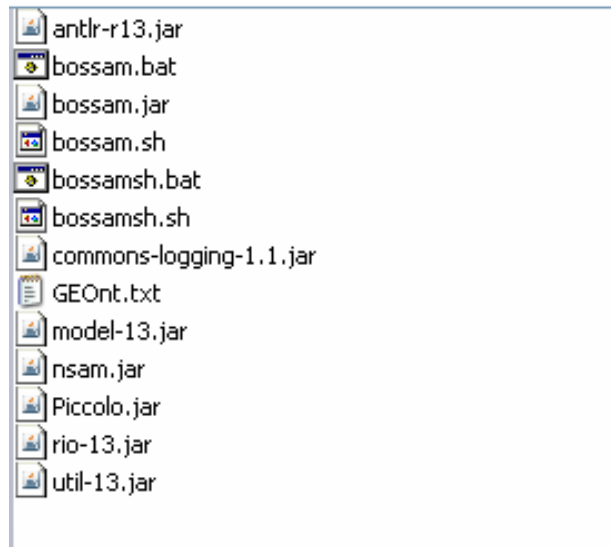
ασχολούνται με την κατασκευή οντολογιών με σκοπό την πλήρη κάλυψη ενός πεδίου. Καθώς αυξάνεται ο αριθμός των κλάσεων και η ποιότητα της ταξινομίας μιας οντολογίας θα προκύψουν και πολλές παραλλαγές στα σενάρια τα οποία επεξεργαστήκαμε στην εργασία μας. Για παράδειγμα, θα μπορούσε να προστεθεί και η κατάσταση στην οποία ο χρήστης μετακινείται καθ' ύψος και βρίσκεται σε σημείο στο οποίο δεν έχει σήμα. Τι μπορεί να γίνει σε αυτήν την περίπτωση; θα σταματήσει η εξέταση; Θα υπάρξει πιθανώς κάποιος ενταμιευτής (buffer) μνήμης στο οποίο θα αποθηκεύονται τα δεδομένα και θα φορτώνονται στο σύστημα καθώς θα ξαναέρχεται το σήμα;

Επίσης μια παραλλαγή που μπορεί να προκύψει στην εφαρμογή μας είναι η μεταφορά της μηχανής αληθειών που περιγράψαμε στο 17.3 σε κάποιον μεμονωμένο εξυπηρέτη με τη μορφή μιας ευρύτερης οντολογίας. Έτσι με κατάλληλα ερωτήματα κάθε φορά θα μπορεί να εξάγεται το κατάλληλο σύνολο αληθειών και κανόνων για την εφαρμογή μας και πάντα σύμφωνα με το σενάριο (εκπαιδευτικό ή οτιδήποτε άλλο) που έχουμε.

Τέτοιες παραλλαγές μπορούν να απασχολήσουν στο μέλλον επέκταση της εργασίας μας σε περαιτέρω έρευνα.

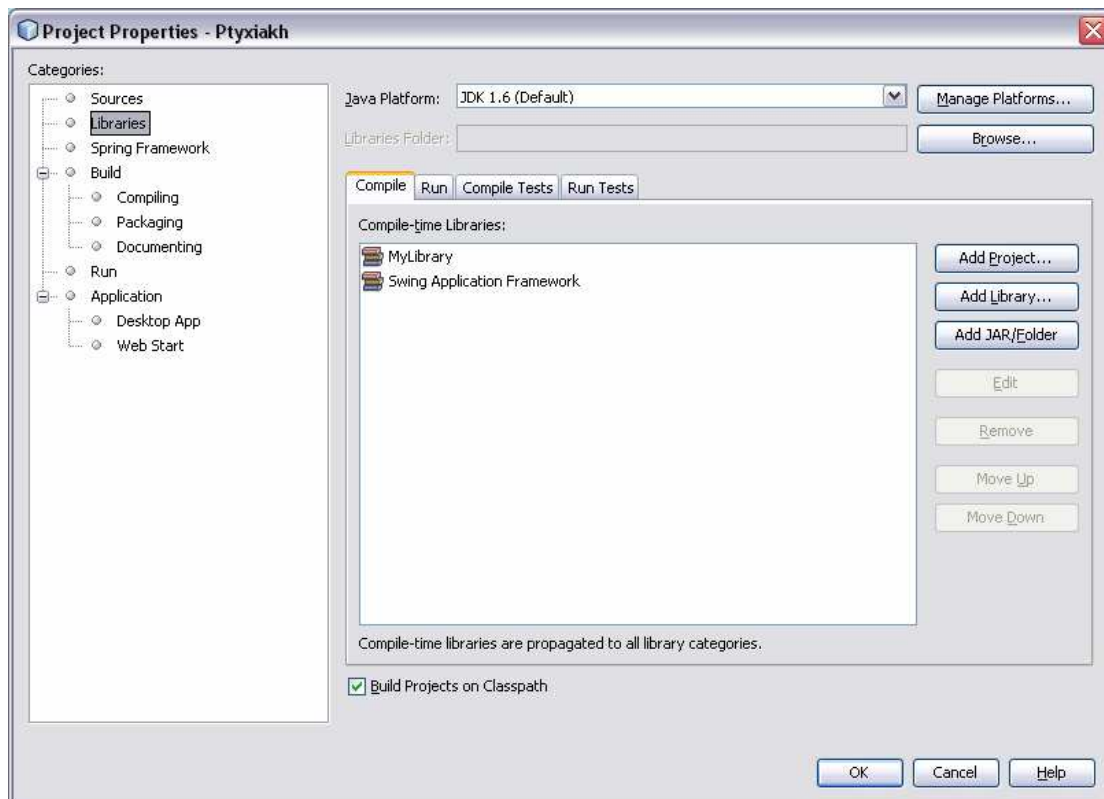
## 19. Μεταγλώττιση και Εκτέλεση της Εφαρμογής

Όπως είδαμε ήδη για την ανάπτυξη μεταγλώττιση και εκτέλεση της εφαρμογής χρησιμοποιήσαμε το περιβάλλον ανάπτυξης Netbeans το οποίο περιγράψαμε στο **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** Η ιδιαιτερότητα που προκύπτει από τη χρήση της εξωτερικής προς την Java μηχανής Bossam είναι το γεγονός ότι πρέπει να ενσωματωθούν οι βιβλιοθήκες της. Για να μπορέσουμε να λειτουργήσουμε τις συναρτήσεις της μηχανής bossam οι οποίες όπως είδαμε είναι γραμμένες σε java, χρειάζεται να ενσωματώσουμε στην εφαρμογή τις βιβλιοθήκες της μηχανής (Εικόνα 34), τις οποίες μπορεί κανείς να βρει στο [40].



Εικόνα 34 : Οι βιβλιοθήκες της μηχανής Bossam

Οι οποίες ενσωματώνονται από το παράθυρο των ιδιοτήτων της εφαρμογής (project) προσθέτοντας την αντίστοιχη βιβλιοθήκη (add library) (Εικόνα 35). Στην εικόνα φαίνεται η βιβλιοθήκη «MyLibrary» η οποία έχει ορισθεί «να δείχνει» στον κατάλογο που περιέχει τα αρχεία της Εικόνα 34.



Εικόνα 35 : Οι ιδιότητες της εφαρμογής με τις επιπρόσθετες βιβλιοθήκες



Κατά τις υπόλοιπες διαδικασίες μεταγλώττισης και εκτέλεσης δεν εμφανίζεται κάποια ιδιαιτερότητα. Τέλος να αναφέρουμε ότι τα εκτελέσιμα jar αρχεία που δημιουργούνται για να εκτελεστούν σωστά θα πρέπει στον ίδιο κατάλογο να έχουν και τον φάκελο με τις βιβλιοθήκες της Bossam.



## Βιβλιογραφία

1. Anagnostopoulos C. and Hadjiefthymiades S., "Enhancing Situation Aware Systems through Imprecise Reasoning", *IEEE Transactions on Mobile Computing*, vol. 7, no. 9, September 2008.
2. Anagnostopoulos C., Tsetsos V., Kikiras P., and Hadjiefthymiades S., "OntoNav: A Semantic Indoor Navigation System", proceedings of the 1st Workshop on Semantics in Mobile Environments (SME), MDM'05, Cyprus, May 2005.
3. Anderson L. W., Krathwohl D. R., Airasian P. W., Cruikshank K. A., Mayer R. E., Pintrich P. R., Raths J. and Wittrock M. C., «A Taxonomy for Learning, Teaching, and Assessing — A Revision of Bloom's Taxonomy of Educational Objectives»; (Eds.) Addison Wesley Longman, Inc. 2001
4. Bloom B. S. «Taxonomy of Educational Objectives: The Classification of Educational Goals» pp. 201-207; (Ed.) Susan Fauer Company, Inc. 1956.
5. Dameron, O., Musen M. A., Gibaud. B., "Using semantic dependencies for consistency management of an ontology of brain-cortex anatomy." *Artif Intell Med* 39(3):217-225, 2007.
6. Davies J., Dieter F. and Frank van Harmelen «Towards the semantic web. Ontology-driven Knowledge Management» Chichester, England: John Wiley & Sons Ltd, <http://eu.wiley.com/WileyCDA/Section/index.html>, 2003.
7. Fok A.W.P. and Ip H.H.S., "Educational Ontologies Construction for Personalized, Learning on the Web",
8. Guangzuo C., Fei C., Hu C., Shufang L., «OntoEdu: A Case Study of Ontology-based Education Grid System for E-Learning», Fifth Agricultural Ontology Service (AOS) Workshop, 27 - 29 April 2004, Beijing, China.
9. Guarino N., «Some Ontological Principles for Designing Upper Level Lexical Resources» *CoRR cmp-lg*, 9809002, 1998
10. Horridge M., Knublauch H., Rector A., Stevens R., Wroe C., «A Practical Guide To Building OWL Ontologies Using The Protege -OWL Plugin and CO-ODE Tools Edition 1.1», The University Of Manchester, Stanford University, 2004
11. Jang M., Sohn J. C., «Bossam: An Extended Rule Engine for OWL Inferencing », *Lecture Notes in Computer Science*, 2004, ISBN 978-3-540-23842-3
12. Jovanović J., Gašević D., Devedžić V., "TANGRAM: An Ontology-based Learning Environment for Intelligent Information Systems," *Proceedings of the World conference on e-Learning in corporate, Government, Healthcare, & Higher Education – E-Learn 2005*, Vancouver , Canada , October, 2005.
13. Milam J., "Ontologies in Higher Education", HigherEd.org, Inc.



14. Mizoguchi R., «Tutorial on ontological engineering - Part 2: Ontology development, tools and languages», New Generation Computing, OhmSha&Springer, Vol.22, No.1, pp.61-96, 2004
15. Nathanail S., Tsetsos V., and Hadjiefthymiades S., "Sensor-driven adaptation of web document presentation", in the proceedings of HCI International (HCII), Beijing, P.R. China, July, 2007.
16. Ogata H. and Yano Y., "JAPELAS, Context-Aware Support for Computer-Supported Ubiquitous Learning", 2<sup>nd</sup> IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'04), 2004.
17. Papataxiarhis V. , "Technologies of Knowledge Representation and Reasoning: the case of an intelligent football manager", M.Sc. thesis, Supervisors: Prof. Izambo Karali, Prof. Panagiotis Stamatopoulos, November 2008
18. Papataxiarhis V. , Riga V., Nomikos V., Sekkas Od., Kolomvatsos K., Tsetsos V., Papageorgas P., Vourakis S., Xouris V., Hadjiefthymiades S., Kouroupetroglou G., "MNISIKLIS Q Indoor Location Based Services for All",
19. Schmidt A., and Winterhalter C., "User Context Aware Delivery of E-Learning Material: Approach and Architecture", Journal of Universal Computer Science, Volume 10 / Issue 1, 2004.
20. Sirin E., Parsia B., "SPARQL-DL: SPARQL Query for OWL-DL"
21. Sure, Y., S. Staab, M. Erdmann, J. Angele, R. Studer and D. Wenke, «OntoEdit, Collaborative ontology development for the semantic web», Proc. of ISWC2002, (2002) 221-235.
22. Tsarkov D. and Horrocks I., «*FaCT++* Description Logic Reasoner: System Description», 2006
23. Tsetsos V., Anagnostopoulos C., Kikiras P., and Hadjiefthymiades S., "Semantically Enriched Navigation for Indoor Environments", International Journal on Web and Grid Services, Vol. 2, No. 4, 2006.
24. Tomai E., Spanaki M., "From ontology design to ontology implementation: A web tool for building geographic ontologies", NTUA
25. Weitzl F., Kammerl R., Göstl M., "Context Aware Reuse of Learning Resources", World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA), Lugano, Switzerland 2004.
26. Yang S. J. H., "Context Aware Ubiquitous Learning Environments for Peer-to-Peer Collaborative Learning", Educational Technology & Society, 9 (1), 2006.
27. Γιαννόπουλος Γ. Π., «GoNToggle: Έξυπνη μηχανή αναζήτησης με χρήση οντολογιών», Επιβλέπων: Ιωάννης Βασιλείου, Καθηγητής Ε.Μ.Π., Αθήνα 2006





28. Μπουρλιός Β., «Δημιουργία Οντολογίας για την ύλη της Φυσικής Λυκείου», Επιβλέπων: Φιτσιλής Πάνος, πτυχιακή εργασία, Πάτρα 2008
29. Πετράκης Σ., «Σύγχρονη Διαχείριση Γνώσης και Ομότιμα Υπολογιστικά Δίκτυα», Διπλωματική Εργασία, Επιβλέπων: Γρηγόρης Μέντζας Καθηγητής Ε.Μ.Π., Αθήνα 2007
30. Ρήγας Π, Τσέτσος Β., Παπαταξιάρχης Β., Πίνο Α., Κολομβάτσος Κ., Χατζηευθυμιάδης Ε., Κουρουπέτρογλου Γ., «ΜΝΗΣΙΚΛΗΣ: Προηγμένες Καθολικές Υπηρεσίες Θέσης σε Εσωτερικούς Χώρους», [<http://speech.di.uoa.gr/mnisiklis/> ], 2007
31. <http://protege.stanford.edu/overview/protege-owl.html>  
The Protégé platform  
(τελευταία πρόσβαση 9/12/2008)
32. EduOnto: An Ontology for Educational Resources  
<http://web.syr.edu/~jqin/eduonto/index.html> : This onology is based on the metadata schmes for the Gateway to Educational Materials and its controlled vocabulary. The class types include reusable classes (Person, Organization, and Contact), resource object classes (instructional, informational, research), and vocabulary classes (subject categories and terms).  
(τελευταία πρόσβαση 9/12/2008)
33. Writing OWL ontologies in ACE  
[http://attempto.ifi.uzh.ch/site/docs/writing\\_owl\\_in\\_ace.html](http://attempto.ifi.uzh.ch/site/docs/writing_owl_in_ace.html)  
(τελευταία πρόσβαση 9/12/2008)
34. «OWL Reasoning Examples and Hands-On Session»,  
<http://owl.man.ac.uk/2005/07/sssw/people.html>  
(τελευταία πρόσβαση 9/12/2008)
35. OWL Web Ontology Language Overview  
<http://www.w3.org/TR/owl-features/>
36. namespace οντολογιών  
<http://www.owl-ontologies.com/>
37. Pellet: The Open Source OWL DL Reasoner  
<http://clarkparsia.com/pellet/>  
(τελευταία πρόσβαση 9/12/2008)
38. The Protégé Ontology Editor and Knowledge Acquisition System  
<http://protege.stanford.edu/>
39. «Bossam» is an inference engine ( a Semantic Reasoner)  
<http://bossam.wordpress.com/>  
(τελευταία πρόσβαση 9/12/2008)



40. «Bossam» Getting Started  
[http://www.geocities.com/jangminsu/getting-started.html#creating\\_a\\_reasoner](http://www.geocities.com/jangminsu/getting-started.html#creating_a_reasoner)  
(τελευταία πρόσβαση 9/12/2008)
41. Buchingae - A Rule Language for The Web  
<http://www.geocities.com/jangminsu/buchingae-rule-language.html>  
(τελευταία πρόσβαση 9/12/2008)
42. Εργαστήριο Ψηφιακών Συστημάτων και Ψηφιακών Μέσων, «Ευφυής πρόσβαση στην Ψηφιακή Πληροφορία»  
<http://dsmc.eap.gr/semweb.php>  
(τελευταία πρόσβαση 9/12/2008)
43. Tim Berners-Lee, the Director of the World Wide Web Consortium  
<http://www.w3.org/People/Berners-Lee/>  
(τελευταία πρόσβαση 26/03/2009)
44. Διαχείριση Γνώσης, Ιστοσελίδα του Μανόλη Γεργατσούλη, Τμήμα Αρχειονομίας & Βιβλιοθηκονομίας, Ιόνιο Πανεπιστήμιο  
<http://www.ionio.gr/~manolis/courses/kmp/kmp.htm>  
(τελευταία πρόσβαση 9/12/2008)
45. Πλήρες xsd data type reference  
<http://books.xmlschemata.org/relaxng/relax-CHP-19.html>  
(τελευταία πρόσβαση 22/01/2009)
46. World Wide Web Consortium (W3C)  
<http://www.w3.org>  
(τελευταία πρόσβαση 22/01/2009)
47. knowledge Systems AI, Laboratory  
<http://www.ksl.stanford.edu/kst/what-is-an-ontology.html>  
(τελευταία πρόσβαση 10/02/2009)
48. First-order logic  
[http://en.wikipedia.org/wiki/First-order\\_logic](http://en.wikipedia.org/wiki/First-order_logic)  
(τελευταία πρόσβαση 26/03/2009)



## Παράρτημα Α

### Κώδικας 1

```
// Με τον κώδικα 1 .....  
  
public class WineQuery01  
{  
  
    // Ορίζουμε το prefix της οντολογίας και παράλληλα  
    // ορίζουμε το URI της οντολογίας  
    // Η οντολογία GEOnt βρίσκεται σε έναν προσωπικό server  
  
    final static String GEOntURI =  
    "http://vsdesign.homelinux.org/~bstefan/GEOnt.owl#";  
  
    // Εδώ λοιπόν ορίζουμε το prefix για το namespace  
  
    final static String ruleURI = "http://www.owl-  
ontologies.com/Ontology1225893950.owl#";  
  
    public static void main(String[] args)  
    {  
        try  
        {  
  
            // δημιουργούμε τον reasoner  
  
            IReasonerFactory factory = ReasonerFactory.getInstance();  
            IReasoner r = factory.createOwlDlTrMReasoner();  
  
            // Φορτώνουμε την οντολογία GEOnt στη μηχανή Bossam  
  
            r.load(IReasoner.OWL, GEOntURI);  
  
            // εκτελούμε το reasoning  
  
            String result = r.run();  
  
            // Εδώ εκτυπώνεται στην οθόνη το αποτέλεσμα που προέκυψε από την  
            // εκτέλεση του reasoning  
            System.out.println("Conclusions: n" + result);  
  
            // θέτουμε το namespace prefix της οντολογίας wine ως «w»  
  
            r.setNamespacePrefix("w", ruleURI);  
            // εκτελούμε ένα ερώτημα και το αποτέλεσμα αποθηκεύεται  
            // στην μεταβλητή answer  
            // Η ερώτηση είναι : βρες τα στιγμιότυπα της οντολογίας  
            // που είναι mathitis και τοποθέτησέ τα στη μεταβλητή x  
  
            Answer answer = r.ask1("query q is w:Mathitis(?x);");  
  
            // Τύπωσε το αποτέλεσμα
```



```
// Prints out the query result
if (answer == null)
{
    System.out.println("The query returns false!");
}
else
{
    System.out.println("Answer (" + answer.getBindings().size() +
":n"
                        + answer);
}
}
catch (Exception e)
{
    e.printStackTrace();
}
}
```



## Κώδικας 2

```
try {  
  
    IReasonerFactory reasonerFactory =  
        ReasonerFactory.getInstance();  
    IReasoner r = reasonerFactory.createOwlDlTrMReasoner();  
    r.load(IReasoner.OWL, GEontURI);  
    r.setSerializationMode(IReasoner.DEFAULT);  
    r.setBaseNamespace(ruleURI);  
    r.setNamespacePrefix("w", ruleURI);  
  
    r.tell("fact a is didaskei(w:John,w:Gymnastikh);");  
    r.tell("fact b is exei_epilexthei(w:Gymnastikh,w:Nil);");  
    String result3 = r.ask("query q is didaskei(?x,?y) and  
        exei_epilexthei(?y,?z);");  
  
    String result2 = r.run();  
  
    if (result2 == null)  
    {  
        System.out.println("The query returns false!");  
    }  
    else  
    {  
        System.out.println(result2);  
        System.out.println(result3);  
    }  
  
}  
catch (Exception e)  
{  
    e.printStackTrace();  
}  
}
```

Η εκτέλεση του παραπάνω κώδικα δίνει :

.....  
.....

```
// old_execution, pali to anapodo paroysiazetai  
// O Petrou htan mathitis mono toy Roumeli  
// O Georgiou htan mathitis mono toy Athanasiou  
  
fact http://bossam.com/default#Fact303 is http://www.owl-  
ontologies.com/Ontology1225893950.owl#einai_mathitis(http://www.owl-  
ontologies.com/Ontology1225893950.owl#Petrou,http://www.owl-  
ontologies.com/Ontology1225893950.owl#Athanasiou);  
  
fact http://bossam.com/default#Fact304 is http://www.owl-  
ontologies.com/Ontology1225893950.owl#einai_mathitis(http://www.owl-
```



```
ontologies.com/Ontology1225893950.owl#Georgiou,http://www.owl-  
ontologies.com/Ontology1225893950.owl#Roumelis);  
  
// Edw einai h ektelesh toy result2  
// O Petrou einai mathitis toy Roumeli  
// Ogeorgiou einai mathitis kai toy Roumeli kai toy Athanasiou  
  
fact http://bossam.com/default#Fact211 is http://www.owl-  
ontologies.com/Ontology1225893950.owl#einai_mathitis(http://www.owl-  
ontologies.com/Ontology1225893950.owl#Petrou,http://www.owl-  
ontologies.com/Ontology1225893950.owl#Athanasiou);  
  
// Edw einai h ektelesh toy result3  
  
{[?z = http://www.owl-  
ontologies.com/Ontology1225893950.owl#Petrou][?y = http://www.owl-  
ontologies.com/Ontology1225893950.owl#Mathimatika][?x =  
http://www.owl-ontologies.com/Ontology1225893950.owl#Roumelis]}  
{[?z = http://www.owl-  
ontologies.com/Ontology1225893950.owl#Georgiou][?y = http://www.owl-  
ontologies.com/Ontology1225893950.owl#Mathimatika][?x =  
http://www.owl-ontologies.com/Ontology1225893950.owl#Roumelis]}  
{[?z = http://www.owl-  
ontologies.com/Ontology1225893950.owl#Georgiou][?y = http://www.owl-  
ontologies.com/Ontology1225893950.owl#Fysikh][?x = http://www.owl-  
ontologies.com/Ontology1225893950.owl#Athanasiou]}
```



### Κώδικας 3

```
// καταρχήν βλέπουμε τα imports

import java.io.FileInputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Hashtable;
import java.util.Iterator;
import java.util.List;
import java.util.Properties;
import java.util.Set;
import java.util.Vector;

import bossam.app.Answer;
import bossam.app.IReasoner;
import bossam.app.IReasonerFactory;
import bossam.app.ReasonerFactory;

// Ο Ορισμός των reasoners

IReasonerFactory reasonerFactory = ReasonerFactory.getInstance();
IReasoner r = reasonerFactory.createOwlDlTrMReasoner();
r.load(IReasoner.OWL, GEontURI);
r.setSerializationMode(IReasoner.DEFAULT);
r.setBaseNamespace(GEontURI);
r.setNamespacePrefix("w", ruleURI);
r.setNamespacePrefix("g", GEontURI);

IReasonerFactory reasonerFactory2 = ReasonerFactory.getInstance();
IReasoner ir = reasonerFactory2.createOwlDlTrMReasoner();
ir.load(IReasoner.OWL, contextURI);
ir.setSerializationMode(IReasoner.DEFAULT);
ir.setBaseNamespace(ruleURI2);
ir.setNamespacePrefix("i", ruleURI2);
ir.setNamespacePrefix("co", contextURI);

// Eisagwgh neas gnwshs se property poy den yparxei stis
ontologies GEont & INO
// kai emfanish mono twn properties pou exoun zititheis sta
facts
// mporoyme dhl na kanoume query mono tis idiothtes didaskei
& exei_epilexthei

//ir.classifyTaxonomy(null);
ir.tell("fact ic is has(i:Noisy,i:Saloni);");
ir.tell("fact ic2 is has(i:Noisy,i:Sala);");
ir.tell("fact id is has(i:quiet,i:Ypnodwmatio);");
r.tell("fact e is
eksetazetai(g:eksetash_Fysikh_metria,g:Georgiou);");
```



```
r.tell("fact e2 is
eksetazetai(g:eksetash_Fysikh_metria,g:Ioannou);");
r.tell("fact e3 is
eksetazetai(g:eksetash_Fysikh_dyskolh,g:Petrou);");
r.tell("fact f is epilegei(w:Mathimatika,w:Remalhs);");

r.tell("fact c is didaskei(g:Vranas,g:Istoria);");
r.tell("fact d is exei_epilexthei(g:Istoria,g:Kolokythas);");
    r.tell("fact a is didaskei(w:John,w:Gymnastikh);");
r.tell("fact b is exei_epilexthei(g:Gymnastikh,g:Nil);");

// Ερωτήματα

String result4 = r.ask("query q is didaskei(?x,?y) and
exei_epilexthei(?y,?z);");
String result6 = ir.ask("query q is has(?x,?z) and
(?x=Noisy);");
Answer result8 = r.ask1("query q is eksetazetai(?x,?y) and
(?x=eksetash_Fysikh_metria);"); // 6
Answer result9 = r.ask1("query q is
w:exei_epilexthei(?x,?z);");
//Answer result6 = r.ask1("query q is w:exei_mathiti(?x);");
String resulti5 = ir.run();
String resulti7 = r.run();

// Diadikasia Diaxwrismou tw n apotelesmatwn

ArrayList actions = new ArrayList(result9.getBindings());

for (Iterator it = actions.iterator(); it.hasNext();) {
    HashMap hm = (HashMap) it.next();
    String segment = hm.get("?x").toString();
    segment = segment.substring(segment.indexOf("#") +
1);

    String service = hm.get("?z").toString();
    service = service.substring(service.indexOf("#") +
1);

    System.out.println("\n Mathima: " + segment);
    System.out.println("\n Mathitis: " + service);
}

// Εκτύπωση Αποτελεσμάτων

if (result6 == null)
{
    System.out.println("The query returns false!");
}
else
{
    //System.out.println("Fourth session:\n" + resulti5);
    System.out.println("Apo GEOnt result4:\n" + result4);
    System.out.println("Apo context result6:\n" + result6);
    System.out.println("Apo GEOnt & context result8:\n" +
result8);
    System.out.println("Apo GEOnt result9:\n" + result9);
}
```





```
    }  
  }  
  catch (Exception e)  
  {  
    e.printStackTrace();  
  }  
}  
  
// Αποτέλεσμα :  
  
Fourth session:  
Mathima: Fysikh  
Mathitis: Georgiou  
Mathima: Mathimatika  
Mathitis: Petrou  
Mathima: Mathimatika  
Mathitis: Georgiou  
Mathima: Fysikh  
Mathitis: Petrou  
  
Apo GEOnt result4:  
  
{[?z = http://www.owl-ontologies.com/Ontology1225893950.owl#Nil][?y =  
http://www.owl-ontologies.com/Ontology1225893950.owl#Gymnastikh][?x =  
http://www.owl-ontologies.com/Ontology1225893950.owl#John]}  
{[?z = http://www.owl-  
ontologies.com/Ontology1225893950.owl#Kolokythas][?y =  
http://www.owl-ontologies.com/Ontology1225893950.owl#Istoria][?x =  
http://www.owl-ontologies.com/Ontology1225893950.owl#Vranas]}  
{[?z =  
http://vsdesign.homelinux.org/~bstefan/GEOnt.owl#Kolokythas][?y =  
http://vsdesign.homelinux.org/~bstefan/GEOnt.owl#Istoria][?x =  
http://vsdesign.homelinux.org/~bstefan/GEOnt.owl#Vranas]}  
  
Apo context result6:  
  
{[?z = http://www.owl-  
ontologies.com/Ontology1229414213.owl#Saloni][?x = http://www.owl-  
ontologies.com/Ontology1229414213.owl#Noisy]}  
{[?z = http://www.owl-ontologies.com/Ontology1229414213.owl#Sala][?x  
= http://www.owl-ontologies.com/Ontology1229414213.owl#Noisy]}  
  
Apo GEOnt & context result8:
```



```
{[?x=http://vsdesign.homelinux.org/~bstefan/GEOnt.owl#eksetash_Fysikh_metria][?y=http://vsdesign.homelinux.org/~bstefan/GEOnt.owl#Georgiou]}
```

```
{[?x=http://vsdesign.homelinux.org/~bstefan/GEOnt.owl#eksetash_Fysikh_metria][?y=http://vsdesign.homelinux.org/~bstefan/GEOnt.owl#Ioannou]}
```

Apo GEOnt result9:

```
{[?x=http://www.owl-ontologies.com/Ontology1225893950.owl#Fysikh][?z=http://www.owl-ontologies.com/Ontology1225893950.owl#Georgiou]}
```

```
{[?x=http://www.owl-ontologies.com/Ontology1225893950.owl#Mathimatika][?z=http://www.owl-ontologies.com/Ontology1225893950.owl#Petrou]}
```

```
{[?x=http://www.owl-ontologies.com/Ontology1225893950.owl#Mathimatika][?z=http://www.owl-ontologies.com/Ontology1225893950.owl#Georgiou]}
```

```
{[?x=http://www.owl-ontologies.com/Ontology1225893950.owl#Fysikh][?z=http://www.owl-ontologies.com/Ontology1225893950.owl#Petrou]}
```



## Παράρτημα Β

### Λίγα λόγια για τις πλατφόρμες ανάπτυξης Eclipse και NetBeans

Το eclipse (<http://www.eclipse.org/>) είναι μια πλατφόρμα με πολύ γνώριμο περιβάλλον ανάπτυξης εφαρμογών για τους προγραμματιστές Java. Είναι πλατφόρμα ανοικτού και ελεύθερου λογισμικού και αυτό από μόνο του προσδίδει ένα τεράστιο πλεονέκτημα στη χρήση της. Το περιβάλλον εργασίας και ο συντάκτης που διαθέτει είναι σύγχρονος και κατάλληλος για ανάπτυξη προγραμμάτων σε Java αφού παρέχει πολυχρωμία ανάλογα τις εντολές και έτσι ξεκουράζει τη διόρθωση λαθών και την παρακολούθηση της ανάπτυξης.

Το Netbeans (<http://www.netbeans.org/>) είναι επίσης μια πολύ σύγχρονη πλατφόρμα η οποία δίνει με ένα σύγχρονο και δυναμικό περιβάλλον που διαθέτει παρέχει δυνατότητες γρήγορης και εύκολης δημιουργίας GUI (στο μοντέλο της Delphi και της VB).

Επιλέχθηκε ως η τελική πλατφόρμα ανάπτυξης της εφαρμογής, λόγω της ευχρηστίας στη δημιουργία Γραφικού περιβάλλοντος Διεπαφής Χρήστη (GUI) και λόγω του σύγχρονου και εξειδικευμένου συντάκτη της γλώσσας που έχει. Ο συγκεκριμένος συντάκτης κάνει error detection κατά τη διάρκεια της συγγραφής του κώδικα, βοηθώντας τον προγραμματιστή να αντιληφθεί συντακτικά λάθη και παραλείψεις.

Φυσικά τα λογικά λάθη φαίνονται κατά τη διάρκεια της εκτέλεσης, όμως και σε αυτήν την ευαίσθητη κατηγορία λαθών, η πλατφόρμα δίνει κάποιες μικρές βοήθειες κατά κάποιον τρόπο και στο βαθμό που αυτό είναι επιτρεπτό. Ως παράδειγμα μπορούμε να αναφέρουμε το γεγονός, ότι βγάζει alarm όταν μια μεταβλητή χρησιμοποιείται χωρίς να έχει οριστεί, ή όταν ορίζεται αλλά δεν χρησιμοποιείται. Επίσης μπορεί να κάνει αυτόματη εισαγωγή βιβλιοθηκών όταν η συνάρτηση που προσπαθούμε να χρησιμοποιήσουμε δεν έχει γίνει import η αντίστοιχη βιβλιοθήκη της. Επίσης όπως φυσικά και η eclipse υποστηρίζει ένα μεγάλο πλήθος παράπλευρων λειτουργιών οι οποίες παραμετροποιούν σε μεγάλο βαθμό και σε πολύ μεγάλη λεπτομέρεια την εφαρμογή μας.



### Κώδικας προγράμματος

Στη συνέχεια παραθέτουμε τον κώδικα του προγράμματος της εφαρμογής μας.

```
/*
 * PtyxiakhView.java
 *
 * by Vassilis Stefanidis
 *

package ptyxiakh;

import java.io.FileInputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Hashtable;
import java.util.Iterator;
import java.util.List;
import java.util.Properties;
import java.util.Set;
import java.util.Vector;

import org.jdesktop.application.Action;
import org.jdesktop.application.ResourceMap;
import org.jdesktop.application.SingleFrameApplication;
import org.jdesktop.application.FrameView;
import org.jdesktop.application.TaskMonitor;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.Timer;
import javax.swing.Icon;
import javax.swing.JDialog;
import javax.swing.JFrame;

import bossam.app.Answer;
import bossam.app.IReasoner;
import bossam.app.IReasonerFactory;
import bossam.app.ReasonerFactory;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.File;
import java.io.FileInputStream;
```



```
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.Reader;
import java.io.StreamTokenizer;

/**
 * The application's main frame.
 */
public class PtyxiakhView extends FrameView {

    public PtyxiakhView(SingleFrameApplication app) {
        super(app);

        initComponents();

        // status bar initialization - message timeout, idle icon and busy animation, etc
        ResourceMap resourceMap = getResourceMap();
        int messageTimeout = resourceMap.getInteger("StatusBar.messageTimeout");
        messageTimer = new Timer(messageTimeout, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                statusMessageLabel.setText("");
            }
        });
        messageTimer.setRepeats(false);
        int busyAnimationRate =
resourceMap.getInteger("StatusBar.busyAnimationRate");
        for (int i = 0; i < busyIcons.length; i++) {
            busyIcons[i] = resourceMap.getIcon("StatusBar.busyIcons[" + i + "]");
        }
        busyIconTimer = new Timer(busyAnimationRate, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                busyIconIndex = (busyIconIndex + 1) % busyIcons.length;
                statusAnimationLabel.setIcon(busyIcons[busyIconIndex]);
            }
        });
        idleIcon = resourceMap.getIcon("StatusBar.idleIcon");
        statusAnimationLabel.setIcon(idleIcon);
        progressBar.setVisible(false);

        // connecting action tasks to status bar via TaskMonitor
        TaskMonitor taskMonitor = new TaskMonitor(getApplication().getContext());
        taskMonitor.addPropertyChangeListener(new
java.beans.PropertyChangeListener() {
```



```
public void propertyChange(java.beans.PropertyChangeEvent evt) {
    String propertyName = evt.getPropertyName();
    if ("started".equals(propertyName)) {
        if (!busyIconTimer.isRunning()) {
            statusAnimationLabel.setIcon(busyIcons[0]);
            busyIconIndex = 0;
            busyIconTimer.start();
        }
        progressBar.setVisible(true);
        progressBar.setIndeterminate(true);
    } else if ("done".equals(propertyName)) {
        busyIconTimer.stop();
        statusAnimationLabel.setIcon(idleIcon);
        progressBar.setVisible(false);
        progressBar.setValue(0);
    } else if ("message".equals(propertyName)) {
        String text = (String)(evt.getNewValue());
        statusMessageLabel.setText((text == null) ? "" : text);
        messageTimer.restart();
    } else if ("progress".equals(propertyName)) {
        int value = (Integer)(evt.getNewValue());
        progressBar.setVisible(true);
        progressBar.setIndeterminate(false);
        progressBar.setValue(value);
    }
}
});
}

@Action
public void showAboutBox() {
    if (aboutBox == null) {
        JFrame mainFrame = PtyxiakhApp.getApplication().getMainFrame();
        aboutBox = new PtyxiakhAboutBox(mainFrame);
        aboutBox.setLocationRelativeTo(mainFrame);
    }
    PtyxiakhApp.getApplication().show(aboutBox);
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
```



```
private void initComponents() {

    mainPanel = new javax.swing.JPanel();
    button1 = new java.awt.Button();
    button2 = new java.awt.Button();
    textArea1 = new java.awt.TextArea();
    button3 = new java.awt.Button();
    button4 = new java.awt.Button();
    button5 = new java.awt.Button();
    button6 = new java.awt.Button();
    button7 = new java.awt.Button();
    button8 = new java.awt.Button();
    checkbox1 = new java.awt.Checkbox();
    checkbox2 = new java.awt.Checkbox();
    checkbox3 = new java.awt.Checkbox();
    checkbox4 = new java.awt.Checkbox();
    checkbox5 = new java.awt.Checkbox();
    button11 = new java.awt.Button();
    button12 = new java.awt.Button();
    button13 = new java.awt.Button();
    label1 = new java.awt.Label();
    button16 = new java.awt.Button();
    button17 = new java.awt.Button();
    checkbox6 = new java.awt.Checkbox();
    button18 = new java.awt.Button();
    menuBar = new javax.swing.JMenuBar();
    javax.swing.JMenu fileMenu = new javax.swing.JMenu();
    javax.swing.JMenuItem exitMenuItem = new javax.swing.JMenuItem();
    javax.swing.JMenu helpMenu = new javax.swing.JMenu();
    javax.swing.JMenuItem aboutMenuItem = new javax.swing.JMenuItem();
    statusPanel = new javax.swing.JPanel();
    javax.swing.JSeparator statusPanelSeparator = new javax.swing.JSeparator();
    statusMessageLabel = new javax.swing.JLabel();
    statusAnimationLabel = new javax.swing.JLabel();
    progressBar = new javax.swing.JProgressBar();
    button10 = new java.awt.Button();
    button9 = new java.awt.Button();
    button14 = new java.awt.Button();
    button15 = new java.awt.Button();

    mainPanel.setName("mainPanel"); // NOI18N

    org.jdesktop.application.ResourceMap resourceMap =
    org.jdesktop.application.Application.getInstance(Ptyxiakh.PtyxiakhApp.class).getContext().getResourceMap(PtyxiakhView.class);
    button1.setLabel(resourceMap.getString("button1.label")); // NOI18N
```



```
button1.setName("button1"); // NOI18N
button1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button1ActionPerformed(evt);
    }
});
```

```
button2.setLabel(resourceMap.getString("button2.label")); // NOI18N
button2.setName("button2"); // NOI18N
button2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button2ActionPerformed(evt);
    }
});
```

```
textArea1.setName("textArea1"); // NOI18N
```

```
button3.setActionCommand(resourceMap.getString("button3.actionCommand")); //
NOI18N
```

```
button3.setLabel(resourceMap.getString("button3.label")); // NOI18N
button3.setName("button3"); // NOI18N
button3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button3ActionPerformed(evt);
    }
});
```

```
button4.setActionCommand("thorivos");
button4.setLabel(resourceMap.getString("button4.label")); // NOI18N
button4.setName("button4"); // NOI18N
button4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button4ActionPerformed(evt);
    }
});
```

```
button5.setLabel(resourceMap.getString("button5.label")); // NOI18N
button5.setName("button5"); // NOI18N
button5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button5ActionPerformed(evt);
    }
});
```

```
button6.setLabel(resourceMap.getString("button6.label")); // NOI18N
```





```
button6.setName("button6"); // NOI18N
button6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button6ActionPerformed(evt);
    }
});

button7.setLabel(resourceMap.getString("button7.label")); // NOI18N
button7.setName("button7"); // NOI18N
button7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button7ActionPerformed(evt);
    }
});

button8.setLabel(resourceMap.getString("button8.label")); // NOI18N
button8.setName("button8"); // NOI18N
button8.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button8ActionPerformed(evt);
    }
});

checkbox1.setLabel(resourceMap.getString("checkbox1.label")); // NOI18N
checkbox1.setName("checkbox1"); // NOI18N

checkbox2.setLabel(resourceMap.getString("checkbox2.label")); // NOI18N
checkbox2.setName("checkbox2"); // NOI18N

checkbox3.setLabel(resourceMap.getString("checkbox3.label")); // NOI18N
checkbox3.setName("checkbox3"); // NOI18N

checkbox4.setLabel(resourceMap.getString("checkbox4.label")); // NOI18N
checkbox4.setName("checkbox4"); // NOI18N

checkbox5.setLabel(resourceMap.getString("checkbox5.label")); // NOI18N
checkbox5.setName("checkbox5"); // NOI18N

button11.setLabel(resourceMap.getString("button11.label")); // NOI18N
button11.setName("button11"); // NOI18N
button11.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button11ActionPerformed(evt);
    }
});
```



```
button12.setLabel(resourceMap.getString("button12.label")); // NOI18N
button12.setName("button12"); // NOI18N
button12.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button12ActionPerformed(evt);
    }
});

button13.setLabel(resourceMap.getString("button13.label")); // NOI18N
button13.setName("button13"); // NOI18N
button13.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button13ActionPerformed(evt);
    }
});

label1.setForeground(resourceMap.getColor("label1.foreground")); // NOI18N
label1.setName("label1"); // NOI18N
label1.setText(resourceMap.getString("label1.text")); // NOI18N

button16.setLabel(resourceMap.getString("All_ontology_content.label")); //
NOI18N
button16.setName("All_ontology_content"); // NOI18N
button16.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button16ActionPerformed(evt);
    }
});

button17.setLabel(resourceMap.getString("button17.label")); // NOI18N
button17.setName("button17"); // NOI18N
button17.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button17ActionPerformed(evt);
    }
});

checkbox6.setLabel(resourceMap.getString("checkbox6.label")); // NOI18N
checkbox6.setName("checkbox6"); // NOI18N

button18.setForeground(resourceMap.getColor("button18.foreground")); //
NOI18N
button18.setLabel(resourceMap.getString("button18.label")); // NOI18N
button18.setName("button18"); // NOI18N
button18.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```



```
        button18ActionPerformed(evt);
    }
});

    javax.swing.GroupLayout mainPanelLayout = new
    javax.swing.GroupLayout(mainPanel);
    mainPanel.setLayout(mainPanelLayout);
    mainPanelLayout.setHorizontalGroup(

mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(mainPanelLayout.createSequentialGroup()

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addGroup(mainPanelLayout.createSequentialGroup()
                    .addComponent(textArea1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 381,
    javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(30, 30, 30)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addGroup(mainPanelLayout.createSequentialGroup()
                    .addComponent(checkbox3,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(checkbox6,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
                .addComponent(checkbox2,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(checkbox1,
    javax.swing.GroupLayout.PREFERRED_SIZE,
```



```
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
    .addComponent(button13,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
    .addComponent(label1,  
javax.swing.GroupLayout.PREFERRED_SIZE, 172,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
    .addGroup(mainPanelLayout.createSequentialGroup()  
    .addComponent(button11,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
    .addComponent(button12,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
    .addComponent(checkbox5,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
    .addComponent(checkbox4,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
    .addComponent(button17,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))  
    .addGroup(mainPanelLayout.createSequentialGroup()  
  
    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignme  
nt.LEADING)  
    .addGroup(mainPanelLayout.createSequentialGroup()  
    .addComponent(button2,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
  
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
    .addComponent(button5,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```



```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(button6,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(mainPanelLayout.createSequentialGroup()
    .addComponent(button1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(125, 125, 125)
    .addComponent(button4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(mainPanelLayout.createSequentialGroup()
    .addGap(167, 167, 167)
    .addComponent(button18,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addGap(20, 20, 20)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
    .addComponent(button8,
javax.swing.GroupLayout.PREFERRED_SIZE, 131,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGroup(mainPanelLayout.createSequentialGroup()
    .addComponent(button7,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(77, 77, 77)
    .addComponent(button16,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))))))
    .addGroup(mainPanelLayout.createSequentialGroup()
    .addGap(93, 93, 93)
    .addComponent(button3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap(267, Short.MAX_VALUE))
);
```



```
mainPanelLayout.setVerticalGroup(

mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(mainPanelLayout.createSequentialGroup())

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(mainPanelLayout.createSequentialGroup())

            .addContainerGap()

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(button1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(button4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(button8,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(button2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(button5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(button6,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(button7,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
```



```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(button3,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(button18,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createSequentialGroup()
    .addContainerGap()
    .addComponent(button16,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(22, 22, 22))

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(textArea1,
        javax.swing.GroupLayout.PREFERRED_SIZE, 255,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addComponent(label1,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(4, 4, 4)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addComponent(checkbox1,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```



```
.addComponent(checkbox2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(checkbox3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addComponent(checkbox6,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(checkbox4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(checkbox5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(button12,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(button11,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(button13,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```





```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(button17,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addGap(13, 13, 13))
);

menuBar.setName("menuBar"); // NOI18N

fileMenu.setText(resourceMap.getString("fileMenu.text")); // NOI18N
fileMenu.setName("fileMenu"); // NOI18N

javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(ptyxiakh.PtyxiakhApp.class).getCon
text().getActionMap(PtyxiakhView.class, this);
exitMenuItem.setAction(actionMap.get("quit")); // NOI18N
exitMenuItem.setName("exitMenuItem"); // NOI18N
fileMenu.add(exitMenuItem);

menuBar.add(fileMenu);

helpMenu.setText(resourceMap.getString("helpMenu.text")); // NOI18N
helpMenu.setName("helpMenu"); // NOI18N

aboutMenuItem.setAction(actionMap.get("showAboutBox")); // NOI18N
aboutMenuItem.setName("aboutMenuItem"); // NOI18N
helpMenu.add(aboutMenuItem);

menuBar.add(helpMenu);

statusPanel.setName("statusPanel"); // NOI18N

statusPanelSeparator.setName("statusPanelSeparator"); // NOI18N

statusMessageLabel.setName("statusMessageLabel"); // NOI18N

statusAnimationLabel.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
statusAnimationLabel.setName("statusAnimationLabel"); // NOI18N

progressBar.setName("progressBar"); // NOI18N

button10.setLabel(resourceMap.getString("button10.label")); // NOI18N
button10.setName("button10"); // NOI18N
```



```
button10.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button10ActionPerformed(evt);
    }
});

button9.setLabel(resourceMap.getString("button9.label")); // NOI18N
button9.setName("button9"); // NOI18N
button9.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button9ActionPerformed(evt);
    }
});

button14.setLabel(resourceMap.getString("button14.label")); // NOI18N
button14.setName("button14"); // NOI18N
button14.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button14ActionPerformed(evt);
    }
});

button15.setActionCommand(resourceMap.getString("button15.actionCommand")); //
NOI18N
button15.setLabel(resourceMap.getString("button15.label")); // NOI18N
button15.setName("button15"); // NOI18N
button15.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        button15ActionPerformed(evt);
    }
});

javax.swing.GroupLayout statusPanelLayout = new
javax.swing.GroupLayout(statusPanel);
statusPanel.setLayout(statusPanelLayout);
statusPanelLayout.setHorizontalGroup(

statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(statusPanelLayout.createSequentialGroup()
        .add(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .add(statusPanelLayout.createSequentialGroup()
                .add(statusMessageLabel)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 893, Short.MAX_VALUE)
            )
        )
    )
);
```



```
.addComponent(statusAnimationLabel)
    .addContainerGap())
    .addGroup(statusPanelLayout.createSequentialGroup())
        .addGap(35, 35, 35)
        .addComponent(button14, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(button10, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(button9, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
        .addGroup(statusPanelLayout.createSequentialGroup())
            .addGap(166, 166, 166)

    .addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
statusPanelLayout.createSequentialGroup())
            .addComponent(progressBar,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
            .addComponent(statusPanelSeparator,
javax.swing.GroupLayout.DEFAULT_SIZE, 395, Short.MAX_VALUE)))
        .addGroup(statusPanelLayout.createSequentialGroup())
            .addGap(20, 20, 20)
            .addComponent(button15,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap()))
);
statusPanelLayout.setVerticalGroup(

statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
```



```
.addGroup(statusPanelLayout.createSequentialGroup())
    .addComponent(statusPanelSeparator,
javax.swing.GroupLayout.PREFERRED_SIZE, 2,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 33,
Short.MAX_VALUE)
    .addComponent(progressBar,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(statusPanelLayout.createSequentialGroup())
        .addContainerGap()

.addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
    .addGroup(statusPanelLayout.createSequentialGroup())
        .addComponent(button15,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap())

.addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
statusPanelLayout.createSequentialGroup())
        .addComponent(button14,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 17,
Short.MAX_VALUE)

.addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.BASELINE)
    .addComponent(statusMessageLabel)
    .addComponent(statusAnimationLabel))
    .addGap(3, 3, 3))
    .addGroup(statusPanelLayout.createSequentialGroup())

.addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
    .addComponent(button10,
javax.swing.GroupLayout.PREFERRED_SIZE,
```



```
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
    .addComponent(button9,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
    .addContainerGap(20, Short.MAX_VALUE))))  
);  
  
setComponent(mainPanel);  
setMenuBar(menuBar);  
setStatusBar(statusPanel);  
} // </editor-fold>  
  
// Button pou dinei apotelesma result olo to perioxomeno ths ontologias  
// To sygkekrimeno button exei kathara ekpaideytiko rolo kai den exei oysiastiko rolo  
// gia to systhma ths ptyxiakhs  
  
private void button1ActionPerformed(java.awt.event.ActionEvent evt) {  
// TODO add your handling code here:  
  
    r.load(IRReasoner.OWL, GEOntURI);  
    r.setSerializationMode(IRReasoner.DEFAULT);  
    r.setBaseNamespace(ruleURI);  
    r.setNamespacePrefix("w", ruleURI);  
    r.setNamespacePrefix("g", GEOntURI);  
  
    ir.load(IRReasoner.OWL, INOURI);  
    ir.setSerializationMode(IRReasoner.DEFAULT);  
    ir.setBaseNamespace(ruleURI3);  
    ir.setNamespacePrefix("i", ruleURI3);  
    ir.setNamespacePrefix("co", INOURI);  
  
    try {  
  
        // Eisagwgh neas gnwshs se property poy den yparxei stis ontologies GEOnt  
& INO  
        // kai emfanish mono twn properties pou exoun zitimhei sta facts  
        // mporoyme dhl na kanoume query mono tis idiothtes didaskei &  
exei_epilexthei  
  
        //ir.classifyTaxonomy(null);  
        ir.tell("fact ic is has(i:Noisy,i:Saloni);");  
        ir.tell("fact ic2 is has(i:Noisy,i:Sala);");  
        ir.tell("fact id is has(i:quiet,i:Ypnodwmatio);");
```



```
r.tell("fact e is eksetazetai(w:eksetash_Fysikh_metria,w:Georgiou);");
r.tell("fact e2 is eksetazetai(w:eksetash_Fysikh_metria,w:Ioannou);");
r.tell("fact e3 is eksetazetai(w:eksetash_Fysikh_dyskolh,w:Petrou);");
r.tell("fact f is epilegei(w:Mathimatika,w:Remalhs);");

r.tell("fact c is didaskei(w:Vranas,w:Istoria);");
r.tell("fact d is exei_epilexthei(g:Istoria,g:Kolokythas);");
r.tell("fact a is didaskei(w:John,w:Gymnastikh);");
r.tell("fact b is exei_epilexthei(w:Gymnastikh,w:Nil);");

//Erwthma sthn GEOnt me logikh synthiki kai filtro mesw ths String kai
energeia ask

String result4 = r.ask("query q is didaskei(?x,?y) and
exei_epilexthei(?y,?z);");

//Erwthma sthn context me logikh synthiki kai filtro mesw ths String kai
energeia ask

String result6 = ir.ask("query q is has(?x,?z) and (?x=Noisy);");

//Erwthma sthn GEOnt me logikh synthiki kai filtro mesw ths Answer kai
energeia ask1

Answer result8 = r.ask1("query q is eksetazetai(?x,?y) and
(?x=eksetash_Fysikh_metria);"); // 6

//Erwthma sthn GEOnt xwris logikh synthiki kai filtro mesw ths Answer kai
energeia ask1
// Prosoxh h endeiksh w: mpainei wste na mh xasoyme ta apotelesmata poy
exoyrn
// kataxwrithei ws instances sthn ontologia

Answer result9 = r.ask1("query q is w:exei_epilexthei(?x,?z)");

//Erwthma sthn context me logikh synthiki kai filtro mesw ths Answer kai
energeia ask1

Answer result10 = ir.ask1("query q is i:has(?x,?z) and (?x=Noisy);");
String resulti5 = ir.run();
String resulti7 = r.run();

// Diadikasia Diaxwrisismou twrn apotelesmatwn

ArrayList actions = new ArrayList(result9.getBindings());
```



```
        for (Iterator it = actions.iterator(); it.hasNext();) {
            HashMap hm = (HashMap) it.next();
            String segment = hm.get("?x").toString();
            segment = segment.substring(segment.indexOf("#") + 1);
            String service = hm.get("?z").toString();
            service = service.substring(service.indexOf("#") + 1);
        }

        if (result6 == null)
        {
            System.out.println("The query returns false!");
        }
        else
        {
            textArea1.setText("Fourth session" + "\nApo GEOnt result4:\n" + result4 +
                "\nApo context result6:\n" + result6 + "\nApo GEOnt & context result8:\n" + result8
                + "\nApo GEOnt result9:\n" + result9 + "\nApo GEOnt result10:\n" + result10);
        }

    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

// Button pou dinei apotelesma sto query tou poios mathitis exei, poio mathima
// To sygkekrimeno button exei kathara ekpaideytiko rolo kai den exei oysiastiko rolo
// gia to systhma ths ptyxiakhs

private void button2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

    r.load(IRReasoner.OWL, GEOntURI);
    r.setSerializationMode(IRReasoner.DEFAULT);
    r.setBaseNamespace(ruleURI);
    r.setNamespacePrefix("w", ruleURI);
    r.setNamespacePrefix("g", GEOntURI);

    ir.load(IRReasoner.OWL, INOURI);
    ir.setSerializationMode(IRReasoner.DEFAULT);
    ir.setBaseNamespace(ruleURI3);
    ir.setNamespacePrefix("i", ruleURI3);
}
```



```
ir.setNamespacePrefix("co", INOURI);

try {

    // Eisagwgh neas gnwshs se property poy den yparxei stis ontologies GEOnt
& INO
    // kai emfanish mono twn properties pou exoun zitithei sta facts
    // mporoyme dhl na kanoume query mono tis idiothtes didaskei &
exei_epilexthei

    ir.tell("fact ic is has(i:Noisy,i:Saloni);");
    ir.tell("fact ic2 is has(i:Noisy,i:Sala);");
    ir.tell("fact id is has(i:quiet,i:Ypnodwmatio);");
    r.tell("fact e is eksetazetai(w:eksetash_Fysikh_metria,w:Georgiou);");
    r.tell("fact e2 is eksetazetai(w:eksetash_Fysikh_metria,w:Ioannou);");
    r.tell("fact e3 is eksetazetai(w:eksetash_Fysikh_dyskolh,w:Petrou);");
    r.tell("fact f is epilegei(w:Mathimatika,w:Remalhs);");

    r.tell("fact c is didaskei(w:Vranas,w:Istoria);");
    r.tell("fact d is exei_epilexthei(g:Istoria,g:Kolokythas);");
    r.tell("fact a is didaskei(w:John,w:Gymnastikh);");
    r.tell("fact b is exei_epilexthei(w:Gymnastikh,w:Nil);");

    String tempString = "";
    //Erwthma sthn GEOnt me logikh synthiki kai filtro mesw ths String kai
energeia ask

    String result4 = r.ask("query q is didaskei(?x,?y) and
exei_epilexthei(?y,?z);");

    //Erwthma sthn context me logikh synthiki kai filtro mesw ths String kai
energeia ask

    String result6 = ir.ask("query q is has(?x,?z) and (?x=Noisy);");

    //Erwthma sthn GEOnt me logikh synthiki kai filtro mesw ths Answer kai
energeia ask1

    Answer result8 = r.ask1("query q is eksetazetai(?x,?y) and
(?x=eksetash_Fysikh_metria);"); // 6

    //Erwthma sthn GEOnt xwris logikh synthiki kai filtro mesw ths Answer kai
energeia ask1
    // Prosoxh h endeiksh w: mpainei wste na mh xasoyme ta apotelesmata poy
exoyrn
    // kataxwritheiw ws instances sthn ontologia
```





```
Answer result9 = r.ask1("query q is w:exei_epilexthei(?x,?z)");

//Erwthma sthn context me logikh synthiki kai filtro mesw ths Answer kai
energeia ask1

Answer result10 = ir.ask1("query q is i:has(?x,?z) and (?x=Noisy);");
String resulti5 = ir.run();
String resulti7 = r.run();

// Diadikasia Diaxwrismou tw n apotelesmatwn

ArrayList actions = new ArrayList(result9.getBindings());
textArea1.setText(" ");
    for (Iterator it = actions.iterator(); it.hasNext();) {
        HashMap hm = (HashMap) it.next();
        String segment = hm.get("?x").toString();
        segment = segment.substring(segment.indexOf("#") + 1);
        String service = hm.get("?z").toString();
        service = service.substring(service.indexOf("#") + 1);
tempString = textArea1.getText() + "\n";
        textArea1.setText( tempString + "\n Mathima: " + segment +
"\n Mathitis: " + service);
    }

    if (result6 == null)
    {
        System.out.println("The query returns false!");
    }
    else
    {
        }

}
catch (Exception e)
{
    e.printStackTrace();
}
}

// Button start to opoio ksekina thn eksetash
```



```
private void button3ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

    started=true;
    thoryvos=false;
    initialize(evt);
    try {

        String tempString = "";

        //Nea Gnwsh
        facts(evt);

        // Erwthma
        Answer result_exam = r.ask1("query q is w:eksetazetai(?y,?x) and
(?x=eksetash_Fysikh_dyskolh);");

        // To apotelesma exei kai th Nea gnwsh alla kai thn hdh enswmatwmenh sthn
ontologia
        // Apo thn ne agnws th emfanisthei to fact e3, enw apo thn ontologia th
emfanisthei
        // h eksetash_Fysikh_dyskolh tou Georgiou

        ArrayList actions = new ArrayList(result_exam.getBindings());
        textArea1.setText(" ");
        for (Iterator it = actions.iterator(); it.hasNext();) {
            HashMap hm = (HashMap) it.next();
            String segment = hm.get("?x").toString();
            segment = segment.substring(segment.indexOf("#") + 1);
            String service = hm.get("?y").toString();
            service = service.substring(service.indexOf("#") + 1);
            tempString = textArea1.getText() + "\n";
            textArea1.setText( tempString + "\n Mathima: " + segment +
"\n Mathitis: " + service);
        }

    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

// Button HXOS - patietai meta apo ta buttons Thoryvos ;h siganos
// wste na ayksomeiwsei to apotelesma ths eksetashs
```



```
private void button4ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

    try {

        if (started){

            String tempString = "";
            facts(evt);

            // Edw to krithrio gia na ektelesthei symperasmos einai
            // an yparxei thoryvos ;h oxi
            // Edw prosomoiwnetai me to pathma enos koumpiou arxika
            // kai epeita kaleitai h royтина button4ActionPerformed (HXOS)

            if (thoryvos){

                // Erwthma
                Answer result_exam = r.ask1("query q is w:eksetazetai(?y,?x) and
(?x=eksetash_Fysikh_eykolh);");

                ArrayList actions = new ArrayList(result_exam.getBindings());
                textArea1.setText(" ");

                for (Iterator it = actions.iterator(); it.hasNext();) {
                    HashMap hm = (HashMap) it.next();
                    String segment = hm.get("?x").toString();
                    segment = segment.substring(segment.indexOf("#") + 1);
                    String service = hm.get("?y").toString();
                    service = service.substring(service.indexOf("#") + 1);
                    tempString = textArea1.getText() + "\n";
                    textArea1.setText( tempString + "\n Mathima: " + segment +
"\n Mathitis: " + service);
                }
            }else if (! thoryvos){

                // Erwthma
                Answer result_exam = r.ask1("query q is w:eksetazetai(?y,?x) and
(?x=eksetash_Fysikh_dyskolh);");

                ArrayList actions = new ArrayList(result_exam.getBindings());
                textArea1.setText(" ");
                for (Iterator it = actions.iterator(); it.hasNext();) {
                    HashMap hm = (HashMap) it.next();
```



```
        String segment = hm.get("?x").toString();
        segment = segment.substring(segment.indexOf("#") + 1);
        String service = hm.get("?y").toString();
        service = service.substring(service.indexOf("#") + 1);
        tempString = textArea1.getText() + "\n";
        textArea1.setText( tempString + "\n Mathima: " + segment +
"\n Mathitis: " + service);
    }
}
}
else if (! started){
    textArea1.setText( "\n Den exei ksekinhsei kapoia eksetash \n");
}

}
catch (Exception e)
{
    e.printStackTrace();
}

}

// Button Thoryvos - meta prepei na patithei to button HXOS

private void button5ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    thoryvos=true;
}

// Button Siganos - meta prepei na patithei to button HXOS

private void button6ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    thoryvos=false;
}

// Button poy deixnei oti perase sto Saloni

private void button8ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

    try {

        if (started){
```



```
thoryvos=true;

String tempString = "";

facts(evt);

// Erwthma
Answer result_xwros = ir.ask1("query q is i:has(?x,?y) and (?x=Noisy);");
Answer result_questions = r.ask1("query q10 is w:has(?k,!l) and (?k=done);");

// r.tell("fact q1 is (w:done,w:question1);");
// Η result_xwros θα έχει όλους εκείνους τους individuals που ικανοποιούν
// το erwthma και αφορούν το xwro στον οποίο έχει μετακινήσει ο μαθητής
// Στο επόμενο τύπωνοyme τα αποτελέσματα της result_xwros αφαιρώντας
// τη URI

ArrayList actions2 = new ArrayList(result_xwros.getBindings());
ArrayList actions3 = new ArrayList(result_questions.getBindings());

//textArea1.setText(" ");
for (Iterator it = actions2.iterator(); it.hasNext();) {
    HashMap hm = (HashMap) it.next();
    String segment = hm.get("?x").toString();
    segment = segment.substring(segment.indexOf("#") + 1);
    String service = hm.get("?y").toString();
    service = service.substring(service.indexOf("#") + 1);
    tempString = textArea1.getText() + "\n";

    // Στο σημείο αυτό εξηγούμε την πληροφορία από την ontologia
    // Η service έχει την τιμή της ?y η οποία είναι ;h το Saloni
    // ;h το ypnodwmatio
    // Επειτα ayth η μεταβλητή θα συγκριθεί και αν η συγκρίση είναι true
    // θα δώσει την κατάλληλη δυσκολία στην εκτέλεση

    xwros=service;

    //katagrafh sthn othoni toy xwrou pou brisketai twra o mathitis
    textArea1.setText( tempString + "\n Apo Hxo ? " + segment +
"\n Xwros: " + service + "\n xwros: " + xwros);
}

for (Iterator it = actions3.iterator(); it.hasNext();) {
    HashMap hm = (HashMap) it.next();
    String segment = hm.get("?k").toString();
    segment = segment.substring(segment.indexOf("#") + 1);
```



```
String service = hm.get("?l").toString();
service = service.substring(service.indexOf("#") + 1);
tempString = textArea1.getText() + "\n";

// Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
// H service exei thn timh ths ?y h opoia einai ;h to Saloni
// ;h to ypnodwmatio
// Epeita ayth h metavlhth tha sygkrithei kai an h sygkrish einai true
// tha dwsei thn katallhlyh dyskolia sthn eksetash

if (service.equals("question_an")){
    checkbox1.setState(true);
}
if (service.equals("question_bn")){
    checkbox2.setState(true);
}
if (service.equals("question_cn")){
    checkbox3.setState(true);
}
if (service.equals("question_dn")){
    checkbox4.setState(true);
}
if (service.equals("question_en")){
    checkbox5.setState(true);
}

}
boolean temp=false;
// H xwros einai global metablhth kai periexei pleon ton xwro ston opoio
vrisketai
// o mathitis
// kalw th royтина symerasmou, gia na vgalw symperasma ean h eksetash tha
dyskolepsei ;h oxi

symperasmos_dwmatiou(evt);

}
else if (! started){
    textArea1.setText( "\n Den exei ksekinhsei kapoia eksetash \n");
}
}
catch (Exception e)
{
    String tempString = textArea1.getText() + "\n";
    textArea1.setText( tempString + "\n kamia apantsh akoma \n");
}
}
```



```
}  
  
// Button poy deixnei oti perase sto Ypnodwmatio  
  
private void button7ActionPerformed(java.awt.event.ActionEvent evt) {  
// TODO add your handling code here:  
  
    try {  
        thoryvos=false;  
        if (started){  
  
            String tempString = "";  
  
            facts(evt);  
  
            // Erwthma  
            Answer result_xwros = ir.ask1("query q is i:has(?x,?y) and (?x=Quiet);");  
            Answer result_questions = r.ask1("query q10 is w:has(?k,?l) and (?k=done);");  
  
            // H result_xwros tha exei olous ekeinous toyw individuals poy ikanopoioun  
            // to erwthma kai aforoun to xwro ston opoio exei metakinitheo o mathitis  
            // Sto epomeno tywnoyme ta apotelesmata ths result_xwros afairontas  
            // th URI  
  
            ArrayList actions2 = new ArrayList(result_xwros.getBindings());  
            ArrayList actions3 = new ArrayList(result_questions.getBindings());  
            for (Iterator it = actions2.iterator(); it.hasNext();) {  
                HashMap hm = (HashMap) it.next();  
                String segment = hm.get("?x").toString();  
                segment = segment.substring(segment.indexOf("#") + 1);  
                String service = hm.get("?y").toString();  
                service = service.substring(service.indexOf("#") + 1);  
                tempString = textArea1.getText() + "\n";  
  
                // Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia  
                // H service exei thn timh ths ?y h opoia einai to dwmatio me thoryvo  
quiet  
                // Epeita ayth h metavlhth tha sygkriethei kai an h sygkrish einai true  
                // tha dwsei thn katallhlyh dyskolia sthn eksetash  
  
                xwros=service;  
  
                textArea1.setText( tempString + "\n Apo Hxo ? " + segment +  
"\n Xwros: " + service + "\n xwros: " + xwros);  
            }  
        }  
    }  
}
```



```
for (Iterator it = actions3.iterator(); it.hasNext();) {
    HashMap hm = (HashMap) it.next();
    String segment = hm.get("?k").toString();
    segment = segment.substring(segment.indexOf("#") + 1);
    String service = hm.get("?l").toString();
    service = service.substring(service.indexOf("#") + 1);
tempString = textArea1.getText() + "\n";

    // Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
    // H service exei thn timh ths ?y h opoia einai ;h to Saloni
    // ;h to ypnodwmatio
    // Epeita ayth h metavlth tha sygkrithe kai an h sygkrish einai true
    // tha dwsei thn katallhlyh dyskolia sthn eksetash

    if (service.equals("question_aq")){
        checkbox1.setState(true);
    }
    if (service.equals("question_bq")){
        checkbox2.setState(true);
    }
    if (service.equals("question_cq")){
        checkbox3.setState(true);
    }
    if (service.equals("question_dq")){
        checkbox4.setState(true);
    }
    if (service.equals("question_eq")){
        checkbox5.setState(true);
    }
}

boolean temp=false;

// H xwros einai global metablth kai periexei pleon ton xwro ston opoio
vrisketai
// o mathitis
// kalw th royтина symerasmou, gia na vgalw symperasma ean h eksetash tha
dyskolepsei ;h oxi

symperasmos_dwmatiou(evt);

}
else if (! started){
    textArea1.setText( "\n Den exei ksekinhsei kapoia eksetash \n");
}
}
```





```
catch (Exception e)
{
    String tempString = textArea1.getText() + "\n";
    textArea1.setText( tempString + "\n kamia apanthsh akoma \n");
}
}

private void Noisy_Place_A_Senariou (java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    try {

        if (started){

            String tempString = "";

            facts(evt);

            // Erwthma
            Answer result_xwros = ir.ask1("query q is i:has(?x,?y) and (?x=Noisy);");

            // H result_xwros tha exei olous ekeinous toyw individuals poy ikanopoioun
            // to erwthma kai aforoun to xwro ston opoio exei metakinitheo o mathitis
            // Sto epomeno typwnoyme ta apotelesmata ths result_xwros afairontas
            // th URI

            ArrayList actions2 = new ArrayList(result_xwros.getBindings());
            for (Iterator it = actions2.iterator(); it.hasNext();) {
                HashMap hm = (HashMap) it.next();
                String segment = hm.get("?x").toString();
                segment = segment.substring(segment.indexOf("#") + 1);
                String service = hm.get("?y").toString();
                service = service.substring(service.indexOf("#") + 1);
                tempString = textArea1.getText() + "\n";

                // Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
                // H service exei thn timh ths ?y h opoia einai ;h to Saloni
                // ;h to ypnodwmatio
                // Epeita ayth h metavlhth tha sygkritheo kai an h sygkrish einai true
                // tha dwsei thn katallhlyh dyskolia sthn eksetash

                xwros=service;

                //katagrafh sthn othoni toy xwrou poy brisketai twra o mathitis
            }
        }
    }
}
```



```
        textArea1.setText( tempString + "\n Apo Hxo ? " + segment +
"\n Xwros: " + service + "\n xwros: " + xwros);
    }
    boolean temp=false;
    // H xwros einai global metablth kai periexei pleon ton xwro ston opoio
vrisketai
    // o mathitis
    // kalw th royтина symerasμου, gia na vgalw symperasma ean h eksetash tha
dyskolepsei ;h oxi

    symperasmos_dwmatiou(evt);

    }
    else if (! started){
        textArea1.setText( "\n Den exei ksekinhsei kapoia eksetash \n");
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
}

private void Quiet_Place_A_Senariou (java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    try {

        if (started){

            String tempString = "";

            facts(evt);

            // Erwthma
            Answer result_xwros = ir.ask1("query q is i:has(?x,?y) and (?x=Quiet);");

            // H result_xwros tha exei olous ekeinous toyw individuals poy ikanopoioun
            // to erwthma kai aforoun to xwro ston opoio exei metakinitheo mathitis
            // Sto epomeno typwnoyme ta apotelesmata ths result_xwros afairontas
            // th URI

            ArrayList actions2 = new ArrayList(result_xwros.getBindings());
            for (Iterator it = actions2.iterator(); it.hasNext();) {
                HashMap hm = (HashMap) it.next();
```



```
String segment = hm.get("?x").toString();
segment = segment.substring(segment.indexOf("#") + 1);
String service = hm.get("?y").toString();
service = service.substring(service.indexOf("#") + 1);
tempString = textArea1.getText() + "\n";

// Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
// H service exei thn timh ths ?y h οποια einai to dwmatio me thoryvo
quiet
// Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
// tha dwsei thn katallhlyh dyskolia sthn eksetash

xwros=service;

textArea1.setText( tempString + "\n Apo Hxo ? " + segment +
"\n Xwros: " + service + "\n xwros: " + xwros);
}
boolean temp=false;

// H xwros einai global metablth kai perixei pleon ton xwro ston οποιο
vrisketai
// o mathitis
// kalw th roytina symerasμου, gia na vgalw symperasma ean h eksetash tha
dyskolepsei ;h oxi

symperasmos_dwmatiου(evt);

}
else if (! started){
textArea1.setText( "\n Den exei ksekinhsei kapoia eksetash \n");
}
}
}
catch (Exception e)
{
e.printStackTrace();
}
}

//button of A_Senario

private void button10ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
try{
button14.setEnabled(false);
button15.setEnabled(false);
button9.setEnabled(false);
```



```
button3.setEnabled(false);
checkbox6.setVisible(false);
label1.setText("Α' Εκπαιδευτικό Σενάριο");
    String tempString = "";

arrayOfIntegers = new double[N]; // allocates memory for N doubles

Reader r = new BufferedReader(new FileReader("c:\\contextdata.txt"));
    StreamTokenizer stok = new StreamTokenizer(r);
    stok.parseNumbers();
    double sum = 0;
    int counter=0;
    int inverse=0;
    stok.nextToken();
    while (stok.ttype != StreamTokenizer.TT_EOF) {
        if (stok.ttype == StreamTokenizer.TT_NUMBER){
            if(counter < N){
                arrayOfIntegers[counter]=stok.nval;
                if (counter%32 == 0 && counter !=0){
                    inverse=32;
                    tempString = textArea1.getText() + "\n";
                    textArea1.setText(tempString + "\n----- next row -----
\n");
                    tempString = textArea1.getText() + "\n";
                    textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Scenario number: " + arrayOfIntegers[((counter-inverse)/32)*32
+(counter-inverse)%32]);
                    inverse=inverse-1;tempString = textArea1.getText() + "\n";
                    textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Repetition number: " + arrayOfIntegers[((counter-inverse)/32)*32
+(counter-inverse)%32]);
                    inverse=inverse-1;tempString = textArea1.getText() + "\n";
                    textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Time (s): " + arrayOfIntegers[((counter-inverse)/32)*32 +(counter-
inverse)%32]);
                    inverse=inverse-1;tempString = textArea1.getText() + "\n";
                    //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Position:DisplayDown: " + arrayOfIntegers[(counter-
inverse)%32]);
                    inverse=inverse-1;tempString = textArea1.getText() + "\n";
                    //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Position:DisplayUp: " + arrayOfIntegers[(counter-
inverse)%32]);
                    inverse=inverse-1;tempString = textArea1.getText() + "\n";
```



```
//textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Position:AntennaDown: " + arrayOfIntegers[(counter-
inverse)%32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Position:AntennaUp: " + arrayOfIntegers[(counter-
inverse)%32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Position:SidewaysRight: " + arrayOfIntegers[(counter-
inverse)%32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Position:SidewaysLeft: " + arrayOfIntegers[(counter-
inverse)%32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Stability:Stable: " + arrayOfIntegers[(counter-inverse)%32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Stability:Unstable: " + arrayOfIntegers[(counter-
inverse)%32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Placement:AtHand: " + arrayOfIntegers[(counter-
inverse)%32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Light:EU: " + arrayOfIntegers[(counter-inverse)%32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Light:USA: " + arrayOfIntegers[(counter-
inverse)%32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Light:Bright: " + arrayOfIntegers[(counter-
inverse)%32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Light:Normal: " + arrayOfIntegers[(counter-
inverse)%32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Light:Dark: " + arrayOfIntegers[(counter-
inverse)%32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
```



```
//textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Light:Natural: " + arrayOfIntegers[(counter-
inverse)%32]);
inverse=inverse-1;tempString = textArea1.getText() + "\n";
//textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Light:TotalDarkness: " + arrayOfIntegers[(counter-
inverse)%32]);
inverse=inverse-1;tempString = textArea1.getText() + "\n";
//textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Temperature:Hot: " + arrayOfIntegers[(counter-
inverse)%32]);
inverse=inverse-1;tempString = textArea1.getText() + "\n";
//textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Temperature:Warm: " + arrayOfIntegers[(counter-
inverse)%32]);
inverse=inverse-1;tempString = textArea1.getText() + "\n";
//textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Temperature:Cool: " + arrayOfIntegers[(counter-
inverse)%32]);
inverse=inverse-1;tempString = textArea1.getText() + "\n";
//textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Temperature:Cold: " + arrayOfIntegers[(counter-
inverse)%32]);
inverse=inverse-1;tempString = textArea1.getText() + "\n";
//textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Humidity:Humid: " + arrayOfIntegers[(counter-
inverse)%32]);
inverse=inverse-1;tempString = textArea1.getText() + "\n";
//textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Humidity:Normal: " + arrayOfIntegers[(counter-
inverse)%32]);
inverse=inverse-1;tempString = textArea1.getText() + "\n";
//textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Humidity:Dry: " + arrayOfIntegers[(counter-
inverse)%32]);
inverse=inverse-1;tempString = textArea1.getText() + "\n";
textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:SoundPressure:Silent: " + arrayOfIntegers[((counter-
inverse)/32)*32 +(counter-inverse)%32]);
inverse=inverse-1;tempString = textArea1.getText() + "\n";
textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:SoundPressure:Modest: " + arrayOfIntegers[((counter-
inverse)/32)*32 +(counter-inverse)%32]);
inverse=inverse-1;tempString = textArea1.getText() + "\n";
```



```
        textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:SoundPressure:Loud: " + arrayOfIntegers[(((counter-
inverse)/32)*32 +(counter-inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " UserAction:Movement:Walking: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " UserAction:Movement:WalkingFast: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " UserAction:Movement:Running: " + arrayOfIntegers[(counter-
inverse)%32]);
        tempString = textArea1.getText() + "\n";
        textArea1.setText(tempString + "----- end of row -----
\n");
        textArea1.setText(tempString + "column : " + (((counter-
inverse)/32)*32 +(counter-inverse)%32 - 4) + "\n");
        if (arrayOfIntegers[(((counter-inverse)/32)*32 +(counter-inverse)%32 -
4)]>=0.3){
            Noisy_Place_A_Senariou(evt);
        } else if (arrayOfIntegers[(((counter-inverse)/32)*32 +(counter-
inverse)%32 - 4)]< 0.3){
            Quiet_Place_A_Senariou(evt);
        }
    }
    counter=counter+1;
    sum += stok.nval;
}
else
    textArea1.setText("Nonnumber: " + stok.sval);
    stok.nextToken();
}
//textArea1.setText("The file sum is " + sum);
}
catch (Exception e)
{
    e.printStackTrace();
}
}
```



```
private void Place_C_Senariou(java.awt.event.ActionEvent evt, int c) {
    try {

        if (started){

            String tempString = "";
            int lock_counter=0;
            facts(evt);

            // Erwthma
            if (c==1){
                session=1;

                Answer result_xwros = ir.ask1("query q is i:is_it(?x,?y) and (?y=hall_1);");
                Answer result_questions = r.ask1("query q10 is w:has(?k,!l) and (?k=done);");
                ArrayList actions2 = new ArrayList(result_xwros.getBindings());
                ArrayList actions3 = new ArrayList(result_questions.getBindings());

                for (Iterator it = actions2.iterator(); it.hasNext();) {
                    HashMap hm = (HashMap) it.next();
                    String segment = hm.get("?x").toString();
                    segment = segment.substring(segment.indexOf("#") + 1);
                    String service = hm.get("?y").toString();
                    service = service.substring(service.indexOf("#") + 1);
                    tempString = textArea1.getText() + "\n";

                    // Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
                    // H service exei thn timh ths ?y h opoia einai ;h to Saloni
                    // ;h to ypnodwmatio
                    // Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
                    // tha dwsei thn katallhlyh dyskolia sthn eksetash

                    xwros=service;

                    //katagrafh sthn othoni toy xwrou poy brisketai twra o mathitis
                    textArea1.setText( tempString + "\n Room Number " +
                    segment + "\n Room Name: " + service + "\n Room Name: " + xwros);
                }

                for (Iterator it = actions3.iterator(); it.hasNext();) {
                    HashMap hm = (HashMap) it.next();
                    String segment = hm.get("?k").toString();
                    segment = segment.substring(segment.indexOf("#") + 1);
                    String service = hm.get("?l").toString();
                    service = service.substring(service.indexOf("#") + 1);
                    tempString = textArea1.getText() + "\n";
                }
            }
        }
    }
}
```





```
// Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
// H service exei thn timh ths ?y h opoia einai ;h to Saloni
// ;h to ypnodwmatio
// Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
// tha dwsei thn katallhlyh dyskolia sthn eksetash

        if(c==1){
if ((service.equals("question_a1")) && (segment.equals("done")) ){
    checkbox1.setState(true);
    checkbox1.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_b1")) && (segment.equals("done")) ){
    checkbox2.setState(true);
    checkbox2.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_c1")) && (segment.equals("done")) ){
    checkbox3.setState(true);
    checkbox3.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_d1")) && (segment.equals("done")) ){
    checkbox4.setState(true);
    checkbox4.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_e1")) && (segment.equals("done")) ){
    checkbox5.setState(true);
    checkbox5.setEnabled(false);
    lock_counter=lock_counter+1;
}

        if(lock_counter==5){
            checkbox6.setState(true);
            lock_answers(evt);
        }
    }
}

if (c==2){
    session=2;
    if (times_in_hall_2 != 5){
```



```
        times_in_hall_2=times_in_hall_2+1;
    }
    Answer result_xwros = ir.ask1("query q is i:is_it(?x,?y) and (?y=hall_2);");
    Answer result_questions = r.ask1("query q10 is w:has(?k,!l) and (?k=done);");
    ArrayList actions2 = new ArrayList(result_xwros.getBindings());
    ArrayList actions3 = new ArrayList(result_questions.getBindings());

    for (Iterator it = actions2.iterator(); it.hasNext();) {
        HashMap hm = (HashMap) it.next();
        String segment = hm.get("?x").toString();
        segment = segment.substring(segment.indexOf("#") + 1);
        String service = hm.get("?y").toString();
        service = service.substring(service.indexOf("#") + 1);
        tempString = textArea1.getText() + "\n";

        // Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
        // H service exei thn timh ths ?y h opoia einai ;h to Saloni
        // ;h to ypnodwmatio
        // Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
        // tha dwsei thn katallhlyh dyskolia sthn eksetash

        xwros=service;

        //katagrafh sthn othoni toy xwrou poy brisketai twra o mathitis
        textArea1.setText( tempString + "\n Room Number " +
        segment + "\n Room Name: " + service + "\n Room Name: " + xwros);
    }

    for (Iterator it = actions3.iterator(); it.hasNext();) {
        HashMap hm = (HashMap) it.next();
        String segment = hm.get("?k").toString();
        segment = segment.substring(segment.indexOf("#") + 1);
        String service = hm.get("?l").toString();
        service = service.substring(service.indexOf("#") + 1);
        tempString = textArea1.getText() + "\n";

        // Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
        // H service exei thn timh ths ?y h opoia einai ;h to Saloni
        // ;h to ypnodwmatio
        // Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
        // tha dwsei thn katallhlyh dyskolia sthn eksetash

        if ((service.equals("question_a2")) && (segment.equals("done"))){
            checkbox1.setState(true);
            checkbox1.setEnabled(false);
            lock_counter=lock_counter+1;
        }
    }
}
```



```
    }
    if ((service.equals("question_b2")) && (segment.equals("done"))) {
        checkbox2.setState(true);
        checkbox2.setEnabled(false);
        lock_counter=lock_counter+1;
    }
    if ((service.equals("question_c2")) && (segment.equals("done"))) {
        checkbox3.setState(true);
        checkbox3.setEnabled(false);
        lock_counter=lock_counter+1;
    }
    if ((service.equals("question_d2")) && (segment.equals("done"))) {
        checkbox4.setState(true);
        checkbox4.setEnabled(false);
        lock_counter=lock_counter+1;
    }
    if ((service.equals("question_e2")) && (segment.equals("done"))) {
        checkbox5.setState(true);
        checkbox5.setEnabled(false);
        lock_counter=lock_counter+1;
    }
    if(lock_counter==5){
        checkbox6.setState(true);
        lock_answers(evt);
    }
    if (times_in_hall_2==5){
        checkbox6.setState(true);
        lock_answers(evt);
    }
}
}

if (c==3){
    session=3;
    Answer result_xwros = ir.ask1("query q is i:is_it(?x,?y) and (?y=hall_3);");
    Answer result_questions = r.ask1("query q10 is w:has(?k,!l) and (?k=done);");
    ArrayList actions2 = new ArrayList(result_xwros.getBindings());
    ArrayList actions3 = new ArrayList(result_questions.getBindings());

    for (Iterator it = actions2.iterator(); it.hasNext(); ) {
        HashMap hm = (HashMap) it.next();
        String segment = hm.get("?x").toString();
        segment = segment.substring(segment.indexOf("#") + 1);
        String service = hm.get("?y").toString();
        service = service.substring(service.indexOf("#") + 1);
    }
}
```



```
tempString = textArea1.getText() + "\n";

// Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
// H service exei thn timh ths ?y h opoia einai ;h to Saloni
// ;h to ypnodwmatio
// Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
// tha dwsei thn katallhlh dyskolia sthn eksetash

xwros=service;

//katagrafh sthn othoni toy xwrou poy brisketai twra o mathitis
textArea1.setText( tempString + "\n Room Number " +
segment + "\n Room Name: " + service + "\n Room Name: " + xwros);
}

for (Iterator it = actions3.iterator(); it.hasNext(); ) {
    HashMap hm = (HashMap) it.next();
    String segment = hm.get("?k").toString();
    segment = segment.substring(segment.indexOf("#") + 1);
    String service = hm.get("?l").toString();
    service = service.substring(service.indexOf("#") + 1);
    //System.out.println("\n Mathima: " + segment);
    //System.out.println("\n Mathitis: " + service);
    tempString = textArea1.getText() + "\n";

    // Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
    // H service exei thn timh ths ?y h opoia einai ;h to Saloni
    // ;h to ypnodwmatio
    // Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
    // tha dwsei thn katallhlh dyskolia sthn eksetash

    if ((service.equals("question_a3")) && (segment.equals("done"))) {
        checkbox1.setState(true);
        checkbox1.setEnabled(false);
        lock_counter=lock_counter+1;
    }
    if ((service.equals("question_b3")) && (segment.equals("done"))) {
        checkbox2.setState(true);
        checkbox2.setEnabled(false);
        lock_counter=lock_counter+1;
    }
    if ((service.equals("question_c3")) && (segment.equals("done"))) {
        checkbox3.setState(true);
        checkbox3.setEnabled(false);
        lock_counter=lock_counter+1;
    }
}
```



```
        if ((service.equals("question_d3")) && (segment.equals("done"))) {
            checkbox4.setState(true);
            checkbox4.setEnabled(false);
            lock_counter=lock_counter+1;
        }
        if ((service.equals("question_e3")) && (segment.equals("done"))) {
            checkbox5.setState(true);
            checkbox5.setEnabled(false);
            lock_counter=lock_counter+1;
        }

        if(lock_counter==5){
            checkbox6.setState(true);
            lock_answers(evt);
        }
    }
}

// r.tell("fact q1 is (w:done,w:question1);");
// Η result_xwros θα έχει όλους εκείνους τους individuals που ικανοποιούν
// το ερώτημα και αφορούν το xwro στον οποίο έχει μετακινήσει ο μαθητής
// Στο επόμενο τυπώνουμε τα αποτελέσματα της result_xwros αφαιρώντας
// τη URI

        boolean temp=false;
        // Η xwros είναι global μεταβλητή και περιέχει πλέον τον xwro στον οποίο
vrisketai
        // ο μαθητής
        // καλώ τη ρούτινα συμπερασμού, για να vgalw συμπεράσμα εάν η εκτέλεση θα
dyskolepsei ;h oxi

        // συμπερασμός_δωμάτιου(evt);

    }
    else if (! started){
        textArea1.setText( "\n Δεν έχει κερδίσει κάποια εκτέλεση \n");
    }
}
}
catch (Exception e)
{
    String tempString = textArea1.getText() + "\n";
    textArea1.setText( tempString + "\n kamia apantsh akoma \n");
}
}

private void Place_C_Senariou_1(java.awt.event.ActionEvent evt) {
```



```
try {  
  
    if (started){  
        session=1;  
  
        String tempString = "";  
        int lock_counter=0;  
        facts(evt);  
  
        // Erwthma  
        Answer result_xwros = ir.ask1("query q is i:is_it(?x,?y) and (?y=hall_1);");  
        Answer result_questions = r.ask1("query q10 is w:has(?k,?l) and (?k=done);");  
  
        // r.tell("fact q1 is (w:done,w:question1);");  
        // Η result_xwros θα έχει όλους εκείνους τους individuals που ικανοποιούν  
        // το erwthma και αφορούν το xwro στον οποίο έχει μετακινήσει ο μαθητής  
        // Στο επόμενο τμήμα τα αποτελέσματα της result_xwros αφαιρούνται  
        // το URI  
  
        ArrayList actions2 = new ArrayList(result_xwros.getBindings());  
        ArrayList actions3 = new ArrayList(result_questions.getBindings());  
  
        for (Iterator it = actions2.iterator(); it.hasNext();) {  
            HashMap hm = (HashMap) it.next();  
            String segment = hm.get("?x").toString();  
            segment = segment.substring(segment.indexOf("#") + 1);  
            String service = hm.get("?y").toString();  
            service = service.substring(service.indexOf("#") + 1);  
            tempString = textArea1.getText() + "\n";  
  
            // Στο σημείο αυτό εξάγουμε την πληροφορία από την ontologia  
            // Η service έχει την τιμή της ?y η οποία είναι ;h το Saloni  
            // ;h το ypnodwmatio  
            // Επειτα αυτή η μεταβλητή θα συγκριθεί και αν η συγκρίση είναι true  
            // θα δώσει την κατάλληλη δυσκολία στην εκτέλεση  
  
            xwros=service;  
  
            //καταγραφή στην οθόνη του xwrou που βρίσκεται τώρα ο μαθητής  
            textArea1.setText( tempString + "\n Room Number " +  
segment + "\n Room Name: " + service + "\n Room Name: " + xwros);  
        }  
  
        for (Iterator it = actions3.iterator(); it.hasNext();) {  
            HashMap hm = (HashMap) it.next();  
            String segment = hm.get("?k").toString();
```



```
        segment = segment.substring(segment.indexOf("#") + 1);
        String service = hm.get("?l").toString();
        service = service.substring(service.indexOf("#") + 1);
tempString = textArea1.getText() + "\n";

// Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
// H service exei thn timh ths ?y h opoia einai ;h to Saloni
// ;h to ypnodwmatio
// Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
// tha dwsei thn katallhlh dyskolia sthn eksetash

if ((service.equals("question_a1")) && (segment.equals("done"))) {
    checkbox1.setState(true);
    checkbox1.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_b1")) && (segment.equals("done"))) {
    checkbox2.setState(true);
    checkbox2.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_c1")) && (segment.equals("done"))) {
    checkbox3.setState(true);
    checkbox3.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_d1")) && (segment.equals("done"))) {
    checkbox4.setState(true);
    checkbox4.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_e1")) && (segment.equals("done"))) {
    checkbox5.setState(true);
    checkbox5.setEnabled(false);
    lock_counter=lock_counter+1;
}

if(lock_counter==5){
    checkbox6.setState(true);
    lock_answers(evt);
}

}
boolean temp=false;
// H xwros einai global metablth kai periexei pleon ton xwro ston opoio
vrisketai
```



```
        // ο mathitis
        // kalw th roytina symerasmou, gia na vgalw symperasma ean h eksetash tha
        dyskolepsei ;h oxi

    }
    else if (! started){
        textArea1.setText( "\n Den exei ksekinhsei kapoia eksetash \n");
    }
}
catch (Exception e)
{
    String tempString = textArea1.getText() + "\n";
    textArea1.setText( tempString + "\n kamia apanthsh akoma \n");
}
}
```

```
private void Place_C_Senariou_2(java.awt.event.ActionEvent evt) {
    try {

        if (started){
            session=2;
            int lock_counter=0;
            String tempString = "";

            facts(evt);
            if (times_in_hall_2 != 5){
                times_in_hall_2=times_in_hall_2+1;
            }
            // Erwthma
            Answer result_xwros = ir.ask1("query q is i:is_it(?x,?y) and (?y=hall_2);");
            Answer result_questions = r.ask1("query q10 is w:has(?k,?l) and (?k=done);");

            // r.tell("fact q1 is (w:done,w:question1);");
            // H result_xwros tha exei olous ekeinous toyw individuals poy ikanopoioun
            // to erwthma kai aforoun to xwro ston opoio exei metakinitheo o mathitis
            // Sto epomeno tywnoyme ta apotelesmata ths result_xwros afairontas
            // th URI

            ArrayList actions2 = new ArrayList(result_xwros.getBindings());
            ArrayList actions3 = new ArrayList(result_questions.getBindings());

            //textArea1.setText(" ");
            for (Iterator it = actions2.iterator(); it.hasNext(); ) {
                HashMap hm = (HashMap) it.next();
                String segment = hm.get("?x").toString();
            }
        }
    }
}
```





```
        segment = segment.substring(segment.indexOf("#") + 1);
        String service = hm.get("?y").toString();
        service = service.substring(service.indexOf("#") + 1);
tempString = textArea1.getText() + "\n";

// Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
// H service exei thn timh ths ?y h opoia einai ;h to Saloni
// ;h to ypnodwmatio
// Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
// tha dwsei thn katallhlyh dyskolia sthn eksetash

xwros=service;

//katagrafh sthn othoni toy xwrou poy brisketai twra o mathitis
textArea1.setText( tempString + "\n Room Number " +
segment + "\n Room Name: " + service + "\n Room Name: " + xwros);
}

for (Iterator it = actions3.iterator(); it.hasNext();) {
    HashMap hm = (HashMap) it.next();
    String segment = hm.get("?k").toString();
    segment = segment.substring(segment.indexOf("#") + 1);
    String service = hm.get("?l").toString();
    service = service.substring(service.indexOf("#") + 1);
tempString = textArea1.getText() + "\n";

// Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
// H service exei thn timh ths ?y h opoia einai ;h to Saloni
// ;h to ypnodwmatio
// Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
// tha dwsei thn katallhlyh dyskolia sthn eksetash

if ((service.equals("question_a2")) && (segment.equals("done"))){
    checkbox1.setState(true);
    checkbox1.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_b2")) && (segment.equals("done"))){
    checkbox2.setState(true);
    checkbox2.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_c2")) && (segment.equals("done"))){
    checkbox3.setState(true);
    checkbox3.setEnabled(false);
    lock_counter=lock_counter+1;
}
```



```
    }
    if ((service.equals("question_d2")) && (segment.equals("done"))) {
        checkbox4.setState(true);
        checkbox4.setEnabled(false);
        lock_counter=lock_counter+1;
    }
    if ((service.equals("question_e2")) && (segment.equals("done"))) {
        checkbox5.setState(true);
        checkbox5.setEnabled(false);
        lock_counter=lock_counter+1;
    }
    if(lock_counter==5){
        checkbox6.setState(true);
        lock_answers(evt);
    }
    if (times_in_hall_2==5){
        checkbox6.setState(true);
        lock_answers(evt);
    }
}
boolean temp=false;
// Η xwros einai global metablth kai periexei pleon ton xwro ston opoio
vrisketai
// o mathitis
// kalw th royтина symerasμου, gia na vgalw symperasma ean h eksetash tha
dyskolepei ;h oxi

}
else if (! started){
    textArea1.setText( "\n Den exei ksekinhsei kapoia eksetash \n");
}
}
}
catch (Exception e)
{
    String tempString = textArea1.getText() + "\n";
    textArea1.setText( tempString + "\n kamia apanthsh akoma \n");
    // e.printStackTrace();
}
}

private void Place_C_Senariou_3(java.awt.event.ActionEvent evt) {
    try {

        if (started){
            session=3;

```



```
int lock_counter=0;
    String tempString = "";

facts(evt);

// Erwthma
Answer result_xwros = ir.ask1("query q is i:is_it(?x,?y) and (?y=hall_3);");
Answer result_questions = r.ask1("query q10 is w:has(?k,!l) and (?k=done);");

// r.tell("fact q1 is (w:done,w:question1);");
// Η result_xwros θα έχει όλους εκείνους τοίχους individuals που ικανοποιούν
// το erwthma και αφορούν το xwros στο οποίο έχει μετακινήσει ο mathitis
// Στο επόμενο τυπώνουμε τα αποτελέσματα της result_xwros αφορώντας
// το URI

ArrayList actions2 = new ArrayList(result_xwros.getBindings());
ArrayList actions3 = new ArrayList(result_questions.getBindings());

//textArea1.setText(" ");
    for (Iterator it = actions2.iterator(); it.hasNext();) {
        HashMap hm = (HashMap) it.next();
        String segment = hm.get("?x").toString();
        segment = segment.substring(segment.indexOf("#") + 1);
        String service = hm.get("?y").toString();
        service = service.substring(service.indexOf("#") + 1);
        tempString = textArea1.getText() + "\n";

        // Στο σημείο αυτό εξασφαλίζουμε την πληροφωρία από την οντολογία
        // Η service έχει την τιμή της ?y η οποία είναι ;h το Saloni
        // ;h το ypnodwmatio
        // Επειτα ayth h metavlhth θα συγκριθεί και αν h sygkrish einai true
        // θα δώσει την κατάλληλη δυσκολία στην εκτέλεση

        xwros=service;

        //katagrafh sthn othoni toy xwrou pou brisketai twra o mathitis
        textArea1.setText( tempString + "\n Room Number " +
segment + "\n Room Name: " + service + "\n Room Name: " + xwros);
    }

    for (Iterator it = actions3.iterator(); it.hasNext();) {
        HashMap hm = (HashMap) it.next();
        String segment = hm.get("?k").toString();
        segment = segment.substring(segment.indexOf("#") + 1);
        String service = hm.get("?l").toString();
```



```
        service = service.substring(service.indexOf("#") + 1);
tempString = textArea1.getText() + "\n";

// Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
// H service exei thn timh ths ?y h opoia einai ;h to Saloni
// ;h to ypnodwmatio
// Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
// tha dwsei thn katallhlyh dyskolia sthn eksetash

if ((service.equals("question_a3")) && (segment.equals("done")) ){
    checkbox1.setState(true);
    checkbox1.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_b3")) && (segment.equals("done")) ){
    checkbox2.setState(true);
    checkbox2.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_c3")) && (segment.equals("done")) ){
    checkbox3.setState(true);
    checkbox3.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_d3")) && (segment.equals("done")) ){
    checkbox4.setState(true);
    checkbox4.setEnabled(false);
    lock_counter=lock_counter+1;
}
if ((service.equals("question_e3")) && (segment.equals("done")) ){
    checkbox5.setState(true);
    checkbox5.setEnabled(false);
    lock_counter=lock_counter+1;
}

if(lock_counter==5){
    checkbox6.setState(true);
    lock_answers(evt);
}
}
boolean temp=false;
// H xwros einai global metablth kai periexei pleon ton xwro ston opoio
vrisketai
// o mathitis
// kalw th royтина symerasmou, gia na vgalw symperasma ean h eksetash tha
dyskolepei ;h oxi
```



```
    }
    else if (! started){
        textArea1.setText( "\n Den exei ksekinhsei kapoia eksetash \n");
    }
}
catch (Exception e)
{
    String tempString = textArea1.getText() + "\n";
    textArea1.setText( tempString + "\n kamia apanthsh akoma \n");
}
}
private void Noisy_Place_B_Senariou_step_by_step (java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:

    try {

        if (started){
            thoryvos=true;

            String tempString = "";

            facts(evt);

            // Erwthma
            Answer result_xwros = ir.ask1("query q is i:has(?x,?y) and (?x=Noisy);");
            Answer result_questions = r.ask1("query q10 is w:has(?k,?l) and (?k=done);");

            // r.tell("fact q1 is (w:done,w:question1);");
            // H result_xwros tha exei olous ekeinous toyw individuals poy ikanopoioun
            // to erwthma kai aforoun to xwro ston opoio exei metakinitheo o mathitis
            // Sto epomeno tywnoyme ta apotelesmata ths result_xwros afairontas
            // th URI

            ArrayList actions2 = new ArrayList(result_xwros.getBindings());
            ArrayList actions3 = new ArrayList(result_questions.getBindings());

            //textArea1.setText(" ");
            for (Iterator it = actions2.iterator(); it.hasNext();) {
                HashMap hm = (HashMap) it.next();
                String segment = hm.get("?x").toString();
                segment = segment.substring(segment.indexOf("#") + 1);
                String service = hm.get("?y").toString();
                service = service.substring(service.indexOf("#") + 1);
```



```
tempString = textArea1.getText() + "\n";

// Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
// H service exei thn timh ths ?y h οποια einai ;h to Saloni
// ;h to ypnodwmatio
// Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
// tha dwsei thn katallhlh dyskolia sthn eksetash

xwros=service;

//katagrafh sthn othoni toy xwrou poy brisketai twra o mathitis
textArea1.setText( tempString + "\n Apo Hxo ? " + segment +
"\n Xwros: " + service + "\n xwros: " + xwros);
}

for (Iterator it = actions3.iterator(); it.hasNext(); ) {
    HashMap hm = (HashMap) it.next();
    String segment = hm.get("?k").toString();
    segment = segment.substring(segment.indexOf("#") + 1);
    String service = hm.get("?l").toString();
    service = service.substring(service.indexOf("#") + 1);
tempString = textArea1.getText() + "\n";

// Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
// H service exei thn timh ths ?y h οποια einai ;h to Saloni
// ;h to ypnodwmatio
// Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
// tha dwsei thn katallhlh dyskolia sthn eksetash

if ((service.equals("question_an")) && (segment.equals("done")) ){
    checkbox1.setState(true);
    checkbox1.setEnabled(false);
}
if ((service.equals("question_bn")) && (segment.equals("done")) ){
    checkbox2.setState(true);
    checkbox2.setEnabled(false);
}
if ((service.equals("question_cn")) && (segment.equals("done")) ){
    checkbox3.setState(true);
    checkbox3.setEnabled(false);
}
if ((service.equals("question_dn")) && (segment.equals("done")) ){
    checkbox4.setState(true);
    checkbox4.setEnabled(false);
}
if ((service.equals("question_en")) && (segment.equals("done")) ){
```



```
        checkbox5.setState(true);
        checkbox5.setEnabled(false);
    }

}
boolean temp=false;
// Η xwros einai global metablth kai perixei pleon ton xwro ston opoio
vrisketai
// o mathitis
// kalw th royтина symerasμου, gia na vgalw symperasma ean h eksetash tha
dyskolepei ;h oxi

    symperasmos_dwmatiου(evt);

}
else if (! started){
    textArea1.setText( "\n Den exei ksekinhsei kapoia eksetash \n");
}
}
}
catch (Exception e)
{
    String tempString = textArea1.getText() + "\n";
    textArea1.setText( tempString + "\n kamia apanthsh akoma \n");
}
}

private void Quiet_Place_B_Senariou_step_by_step (java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:

    try {
        thoryvos=false;
        if (started){

            String tempString = "";

            facts(evt);

            // Erwthma
            Answer result_xwros = ir.ask1("query q is i:has(?x,?y) and (?x=Quiet);");
            Answer result_questions = r.ask1("query q10 is w:has(?k,!l) and (?k=done);");

            // Η result_xwros tha exei olous ekeinous toyw individuals poy ikanopoioun
            // to erwthma kai aforoun to xwro ston opoio exei metakinithi o mathitis
            // Sto epomeno typwnoyme ta apotelesmata ths result_xwros afairontas
```



```
// th URI

ArrayList actions2 = new ArrayList(result_xwros.getBindings());
ArrayList actions3 = new ArrayList(result_questions.getBindings());
    for (Iterator it = actions2.iterator(); it.hasNext();) {
        HashMap hm = (HashMap) it.next();
        String segment = hm.get("?x").toString();
        segment = segment.substring(segment.indexOf("#") + 1);
        String service = hm.get("?y").toString();
        service = service.substring(service.indexOf("#") + 1);

        tempString = textArea1.getText() + "\n";

        // Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
        // H service exei thn timh ths ?y h οποια einai to dwmatio me thoryvo
quiet
        // Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
        // tha dwsei thn katallhlyh dyskolia sthn eksetash

        xwros=service;

        textArea1.setText( tempString + "\n Apo Hxo ? " + segment +
"\n Xwros: " + service + "\n xwros: " + xwros);
    }
    for (Iterator it = actions3.iterator(); it.hasNext();) {
        HashMap hm = (HashMap) it.next();
        String segment = hm.get("?k").toString();
        segment = segment.substring(segment.indexOf("#") + 1);
        String service = hm.get("?l").toString();
        service = service.substring(service.indexOf("#") + 1);
        tempString = textArea1.getText() + "\n";

        // Sto shmeio ayto eksagoyme thn plhroforia apo thn ontologia
        // H service exei thn timh ths ?y h οποια einai ;h to Saloni
        // ;h to ypnodwmatio
        // Epeita ayth h metavlth tha sygkrithei kai an h sygkrish einai true
        // tha dwsei thn katallhlyh dyskolia sthn eksetash

        if ((service.equals("question_aq")) && (segment.equals("done")) ){
            checkbox1.setState(true);
            checkbox1.setEnabled(false);
        }
        if ((service.equals("question_bq")) && (segment.equals("done")) ){
            checkbox2.setState(true);
            checkbox2.setEnabled(false);
        }
    }
}
```





```
        if ((service.equals("question_cq")) && (segment.equals("done"))) {
            checkbox3.setState(true);
            checkbox3.setEnabled(false);
        }
        if ((service.equals("question_dq")) && (segment.equals("done"))) {
            checkbox4.setState(true);
            checkbox4.setEnabled(false);
        }
        if ((service.equals("question_eq")) && (segment.equals("done"))) {
            checkbox5.setState(true);
            checkbox5.setEnabled(false);
        }
    }

    boolean temp=false;

    // Η xwros einai global metablth kai periexei pleon ton xwro ston opoio
    vrisketai
    // o mathitis
    // kalw th royтина symerasμου, gia na vgalw symperasma ean h eksetash tha
    dyskolepsei ;h oxi

    symperasmos_dwmatiou(evt);

    }
    else if (! started){
        textArea1.setText( "\n Den exei ksekinhsei kapoia eksetash \n");
    }
    }
    catch (Exception e)
    {
        String tempString = textArea1.getText() + "\n";
        textArea1.setText( tempString + "\n kamia apanthsh akoma \n");
    }
}

//button of B_Senario step by step

private void button9ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    try{

        // Arxikopoihsh othonhs 2oy eksp. senariou

        visible_checkboxes(evt);
        clear_checkboxes(evt);
```



```
unlock_answers(evt);
button3.setEnabled(false);
button14.setEnabled(false);
button10.setEnabled(false);
button15.setEnabled(false);
checkbox6.setVisible(false);
label1.setText("Answer an click button to save");
button13.setVisible(true);
button11.setVisible(false);
//button12.setVisible(true);

// Η μεταβλητή time_step δίνει βηματική εκτέλεση στο senario
// Στὴν οὐσία καθὲν ἕνα time_step εἶναι 1 sec
time_step=time_step+1;

String tempString = "";
textArea1.setText(" ");
arrayOfIntegers = new double[32*time_step]; // allocates memory for N
doubles
Reader r = new BufferedReader(new FileReader("c:\\contextdata.txt"));
StreamTokenizer stok = new StreamTokenizer(r);
stok.parseNumbers();
double sum = 0;
int counter=0;
int inverse=0;
stok.nextToken();
while (stok.ttype != StreamTokenizer.TT_EOF) {
    if (stok.ttype == StreamTokenizer.TT_NUMBER){
        if(counter < 32*time_step){
            arrayOfIntegers[counter]=stok.nval;
            if (counter%32 == 0 && counter !=0 && counter >= 32*(time_step-
1)) {
                inverse=32;
                tempString = textArea1.getText() + "\n";
                textArea1.setText(tempString + "\n----- next row -----
\n");
                tempString = textArea1.getText() + "\n";
                textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Scenario number: " + arrayOfIntegers[((counter-inverse)/32)*32
+(counter-inverse)%32]);
                inverse=inverse-1;tempString = textArea1.getText() + "\n";
                textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Repetition number: " + arrayOfIntegers[((counter-inverse)/32)*32
+(counter-inverse)%32]);
                inverse=inverse-1;tempString = textArea1.getText() + "\n";
```



```
        textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Time (s): " + arrayOfIntegers[((counter-inverse)/32)*32 +(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Position:DisplayDown: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Position:DisplayUp: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Position:AntennaDown: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Position:AntennaUp: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Position:SidewaysRight: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Position:SidewaysLeft: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Stability:Stable: " + arrayOfIntegers[(counter-inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Stability:Unstable: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Device:Placement:AtHand: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Light:EU: " + arrayOfIntegers[(counter-inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Light:USA: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
```



```
//textArea1.setText( tempString + "column " + (counter-
inverse)% 32 + " Environment:Light:Bright: " + arrayOfIntegers[(counter-
inverse)% 32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)% 32 + " Environment:Light:Normal: " + arrayOfIntegers[(counter-
inverse)% 32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)% 32 + " Environment:Light:Dark: " + arrayOfIntegers[(counter-
inverse)% 32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)% 32 + " Environment:Light:Natural: " + arrayOfIntegers[(counter-
inverse)% 32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)% 32 + " Environment:Light:TotalDarkness: " + arrayOfIntegers[(counter-
inverse)% 32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)% 32 + " Environment:Temperature:Hot: " + arrayOfIntegers[(counter-
inverse)% 32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)% 32 + " Environment:Temperature:Warm: " + arrayOfIntegers[(counter-
inverse)% 32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)% 32 + " Environment:Temperature:Cool: " + arrayOfIntegers[(counter-
inverse)% 32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)% 32 + " Environment:Temperature:Cold: " + arrayOfIntegers[(counter-
inverse)% 32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)% 32 + " Environment:Humidity:Humid: " + arrayOfIntegers[(counter-
inverse)% 32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
    //textArea1.setText( tempString + "column " + (counter-
inverse)% 32 + " Environment:Humidity:Normal: " + arrayOfIntegers[(counter-
inverse)% 32]);
    inverse=inverse-1;tempString = textArea1.getText() + "\n";
```



```
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:Humidity:Dry: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:SoundPressure:Silent: " + arrayOfIntegers[((counter-
inverse)/32)*32 +(counter-inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:SoundPressure:Modest: " + arrayOfIntegers[((counter-
inverse)/32)*32 +(counter-inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " Environment:SoundPressure:Loud: " + arrayOfIntegers[((counter-
inverse)/32)*32 +(counter-inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " UserAction:Movement:Walking: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " UserAction:Movement:WalkingFast: " + arrayOfIntegers[(counter-
inverse)%32]);
        inverse=inverse-1;tempString = textArea1.getText() + "\n";
        //textArea1.setText( tempString + "column " + (counter-
inverse)%32 + " UserAction:Movement:Running: " + arrayOfIntegers[(counter-
inverse)%32]);
        tempString = textArea1.getText() + "\n";
        textArea1.setText(tempString + "----- end of row -----
\n");
        textArea1.setText(tempString + "column : " + (((counter-
inverse)/32)*32 +(counter-inverse)%32 - 4) + "\n");
        if (arrayOfIntegers[((counter-inverse)/32)*32 +(counter-inverse)%32 -
4])>=0.3){

                // Noisy Saloni

                Noisy_Place_B_Senariou_step_by_step(evt);

        } else if (arrayOfIntegers[((counter-inverse)/32)*32 +(counter-
inverse)%32 - 4])< 0.3){

                // Quiet Ypnodwmatio

                Quiet_Place_B_Senariou_step_by_step(evt);
```



```
        }
    }

    counter=counter+1;
    }
    sum += stok.nval;
    }
    else
        textArea1.setText("Nonnumber: " + stok.sval);
        stok.nextToken();
    }
    //textArea1.setText("The file sum is " + sum);
}
catch (Exception e)
{
    e.printStackTrace();
}
}

private void button11ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    try{
        if (checkbox1.getState()==true){
            checkbox1.setState(false);
        }else if(checkbox1.getState()==false){
            checkbox1.setState(true);
        }
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

private void button12ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    clear_checkboxes(evt);

}

private void button13ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    if (thoryvos==true){
        question_facts_noisy(evt);
    }else if(thoryvos==false){
        question_facts_quiet(evt);
    }
}
```



```
}  
}
```

```
private void button14ActionPerformed(java.awt.event.ActionEvent evt) {  
// TODO add your handling code here:
```

```
    button15.setEnabled(false);  
    button10.setEnabled(false);  
    button9.setEnabled(false);  
    checkbox6.setVisible(false);  
    label1.setText("Κατάσταση Δοκιμών");  
    button1.setVisible(true);  
    button2.setVisible(true);  
    button4.setVisible(true);  
    button5.setVisible(true);  
    button6.setVisible(true);  
    button7.setVisible(true);  
    button8.setVisible(true);  
    //button12.setVisible(true);  
}
```

```
private void button16ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    r.load(IRReasoner.OWL, GEontURI);  
    r.setSerializationMode(IRReasoner.DEFAULT);  
    r.setBaseNamespace(ruleURI);  
    r.setNamespacePrefix("w", ruleURI);  
    r.setNamespacePrefix("g", GEontURI);  
  
    ir.load(IRReasoner.OWL, INOURI);  
    ir.setSerializationMode(IRReasoner.DEFAULT);  
    ir.setBaseNamespace(ruleURI3);  
    ir.setNamespacePrefix("i", ruleURI3);  
    ir.setNamespacePrefix("co", INOURI);  
  
    try{  
        String result_all_GEont = r.run();  
        String result_all_INO = inor.run();  
        String tempString;  
        // Diadikasia Diaxwrismou tw n apotelesmatwn  
  
        //System.out.println("Fourth session:\n" + resulti5);  
        textArea1.setText("\nApo GEont :\n" + result_all_GEont);  
        tempString = textArea1.getText() + "\n";  
        textArea1.setText(tempString + "\nApo INO :\n" + result_all_INO);
```



```
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

private void button15ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    try{

        unlock_answers(evt);
        // Arxikopoihsh othonhs 2oy ekp. senariou
        button14.setEnabled(false);
        button10.setEnabled(false);
        button9.setEnabled(false);
        button3.setEnabled(false);
        visible_checkboxes(evt);
        clear_checkboxes(evt);
        checkbox6.setVisible(true);
        checkbox6.setState(false);
        label1.setText("Answer an click button to save");
        button17.setVisible(true);
        button11.setVisible(false);
        // button12.setVisible(true);

        // H metablhth time_step dinei bhmatikh ektelesh sto senario
        // Sthn ousia kathe ena time_step einai 1 sec
        time_step=time_step+1;

        String tempString = "";
        textArea1.setText(" ");
        arrayOfIntegers = new double[4*time_step]; // allocates memory for N
doubles
        Reader r = new BufferedReader(new FileReader("c:\\RFIDcontextdata.txt"));
        StreamTokenizer stok = new StreamTokenizer(r);
        stok.parseNumbers();
        double hall=0;
        double sum = 0;
        int counter=0;
        int inverse=0;
        stok.nextToken();
        while (stok.ttype != StreamTokenizer.TT_EOF) {
            if (stok.ttype == StreamTokenizer.TT_NUMBER){
```





```
        if(counter < 4*time_step){
            arrayOfIntegers[counter]=stok.nval;
            if (counter%4 == 0 && counter !=0 && counter >= 4*(time_step-
1)) {
                inverse=4;
                tempString = textArea1.getText() + "\n";
                textArea1.setText(tempString + "\n----- next row -----
\n");
                tempString = textArea1.getText() + "\n";
                textArea1.setText( tempString + "column " + (counter-inverse)%4
+ " Scenario number: " + arrayOfIntegers[((counter-inverse)/4)*4 +(counter-
inverse)%4]);
                inverse=inverse-1;tempString = textArea1.getText() + "\n";
                textArea1.setText( tempString + "column " + (counter-inverse)%4
+ " Repetition number: " + arrayOfIntegers[((counter-inverse)/4)*4 +(counter-
inverse)%4]);
                inverse=inverse-1;tempString = textArea1.getText() + "\n";
                textArea1.setText( tempString + "column " + (counter-inverse)%4
+ " Time (s): " + arrayOfIntegers[((counter-inverse)/4)*4 +(counter-inverse)%4]);
                inverse=inverse-1;tempString = textArea1.getText() + "\n";
                hall=arrayOfIntegers[((counter-inverse)/4)*4 +(counter-inverse)%4];
                textArea1.setText( tempString + "column " + (counter-inverse)%4
+ " Hall: " + arrayOfIntegers[((counter-inverse)/4)*4 +(counter-inverse)%4]);
                inverse=inverse-1;tempString = textArea1.getText() + "\n";
                tempString = textArea1.getText() + "\n";
                textArea1.setText(tempString + "----- end of row -----
\n");
                textArea1.setText(tempString + "column : " + (((counter-inverse)/4)*4
+(counter-inverse)%4) + "\n");
                tempString = textArea1.getText() + "\n";

                if (hall==45){

                    // Hall No 45
                    Place_C_Senariou(evt, 1);
                    // Place_C_Senariou_1(evt);

                } else if (hall==46){

                    // Hall No 46
                    Place_C_Senariou(evt, 2);
                    // Place_C_Senariou_2(evt);

                }else if (hall==47){

                    // Hall No 47
```



```
        Place_C_Senariou(evt, 3);
        // Place_C_Senariou_3(evt);
    }
}

        counter=counter+1;
    }
    sum += stok.nval;
}
else
    textArea1.setText("Nonnumber: " + stok.sval);
    stok.nextToken();
}
//textArea1.setText("The file sum is " + sum);
}
catch (Exception e)
{
    e.printStackTrace();
}
}

private void button17ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    if (session==1){
        question_facts_session1(evt);
    }else if(session==2){
        question_facts_session2(evt);
    }else if(session==3){
        question_facts_session3(evt);
    }
}

private void button18ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    System.exit(0);
}

private void visible_checkboxes (java.awt.event.ActionEvent evt) {
    try{
        checkbox1.setVisible(true);
        checkbox2.setVisible(true);
        checkbox3.setVisible(true);
        checkbox4.setVisible(true);
        checkbox5.setVisible(true);
    }
```



```
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

private void not_visible_checkboxes (java.awt.event.ActionEvent evt) {
    try{
        checkbox1.setVisible(false);
        checkbox2.setVisible(false);
        checkbox3.setVisible(false);
        checkbox4.setVisible(false);
        checkbox5.setVisible(false);
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

private void clear_checkboxes(java.awt.event.ActionEvent evt) {
    try{
        checkbox1.setState(false);
        checkbox2.setState(false);
        checkbox3.setState(false);
        checkbox4.setState(false);
        checkbox5.setState(false);
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

private void question_facts_session1(java.awt.event.ActionEvent evt) {
    try{
        // Eisagwgh neas gnwshs se property poy den yparxei stis ontologies GEOnt
        & context
        // kai emfanish mono twn properties pou exoun zititheis sta facts
        // mporoyme dhl na kanoume query mono tis idiothtes didaskei &
        exei_epilexthei
        if (checkbox1.getState()==true){
            r.tell("fact s1q1 is has(w:done,w:question_a1);");
        }
        if (checkbox2.getState()==true){
```



```
        r.tell("fact s1q2 is has(w:done,w:question_b1);");
    }
    if (checkbox3.getState()==true){
        r.tell("fact s1q3 is has(w:done,w:question_c1);");
    }
    if (checkbox4.getState()==true){
        r.tell("fact s1q4 is has(w:done,w:question_d1);");
    }
    if (checkbox5.getState()==true){
        r.tell("fact s1q5 is has(w:done,w:question_e1);");
    }
    if (checkbox6.getState()==true){
        r.tell("fact s1q6 is has(w:locked,w:question_f1);");
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
}

private void question_facts_session2(java.awt.event.ActionEvent evt) {
    try{
        // Eisagwgh neas gnwshs se property poy den yparxei stis ontologies GEOnt
        & context
        // kai emfanish mono twn properties pou exoun zitithei sta facts
        // mporoyme dhl na kanoume query mono tis idiothtes didaskei &
        exei_epilexthei
        if (checkbox1.getState()==true){
            r.tell("fact s2q1 is has(w:done,w:question_a2);");
        }
        if (checkbox2.getState()==true){
            r.tell("fact s2q2 is has(w:done,w:question_b2);");
        }
        if (checkbox3.getState()==true){
            r.tell("fact s2q3 is has(w:done,w:question_c2);");
        }
        if (checkbox4.getState()==true){
            r.tell("fact s2q4 is has(w:done,w:question_d2);");
        }
        if (checkbox5.getState()==true){
            r.tell("fact s2q5 is has(w:done,w:question_e2);");
        }
        if (checkbox6.getState()==true){
```



```
        r.tell("fact s2q6 is has(w:locked,w:question_f2);");
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
}

private void question_facts_session3(java.awt.event.ActionEvent evt) {
    try{
        // Eisagwgh neas gnwshs se property poy den yparxei stis ontologies GEOnt
        & context
        // kai emfanish mono twn properties pou exoun zitithei sta facts
        // mporoyme dhl na kanoume query mono tis idiothtes didaskei &
        exei_epilexthei
        if (checkbox1.getState()==true){
            r.tell("fact s3q1 is has(w:done,w:question_a3);");
        }
        if (checkbox2.getState()==true){
            r.tell("fact s3q2 is has(w:done,w:question_b3);");
        }
        if (checkbox3.getState()==true){
            r.tell("fact s3q3 is has(w:done,w:question_c3);");
        }
        if (checkbox4.getState()==true){
            r.tell("fact s3q4 is has(w:done,w:question_d3);");
        }
        if (checkbox5.getState()==true){
            r.tell("fact s3q5 is has(w:done,w:question_e3);");
        }
        if (checkbox6.getState()==true){
            r.tell("fact s3q6 is has(w:locked,w:question_f3);");
        }
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

private void question_facts_noisy(java.awt.event.ActionEvent evt) {

    try{
        // Eisagwgh neas gnwshs se property poy den yparxei stis ontologies GEOnt
        & context
```



```
// kai emfanish mono twm properties pou exoun zitithei sta facts
// mporoyme dhl na kanoume query mono tis idiothtes didaskei &
exeipilexthei
    if (checkbox1.getState()==true){
        r.tell("fact q1 is has(w:done,w:question_an);");
    }
}
    if (checkbox2.getState()==true){
        r.tell("fact q2 is has(w:done,w:question_bn);");
    }
    if (checkbox3.getState()==true){
        r.tell("fact q3 is has(w:done,w:question_cn);");
    }
    if (checkbox4.getState()==true){
        r.tell("fact q4 is has(w:done,w:question_dn);");
    }
    if (checkbox5.getState()==true){
        r.tell("fact q5 is has(w:done,w:question_en);");
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
}

private void question_facts_quiet(java.awt.event.ActionEvent evt) {

    try{
        // Eisagwgh neas gnwshs se property poy den yparxei stis ontologies GEOnt
        & context
        // kai emfanish mono twm properties pou exoun zitithei sta facts
        // mporoyme dhl na kanoume query mono tis idiothtes didaskei &
        exeipilexthei
        if (checkbox1.getState()==true){
            r.tell("fact q11 is has(w:done,w:question_aq);");
        }
        if (checkbox2.getState()==true){
            r.tell("fact q12 is has(w:done,w:question_bq);");
        }
        if (checkbox3.getState()==true){
            r.tell("fact q13 is has(w:done,w:question_cq);");
        }
        if (checkbox4.getState()==true){
            r.tell("fact q14 is has(w:done,w:question_dq);");
        }
    }
```



```
        if (checkbox5.getState()==true){
            r.tell("fact q15 is has(w:done,w:question_eq);");
        }
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

private void clear_answers_in_second(java.awt.event.ActionEvent evt) {
    try{
        r.tell("fact q1 is has(w:doesnt,w:question_an);");
        r.tell("fact q2 is has(w:doesnt,w:question_bn);");
        r.tell("fact q3 is has(w:doesnt,w:question_cn);");
        r.tell("fact q4 is has(w:doesnt,w:question_dn);");
        r.tell("fact q5 is has(w:doesnt,w:question_en);");
        r.tell("fact q11 is has(w:doesnt,w:question_aq);");
        r.tell("fact q12 is has(w:doesnt,w:question_bq);");
        r.tell("fact q13 is has(w:doesnt,w:question_cq);");
        r.tell("fact q14 is has(w:doesnt,w:question_dq);");
        r.tell("fact q15 is has(w:doesnt,w:question_eq);");
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

private void facts(java.awt.event.ActionEvent evt) {

    // Dhmioyrgia twn Reasoners twn dyo ontologiwn

    r.load(IRReasoner.OWL, GEOntURI);
    r.setSerializationMode(IRReasoner.DEFAULT);
    r.setBaseNamespace(ruleURI);
    r.setNamespacePrefix("w", ruleURI);
    r.setNamespacePrefix("g", GEOntURI);

    ir.load(IRReasoner.OWL, INOURI);
    ir.setSerializationMode(IRReasoner.DEFAULT);
    ir.setBaseNamespace(ruleURI3);
    ir.setNamespacePrefix("i", ruleURI3);
    ir.setNamespacePrefix("co", INOURI);

    try{
```



```
// Eisagwgh neas gnwshs se property pou den yparxei stis ontologies GEOnt
& context
// kai emfanish mono twn properties pou exoun zitimethi sta facts
// mporoyme dhl na kanoume query mono tis idiothtes didaskei &
exei_epilexthei

ir.tell("fact ic is has(i:Noisy, i:Saloni);");
ir.tell("fact ic2 is has(i:Quiet, i:Ypnodwmatio);");
ir.tell("fact ic3 is is_it(i:n45, i:hall_1);");
ir.tell("fact ic4 is is_it(i:n46, i:hall_2);");
ir.tell("fact ic5 is is_it(i:n47, i:hall_3);");
r.tell("fact e4 is eksetazetai(w:Dhmhtriu, w:eksetash_Fysikh_eykolh);");
r.tell("fact e is eksetazetai(w:Georgiou, w:eksetash_Fysikh_metria);");
r.tell("fact e2 is eksetazetai(w:Ioannou, w:eksetash_Fysikh_metria);");
r.tell("fact e3 is eksetazetai(w:Dhmhtriu, w:eksetash_Fysikh_dyskolh);");

}
catch (Exception e)
{
    e.printStackTrace();
}
}

private void symperasmos_dwmatiou (java.awt.event.ActionEvent evt) {
try{
    String tempString = "";
    if (xwros.equals("Saloni")){

        r.tell("fact e5 is eksetazetai(w:Dhmhtriu, w:eksetash_Fysikh_eykolh);");

        // Erwthma
        Answer result_exam = r.ask1("query q is w:eksetazetai(?y,?x) and
(?x=eksetash_Fysikh_eykolh);");

        ArrayList actions = new ArrayList(result_exam.getBindings());
        for (Iterator it = actions.iterator(); it.hasNext(); ) {
            HashMap hm = (HashMap) it.next();
            String segment = hm.get("?x").toString();
            segment = segment.substring(segment.indexOf("#") + 1);
            String service = hm.get("?y").toString();
            service = service.substring(service.indexOf("#") + 1);
            tempString = textArea1.getText() + "\n";
            textArea1.setText( tempString + "\n Mathima: " + segment +
"\n Mathitis: " + service);
        }
    }
}
}
```





```
}else if (xwros.equals("Υπνοδωμάτιο")){
    r.tell("fact e5 is eksetazetai(w:Dhnhtriu, w:eksetash_Fysikh_dyskolh);");

    // Erwthma
    Answer result_exam = r.ask1("query q is w:eksetazetai(?y,?x) and
(?x=eksetash_Fysikh_dyskolh);");
    ArrayList actions = new ArrayList(result_exam.getBindings());
    for (Iterator it = actions.iterator(); it.hasNext();) {
        HashMap hm = (HashMap) it.next();
        String segment = hm.get("?x").toString();
        segment = segment.substring(segment.indexOf("#") + 1);
        String service = hm.get("?y").toString();
        service = service.substring(service.indexOf("#") + 1);
        tempString = textArea1.getText() + "\n";
        textArea1.setText( tempString + "\n Mathima: " + segment +
"\n Mathitis: " + service);
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
}

private void unlock_answers(java.awt.event.ActionEvent evt) {
    checkbox1.setEnabled(true);
    checkbox2.setEnabled(true);
    checkbox3.setEnabled(true);
    checkbox4.setEnabled(true);
    checkbox5.setEnabled(true);
    checkbox6.setEnabled(true);
}

private void lock_answers(java.awt.event.ActionEvent evt) {
    checkbox1.setEnabled(false);
    checkbox2.setEnabled(false);
    checkbox3.setEnabled(false);
    checkbox4.setEnabled(false);
    checkbox5.setEnabled(false);
    checkbox6.setEnabled(false);
}

private void initialize(java.awt.event.ActionEvent evt) {
    try{
```



```
not_visible_checkboxes(evt);
button15.setEnabled(true);
button10.setEnabled(true);
button9.setEnabled(true);
button14.setEnabled(true);
checkbox6.setVisible(false);
label1.setText("Διαλέξτε Εκπ. Σενάριο");
button13.setVisible(false);
button11.setVisible(false);
button12.setVisible(false);
button1.setVisible(false);
button2.setVisible(false);
button4.setVisible(false);
button5.setVisible(false);
button6.setVisible(false);
button7.setVisible(false);
button8.setVisible(false);
button12.setVisible(false);
button17.setVisible(false);

}
catch (Exception e)
{
    e.printStackTrace();
}
}

// Variables declaration - do not modify
private java.awt.Button button1;
private java.awt.Button button10;
private java.awt.Button button11;
private java.awt.Button button12;
private java.awt.Button button13;
private java.awt.Button button14;
private java.awt.Button button15;
private java.awt.Button button16;
private java.awt.Button button17;
private java.awt.Button button18;
private java.awt.Button button2;
private java.awt.Button button3;
private java.awt.Button button4;
private java.awt.Button button5;
private java.awt.Button button6;
private java.awt.Button button7;
private java.awt.Button button8;
private java.awt.Button button9;
```



```
private java.awt.Checkbox checkbox1;
private java.awt.Checkbox checkbox2;
private java.awt.Checkbox checkbox3;
private java.awt.Checkbox checkbox4;
private java.awt.Checkbox checkbox5;
private java.awt.Checkbox checkbox6;
private java.awt.Label label1;
private javax.swing.JPanel mainPanel;
private javax.swing.JMenuBar menuBar;
private javax.swing.JProgressBar progressBar;
private javax.swing.JLabel statusAnimationLabel;
private javax.swing.JLabel statusMessageLabel;
private javax.swing.JPanel statusPanel;
private java.awt.TextArea textArea1;
// End of variables declaration

private final Timer messageTimer;
private final Timer busyIconTimer;
private final Icon idleIcon;
private final Icon[] busyIcons = new Icon[15];
private int busyIconIndex = 0;

private JDialog aboutBox;

final static String GEOntURI =
"http://vsdesign.homelinux.org/~bstefan/GEOnt.owl#";
final static String contextURI =
"http://vsdesign.homelinux.org/~bstefan/context.owl#";
final static String INOURI = "http://vsdesign.homelinux.org/~bstefan/INO.owl#";

//The following URI occurs from execution. We can find it into the area of facts
// of execution. Also we can find it in protege first screen

final static String ruleURI = "http://www.owl-
ontologies.com/Ontology1225893950.owl#";
final static String ruleURI2 = "http://www.owl-
ontologies.com/Ontology1229414213.owl#";
final static String ruleURI3 = "http://vsdesign.homelinux.org/~bstefan/INO.owl#";
// Oles oi metavlhtes se ayton ton xwro einai global

boolean started=false;
boolean thoryvos=false;

String xwros="";
String question_q="";
```



```
int times_in_hall_2=0;
int time_step=0;
//int N=32*46045;
int N=32*50;
int session=0;
double[] arrayOfIntegers; // declares an array of doubles

IRReasonerFactory reasonerFactory = ReasonerFactory.getInstance();
IRReasoner r = reasonerFactory.createOwlDITrMReasoner();

IRReasonerFactory reasonerFactory2 = ReasonerFactory.getInstance();
IRReasoner ir = reasonerFactory2.createOwlDITrMReasoner();

IRReasonerFactory reasonerFactory3 = ReasonerFactory.getInstance();
IRReasoner inor = reasonerFactory3.createOwlDITrMReasoner();

}
```