

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών
Πληροφοριακά και Επικοινωνιακά Συστήματα
Κατεύθυνση: *Ευφυή Συστήματα*

Μεταπτυχιακή Διατριβή



ΔΗΜΙΟΥΡΓΙΑ ΕΞΥΠΝΟΥ ΒΟΗΘΟΥ ΓΙΑ ΕΚΠΑΙΔΕΥΤΙΚΟΥΣ

Μαρία Χατζηιωάννου

Επιβλέπων Καθηγητής
Λοΐζος Μιχαήλ

Μάϊος 2018

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών Επιστημών

**Μεταπτυχιακό Πρόγραμμα Σπουδών
Πληροφοριακά και Επικοινωνιακά Συστήματα
Κατεύθυνση: Ευφυή Συστήματα**

Μεταπτυχιακή Διατριβή

ΔΗΜΙΟΥΡΓΙΑ ΕΞΥΠΝΟΥ ΒΟΗΘΟΥ ΓΙΑ ΕΚΠΑΙΔΕΥΤΙΚΟΥΣ

Μαρία Χατζηιωάννου

**Επιβλέπων Καθηγητής
Λοΐζος Μιχαήλ**

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου σπουδών στα Ευφυή Συστήματα από τη Σχολή Θετικών Επιστημών του Ανοικτού Πανεπιστημίου Κύπρου.

Μάιος 2018

ΠΕΡΙΛΗΨΗ

Η παρούσα διατριβή ασχολείται με την δημιουργία ενός έξυπνου βοηθού για εκπαιδευτικούς σχετικά με συμπεριφορές μαθητών όλων των ηλικιών και το λόγο που προκαλεί αυτές τις συμπεριφορές, όπως αναφέρεται και στον τίτλο της. Σκοπός της διατριβής ήταν να συλλεχθεί η γνώση που κατέχουν κάποιοι εκπαιδευτικοί από διάφορα περιστατικά με προβληματικές ή αξιοπρόσεκτες συμπεριφορές των μαθητών τους και τους λόγους που προκαλούν αυτές τις συμπεριφορές, για να μπορούν να βοηθηθούν άλλοι εκπαιδευτικοί σε ανάλογες περιπτώσεις ώστε να μπορέσουν να διαχειριστούν αυτού του είδους τις καταστάσεις. Υπάρχει επιστημονικό/ερευνητικό ενδιαφέρον στη μελέτη νέων τεχνολογιών που μπορούν να εφαρμοστούν στον τομέα των ευφυή συστημάτων. Ένας έξυπνος βοηθός μπορεί να μαθαίνει εμπειρικά από τα δεδομένα που του δίνονται σε βάθος χρόνου και να τα καταχωρεί στη βάση γνώσης του καθώς και να εξάγει αποτελέσματα/συμπεράσματα όταν του ζητηθεί.

Στην εργασία αξιολογούνται ποιοτικά και πειραματικά οι υπάρχουσες τεχνολογίες αναπαράστασης γνώσης και συμπερασμού στα πλαίσια ανάπτυξης ενός συστήματος βασισμένου σε γνώση με καθορισμένες απαιτήσεις. Πρόκειται για έναν έξυπνο βοηθό που καταγράφει νέα γνώση από νέα περιστατικά που καταχωρούνται, με σκοπό την μάθηση σε βάθος χρόνου (μηχανική μάθηση), αλλά ταυτόχρονα θα μπορεί να εξάγει συμπεράσματα με βάση την ήδη καταγραμμένη γνώση στην κεντρική του αποθήκη (κανόνες – λογικός προγραμματισμός). Ο σχεδιασμός και η υλοποίηση του συστήματος βασίζονται στα αποτελέσματα που προέκυψαν από τη διαδικασία αξιολόγησης διαφόρων γλωσσών και εργαλείων. Το σύστημα κάνει χρήση σύγχρονων τεχνολογιών που επιτρέπουν την αναπαράσταση γνώσης μέσω εκφραστικών λογικών και την εφαρμογή αποδοτικών μεθόδων συμπερασμού.

Κατά την διάρκεια της παρούσας διατριβής έγινε μια προσπάθεια να παραχθεί ένα σύστημα με φιλικό γραφικό περιβάλλον επικοινωνίας με τους χρήστες το οποίο να μπορεί να καταχωρεί νέα περιστατικά αλλά να δίνει και οδηγίες ή πιθανούς λόγους για τους οποίους αντιδρά με συγκεκριμένο τρόπο κάποιος μαθητής σύμφωνα με την γνώση που έχει συλλέξει το σύστημα ήδη. Μία από τις βασικές απαιτήσεις του συστήματος είναι να μπορεί να αποκτήσει επιπλέον γνώση σε βάθος χρόνου, να μαθαίνει, ενώ ταυτόχρονα να είναι σε θέση να αξιοποιεί την αποκτηθείσα γνώση για να διευκολύνει την ήδη δύσκολη δουλειά των εκπαιδευτικών που όχι μόνο προσπαθούν να διδάξουν τα παιδιά μας, αλλά πλάθουν και τους χαρακτήρες των νέων μας για την καλύτερευση του εκπαιδευτικού μας συστήματος. Δόθηκε έμφαση στην συλλογή πληροφοριών και στην διαδικασία υλοποίησης – δημιουργίας ενός έξυπνου βοηθού. Η μελέτη των βασικών αρχών εκπαιδευτικής ψυχολογίας είναι βασικό βήμα για την ολοκλήρωση της παρούσας διατριβής, αλλά αποφεύχθηκε η εμβάθυνση στην επιστήμη της ψυχολογίας των μαθητών. Επίσης αντιμετωπίστηκαν κάποια προβλήματα συμβατότητας μεταξύ κάποιων εργαλείων και τεχνολογιών που επιλέχθηκαν αρχικά για να χρησιμοποιηθούν.

SUMMARY

This thesis deals with the creation of an intelligent assistant for teachers on behaviour of students of all ages and the reason that causes these behaviours, as it is mentioned in the title. The purpose of the dissertation was to collect the knowledge of some teachers from various incidents with problematic or remarkable behaviours of their students and the reasons for these behaviours so that other teachers can be assisted in such cases in order to be able to manage this kind of behaviour situations.

ΕΥΧΑΡΙΣΤΙΕΣ

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω την οικογένειά μου και το φιλικό μου περιβάλλον για την κατανόηση και την στήριξη που μου παρείχαν. Ενώ, θα ήθελα να ευχαριστήσω ιδιαίτερα να ευχαριστήσω τα άτομα που με βοήθησαν να συλλέξω με ερωτηματολόγια την αρχική γνώση που χρησιμοποιήθηκε για να μπορώ να αρχίσω να δουλεύω.

Επίσης, για να είμαι δίκαιη ευχαριστίες θα ήθελα να απευθύνω στο προσωπικό του Ανοικτού Πανεπιστημίου Κύπρου που βοήθησε να μεταφερθεί η Διατριβή μου από το ακαδημαϊκό έτος 2016 - 2017 στο έτος 2017 - 2018 μετά από κάποιο πρόβλημα που αντιμετώπιζα καθώς και τον επιβλέποντα καθηγητή μου κύριο Λοίζο Μιχαήλ για όλα όσα έκανε και την υπομονή του.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	ΕΙΣΑΓΩΓΗ.....	7
1.1	ΕΡΕΥΝΗΤΙΚΑ ΕΡΩΤΗΜΑΤΑ	8
1.2	ΑΝΑΓΚΑΙΟΤΗΤΑ ΚΑΙ ΣΠΟΥΔΑΙΟΤΗΤΑ ΕΡΕΥΝΑΣ	12
2	ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ	13
2.1	ΠΕΡΙΟΧΕΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΟΥ ΕΜΠΛΕΚΟΝΤΑΙ	13
2.2	ΤΕΧΝΟΛΟΓΙΕΣ, ΠΕΡΙΒΑΛΛΟΝΤΑ ΚΑΙ ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	18
3	ΜΕΘΟΔΟΛΟΓΙΑ	20
3.1	ΜΟΝΤΕΛΟ ΚΥΚΛΟΥ ΑΝΑΠΤΥΞΗΣ ΣΥΣΤΗΜΑΤΟΣ	21
3.2	ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ	24
4	ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΕΞΥΠΝΟΥ ΒΟΗΘΟΥ	26
4.1	ΠΡΟΔΙΑΓΡΑΦΕΣ ΣΥΣΤΗΜΑΤΟΣ	26
4.2	ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΕΞΥΠΝΟΥ ΒΟΗΘΟΥ	30
5	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ.....	44
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	45
	ΠΑΡΑΡΤΗΜΑΤΑ	
A	ΤΜΗΜΑΤΑ ΚΩΔΙΚΑ ΚΑΙ ΟΘΟΝΕΣ ΣΥΣΤΗΜΑΤΟΣ.....	48
A.1	ΜΕΤΑΤΡΟΠΗ ΕΜΠΕΙΡΟΥ ΣΥΣΤΗΜΑΤΟΣ ΣΕ ΚΩΔΙΚΑ PROLOG.....	48
A.2	ΟΘΟΝΕΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	62

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Γίνεται αντιληπτό ότι στις μέρες μας, όπου υπάρχει ραγδαία τεχνολογική ανάπτυξη και οι ανάγκες μας αυξάνονται, η ανάπτυξη της επιστήμης της Τεχνητής Νοημοσύνης¹ αποτελεί σημαντικό σταθμό. Τα Ευφυή συστήματα που μέχρι στιγμής τα βλέπαμε σε ταινίες επιστημονικής φαντασίας αποτελούν πλέον πραγματικότητα και υπόσχονται πολλά για το μέλλον της ανθρωπότητας αν χρησιμοποιηθούν ορθά. Η τεχνητή νοημοσύνη (AI - Artificial Intelligence) καθιστά τις μηχανές ικανές να μαθαίνουν από την εμπειρία, να προσαρμόζονται σε νέα εισαγόμενα δεδομένα και να εκτελούν ανθρωπομορφικά έργα. Τα περισσότερα παραδείγματα AI για τα οποία συναντώνται στις μέρες μας – από τους υπολογιστές που παίζουν σκάκι έως τα αυτο-οδηγούμενα αυτοκίνητα – βασίζονται σε μεγάλο βαθμό στο deep learning² και την επεξεργασία φυσικής γλώσσας. Με τη χρήση των τεχνολογιών αυτών, οι υπολογιστές μπορούν να εκπαιδευτούν ώστε να επιτελούν συγκεκριμένα καθήκοντα με επεξεργασία μεγάλων ποσοτήτων δεδομένων και αναγνώριση μορφών στα δεδομένα. Το AI είναι ιδιαίτερα χρήσιμο σε ό,τι αφορά εφαρμογές διαχείρισης ροών δεδομένων σε πραγματικό χρόνο, "έξυπνης βοήθειας" για την υποστήριξη πελατών κ.α.

Η παρούσα μεταπτυχιακή διατριβή, η οποία πραγματοποιείται στα πλαίσια του προγράμματος σπουδών στα επικοινωνιακά και πληροφοριακά συστήματα του ανοικτού πανεπιστημίου Κύπρου με ειδίκευση τα ευφυή συστήματα, αξιολογεί ποιοτικά και πειραματικά τις υπάρχουσες τεχνολογίες αναπαράστασης γνώσης και συμπερασμού στα πλαίσια ανάπτυξης ενός συστήματος

¹ **Τεχνητή Νοημοσύνη:** Ο όρος τεχνητή νοημοσύνη αναφέρεται στον κλάδο της πληροφορικής ο οποίος ασχολείται με τη σχεδίαση και την υλοποίηση υπολογιστικών συστημάτων που μιμούνται στοιχεία της ανθρώπινης συμπεριφοράς τα οποία υπονοούν έστω και στοιχειώδη ευφυΐα: μάθηση, προσαρμοστικότητα, εξαγωγή συμπερασμάτων, κατανόηση από συμφραζόμενα και την επίλυση προβλημάτων. Η Τεχνητή Νοημοσύνη επιχειρεί όχι μόνο να κατανοήσει αλλά και να κατασκευάσει νοήμονες οντότητες.

² **Deep Learning:** είναι τεχνική/υποπεδίο της μηχανικής μάθησης όπου οι υπολογιστές εκπαιδεύονται μέσα από παραδείγματα.

βασισμένου σε γνώση³ με καθορισμένες απαιτήσεις. Πρόκειται για έναν έξυπνο βοηθό (smart agent) που καταγράφει νέα γνώση με σκοπό την απόκτηση γνώσης σε βάθος χρόνου (μηχανική μάθηση – machine learning⁴), αλλά ταυτόχρονα μπορεί να εξάγει συμπεράσματα με βάση την ήδη καταγεγραμμένη γνώση στην κεντρική του αποθήκη από κανόνες, όπως γίνεται στο λογικό προγραμματισμό⁵. Ο σχεδιασμός και η υλοποίηση του συστήματος βασίζονται στα αποτελέσματα που προέκυψαν από τη διαδικασία αξιολόγησης των διαφόρων γλωσσών και εργαλείων. Το σύστημα θα κάνει χρήση σύγχρονων τεχνολογιών που επιτρέπουν την αναπαράσταση γνώσης⁶ μέσω εκφραστικών λογικών και την εφαρμογή αποδοτικών μεθόδων συμπερασμού. Σε φοιτητές του μεταπτυχιακού προγράμματος στα πληροφοριακά και επικοινωνιακά προγράμματα υπάρχει το επισημονικό/ερευνητικό ενδιαφέρον η μελέτη νέων τεχνολογιών και η χρήση τους σε μια προσπάθεια υλοποίησης ενός έξυπνου βοηθού. Ένας έξυπνος βοηθός μπορεί να μαθαίνει εμπειρικά από τα δεδομένα που του δίνονται σε βάθος χρόνου και να τα καταχωρεί στη βάση γνώσης του καθώς και να εξάγει αποτελέσματα/συμπεράσματα όταν του ζητηθεί.

Επίσης, στην έρευνα η οποία πραγματοποιήθηκε εμπλέκονται θέματα μαθησιακών δυσκολιών⁷, σχολικού εκφοβισμού⁸ και εκπαιδευτικής ψυχολογίας, αφού αυτό είναι το αντικείμενο μελέτης και

³ **Γνώση (knowledge):** είναι πληροφορία η οποία έχει υποστεί μία σειρά ειδικών ελέγχων για την πιστοποίησή της.

⁴ **Μηχανική μάθηση (machine learning):** είναι μια περιοχή της τεχνητής νοημοσύνης η οποία αφορά αλγορίθμους και μεθόδους που επιτρέπουν στους υπολογιστές να «μαθαίνουν». Με τη μηχανική μάθηση καθίσταται εφικτή η κατασκευή προσαρμοσίμων (adaptable) προγραμμάτων υπολογιστών τα οποία λειτουργούν με βάση την αυτοματοποιημένη ανάλυση συνόλων δεδομένων και όχι τη διαίσθηση των μηχανικών που τα προγραμματίσαν. Η μηχανική μάθηση επικαλύπτεται σημαντικά με τη στατιστική, αφού και τα δύο πεδία μελετούν την ανάλυση δεδομένων.

⁵ **Λογικός Προγραμματισμός:** είναι μια υπολογιστική κωδικοποίηση που συνδυάζει την χρήση της λογική για να αναπαραστήσει τη γνώση και την χρήση του μηχανισμού των αποδείξεων και των συμπερασμάτων για να επεξεργαστεί γνώση. Ο κώδικας σε μια γλώσσα λογικού προγραμματισμού αποτελείται από γεγονότα (facts) και κανόνες (rules).

⁶ **Αναπαράσταση γνώσης:** είναι ένα σύνολο συντακτικών και σημασιολογικών παραδοχών, οι οποίες καθιστούν δυνατή την περιγραφή ενός «κόσμου». Μία μέθοδος αναπαράστασης γνώσης έχει συντακτικό (syntax) και σημασιολογία (semantics). Για τα συστήματα Τεχνητής Νοημοσύνης πρέπει να χρησιμποιηθεί ένας μονοσήμαντος και τυποποιημένος συμβολισμός. Η φυσική γλώσσα είναι ακατάλληλη για αναπαράσταση γνώσης λόγω της πολυσημαντικότητας (ambiguity) και της ερμηνείας με βάση τα συμφραζόμενα (context).

⁷ **Μαθησιακές Δυσκολίες:** είναι ένας γενικός όρος που αναφέρεται σε μια ανομοιογενή ομάδα διαταραχών, οι οποίες εκδηλώνονται με σημαντικές δυσκολίες στην πρόσκτηση και χρήση ικανοτήτων ακρόασης, ομιλίας, ανάγνωσης, γραφής, συλλογισμού ή μαθηματικής ικανότητας.

⁸ **Σχολικός εκφοβισμός (bullying):** είναι ένα φαινόμενο νεανικής παραβατικότητας, αναφέρεται στη χρήση βίας μεταξύ μαθητών ή συνομηλίκων παιδιών, με στόχο να προκληθεί πόνος ή αναστάτωση. Εμφανίζεται με τη μορφή του λεκτικού εκφοβισμού (κοροϊδία, διακρίσεις, ντροπιαστικά σχόλια), του κοινωνικού εκφοβισμού (διάδοση φημών, καταστροφή προσωπικών αντικειμένων, απομόνωση από την ομάδα), του σωματικού εκφοβισμού (χτυπήματα, σπρωξιματα, κλωτσιές) και του ηλεκτρονικού εκφοβισμού (εκβιασμός μέσω Διαδικτύου και ηλεκτρονικού ταχυδρομείου, μέσω μηνυμάτων στο κινητό τηλέφωνο).

γνώσης του έξυπνου βοηθού που προκύπτει από την παρούσα μεταπτυχιακή διατριβή. Θεωρούνται υψίστης σημασίας θέματα που απασχολούν τον τομέα της εκπαίδευσης αφού από αυτά προκύπτουν αρκετά προβλήματα στην εκπαίδευση και την συμπεριφορά των παιδιών τα οποία έχουν επιπτώσεις στην ενήλικη ζωή τους στο μέλλον.



Εικόνα 1. Ιστορικά στοιχεία (Πηγή: <https://www.sas.com/>)

1.1 ΕΡΕΥΝΗΤΙΚΑ ΕΡΩΤΗΜΑΤΑ

Τα ερευνητικά ερωτήματα τα οποία πραγματεύεται η παρούσα διατριβή σχετίζονται περισσότερο με την μελέτη για τις τεχνολογίες που είναι υποψήφιες για χρήση στην δημιουργία του έξυπνου βοηθού (Smart Agent), ποιες είναι κατάλληλες για την υλοποίηση του συστήματος αλλά και με ποιο τρόπο θα μπορέσουν να επικοινωνήσουν γλώσσες λογικού προγραμματισμού (Logic Programming) με γλώσσες που προσφέρονται περισσότερο για εφαρμογές με φιλικό γραφικό περιβάλλον επικοινωνίας⁹ (Graphical User Interface - GUI), αλλά ταυτόχρονα αυτός ο έξυπνος βοηθός θα πρέπει να είναι μια εφαρμογή δικτύου¹⁰ (Network Application).

⁹ **Γραφικό Περιβάλλον Επικοινωνίας (Graphical User Interface - GUI):** καλείται στην πληροφορική ένα σύνολο γραφικών στοιχείων, τα οποία εμφανίζονται στην οθόνη κάποιας ψηφιακής συσκευής (π.χ. Η/Υ) και χρησιμοποιούνται για την αλληλεπίδραση του χρήστη με τη συσκευή αυτή. Παρέχουν στον τελευταίο, μέσω γραφικών, ενδείξεις και εργαλεία προκειμένου αυτός να φέρει εις πέρας κάποιες επιθυμητές λειτουργίες. Για τον λόγο αυτό δέχονται και είσοδο από τον χρήστη και αντιδρούν ανάλογα στα συμβάντα που αυτός προκαλεί με τη βοήθεια κάποιας συσκευής εισόδου (π.χ. πληκτρολόγιο, ποντίκι). Το βασικό συστατικό των GUI είναι τα εικονίδια και είναι απαραίτητη η χρήση κάποιας συσκευής κατάδειξης (pointing device), όπως είναι το ποντίκι στην περίπτωση του Ηλεκτρονικού Υπολογιστή.

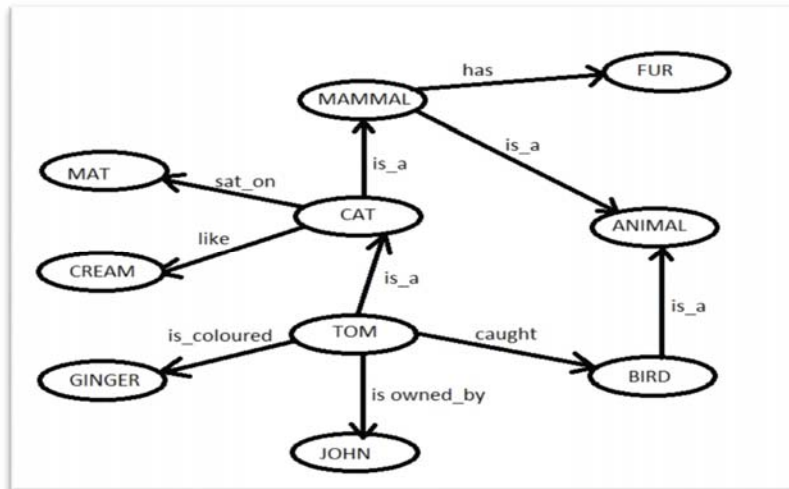
¹⁰ **Εφαρμογή Δικτύου (Network Application):** είναι μια εφαρμογή η οποία απαιτεί σύνδεση δικτύου για να μπορεί να στέλνει και να λαμβάνει δεδομένα. Συνήθως αναφερόμαστε σε εφαρμογές που εφαρμόζουν το μοντέλο αρχιτεκτονικής πελάτη – εξυπηρετητή (client – server), όπου ο πελάτης «αιτείται» και ο εξυπηρετητής «απαντά» ανάλογα με το αίτημα του πελάτη.

Ποιο συγκεκριμένα τα ερευνητικά ερωτήματα αυτά είναι:

- Ερωτήματα που αφορούν την μελέτη τεχνολογιών, εργαλείων και γλωσσών προγραμματισμού που υπάρχουν σήμερα και μπορούν να βοηθήσουν στην υλοποίηση ενός έξυπνου βοηθού. Είναι σημαντικό να γίνει μελέτη και να βρεθεί τρόπος επικοινωνίας των διαφόρων γλωσσών και τεχνολογιών που μετέχουν τελικά στην υλοποίηση του συστήματος του παρόντος έξυπνου βοηθού, καθώς και να γίνει μελέτη που θα βοηθήσει να εξαλειφθούν τυχόν προβλήματα συμβατότητας μεταξύ των εργαλείων, τεχνολογιών και γλωσσών προγραμματισμού¹¹.
- Μελέτη συμπεριφορών μαθητών μέσα από καταγραφή αληθινών ιστοριών που αποτελούν μέρος της γνώσης που κατέχουν κάποιοι εκπαιδευτικοί με σκοπό την δημιουργία κάποιας κεντρικής αποθήκης/βιβλιοθήκης (repository) γεγονότων, την οποία θα μπορεί να επεκτείνει και να μαθαίνει συνεχώς.
- Εύρεση καλύτερου τρόπου συλλογής της γνώσης. Μελέτη τρόπου μεταφοράς της φυσικής γλώσσας σε κανόνες λογικού προγραμματισμού (logic programming) με σκοπό την δημιουργία/επέκταση ενός σημασιολογικού δικτύου¹² (semantic network) που θα αποτελέσει την αποθήκη γνώσης.

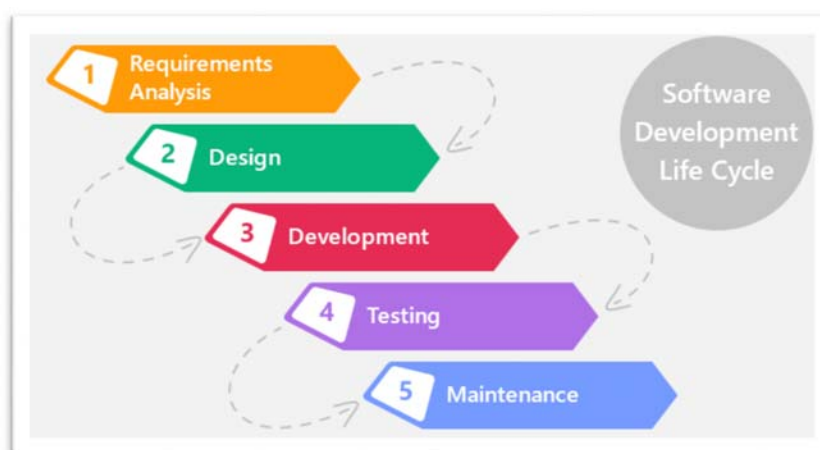
¹¹ **Γλώσσα Προγραμματισμού:** Γλώσσα προγραμματισμού λέγεται μια τεχνητή γλώσσα που μπορεί να χρησιμοποιηθεί για τον έλεγχο μιας μηχανής, συνήθως ενός υπολογιστή. Οι γλώσσες προγραμματισμού ορίζονται από ένα σύνολο συντακτικών και εννοιολογικών κανόνων, που ορίζουν τη δομή και το νόημα των προτάσεων της γλώσσας. Οι γλώσσες προγραμματισμού χρησιμοποιούνται για να διευκολύνουν την οργάνωση και διαχείριση πληροφοριών, αλλά και για την ακριβή διατύπωση αλγορίθμων.

¹² **Σημασιολογικό δίκτυο (semantic network):** Ένα σημασιολογικό δίκτυο (semantic net) αποτελείται από κόμβους (nodes) και δεσμούς (links) ανάμεσά τους. Οι κόμβοι υποδηλώνουν κλάσεις αντικειμένων (classes), αντικείμενα (objects), έννοιες (concepts) και τιμές ιδιοτήτων (values), ενώ οι δεσμοί τις σχέσεις (relations) μεταξύ αυτών των αντικειμένων ή ιδιότητες που συνδέουν αντικείμενα με τιμές.



Εικόνα 2. Παράδειγμα Σημασιολογικού Δικτύου (Πηγή: <http://btechmduis.blogspot.com/2015/07/semantic-nets.html>)

- Ανάπτυξη ενός «έξυπνου βοηθού» για εκπαιδευτικούς με έμφαση στην διαδικασία δημιουργίας ενός έξυπνου βοηθού.
- Δημιουργία φιλικού γραφικού περιβάλλοντος επικοινωνίας (GUI) που να προτρέπει τους εκπαιδευτικούς να χρησιμοποιήσουν την παρούσα εφαρμογή ή τουλάχιστον να είναι αρκετά απλό που να τους διευκολύνει στην χρήση της εφαρμογής.
- Εφαρμογή τεχνικών και βασικών αρχών που διέπουν την ανάπτυξη ενός νέου συστήματος από την συλλογή προδιαγραφών και την σχεδίαση μέχρι την υλοποίηση και την συντήρηση του συστήματος.



Εικόνα 3. Φάσεις Κύκλου Ανάπτυξης Πληροφοριακού Συστήματος

Ανακεφαλαιώνοντας, θα λέγαμε ότι το ζητούμενο είναι η μελέτη και εύρεση των κατάλληλων τεχνολογιών και γλωσσών προγραμματισμού για την δημιουργία ενός συστήματος έξυπνου βοηθού και τέλος η χρήση αυτών που επιλέχθηκαν για την υλοποίηση ενός συστήματος που να μπορεί να μαθαίνει, δηλαδή να καταγράφει νέα γεγονότα και κανόνες και ταυτόχρονα να εξάγει συμπεράσματα οπότε απαιτείται συνεργασία της γλώσσας καταγραφής της γνώσης με αυτή που θα μας προσφέρει το γραφικό περιβάλλον επικοινωνία του συστήματος.

1.2 ΑΝΑΓΚΑΙΟΤΗΤΑ ΚΑΙ ΣΠΟΥΔΑΙΟΤΗΤΑ ΕΡΕΥΝΑΣ

Λόγοι για τους οποίους είναι χρήσιμο να δημιουργηθεί ο «έξυπνος βοηθός» για εκπαιδευτικούς:

- Στα σχολεία της Κύπρου αυτοί την στιγμή εργοδοτούνται αρκετοί πτυχιούχοι όπως γυμναστές, φιλόλογοι, μαθηματικοί, φυσικοί, καθηγητές πληροφορικής κ.α. Άτομα τα οποία ειδικεύονται σε αυτό που διδάσκουν αλλά δεν είναι παιδαγωγοί παρόλο που η νομοθεσία προϋποθέτει την παρακολούθηση κάποιων μαθημάτων παιδαγωγικής και παιδικής ψυχολογίας πριν τον διορισμό του σε δημόσια σχολεία.
- Υπάρχουν άτομα τα οποία διδάσκουν σε απογευματινά φροντιστήρια ή ιδιωτικά σχολεία και δεν έχουν παρακολουθήσει τα μαθήματα παιδαγωγικών που είναι απαραίτητα για εργοδότηση στο δημόσιο, το μοναδικό χαρτί που χρειάζονται είναι το πτυχίο τους.
- Οι δασκάλοι Προδημοτικής και Δημοτικής εκπαίδευσης έχουν παρακολουθήσει μαθήματα σχετικά με παιδαγωγικά και παιδική ψυχολογία κατά την διάρκεια των σπουδών τους. Ακόμη όμως και αυτοί οι οποίοι είναι παιδαγωγοί υπάρχουν στιγμές που χρειάζονται βοήθεια και κάποια καθοδήγηση για τον τρόπο με τον οποίο πρέπει να αντιμετωπίσουν κάποιες ιδιαίτερες συμπεριφορές μαθητών.
- Υπάρχουν εκπαιδευτικοί οι οποίοι είναι νέοι στο επάγγελμα και δεν έχουν την γνώση και τις εμπειρίες που έχει ένας εκπαιδευτικός που ασκεί για χρόνια το συγκεκριμένο επάγγελμα και θα μπορούσαν μέσα από την εφαρμογή που πρόκειται να παραχθεί να βοηθηθούν ενώ ταυτόχρονα κάποιοι θα μπορούν να μεταφέρουν τις γνώσεις τους σε άλλους.

Καθώς το πλήθος της πληροφορίας που αφορά σε εκπαιδευτικά ζητήματα αυξάνεται με ραγδαίους ρυθμούς, καθίσταται όλο και πιο δύσκολο να εντοπισθεί ποιο τμήμα της είναι πραγματικά αξιόλογο και δημιουργεί νέα γνώση προς όφελος των εκπαιδευτικών και των ιδρυμάτων που ασχολούνται με την εκπαίδευση. Το παρόν σύστημα θα αποτελέσει γέφυρα επικοινωνίας μεταξύ των εκπαιδευτικών και μέσα από την συλλογή δεδομένων από εμπειρίες εκπαιδευτικών θα παράγει γνώση η οποία θα είναι χρήσιμο σε όλους τους εκπαιδευτικούς αλλά και σε επιστήμονες που ασχολούνται με το τομέα της εκπαιδευτικής ψυχολογίας. Το σύστημα

που πρόκειται να παραχθεί με την ολοκλήρωση αυτής της διατριβής θα διευκολύνει την εργασία των εκπαιδευτικών σε όποια ηλικιακή ομάδα και αν διδάσκουν (νήπια, παιδιά στο δημοτικό, γυμνάσιο, λύκειο ή ακόμα και φοιτητές πανεπιστημίου), ενώ ταυτόχρονα θα καλυτερεύσουν τις συνθήκες κάτω από τις οποίες λειτουργεί το εκπαιδευτικό σύστημα στον τόπο μας.

Ήδη αναφέρθηκε γιατί θα είναι χρήσιμη η δημιουργία ενός τέτοιου συστήματος στην εκπαιδευτική κοινότητα. Η παρούσα διατριβή επικεντρώνεται περισσότερο στη μελέτη σύγχρονων τεχνολογιών και των δυνατοτήτων τους. Είναι σημαντικό να γίνουν κατανοητές έννοιες που σχετίζονται με τους εμπλεκόμενους τομείς της πληροφορικής θεωρικά για να είναι εφικτή η υλοποίηση ενός τέτοιου συστήματος. Σκοπός είναι η εφαρμογή όσων μελετήθηκαν στα συγκεκριμένα πεδία, καθώς και η επαφή με τις νέες τεχνολογίες που μελετήθηκαν μέσα από αυτήν την εφαρμογή.

ΚΕΦΑΛΑΙΟ 2

ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

Για την υλοποίηση μιας τέτοιας μεταπτυχιακής διατριβής απαιτείται πρώτα να γίνει κάποια βιβλιογραφική μελέτη για να γίνει πλήρως κατανοητό από τον μελετητή/συγγραφέα το αντικείμενο με το οποίο πρόκειται να ασχοληθεί. Σε αυτό το κεφάλαιο της διατριβής παραθέτονται κάποια στοιχεία σχετικά με το υλικό που εντοπίστηκε και μελετήθηκε.

2.1 ΠΕΡΙΟΧΕΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΟΥ ΕΜΠΛΕΚΟΝΤΑΙ

Στα πλαίσια έρευνας για υποστηρικτικό υλικό για τις ανάγκες της παρούσας διατριβής εντοπίστηκε υλικό το οποίο αφορά όλους τους εμπλεκόμενους τομείς/θέματα της πληροφορικής.

Οι τομείς αυτοί είναι οι ακόλουθοι:

- Artificial Intelligence (Τεχνητή νοημοσύνη)
- Cognitive Programming (Γνωστικός Προγραμματισμός)
- Cognitive Reasoning and Learning Mechanisms (Γνωστική Αιτιολόγηση και Μηχανισμοί μάθησης)
- Logic Programming (Λογικός Προγραμματισμός)
- Machine Learning (Μηχανική Μάθηση)
- Εξόρυξη Δεδομένων¹³
- Human Computer Interaction (Επικοινωνία Ανθρώπου και Υπολογιστή)

¹³ **Εξόρυξη δεδομένων** (ή ανακάλυψη γνώσης από βάσεις δεδομένων - data mining): είναι η εξεύρεση μιας (ενδιαφέρουσας, αυτονόητης, μη προφανούς και πιθανόν χρήσιμης) πληροφορίας ή προτύπων από μεγάλες βάσεις δεδομένων με χρήση αλγορίθμων ομαδοποίησης ή κατηγοριοποίησης και των αρχών της στατιστικής, της τεχνητής νοημοσύνης, της μηχανικής μάθησης και των συστημάτων βάσεων δεδομένων. Στόχος της εξόρυξης δεδομένων είναι η πληροφορία που θα εξαχθεί και τα πρότυπα που θα προκύψουν να έχουν δομή κατανοητή προς τον άνθρωπο έτσι ώστε να τον βοηθήσουν να πάρει τις κατάλληλες αποφάσεις.

- System Analysis (Ανάλυση Συστημάτων¹⁴)
- Software Engineering (Τεχνολογία Λογισμικού¹⁵)

Η ενδεικτική βιβλιογραφία που εντοπίστηκε στα πλαίσια έρευνας που πρόκειται να βοηθήσει στην υλοποίηση του παρόντος λογισμικού (έξυπνου βοηθού) ορίζει τις πιο πάνω περιοχές της πληροφορικής, παρουσιάζει τα κύρια χαρακτηριστικά τους ενώ βοήθησε να εντοπιστούν ποια στοιχεία απο τον κάθε τομέα που αναφέραμε πιο πάνω πρόκειται να εφαρμοστούν. Επίσης, από το υλικό που μελετήθηκε διευκρινίστηκε η έννοια του σημασιολογικού δικτύου, η έννοια του έμπειρου συστήματος σαν τομείς της επιστήμης της Τεχνητής Νοημοσύνης μέσα απο διάφορα βιβλία, ερευνητικά άρθρα και σελίδες στο διαδίκτυο.

Μερικές απο τις περιοχές που μελετήθηκαν εντάσσονται σε μια άλλη μεγαλύτερη απο αυτές που αναφέραμε. Για παράδειγμα, το Cognitive Programming και το Logic Programming εντάσσονται στην ευρύτερη περιοχή της τεχνητής νοημοσύνης. Το Cognitive Reasoning and Learning Mechanisms αποτελούν τμήματα της Μηχανικής Μάθησης, όπου με τον όρο Μηχανική Μάθηση γίνεται αναφοράστο τομέας της πληροφορικής που διερευνά τη μελέτη και την κατασκευή αλγορίθμων που μπορούν να μαθαίνουν από τα δεδομένα και να κάνουν προβλέψεις σχετικά με αυτά. Η Μηχανική μάθηση είναι και αυτή ένα υποπεδίο της Τεχνητής Νοημοσύνης.

Όσον αφορά τη Τεχνητή νοημοσύνη θα λέγαμε ότι είναι ένα από τα πιο νέα ερευνητικά πεδία της επιστήμης της Πληροφορικής. Η διεξοδική αναζήτηση στη βιβλιογραφία οδηγεί σε μια σειρά από ορισμούς της Τεχνητής Νοημοσύνης. Σύμφωνα με τους Barr και Feigenbaum, η Τεχνητή Νοημοσύνη είναι ο τομέας της επιστήμης των υπολογιστών, που ασχολείται με τη σχεδίαση ευφυών (νοημόνων) υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζονται με την ανθρώπινη νοημοσύνη και την ανθρώπινη συμπεριφορά. Γενικά, η Τεχνητή Νοημοσύνη μελετά και δημιουργεί συστήματα που σκέπτονται σαν τον άνθρωπο, συστήματα που σκέπτονται ορθολογικά, συστήματα που ενεργούν σαν τον άνθρωπο ή συστήματα που ενεργούν ορθολογικά. Ακολουθεί αναφορά και σχολιασμός για μερικούς απο τους ορισμούς που δόθηκαν για τις έννοιες αυτές ξεκινώντας από την έννοια της

¹⁴ **Ανάλυση Συστημάτων (System Analysis):** αποτελεί ένα αντικείμενο της Πληροφορικής που ασχολείται με τις τεχνικές ανάλυσης και σχεδίασης συστημάτων πληροφόρησης, για τις ανάγκες ενός Οργανισμού ή μιας Επιχείρησης. Λέγοντας σύστημα εννοούμε ένα σύνολο αλληλεπιδρόντων μεταξύ τους συστατικών στοιχείων (άνθρωποι - μηχανές - διαδικασίες που διέπονται από συγκεκριμένους νόμους). Τα συστατικά αυτά στοιχεία παράγουν πληροφορίες , χρησιμοποιώντας τα δεδομένα εισόδου, με βάση μια προκαθορισμένη διαδικασία. Ο αναλυτής συστημάτων προσπαθεί να αναλύσει ένα σύστημα καθορίζοντας τα όριά του, τους νόμους και τις διαδικασίες που το συνθέτουν. Με τον τρόπο αυτό ελέγχει την αποτελεσματικότητά του ώστε να μπορεί να τη βελτιώσει τροποποιώντας ή προσθέτοντας νέες διαδικασίες.

¹⁵ **Τεχνολογία Λογισμικού (Software Engineering):** είναι ο κλάδος της πληροφορικής που ασχολείται με τη μελέτη και την εφαρμογή συστηματικών, μεθοδικών και ποσοτικοποιημένων προσεγγίσεων για την ανάπτυξη, λειτουργία και συντήρηση του λογισμικού.

Τεχνητής Νοημοσύνης (Πίνακας 1) που αποτελεί τον βασικό τομέα της πληροφορικής που εμπλέκεται στην εκπόνηση αυτής της εργασίας.

Τεχνητή Νοημοσύνη

«Η συναρπαστική νέα προσπάθεια για να κάνουμε τους υπολογιστές να σκέπτονται... μηχανές με νόηση, με πλήρη και κυριολεκτική έννοια» (Haugeland, 1985).

«Η αυτοματοποίηση των δραστηριοτήτων που συσχετίζουμε με την ανθρώπινη σκέψη, όπως η λήψη αποφάσεων, η επίλυση προβλημάτων, η μάθηση...» (Bellman, 1978)

«Η τεχνη της δημιουργίας μηχανών που πραγματοποιούν λειτουργίες οι οποίες απαιτούν νοημοσύνη όταν πραγματοποιούνται απο ανθρώπους.» (Kuzweil, 1990)

«Η μελέτη του πώς μπορούμε να κάνουμε τους υπολογιστές να κάνουν πράγματα στα οποία προς το παρόν οι άνθρωποι είναι καλύτεροι» (Rich και Knight, 1991)

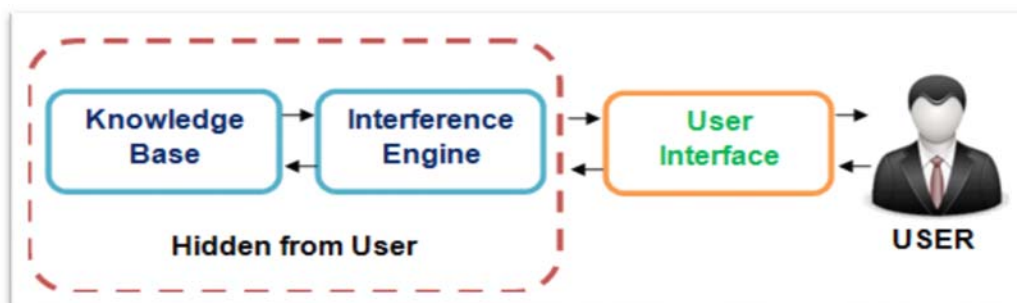
«Η μελέτη των νοητικών ικανοτήτων με τη χρήση υπολογιστικών μοντέλων» (Charniak και McDermott, 1985)

«Η μελέτη των υπολογιστικών εργασιών που μας δίνουν τη δυνατότητα να αντιλαμβανόμαστε, να συλλογιζόμαστε και να ενεργούμε.» (Winston, 1992)

«Η τεχνητή νοημοσύνη ασχολείται με την ευφυή συμπεριφορά των τεχνουργημάτων» (Nilsson, 1998)

Πίνακας 1. Μερικοί ορισμοί της Τεχνητής Νοημοσύνης

Στην συνέχεια γίνεται αναφορά στην έννοια του εννοιολογικού δικτύου. Ένα σημασιολογικό δίκτυο (semantic network) είναι μια κλασική τεχνική αναπαράστασης γνώσης της Τεχνητής Νοημοσύνης. Τα δίκτυα αυτά αναπτύχθηκαν αρχικά με σκοπό την αναπαράσταση της ανθρώπινης μνήμης και γνώσης καθώς και την κατανόηση της γλώσσας στα πλαίσια της επιστήμης της Τεχνητής Νοημοσύνης. Η δομή ενός Semantic Network απεικονίζεται γραφικά με τη χρήση κόμβων και τόξων που τα συνδέουν. Οι κόμβοι συνήθως ονομάζονται αντικείμενα και τα τόξα καλούνται συνδέσμοι ή ακμές. Οι σύνδεσμοι αυτοί χρησιμοποιούνται για να δηλώσουν συσχέτιση μεταξύ δύο κόμβων. Οι κόμβοι με την σειρά τους χρησιμοποιούνται για να αναπαραστήσουν αντικείμενα, έννοιες και συγκεκριμένες καταστάσεις.



Εικόνα 4. Τα μέρη ενός Έμπειρου συστήματος

Επίσης μελετήθηκε η έννοια του έξυπνου (έμπειρου) συστήματος. Υπάρχει πληθώρα ορισμών για τα έμπειρα συστήματα που εξαρτώνται από τη γωνία, που ο καθένας τα προσεγγίζει. Στην Εικόνα 4, η οποία φτιάχτηκε στο Microsoft Power Point παρουσιάζονται τα μέρη ενός έμπειρου συστήματος. Επομένως, είναι εμφανές ότι τα μέρη του έμπειρου συστήματος όπως παρουσιάζονται στην Εικόνα 4 είναι η Βάση γνώσης¹⁶ (Knowledge Base), ο μηχανισμός εξαγωγής συμπερασμάτων¹⁷ (Inference Engine) και το περιβάλλον διεπαφής με τον χρήστη (User Interface) του έμπειρου συστήματος.

Έμπειρα Συστήματα

Ο Feigenbaum (1982) θεωρεί ότι ένα έμπειρο σύστημα είναι ένα έξυπνο πρόγραμμα το οποίο για να λύσει κάποιο πρόβλημα, που είναι αρκετά δύσκολο ώστε να απαιτεί σημαντική εμπειρία για την επίλυσή του, χρησιμοποιεί ειδική γνώση και διαδικασίες εξαγωγής συμπερασμάτων. Η γνώση που είναι απαραίτητη για τη λειτουργία σε αυτό το επίπεδο, μαζί με τις διαδικασίες εξαγωγής συμπερασμάτων, μπορεί να θεωρηθούν σαν ένα μοντέλο εμπειρίας των καλύτερων ειδικών του χώρου.

Ο Hart (1986) εκτιμά ότι τα Έμπειρα Συστήματα είναι προγράμματα, τα οποία εκτελούν εργασίες τις οποίες συνήθως τις κάνουν ειδικοί. Αυτά ενσωματώνουν τη γνώση ειδικών και την ικανότητά τους να χρησιμοποιούν αυτή τη γνώση για να επιλύουν προβλήματα. Τα προγράμματα αυτά περιορίζονται από τον τύπο των εργασιών που μπορούν να εκτελέσουν, αλλά παρουσιάζουν την εξειδίκευσή τους κατά την προσέγγιση κατάλληλων προβλημάτων. Σε τέτοια προγράμματα η γνώση κωδικοποιείται δημιουργώντας έτσι ισχυρά εργαλεία.

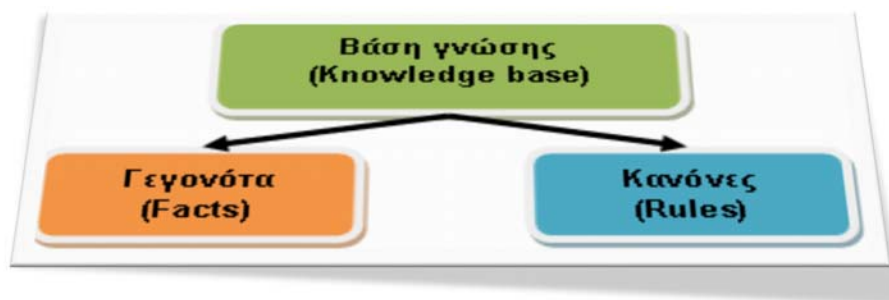
Οι Curry και Moutinho (1991) ορίζουν ως έμπειρα συστήματα τα προγράμματα με τα οποία γίνεται προσπάθεια να αναπαρασταθεί η συμπεριφορά ενός ειδικού κάποιου τομέα. Τα συστήματα αυτά διαθέτουν γνώση και εμπειρία και είναι ικανά με τη χρήση λογικής να παρέχουν συμβουλές ή να παίρνουν αποφάσεις έχοντας την ικανότητα να αιτιολογούν το πώς οδηγούνται σε αυτές.

¹⁶ **Βάση γνώσης (Knowledge Base):** είναι ένα σύνολο από αναπαραστάσεις και γεγονότα σχετικά με τον κόσμο και η **Γλώσσα Αναπαράστασης Γνώσης (Knowledge Representation Language)** που χρησιμοποιείται σε μια βάση γνώσης είναι μια γλώσσα της οποίας οι προτάσεις αντιπροσωπεύουν γεγονότα σχετικά με τον κόσμο.

¹⁷ **μηχανισμός εξαγωγής συμπερασμάτων (inference mechanism):** είναι ένας μηχανισμός που προσδιορίζει τι συνεπάγεται από την πληροφορία που έχει προστεθεί στη βάση γνώσης. Οι ερωτήσεις που γίνονται στη βάση γνώσης χρησιμοποιούν αυτό τον μηχανισμό.

Οι Doukidis και Paul (1992), συνδυάζοντας το τι και το πώς κάνει ένα έμπειρο σύστημα ορίζουν ότι έμπειρο σύστημα είναι ένα πρόγραμμα που βοηθά το χρήστη, παρέχοντας πληροφορίες σε κάποιο ιδιαίτερο πεδίο. Αυτό το πετυχαίνει χειριζόμενο πληροφορίες, σχετικές με ένα χώρο, που παρέχονται από μια σειρά από "ειδικούς" του χώρου αυτού. Ένα άλλο σημαντικό χαρακτηριστικό ενός έμπειρου συστήματος είναι ότι αυτό έχει τη δυνατότητα να εξηγεί/αιτιολογεί τις χρησιμοποιηθείσες μεθόδους για την παροχή πληροφοριών

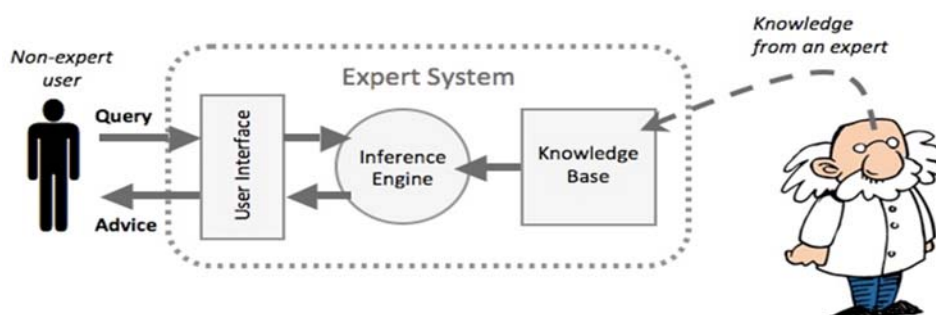
Πίνακας 2. Μερικοί ορισμοί για τα έμπειρα συστήματα



Εικόνα 5. Η βάση γνώσης ενός έμπειρου συστήματος αποτελείται από γεγονότα και κανόνες

Κάποια άλλα βασικά στοιχεία για τα έμπειρα συστήματα επεξηγούν την δομή ενός έμπειρου συστήματος. Ένα τέτοιο σύστημα αποτελείται από 3 μέρη:

1. Την **βάση γνώσης (Knowledge base)** που αποτελείται και αυτή με την σειρά της από κανόνες και γεγονότα.
2. Τον μηχανισμό εξαγωγής συμπερασμάτων (**inference engine**)
3. Και την διεπαφή με τον χρήστη (**User Interface**)



Εικόνα 6. Τα μέρη ενός έμπειρου συστήματος και η σχέση του με τους ανθρώπους

Το Human Computer Interaction (Επικοινωνία Ανθρώπου και Υπολογιστή) μελετάει το σχεδιασμό, ανάπτυξη και αξιολόγηση διαδραστικών υπολογιστικών συστημάτων. Επικεντρώνεται δηλαδή στο πώς πρέπει να είναι το γραφικό περιβάλλον επικοινωνίας ενός συστήματος για καλύτερη επικοινωνία μεταξύ Άνθρωπου και Υπολογιστή. Ενω, η Ανάλυση συστημάτων περιγράφει ένα σύστημα όχι απαραίτητα μηχανογραφημένο και η Τεχνολογία Λογισμικού ασχολείται με την διαδικασία ανάπτυξης λογισμικού συστήματος και τις μεθόδους που ακολουθούνται σε κάθε φάση της ανάπτυξης ενός λογισμικού.

2.2 ΤΕΧΝΟΛΟΓΙΕΣ, ΠΕΡΙΒΑΛΛΟΝΤΑ ΚΑΙ ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Επίσης, υλικό εντοπίστηκε για τις πιθανές τεχνολογίες, εργαλεία και γλώσσες προγραμματισμού που ίσως χρησιμοποιηθούν για την φάση της υλοποίησης του συστήματος έξυπνου βοηθού.

- Visual Studio
- C#
- NetBeans
- Java
- JSP
- Prolog

Οι τεχνολογίες που αναφέρονται/προτείνονται στο έγγραφο αυτό είναι ευρέως χρησιμοποιημένες και «κουβαλούν» αρκετή βιβλιογραφία. Υπάρχουν βιβλία που εκπαιδεύουν αρχάριους βήμα βήμα να τις χρησιμοποιήσουν, υπάρχουν βιβλία που περιγράφουν τα βασικά χαρακτηριστικά και τις δυνατότητές τους, καθώς και το είδος των εφαρμογών που μπορούν να δημιουργηθούν χρησιμοποιώντας αυτές τις τεχνολογίες.

Πιο συγκεκριμένα, το Visual Studio είναι ένα λογισμικό της Microsoft στο οποίο μπορείς να αναπτύξεις εφαρμογές σε διάφορες γλώσσες προγραμματισμού και μία γλώσσα που υποστηρίζει είναι η C#. Το Visual Studio, είναι ένα εύχρηστο περιβάλλον ανάπτυξης εφαρμογών, που αναπτύχθηκε από την εταιρεία Microsoft Corporation. Υποστηρίζει την ανάπτυξη προγραμμάτων σε κονσόλα, οπτικές εφαρμογές, ιστοσελίδες, υπηρεσίες WEB και άλλα. Το περιβάλλον ανάπτυξης Visual Studio βοηθά τον προγραμματιστή να αναπτύξει τα προγράμματά του με σχετική ευκολία, καθώς η τεχνολογία Microsoft IntelliSense βοηθά τον προγραμματιστή να κατανοήσει με ευκολία πιθανά λάθη του κώδικά του, υπογραμμίζοντας τα με κόκκινη γραμμή δυναμικά κατά το χρόνο συγγραφής του προγράμματος. Αυτή η τεχνολογία είναι ικανή να εντοπίσει λάθη τα οποία μπορεί να είναι είτε συντακτικά, όπως για παράδειγμα η χρήση μιας εντολής με εσφαλμένο τρόπο, είτε λογικά, όπως για παράδειγμα η δήλωση ενός αντικειμένου χωρίς να αυτό να χρησιμοποιείται. Το Visual Studio υποστηρίζει την ανάπτυξη προγραμμάτων στις C++, C#, Visual Basic, F# και τη μεταφορά προγραμμάτων από την μία γλώσσα στην άλλη.

Με άλλα λόγια συγγραφεί ένα πρόγραμμα σε γλώσσα C++ μπορεί απλά και εύκολα να μετατραπεί αυτόματα σε κάποια εκ των γλωσσών που υποστηρίζει το Visual Studio.

Το NetBeans είναι ένα ανοιχτής πηγής (open source) ενσωματωμένο περιβάλλον ανάπτυξης εφαρμογών. Αρχικά χρησιμοποιήθηκε για την ανάπτυξη εφαρμογών σε Java αλλά πλέον χρησιμοποιείται και για άλλες γλώσσες και τεχνολογίες όπως η JSP (Java Server Pages). Η Sun Microsystems ίδρυσε το ερευνητικό έργο ανοιχτής πηγής NetBeans τον Ιούνιο του 2000 και συνεχίζει να είναι ο κύριος ανάδοχος.

Η C#, Java, JSP και Prolog είναι γλώσσες προγραμματισμού για διαφορετικού τύπου εφαρμογές η κάθε μία από αυτές. Η Prolog είναι μια γλώσσα Λογικού προγραμματισμού η οποία περιγράφει την λογική σκέψη με την χρήση κανόνων της μορφής «Αν ισχύει αυτό και εκείνο τότε εξάγεται το συμπέρασμα τάδε». Η C# είναι μια γλώσσα προγραμματισμού στην οποία μπορείς εύκολα να φτιάξεις ένα γραφικό περιβάλλον επικοινωνίας(GUI)και είναι κατάλληλη για εφαρμογές που τρέχουν σε περιβάλλον Windows,είτε πρόκειται για δυκτιακές εφαρμογές, είτε όχι.Η Javaείναι μια γλώσσα κατάλληλη για περιπτώσεις αντικειμενοστρεφούςπρογραμματισμού (Object-Oriented programming), ενώ η JSP (JavaServer Pages) είναι τεχνολογία/γλώσσα κατάλληλη για εφαρμογές ιστού (web-applications) όπου συνδιάζονται τμήματα κώδικα html (hypertext markup language) και για κάποια πιο δυναμικά μέρη και υπολογιστικά μέρη των δικτυακών εφαρμογών Java.

Ανακεφαλαιώνοντας, το υλικό που μελετήθηκε με σκοπό την ολοκλήρωση αυτής της εργασίας αναφέρεται στην Τεχνητής Νοημοσύνης και εννοιών που σχετίζονται με αυτή την περιοχή της πληροφορικής, αλλά και με περιοχές που αφορούν την ανάπτυξη ενός νέου συστήματος όπως είναι η Τεχνολογία Λογισμικού και η Ανάλυση Συστημάτων.

ΚΕΦΑΛΑΙΟ 3

ΜΕΘΟΔΟΛΟΓΙΑ

Για την ολοκλήρωση της παρούσας διατριβής γίνεται συνδυασμός ποσοτικής¹⁸ και ποιοτικής¹⁹ προσέγγισης. Έγινε μελέτη για τον καταλληλότερο τρόπο να συλλεγεται η γνώση και πώς αυτή η γνώση είναι εφικτό να μετατρέπεται σε κανόνες. Επίσης έγινε μελέτη ανάμεσα σε τεχνολογίες που προτάθηκαν για εφαρμογή και χρήση κατά την υλοποίηση κάθε τμήματος του συστήματος, εντοπίστηκαν οι δυνατότητες και η αδυναμίες τις κάθε τεχνολογίας και επιλέγηκαν οι κατάλληλες τεχνολογίες για το σύστημα. Στην συνέχεια γίνεται ανάπτυξη του συστήματος με τις επιλεχθείσες τεχνολογίες, εργαλεία, γλώσσες προγραμματισμού και διαδικασίες. Πρόκειται για μια δικτυακή εφαρμογή (Network Application) για χρήση από ηλεκτρονικό υπολογιστή.

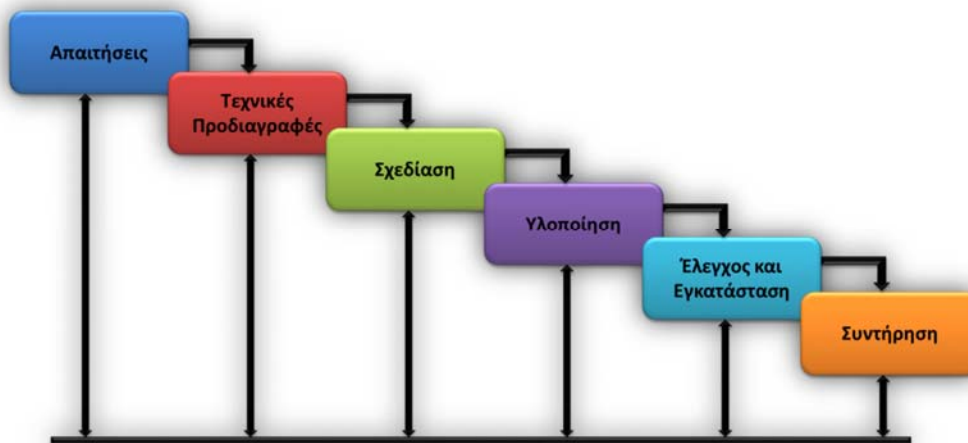
Πριν αρχίσει η διαδικασία ανάπτυξης ενός συστήματος πρέπει να επιλεγεί ποια μεθοδολογία και ποιο μοντέλο ανάπτυξης συστημάτων θα ακολουθηθεί για να οργανώσει τις διαδικασίες που πρόκειται να ακολουθηθούν. Αυτό υπάγεται στον τομέα της Πληροφορικής που λέγεται

¹⁸ **Ποσοτική έρευνα:** Σκοπός της ποσοτικής έρευνας είναι η εύρεση σχέσεων μεταξύ διαφόρων παραγόντων. Η ποσοτική έρευνα αναφέρεται στη συστηματική διερεύνηση φαινομένων με στατιστικές μεθόδους, μαθηματικά μοντέλα και αριθμητικά δεδομένα. Χρησιμοποιείται συνήθως αντιπροσωπευτικό δείγμα παρατηρήσεων και επιδιώκεται γενίκευση σε ένα ευρύτερο πληθυσμό. Η συλλογή δεδομένων γίνεται με δομημένα πρωτόκολλα, όπως ερωτηματολόγια, κλίμακες και δοκίμια επιτευγμάτων. Οι Μικτές προσεγγίσεις συνδυάζουν ποσοτικές και ποιοτικές μεθόδους κατά το μεθοδολογικό τους σχεδιασμό για να αξιοποιούν καλύτερα τα πλεονεκτήματα της κάθε μεθόδου και αντιμετωπίζουν αποτελεσματικότερα τις αδυναμίες της κάθε μιας.

¹⁹ **Ποιοτικής έρευνα:** Μέσω της ποιοτικής έρευνας συλλέγονται δεδομένα τα οποία περιγράφουν προβλήματα κι έννοιες από τη ζωή των ατόμων. Τα δεδομένα μπορούν να προέλθουν από συνεντεύξεις, παρατηρήσεις, συμμετοχικές παρατηρήσεις, ιστορίες, αλληλεπιδράσεις, περιπτωσιολογικές μελέτες, προσωπικές εμπειρίες, ιστορίες ζωής, αναλύσεις αρχείων, οπτικό υλικό κι ενδοσκοπήσεις. Η ποιοτική έρευνα έχει δύο βασικά μοναδικά χαρακτηριστικά. Το πρώτο είναι ότι ο ερευνητής αποτελεί το μέσο με το οποίο διεξάγεται η έρευνα, και το δεύτερο είναι ότι ο κύριος σκοπός της είναι να διερευνήσει κάποιες πλευρές του κοινωνικού συστήματος που μελετά.

Τεχνολογία Λογισμικού²⁰ (Software Engineering). Ο έξυπνος βοηθός είναι επίσης ένα είδος πληροφοριακού συστήματος, έτσι θα ακολουθήσει τις φάσεις του κύκλου ανάπτυξης πληροφοριακών συστημάτων όπως όλα τα πληροφοριακά συστήματα.

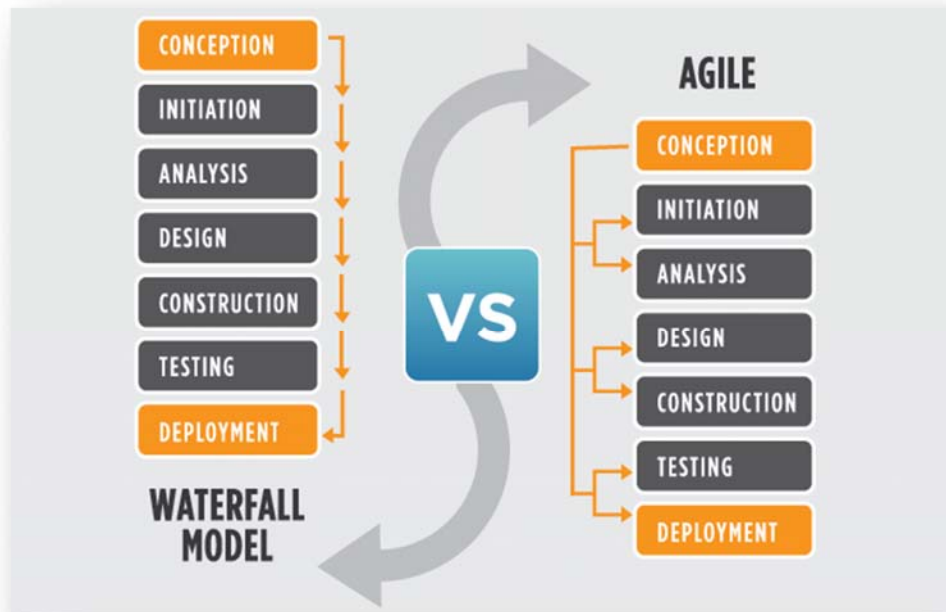
3.1 ΜΟΝΤΕΛΟ ΚΥΚΛΟΥ ΑΝΑΠΤΥΞΗΣ ΣΥΣΤΗΜΑΤΟΣ



Εικόνα 7. Μοντέλο Καταρράκτη

Το μοντέλο ανάπτυξης που είχε αρχικά επιλεγεί για να εφαρμοστεί ήταν το μοντέλο καταρράκτη (Waterfall lifecycle model) γιατί το συγκεκριμένο μοντέλο ανάπτυξης διασφαλίζει ότι προτού αρχίσει η επόμενη φάση του κύκλου ανάπτυξης έχει ολοκληρωθεί και ελεγχθεί η προηγούμενη ενώ ταυτόχρονα μας δίνει την δυνατότητα να επιστρέψουμε σε όποια φάση κριθεί αναγκαίο και απο όποια φάση και αν βρίσκεται η διαδικασία ανάπτυξης την δεδομένη στιγμή. Στην πορεία όμως προτάθηκε και εφαρμόστηκε ένα ευέλικτο μοντέλο (agile), το οποίο είναι ένα μοντέλο ταχείας ανάπτυξης αρκετά νεότερο από τον καταρράκτη και φάνηκε να είναι για την ολοκλήρωση της μεταπτυχιακής διατριβής.

²⁰ **Τεχνολογία Λογισμικού (Software engineering):** είναι η περιοχή εκείνη της επιστήμης της πληροφορικής η οποία ασχολείται με την εύρεση και θεμελίωση μεθόδων για να περιγράφεται, να κατασκευάζεται/αναπτύσσεται και να συντηρείται λογισμικό.



Εικόνα 8. Φάσεις Μοντέλου Καταρράκτη και Ευέλικτου Μοντέλου (Πηγή: <https://www.seguetech.com/waterfall-vs-agile-methodology/>)

Το μοντέλο Καταρράκτης είναι μια γραμμική/σειριακή προσέγγιση στην ανάπτυξη Λογισμικού, στην οποία μπορεί όμως να επιστρέψει σε οποιαδήποτε φάση χρειαστεί και ακολούθως να προχωρήσει και πάλι στις επόμενες φάσεις, ενώ κάθε φάση/στάδιο ανάπτυξης λογισμικού ολοκληρώνεται πριν την έναρξη του επόμενου σταδίου και χρειάζεται έγκριση από τον «πελάτη». Οπότε γίνεται αντιληπτό ότι υπάρχουν θετικά και αρνητικά στοιχεία στο μοντέλο Καταρράκτη. Τα θετικά αυτά χαρακτηριστικά είναι:

- Οι προγραμματιστές και οι πελάτες συμφωνούν για το τι θα παραδοθεί νωρίτερα στον αναπτυξιακό κύκλο ζωής. Αυτό κάνει τον προγραμματισμό και το σχεδιασμό πιο απλό.
- Η πρόοδος μετράται ευκολότερα, καθώς το πλήρες φάσμα των εργασιών είναι γνωστό εκ των προτέρων.
- Εκτός από τις αναθεωρήσεις και τις εγκρίσεις η παρουσία του πελάτη δεν απαιτείται αυστηρά μετά τη φάση των απαιτήσεων.
- Επειδή ο σχεδιασμός ολοκληρώνεται νωρίς στον αναπτυξιακό κύκλο ζωής, η προσέγγιση αυτή προσφέρεται σε έργα όπου πολλά στοιχεία λογισμικού πρέπει να σχεδιαστούν για την ενσωμάτωση με εξωτερικά συστήματα.
- Τέλος, το λογισμικό μπορεί να σχεδιαστεί πλήρως και πιο προσεκτικά, με βάση την πληρέστερη κατανόηση όλων των παραδοτέων λογισμικού.

Τα αρνητικά στοιχεία του μοντέλου Καταρράκτη είναι τα πιο κάτω:

- Ένας τομέας ο οποίος σχεδόν πάντα υπολείπεται είναι η αποτελεσματικότητα των απαιτήσεων. Η συλλογή και η τεκμηρίωση των απαιτήσεων με τρόπο που έχει νόημα για έναν πελάτη είναι συχνά το πιο δύσκολο κομμάτι της ανάπτυξης λογισμικού, κατά τη γνώμη μου. Οι πελάτες μερικές φορές εκφοβίζονται από λεπτομέρειες και απαιτούνται συγκεκριμένες λεπτομέρειες, που παρέχονται νωρίς στο έργο, με αυτήν την προσέγγιση. Επιπλέον, οι πελάτες δεν είναι πάντα σε θέση να απεικονίσουν μια εφαρμογή από ένα έγγραφο απαιτήσεων.
- Ένα άλλο πιθανό μειονέκτημα της καθαρής ανάπτυξης Waterfall είναι η πιθανότητα ότι ο πελάτης θα είναι δυσαρεστημένος με το παραδιδόμενο προϊόν λογισμικού. Καθώς όλα τα παραδοτέα βασίζονται σε τεκμηριωμένες απαιτήσεις, ο πελάτης μπορεί να μην δει τι θα παραδοθεί μέχρι να ολοκληρωθεί. Μέχρι τότε, οι αλλαγές μπορεί να είναι δύσκολες (και δαπανηρές) για την εφαρμογή τους.

Ένα ευέλικτο μοντέλο (agile), είναι περισσότερο διαδραστικό και οι πελάτες συνεργάζονται με τους προγραμματιστές και εμπλέκονται περισσότερο στην διαδικασία ανάπτυξης του συστήματος. Αυτή η προσέγγιση τονίζει την ταχεία παράδοση μιας εφαρμογής με πλήρη λειτουργικά στοιχεία. Όπως αντιλαμβάνεστε υπάρχουν κάποια θετικά χαρακτηριστικά στο ευέλικτο μοντέλο που είναι εμφανής, όπως τα πιο κάτω:

- Ο πελάτης έχει συχνές ευκαιρίες να έρθει σε επαφή με το σύστημα που παράγεται και να λαμβάνει αποφάσεις για αλλαγές στο αναπτυξιακό έργο.
- Ο πελάτης αποκτά έντονη αίσθηση ιδιοκτησίας, αφού συμμετέχει άμεσα στην παραγωγή του συστήματος, δουλεύοντας εκτεταμένα και άμεσα με την ομάδα του έργου καθ' όλη τη διάρκεια του έργου.
- Η ανάπτυξη συχνά επικεντρώνεται περισσότερο στο χρήστη, πιθανότατα αυτό να είναι αποτέλεσμα συχνότερης κατεύθυνσης από τον πελάτη.
- Ξεν είναι απαραίτητο να ακολουθηθούν όλα τα στάδια ανάπτυξης με την σειρά όπως στο παραδοσιακό μοντέλο του καταρράκτη.

Συμπερασματικά, καταλήγουμε στο γεγονός ότι η ευέλικτη προσέγγιση είναι πολύ διαφορετική από αυτήν του μοντέλου καταρράκτη και δεν είναι εύκολη η επιλογή μεταξύ τους. Ο καταρράκτης θα μπορούσε να εφαρμοστεί σε σχεδόν οποιοδήποτε είδος έργου, ενώ στο ευέλικτο μοντέλο πρέπει να υπάρχει ευελιξία γύρω από τις απαιτήσεις και την σειρά με την οποία θα διεκπεραιωθούν κάποιες διαδικασίες για την ολοκλήρωση του έργου.

Η κεντρική ιδέα του μοντέλου του καταρράκτη είναι ότι το σύστημα λογισμικού αναπτύσσεται περνώντας ολόκληρο από διαδοχικές επιμέρους φάσεις, καθεμία από τις οποίες θεωρείται

περατωμένη με την παραγωγή ορισμένων συστατικών λογισμικού. Κάθε επιμέρους φάση ολοκληρώνεται με μια εργασία επαλήθευσης/επικύρωσης των προϊόντων της κατά την οποία αποφασίζεται η μετάβαση ή όχι στην επόμενη. Το λογισμικό εμφανίζεται πλήρες, δηλαδή με όλα τα λειτουργικά του χαρακτηριστικά, από την επιμέρους φάση της συνένωσης και μετά. Χαρακτηριστικό του μοντέλου του καταρράκτη είναι ότι για να ξεκινήσει μία φάση πρέπει να έχει ολοκληρωθεί πλήρως η προηγούμενη. Η ανάπτυξη με τον τρόπο αυτό χαρακτηρίζεται ακολουθιακή, διότι οι επιμέρους φάσεις από τις οποίες διέρχεται είναι διακριτές και ακολουθούν η μία την άλλη.

Στην περίπτωση της παρούσας διατριβής εργασίας κρίθηκε πιο κατάλληλο το ευέλικτο μοντέλο, αφού εφαρμόζοντάς το είχα την δυνατότητα να προασμόσω κάποιες από τις απαιτήσεις του συστήματος στις δικές μου δυνατότητες αλλά και των μελλοντικών μου χρηστών.

3.2 ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ

Όπως αναφέραμε σε προηγούμενο σημείο της παρούσας έκθεσης μεταπτυχιακής διατριβής προτάθηκαν κάποιες γλώσσες προγραμματισμού διαφορετικών κατηγοριών για την υλοποίηση του συστήματος. Για να διατηρηθεί όμως σε λειτουργία το mini Expert System που είχαμε ήδη παράξει σε προηγούμενο μάθημα του μεταπτυχιακού προγράμματος, έπρεπε να διατηρηθεί η χρήση της Prolog για την διαχείριση της γνώσης, δηλαδή στο κομμάτι που αφορά τον λογικό προγραμματισμό. Αυτό είχε ως αποτέλεσμα να αφιερωθεί αρκετός χρόνος στο ψάξιμο τρόπου «επικοινωνίας» μεταξύ:

- Java με Prolog ίσως με την χρήση του εργαλείου NetBeans που αναφέραμε πιο πάνω και
- C# με Prolog με την χρήση του πακέτου λογισμικού ανάπτυξης συστημάτων του Visual Studio.



Εικόνα 9. Εργαλεία και γλώσσες που επιλέχθηκαν

Αυτό έγινε για να επιλεγεί ποια θα ήταν η 2^η γλώσσα προγραμματισμού που θα χρησιμοποιείτω στην υλοποίηση του project, αλλά και με την χρήση ποιών τεχνολογιών και εργαλείων θα είτω εφικτή η υλοποίηση. Σε αυτή την φάση και αφού είχε ήδη χαθέ αρκετός χρόνος σε επίπεδο μελέτης και έρευνας επιλέχθηκε η χρήση της C# σε συνδιασμό με Prolog, όπου η Prolog δεν θα είναι εμφανές ότι χρησιμοποιείται και για την αποθήκευση κάποιων επιπρόσθετων στοιχείων που απαιτούνται θα γίνει χρήση απλών αρχείων κειμένου και όχι Βάσης Δεδομένων (Database).

ΚΕΦΑΛΑΙΟ 4

ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ

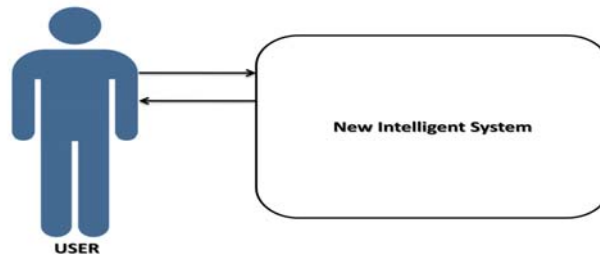
ΕΞΥΠΝΟΥ ΒΟΗΘΟΥ

Όπως αναφέραμε ήδη ακολουθούμε ένα ευέλικτο μοντέλο ανάπτυξης και όχι αυστηρά το μοντέλο καταράκτη με τις αυστηρές φάσεις ανάπτυξης του λογισμικού οπότε προσαρμόζουμε τα στάδια της ανάπτυξης στο χρόνο που διαθέτουμε και τις δικές μας προσωπικές ανάγκες. Οι γλώσσες που θα χρησιμοποιηθούν είναι η Prolog και η C# μέσα από το πακέτο λογισμικού Microsoft Visual Studio 2017, ενώ με την χρήση αρχείων κειμένου, αποφέχθηκε η χρήση κάποιου άλλου πακέτου όπως η MySQL που είναι Structured Query Language για την δημιουργία δομημένων βάσεων δεδομένων (relational database management system - RDBMS)²¹.

4.1 ΠΡΟΔΙΑΓΡΑΦΕΣ ΣΥΣΤΗΜΑΤΟΣ

Το σύστημα που μελετάται η ανάπτυξή του πρόκειται να είναι ένα ολοκληρωμένο πακέτο λογισμικού με διάφορα τμήματα. Σε αυτή την υποενότητα θα αναφέρουμε τα επιμέρους τμήματα και πώς αυτά θα επικοινωνούν μεταξύ τους, ενώ ταυτόχρονα θα δούμε πώς ο χρήστης αλληλεπιδρά με το σύστημά μας.


²¹ **Δομημένες βάσεις δεδομένων (relational database management system - RDBMS):** είναι ένα σύστημα διαχείρισης σχεσιακής βάση (relational database management system - RDBMS) που χρησιμοποιεί κάποια γλώσσα για την προσθήκη, την πρόσβαση και την επεξεργασία δεδομένων σε μία Βάση Δεδομένων, περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα.



Εικόνα 10. Ο χρήστης και το έξυπνο μας σύστημα αλληλεπιδρούν

Ο χρήστης ο οποίος πρέπει να κατέχει το πρόγραμμα και να το έχει εγκατεστημένο στον υπολογιστή του για να μπορεί να έχει πρόσβαση στο σύστημά μας. Αφού το εγκαταστήσει πρέπει να δημιουργήσει ένα λογαριασμό χρήστη με όνομα χρήστη (username) και κωδικό πρόσβασης (password). Στην σχεδίαση και υλοποίηση του συστήματος δίνεται έμφαση στις βασικές αρχές του human-computer Interaction²² έτσι ώστε να επιτευχθεί ένα φιλικό graphical user interface (GUI). Ένας βασικός στόχος της αλληλεπίδρασης ανθρώπου-υπολογιστή είναι να βελτιώσει την επικοινωνία μεταξύ χρηστών και υπολογιστών, μέσω της ορθής σχεδίασης εύχρηστων και εργονομικών υπολογιστών, προσανατολισμένων στις ανθρώπινες ανάγκες.

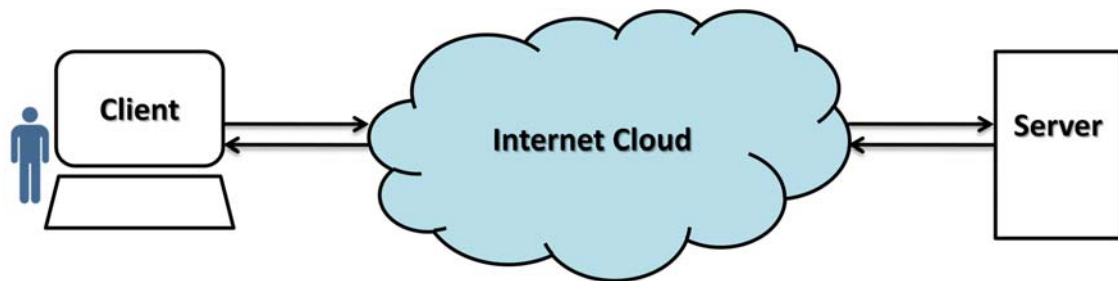
Κάποιοι κανόνες του human-computer interaction οι οποίοι εφαρμόζονται είναι:

- ✓ Μεγάλα κουμπιά με ευανάγνωστα ως προς την γραμματοσειρά γράμματα.
 - ✓ Συνέπεια σε σχέση με το τι εφαρμόζεται συνήθως σε παρόμοιες εφαρμογές για να χρειάζεται λιγότερο χρόνο για να μάθει κάποιος χρήστης να χρησιμοποιεί την εφαρμογή μας. Για παράδειγμα, για το search (αναζήτηση) χρησιμοποιείται το εικονίδιο με το φακό
- 
- ✓ Αποφυγή των έντονων χρωμάτων. Προτίμηση χρωμάτων που δεν κουράζουν το μάτι όταν χρησιμοποιείται παρατεταμένα η εφαρμογή.

4.1.1 Δομή/Αρχιτεκτονική Συστήματος

²² **αλληλεπίδραση ανθρώπου-υπολογιστή (HCI - Human Computer Interaction):** είναι το επιστημονικό πεδίο της πληροφορικής που μελετά την αλληλεπίδραση μεταξύ ανθρώπων (χρηστών) και υπολογιστών. Η αλληλεπίδραση μεταξύ χρηστών και υπολογιστών γίνεται στο επίπεδο της διεπαφής χρήστη (user interface), μέσω κατάλληλου λογισμικού και υλικού.

Αρχικά ας αναφερθούμε στο είδος της αρχιτεκτονικής του συστήματος και πώς αυτό κρίθηκε κατάλληλο για το συγκεκριμένο σύστημα. Αφού θέλουμε κάπου να υπάρχει μια κεντρική αποθήκη γνώσεων στην οποία να ψάχνουν απαντήσεις οι χρήστες μας, είτε να καταχωρούν νέα συμβάντα θα πρέπει να έχουμε κάποιο εξυπηρετητή/διακομιστή²³ (Server), στον οποίο θα βρίσκονται αποθηκευμένα όλα τα αρχεία κειμένου αλλά και Prolog ενώ ταυτόχρονα αφού θα έχει πρόσβαση σε αυτά τα αρχεία, θα ελέγχει τις απαιτήσεις των χρηστών με σκοπό να τους εξυπηρετήσει κατάλληλα στέλνοντας τους πίσω τις πληροφορίες που ζήτησαν. Στην πλευρά του Client θα υπάρχει μια εφαρμογή με ένα πιο ευχάριστο γραφικό περιβάλλον επικοινωνίας που σκοπό έχει να βοηθήσει το χρήστη να χρησιμοποιήσει την εφαρμογή μας.

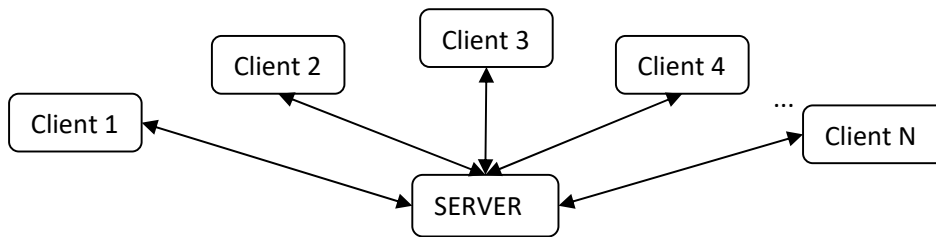


Εικόνα 11. Αφαιρετικό μοντέλο αρχιτεκτονικής Client-Server²⁴

Στην Εικόνα 11, μπορούμε να διακρίνουμε ένα αφαιρετικό μοντέλο της Client-Server αρχιτεκτονικής η οποία εφαρμόζεται σε αυτό το σύστημα, ενώ θα ήταν καλό να αναφαίρουμε ότι όπως και στην Εικόνα 12, ένα τέτοιο σύστημα μπορεί να εξυπηρετήσει πολλούς χρήστες ταυτόχρονα.

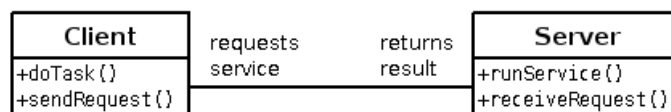
²³ **Διακομιστής (Server):** Ο Server ή αλλιώς εξυπηρετητής ή διακομιστής, στην πιο απλή του μορφή είναι ένας ηλεκτρονικός υπολογιστής που τρέχει κατάλληλο λογισμικό ώστε να εξυπηρετεί τους χρήστες που συνδέονται με αυτόν για κάποιο σκοπό. Ανάλογα με τον σκοπό, ο server τρέχει και τις κατάλληλες υπηρεσίες και έχει και κατάλληλη ονομασία.

²⁴ **Αρχιτεκτονική Πελάτη-Διακομιστή (Client-Server model):** Στην αρχιτεκτονική αυτή το σύστημα δέχεται αιτήσεις για τις υπηρεσίες του από μια οντότητα εκτός συστήματος και προωθεί τις απαντήσεις στην οντότητα αυτή.



Εικόνα 12. Client - Server μοντέλο με πολλούς clients συνδεδεμένους ταυτόχρονα

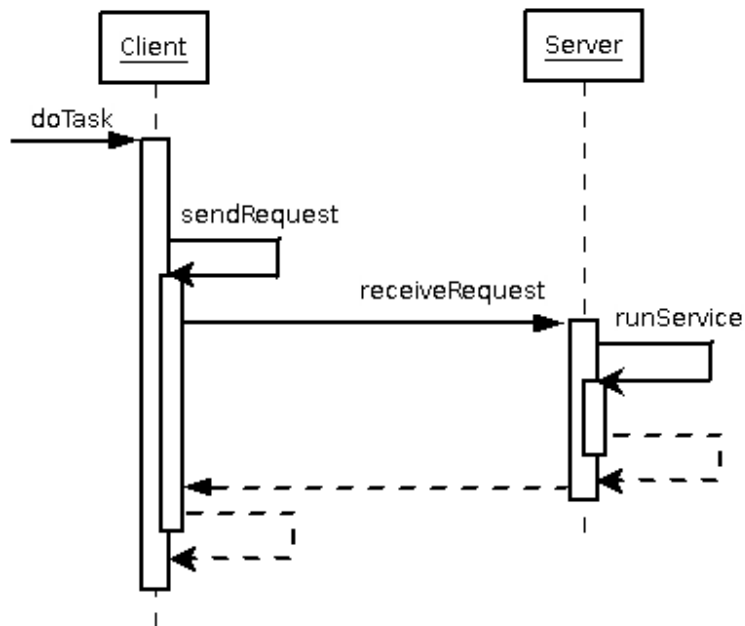
Πιο απλά και συνοπτικά, ο εξυπηρετητής (Server) θα είναι υπεύθυνος για να διαχειρίζεται τα αιτήματα (requests) των clients που συνδέονται στο σύστημα και να παρέχει κάποια ανταπόκριση για να επιτυγχάνεται η επικοινωνία των χρηστών με το σύστημα. Στον υπολογιστή του χρήστη θα είναι εμφανές μόνο το η εφαρμογή που τον αφορά, η οποία θα διαθέτει ένα αρκετά φιλικό γραφικό περιβάλλον επικοινωνίας για να τον διευκολύνει στην χρήση αλλά και για να προστατεύσει τα υπόλοιπα μέρη του συστήματος από ένα άπειρο χρήστη που μπορεί να τα καταστρέψει χωρίς όμως να το καταλάβει.



Εικόνα 13. Διάγραμμα κλάσεων (class diagram)²⁵ αρχιτεκτονικής πελάτη-εξυπηρετητή

Στο διάγραμμα κλάσεων της Εικόνας 13, βλέπουμε ένα αφαιρετικό διάγραμμα, με 2 κύριες κλάσεις τον πελάτη (client) και τον εξυπηρετητή (server). Ο client κάνει διάφορες κινήσεις για να βρεί αυτό που θέλει στην εφαρμογή που τον προμηθεύσαμε (doTask()), ενώ ταυτόχρονα μια κίνησή του μπορεί να πυροδοτήσει κάποιο αίτημα (sendRequest()) προς τον εξυπηρετητή, όπου κάθε υπηρεσία ή επεξεργασία που ζητάμε από το server σαν χρήστες καλύπτεται από την μέθοδο runService(), αφού πρώτα λάβει κάποιο αίτημα (receiveRequest()).

²⁵ **Διάγραμμα κλάσεων (class diagram):** Τα διαγράμματα κλάσεων περιγράφουν τις οντότητες που απαρτίζουν ένα σύστημα και τις στατικές συσχετίσεις μεταξύ τους. Είναι το βασικότερο είδος των διαγραμμάτων της UML. Ένα διάγραμμα κλάσεων δείχνει την στατική δομή των κλάσεων του συστήματος και των σχέσεων μεταξύ τους.



Εικόνα 14. Διάγραμμα Ακολουθίας (Sequence Diagram)²⁶ μοντέλου πελάτη - εξυπηρετητή

4.2 ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΕΞΥΠΝΟΥ ΒΟΗΘΟΥ

4.2.1 Βήματα που έγιναν στην πρώτη φάση της εργασίας

Όπως ήδη έχει αναφερθεί πιο πάνω η εργασία ξεκίνησε με βάση ένα μικρό expert system υλοποιημένο εξ' ολοκλήρου σε γλώσσα Prolog η οποία χρησιμοποιείται κυρίως για Logic Programming.

Ας ξεκινήσουμε με τους τρόπους συλλογής γνώσης που χρησιμοποιήθηκαν στην αρχική εργασία αλλά και στην συνέχεια για να εμπλουτιστεί η αρχική βάση γνώσης. Για την συλλογή της γνώσης έγινε χρήση περισσότερων από μία μεθόδων. Οι μέθοδοι αυτοί είναι:

- Έγιναν προσωπικές συναντήσεις όπου ερωτήθηκαν έμπειροι εκπαιδευτικοί για δύσκολα περιστατικά που αντιμετώπισαν με τους μαθητές τους.

²⁶ **Διάγραμμα Ακολουθίας (Sequence Diagram):** Τα διαγράμματα ακολουθίας παρουσιάζουν την αλληλεπίδραση των αντικειμένων μέσω της ανταλλαγής μηνυμάτων. Δίνουν έμφαση στη χρονική αλληλουχία των γεγονότων. Ένα διάγραμμα ακολουθίας περιέχει χειριστές, αντικείμενα και μηνύματα που ανταλλάσσουν τα αντικείμενα. Τα αντικείμενα οργανώνονται στον οριζόντιο άξονα και τα γεγονότα στον κάθετο άξονα του χρόνου. Τα διαγράμματα ακολουθίας χρησιμοποιούνται για να περιγράψουν τη ροή του ελέγχου μέσα στο σύστημα.

- Φτιάχτηκε ένα πολύ απλό ερωτηματολόγιο (5 ερωτήσεις) το οποίο δημοσιεύτηκε στο διαδύκτιο (15 απαντήσεις) και απευθυνόταν επίσης σε εκπαιδευτικούς, ενώ ταυτόχρονα δώθηκε και σε εκτυπωμένο σε χαρτί (12 απαντήσεις).
- Διαβάστηκε έντυπο υλικό (βιβλία για ψυχολογία μαθητών και άρθρα για περιπτώσεις σχολικού εκφοβισμού - bullying)
- Διενεργήθηκε αναζήτηση για υλικό στο διαδύκτιο.

Αξιζει να σημειωθεί ότι το δείγμα που πάρθηκε από τις συνεντεύξεις και το ερωτηματολόγιο Online και σε χαρτί είναι αρκετά μικρό έτσι αφού δεν υπήρξε ιδιαίτερο ενδιαφέρον, έγινε μια περισσότερη εκτενής έρευνα σε βιβλία και online υλικό στο διαδύκτιο.

Έρευνα με σκοπό την συλλογή γνώσης για σκοπούς δημιουργίας ενός έξυπνου βοηθού

Η παρούσα έρευνα έχει σκοπό την συλλογή γνώσης που αφορά τον χώρο περιέργια περιστατικά που έχουν αντιμετωπίσει κάποιος εκπαιδευτικός. Η γνώση που θα συλλεχθεί θα αποσταλεί στο knowledge base ενός μικρού expert system. Ενώ στην συνέχεια πρόκειται να αποταλέσει την αρχική γνώση που θα χρησιμοποιηθεί για την δημιουργία ενός έξυπνου βοηθού (smart agent), ο οποίος θα μπορεί να μαθαίνει από νέα περιστατικά που θα καταχωρούν κάποιος εκπαιδευτικός αλλά και να δίνει απαντήσεις για πιθανές αιτίες που προκαλούν κάποια συμπεριφορά σε άλλους εκπαιδευτικούς που δεν έχουν συναντήσει προηγουμένως τις ίδιες συμπεριφορές.

* Required

Ηλικίες μαθητών που διδάσκετε *

Δημοτικό (6-11 χρόνων)

Γυμνάσιο (12-14 χρόνων)

Λύκειο (15-18 χρόνων)

Φοιτητές Πανεπιστημίου (18+ χρόνων)

Έχετε αντιμετωπίσει προβληματικές συμπεριφορές ή περιστατικά κατά την διάρκεια των χρόνων που διδάσκετε; *

Ναι

Όχι

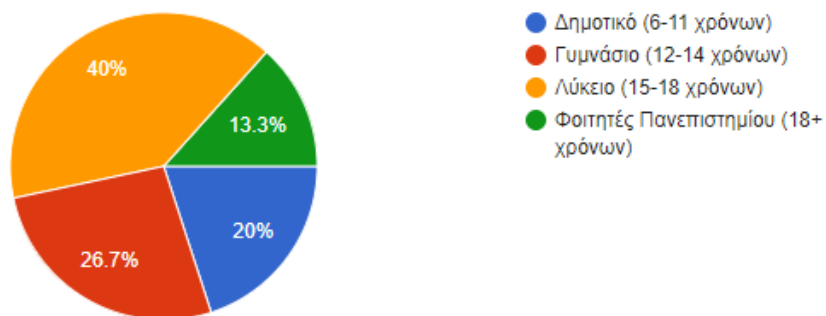
Εικόνα 15. Το ερωτηματολόγιο που φτιάχτηκε με την χρήση Google forms

(https://docs.google.com/forms/d/e/1FAIpQLSeNXn1Jwe8viAsEEmv1EVVAOXHhoNUf45sL6AM2T_xkGM1AeA/viewform)

Ακολουθούν στιγμιότυπα (screenshots) με τα στατιστικά αποτελέσματα και λίστες απαντήσεων του online ερωτηματολογίου ανα ερώτηση.

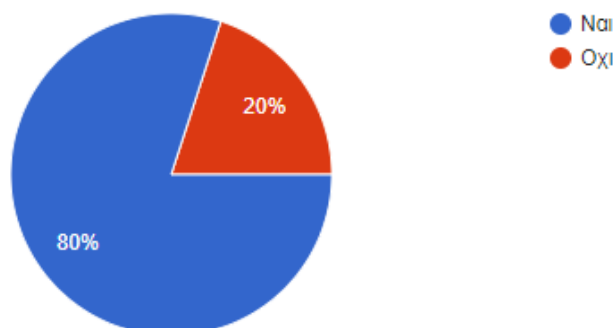
Ηλικίες μαθητών που διδάσκετε

15 responses



Έχετε αντιμετωπίσει προβληματικές συμπεριφορές ή περιστατικά κατά την διάρκεια των χρόνων που διδάσκετε;

15 responses



Περιγραφή κάποιου παράξενου περιστατικού που αντιμετωπίσατε σε σχέση με κάποιο μαθητή σας (αν έχετε απαντήσει όχι στην προηγούμενη ερώτηση όχι αγνοείτε την ερώτηση αυτή)

10 responses

Υπερκινητικους mathites, na perpatane dixos adeia stin taxi.. na miloun stin othoni tou computer.

Αναρμοστη συμπεριφορα

Υπερεγώ και υπεροψία. Ο φοιτητής νόμιζε ότι έκανε 2ο πτυχίο ενώ στην πραγματικότητα έκανε επιπλέον μαθήματα για να αναγνωριστεί το πτυχίο του και είχε τόσα κενά που του ήταν αδύνατο να κάνει την πτυχιακή του μόνος του.

Your answer
θυμός και εγκατάλειψη της τάξης

Δεν μπορούσε να διαχειριστεί τον θυμό του και λιποθυμούσε.

μαθητής μου 16 χρονών έχει περίεργη ιδιόρρυθμη συμπεριφορά με τάσεις αυτοαπομόνωσης

Μαθητής που αρνείται να παρακολουθεί το μάθημα στην τάξη. Όταν βρίσκεται στην τάξη προσπαθεί συνεχώς να συγκεντρώνει την προσοχή των συμμαθητών του προκαλώντας τους.

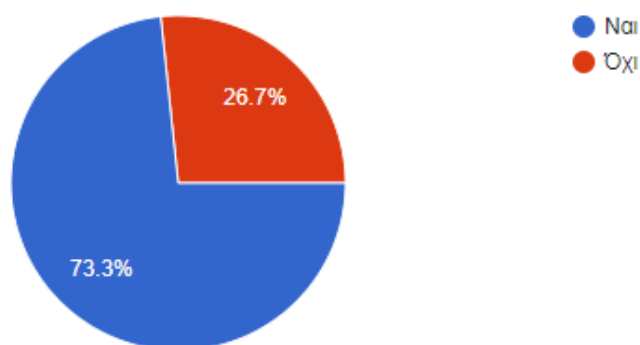
Απειλές, θυμός και οργή κατα το πρόσωπό μου κατα τη διδασκαλία.

Να σηκώνεται απο τη θέση του κατα τη διάρκεια του μαθήματος και να μιλά με χυδαίο τρόπο

Άλλο μαθήτεις τρακώθηκαν μεταξύ τους και υπήρξαν εν μία την άλλη

Γνωρίζετε τον πιθανό λόγο που οδήγησε τον μαθητή σε αυτή την συμπεριφορά

15 responses



Δώστε τον λόγο για την συσχεκκριμένη συμπεριφορά (αν απαντήσατε ναι στην προηγούμενη ερώτηση) π.χ. προβλήματα στην οικογένεια, κάποια μαθησιακή δυσκολία, κακός χαρακτήρας κ.α.

12 responses

Dispraxia, dislexia k provlimata sto oikogeniako perivalon

Οικογενειακα προβληματα

Η συμπεριφορά των γονέων του προς αυτόν που τον επαινούσαν πάντα, ακόμη και χωρίς λόγο. Ήταν αυτό που λέμε στην Κυπριακή "απομόνος"!

Μαθησιακά προβλήματα, τσακωμός με τους φίλους του και την κοπέλα του, εθισμός στα ναρκωτικά.

Αυτισμος

Αυτό θα το κρίνουν οι ειδικοί

ιδιορρυθμία του χαρακτήρα του, έλλειψη προσαρμοστικότητας και ευελιξίας (χαρακτήρας και μαθησιακή δυσκολία)

Προβλήματα στην οικογένεια

για να επιδείξουν ότι έχουν ισχύ προσπαθώντας να επιβάλουν και προβάλλοντας δύναμη σωματική, για να μην νιώσουν κάπως ταπεινωτικά απέναντι σε άλλες ομάδες μαθητών, η δική μου επιμονή κάποιες φορές να κάνουν τις εργασίες τους προκαλεί εκνευρισμό.

κακός χαρακτήρας, μη έλεγχος απο το σπίτι

Αυτές τις απαντήσεις μας δίνεται δυνατότητα από την Google να τις κάνουμε download υπο μορφή sheets και μπορούμε να τα ανοίξουμε στον υπολογιστή μας με Microsoft Excel.

Timestamp	Ηλικία μαθητών που δόε	Έχετε αντιμετωπίσει προς	Περιγραφή κάποιου παρά Γνωρίζετε τον πιθανό λόγ	Δώστε τον λόγο για την συσχεκκριμένη συμπεριφορά (αν απαντήσατε ναι στην προηγούμενη ερώτηση) π.χ. προβλήματα στην οικογένεια,
1/5/2017 18:05:29	Λύκειο (15-18 χρόνων)	Ναι	Υπερκινητικους μαθητες. η Ναι	Dispraxia, dislexia k provlimata sto oikogeniako perivalon
1/5/2017 18:38:52	Δημοτικό (6-11 χρόνων)	Ναι	Αναρμοστη συμπεριφορας Ναι	Οκαγενειακα προβληματα
1/5/2017 22:18:51	Φοιτητές Πανεπιστημίου (Ναι)	Ναι	Υπεργώ και υπερροαία C Ναι	Η συμπεριφορά των γονέων του προς αυτόν που τον επαινούσαν πάντα, ακόμη και χωρίς λόγο. Ήταν αυτό που λέμε στην Κυπριακή "απο
1/5/2017 22:52:33	Λύκειο (15-18 χρόνων)	Ναι	Υστερ απαντες θυμός και εγκατάλειψη τη Ναι	Μαθησιακά προβλήματα, τσακωμός με τους φίλους του και την κοπέλα του, εθισμός στα ναρκωτικά.
1/7/2017 11:17:17	Δημοτικό (6-11 χρόνων)	Όχι	Όχι	
1/7/2017 11:40:57	Δημοτικό (6-11 χρόνων)	Ναι	Ναι	Αυτισμος
1/7/2017 21:55:37	Γυμνάσιο (12-14 χρόνων)	Όχι	Όχι	
1/14/2017 9:05:12	Γυμνάσιο (12-14 χρόνων)	Ναι	Δεν μπορούσα να διαχωρι Όχι	Αυτό θα το κρίνουν οι ειδικοί
1/22/2017 23:41:50	Λύκειο (15-18 χρόνων)	Ναι	μαθητης μου 16 χρόνων (Ναι	ιδιορρυθμία του χαρακτήρα του, έλλειψη προσαρμοστικότητας και ευελιξίας (χαρακτήρας και μαθησιακή δυσκολία)
1/22/2017 23:44:44	Γυμνάσιο (12-14 χρόνων)	Ναι	Μαθητης που αφηλεια να Ναι	Προβλήματα στην οικογένεια
1/23/2017 0:22:43	Λύκειο (15-18 χρόνων)	Ναι	Απαλεις, θυμός και οργη + Ναι	για να επιδείξουν ότι έχουν ισχύ προσπαθώντας να επιβάλουν και προβάλλοντας δύναμη σωματική, για να μην νιώσουν κάπως ταπεινωτικά
1/23/2017 0:27:31	Λύκειο (15-18 χρόνων)	Ναι	Να σηκώνετε απο τη βία Ναι	κακός χαρακτήρας, μη έλεγχος απο το σπίτι
1/23/2017 0:30:54	Γυμνάσιο (12-14 χρόνων)	Όχι	Ναι	οκαγενειακά προβλήματα, έλλειψη αυτοεκτιμησης, χαμηλή αυτοπεπιθωση, αδύνατοι μαθητές
1/23/2017 0:33:00	Λύκειο (15-18 χρόνων)	Ναι	δύο μαθητές τσακωδη Ναι	Προβλήματα στην οικογένεια, μαθητρια με παραβατική συμπεριφορά
1/23/2017 21:20:05	Φοιτητές Πανεπιστημίου (Ναι)	Ναι	Όχι	

Αφού έγιναν τα πιο πάνω έγινε επιλογή γνώσης για την διευκόλυνση της δημιουργίας του semantic network και του τελικού expert system. Αξίζει να σημειωθεί ότι απο τις απαντήσεις του ερωτηματολογίου που αναρτήθηκε στο διαδύκτιο και απο την έρευνα που έκανα σε θέματα ψυχολογίας τα στοιχεία που συλλέχθηκαν συγκλίνουν. Για να διευκόλυνση τις διαδικασίας εντάχθηκαν κάποια συγκεκριμένα περιστατικά σε κατηγορίες συμπεριφορών ή χαρακτηριστικών παιδιών. Για παράδειγμα ένα παιδί που σκίζει το διαγώνισμα και φεύγει χτυπώντας την πόρτα απο την τάξη έχει επιθετική συμπεριφορά, ή ένα παιδί που του γίνεται παρατήρηση από εκπαιδευτικό στην τάξη και απειλεί τον εκπαιδευτικό έχει επίσης επιθετική συμπεριφορά.

ΣΥΛΛΕΧΘΗΣΑ ΓΝΩΣΗ

Ακολουθεί το κείμενο που συντάχθηκε απο την συλλεχθήσα γνώση:

Οι τύποι γονεϊκής συμπεριφοράς επηρεάζουν την μετέπειτα συμπεριφορά των μαθητών. Οι **Κυριαρχικοί** Γονείς οδηγούν σε παιδιά ανεξάρτητα, διεκδικητικά, φιλικά προς τους συνομήλικους, συνεργατικά με τους γονείς, ευτυχισμένα, με κίνητρα επίτευξης, επιτυχημένα. Οι **αυταρχικοί** γονείς έχουν παιδιά με τάση προς κοινωνική απόσυρση, έλλειψη αυθορμητισμού, τα κορίτσια είναι εξαρτημένα και χωρίς κίνητρα επίτευξης και τα αγόρια έχουν την τάση να είναι επιθετικά προς τους συνομήλικους. Τα παιδιά των **επιεικών** γονέων έχουν θετική και ζωηρή διάθεση, είναι όμως ανώριμα, και υστερούν σε αυτοέλεγχο, κοινωνική υπευθυνότητα και αυτοδυναμία, τάση προς επιθετικότητα. **Παραμελητικοί** γονείς σημαίνει παιδιά με τάση προς δυσθυμία και έλλειψη αυτοσυγκέντρωσης, παραβατικά, με αδύναμο έλεγχο των συγκινήσεων τους, συχνές απουσίες από το σχολείο και λήψη ναρκωτικών ουσιών.

Σημάδια μαθησιακών δυσκολιών είναι η δυσκολία στην ανάγνωση και/ή τη γραφή, οι δυσκολίες στα Μαθηματικά, η δυσκολία απομνημόνευσης, η δυσκολία εστίασης της προσοχής, η δυσκολία τήρησης οδηγιών, ο κακός συντονισμός, η δυσκολία διαχείρισης εννοιών που αφορούν στο χρόνο και η δυσκολία οργάνωσης. Συμπτώματα μαθησιακών δυσκολιών είναι η παρορμητική συμπεριφορά, η απρεπή ανταπόκριση σε ερωτήσεις (στο σχολείο ή σε άλλο κοινωνικό πλαίσιο), η δυσκολία συγκέντρωσης (διάσπαση προσοχής), η δυσκολία έκφρασης (το παιδί δυσκολεύεται να βρει τον τρόπο να εκφράσει αυτό που σκέφτεται), η ασταθής επίδοση στο σχολείο, η ανωριμότητα στην ομιλία, η ελλιπής αντίληψη μέσω της ακοής, η δυσκολία διαχείρισης νέων καταστάσεων και η δυσκολία κατανόησης λέξεων ή εννοιών.

Αν το παιδί έχει κακοποιηθεί ή αν είναι μάρτυρας σκηνών βίας μεταξύ των γονέων ίσως έχει παρόμοια συμπεριφορά, άρα μπορεί να γίνει θύτης σε σχολικό εκφοβισμό. Η παραβατική συμπεριφορά μπορεί να είναι μέσο επιβίωσης για το μαθητή που

μεγαλώνει από μικρός σε ένα περιβάλλον στερημένο από αγάπη, παραμελημένος από την οικογένεια, προσπαθώντας να βρει το δικό του μονοπάτι για την επιβίωση, αρα επηρεάζεται από το περιβάλλον του. Υπάρχουν και εκείνα τα παιδιά που εκδηλώνουν παραβατικές συμπεριφορές, ενώ μεγαλώνουν σε ευοίωνες συνθήκες, με γονείς που προσπαθούν για ό,τι καλύτερο. Σύμφωνα με την επιστημονική βιβλιογραφία, οι ακραίες παραβατικές συμπεριφορές σε παιδιά και εφήβους που δεν οφείλονται σε περιβαλλοντολογικούς παράγοντες συνήθως είναι αποτέλεσμα διαταραχής διαγωγής. Πιθανά θύματα σχολικού εκφοβισμού είναι παιδιά ευαίσθητα στα πειράγματα, που θυμώνουν και κλαίει εύκολα, απομονώνονται. Οι θύτες συνήθως είναι παιδιά που δείχνουν έντονη αυτοπεποίθηση, δεν πιθαρχούν σε κανόνες, θέλουν να επιβάλλονται, θεωρούν ότι η βία είναι κοινωνικά αποδεκτή, υιοθετούν ρόλο αρχηγού στις παρέες. Οι θύτες και τα θύματα μπορούν να έχουν είτε υπερπροστατευτικούς, είτε αυταρχικούς γονείς. Οι αντιδράσεις θυμάτων σχολικού εκφοβισμού είναι επιθετικότητα, υπερβολική ανησυχία, το παιδί φοβάται να χρησιμοποιήσει το κινητό του, κάνει αδικαιολόγητες απουσίες από το σχολείο, έχει εφιάλτες, πέφτει η σχολική του επίδοση, έχει τάσεις αυτοκτονίας, έλλειψη αυτοεκτίμησης, χάνει συχνά τα πράγματά του, επιστρέφει σπíti με σημάδια και σχισμένα ρούχα. Τα παιδιά θύτες είναι αποτέλεσμα απουσίας ισορροπίας μέσα στην οικογένεια, όπως όταν κάποιος από τους γονείς λείπει ή έχει πεθάνει, αν η οικογένεια περνά πολλές οικονομικές δυσκολίες, αν κάποιος είναι άρρωστος και το παιδί νιώθει ανασφάλεια. Τα παιδιά θύτες και θύματα σχολικού εκφοβισμού είναι παιδιά που ίσως καταλήξουν χρήστες ναρκωτικών.

ΑΝΑΠΑΡΑΣΤΑΣΗ ΓΝΩΣΗΣ - ΣΗΜΑΣΙΟΛΟΓΙΚΟ ΔΙΚΤΥΟ (SEMANTIC NETWORK)

Σε αυτή την ενότητα της εργασίας παραθέτονται τα όσα περιγράφονται με λόγια (φυσική γλώσσα πιο πάνω υπο μορφή γράφου σημασιολογικού δικτύου. Το σημασιολογικό δίκτυο ήταν αρκετά μεγάλο και αφού κάθε παράγραφος της γνώσης που παρουσιάζεται πιο πάνω αφορά ένα ανεξάρτητο ξεχωριστό θέμα, διαιρέθηκε σε τρία μικρότερα επίσης ανεξάρτητα σημασιολογικά δίκτυα.

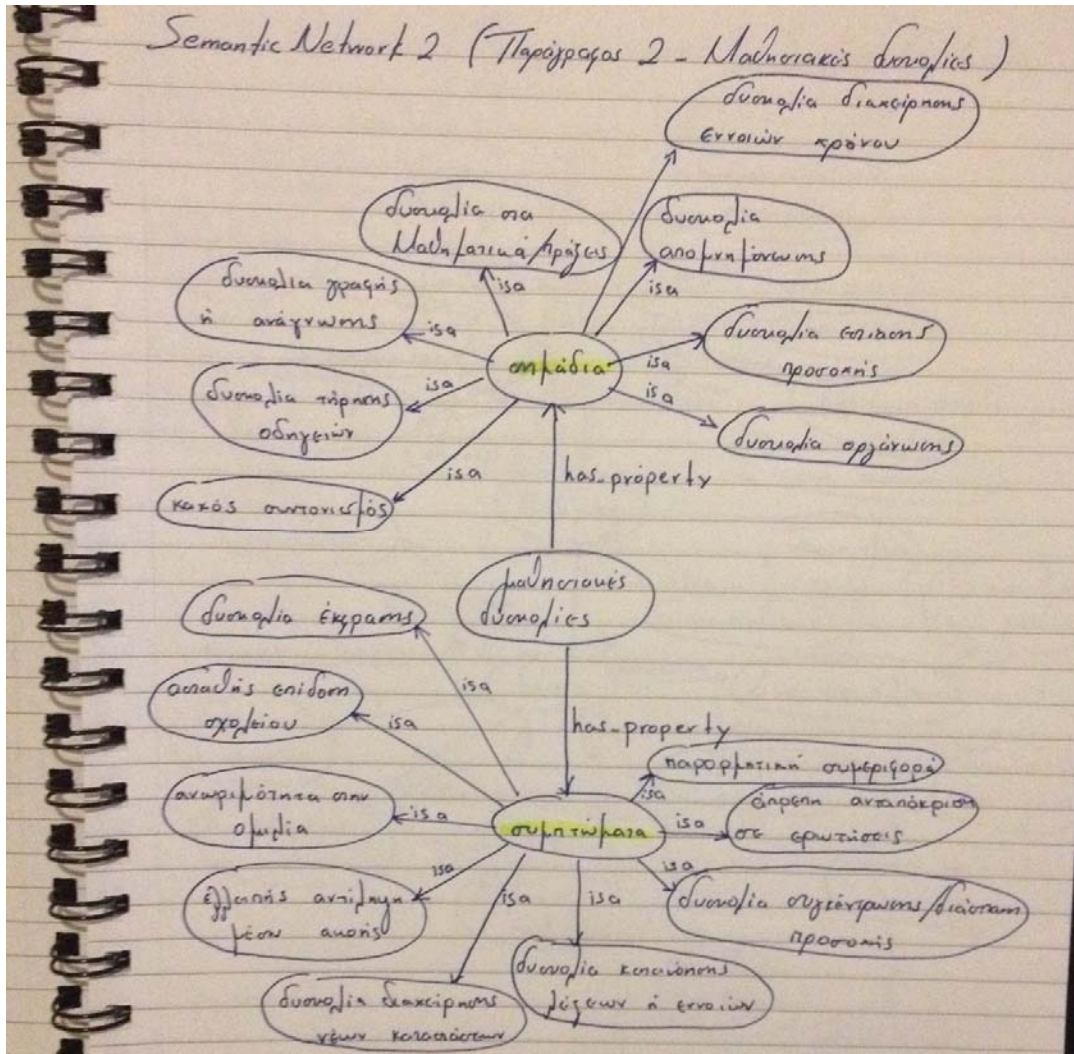
Αξίζει να σημειωθεί ότι το πιο πολύπλοκο από τα 3 αυτά σημασιολογικά δίκτυα στην δημιουργία του από την παράγραφο της συλλεχθήσας γνώσης σε φυσική γλώσσα ήταν η 3^η παράγραφος που αφορά τον σχολικό εκφοβισμό (bullying).

Semantic Network 1 – Τύποι γονέων – Γράφος



Οι τύποι γονεϊκής συμπεριφοράς επηρεάζουν την μετέπειτα συμπεριφορά των μαθητών. Οι **Κυριαρχικοί** Γονείς οδηγούν σε παιδιά ανεξάρτητα, διεκδικητικά, φιλικά προς τους συνομήλικους, συνεργατικά με τους γονείς, ευτυχισμένα, με κίνητρα επίτευξης, επιτυχημένα. Οι **αυταρχικοί** γονείς έχουν παιδιά με τάση προς κοινωνική απόσυρση, έλλειψη αυτορρητισμού, τα κορίτσια είναι εξαρτημένα και χωρίς κίνητρα επίτευξης και τα αγόρια έχουν την τάση να είναι επιθετικά προς τους συνομήλικους. Τα παιδιά των **επιεικών** γονέων έχουν θετική και ζωηρή διάθεση, είναι όμως ανώριμα, και υστερούν σε αυτοέλεγχο, κοινωνική υπευθυνότητα και αυτοδυναμία, τάση προς επιθετικότητα. **Παραμελητικοί** γονείς σημαίνει παιδιά με τάση προς δυσθυμία και έλλειψη αυτοσυγκέντρωσης, παραβατικά, με αδύναμο έλεγχο των συγγκινήσεων τους, συχνές απουσίες από το σχολείο και λήψη ναρκωτικών ουσιών.

Semantic Network 2 – Μαθησιακές Δυσκολίες – Γράφος



Σημάδια μαθησιακών δυσκολιών είναι η δυσκολία στην ανάγνωση και/ή τη γραφή, οι δυσκολίες στα Μαθηματικά, η δυσκολία απομνημόνευσης, η δυσκολία επίτασης της προσοχής, η δυσκολία τήρησης οδηγιών, ο κακός συντονισμός, η δυσκολία διαχείρισης εννοιών που αφορούν στο χρόνο και η δυσκολία οργάνωσης. Συμπτώματα μαθησιακών δυσκολιών είναι η παραρρητική συμπεριφορά, η απρεπτή ανταπόκριση σε ερωτήσεις (στο σχολείο ή σε άλλο κοινωνικό πλαίσιο), η δυσκολία συγκέντρωσης (διάσπαση προσοχής), η δυσκολία έκφρασης (το παιδί δυσκολεύεται να βρει τον τρόπο να εκφράσει αυτό που σκέφτεται), η ασταθής επίδοση στο σχολείο, η ανωριμότητα στην ομιλία, η ελλιπής αντίληψη μέσω της ακοής, η δυσκολία διαχείρισης νέων καταστάσεων και η δυσκολία κατανόησης λέξεων ή εννοιών.

βιβλιογραφία, οι ακραίες παραβατικές συμπεριφορές σε παιδιά και εφήβους που δεν οφείλονται σε περιβαλλοντολογικούς παράγοντες συνήθως είναι αποτέλεσμα διαταραχής διαγωγής. Πιθανά θύματα σχολικού εκφοβισμού είναι παιδιά ευαίσθητα στα πειράγματα, που θυμώνουν και κλαίει εύκολα, απομονώνονται. Οι θύτες συνήθως είναι παιδιά που δείχνουν έντονη αυτοπεποίθηση, δεν πειθαρχούν σε κανόνες, θέλουν να επιβάλλονται, θεωρούν ότι η βία είναι κοινωνικά αποδεκτη, υιοθετούν ρόλο αρχηγού στις παρέες. Οι θύτες και τα θύματα μπορούν να έχουν είτε υπερπροστατευτικούς, είτε αυταρχικούς γονείς. Οι αντιδράσεις θυμάτων σχολικού εκφοβισμού είναι επιθετικότητα, υπερβολική ανησυχία, το παιδί φοβάται να χρησιμοποιήσει το κινητό του, κάνει αδικαιολόγητες απουσίες από το σχολείο, έχει εφιάλτες, πέφτει η σχολική του επίδοση, έχει τάσεις αυτοκτονίας, έλλειψη αυτοεκτίμησης, χάνει συχνά τα πράγματά του, επιστρέφει σπίτι με σημάδια και σχισμένα ρούχα. Τα παιδιά θύτες είναι αποτέλεσμα απουσίας ισορροπίας μέσα στην οικογένεια, όπως όταν κάποιος από τους γονείς λείπει ή έχει πεθάνει, αν η οικογένεια περνά πολλές οικονομικές δυσκολίες, αν κάποιος είναι άρρωστος και το παιδί νιώθει ανασφάλεια. Τα παιδιά θύτες και θύματα σχολικού εκφοβισμού είναι παιδιά που ίσως καταλήξουν χρήστες ναρκωτικών.

ΥΛΟΠΟΙΗΣΗ ΕΜΠΕΙΡΟΥ ΣΥΣΤΗΜΑΤΟΣ

Όπως ήδη έχουμε αναφέρει σε προηγούμενη ενότητα της εργασίας, ένα έμπειρο σύστημα αποτελείται από 3 μέρη (Knowledge base, interference engine, user interface). Το κάθε ένα από αυτά τα μέρη έχει ένα ιδιαίτερο και ξεχωριστό ρόλο.

Η βάση γνώσης (knowledge base) και ο μηχανισμός εξαγωγής συμπερασμάτων (Interference engine) είναι κρυμμένα από τον χρήστη του συστήματος ο οποίος αγνωρίζει στοιχεία για το έμπειρο σύστημα, όπως ποια η δομή του, ποια τα επιμέρους τμήματά του και πώς επιτυγχάνεται η επικοινωνία μεταξύ αυτών των τμημάτων. Ο χρήστης μπορεί να «δεί» κάποιο περιβάλλον επικοινωνίας που του παρέχει το User Interface.

Σχετικά με το υλοποιηθέν expert system της παρούσας εργασίας παρουσιάζονται αναλυτικά τα επιμέρους τμήματά του στο Παράρτημα Α της εργασίας. Επίσης, αξίζει να σημειωθεί ότι ο Prolog κώδικας του συστήματος παρόλο που θα μπορούσε να βρίσκεται σε ένα αρχείο έχει χωριστεί σε τρία μικρότερα αρχεία. Το καθένα από αυτά τα αρχεία περιέχει κώδικα για ένα από τα 3 μέρη του expert system.

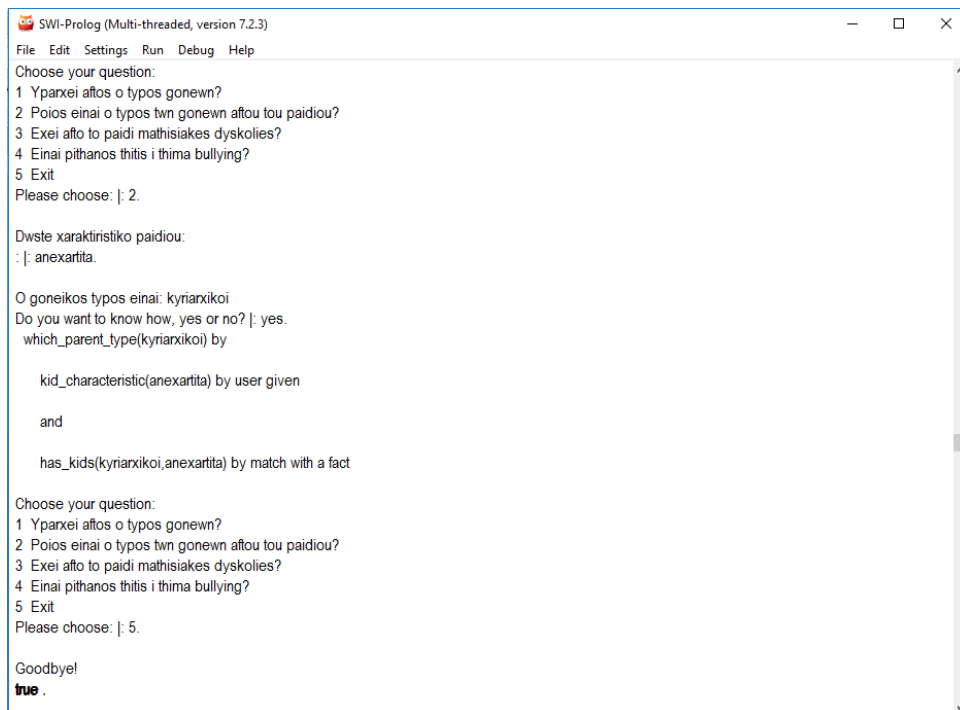
ΤΟ EXPERT SYSTEM ΠΟΥ ΠΑΡΑΧΘΗΚΕ

Επιτεύχθηκε η κατασκευή ενός έμπειρου συστήματος με την χρήση μιας γλώσσας Λογικού προγραμματισμού (Prolog). Παράχθηκε μια χαμηλού επιπέδου μορφή τεχνητής νοημοσύνης. Μέσα απο μια μικρή σχετικά αποθήκη γνώσης – δεδομένων και μια έτοιμη μηχανή συμπερασμού επιτεύχθηκε να μπορούμε να πάρουμε απαντήσεις σε μερικά ερωτήματα, όπως:

- 1) Ποιός ο τύπος των γονέων του παιδιού με αυτό το χαρακτηριστικό;
- 2) Υπάρχει αυτός ο συγκεκριμένος τύπος γονέων;
- 3) Έχει το παιδί με κάποια συγκεκριμένη ιδιότητα - χαρακτηριστικό μαθησιακές δυσκολίες;
- 4) Ένα παιδία με κάποια ιδιότητα είναι πιθανός θύτης ή θήμα σχολικού εκφοβισμού;

Σε αυτό το σημείο αξίζει να σημειωθεί ότι το σύστημα κρατάει τις προηγούμενες απαντήσεις σας και εαν επιλέξετε απο το μενού επιλογών του κάποια συγκεκριμένη επιλογή 2^η φορά θα σας δώσει απευθείας την απάντηση. Επίσης, για να δώσει απάντηση το σύστημα αυτό σε κάποιο απο τα ερωτήματά μας πρέπει να καταχωρήσουμε το δεδομένο εισόδου όπως είναι γραμμένο στην βάση γνώσης αλλιώς δεν θα μας δώσει ορθή απάντηση.

Στην εικόνα που ακολουθεί παρουσιάζεται ένα παράδειγμα εκτέλεσης του συστήματος.



```
SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help
Choose your question:
1 Υπαρχει αλλος ο typos gonewn?
2 Ποιος einai ο typos twν gonewn αττου του παιδιου?
3 Exei αττο το παιδι mathsiakes dyskoliες?
4 Einai pithanos thitis i thima bullying?
5 Exit
Please choose: |: 2.

Dwste xarakteristiko παιδιου:
: |: anexartita.

O goneikos typos einai: kyriarxikoι
Do you want to know how, yes or no? |: yes.
which_parent_type(kyriarxikoι) by

    kid_characteristic(anexartita) by user given

and

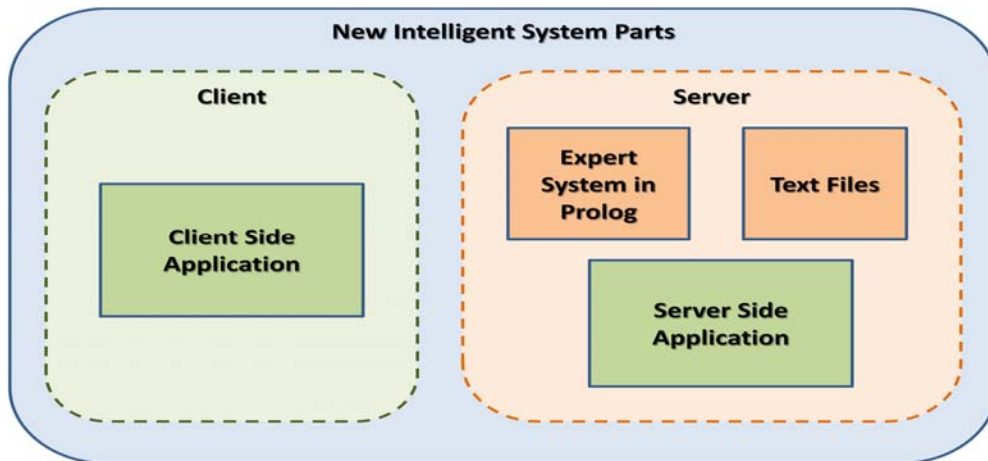
    has_kids(kyriarxikoι,anexartita) by match with a fact

Choose your question:
1 Υπαρχει αλλος ο typos gonewn?
2 Ποιος einai ο typos twν gonewn αττου του παιδιου?
3 Exei αττο το παιδι mathsiakes dyskoliες?
4 Einai pithanos thitis i thima bullying?
5 Exit
Please choose: |: 5.

Goodbye!
true.
```

Εικόνα 16. Παράδειγμα εκτέλεσης έμπειρου συστήματος

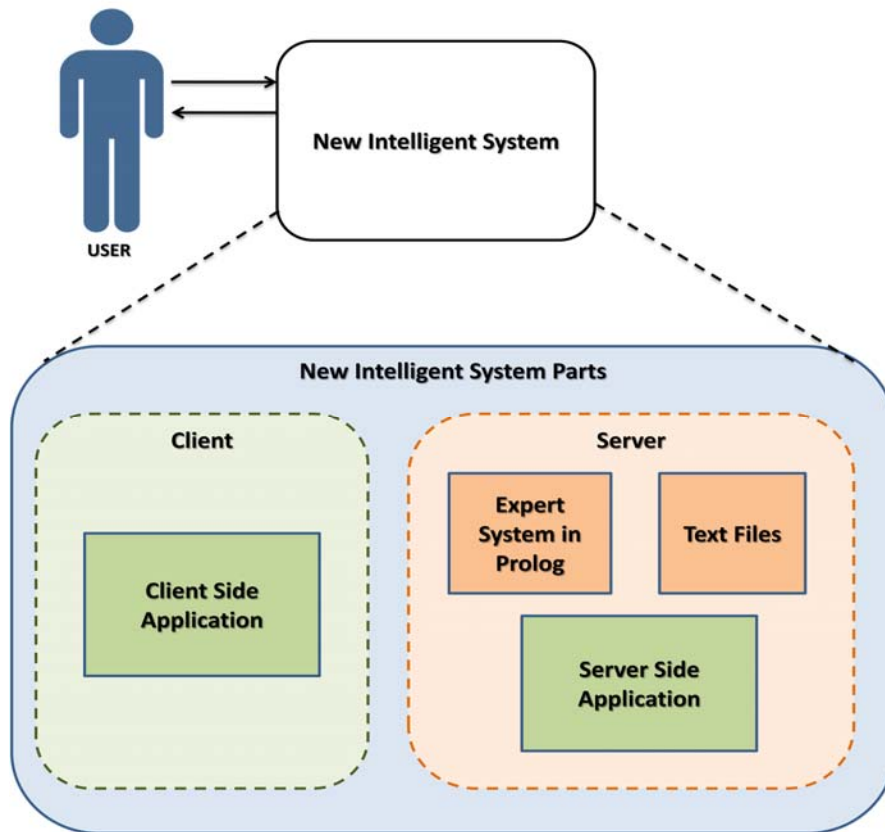
4.2.2 Βήματα δευτέρου μέρους της εργασίας



Εικόνα 17. Τα επιμέρους τμήματα του συστήματος

Αφού ολοκληρώθηκε η μελέτη για υλικό εμπλουτισμού της υπάρχουσας βάσης γνώσης και για τις τεχνολογίες και γλώσσες προγραμματισμού που θα χρησιμοποιούνταν στην υλοποίηση του νέου μέρους του συστήματος του έξυπνου βοηθού που θα αξιοποιούσε την γνώση που συλλέχθηκε αρχικά και θα μπορούσε επίσης να την επεκτείνει, επιλέχθηκαν οι τεχνολογίες και οι γλώσσες προγραμματισμού για την υλοποίηση και άρχισε η σχεδίαση και η ανεύρεση των επιμέρους τμημάτων του συστήματος.

Όπως ήδη αναφέρθηκαν στις προδιαγραφές του συστήματος η αρχιτεκτονική του συστήματος αναμένεται να είναι ένα client – server μοντέλο, όπου ότι υπάρχει στην εφαρμογή του χρήστη είναι αυτό που μπορεί να δει όποιος χρησιμοποιεί την εφαρμογή, ενώ τα τμήματα που είναι στην μεριά του server είναι κρυμμένα από κάθε χρήστη.



Για τον Client και τον Server υλοποιήθηκαν 2 ξεχωριστά C# Windows Forms Application projects. Πιο κάτω παρουσιάζονται κάποιες από τις οθόνες/φόρμες του application για το Clients' side.

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ

ΑΠΟΤΕΛΕΣΜΑΤΑ

Με την ολοκλήρωση της παρούσας διατριβής επιτεύχθηκε η μελέτη διαφόρων περιοχών/τομέων και εννοιών της πληροφορικής που σχετίζονται με την Τεχνητή Νοημοσύνη, ενώ παράλληλα μελετήθηκαν και θέματα ψηυχολογίας παιδιών, μαθησιακών δυσκολιών και σχολικού εκφοβισμού. Είναι εμφανές ότι μέσα από αυτή την εργασία κέρδιθηκαν πολύ περισσότερα σε θεωρητικό επίπεδο μέσα από τη βιβλιογραφική μελέτη που διενεργήθηκε, από όσα κερδίθηκαν σε πρακτικό επίπεδο από την εφαρμογή των όσων μελέτηθηκαν στην υλοποίηση του συστήματος.

Επίσης, κρίθηκε αναγκαίο να αναφερθεί ότι υπήρξαν δυσκολίες και προβλήματα στο θέμα επικοινωνίας μεταξύ γλωσσών προγραμματισμού έτσι η εργασία υλοποιήθηκε βρίσκοντας λύσεις όσο πιο απλές γίνεται κάθε φορά ενώ χάθηκε αρκετός χρόνος και σκάλωνε η εργασία για μέρες σε τέτοιες περιπτώσεις.

Θα μπορούσε να χρησιμοποιηθεί το πακέτο λογισμικού WAMP που περιέχει Apache Server και MySQL για την δημιουργία βάσης δεδομένων, αλλά για αποφυγή αύξησης της πολυπλοκότητας και εμπλοκής περισσότερων τεχνολογιών/εργαλείων, μεθόδων και γλωσσών προγραμματισμού που θα έκαναν το πρόβλημα συμβατότητας και επικοινωνίας μεταξύ των επιμέρους τμημάτων ακόμα δυσκολότερη επιλέχθηκαν να χρησιμοποιηθούν αρχεία κειμένου για την αποθήκευση των δεδομένων.

Bibliography

ALAN DIX, J. F. (2004). HUMAN - COMPUTER INTERACTION (3rd ed.). Pearson Education Limited.

Andrew Sears, J. A. (Ed.). (2007). The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications (2nd ed.). CRC Press.

Bharat Bhushan Agarwal, S. P. (2009). Software Engineering. Laxmi Publications.

Boeck, H. (2009). The Definitive Guide to NetBeans Platform. APRESS.

Bratko, I. (2012). Prolog Programming for Artificial Intelligence (4th ed.). Addison-Wesley.

Claes Wohlin, P. R. (2012). Experimentation in Software Engineering. Springer Science & Business Media.

Dixit, J. B. (2007). Structured System Analysis and Design. Luxmi Publications.

ECKEL, B. (2006). THINKING IN JAVA (4th ed.). PRENTICE HALL.

Graba, J. (2013). An Introduction to Network Programming with Java: Java 7 Compatible (3rd ed.). Springer Science & Business Media.

Ian H. Witten, E. F. (2011). Data Mining: Practical Machine Learning Tools and Techniques (3rd ed.). Burlington, MA : Morgan Kaufmann.

Johnson, B. (2014). Professional Visual Studio 2013. John Wiley & Sons.

Jones, M. T. (2009). Artificial Intelligence: A Systems Approach. Jones & Bartlett Learning.

Kevin A. Gluck, R. W. (Ed.). (2006). Modeling Human Behavior With Integrated Cognitive Architectures: Comparison, Evaluation, and Validation. Psychology Press.

Kurniawan, B. (2012). Servlet & JSP. Brainy Software Inc.

Lloyd, J. W. (2012). Foundations of Logic Programming. Springer Science & Business Media.

LONNIE D. BENTLEY, J. L. (2007). *SYSTEM ANALYSIS & DESIGN FOR THE GLOBAL ENTERPRICE* (7th ed.). McGraw Hill/Irwin.

Michael Stiefel, R. J. (2002). *Application Development Using C# and .NET*. PRENTICE HALL.

Oleg M. Anshakov, T. G. (2010). *Cognitive Reasoning: A Formal Approach*. Springer Science & Business Media.

P.J. DEITEL, H. D. (2008). *INTERNET & WORLD WIDE WEB: HOW TO PROGRAM* (4th ed.). Pearson Education.

Patrick Blackburn, J. B. (2006). *Learn Prolog Now!* College Publications.

Sarmiento, A. (Ed.). (2005). *Issues of Human Computer Interaction*. Idea Group Inc (IGI).

SCHACH, S. R. (1999). *CLASSICAL AND OBJECT ORIENTED SOFTWARE ENGINEERING: WITH UML AND C++* (4th ed.). McGraw Hill/Irwin.

Sebe, N., Cohen, I., Garg, A., & Huang, T. S. (2005). *Machine Learning in Computer Vision*. Dordrecht : Springer.

SEBESTA, R. W. (2010). *CONCEPT OF PROGRAMMING LANGUAGES* (9th ed.). Pearson Education.

Stuart Jonathan Russell, P. N. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Tim Boudreau, J. G. (2003). *NetBeans: The Definitive Guide*. O'Reilly & Associates .

Zain Naboulsi, S. F. (2011). *Coding Faster: Getting More Productive with Microsoft® Visual Studio®: Covers Microsoft® Visual Studio® 2005, 2008, and 2010*. Microsoft Press.

Zambon, G. (2012). *Beginning JSP, JSF and Tomcat: Java Web Development* (2nd ed.). APRESS.

Bratko, I. (2012). *Prolog Programming for Artificial Intelligence* (4th ed.). Addison-Wesley.

Jones, M. T. (2009). Artificial Intelligence: A Systems Approach. Jones & Bartlett Learning.

Patrick Blackburn, J. B. (2006). Learn Prolog Now! College Publications.

Stuart Jonathan Russell, P. N. (2010). Artificial Intelligence: A Modern Approach. Prentice Hall.

ΠΑΡΑΡΤΗΜΑ Α

ΤΜΗΜΑΤΑ ΚΩΔΙΚΑ

A.1 ΜΕΤΑΤΡΟΠΗ ΕΜΠΕΙΡΟΥ ΣΥΣΤΗΜΑΤΟΣ ΣΕ ΚΩΔΙΚΑ PROLOG

Σε αυτό το τμήμα της εργασίας παρουσιάζεται ο κώδικας που προκύπτει από την μετατροπή των τριών ενοιολογικών δικτύων (Semantic Networks) σε κώδικα prolog. Αξίζει να σημειωθεί ότι η βάση της Prolog είναι τα γεγονότα (*facts*), οι κανόνες (*rules*) και οι ερωτήσεις (*queries*). Όταν γίνει consult στο SWI-Prolog των τριών αρχείων που παράχθηκαν μετά την μετατροπή μπορούν να εκτελεστούν διάφορα ερωτήματα (*queries*).

Βάση γνώσης (Knowledge Base):

Η βάση γνώσης είναι η αποθήκη της γνώσης που συλλέχθηκε μέσα από ερωτηματολόγια (online και paper based), προσωπικές συνεντεύξεις, βιβλία και αναζήτηση στο διαδίκτυο. Η γνώση αυτή αρχικά μετατράπηκε από φυσική γλώσσα σε 3 ανεξάρτητα σημασιολογικά δίκτυα υπό μορφή γράφου και ακολούθως μεταφράστηκε σε γλώσσα prolog. Το αποτέλεσμα από την μετατροπή σε κώδικα prolog, το οποίο είναι ένα σύνολο γεγονότων, μαζί με κάποιους κανόνες.

Semantic Network 1 – Κώδικας Prolog

```
parent_category(kyriarxikoi).
```

```
parent_category(epieikis).
```

```
parent_category(paramelitikoi).
```

```
parent_category(aftarxikoi).
```

```
has_kids(kyriarxikoi, sinergatika_me_tous_goneis_tous).
```

```
has_kids(kyriarxikoi, anexartita).
```

```
has_kids(kyriarxikoi, diekdikitika).
```

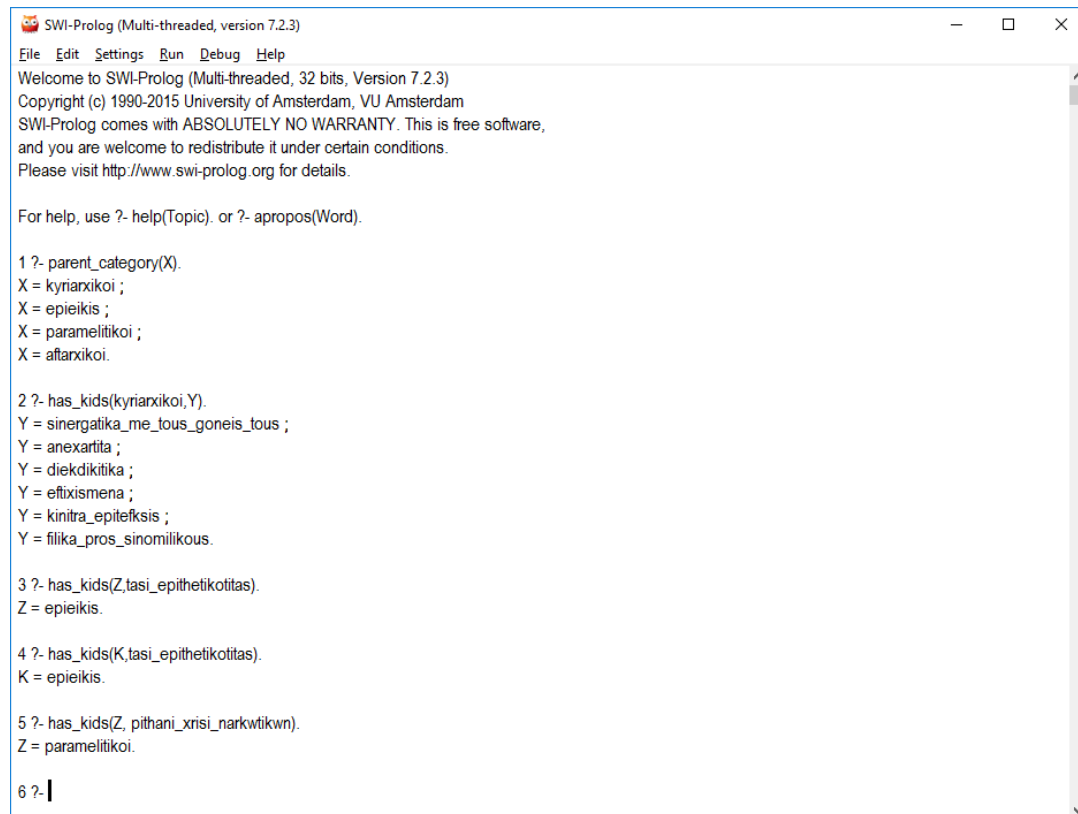
```
has_kids(kyriarxikoi, eftixismena).
```

has_kids(kyriarxikoi, kinitra_epitefksis).
has_kids(kyriarxikoi, filika_pros_sinomilikous).

has_kids(epieikis, tasi_epithetikotitas).
has_kids(epieikis, thetiki_kai_zwiri_diathesi).
has_kids(epieikis, anwrima).
has_kids(epieikis, xwris_aftoelegxo).
has_kids(epieikis, xwris_koinwniki_ypefthinotita).
has_kids(epieikis, xwris_aftodynamia).

has_kids(paramelitikoi, elleipsi_aftosygkentrwsis).
has_kids(paramelitikoi, syxnes_apousies_apo_sxoleio).
has_kids(paramelitikoi, pithani_xrisi_narkwtikwn).
has_kids(paramelitikoi, adynamo_elegxo_sygkinisewn).
has_kids(paramelitikoi, dysthimia).
has_kids(paramelitikoi, paravatika).

has_kids(aftarxikoi, tasi_pros_koinwniki_aposyrsi).
has_kids(aftarxikoi, elleipsi_afthormitismou).
has_kids(aftarxikoi, agoria_me_tasi_epithetikotitas_stous_synimilikous).
has_kids(aftarxikoi, koritsia_eksartimena_xwris_kinitra_epitefksis).



```
SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 7.2.3)
Copyright (c) 1990-2015 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- parent_category(X).
X = kyriarxikoi ;
X = epieikis ;
X = paramelitikoi ;
X = aftarxikoi.

2 ?- has_kids(kyriarxikoi,Y).
Y = sinergatika_me_tous_goneis_tous ;
Y = anexartita ;
Y = diekdikitika ;
Y = eftixismena ;
Y = kinitra_epitefksis ;
Y = filika_pros_sinomilikous.

3 ?- has_kids(Z,tasi_epithetikotitas).
Z = epieikis.

4 ?- has_kids(K,tasi_epithetikotitas).
K = epieikis.

5 ?- has_kids(Z, pithani_xrisi_narkwtikwn).
Z = paramelitikoi.

6 ?- |
```

Εικόνα 18. Παραδείγματα ερωτημάτων στον κώδικα του semantic network 1

Semantic Network 2 – Κώδικας Prolog

```
has_property(mathisiakes_dyskolies, simadia).
has_property(mathisiakes_dyskolies, symptwmata).

isa(simadia, dyskolia_organwsis).
isa(simadia, dyskolia_estiasis_prosoxis).
isa(simadia, dyskolia_apomnimonefsis).
isa(simadia, dyskolia_diaxeirisis_ennoiwn_xronou).
isa(simadia, dyskolia_sta_mathimatika).
isa(simadia, dyskolia_grafis_i_anagnwsis).
isa(simadia, dyskolia_tirisis_odigeiwn).
isa(simadia, kakos_sintonismos).

isa(symptwmata, dyskolia_ekfrasis).
isa(symptwmata, astathis_epidwsi_sxoleiou).
```

isa(symptwmata, anwrimotita_stin_omilia).
 isa(symptwmata, elleipsis_antilipsi_meso_akois).
 isa(symptwmata, dyskolia_diaxeirisis_newn_katastasewn).
 isa(symptwmata, dyskolia_katanoisis_leksewn_i_ennoiwn).
 isa(symptwmata, dyskolia_sygkentrwsis_diaspasi_prosoxis).
 isa(symptwmata, aprepi_antapokrisi_se_erwtiseis).
 isa(symptwmata, parormitiki_symperifora).

```

SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help

1 ?- has_property(X,Y).
X = mathisiakes_dyskolies,
Y = simadia ;
X = mathisiakes_dyskolies,
Y = symptwmata.

2 ?- has_property(mathisiakes_dyskolies,X).
X = simadia ;
X = symptwmata.

3 ?- isa(simadia,Y).
Y = dyskolia_organwsis ;
Y = dyskolia_estiasis_prosoxis ;
Y = dyskolia_apomnimonefsis ;
Y = dyskolia_diaxeirisis_ennoiwn_xronou ;
Y = dyskolia_sta_mathimatika ;
Y = dyskolia_grafis_i_anagnwsis ;
Y = dyskolia_tirisis_odigeiwn ;
Y = kakos_sintonismos.

4 ?- isa(symptwmata,Y).
Y = dyskolia_ekfrasis ;
Y = astathis_epidwsi_sxoleiou ;
Y = anwrimotita_stin_omilia ;
Y = elleipsis_antilipsi_meso_akois ;
Y = dyskolia_diaxeirisis_newn_katastasewn ;
Y = dyskolia_katanoisis_leksewn_i_ennoiwn ;
Y = dyskolia_sygkentrwsis_diaspasi_prosoxis ;
Y = aprepi_antapokrisi_se_erwtiseis ;
Y = parormitiki_symperifora.
  
```

Εικόνα 19. Παραδείγματα ερωτημάτων στον κώδικα του semantic network 2

Semantic Network 3 – Κώδικας Prolog

has_member(bullying,thima).
 has_member(bullying,thitis).

 property(thima, thimwnoun_kai_klaine_efkola).
 property(thima, apomononontai).
 property(thima, paidia_evaisthita_sta_peiragmata).

property(thitis, thewroun_tin_via_koinwnika_apodekti).
property(thitis, entoni_aftopepithisi).
property(thitis, den_pitharxoun_se_kanones).
property(thitis, rolos_arxigou_pareas).
property(thitis, theloun_na_epivallontai).
property(thitis, xristis_narkotikwn).

has_reaction(thima, elleipsi_aftoektimisis).
has_reaction(thima, fovatai_na_xrismopoiisei_to_kinito_tou).
has_reaction(thima, ypervoliki_anisixia).
has_reaction(thima, xanei_sixna_ta_pragmata_tou).
has_reaction(thima, epistrefei_me_simadia_kai_sxismena_rouxa).
has_reaction(thima, adikaiologites_apousies_sto_sxoleio).
has_reaction(thima, taseis_aftoktonias).
has_reaction(thima, peftei_sxoliki_epidosi).
has_reaction(thima, exei_efialtes).
has_reaction(thima, epithetikotita).

has_reason(thitis, vasi_perivallontos).
has_reason(thitis, oxi_vasi_perivallontos).

reason(vasi_perivallontos, kakopoiimeno_paidi).
reason(vasi_perivallontos, martiras_vias_metaksi_gonewn).
reason(vasi_perivallontos, megalwnei_xwris_agapi).
reason(vasi_perivallontos, paramelimenos_apo_oikogenia).
reason(vasi_perivallontos, apousia_isorropias_oikogeneias).
reason(oxi_vasi_perivallontos, diataraksi_diagwgis).

no_balance_by(apousia_isorropias_oikogeneias, kapoios_goneas_apousiazei_i_pethane).
no_balance_by(apousia_isorropias_oikogeneias, kapoios_einai_arrwstos_anasfaleia).

why(thitis, tropos_epiviosis).

```

SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 7.2.3)
Copyright (c) 1990-2015 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- has_member(X,Y).
X = bullying,
Y = thima ;
X = bullying,
Y = thitis.

2 ?- property(thitis,Y).
Y = thewroun_tin_via_koinwnika_apodekti ;
Y = entoni_aftopepithisi ;
Y = den_pitharxoun_se_kanones ;
Y = rolos_arxigou_pareas ;
Y = theloun_na_epivallontai ;
Y = xristis_narkotikwn.

3 ?- property(X,thimwnoun_kai_klaine_efkola).
X = thima.

4 ?- has_reaction(X, exei_efialtes).
X = thima.

5 ?- has_reaction(thima, exei_efialtes).
true.

6 ?- |

```

Εικόνα 20. Παραδείγματα ερωτημάτων στον κώδικα του semantic network 3

Ο κώδικας της αρχικής βάσης γνώσης:

```

:- dynamic kid_characteristic(1).
:- dynamic parents_type(1).
:- dynamic sign_or_symptom(1).
:- dynamic victim_or_victimizer(1).

tobe_filled(parents_type(X)).
tobe_filled(kid_characteristic(X)).
tobe_filled(sign_or_symptom(X)).
tobe_filled(victim_or_victimizer(X)).

% semantic network1
parent_category(kyriarxikoi).
parent_category(epieikis).
parent_category(paramelitikoi).
parent_category(aftarxikoi).

```

has_kids(kyriarxikoi, sinergatika_me_tous_goneis_tous).
has_kids(kyriarxikoi, anexartita).
has_kids(kyriarxikoi, diekdikitika).
has_kids(kyriarxikoi, eftixismena).
has_kids(kyriarxikoi, kinitra_epitefksis).
has_kids(kyriarxikoi, filika_pros_sinomilikous).

has_kids(epieikis, tasi_epithetikotitas).
has_kids(epieikis, thetiki_kai_zwiri_diathesi).
has_kids(epieikis, anwrima).
has_kids(epieikis, xwris_aftoelegxo).
has_kids(epieikis, xwris_koinwniki_ypeffthinotita).
has_kids(epieikis, xwris_aftodynamia).

has_kids(paramelitikoi, elleipsi_aftosygkentrwsis).
has_kids(paramelitikoi, syxnes_apousies_apo_sxoleio).
has_kids(paramelitikoi, pithani_xrisi_narkwtikwn).
has_kids(paramelitikoi, adynamo_elegxo_sygkinisewn).
has_kids(paramelitikoi, dysthimia).
has_kids(paramelitikoi, paravatika).

has_kids(aftarxikoi, tasi_pros_koinwniki_aposyrsi).
has_kids(aftarxikoi, elleipsi_afthormitismou).
has_kids(aftarxikoi, agoria_me_tasi_epithetikotitas_stous_synimilikous).
has_kids(aftarxikoi, koritsia_eksartimena_xwris_kinitra_epitefksis).

% semantic network 2

has_property(mathisiakes_dyskolies, simadia).
has_property(mathisiakes_dyskolies, symptwmata).

isa(simadia, dyskolia_organwsis).
isa(simadia, dyskolia_estiasis_prosoxis).
isa(simadia, dyskolia_apomnimonefsis).
isa(simadia, dyskolia_diaxeirisis_ennoiwn_xronou).
isa(simadia, dyskolia_sta_mathimatika).
isa(simadia, dyskolia_grafis_i_anagnwsis).

isa(simadia, dyskolia_tirisis_odigeiwn).

isa(simadia, kakos_sintonismos).

isa(symptwmata, dyskolia_ekfrasis).

isa(symptwmata, astathis_epidwsi_sxoleiou).

isa(symptwmata, anwrimotita_stin_omilia).

isa(symptwmata, elleipsis_antilipsi_meso_akois).

isa(symptwmata, dyskolia_diaxeirisis_newn_katastasewn).

isa(symptwmata, dyskolia_katanoisis_leksewn_i_ennoiwn).

isa(symptwmata, dyskolia_sygkentrwsis_diaspasi_prosoxis).

isa(symptwmata, aprepi_antapokrisi_se_erwtiseis).

isa(symptwmata, parormitiki_symperifora).

% semantic network 3

has_member(bullying,thima).

has_member(bullying,thitis).

property(thima, thimwnoun_kai_klaine_ekola).

property(thima, apomononontai).

property(thima, paidia_evaisthita_sta_peiragmata).

property(thitis, thewroun_tin_via_koinwnika_apodekti).

property(thitis, entoni_aftoepithisi).

property(thitis, den_pitharxoun_se_kanones).

property(thitis, rolos_arxigou_pareas).

property(thitis, theloun_na_epivallontai).

property(thitis, xristis_narkotikwn).

has_reaction(thima, elleipsi_aftoektimisis).

has_reaction(thima, fovatai_na_xrisimopoiisei_to_kinito_tou).

has_reaction(thima, ypervoliki_anisixia).

has_reaction(thima, xanei_sixna_ta_pragmata_tou).

has_reaction(thima, epistrefei_me_simadia_kai_sxismena_rouxa).

has_reaction(thima, adikaiologites_apousies_sto_sxoleio).

has_reaction(thima, taseis_aktontias).

has_reaction(thima, peftei_sxoliki_epidosi).

has_reaction(thima, exei_efialtes).

has_reaction(thima, epithetikotita).

has_reason(thitis, vasi_perivallontos).

has_reason(thitis, oxi_vasi_perivallontos).

reason(vasi_perivallontos, kakopoiimeno_paidi).

reason(vasi_perivallontos, martiras_vias_metaksi_gonewn).

reason(vasi_perivallontos, megalwnei_xwris_agapi).

reason(vasi_perivallontos, paramelimenos_apo_oikogenia).

reason(vasi_perivallontos, apousia_isorropias_oikogeneias).

reason(oxi_vasi_perivallontos, diataraksi_diagwgis).

no_balance_by(apousia_isorropias_oikogeneias, kapoios_goneas_apousiazei_i_pethane).

no_balance_by(apousia_isorropias_oikogeneias, kapoios_einai_arrwstos_anasfaleia).

why(thitis, tropos_epiviosis).

isa(epithetikes_antidraseis, skisimo_diagwnismatos_arnisi_na_grapsei_esto_onoma).

isa(epithetikes_antidraseis, apeiles_pros_ton_ekpaideftiko).

isa(epithetikes_antidraseis, xeirodikia_apenanti_se_sinomilikous).

isa(epithetikes_antidraseis, thimos_kai_egkatalipsi_tis_taksis).

%rules

append([],X,X). % base (for lists)

append([X|Y],Z,[X|W]) :- append(Y,Z,W). % recursive

insertAtEnd(X,[],[X]). % base (for lists)

insertAtEnd(X,[H|T],[H|Z]) :- insertAtEnd(X,T,Z). % recursive

parent_type_exist(X):-

parents_type(X),parent_category(X).

which_parent_type(Y):-

kid_characteristic(X),has_kids(Y, X).

has_mathisiakes_dyskolies(X):-

isa(X,Y),sign_or_symptom(Y).

```
scholikos_ekfovismos_member(X):-  
  property(X,Y),victim_or_victimizer(Y).
```

Μηχανισμός Εξαγωγής Συμπερασμάτων (Interference Engine):

Για το τμήμα αυτό υπάρχουν διάφοροι έτοιμοι μηχανισμοί διαθέσιμοι στο διαδίκτυο σε αυτή την γλώσσα η οποία διδάσκεται σε διάφορες σχολές και πανεπιστήμια στο μάθημα του λογικού προγραμματισμού. Στο συγκεκριμένο σύστημα χρησιμοποιήθηκε ένας τέτοιος μηχανισμός.

Ο μηχανισμός συμπεράσματος είναι το πρόγραμμα που εντοπίζει την κατάλληλη γνώση στη βάση γνώσεων, και συμπεραίνει τη νέα γνώση με την εφαρμογή των λογικών στρατηγικών επεξεργασίας και επίλυσης προβλήματος.

Ακολουθεί ο κώδικας του μηχανισμού αυτού:

```
:- dynamic already_asked(1).  
:- dynamic already_answered(1).  
  
process(true, 'match with a fact', Rules) :- !.  
  
process((Goal, Goals), and(Proof, Proofs), Rules) :- !,  
  process(Goal, Proof, Rules), process(Goals, Proofs, Rules).  
  
process(Goal, by(Goal, Proof), Rules) :-  
  clause(Goal, SubGoals), process(SubGoals, Proof, [(Goal :- SubGoals)|Rules]).  
  
process(Goal, by(Goal, user_given), Rules) :-  
  tobe_filled(Goal), Goal =.. [Pred, Arg],  
  already_asked(Pred), !, already_answered(Goal).  
  
process(Goal, by(Goal, user_given), Rules) :-  
  tobe_filled(Goal),  
  Goal =.. [Pred, Arg], Goalx =.. [Pred, X],  
  ask_user(Goalx, Rules), assert(already_asked(Pred)),  
  assert(already_answered(Goalx)), Goal = Goalx.
```

Διεπαφή Χρήστη (User Interface):

Αυτό το τμήμα αποτελείται από ένα μικρό menu επιλογών με πιθανές ερωτήσεις χρηστών προς το έμπειρο σύστημα. Η διεπαφή με τον χρήστη είναι ο τρόπος επικοινωνίας μεταξύ ενός χρήστη και των διαδικασιών επίλυσης προβλήματος έμπειρων συστημάτων. Ένα καλό έμπειρο σύστημα δεν είναι πολύ χρήσιμο εκτός αν έχει μια αποτελεσματική διεπαφή.

Ακολουθεί ο κώδικας της διεπαφής χρήστη:

```
expert :- write('Hello!!! This is an expert system!!!'), menu.
```

```
menu :- nl, write('Choose your question:'),
        nl, write('1 Yparxei aftos o typos gonewn?'),
        nl, write('2 Poios einai o typos twn gonewn aftou tou paidiou?'),
        nl, write('3 Exei afto to paidi mathisiakes dyskoliies?'),
        nl, write('4 Einai pithanos thitis i thima bullying?'),
        nl, write('5 Exit'),
        nl, write('Please choose: '), read(X), option(X).
```

```
option(1) :- process(parent_type_exist(X), Proof, []), nl,
              write('Yparxei o typos: '),
              write(X), can_explain(Proof), menu.
```

```
option(2) :- process(which_parent_type(X), Proof, []), nl,
              write('O goneikos typos einai: '),
              write(X), can_explain(Proof), menu.
```

```
option(3) :- process(has_mathisiakes_dyskoliies(X), Proof, []), nl,
              write('Anikei sta '),
              write(X), can_explain(Proof), menu.
```

```
option(4) :- process(sxolikos_ekfovismos_member(X), Proof, []), nl,
              write('Anikei stis idiotites tou '),
              write(X), can_explain(Proof), menu.
```

```
option(5) :- retract(already_asked(X)), retract(already_answered(Y)), fail.
```

```
option(5) :- nl, write('Goodbye!').
```

```
option(X) :- out_of_range(X), nl, write('No such option - please try again.'), menu.
```

```
option(_) :- nl, write('Sorry Den mporw na vrw tetoia dedomena sti vasi gnwsis'), menu.
```

```
out_of_range(X) :- not(member(X, [1, 2, 3, 4, 5, 6])).
```

```
can_explain(Proof) :- nl, write('Do you want to know how, yes or no? '),  
                    read(X), explain_respond(X, Proof).
```

```
explain_respond(no, _).
```

```
explain_respond(yes, Proof) :-  
    explain(Proof, 2).
```

```
explain(by(Goal, Proof), Intent) :-  
    tab(Intent), write(Goal), write(' by '),  
    explain_step(Proof, Intent).
```

```
explain(and(Proof1, Proof2), Intent) :-  
    explain(Proof1, Intent), nl, tab(Intent),  
    write('and'), nl, nl, explain(Proof2, Intent).
```

```
explain_step('match with a fact', _) :- !, write('match with a fact'), nl.
```

```
explain_step(user_given, _) :- !, write('user given'), nl.
```

```
explain_step(Proof, Intent) :- NewIn is Intent + 4, nl, nl, explain(Proof, NewIn).
```

```
ask_user(parents_type(X), Rules) :-  
    nl, write('Dwste pithano typo gonewn:'),  
    nl, write(': '), read(Y), respond(Y, parents_type(X), Rules).
```

```
ask_user(kid_characteristic(X), Rules) :-  
    nl, write('Dwste xarakteristiko paidiou:'),  
    nl, write(': '), read(Y), respond(Y, kid_characteristic(X), Rules).
```

```
ask_user(sign_or_symptom(X), Rules) :-
    nl, write('Dwste pithano simadi i symptwma:'),
    nl, write(': '), read(Y), respond(Y, sign_or_symptom(X), Rules).

ask_user(victim_or_victimizer(X), Rules) :-
    nl, write('Dwste pithani idiotita thiti i thimatos:'),
    nl, write(': '), read(Y), respond(Y, victim_or_victimizer(X), Rules).

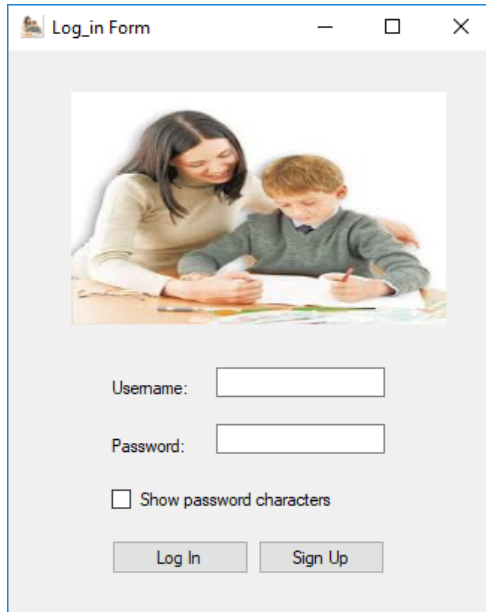
respond(why, Goal, [Rule|Rules]) :- !, nl, display_rule(Rule), nl, ask_user(Goal, Rules).

respond(why, Goal, []) :- !, nl, write('No more explanation!'), nl, ask_user(Goal, []).

respond(Ans, Goal, Rules) :- Goal =.. [Pred, Arg], Arg = Ans.

display_rule((Goal :- SubGoals)) :- write(Goal), write(' :- '), write(SubGoals).
```

A.2 ΟΘΟΝΕΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ



Log_in Form

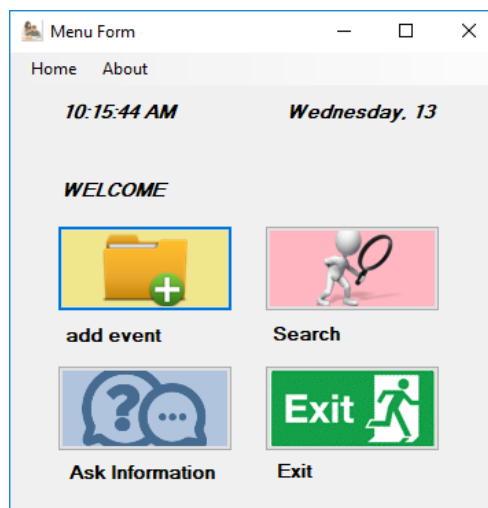
Username:

Password:

Show password characters

Log In Sign Up

Εικόνα 21. Φόρμα για το Log in στο σύστημα



Menu Form

Home About

10:15:44 AM Wednesday, 13

WELCOME

add event Search

Ask Information Exit

Εικόνα 22. Αρχική φόρμα του έξυπνου βοηθού μετά το Log in

Registration Form

Username

Password

Confirm Password:

Εικόνα 23. Φόρμα εγγραφής/δημιουργίας λογαριασμού νέου χρήστη