



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Prolog Tutor: Ένα προσαρμοστικό και αλληλεπιδραστικό σύστημα για τη διδασκαλία της Prolog
Όνοματεπώνυμο Φοιτητή	Ορουγιέ Εκινέ Ρενάτα Ειρήνη
Πατρώνυμο	Philip
Αριθμός Μητρώου	ΜΠΠΛ/09043
Επιβλέπων	Θέμης Παναγιωτόπουλος, Βαθμίδα

Ημερομηνία Παράδοσης

Δεκέμβριος 2012

Prolog Tutor: Ένα προσαρμοστικό και αλληλεπιδραστικό σύστημα για τη διδασκαλία της Prolog

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Ευχαριστίες

Η παρούσα μεταπτυχιακή διατριβή πραγματοποιήθηκε υπό την επίβλεψη του καθηγητή Θεμιστοκλή Παναγιωτόπουλου τον οποίο θα ήθελα να ευχαριστώ θερμά για την συμπαράστασή του καθ' όλη τη διάρκεια της συνεργασίας μας. Επίσης, θα ήθελα να ευχαριστήσω και τον Αναστασάκη Γεώργιο για τις πολύτιμες συμβουλές και υποδείξεις του.

Τέλος, θα ήθελα να ευχαριστήσω όσους ήταν κοντά μου κατά την εκπόνηση της διατριβής μου, για την υπομονή, τη συμπαράσταση και την κατανόησή τους.

Περίληψη

Αντικείμενο της παρούσας μεταπτυχιακής διατριβής αποτελεί η δημιουργία ενός web-based ευφυούς προσαρμοστικού συστήματος για την αλληλεπιδραστική εκμάθηση της γλώσσας προγραμματισμού Prolog.

Το προσαρμοστικό αυτό σύστημα δύναται να χρησιμοποιηθεί από φοιτητές προπτυχιακών και μεταπτυχιακών πανεπιστημιακών τμημάτων όπου διδάσκεται η Prolog τόσο για αυτόνομη , εξ' αποστάσεως διδασκαλία, όσο και για συμπληρωματική χρήση στις πανεπιστημιακές διαλέξεις.

Η ύλη και τα παραδείγματα που διατίθενται, έχουν χωριστεί σε τέσσερα επίπεδα δυσκολίας ανάλογα τον τύπο μοντέλου που υπάγονται οι φοιτητές. Μέτριος, καλός, πολύ καλός και άριστος. Βάσει το μοντέλου αυτού ο γεννήτορας των ερωτήσεων του τεστ αξιολόγησης, η παρεχόμενη θεωρία και παραδείγματα προσαρμόζονται στο κάθε χρήστη ξεχωριστά.

Abstract

Objective of this master thesis is to create a web-based intelligent adaptive system for the interactive learning of the programming language Prolog.

This adaptive system can be used by students of undergraduate and postgraduate university departments which teach Prolog both for autonomous, remote teaching, and for complementary use of university lectures.

The material and examples available are divided into four levels of difficulty depending on the type of model users fall into. The user models are average, good, very good and excellent. Based on users' model, the questions generator provides theory and examples fully tailored to each user's needs.

Περιεχόμενα

1. Εισαγωγή	7
2. Θεωρητικό Υπόβαθρο	8
2.1. Εκπαιδευτικό λογισμικό και εξ' αποστάσεως εκπαίδευση.....	8
2.2. Μορφές εξ' αποστάσεως εκπαίδευσης.	8
2.3. Προσαρμοστικά εκπαιδευτικά συστήματα διδασκαλίας	9
3. Λογικός προγραμματισμός και Prolog: Η ιστορία, η «ιδιομορφία» της και η αξία της	11
3.1. Prolog: Η γένεσή της.....	11
3.2. «Συμβατικές» γλώσσες προγραμματισμού και Prolog.....	11
3.3. Prolog: Η ιδιομορφία της και η αξία της.....	11
4. Ανάπτυξη ευφυούς προσαρμοστικού web-based εκπαιδευτικού συστήματος για την αλληλεπιδραστική εκμάθηση της Prolog.....	13
4.1. Σύλληψη και Ανάλυση Απαιτήσεων του Συστήματος	13
4.1.1. Γενική θεώρηση Prolog Tutor	15
4.1.2. Λειτουργικές απαιτήσεις Prolog Tutor	15
4.1.3. Μη λειτουργικές απαιτήσεις του Prolog Tutor	16
4.1.4. Διαγράμματα περιπτώσεων χρήσης (use case diagrams)	17
4.2. Σχεδιασμός του συστήματος.....	18
4.2.1 Βάση Δεδομένων	18
4.2.2 Μοντελοποίηση χρήστη και γεννήτορας ασκήσεων.....	18
4.3. Υλοποίηση του συστήματος διδασκαλίας Prolog, Prolog Tutor ...	20
4.3.1. Βάση Δεδομένων	20
4.3.2. Παρουσίαση του συστήματος διδασκαλίας Prolog, Prolog Tutor	23

Prolog Tutor: Ένα προσαρμοστικό και αλληλεπιδραστικό σύστημα για τη διδασκαλία της Prolog

Μεταπτυχιακή Διατριβή

5. Συμπεράσματα.....	52
6. Βιβλιογραφία.....	54

1. Εισαγωγή

Κύριο αντικείμενο της παρούσας μεταπτυχιακής διατριβής, όπως ορίζεται και στον τίτλο της είναι η ανάλυση, ο σχεδιασμός και τελικά η δημιουργία ενός προσαρμοστικού και αλληλεπιδραστικού web based εκπαιδευτικού λογισμικού για την εκμάθηση της γλώσσας προγραμματισμού Prolog.

Η επιλογή της εν λόγω γλώσσας έγινε για δυο λόγους. Πρώτον γιατί η Prolog είναι άρρηκτα συνδεδεμένη με την έννοια της Τεχνητής Νοημοσύνης και άρα η εκμάθηση και συνεπακόλουθα η χρήση της κρίνεται επιτακτική και δεύτερον, ο τρόπος σκέψης που επιβάλλει η Prolog (mindset) είναι δύσκολο να κατανοηθεί από φοιτητές που έχουν ήδη γνώση κλασικών γλωσσών προγραμματισμού.

Στο σύστημά μας κάθε κεφάλαιο της παρεχόμενης ύλης απαρτίζεται όχι μόνο από τη θεωρία, τα παραδείγματα και πολυμεσικά υποστηρικτικά μέσα (animation, video, κλπ) που προσφέρουν διάφορα υπάρχοντα συστήματα διδασκαλίας Prolog (π.χ. αυτό των (Βλαχάβας, Ρεφανίδης, & Σακελλαρίου) αλλά και από ασκήσεις εμπέδωσης εννοιών που εξετάζει το εκάστοτε κεφάλαιο δίνοντας τη δυνατότητα να εκτελεί εκείνη τη στιγμή ο χρήστης τα ερωτήματα και να διαμορφώνει έτσι μια πολύ στερεή κατανόηση τους.

Αυτή η νέα δυνατότητα που παρέχεται από το σύστημά μας, η αλληλεπιδραστική εκτέλεση ερωτημάτων και η εμφάνισή τους στο χρήστη τη στιγμή που τα εκτελεί μπορεί να κάνει το χρήστη να συμμετέχει πιο ενεργά στη διαδικασία μάθησης της Prolog.

Η εκμάθηση της Prolog από τους φοιτητές διαφαίνεται να είναι ένα αρκετά δύσκολο εγχείρημα και αυτή η δυσκολία έγκειται στη φύση της εν λόγω γλώσσας, η οποία και την διαφοροποιεί από τις υπόλοιπες, τις «συμβατικές» γλώσσες προγραμματισμού.

Στη συνέχεια της μεταπτυχιακής διατριβής εξετάζεται το θεωρητικό υπόβαθρο που άπτεται του εκπαιδευτικού λογισμικού, της εξ' αποστάσεως εκπαίδευσης (e-learning) καθώς και των προσαρμοστικών εκπαιδευτικών συστημάτων στο Διαδίκτυο (Κεφάλαιο 2). Το κεφάλαιο 3 πραγματεύεται τη «γένεση» και τις ιδιομορφίες που χαρακτηρίζουν τη Prolog ,τις ειδοποιούς διαφορές σε σχέση με άλλες γλώσσες προγραμματισμού καθώς και τους λόγους για τους οποίους επελέγη η συγκεκριμένη γλώσσα προγραμματισμού.

Ακολουθεί το κεφάλαιο 4 στο οποίο γίνεται καταγραφή της Ανάλυσης απαιτήσεων του Συστήματός μας, όπου αναλύονται οι λόγοι που κατά τη συγγραφέα κρίνεται επιτακτική η αλληλεπιδραστική εκμάθηση της εν λόγω γλώσσας και ακολουθούν ο σχεδιασμός και η υλοποίησή του.

2. Θεωρητικό Υπόβαθρο

Η παρουσία των προσωπικών υπολογιστών στην εκπαίδευση ήταν αρχικά συνυφασμένη με διαχειριστικά και οργανωτικά ζητήματα. Με την πάροδο όμως των χρόνων και την ολοένα αυξανόμενη και αναδυόμενη ανάγκη του πληροφορικού αλφαριθμητισμού, οι υπολογιστές συμπεριλήφθηκαν στην εκπαιδευτική διαδικασία, όχι μόνο ως εργαλείο διδασκαλίας και μάθησης στα διάφορα γνωστικά αντικείμενα, αλλά και ως γνωστικό αντικείμενο καθεαυτό και ως γνωστικό εργαλείο¹ για την έκφραση και οικοδόμηση γνώσεων. Η εξέλιξη των δικτύων υπολογιστών (γνωστό και ως διαδίκτυο) αλλά και η ανάπτυξη των πολυμέσων ήταν παράγοντες που βοήθησαν στην ευρεία χρήση των υπολογιστών στην εκπαίδευση.

Οι καθηγητές προκειμένου να κρατούν αμείωτο το ενδιαφέρον των μαθητών τους καθ' όλη τη διάρκεια της εκπαιδευτικής διαδικασίας, να βοηθήσουν στην ενεργή και στέρεη οικοδόμηση γνώσης από τους μαθητές αλλά και για να επιτύχουν καλύτερα μαθησιακά αποτελέσματα κάνουν χρήση των παρεχόμενων τεχνολογιών, παράδειγμα των οποίων αποτελεί και το εκπαιδευτικό λογισμικό.

2.1. Εκπαιδευτικό λογισμικό και εξ' αποστάσεως εκπαίδευση.

Με τον όρο εκπαιδευτικό λογισμικό εννοούμε όλα εκείνα τα προγράμματα τα οποία κάνουν χρήση διαφόρων μέσων για την παρουσίαση της προς μετάδοση πληροφορίας, όπως αυτά του κειμένου, του ήχου, της εικόνας και του βίντεο με κύριο στόχο τους τη πρόσβαση του χρήστη στην πληροφορία και την αποτελεσματική μάθηση.

Θα μπορούσε με ασφάλεια να λεχθεί πως το εκπαιδευτικό λογισμικό αποτελεί πρωτόλεια μορφή της (ασύγχρονης) εξ' αποστάσεως εκπαίδευσης.

Με το πέρασμα των χρόνων και τη ραγδαία ανάπτυξη της τεχνολογίας και του Διαδικτύου, η χρήση του οποίου υπεισέρχεται σε όλες τις εκφάνσεις της ανθρώπινης δραστηριότητας, η εκπαίδευση από απόσταση παίρνει πλέον νέα μορφή. Μιλάμε πλέον για e-learning όρος ο οποίος αποδίδει τη διάχυση του εκπαιδευτικού περιεχομένου με τη βοήθεια ηλεκτρονικών μέσων.

2.2. Μορφές εξ' αποστάσεως εκπαίδευσης.

Οι μορφές μάθησης που μπορεί να λάβει αυτού του τύπου η εκπαίδευση με βάση τον τόπο και το χρόνο που λαμβάνει χώρα η μάθηση αλλά και το αν βασίζονται στους ηλεκτρονικούς υπολογιστές ή/και τις τεχνολογίες του Διαδικτύου μπορούν να κατηγοριοποιηθούν ως εξής:

- Με βάση το «χωροχρόνο» της μάθησης έχουμε:
 - τη σύγχρονη και
 - ασύγχρονη εκπαίδευση
- Με βάση το αν βασίζονται στους ηλεκτρονικούς υπολογιστές ή/και τις τεχνολογίες του Διαδικτύου έχουμε:
 - Computer-based learning: όπως ελέχθη και νωρίτερα αυτός ο τύπος εκπαίδευσης αποτελεί πρωτόλεια μορφή του σημερινού e-learning. Βασίζεται κυρίως στην τεχνολογία των υπολογιστών. Παράδειγμα τέτοιας μορφής αποτελεί το εκπαιδευτικό λογισμικό (cd rom).

¹ Με τον όρο γνωστικό εργαλείο (cognitive tool) ή νοητικό εργαλείο (mind tool) εννοούμε τις εφαρμογές και τα περιβάλλοντα που έχουν δημιουργηθεί ή προσαρμοστεί έτσι ώστε να λειτουργούν ως διανοητικοί συνεργάτες του μαθητή υποστηρίζοντας και ενισχύοντας την κριτική σκέψη και την ανάπτυξη γνώσεων και δεξιοτήτων υψηλού επιπέδου. (σελ.114, (Κόμης, 2004))

- Web-based learning: σε αυτή τη μορφή εκπαίδευσης γίνεται χρήση κυρίως των τεχνολογιών Διαδικτύου ή ενός δικτύου που ανήκει σε έναν φορέα όπως για παράδειγμα μια εταιρεία ή ένας εκπαιδευτικός οργανισμός (σχολείο, πανεπιστήμιο κλπ). Σε αυτού του τύπου μάθησης δίνεται η δυνατότητα επικοινωνίας με τους διδάσκοντες αλλά και άλλους εκπαιδευόμενους μέσω chat rooms και forums. Μερικά από τα θετικά που προσφέρονται είναι εύκολη πρόσβαση στην παρεχόμενη πληροφορία ανεξάρτητα από χωρο-χρονικούς περιορισμούς. Anytime και anywhere πρόσβαση. Δίνεται η δυνατότητα στο υποκείμενο να μαθαίνει με τους δικούς του «όρους», είτε αυτό σημαίνει από τη βολή του σπιτιού του, είτε την ώρα που πηγαίνει στη δουλειά του, είτε την ώρα που ξεκουράζεται. Ακόμη, κάτι αρκετά σημαντικό είναι η γρηγορότερη ανανέωση του περιεχομένου και έτσι η μεταδιδόμενη γνώση είναι εύκολο να επικαιροποιηθεί.

2.3. Προσαρμοστικά εκπαιδευτικά συστήματα διδασκαλίας.

Τελευταία παρατηρείται έντονα η εισχώρηση του Διαδικτύου στις περισσότερες εκφάνσεις της ανθρώπινης δραστηριότητας. Το Διαδίκτυο χρησιμοποιείται στην επικοινωνία, στην αναζήτηση πληροφοριών, στη διαφήμιση, στο εμπόριο (Chrysafiadi & Virvou, 2008) και όπως ήταν αναμενόμενο εισχώρησε και στο χώρο της εκπαίδευσης.

Ενώ τα εκπαιδευτικά συστήματα διδασκαλίας που βασίζονται στο Διαδίκτυο παρέχουν στο χρήστη τους την πρόσβαση σε αυτά ασύγχρονα είναι η απουσία του δασκάλου (Chrysafiadi & Virvou, 2008) καθώς και η προσέγγιση one-size-fits-all που καθιστούν επιτακτική την ανάγκη για τη δημιουργία διαδικτυακών εκπαιδευτικών συστημάτων που θα λαμβάνουν υπόψη τους μοναδικά χαρακτηριστικά, ανάγκες, στόχους, ενδιαφέροντα και προτιμήσεις των χρηστών του (Brusilovsky, 1998).

Στα «απλά» διαδικτυακά εκπαιδευτικά συστήματα η ύλη ενός κεφαλαίου μπορεί να είναι κατανοητή από ένα καλό μαθητή, βαρετή για κάποιον που είναι άριστος ή να μπερδεύει κάποιον ο οποίος δεν είναι και τόσο καλός μαθητής. Θα πρέπει να βρεθεί ένας τρόπος όπου το εκπαιδευτικό σύστημα να ικανοποιεί τους ετερογενείς χρήστες του που μεταξύ των άλλων έχουν ως κύριο στόχο τη μάθηση.

Αυτό στο οποίο τα παραπάνω συστήματα αποτυγχάνουν να ικανοποιήσουν και τα προσαρμοστικά εκπαιδευτικά συστήματα επιχειρούν να λύσουν είναι η αδυναμία των πρώτων να μοντελοποιήσουν τους χρήστες τους και να προσαρμόσουν τη συμπεριφορά τους βάσει του μοντέλου στο οποίο υπάγονται.

Ένα προσαρμοστικό σύστημα μπορεί να προσαρμοστεί με βάση τα εξής:

- ✓ Τους στόχους των χρηστών του
- ✓ Την πρότερη γνώση τους
- ✓ Το υπόβαθρό τους
- ✓ Τις προτιμήσεις τους

Κάτι επίσης σημαντικό στα προσαρμοστικά συστήματα πέρα από τα κριτήρια προσαρμογής τους είναι και το TI μπορεί να προσαρμοστεί. Ποιο μέρος, δηλαδή, του συστήματος μπορεί να αλλάξει/προσαρμόζεται σε διαφορετικούς χρήστες.

Με βάση λοιπόν αυτό που προσαρμόζεται έχουμε την προσαρμοστική παρουσίαση και την προσαρμοστική πλοήγηση.

Στην προσαρμοστική παρουσίαση, όπως είναι ευκόλως εννοούμενο, προσαρμόζεται το περιεχόμενο της σελίδας σύμφωνα με τα ιδιαίτερα χαρακτηριστικά του χρήστη (Brusilovsky, 1998) (γνώση, στόχοι κλπ). Για παράδειγμα ένας προχωρημένος χρήστης θα έχει πρόσβαση σε περιεχόμενο που είναι πιο ενδιαφέρον για αυτόν. Μπορεί δηλαδή να έχεις λιγότερες λεπτομέρειες, να κάνει πιο δύσκολα παραδείγματα ενώ ένας λιγότερο προχωρημένος χρήστης θα μπορεί να έχει πρόσβαση σε περιεχόμενο όπου θα αποσαφηνίζονται διάφοροι περίπλοκοι όροι και θα κάνει περισσότερα παραδείγματα.

Έτσι λοιπόν, χρήστες που υπάγονται σε διαφορετικά μοντέλα χρηστών θα έχουν πρόσβαση σε διαφορετική θεωρία και παραδείγματα.

Στην προσαρμοστική πλοήγηση, επιχειρείται η παράθεση συνδέσμων που ο χρήστης μπορεί να ακολουθήσει σύμφωνα με το μοντέλο στο οποίο υπάγεται. Για παράδειγμα, κάποιος ο οποίος δεν είναι προχωρημένος το σύστημα θα του υποδεικνύει τι θα ήταν καλό να διαβάσει στη συνέχεια, να το αποκρύψει κάποιους συνδέσμους ή ακόμη, με τη μέθοδο του adaptive annotation να παρέχει σχόλια δίπλα από κάθε σύνδεσμο για το αν χρειάζεται να ξαναδιαβάσει μια ενότητα, αν η εκμάθηση κάποιας άλλης ήταν επιτυχής κλπ.

3. Λογικός προγραμματισμός και Prolog: Η ιστορία, η «ιδιομορφία» της και η αξία της

3.1. Prolog: Η γένεσή της.

Η γένεση της Prolog έλαβε χώρα στη Γαλλία και αποτελεί παράπλευρο προϊόν ενός εγχειρήματος των Alain Colmerauer και Philippe Roussel που σκοπό είχε τη δημιουργία ενός επεξεργαστή της γαλλικής γλώσσας. Έτσι λοιπόν, το 1971 έχουμε μια πρωτόλεια μορφή της Prolog η οποία προς τα τέλη του 1972 θα λάβει την τελική της μορφή (Colmerauer & Roussel, 1992). Σήμερα η Prolog θεωρείται μια από τις πιο επιτυχημένες γλώσσες προγραμματισμού για τον προγραμματισμό εφαρμογών Τεχνητής Νοημοσύνης, όπως είναι τα έμπειρα συστήματα και τα συστήματα διδασκαλίας.

Η ονομασία Prolog επιλέχθηκε από τον Philippe Roussel ως σύντμηση των λέξεων PROgrammation en LOGique, Προγραμματίζοντας δηλαδή στη Λογική, για να δηλώσει το εργαλείο εκείνο που καθιστά δυνατή την επικοινωνία ανθρώπου και υπολογιστή διαμέσου της φυσικής γλώσσας και εν προκειμένω της γαλλικής.

3.2. «Συμβατικές» γλώσσες προγραμματισμού και Prolog.

Οι διαδικαστικές γλώσσες προγραμματισμού ήταν για πάρα πολλά χρόνια το μόνο «είδος» γλωσσών που χρησιμοποιούσαν οι προγραμματιστές για να αναπτύξουν τα προγράμματά τους. Μάλιστα η επίδρασή τους ήταν τέτοια που μέχρι και σήμερα πολλοί αγνοούν την ύπαρξη και άλλου τύπου γλωσσών προγραμματισμού.

Με τον όρο διαδικαστικός ή αλγοριθμικός προγραμματισμός εννοούμε την εκτέλεση μιας αυστηρά καθορισμένης ακολουθίας ενεργειών (εντολών) που θα οδηγήσουν στην επιτυχή επίλυση του προβλήματος. Αυτού του τύπου ο προγραμματισμός προϋποθέτει πως ο προγραμματιστής γνωρίζει εκ των προτέρων τον αλγόριθμο επίλυσης του προβλήματος.

Σε μια τέτοια περίπτωση θα μπορούσαμε ενδεχομένως να προβούμε στην επίλυσή του και χωρίς τη χρήση υπολογιστή αρκεί να διαθέτουμε μολύβι, χαρτί και πολύ χρόνο. Στην ουσία λοιπόν ο υπολογιστής περισσότερο χρησιμοποιείται σαν μια γρήγορη αριθμομηχανή παρά σαν ένα «έξυπνο» μηχάνημα. Είναι ο άνθρωπος που βρίσκει τους αλγορίθμους, είναι αυτός που λύνει το πρόβλημα και όχι ο υπολογιστής. (Σγάμμας, 2006).

Έτσι γεννήθηκε ο δηλωτικός προγραμματισμός. Στο δηλωτικό ή περιγραφικό προγραμματισμό το «βάρος» εξεύρεσης λύσης μετατοπίζεται πλέον στον υπολογιστή αφού ο προγραμματιστής ορίσει επαρκώς το πρόβλημα.

3.3. Prolog: Η ιδιομορφία της και η αξία της.

Η Prolog, σύμφωνα με τα όσα ελέχθησαν στην προηγούμενη ενότητα, είναι μια δηλωτική γλώσσα προγραμματισμού που χρησιμοποιείται κυρίως σε εφαρμογές τεχνητής νοημοσύνης, όπως οι γνωσιακές βάσεις και τα έμπειρα συστήματα. Με την έννοια «δηλωτική» εννοούμε πως η Prolog καθορίζει ΤΙ πρέπει να γίνει και υπό ποιους όρους και όχι το πώς, με το οποίο καταπιάνονται οι «συμβατικές» γλώσσες προγραμματισμού. Σε αντίθεση με τις διαδικαστικές και αντικειμενοστραφείς γλώσσες, ενώ είναι φαινομενικά απλή η εκμάθησή της, λόγω του εξαιρετικά απλού συντακτικού της και τον τρόπο δημιουργίας προγραμμάτων, είναι ο τρόπος σκέψης (mindset) που καλείται ο χρήστης να αναπτύξει που αποτελεί μεγάλη πρόκληση. Και

αποτελεί μεγαλύτερη πρόκληση δε, όχι στους αδαής σε γλώσσες προγραμματισμού, όπως θα ήταν λογικό να νομίσει κάποιος, αλλά σε αυτούς που έχουν ήδη διδαχθεί κάποια διαδικαστική ή αντικειμενοστραφή γλώσσα!

Οι έμπειροι γνώστες γλωσσών προγραμματισμού έχουν ήδη μια ιδέα του τι θα πει «προγραμματισμός» και είναι εφοδιασμένοι με εργαλεία που θα τους βοηθήσουν να λύσουν «κλασσικά CS» προβλήματα (Nalera & Wonjincki).

Η κύρια ασχολία ενός προγραμματιστή Prolog είναι η καταγραφή αληθών προτάσεων (clauses) καθώς και των μεταξύ τους συσχετίσεων/σχέσεων (Brna, Pain, & Du Bulaiy, 1990). Από κει και πέρα επαφίεται στην Prolog να συμπεράνει την ορθότητα ή μη των τιθέμενων ερωτημάτων.

Από τη στιγμή που θα της πούμε ποιό είναι το πρόβλημα, αναλαμβάνει να κάνει πλήρη διερεύνηση και να βρει όλες τις δυνατές λύσεις του. Αυτό το πετυχαίνει α) με τη βοήθεια μιας ενσωματωμένης μηχανής αναζήτησης τύπου depth-first η οποία ενεργοποιείται αυτόματα κάθε φορά που ρωτάμε κάτι τη γλώσσα μέσω ερωτημάτων (queries) αλλά και β) μέσω του μηχανισμού συμπερασμού της, με τη βοήθεια του οποίου μπορεί να συμπεράνει νέα γεγονότα από ήδη υπάρχοντα. (Fischetti & Gisolfi, 1990) και (Σγάρμπας, 2006).

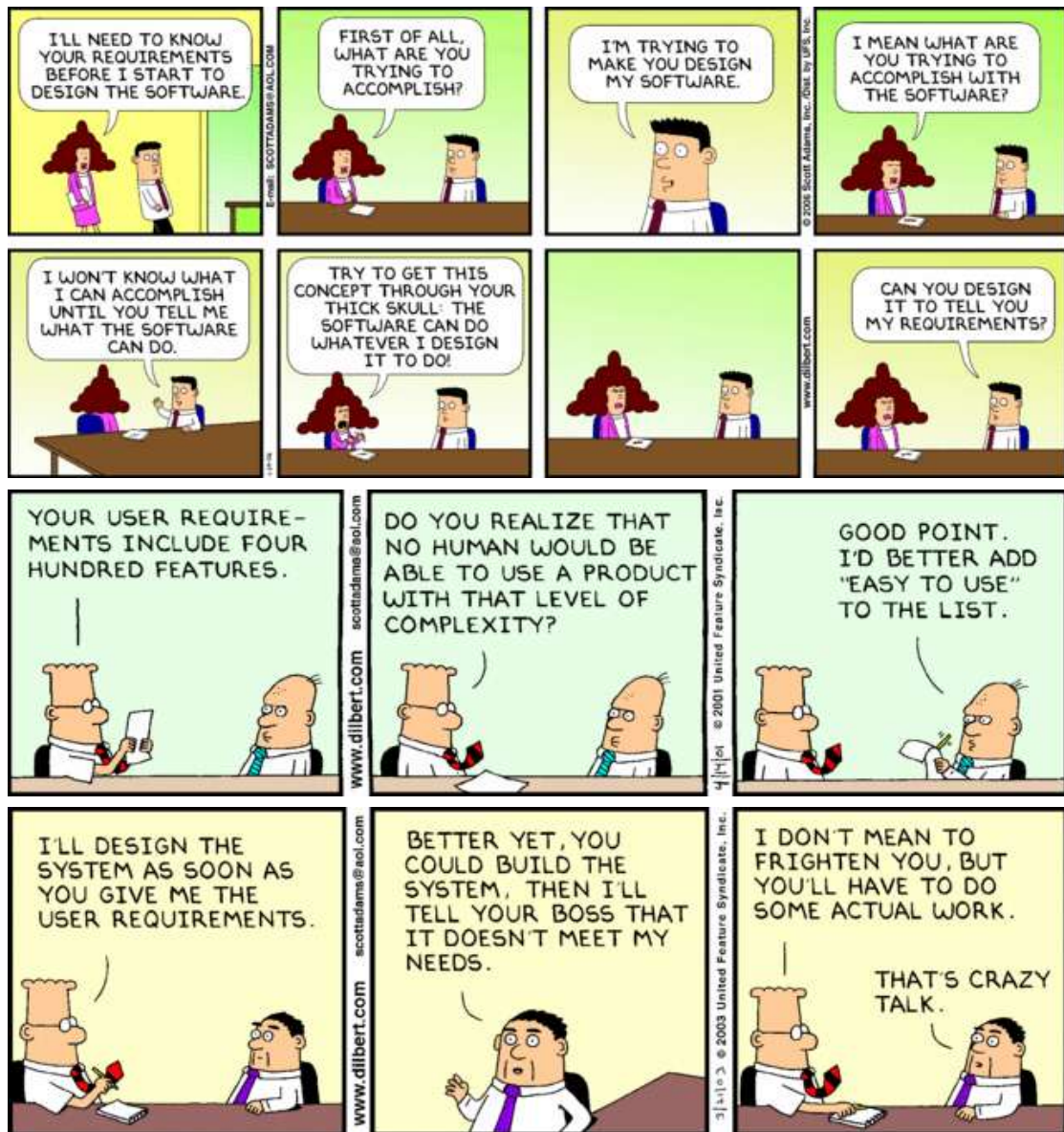
Ένα πρόγραμμα prolog αποτελείται από ένα σύνολο προτάσεων-δηλώσεων που μπορεί να είναι είτε κανόνες είτε γεγονότα τα οποία ακολουθούνται από μια τελεία που σηματοδοτεί το τέλος τους. Τα γεγονότα χρησιμοποιούνται για να περιγράψουν το πρόβλημα (πχ. Ο John είναι γονιός του Paul, ο Paul είναι γονιός του Charles) ενώ οι κανόνες ορίζουν συμπεράσματα που μπορεί να προκύψουν από ήδη γνωστές προτάσεις-δηλώσεις (πχ. Ποιος είναι ο πατέρας του Charles?) (Fischetti & Gisolfi, 1990).

Ενώ τα παραπάνω μπορούν να αποτελέσουν ανασταλτικό παράγοντα και να αποθαρρύνουν κάποιους από την εκμάθηση της, υπάρχουν, ωστόσο, σπουδαίοι λόγοι για τους οποίους πρέπει να μάθει κανείς Prolog:

- Η Prolog χρησιμοποιείται κατά κόρον σε εφαρμογές τεχνητής νοημοσύνης.
- Μέσω αυτής δίνεται η δυνατότητα έκφρασης γεγονότων και σχέσεων ασχέτως της πολυπλοκότητάς τους. Αυτή η δυνατότητα εξασφαλίζεται μέσω της quantification logic η οποία την διακατέχει. Με την έννοια αυτή εννοούμε ότι μια μεταβλητή μπορεί να έχει ιδιότητες (attributes). Έτσι μπορούμε να εκφράσουμε πλέον όχι μόνο κανόνες του τύπου: if A and B then C αλλά και πιο προχωρημένα όπως: if A(a,b,c) and B(m,n) then C(a,b,n). Για παράδειγμα, if Between(x,y,z) and Between(x,k,y) then Between(x,k,z) (Mano, 2012).
- Άλλος ένα λόγος χρήσης της Prolog είναι ο μηχανισμός ενοποίησης της. Με το που εισάγει ο χρήστης ένα ερώτημα, η Prolog θα εκκινήσει αυτόματα τη διαδικασία εύρεσης λύσης. Σύμφωνα με τον παραπάνω μηχανισμό για ν' αποφασίσει η Prolog την ορθότητα του ερωτήματός μας, ο μηχανισμός ελέγχου της Prolog προσπαθεί να ενοποιήσει (στη βιβλιογραφία υπάρχει και ο όρος ταύτιση ή ταυτοποίηση) κάθε γεγονός της πρότασης με τουλάχιστον ένα γεγονός της βάσης δεδομένων ή με τουλάχιστον έναν κανόνα. Η ενοποίηση αυτή πραγματοποιείται με σειριακό τρόπο εξετάζοντας όλα τα γεγονότα και τους κανόνες από πάνω προς τα κάτω. Για το λόγο αυτό η σειρά με την οποία εισάγουμε τα γεγονότα και τους κανόνες στη βάση δεδομένων έχει σημασία διότι καθορίζει τη σειρά με την οποία θα γίνουν οι ενοποιήσεις και άρα και τα αποτελέσματα που θα πάρουμε. (Κουράκος Μαυρομιχάλης, 2006)

4. Ανάπτυξη ευφυούς προσαρμοστικού web-based εκπαιδευτικού συστήματος για την αλληλεπιδραστική εκμάθηση της Prolog

4.1. Σύλληψη και Ανάλυση Απαιτήσεων του Συστήματος.



Η παρούσα ενότητα πραγματεύεται τη σύλληψη καθώς και την ανάλυση απαιτήσεων του προς ανάπτυξη διαδικτυακού προσαρμοστικού και αλληλεπιδραστικού λογισμικού για την εκμάθηση της Prolog.

Το Prolog Tutor, είναι ένα προσαρμοστικό εκπαιδευτικό λογισμικό που προάγει την ενεργή και στέρεη οικοδόμηση της γνώσης μέσω της αλληλεπίδρασης που αναπτύσσει με τον χρήστη. Κάνει χρήση της τεχνολογίας του διαδικτύου ευνοώντας έτσι την εξ' αποστάσεως και αυτόνομη εκπαίδευση/μάθηση. Το σύστημα αυτό απευθύνεται σε προπτυχιακούς και μεταπτυχιακούς φοιτητές πληροφορικής χωρίς ωστόσο να αποκλείεται η χρήση και από φοιτητές άλλων ειδικοτήτων καθώς το σύστημα δεν απαιτεί από το χρήστη πρότερες γνώσεις επί του αντικείμενου αλλά ούτε κάποια γλώσσα προγραμματισμού! Το εκπαιδευτικό αυτό σύστημα διακρίνεται από προσαρμοστικότητα καθώς και αλληλεπιδραστικότητα. Προσαρμοστικότητα γιατί ο κάθε χρήστης που συνδέεται στην πλατφόρμα εκμάθησης Prolog είναι μια ξεχωριστή οντότητα με ή χωρίς πρότερες γνώσεις επί της Prolog ή/και άλλων γλωσσών προγραμματισμού και διαφορετικό υπόβαθρο πάνω στο αντικείμενο του της πληροφορικής γενικότερα.

Για να μπορέσει ο χρήστης λοιπόν, να επιτύχει το στόχο του, που δεν είναι άλλος από την εκμάθηση της εν λόγω γλώσσας, θα πρέπει να διαθέτει έναν οδηγό ο οποίος θα τον καθοδηγεί σε όλη αυτή τη διαδικασία μειώνοντάς του στο προς εξερεύνηση πεδίο (browsing space), προτείνοντάς του τι συνδέσμους να ακολουθήσει ή σε ποιον να επιστήσει την προσοχή του, ή τέλος να του παρέχει προσαρμοστικά σχόλια σε ήδη ενεργούς συνδέσμους. Το ρόλο αυτό έχει η προσαρμοστικότητα.

Επειδή είναι αρκετά σημαντική και η εκμάθηση με ενεργό τρόπο από το χρήστη, δίνεται η δυνατότητα από το σύστημα, μέσω της αλληλεπιδραστικότητας, να του απαντά σε κάθε ερώτηση που του τίθεται καθώς και να εφαρμόζει online τα παραδείγματα σε αντίθεση με την πλειονότητα των διαδικτυακών πολυμεσικών συστημάτων για την εκμάθηση της Prolog που παραθέτουν μόνο θεωρία και πολυμεσικό υλικό για την παρουσίαση της καθώς επίσης και παραδείγματα.

Το πεδίο γνώσης (domain knowledge) του συστήματός μας είναι η εισαγωγή σε αρχικές έως και αρκετά προχωρημένες έννοιες της Prolog.

Πιο συγκεκριμένα έχει ακολουθηθεί η εξής ομαδοποίηση εννοιών της Prolog σε 4 επίπεδα δυσκολίας. Το επίπεδο του αρχάριου αντιστοιχεί σε βασικές έννοιες όπως αυτές των εννοιών facts, rules, queries, μπορεί να συντάσσει γνωσιακές βάσεις και κατέχει σε βασικό επίπεδο την έννοια της αναδρομής. Ο καλός γνώστης της γλώσσας επιπρόσθετα γνωρίζει βασικές αλλά προχωρημένες χρήσεις των λιστών. Ο πολύ καλός χρήστης γνωρίζει πως μπορεί να εκφράσει διαδικαστικούς αλγόριθμους στην Prolog αλλά έχει και γνώση ειδικών τελεστών όπως αυτών της αποκοπής. Τέλος ο άριστος χρήστης είναι σε θέση να χειριστεί προηγμένες εντολές που έχουν να κάνουν με τη δυναμική τροποποίηση γνωσιακών βάσεων δεδομένων καθώς και τον χειρισμό αρχείων.

Για την ουσιαστική εμπέδωση της ύλης από τους φοιτητές όλων των επιπέδων κάθε κεφάλαιο της ύλης συνοδεύεται από θεωρία αλλά και ασκήσεις εμπέδωσης ποικίλης δυσκολίας, όπου ο χρήστης μπορεί να «τρέχει» εκείνη τη στιγμή τα ερωτήματα που έχει μάθει και να βλέπει τα αποτελέσματά τους.

Στο τέλος κάθε κεφαλαίου υπάρχει ένα υποχρεωτικό επαναληπτικό τεστ πλήρως προσαρμοσμένο στο μοντέλο χρήστη στο οποίο ανήκει ο χρήστης κάθε φορά και το οποίο εξετάζει το επίπεδο γνώσης του και η επιτυχία του καθορίζει τη μετάβασή του στο επόμενο προσαρμοσμένο κεφάλαιο. Αν το επίπεδο του χρήστη είναι πάνω ή κάτω από ένα προκαθορισμένο κατώφλι (το οποίο είναι αποτέλεσμα αρκετών παραγόντων όπως θα αναλυθεί

στη συνέχεια) τότε ο χρήστης μεταβαίνει ή μπορεί ακόμη και να παραμένει σε ένα (νέο) επίπεδο γνώσης.

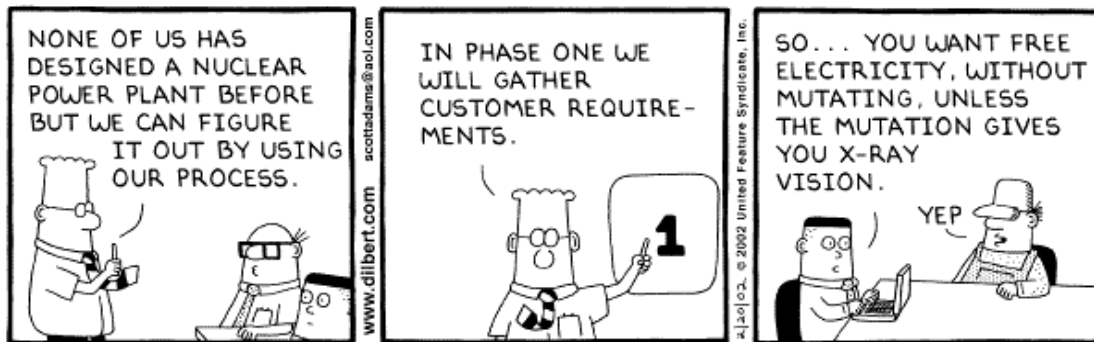
Ο βαθμός γνώσης των εννοιών καθορίζεται από τα εξής:

- ✓ Τον αριθμό των σωστών απαντήσεων που αφορούν την εν λόγω έννοια (πχ. Αναδρομή, χειρισμός αρχείων κλπ.),
- ✓ Ο χρόνος ολοκλήρωσης του τεστ.

Τα διακριτά επίπεδα γνώσης των εννοιών του κεφαλαίου είναι:

- ✓ Μέτρια γνώση
- ✓ Καλή γνώση
- ✓ Πολύ καλή γνώση
- ✓ Άριστη γνώση

Για την αρχικοποίηση του μοντέλου θεωρούμε πως κάθε χρήστης κατά την πρώτη επαφή του με το σύστημα δεν γνωρίζει καθόλου Prolog και ότι υπάγεται στο μοντέλο του μέτριου.



Copyright © 2002 United Feature Syndicate, Inc.

4.1.1. Γενική Θεώρηση Prolog Tutor.

Το Prolog Tutor είναι ένα ευφυές, προσαρμοστικό και αλληλεπιδραστικό εκπαιδευτικό λογισμικό που έχει ως γνωστικό του αντικείμενο τη γλώσσα προγραμματισμού Prolog.

Απευθύνεται σε φοιτητές σχολών Πληροφορικής, χωρίς όμως να αποκλείονται οι φοιτητές άλλων σχολών αλλά και χρήστες με άλλη ιδιότητα πέραν της φοιτητικής. Το σύστημα διακρίνεται από προσαρμοστικότητα και αλληλεπιδραστικότητα παρέχοντας έτσι ελευθερία πλοήγησης αλλά και προσαρμογή του παρεχόμενου εκπαιδευτικού λογισμικού στις ιδιαίτερες ανάγκες του κάθε χρήστη χωρίς να του παρουσιάζει ένα one-size-fits-all περιεχόμενο.

Για την εξάσκηση των χρηστών υπάρχει μια πληθώρα παραδειγμάτων εμπέδωσης της ύλης τα οποία είναι απόλυτα προσαρμοσμένα στο μοντέλο χρήστη το οποίο υπάγεται.

Το γνωστικό πεδίο αναπαρίστανται ως ένα εννοιολογικό δίκτυο, το οποίο απεικονίζει σχέσεις μεταξύ του συνόλου των εννοιών του αντικειμένου που διδάσκεται.

Όλες οι παραπάνω δυνατότητες παρέχονται κατόπιν log-in του χρήστη στο σύστημα.

4.1.2. Λειτουργικές απαιτήσεις Prolog Tutor.

Επειδή έχουμε 1 ομάδα χρηστών της διαδικτυακής εφαρμογής έχουμε 1 ομάδα λειτουργικών απαιτήσεων.

Λειτουργικές απαιτήσεις για την ομάδα χρηστών ΦΟΙΤΗΤΗΣ

- Λειτουργική Απαιτήση 1.** **Ανάγνωση Ύλης και Εφαρμογή Παραδειγμάτων.**
Μέσω της εφαρμογής ο φοιτητής μπορεί να περιηγηθεί στη διδασκόμενη ύλη. Με την επιλογή του κεφαλαίου παρουσιάζεται η θεωρία καθώς και τα παραδείγματα τα οποία μπορεί να κάνει ο χρήστης μέσω της φόρμας ερωτημάτων που βρίσκεται στο κάτω μέρος της οθόνης του.
- Λειτουργική Απαιτήση 2.** **Εκπλήρωση τεστ.**
Δίνεται στο χρήστη η δυνατότητα να πραγματοποιεί τεστ μετά την ολοκλήρωση του κάθε κεφαλαίου.
- Λειτουργική Απαιτήση 3.** **Εμφάνιση Επίδοσης.**
Ο χρήστης έχει τη δυνατότητα να βλέπει την πρόοδό του καθ' όλη τη διάρκεια παραμονής του στην εφαρμογή. Εδώ παρουσιάζονται στο χρήστη τα τεστ τα οποία έχει εκπληρώσει καθώς και το σκορ που έχει επιτύχει σε αυτά
- Λειτουργική Απαιτήση 4.** **Αποσύνδεση χρήστη.**
Η εφαρμογή δίνει τη δυνατότητα στο χρήστη να αποσυνδεθεί ανά πάσα ώρα και στιγμή, ανεξάρτητα από το που βρίσκεται.
- Λειτουργική Απαιτήση 5.** **Επιστροφή στην κυρίως εφαρμογή.**
Δίνεται η δυνατότητα στο χρήστη να επιστρέφει στην κυρίως εφαρμογή από όποιο σημείο και αν βρίσκεται.

4.1.3. Μη λειτουργικές απαιτήσεις του Prolog Tutor.

Απαιτήσεις χρήσης

- Η πρόσβαση στο σύστημα γίνεται μόνο από εξουσιοδοτημένους χρήστες.
- Η εισαγωγή των δεδομένων θα γίνεται από το ποντίκι και το πληκτρολόγιο.

Απαιτήσεις επιδόσεων

- Ο χρόνος αναζήτησης και ανάκτησης από τη βάση δεδομένων μιας εγγραφής με κλειδί τον κωδικό δε θα πρέπει να είναι πολύ μεγάλος.

Απαιτήσεις υλοποίησης

- Θα γίνει χρήση της γλώσσας σήμανσης υπερκειμένων HTML, της PHP, Javascript και CSS για την υλοποίηση της διαδικτυακής εφαρμογής με τη βοήθεια του εργαλείου Dreamweaver και Apache Server.

Απαιτήσεις Βάσεων Δεδομένων

- Η εφαρμογή θα πρέπει να διατηρεί αρχείο φοιτητών, καθώς και των μαθημάτων, των τεστ και των αποτελεσμάτων.

Φυσικές Απαιτήσεις

- Η εφαρμογή θα πρέπει να είναι συμβατή με όλους τους γνωστούς φυλλομετρητές, όπως Opera, Mozilla, Explorer, Google Chrome.

4.1.4. Διαγράμματα περιπτώσεων χρήσης (use case diagrams).

Το μοντέλο των περιπτώσεων χρήσης

Η σπουδαιότητα του μοντέλου περιπτώσεων χρήσης του συστήματος έγκειται στην περιγραφή του υποκειμένου για τον οποίο αναπτύσσεται το σύστημα καθώς και των λειτουργιών που πρέπει αυτό να επιτελεί.

Η ικανοποίηση των λειτουργικών απαιτήσεων του χρήστη από μια εφαρμογή λογισμικού υλοποιείται ως μια αλληλουχία ενεργειών που εκτελούνται από το λογισμικό αλληλεπιδρώντας είτε με κάποιο χρήστη είτε με άλλα συστήματα.

Με βάση τα όσα ειπώθηκαν παραπάνω θα μπορούσαμε να πούμε ότι:

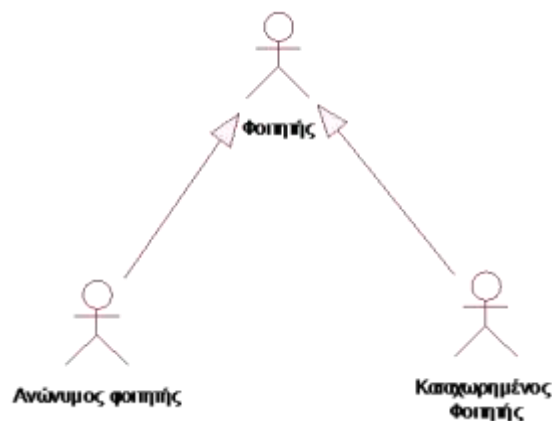
Μια περίπτωση χρήσης (use case) είναι μια αλληλουχία ενεργειών που εκτελεί το λογισμικό αλληλεπιδρώντας με το χρήστη ή με εξωτερικά συστήματα, προκειμένου να ικανοποιήσει μια λειτουργική απαίτηση (Βεσκούκης , 2001).

Προσδιορισμός των ενεργοποιών

Τονίζοντας προηγουμένως, την σπουδαιότητα του μοντέλου περιπτώσεων χρήσης του συστήματος έγινε αναφορά και στο υποκείμενο, η ανάγκη του οποίου έθεσε σε κίνηση τους μηχανισμούς για την κατασκευή του λογισμικού που θα πραγματοποιεί επιθυμητές για αυτόν λειτουργίες. Το υποκείμενο-χρήστης του συστήματος, καθώς και τα όποια εξωτερικά συστήματα με τα οποία αλληλεπιδρά το σύστημα ονομάζονται ενεργοποιοί (actors) του συστήματος.

Στην εργασία μας, έχουμε εντοπίσει τους εξής ενεργοποιοί:

- i. Ο φοιτητής.
 - a. Ανώνυμος φοιτητής
 - b. Καταχωρημένος φοιτητής



Σχήμα 1 Οι ενεργοποιοί του συστήματος

4.2. Σχεδιασμός του συστήματος.

Στην ενότητα αυτή αναλύεται ο σχεδιασμός του Prolog Tutor βάσει των απαιτήσεων που προηγήθηκαν στην προηγούμενη ενότητα.

Οι φοιτητές θα έχουν τη δική τους διεπαφή μέσω της οποίας θα αλληλεπιδρούν με το σύστημα. Αναγκαία προϋπόθεση για την αλληλεπίδραση χρήστη – συστήματος είναι η σύνδεσή τους (log in) σε αυτό. Η διεπαφή του φοιτητή, η οποία και αποτελεί το βασικό κομμάτι του συστήματός μας, θα του δίνει τη δυνατότητα να μελετά τη παρεχόμενη θεωρία, να εφαρμόζει τα παραδείγματα, να εισάγει και να εκτελεί ερωτήματα, να λύνει τεστ και προσαρμογής στο επίπεδο του χρήστη. Επίσης δίνεται η δυνατότητα τροποποιεί το προφίλ και να αποσυνδέεται ανά πάσα ώρα και στιγμή, σε όποιο σημείο του συστήματος και αν βρίσκεται.

4.2.1 Βάση Δεδομένων.

Η βάση δεδομένων του συστήματος διατηρεί όλα εκείνα τα απαραίτητα στοιχεία των χρηστών του συστήματος, όπως είναι τα στοιχεία του λογαριασμού σύνδεσής τους, το προφίλ τους καθώς και την αλληλεπίδρασή τους με το σύστημα (τι τεστ έλυσε, την πρόδοό του κλπ).

Οι κύριες οντότητες της εκπαιδευτικής εφαρμογής μας είναι ο φοιτητής, όπου έχει ως χαρακτηριστικά του έναν κωδικό που τον προσδιορίζει μοναδικά, το ονοματεπώνυμό του, την ημερομηνία εγγραφής του στο σύστημα καθώς και ένα όνομα και έναν κωδικό χρήστη, το κεφάλαιο, με κύρια χαρακτηριστικά του ένα αναγνωριστικό που το χαρακτηρίζει μοναδικά, τον τίτλο, το επίπεδο δυσκολίας του, το υποκεφάλαιο που συνδέεται με την οντότητα κεφάλαιο με τη σχέση έχει. Η μεταξύ του σχέση είναι 1 προς πολλά καθώς ένα κεφάλαιο μπορεί να έχει πολλά υποκεφάλαια. Για το υποκεφάλαιο οι πληροφορίες που κρατούνται στη βάση δεδομένων μας είναι ένα μοναδικό αναγνωριστικό, ο τίτλος του, το κεφάλαιο στο οποίο ανήκει (ξένο κλειδί), το κυρίως περιεχόμενο, το κυρίως σετ παραδειγμάτων καθώς και μερικά έξτρα παραδείγματα και περιεχόμενα τα οποία θα εμφανίζονται στο χρήστη βάσει του μοντέλου στο οποίο υπάγεται.

Επίσης υπάρχουν οι οντότητες αποτελέσματα και τεστ. Η οντότητα αποτέλεσμα έχει ως χαρακτηριστικά της ένα μοναδικό αναγνωριστικό, το αναγνωριστικό του χρήστη που επιλύει μια άσκηση, το κεφάλαιο στο οποίο ανήκει η άσκηση, η βαθμολογία, ημερομηνία εξέτασης καθώς και ο χρόνος που χρειάστηκε από τη μεριά του φοιτητή για την ολοκλήρωση του τεστ.

Τέλος, η οντότητα τεστ περιέχει ένα μοναδικό αναγνωριστικό, καθώς και το αναγνωριστικό του κεφαλαίου στο οποίο ανήκει (ξένο κλειδί).

4.2.2 Μοντελοποίηση χρήστη και γεννήτορας ασκήσεων.

Το Prolog Tutor είναι ένα προσαρμοστικό και αλληλεπιδραστικό web based λογισμικό το οποίο δίνει τη δυνατότητα στους χρήστες του να μάθουν την εν λόγω γλώσσα προγραμματισμού χωρίς όμως να είναι αναγκαία προϋπόθεση η πρότερη γνώση της.

Το σύστημα ως προσωπικός καθηγητής του κάθε χρήστη παρακολουθεί την επίδοσή του και τον κατατάσσει σε κάποιο από τα 4 μοντέλα χρηστών. Τα τέσσερα μοντέλα χρηστών που μπορεί να υπαχθεί ο χρήστης τα εξής:

- ✓ Μέτριος, με βαθμολογία κάτω από 50%
- ✓ Καλός, με βαθμολογία μεταξύ 50% και 70%
- ✓ Πολύ καλός, με βαθμολογία μεταξύ 70% και 80% και τέλος
- ✓ Άριστος, με βαθμολογία μεταξύ 80% και 100%.

Πέρα από την επίδοσή των χρηστών στο τεστ λαμβάνεται υπόψη και ο χρόνος ολοκλήρωσής του καθώς θεωρούμε πως η επίδοση είναι άρρηκτα συνδεδεμένη με το χρόνο. Έτσι λοιπόν, ένας χρήστης που έχει ολοκληρώσει το τεστ με άριστο βαθμό, δηλαδή πάνω από 80%, σε πάρα πολύ μεγάλο χρονικό διάστημα δεν μπορεί να θεωρηθεί εξίσου άριστος με κάποιον που ενώ έχει επιτύχει άριστη βαθμολογία έχει ολοκληρώσει το τεστ σε φυσιολογικό διάστημα χρόνου.

Βαθμός Χρόνος	50%>βαθμολογία	50%<βαθμολογία <70%	70%<βαθμολογία <80%	80%<βαθμολογία< 100%
50%>χρόνος	Μέτριος, βιάστηκε πάρα πολύ	Καλός, βιάστηκε πάρα πολύ	Πολύ καλός, βιάστηκε πάρα πολύ	Άριστος, βιάστηκε πάρα πολύ
50%<χρόνος<70%	Μέτριος, βιάστηκε	Καλός, βιάστηκε	Πολύ καλός, βιάστηκε	Άριστος, βιάστηκε
70%<χρόνος<80%	Μέτριος	Καλός	Πολύ καλός	Άριστος
70%<χρόνος<100%	Μέτριος, άργησε	Καλός, άργησε	Πολύ καλός, άργησε	Άριστος, άργησε

Το σύστημα έχει τη δυνατότητα να προσαρμόζει τη θεωρία αλλά και τις ασκήσεις που παρουσιάζονται σύμφωνα με το μοντέλο που υπάγεται ο χρήστης. Έτσι λοιπόν ένας μέτριος χρήστης θα έχει πρόσβαση σε θεωρία και παραδείγματα απόλυτα προσαρμοσμένα στο επίπεδό του χωρίς να πασχίζει να κατανοήσει θεωρίες που δεν του είναι εύκολο να επεξεργαστεί αλλά και να εξασκείται με περισσότερα παραδείγματα από ότι κάποιος άλλος χρήστης που ανήκει στο μοντέλο του άριστου.

Έτσι λοιπόν, με βάση το μοντέλο στο οποίο εντάσσεται ο χρήστης η κυρίως θεωρία και τα κυρίως παραδείγματα συνοδεύονται και από ένα ακόμη (extra) κομμάτι θεωρίας αλλά και παραδειγμάτων προσαρμοσμένα στο επίπεδό του. Το νέο κομμάτι εισάγεται με ειδική μορφοποίηση (με έντονη ή πλάγια γραφή) ώστε μπορεί να ξεχωρίζει από το κυρίως κείμενο.

Η προσαρμογή που προσφέρει το σύστημά μας δεν μένει μόνο στην παρουσίαση της θεωρίας, των παραδειγμάτων και των συμβουλών αλλά προχωρά και στη δόμηση του τεστ αξιολόγησης κάθε κεφαλαίου.

Με βάση λοιπόν το μοντέλο που υπάγεται ο χρήστης ο γεννήτορας ασκήσεων δομεί ένα τεστ αξιολόγησης απόλυτα προσαρμοσμένο σε αυτό. Κάθε τεστ αξιολόγησης αποτελείται από ένα σετ 10 ερωτήσεων τις οποίες καλείται να απαντήσει ο χρήστης. Ο προτεινόμενος χρόνος υλοποίησης του τεστ έχει οριστεί στα 5 λεπτά.

Με βάση τα παραπάνω για κάθε μοντέλο χρήστη έχουμε προσδιορίσει τα εξής:

Μοντέλο Χρήστη	Μέτριος	Καλός	Πολύ καλός	Άριστος
Αριθμός ερωτήσεων				
Αριθμός ερωτήσεων επιπέδου 1	6	4	2	1

Αριθμός ερωτήσεων επιπέδου 2	3	4	5	5
Αριθμός ερωτήσεων επιπέδου 3	1	2	3	4

Όσο καλύτερος είναι ο χρήστης τόσο μειώνεται ο αριθμός των εύκολων ερωτήσεων και αυξάνεται ο αριθμός των δύσκολων ερωτήσεων.

Κάθε ερώτηση επιπέδου 1 βαθμολογείται με 10 μονάδες, κάθε ερώτηση επιπέδου 2 βαθμολογείται με 20 μονάδες και τέλος κάθε ερώτηση επιπέδου 3 βαθμολογείται με 30 μονάδες.

Έτσι λοιπόν σύμφωνα με τα παραπάνω αλλά και τον πίνακα που προηγήθηκε το άριστο σκορ για κάθε επίπεδο είναι 150 για το μοντέλο χρήστη μέτριος, 180 για το μοντέλο χρήστη καλός, 210 για το μοντέλο χρήστη πολύ καλός και τέλος 230 για το μοντέλο χρήστη άριστος.

Η τελική βαθμολογία προκύπτει από τον τύπο: $final_{score} = \frac{score\ που\ πέτυχε\ στο\ τεστ}{\acute{\alpha}ριστη\ βαθμολογία\ \acute{\alpha}σει\ μοντέλου} * 100.$

4.3. Υλοποίηση του συστήματος διδασκαλίας Prolog, Prolog Tutor

Στην προηγούμενη ενότητα παρουσιάστηκε ο σχεδιασμός του συστήματος διδασκαλίας Prolog, Prolog Tutor. Τώρα θα μεταβούμε στο στάδιο της υλοποίησης.

Για τη δημιουργία των ιστοσελίδων που θα απαρτίζουν το σύστημά μας θα γίνει εκτενής χρήση της γλώσσας σήμανσης υπερκειμένων HTML καθώς και PHP έκδοσης 5.3.5. Ο web-server που θα χρησιμοποιηθεί θα είναι ο Apache Server έκδοσης 2.2.17.

Για τη δημιουργία της Βάσης Δεδομένων του συστήματός μας θα κάνουμε χρήση MySQL έκδοσης 5.5.8.

4.3.1. Βάση Δεδομένων.

Η Βάση Δεδομένων αποτελεί ακρογωνιαίο λίθο της εφαρμογής μας καθώς σε αυτή διατηρούνται όλα εκείνα τα στοιχεία που καθιστούν εύρυθμη και ομαλή τη λειτουργία του. Μεταφράζοντας το διάγραμμα οντοτήτων – συσχετίσεων έχουμε τους κάτωθι πίνακες:

chapter		
Field	Type	Null
<u>chapterId</u>	int(10)	No
Title	varchar(100)	No
Level	varchar(50)	No

Εικόνα 1 Πίνακας όπου περιέχονται τα κεφάλαια

questions	
Field	Type
<u>question_id</u>	int(11)
question	text
right_answer	text
answer1	text
answer2	text
answer3	text
level	int(11)
testId	int(11)

Εικόνα 2 Πίνακας όπου περιέχονται οι ερωτήσεις του τεστ αξιολόγησης

results	
Field	Type
<u>id</u>	int(11)
userId	int(11)
chapterTest	varchar(100)
score	int(11)
datetime_test	datetime
succeded	enum('YES', 'NO')
completedtest	enum('YES', 'NO')
totalminutes	int(11)
totalseconds	int(11)

Εικόνα 3 Πίνακας όπου αποθηκεύονται τα αποτελέσματα του τεστ

student	
Field	Type
<u>studentId</u>	int(10)
name	varchar(50)
surname	varchar(50)
userName	varchar(20)
passWord	varchar(10)
email	varchar(60)
registrationDate	date

Εικόνα 4 Πίνακας όπου αποθηκεύονται οι φοιτητές

sub_chapters	
Field	Type
<u>sub_chapterId</u>	int(10)
title	varchar(100)
description	text
mainContent	text
extraContent1	text
extraContent2	text
mainExample	text
extraExample1	text
extraExample2	text
chapterId	int(10)
link	varchar(100)

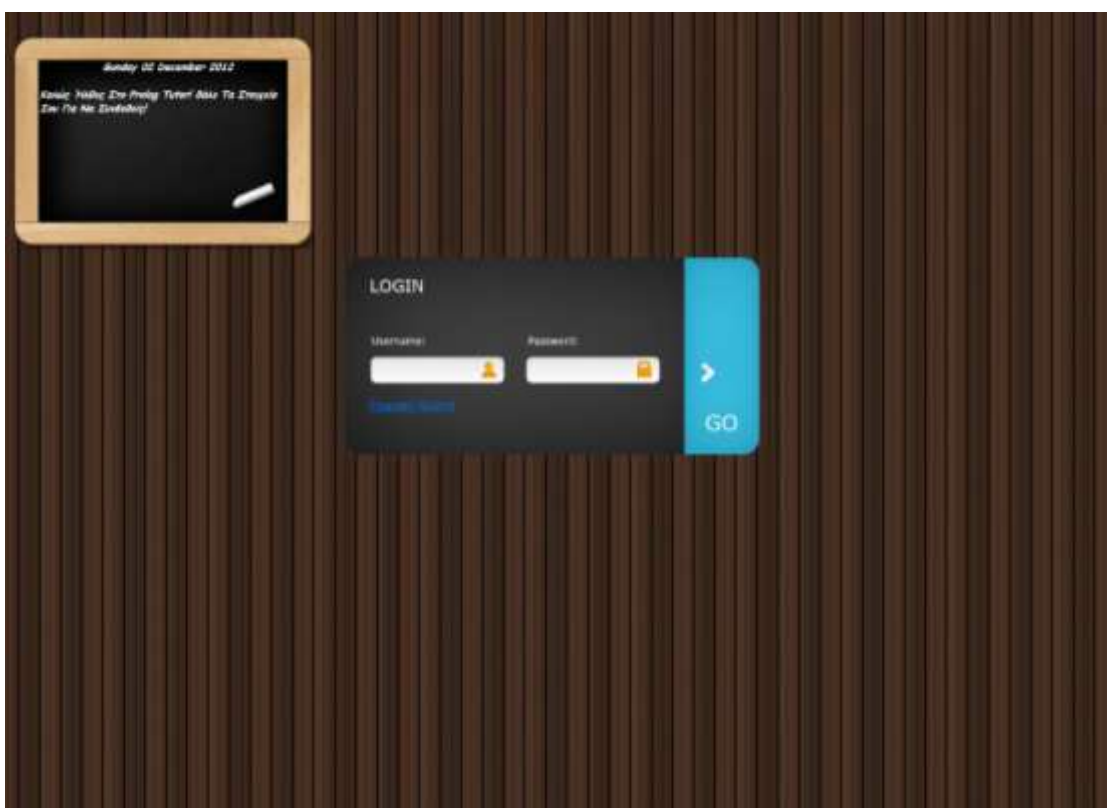
Εικόνα 5 Πίνακας όπου αποθηκεύονται τα υποκεφάλαια

test	
Field	Type
<u>id</u>	int(11)
chapterid	int(11)

Εικόνα 6 Πίνακας όπου αποθηκεύονται πληροφορίες για τα τεστ

4.3.2. Παρουσίαση του συστήματος διδασκαλίας Prolog, Prolog Tutor.

Σε αυτή την ενότητα παρουσιάζονται στιγμιότυπα χρήσης του συστήματος Prolog Tutor.



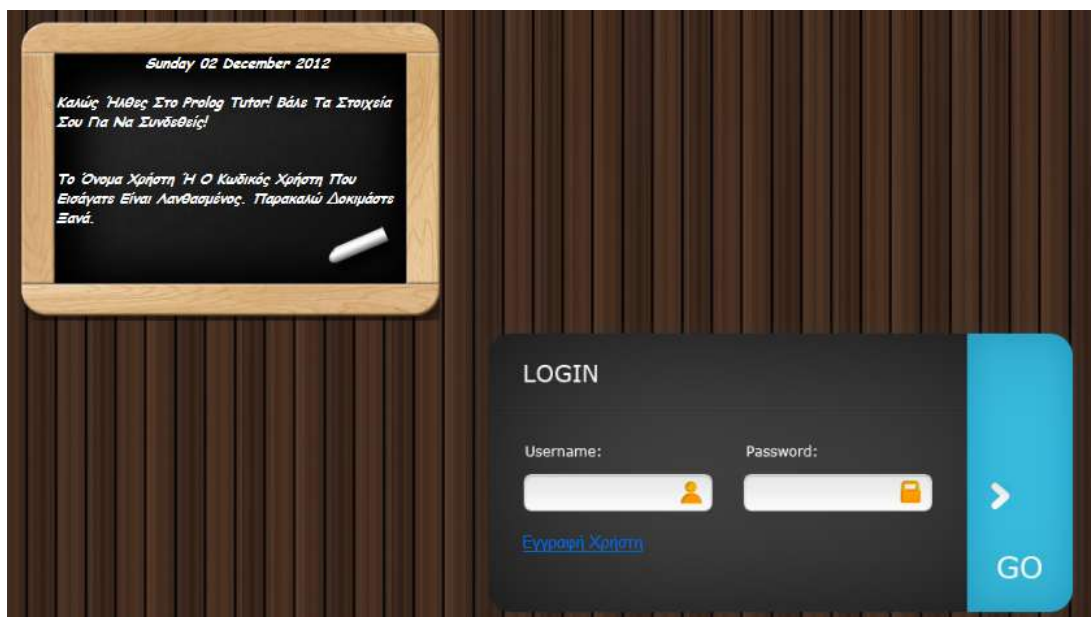
Εικόνα 7 Η αρχική σελίδα του Prolog Tutor

Το Prolog Tutor είναι ένα σύστημα που επιτρέπει την πρόσβαση μόνο σε εξουσιοδοτημένου χρήστες. Για αυτό το λόγο κατά την είσοδο του χρήστη στο σύστημα, η πρώτη σελίδα που εμφανίζεται είναι η σελίδα σύνδεσης χρήστη. Σε αυτή τη σελίδα ο χρήστης μπορεί να δώσει τα στοιχεία σύνδεσής του εφόσον είναι ήδη εγγεγραμμένος ή μπορεί να δημιουργήσει ένα νέο λογαριασμό.

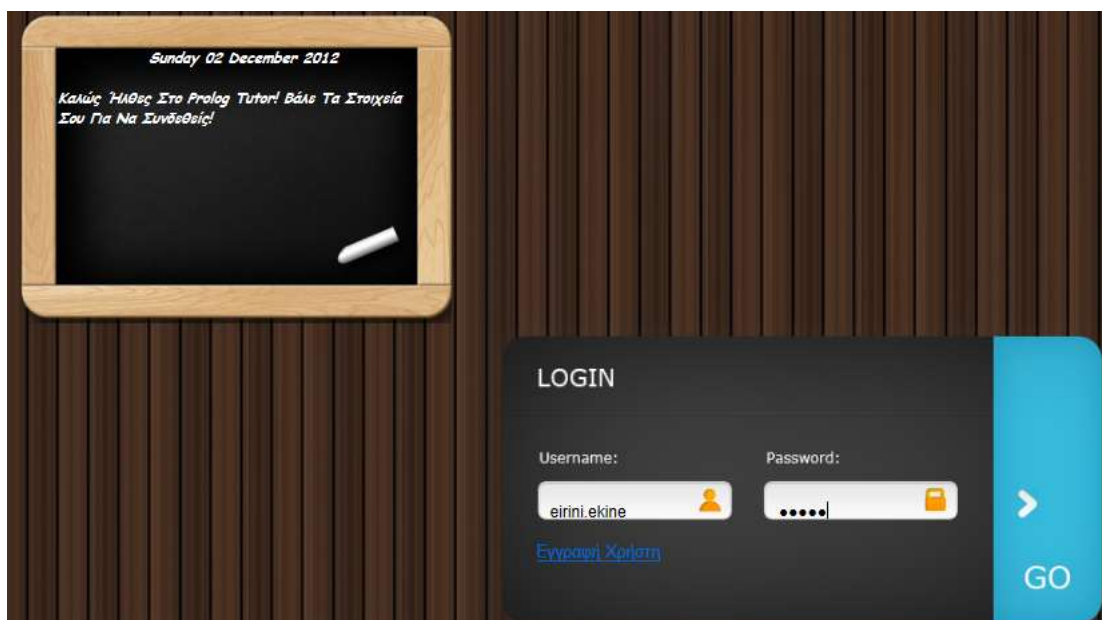
Μεταπτυχιακή Διατριβή

Αφού ο χρήστης υποβάλλει τα στοιχεία σύνδεσής του, το σύστημα πραγματοποιεί έλεγχο προκειμένου να ταυτοποιηθεί ο χρήστης με τα στοιχεία των χρηστών που είναι αποθηκευμένα στη βάση δεδομένων του συστήματος. Σε περίπτωση που εισάγει λανθασμένα στοιχεία, είτε όνομα χρήστη είτε κωδικό είτε και τα δύο, το σύστημα τον ενημερώνει σχετικά και τον προτρέπει να εισάγει εκ νέου τα στοιχεία του.

Σε περίπτωση ο χρήστης είναι καινούριος έχει τη δυνατότητα να πραγματοποιήσει εγγραφή.



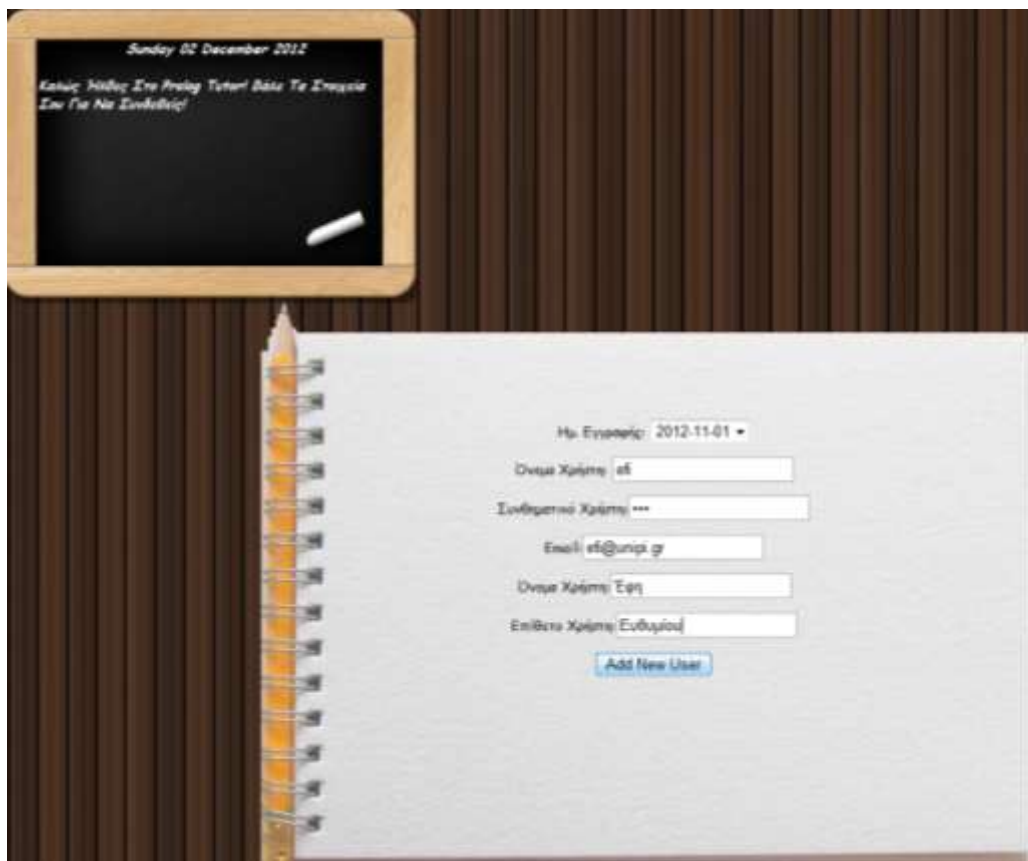
Εικόνα 8 Εισαγωγή λανθασμένων ή κενών στοιχείων στη φόρμα σύνδεσης



Εικόνα 9 Εισαγωγή στο σύστημα ως eirini.ekine



Εικόνα 10 Φόρμα εγγραφής νέου χρήστη στο σύστημα



Εικόνα 11 Εγγραφή νέου χρήστη στο σύστημα

Αφού συνδεθεί επιτυχώς, έχει πρόσβαση στην κεντρική οθόνη του συστήματος.



Εικόνα 12 Κεντρική οθόνη του Prolog Tutor

Η διεπαφή του Prolog tutor

Όπως ήδη ελέχθη, αφού ο χρήστης συνδεθεί επιτυχώς παρουσιάζεται σε αυτόν η κεντρική σελίδα της εφαρμογής. Ο χρήστης μπορεί να πλοηγηθεί μέσω του μενού γρήγορης πρόσβασης που παρέχεται σε αυτόν στην πάνω δεξιά μεριά του συστήματος αλλά και του μενού που δίνεται στα δεξιά για να πλοηγηθεί στην παρεχόμενη ύλη καθώς και τα παραδείγματα εμπέδωσης.

Το μενού γρήγορης πρόσβασης δίνει στο χρήστη τις εξής δυνατότητες μέσω των επιλογών του:

1. Η ταυτότητα. Το κουμπί του με την ταυτότητα δίνει τη δυνατότητα στο χρήστη να δει τα στοιχεία που αφορούν το προφίλ του και να πραγματοποιήσει αλλαγές επί αυτών. Τα στοιχεία που εμφανίζονται είναι τα εξής: το ονοματεπώνυμό του, η ημερομηνία εγγραφής του στο σύστημα, ο κωδικός πρόσβασης καθώς και το όνομα χρήστη.



Εικόνα 13 Επεξεργασία στοιχείων χρήστη

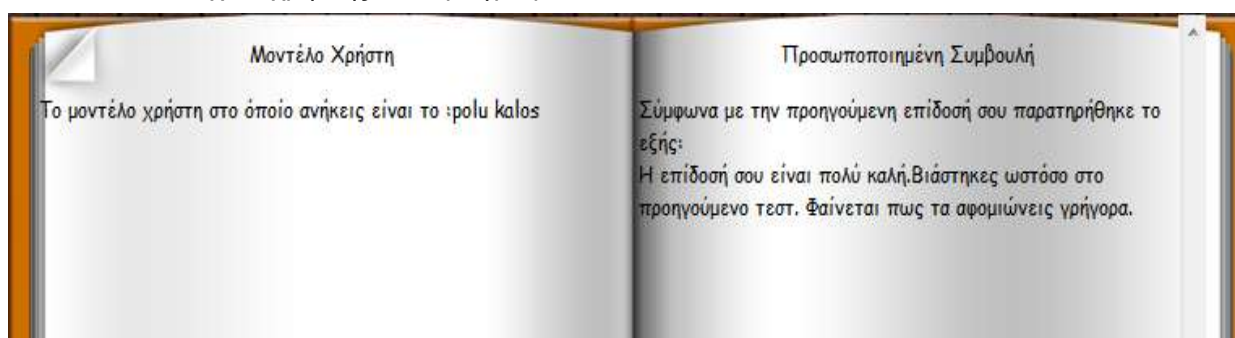
Εδώ δίνεται η δυνατότητα επεξεργασίας των στοιχείων του χρήστη. Αρχικά, όλα τα πεδία είναι απενεργοποιημένα προκειμένου να αποφευχθεί η από λάθος αλλαγή των στοιχείων. Με το κουμπί edit user info ενεργοποιούνται όλα τα πεδία της φόρμας και αφού πραγματοποιηθούν όλες οι επιθυμητές αλλαγές ο χρήστης μπορεί να τις αποθηκεύσει στη βάση με το κουμπί commit changes.

2. Το σημειωματάριο. Η επιλογή αυτή δίνει τη δυνατότητα στο χρήστη να διεξάγει προσωποποιημένα τεστ ανάλογα με το μοντέλο χρήστη στο οποίο ανήκει. Εδώ εμφανίζεται το τεστ που δεν έχει ακόμη ο χρήστης κάνει.



Εικόνα 14 Η σελίδα του τεστ αξιολόγησης

3. Η ατζέντα. Η ατζέντα δίνει τη δυνατότητα στο χρήστη να λαμβάνει πληροφορίες για το μοντέλο χρήστη στο οποίο ανήκει καθώς και να λαμβάνει μια προσωποποιημένη συμβουλή. Τα προαναφερθέντα προκύπτουν από το την βαθμολογία που έχει επιτύχει ο χρήστης στο προηγούμενο τεστ.



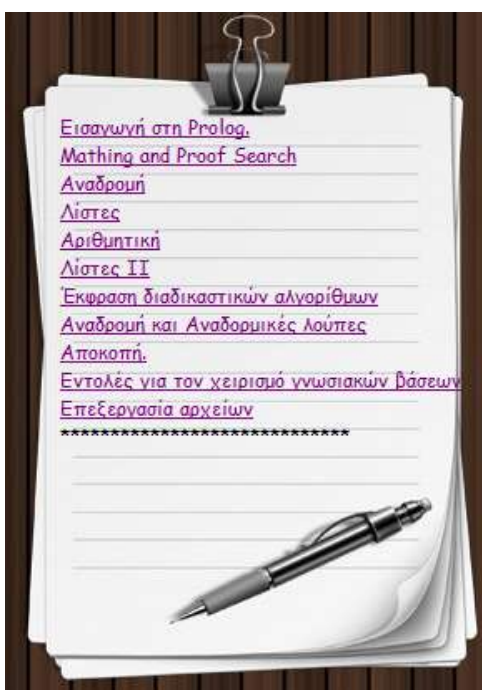
Εικόνα 15 Μοντέλο χρήστη στο οποίο ανήκει ο συγκεκριμένος χρήστης και προσωποποιημένη συμβουλή

4. Το σπίτι. Δίνεται η δυνατότητα να επιστρέφει ο χρήστης στην κεντρική σελίδα της εφαρμογής.
5. Exit. Δίνει τη δυνατότητα εξόδου από το σύστημα.
Με την επιλογή αυτή αποσυνδέεται ο χρήστης από το σύστημα και παραπέμπεται στην αρχική σελίδα προκειμένου να εισάγει τα στοιχεία του.

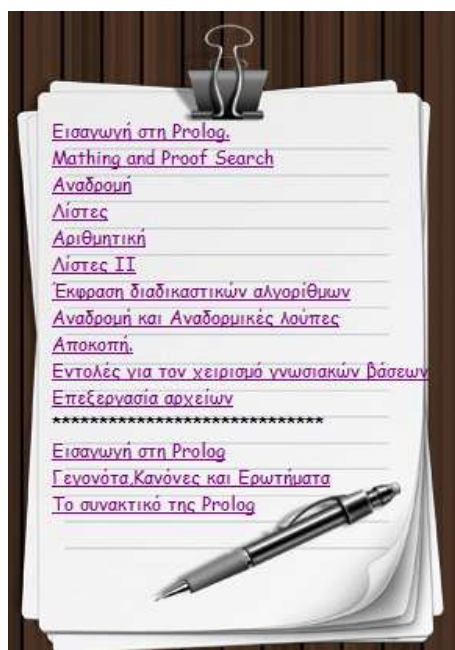


Εικόνα 16 Το μενού γρήγορης πρόσβασης του Prolog Tutor

Το μενού στα δεξιά δίνει στο χρήστη τη δυνατότητα αφού επιλέξει το κεφάλαιο και στη συνέχεια υποκεφάλαιο της αρεσκείας του να διαβάσει τη θεωρία και να κάνει τις ασκήσεις που αντιστοιχούν σε αυτό.

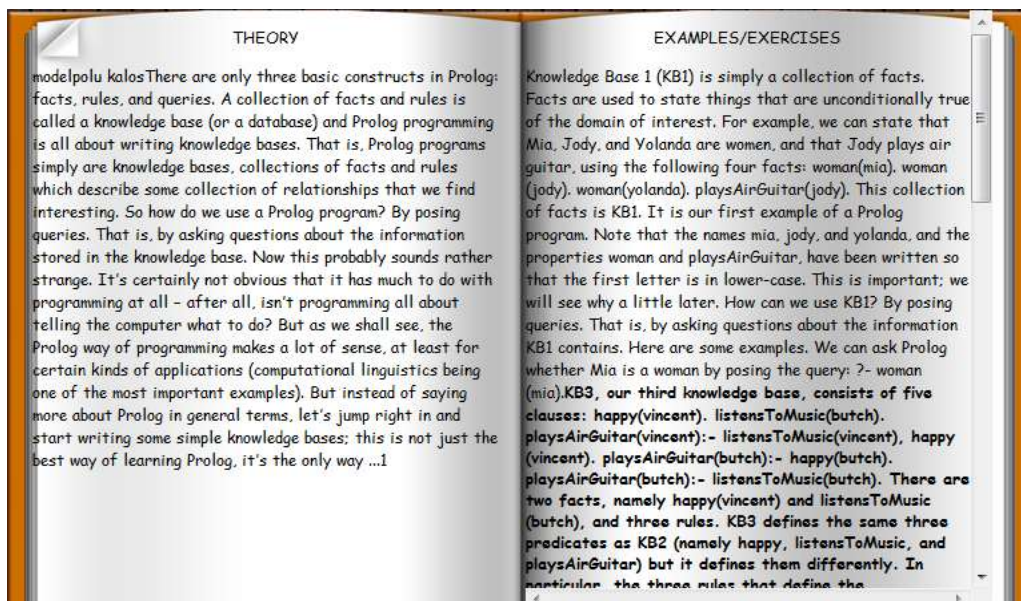


Εικόνα 17 Το μενού με τα κεφάλαια της θεωρίας



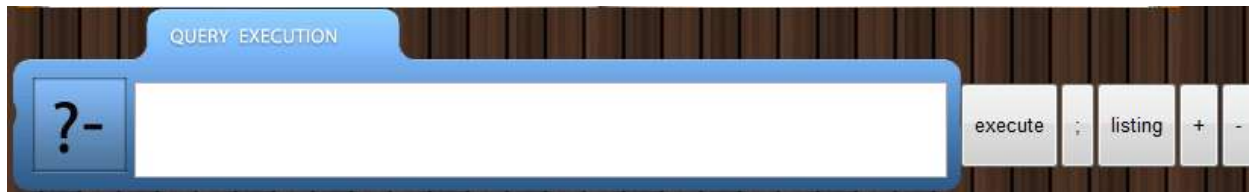
Εικόνα 18 Εμφάνιση των υποκεφαλαίων του κεφαλαίου Εισαγωγή στη Prolog

Αφού επιλέξει ο χρήστης το επιθυμητό υποκεφάλαιο τότε η θεωρία και τα παραδείγματα εμφανίζονται στην κυρίως περιοχή της εφαρμογής.



Εικόνα 19 Η θεωρία και τα παραδείγματα του κεφαλαίου 1

Στο κάτω μέρος της οθόνης παρουσιάζεται η φόρμα εκτέλεσης ερωτημάτων μέσω της οποίας ο χρήστης έχει τη δυνατότητα να εκτελεί τα παραδείγματα που του παρουσιάζονται.



Εικόνα 20 Η φόρμα εκτέλεσης ερωτημάτων

Με την εισαγωγή ενός ερωτήματος (query) στην φόρμα εκτέλεσης ερωτημάτων ο χρήστης έχει την δυνατότητα να επιλέξει μια από τις ακόλουθες λειτουργίες-κουμπιά:

1. «execute», που εκτελεί τα ερωτήματα που εισάγει ο χρήστης,
2. «;», εμφανίζει εναλλακτικές λύσεις στο ερώτημά του, εάν υπάρχουν,
3. «+», προσθέτει νέα κατηγορήματα στη γνωσιακή βάση,
4. «-», αφαιρεί προτάσεις (clauses) από τη γνωσιακή βάση,
5. «listing», όπου παρατίθενται όλα τα περιεχόμενα της γνωσιακής βάσης.

Ο χρήστης σε κάθε κεφάλαιο έχει τη δυνατότητα να μελετά τη θεωρία και να εξασκείται εκτελώντας τα δοθέντα παραδείγματα και ασκήσεις. Είναι πολύ σημαντικό να τονιστεί ότι ο χρήστης έχει τη δυνατότητα να εισάγει και διαγράφει δικά του γεγονότα και κανόνες στη βάση και να θέτει ερωτήματα πέραν αυτών που «υπαγορεύονται» στις ασκήσεις εμπέδωσης. Με αυτό τον τρόπο υποστηρίζουμε την ενεργή οικοδόμηση της γνώσης αλλά την βαθύτερη και καλύτερη κατανόηση μέσω του learning by doing.

Ας υποθέσουμε ότι ο χρήστης θέλει να τρέξει το ερώτημα: «hungry(mary)». Με την επιλογή του κουμπιού execute στο παρασκήνιο εκτελείται μια ακολουθία από τις εξής λειτουργίες:


```
existence(A) :-
    ( functor(A, B, C),
      current_predicate(B/C)
    -> write('predicate exists')
    ; write('p redicate does not exist')
    ).

check_clause_existance(A) :-
    ( \+ call(A)
    -> write('Clause doesnt exist.')
    ; write('Clause does exist.')
    ).
```

Εικόνα 21 Ο κώδικας check existence για τα κατηγορήματα και check clause existence για τις προτάσεις.

Αρχικά, γίνεται έλεγχος του κατηγορήματος που εισήγαγε ο χρήστης, αν υπάρχει δηλαδή το κατηγορήμα hungry στη γνωσιακή βάση. Αν υπάρχει το κατηγορήμα εμφανίζει στο χρήστη ένα μήνυμα όπου τον ενημερώνει σχετικά και ακολουθεί μια άλλη «συνάρτηση» η οποία ελέγχει εάν υπάρχει ήδη η συγκεκριμένη «πρόταση» (clause) αποθηκευμένη στη γνωσιακή βάση. Εάν υπάρχει τότε εμφανίζεται στον χρήστη το μήνυμα true (ότι αυτό δηλαδή που εισήγαγε είναι πράγματι αληθές σύμφωνα με τους κανόνες και τα γεγονότα που είναι αποθηκευμένα). Σε αντίθετη περίπτωση εμφανίζεται στον χρήστη το μήνυμα false.

Σε περίπτωση που ο χρήστης θελήσει να εισάγει το προαναφερθέν στη γνωσιακή βάση τότε εισάγει εκ νέου το ερώτημα και κατόπιν επιλέγει το κουμπί «+» του οποίου ο σκοπός είναι να αποθηκεύει νέα δεδομένα στη βάση. Εφόσον πατηθεί το κουμπί τότε εκτελείται η εξής διαδικασία: πραγματοποιείται έλεγχος αν αυτό που έχει εισάγει ο χρήστης υπάρχει ήδη στη βάση. Σε περίπτωση που υπάρχει, ο χρήστης ενημερώνεται σχετικά και το σύστημα δεν προχωρά στην προσθήκη του στη βάση. Σε περίπτωση όμως που δεν υπάρχει τότε το σύστημα προβαίνει στην προσθήκη του.

```
add_predicate(B, A) :-  
    append(A,  
        nl,  
        write(':-dynamic('),  
        functor(B, C, D),  
        write(C),  
        write('/),  
        write(D),  
        write(').'),  
        nl,  
        write(B),  
        write('.'),  
        told.
```

Εικόνα 22 Ο κώδικας για την προσθήκη clause στη γνωσιακή βάση

Μπορεί στη συνέχεια, αν το θελήσει να εισάγει και άλλα clauses που αφορούν το συγκεκριμένο κατηγορήμα ακολουθώντας την προαναφερθείσα διαδικασία. Παραδείγματος χάριν hungry(george), hungry(kate).

Ο χρήστης μπορεί να εισάγει και ερωτήματα που απαντούν στην ερώτηση hungry(X), δηλαδή ποιοι, σύμφωνα με τη γνωσιακή βάση μας, πεινάνε. Σε αυτή τη περίπτωση εισάγει στη φόρμα εκτέλεσης ερωτημάτων το ερώτημα: hungry(X). Πατώντας execute εμφανίζεται στον χρήστη hungry(mary). Πατώντας το κουμπί «;» διαδοχικά ο χρήστης έχει πρόσβαση και στις εναλλακτικές λύσεις hungry(george) και hungry(kate). Επειδή όμως σύμφωνα με αυτά που έχουν καταχωρηθεί στη βάση δεν υπάρχει κάτι άλλο που να ικανοποιεί το παραπάνω ερώτημα εμφανίζει στο χρήστη το μήνυμα «no more solutions» .

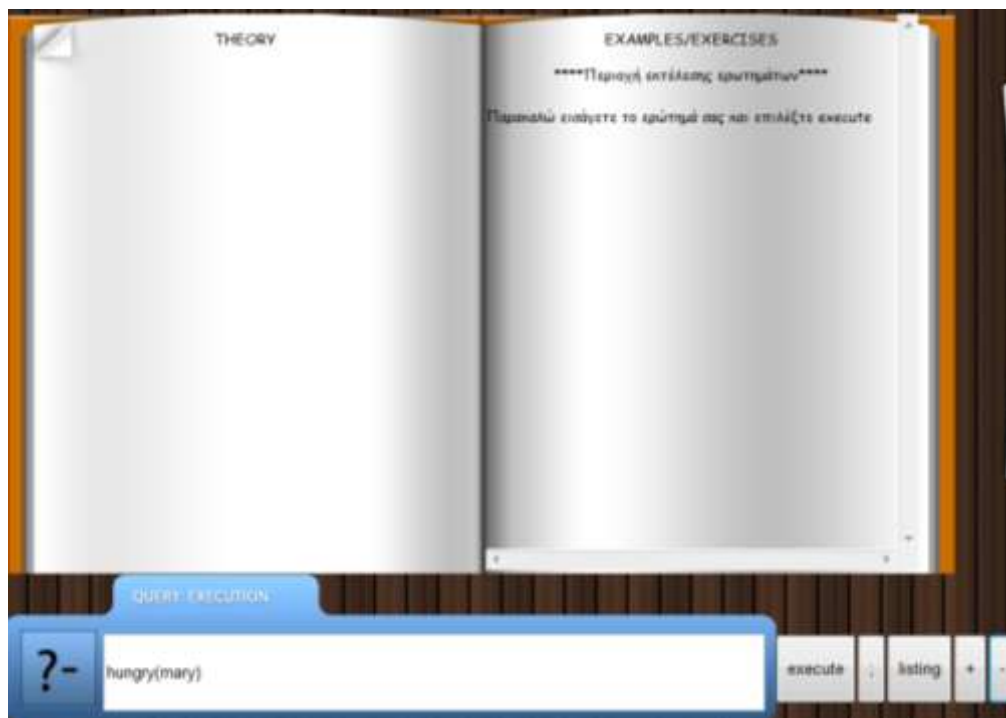
```
find_solutions(A) :-  
    call(A),  
    write(A),  
    nl,  
    !.  
  
find_solutions(_) :-  
    write('no (more) solutions'),  
    nl.
```

Εικόνα 23 Ο κώδικας για την εύρεση παραπάνω της μιας λύσης

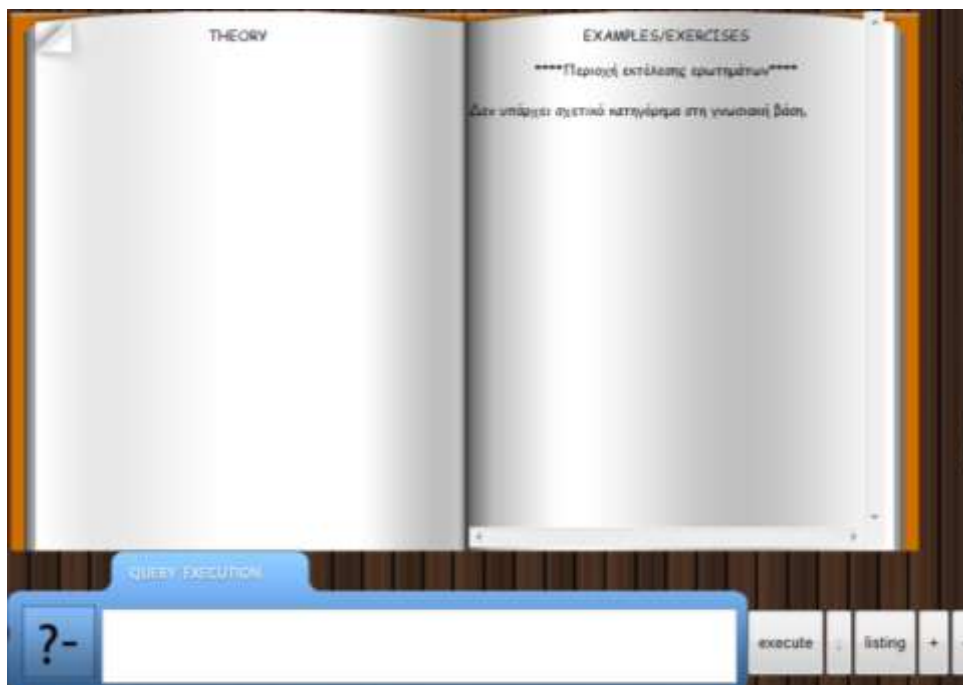
Σε περίπτωση που ο χρήστης θέλει να διαγράψει κάτι από τη βάση, για παράδειγμα το `hungry(mary)` το σύστημα διενεργεί έλεγχο για το κατά πόσο υπάρχει αυτό που θέλει να διαγράψει και σε περίπτωση αδυναμίας εύρεσης ενημερώνει το χρήστη σχετικά. Σε αντίθετη περίπτωση όμως, που το σύστημα εντόπισε το ζητούμενο μέσα στη βάση, τον ενημερώνει για την επιτυχή διαγραφή.

```
retract_predicate(A, B) :-  
    retract(A),  
    tell(B),  
    write(':-ensure_loaded('\add_remove_predicates.pl\').'),  
    nl,  
    listing,  
    told.
```

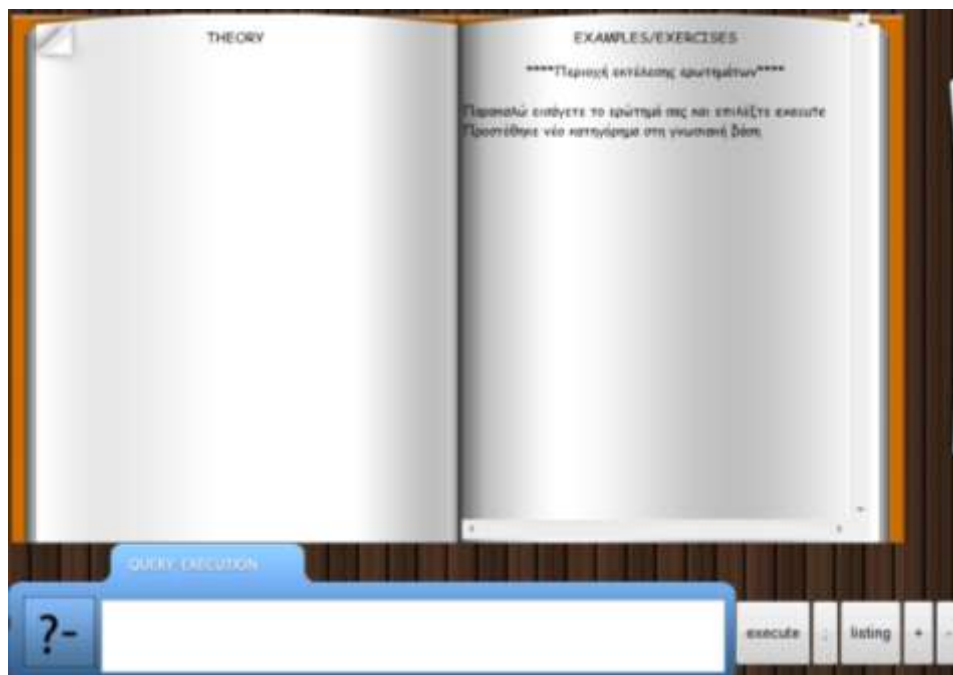
Εικόνα 24 Ο κώδικας για την αφαίρεση προτάσεων από τη γνωσιακή βάση



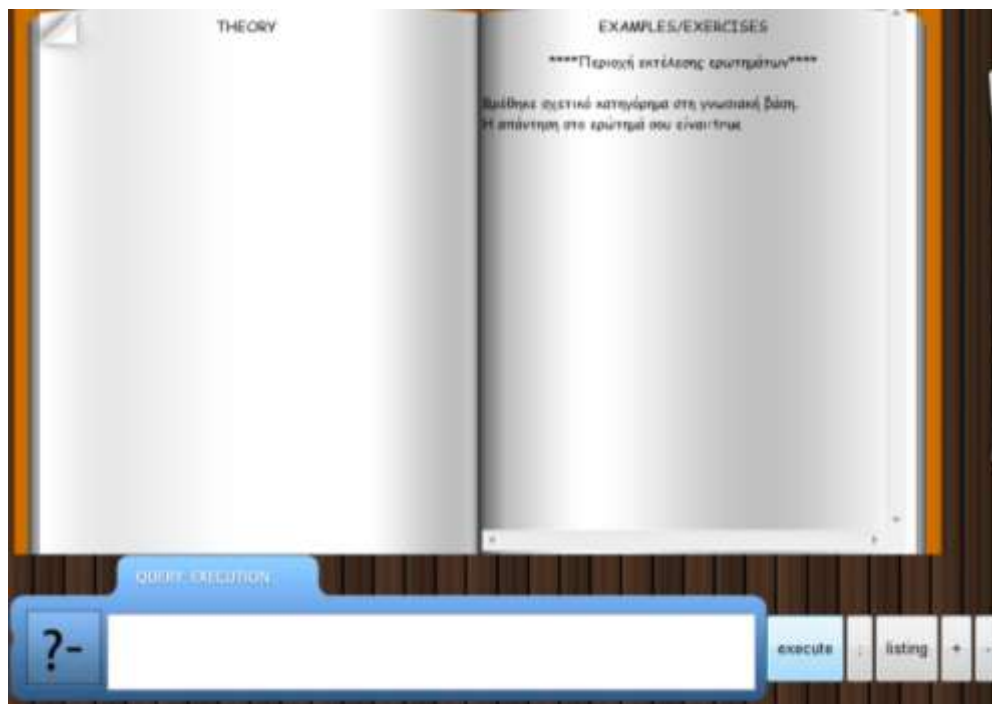
Εικόνα 25 Εκτέλεση του ερωτήματος ?-hungry(mary).



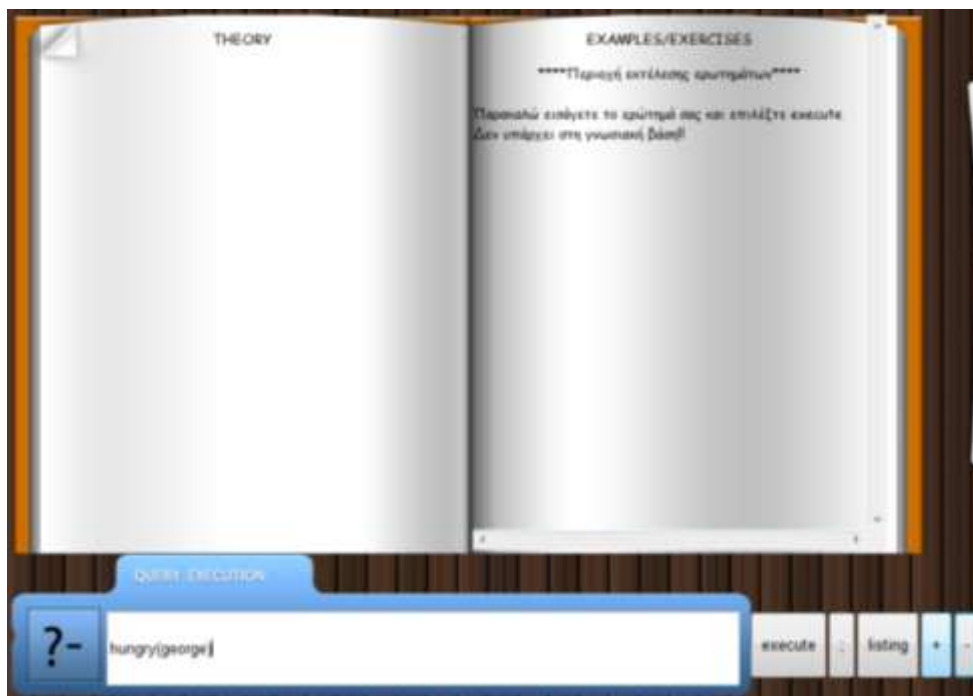
Εικόνα 26 Απάντηση στο ερώτημα ?-hungry(mary).



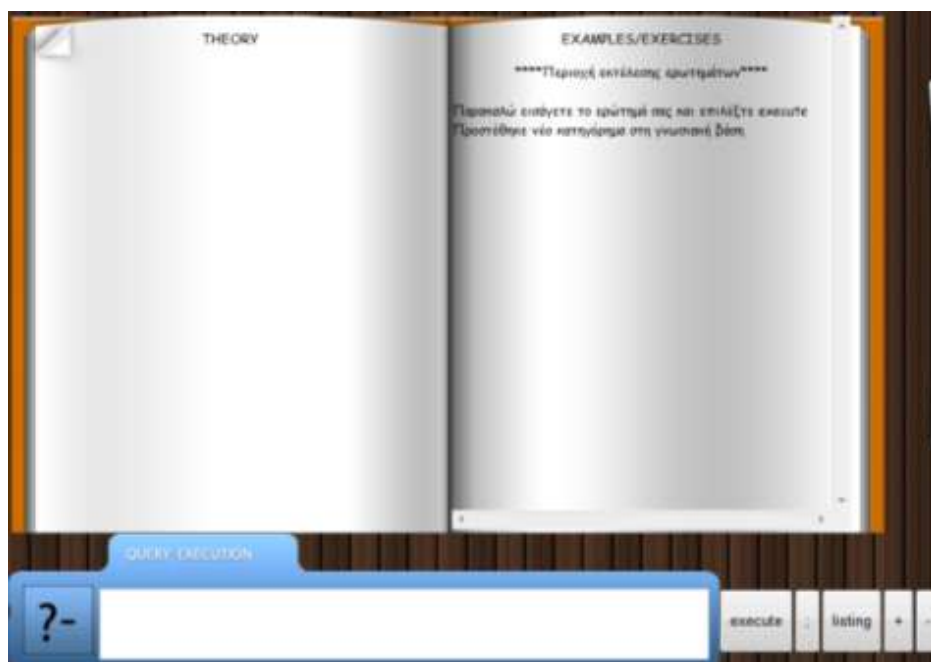
Εικόνα 27 Επιτυχής προσθήκη του hungry(mary) στη γνωσιακή βάση.



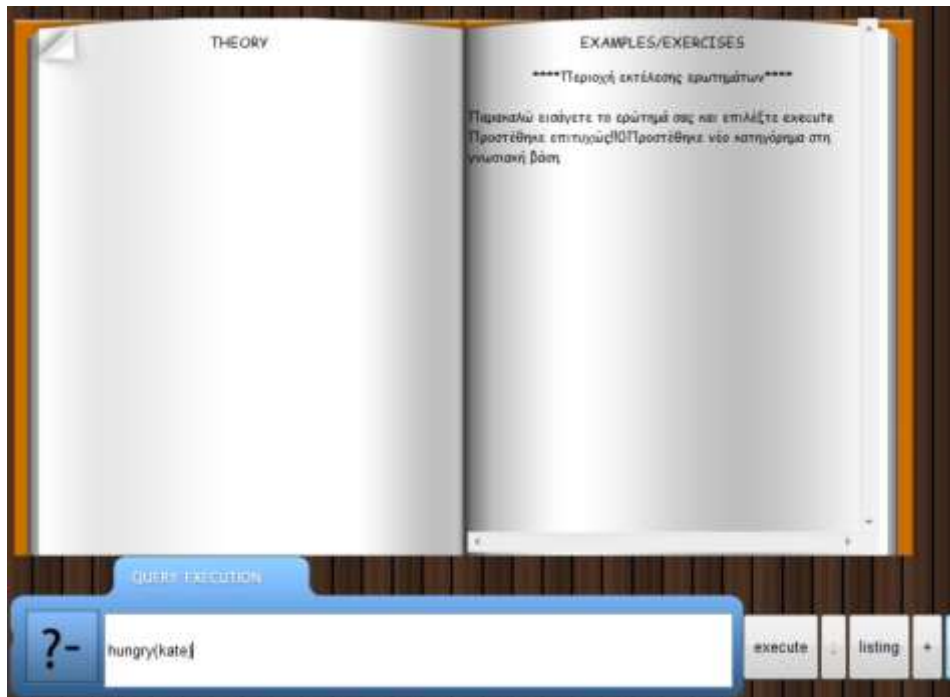
Εικόνα 28 Επανεκτέλεση του ερωτήματος ?-hungry(mary)



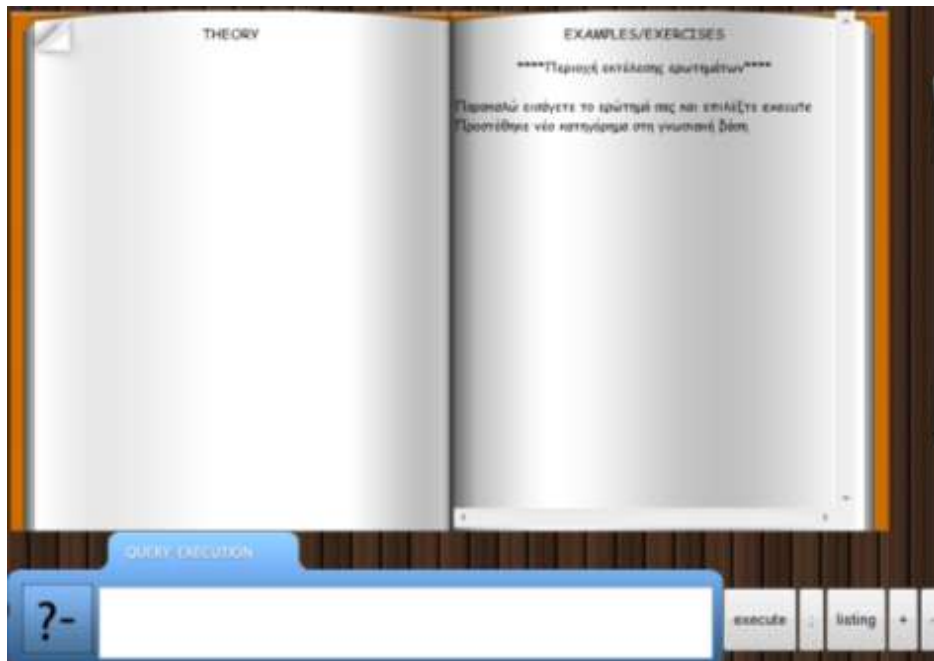
Εικόνα 29 Εισαγωγή του hungry(george) στη γνωσιακή βάση



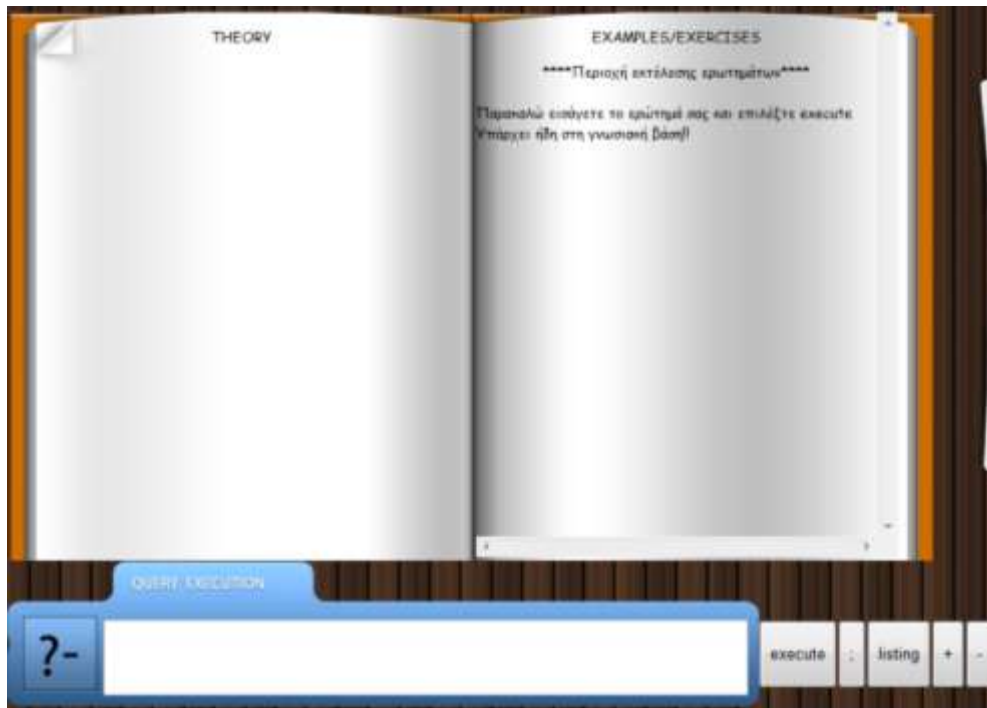
Εικόνα 30 Επιτυχής εισαγωγή του hungry(george) στη γνωσιακή βάση



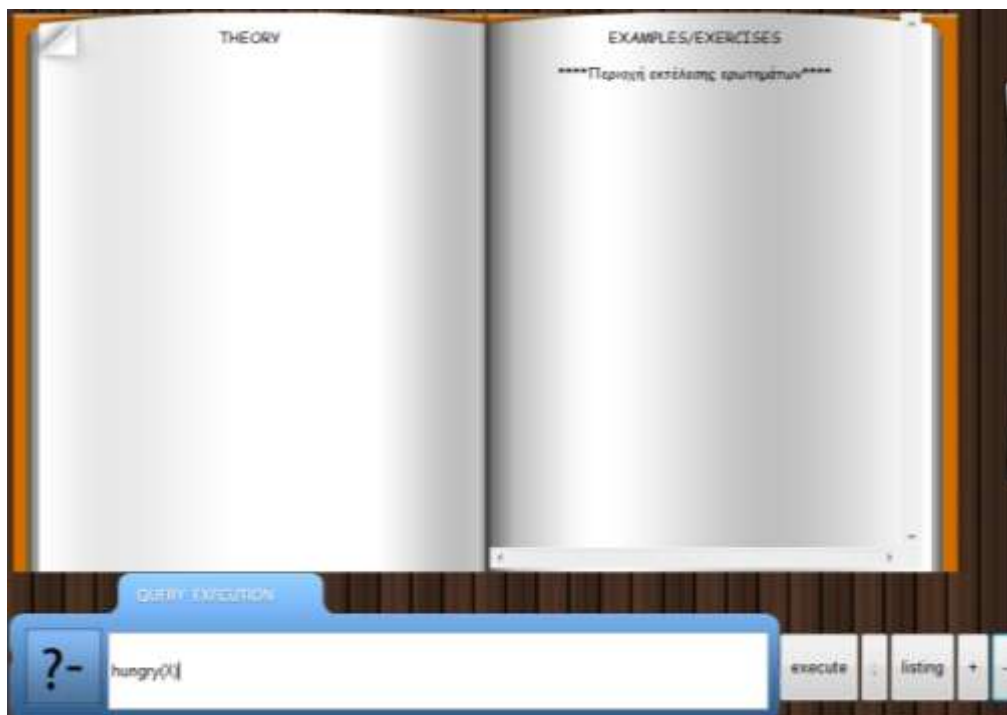
Εικόνα 31 Εισαγωγή του hungry(kate) στη γνωσιακή βάση



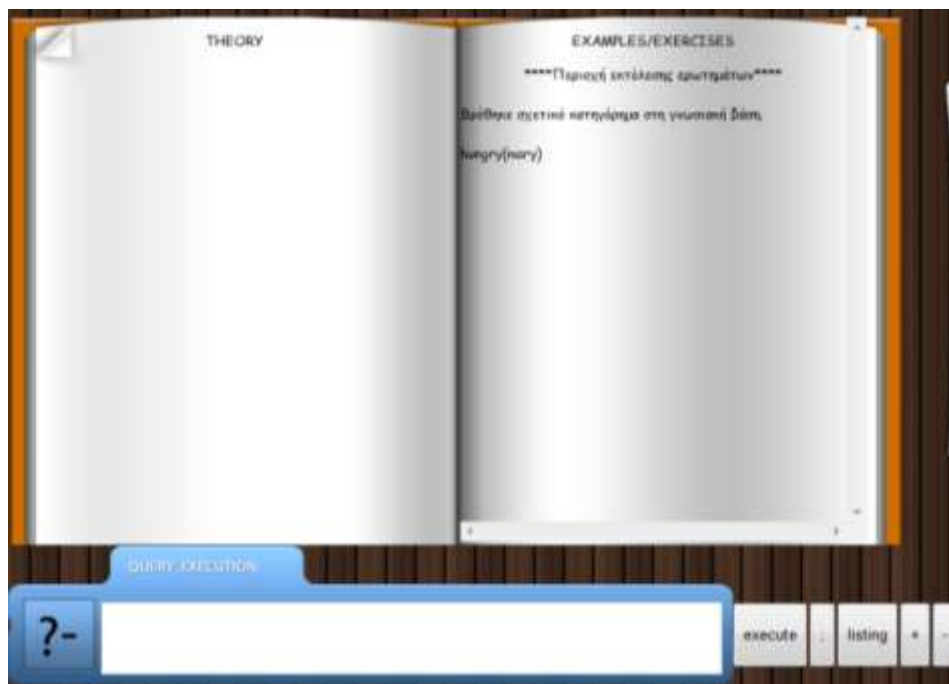
Εικόνα 32 Επιτυχής εισαγωγή του hungry(kate) στη γνωσιακή βάση



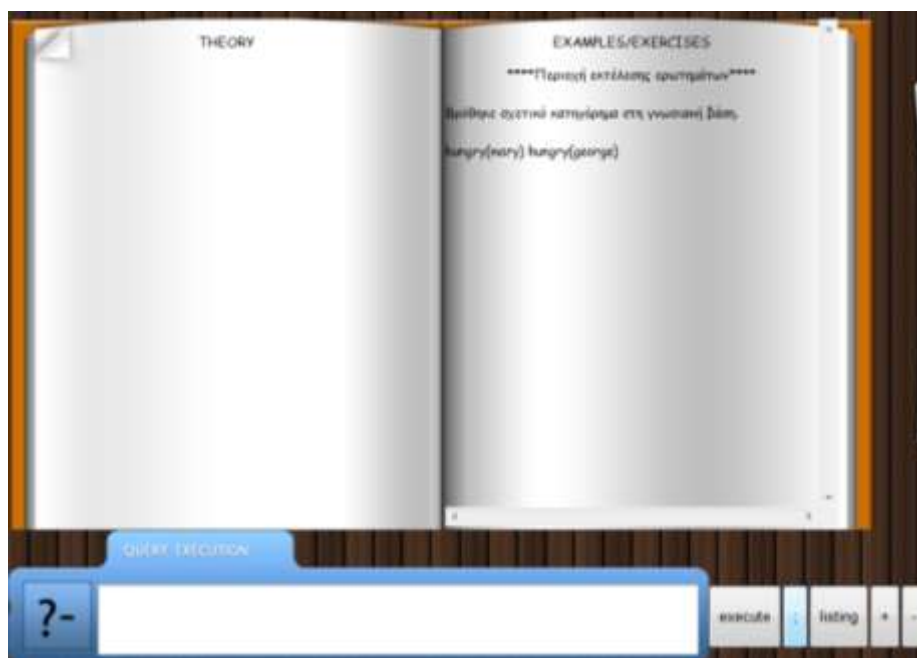
Εικόνα 33 Απάντηση συστήματος στην επανεκτέλεση του ?-hungry(kate)



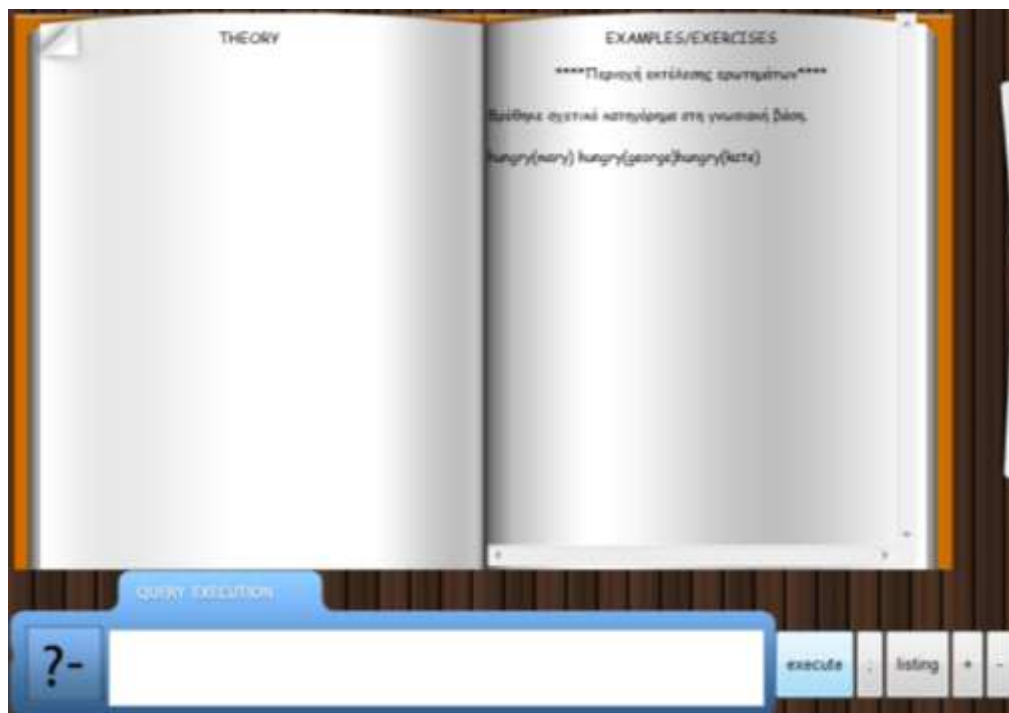
Εικόνα 34 Εκτέλεση του ερωτήματος ?-hungry(X)



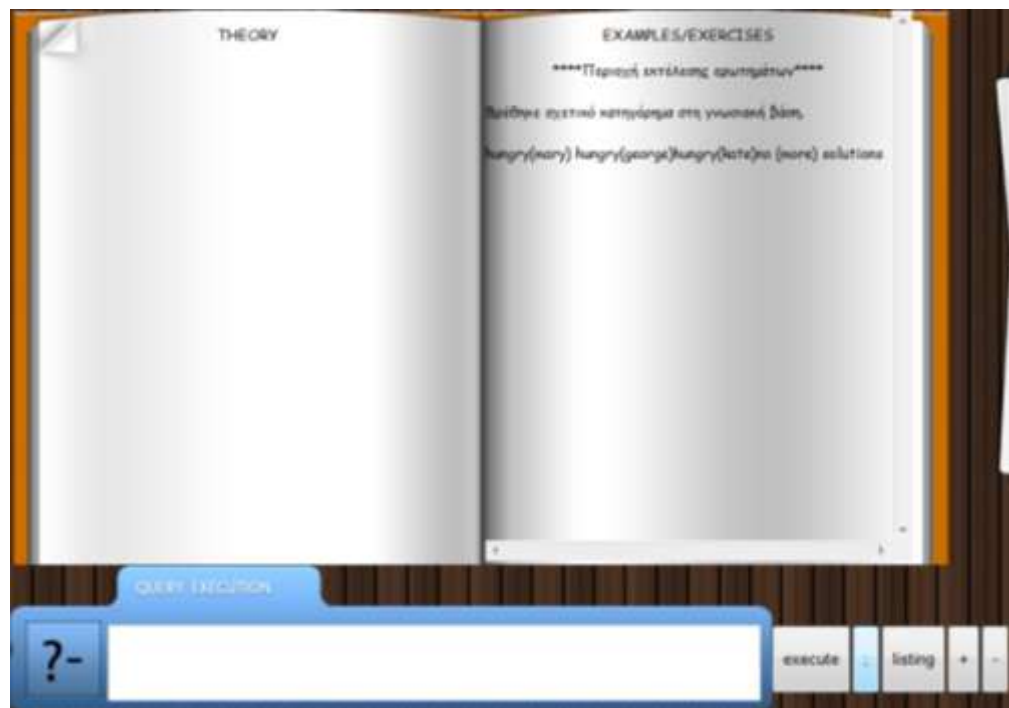
Εικόνα 35 Η πρώτη απάντηση στο ?-hungry(X)



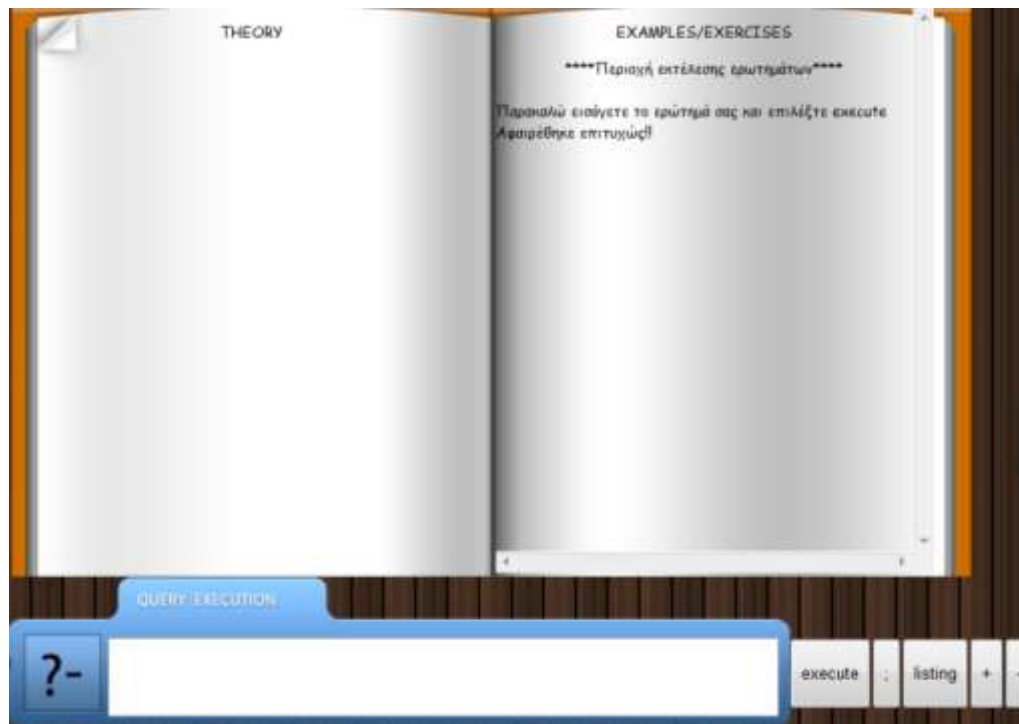
Εικόνα 36 Η δεύτερη απάντηση στο ?-hungry(X)



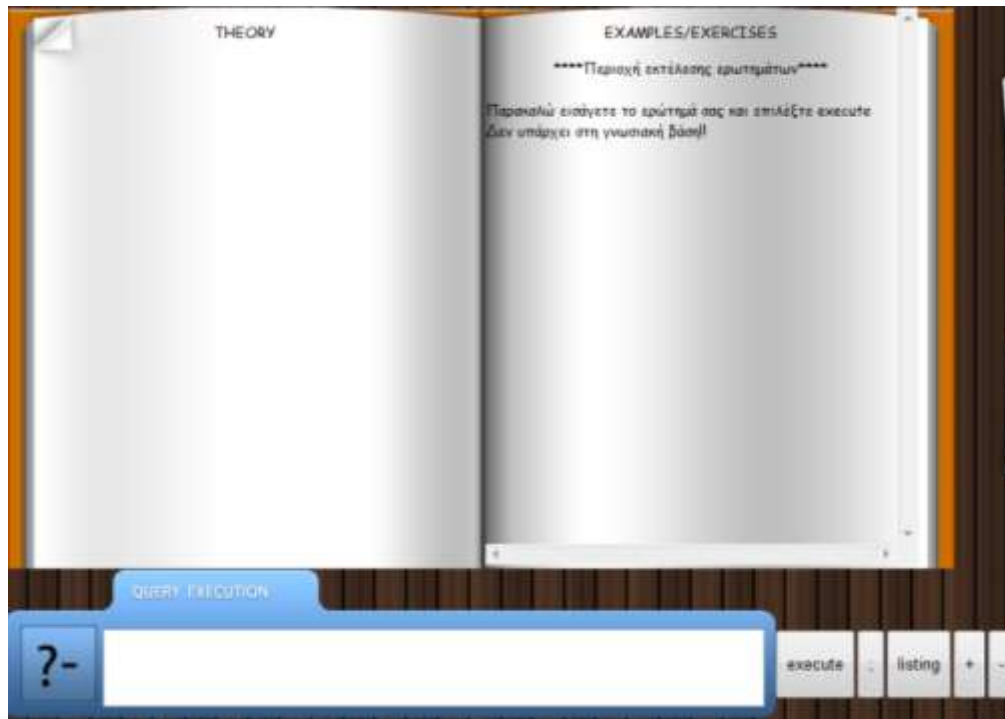
Εικόνα 37 Η τρίτη απάντηση στο hungry(X)



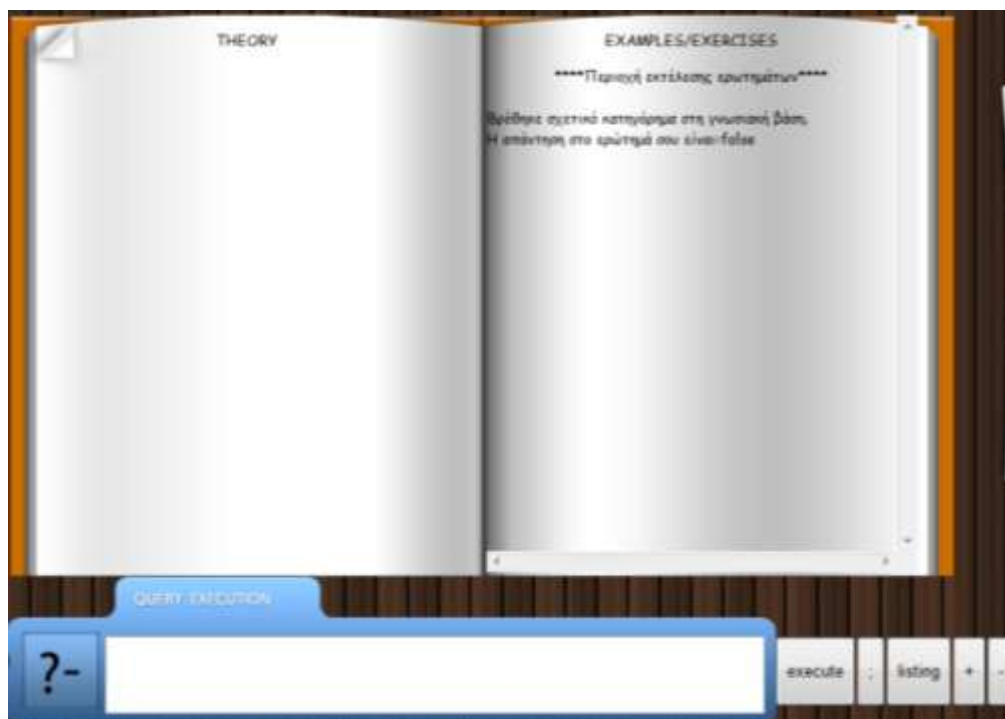
Εικόνα 38 Τέλος εναλλακτικών απαντήσεων στο ερώτημα ?-hungry(X)



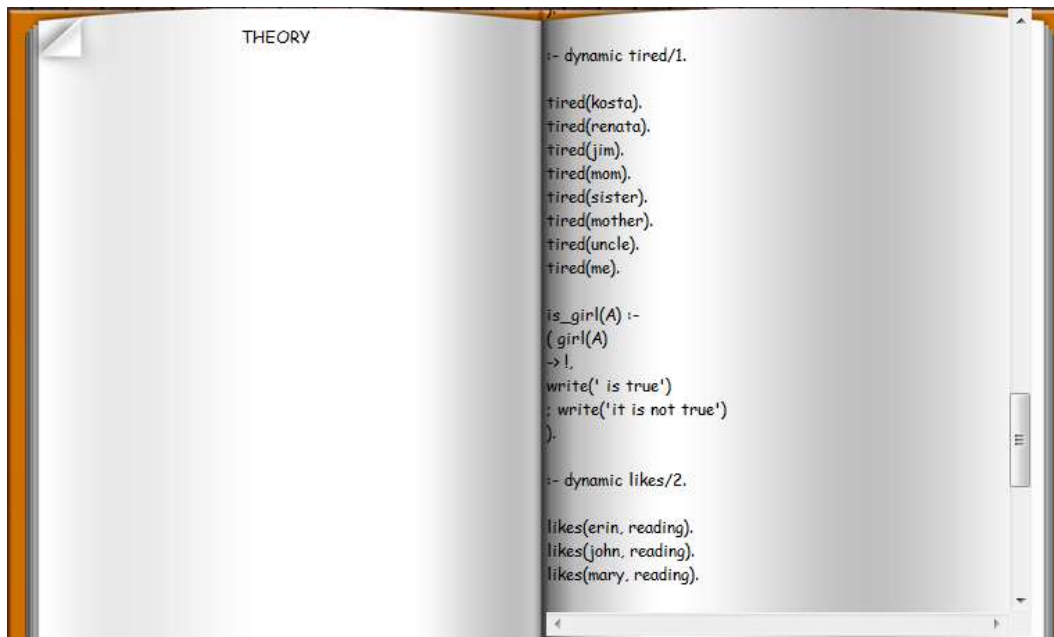
Εικόνα 39 Διαγραφή του hungry(mary) από τη γνωσιακή βάση



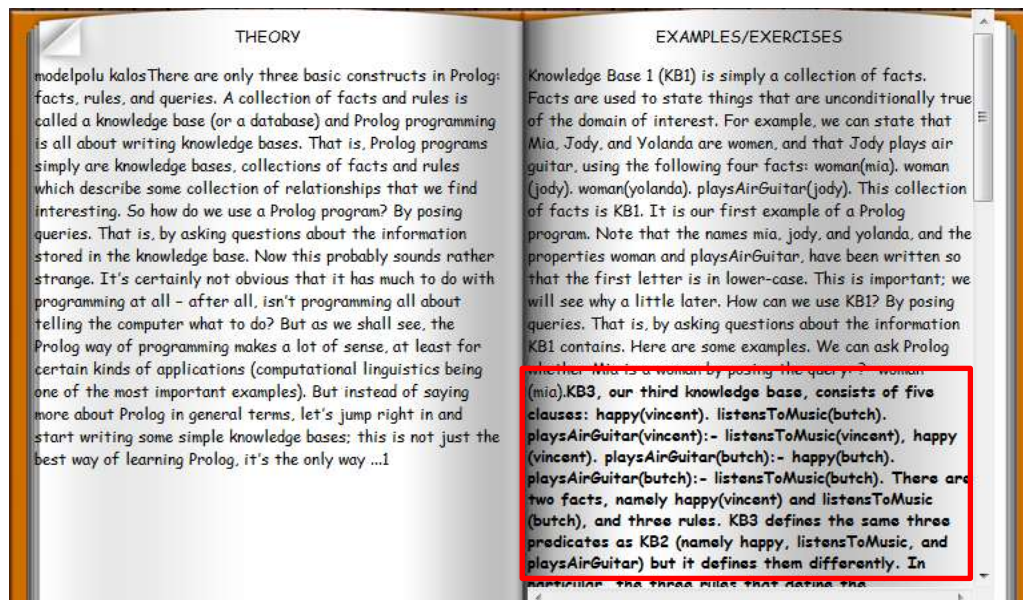
Εικόνα 40 Επιχειρώντας να ξαναδιαγράψουμε το hungry(mary)



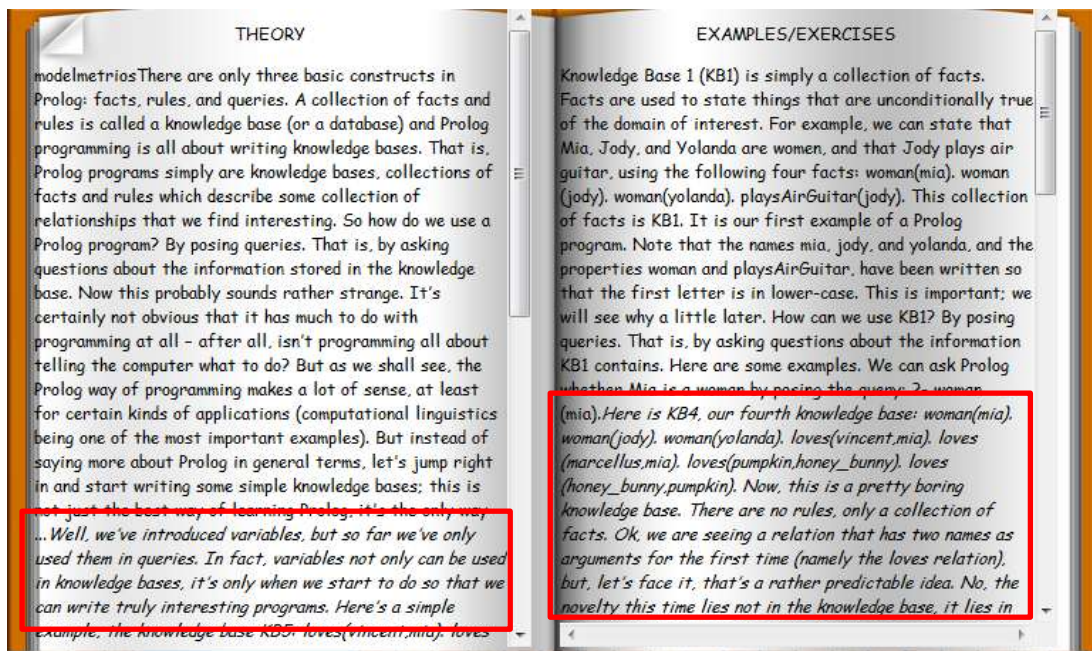
Εικόνα 41 Επανεκτέλεση του hungry(mary) αφού το διαγράψαμε



Εικόνα 42 Απόσπασμα από το listing



Εικόνα 43 Προσαρμοστική παρουσίαση για το μοντέλο χρήστη πολύ καλός



Εικόνα 44 Προσαρμοστική παρουσίαση για το μοντέλο χρήστη μέτριος



Εικόνα 45 Προσαρμοσμένο τεστ και παροχή προσωποποιημένης συμβουλής

Επαναληπτικό Τεστ Κεφαλαίου

Με ποιο σύμβολο τελιώνουν όλα τα ερωτήματα?

- Κόμα
- Τελεία
- Δεν χρειάζεται κάτι στο τέλος
- ΜΕ ΤΟ ?-

Ποιά η σημασία του ":"

- Σημαίνει ή
- Σηματοδοτεί το τέλος του ερωτήματος
- Διαχωρίζει την κεφαλή από την ουρά σε έναν κανόνα.
- Δηλώνει το τέλος ενός κανόνα (rule)

Ποιο από τα παρακάτω είναι μεταβλητή?

- Variable
- v_variable
- vARIABLE
- This Is A Variable

Τι σημαίνει το :-

- πιθανον
- εαν
- ισως
- αφου

Εικόνα 46 Οι ερωτήσεις 1-4 του τεστ για το κεφάλαιο 1

Τι είναι η γνωσιακή βάση?

- Μια συλλογή δεδομένων και κανόνων
- Μια συλλογή δεδομένων που αφορούν τον πραγματικό κόσμο.
- Μια συλλογή δεδομένων και κανόνων που αφορούν τον πραγματικό κόσμο.
- Μια συλλογή δεδομένων

Τποια είναι τα "δομικά συστατικά" ενός προγράμματος Prolog?

- Κανόνες και ερωτήματα
- Ερωτήματα και σχέσεις
- Ερωτήματα
- Κανόνες, γεγονότα και ερωτήματα

Τποιο από τα παρακάτω αντιπροσωπεύει σωστά τον όρο love(vincent,marcellus,mia)?

- love/3
- :-love/3.
- love.
- love/2

Εικόνα 47 Οι ερωτήσεις 5-7

Δίνεται η εξής βάση: `woman(vincent), woman(mia), man(jules), person(X) :- man(X); woman(X), loves(X,Y) :- knows(Y,X), father(Y,Z) :- man(Y), son(Z,Y), father(Y,Z) :- man(Y), daughter(Z,Y)`. Τι επιστρέφει το κάτωθι ερώτημα? `?-person(X)`.

- X=vincent, X=mia.
- Τίποτα. Δεν υπάρχει κάποιος-α στη βάση που να είναι ταυτόχρονα και άντρας και γυναίκα.
- X=vincent, X=mia, X=jules.
- X=jules.

Πως μπορεί να γραφεί διαφορετικά το κάτωθι:
`playsAirGuitar(butch):- happy(butch), playsAirGuitar(butch):- listensToMusic(butch)`.

- `playsAirGuitar(butch), happy(butch), playsAirGuitar(butch), listensToMusic(butch)`.
- `playsAirGuitar(butch):- happy(butch), playsAirGuitar(butch):- listensToMusic(butch)`.
- `playsAirGuitar(butch):- happy(butch), listensToMusic(butch)`.
- `playsAirGuitar(butch):- happy(butch); listensToMusic(butch)`.

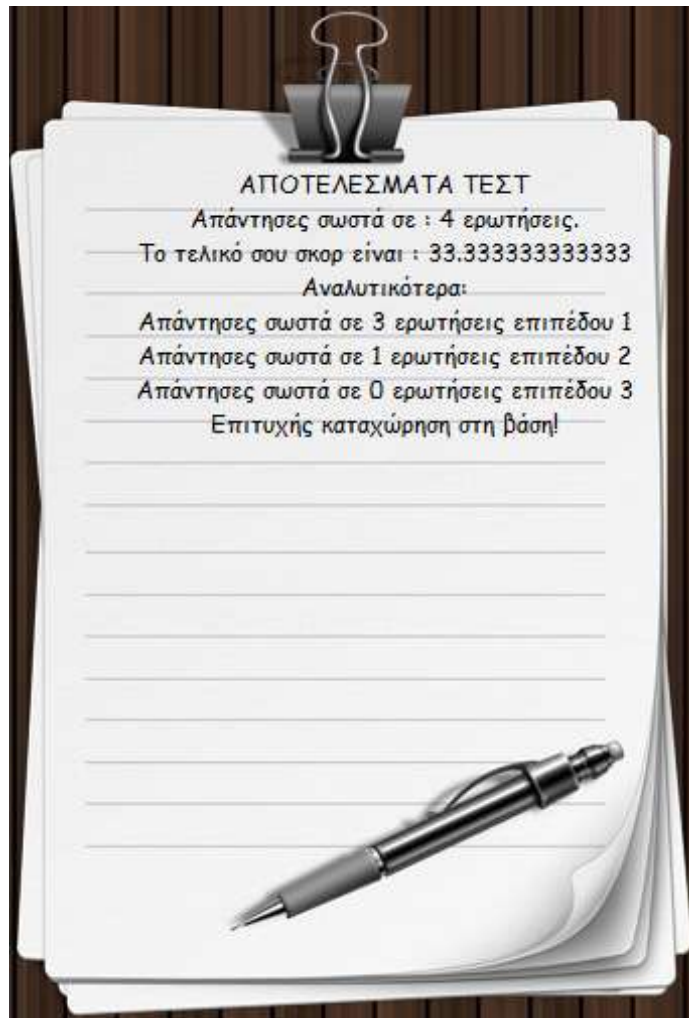
Εικόνα 48 Οι ερωτήσεις 8-9

Τι πληθικότητας (arity) είναι το `hide(X,father(father(father(butch))))` ?

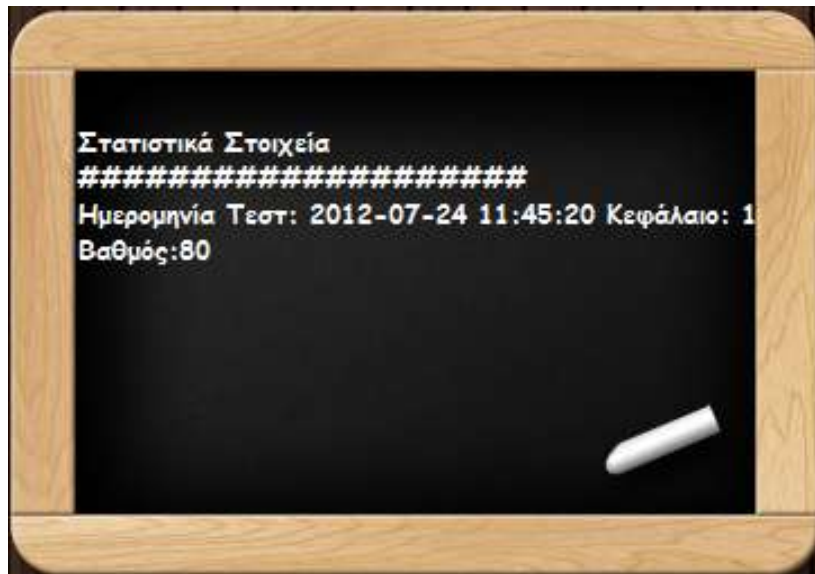
- 6
- 3
- 2
- 5

Υποβολή

Εικόνα 49 Η ερώτηση 10



Εικόνα 50 Αποτελέσματα τεστ



Εικόνα 51 Εμφάνιση στατιστικών χρήστη

5. Συμπεράσματα

Η ανάγκη για προσαρμοστικά διαδικτυακά συστήματα διδασκαλίας κρίνεται άκρως επιτακτική και αυτό λόγω του όγκου, της πολυπλοκότητας και ετερογένειας της παρεχόμενης πληροφορίας που κατακλύζει τον Παγκόσμιο Ιστό (Brusilovsky, 1998).

Το προσαρμοστικό σύστημα Web Tutor το οποίο σχεδιάστηκε και υλοποιήθηκε στην παρούσα μεταπτυχιακή διατριβή διαφέρει από όλα τα υπάρχοντα συστήματα διδασκαλίας Prolog και η διαφορά του έγκειται στο γεγονός ότι αφενός λαμβάνεται υπόψη η διαφορετικότητα του κάθε χρήστη (επίπεδο γνώσης της εν λόγω αλλά και άλλων γλωσσών προγραμματισμού, ο προσωπικός ρυθμός μάθησης) και αφετέρου η χρήση αλληλεπίδρασης η οποία υποβοηθά την ενεργή οικοδόμηση της γνώσης καθώς και την εις βάθος κατανόηση.

Η διάθεση του συστήματος μέσω του Διαδικτύου το καθιστά εύκολα διαθέσιμο σε πλειάδα χρηστών πανεπιστημιακού και μη χώρου.

Το παρόν σύστημα θα μπορούσε να ενισχύσει την περιορισμένη ελληνική βιβλιογραφία όσον αφορά στην εκμάθηση της Prolog.

6. Παράρτημα

Για την εγκατάσταση της εφαρμογής στον υπολογιστή του χρήστη θα πρέπει να διαθέτει τα εξής:

Wamp Server εκδόσεως 2.1 και άνω

SWI-Prolog έκδοσης 6.1.11

Μετά την επιτυχή εγκατάσταση των ανωτέρω προγραμμάτων θα πρέπει ο χρήστης να αντιγράψει και να επικολλήσει τον φάκελο `ad_web` στη διαδρομή «C:\wamp\www».

7. Βιβλιογραφία

- Κουράκος Μαυρομιχάλης, Ε. (2006). *Ο μηχανισμός ελέγχου της Prolog*. Ανάκτηση από Εισαγωγή στη Γλώσσα Προγραμματισμού Prolog : http://www.icsd.aegean.gr/emav/prolog/Ch3/Ch3_1.htm
- Beaumont, I. (1998). User Modelling in the Interactive Anatomy Tutoring System ANATOM-TUTOR. In P. Brusilovsky, A. Kobsa, & J. Vassileva, *Adaptive Hypertext and Hypermedia* (pp. 91-115). Netherlands: Kluwer Academic Publishers.
- Blackburn, P., Bos, J., & Striegnitz, K. (2001). *Learn Prolog Now!*
- Boyle, G., & Encarnacion, A. (1998). Metadoc: An Adaptive Hypertext Reading System. In P. Brusilovsky, A. Kobsa, & J. Vassileva, *Adaptive Hypertext and Hypermedia* (pp. 71-89). Netherlands: Kluwer Academic Publishers.
- Brna, P., Pain, H., & Du Bulaiy, B. (1990). Teaching, learning and using Prolog: Understanding Prolog. *Instructional Science*(19), pp. 247-256.
- Brusilovsky, P. (1998). Methods and Techniques of Adaptive Hypermedia. In P. Brusilovsky, A. Kobsa, & J. Vassileva, *Adaptive Hypertext and Hypermedia* (pp. 1-39). Netherlands: Kluwer Academic Publishers.
- Chrysafiadi, K., & Virvou, M. (2008). Personalized Teaching of a Programming language over the web: Stereotypes and rule-based mechanisms. In M. Virvou, & T. Nakamura (Ed.), *Eighth Joint Conference on Knowledge-based Software Engineering* (pp. 484-492). Netherlands: IOS PRESS.
- Colmerauer, A., & Roussel, P. (1992). The birth of Prolog.
- Cope, P. (1989). Teaching PROLOG Programming in the secondary School. *Journal of Computer Assisted Learning*(5), 223-230.
- Fischetti, E., & Gisolfi, A. (1990). Logprimer: A Tutoring System For Prolog Learning. *Computer Education*.
- Fregier, M. (n.d.). Teaching Prolog Programming to CAD Design Students.
- Hohl, H., Bocker, H.-D., & Gunzenhauser, R. (1998). Hypadapter: An adaptive Hypertext System for Exploratory Learning and Programming. In P. Brusilovsky,

- A. Kobsa, & J. Vassileva, *Adaptive Hypertext and Hypermedia* (pp. 117-142). Netherlands: Kluwer Academic Publishers.
- Hook, K., Karlgren, J., Waern, A., Dahlback, N., Jansson, C., Karlgren, K., et al. (1998). A Glass Box Approach to Adaptive Hypermedia. In P. Brusilovsky, A. Kobsa, & J. Vassileva, *Adaptive Hypertext and Hypermedia* (pp. 143-170). Netherlands: Kluwer Academic Publishers.
- Kaplan, G., Fenwick, J., & Chen, J. (1998). Adaptive Hypertext Navigation Based On User Goals and Context. In P. Brusilovsky, A. Kobsa, & J. Vassileva, *Adaptive Hypertext and Hypermedia* (pp. 45-69). Netherlands: Kluwer Academic Publisher.
- Mano, C. (2012). *Reasons to use Prolog*. Retrieved from Ehow:
http://www.ehow.com/list_7396622_reasons-use-prolog.html
- Nalepa, G. J., & Wonjincki, I. (n.d.). Concept of an Interactive Web Portal for Teaching Prolog.
- Nalepa, G., & Wojnicki, I. (2008). Concept of an Interactive Web Portal for Teaching Prolog. *Association for the Advancement of Artificial Intelligence*.
- Neumerkel, U. (n.d.). Teaching beginners Prolog. How to teach Prolog.
- Oppermann, R., Rashev, R., & Kinshuk. (n.d.). Adaptability and adaptivity in Learning Systems.
- Yang, S., & Joy, M. (2006). Approaches for Teaching Prolog to Beginners. *Higher Education Academy*, pp. 106-110.
- Βλαχάβας, Ι., Ρεφανίδης, Ι., & Σακελλαρίου, Η. (n.d.). Πολυμεσικό Σύστημα για τη Διδασκαλία της γλώσσας Προγραμματισμού Prolog.
- Σγάρμπας, Κ. (2006). *Η γλώσσα προγραμματισμού Prolog*. Πατρα: Πανεπιστήμιο Πατρών.