

ΥΛΟΠΟΙΗΣΗ ΦΩΝΗΤΙΚΗΣ ΠΥΛΗΣ ΠΟΛΥΤΕΧΝΕΙΟΥ
ΚΡΗΤΗΣ

Γερμάνης Γ. Θεόδωρος

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΣΤΟ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΙΟΥΛΙΟΣ 2006

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ

ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Αφιερωμένο σε όλη την τρελοπαρέα των Χανίων, και σε όλους βρέθηκαν έστω και μια στιγμή σε αυτή την όμορφη πόλη κατά την διάρκεια αυτών των αξέχαστων 5 χρόνων.

Ευχαριστίες

Θα ήθελα με την ευκαιρία αυτή να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Βασίλη Διγαλάκη για την εμπιστοσύνη που έδειξε στ πρόσωπό μου και τις πάντα εύστοχες παρατηρήσεις και υποδείξεις του. Η βοήθεια που μου παρείχε μέσω της εμπειρίας του στον τομέα εφαρμογών φωνής ήταν ανεκτίμητης αξίας. Θα ήθελα επίσης να ευχαριστήσω τον υποψήφιο διδάκτορα Χρήστο Βοσνίδη για την συνεργασία του και την καθοδήγησή του κατά την διάρκεια εκπόνησης της εργασίας. Χωρίς την βοήθεια και την φιλία του ακόμα θα βρισκόμουν χαμένος στις συμπληγάδες της υλοποίησης μια τέτοιας εφαρμογής.. Επίσης οφείλω ευχαριστίες στον υποψήφιο διδάκτορα Κώστα Χαριζάκη για την αμέριστη βοήθειά του σε πολλά δύσκολα και “πονηρά” σημεία της υλοποίησης και τον διδάκτορα Βασίλη Διακολουκά για την πολύτιμη βοήθειά του κατά την διάρκεια της αξιολόγησης του συστήματός.

Επίσης θα ήθελα να ευχαριστήσω τον κ. Αλέξανδρο Ποταμιάνο για την βοήθεια που μου παρείχε κατά την διάρκεια των μαθημάτων σχετικών με την εργασία και για την ανάγνωση της διπλωματικής εργασίας και για τα σχόλιά του.

Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου για την αμέριστη υποστήριξη και αγάπη τους σε όλες τις δύσκολες στιγμές. Την φίλη μου Ελένη (για όσα δεινά της προξένησα).Και τους φίλους μου Βάρδο (για τον όγκο του), Γκούνη (για την απλότητά της σκέψης του), Κομμάτια (για την αξέχαστη συγκατοίκηση), Κροκόδειλο (για το γέλιο...), Λύκο (για την απουσία του), Μπόκα (νεβερμάνιτ), Τροπ (για την γκρίνια του), Τσίκο Τσίκο (για τις θεωρητικό-οικονομικές αναζητήσεις), Στρατηγό (για τους καφέδες!),τρόμο-Μύγα και όλους τους άλλους που, αν ήταν να τους αναφέρω θα χρειαζόταν μια νέα διπλωματική εργασία.

Περιεχόμενα

Εισαγωγή	7
1. Προδιαγραφές Φωνητικής Πύλης για το Πολυτεχνείο Κρήτης.....	11
1.1 Απαιτήσεις Συστήματος Web-based (HTML).....	11
1.2 Απαιτήσεις Φωνητικού Συστήματος	13
2. Τεχνολογίες Επεξεργασίας Φωνής	17
2.1 Αναγνώριση Φωνής	17
2.2 Σύνθεση Φωνής.....	20
2.3 Συστήματα Διαλόγου	24
2.4 Αρχές Ανάπτυξης Εφαρμογών Φωνής	27
2.3.1 Φωνητική Είσοδος	28
2.3.2 Σχεδιασμός Για Φωνή – Χαρακτηριστικά και απαιτήσεις	30
2.3.3 Θέματα που αφορούν Εφαρμογές Φωνής.....	35
2.3.4 Εμπλέκοντας τους χρήστες	41
3. Υλοποίηση της Εφαρμογής Φωνητικής Πύλης Πολυτεχνείου Κρήτης...44	44
3.1 Το Αρχιτεκτονικό Μοντέλο Model View Controller	45
3.1.1 Πρότυπα σχεδίασης	48
3.2 Υλοποίηση Front End	50
3.2.1 Η γλώσσα VoiceXML	50
3.2.2 Η Φωνητική Εφαρμογή.....	56
3.2.2.1 Σχεδίαση - Υλοποίηση Διαλόγων	56
3.2.2.2 Υλοποίηση Γραμματικών	62
3.2.3 Εφαρμογή Web-based (HTML).....	69
3.3 Υλοποίηση Back End.....	70
3.3.1 Παρουσίαση της Βάσης Δεδομένων	70
3.3.2 Ανάπτυξη Δυναμικών Εφαρμογών	72
4. Αξιολόγηση	76
4.1 Δοκιμές Χρηστικότητας.....	76

4.2	Εκτίμηση Επιτυχούς Αναγνώρισης.....	77
5.	Συμπεράσματα και Επεκτασιμότητα	84
	Appendix A – Dialog Specification	86
	Appendix B – Usability Questionnaire.....	120
	Appendix C – Technical Issues	124
i.	Η εφαρμογή του MVC στο σύστημά μας.....	125
ii.	VoiceXML Specification.....	132
iii.	Web Based system Specification.....	135
iv.	Data Base	141

Εισαγωγή

Τα τελευταία χρόνια η χρήση Διαδραστικών Συστημάτων Φωνητικής Απόκρισης (Interactive Voice Response Systems – IVR Systems) [1] [2] είναι αρκετά διαδεδομένη ως μέσο επικοινωνίας ανθρώπου - υπολογιστή.

Τα IVR Systems είναι αυτοματοποιημένα συστήματα, τα οποία μπορούν να αλληλεπιδρούν με τους χρήστες μέσω συσκευών τηλεφώνου, παρέχοντάς τους διάφορες πληροφορίες ή υπηρεσίες. Άρχισαν να χρησιμοποιούνται από τα τέλη της δεκαετίας του 80' και η χρήση τους αφορούσε την διαχείριση του μεγάλου αριθμού κλήσεων στα τηλεφωνικά κέντρα των τμημάτων εξυπηρέτησης πελατών μεγάλων εταιρειών και επιχειρήσεων. Άλλες χρήσεις των συστημάτων αυτών είναι στις τραπεζικές συναλλαγές μέσω τηλεφώνου (telephone banking), σε διάφορες υπηρεσίες εξυπηρέτησης κρατήσεων μέσω τηλεφώνου, καθώς και για πληροφορίες χρηματιστηρίου.

Τα περισσότερα IVR systems χρησιμοποιούν dual tone multiple frequency (D. T. M. F.) [2] [3] [4] διεπαφή χρήστη (user interface). Σε αυτές τις περιπτώσεις, ο χρήστης καλεί ένα συγκεκριμένο νούμερο τηλεφώνου και το σύστημα απαντάει αρχίζοντας την αυτόματη διαδικασία διαλόγων. Η αυτοματοποιημένη φωνητική διεπαφή οδηγεί τον χρήστη να κάνει επιλογές χρησιμοποιώντας τα πλήκτρα του τηλεφώνου. Όταν η συνομιλία τελειώσει και ο χρήστης κλείνει το τηλέφωνο, το σύστημα σταματάει και την κλήση. Στην δεκαετία του 80' τα συστήματα βασισμένα σε DTMF είσοδο θεωρούνταν η πιο ενδεδειγμένη λύση για την υλοποίηση διαδραστικών συστημάτων φωνητικής απόκρισης. Σύντομα όμως, άρχισαν μελέτες πάνω σε τεχνικές που είχαν τη δυνατότητα να αναγνωρίσουν την ανθρώπινη φωνή με μεγαλύτερη ακρίβεια, και που σύντομα θα αντικαθιστούσαν τα συστήματα βασισμένα σε DTMF είσοδο. Στα τέλη της δεκαετίας του 80 η ARPA (Advanced Research Project Agency) [7] χρηματοδότησε κάποιους από τους οργανισμούς που δραστηριοποιούνται στην ανάπτυξη συστημάτων αναγνώρισης φωνής με την χρήση ηλεκτρονικών υπολογιστών. Οι οργανισμοί αυτοί προχώρησαν στην δημιουργία κάποιων πρώιμων συστημάτων αναγνώρισης φωνής που μπορούσαν να αναγνωρίσουν ένα πολύ μικρό φάσμα ανθρώπινων φωνών. Ωστόσο τα νέα εξελιγμένα IVR systems μπορούν να διαχειρίζονται αποτελεσματικά πολύ μεγάλο αριθμό χρηστών, καθώς έχουν την δυνατότητα να αναγνωρίσουν φωνητικές εισόδους

από διαφορετικούς χρήστες, κάνοντας χρήση πολύ εξελιγμένων τεχνικών, όπως είναι η τεχνητή νοημοσύνη. Παρακάτω θα δούμε ποιοι είναι λόγοι που κάνουν σήμερα δυνατή την υλοποίηση υψηλής ποιότητας εφαρμογών φωνής.

Αυξημένη υπολογιστική ισχύς με λιγότερο κόστος.

Παίρνοντας υπόψη το γεγονός ότι η υπολογιστική ισχύς των ολοκληρωμένων κυκλωμάτων (chip) διπλασιάζεται περίπου κάθε δεκαοχτώ μήνες-σύμφωνα με τον νόμο του Moore- καταλαβαίνουμε την διαφορά υπολογιστικής ισχύος μεταξύ των σημερινών συστημάτων και των συστημάτων που χρησιμοποιούνταν είκοσι χρόνια πριν. Η υπολογιστική αυτή ισχύς επέτρεψε στην δημιουργία πιο πολύπλοκων και εξειδικευμένων αλγορίθμων τόσο στην αναγνώριση όσο και στην σύνθεση φωνής. Υπολογισμοί, που πριν απαιτούσαν μέρες επεξεργασίας, σήμερα εκτελούνται σε κλάσματα δευτερολέπτων. [4]

Βελτιωμένοι αλγόριθμοι

Οι αλγόριθμοι αναγνώρισης και σύνθεσης φωνής, καθώς και των διαλόγων χρήστη-υπολογιστή έχουν εξελιχθεί σε σημαντικό βαθμό. Έχουν γίνει πολύ πιο εξεζητημένοι και αποτελεσματικοί. Μέσα από μελέτες χρόνων οι αλγόριθμοι επεκτάθηκαν, προσαρμόστηκαν και βελτιστοποιήθηκαν με αποτέλεσμα να είναι πιο γρήγοροι και πιο αποτελεσματικοί. Επίσης, ο ολοένα αυξανόμενος και φτηνός διαθέσιμος αποθηκευτικός χώρος των σύγχρονων συστημάτων σε συνδυασμό με τις βελτιωμένες μεθόδους αναζήτησης, βοήθησαν στην διατήρηση μεγαλύτερων λεξιλογίων και δεδομένων, σχετικών με τους αλγορίθμους αναγνώρισης φωνής. Είναι ενδεικτικό ότι οι σημερινοί αλγόριθμοι έχουν την δυνατότητα να πραγματοποιούν αναζητήσεις σε λεξιλόγια μερικών δεκάδων χιλιάδων λέξεων.

Οι παλιότεροι αλγόριθμοι ήταν πολύ περιοριστικοί για τους χρήστες. Οι χρήστες έπρεπε να λένε κάθε λέξη ξεχωριστά αφήνοντας μια μικρή παύση ανάμεσα στις λέξεις, μια μορφή ομιλίας που λέγεται **μεμονωμένη ομιλία** (isolated speech). Αυτή η μορφή ομιλίας είναι αφύσικη και δύσχρηστη. Αντίθετα, οι σύγχρονοι αλγόριθμοι αναγνώρισης φωνής επιτρέπουν στους χρήστες να μιλούν με φυσικό και συνεχόμενο τρόπο, μια μορφή ομιλίας που λέγεται **συνεχής ομιλία** (continuous

speech), στην οποία οι λέξεις ακολουθούν η μια την άλλη χωρίς κάποια παύση ανάμεσά τους. Επίσης στα σύγχρονα συστήματα αναγνώρισης, δεν επιστρέφεται μόνο η λέξη με την μεγαλύτερη πιθανότητα επιτυχούς αναγνώρισης, αλλά μια λίστα με τις πιο πιθανές λέξεις που μπορεί να έχει εισάγει ο χρήστης. Έτσι αν αστοχήσουμε στην πρώτη μας επιλογή, μπορούμε να ξαναδοκιμάσουμε με τις υπόλοιπες καταχωρήσεις της λίστας, κάτι που είναι πολύ φυσικό και όμοιο με την ανθρώπινη συμπεριφορά. Μια άλλη επιλογή που έχει δώσει μια μεγαλύτερη φυσικότητα και ευελιξία στις εφαρμογές επεξεργασίας φωνής, είναι η δυνατότητα που παρέχεται στον χρήστη να μιλούν την ώρα που μιλά και ο υπολογιστής, διακόπτοντάς τον. Η τεχνική αυτή ονομάζεται **παρεμβολή** (barge in). Παρόλο που θεωρείται αγένεια να διακόπτεις κάποιον όταν μιλάει, ο υπολογιστής δεν προσβάλλεται και επιπλέον αυτό επιταχύνει τον διάλογο κάνοντας τις εργασίες του χρήστη πιο γρήγορες.

Τέλος, είναι πλέον εφικτή η ανάπτυξη πολύπλοκων γραμματικών, οι οποίες περιγράφουν τις δυνατές εισόδους του χρήστη σε κάθε σημείο της εφαρμογής. Έτσι και ο λόγος του χρήστη γίνεται πιο φυσικός και ο υπολογιστής μπορεί να καταλαβαίνει καλύτερα, με αποτέλεσμα να αποφεύγονται οι παρανοήσεις. [7]

Βελτιωμένες στρατηγικές σχεδιασμού διαλόγων

Η σχεδίαση αποτελεσματικών διαλόγων μπορεί να μετριάσει σε σημαντικό βαθμό τις αδυναμίες που έχουν τα συστήματα αναγνώρισης. Για αυτό απαιτείται προσεχτικός σχεδιασμός των διαλόγων, που αντανakλούν τον τρόπο με τον οποίο σύστημα και χρήστης ανταλλάσσουν πληροφορίες και αιτήσεις. Στους application-directed διαλόγους οι αποκρίσεις του χρήστη περιορίζονται σε ένα συγκεκριμένο πλαίσιο λέξεων και σύντομων φράσεων για να διευκολύνουν το έργο του αναγνωριστή που δεν είναι αναγκασμένος να κάνει αναζήτηση σε κάποιο εκτεταμένο λεξιλόγιο. Στους mixed-initiative διαλόγους, δίνεται η δυνατότητα τόσο στον χρήστη όσο και στο σύστημα να απευθύνει ερωτήσεις και να κατευθύνει την συζήτηση. [7]

Εκτεταμένη χρήση συσκευών τηλεφώνου

Η χρήση τηλεφώνου σε παγκόσμιο επίπεδο αυξάνει με ταχείς ρυθμούς ενώ ήδη τα κινητά τηλέφωνα έχουν κατακλύσει την καθημερινή ζωή στις ανεπτυγμένες

κοινωνίες. Η ευρεία διαθεσιμότητα των συσκευών τηλεφωνίας, επιτρέπει την πρόσβαση στο διαδίκτυο από όλους και από οπουδήποτε. Και από τη στιγμή που η φωνή είναι ο παραδοσιακός και πιο φυσικός τρόπος επικοινωνίας στις συσκευές αυτές, καταλαβαίνουμε πόσο επιτακτική είναι η ανάγκη υλοποίησης αποτελεσματικών φωνητικών εφαρμογών.

Στο πρώτο κεφάλαιο αυτής της εργασίας θέτουμε τις προδιαγραφές και τις απαιτήσεις της υπηρεσίας που θα υλοποιήσουμε ενώ θα γίνει και μια σύντομη περιγραφή της δομής της. Στο δεύτερο κεφάλαιο αναλύουμε τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος, όπως είναι οι τεχνολογίες αναγνώρισης και σύνθεσης φωνής, συστήματα διαλόγων καθώς και τις βασικές αρχές ανάπτυξης φωνητικών εφαρμογών. Στο τρίτο κεφάλαιο αναλύουμε τις τεχνικές και τεχνολογίες που χρησιμοποιήθηκαν για το στάδιο της υλοποίησης της εφαρμογής. Δίνεται το βασικό σχέδιο της εφαρμογής καθώς και κομμάτια στα οποία αυτή είναι οργανωμένη. Στο τρίτο κεφάλαιο παρουσιάζονται τα αποτελέσματα από την χρήση του συστήματος ύστερα και από τις απαραίτητες δοκιμές χρηστικότητας για την διόρθωση διάφορων προβλημάτων και την ρύθμιση των παραμέτρων αναγνώρισης. Στο τέταρτο κεφάλαιο παρουσιάζονται τα αποτελέσματα της εργασίας καθώς και η επεκτασιμότητα της εργασίας αυτής. Στο Appendix A – Dialog Specification υπάρχουν τα dialog specifications της υπηρεσίας. Στο Appendix B – Usability Questionnaire δίνεται το ερωτηματολόγιο που χρησιμοποιήσαμε κατά τις δοκιμές χρηστικότητας. Στο Appendix C – Technical Issues παρατίθενται διάφορες τεχνικές λεπτομέρειες που αφορούν θέματα υλοποίησης. [7]

Κεφάλαιο 1:

1. Προδιαγραφές Φωνητικής Πύλης για το Πολυτεχνείο Κρήτης

Σκοπός αυτής της μελέτης είναι η προώθηση στον εκπαιδευτικό τομέα της προσέγγισης ανάκτησης πληροφοριών με την χρήση φωνής. Η μελέτη αυτή παρουσιάζει μια εφαρμογή σχεδιασμένη σε μια γλώσσα της μορφής, Voice Extensible Markup Language (VoiceXML) και η οποία εξυπηρετεί την αλληλεπίδραση με μια βάση πληροφοριών σχετική με του φοιτητές του ακαδημαϊκού ιδρύματος του Πολυτεχνείου Κρήτης. Προβλέπει την υλοποίηση και σύνδεση μιας βάσης δεδομένων, ενός Web-based συστήματος και ενός φωνητικού συστήματος ως μέρη της μελέτης. Το Web-based σύστημα επιτρέπει στους καθηγητές και άλλους διαχειριστές του συστήματος να εισάγουν και να τροποποιούν δεδομένα σχετικά με φοιτητές και θέματα που τους αφορούν, όπως μαθήματα και προγράμματα μαθημάτων, και να δίνουν στους εμπλεκόμενους με την εφαρμογή χρήστες (καθηγητές και φοιτητές) τον απαραίτητο κωδικό πρόσβασης για την εισαγωγή τους στο σύστημα. Η φωνητική εφαρμογή παρέχει την δυνατότητα σε όλους τους χρήστες να κάνουν ανάκτηση των δεδομένων της βάσης χρησιμοποιώντας μια απλή τηλεφωνική συσκευή. Η εφαρμογή έχει την δυνατότητα να παρέχει και προσωποποιημένες πληροφορίες στους φοιτητές, όπως είναι οι βαθμολογίες μαθημάτων και εργαστηρίων, ενώ προσφέρει μια φιλική και εύκολα προσπελάσιμη φωνητική διεπαφή στους χρήστες του συστήματος και τους επιτρέπει να κάνουν αναζήτηση των πληροφοριών που αυτή προσφέρει.

1.1 Απαιτήσεις Συστήματος Web-based (HTML)

Μέσα από την χρήση του Web-based συστήματος γίνεται η διατήρηση και ενημέρωση της βάσης δεδομένων καθώς και διάφορες άλλες λειτουργίες για την σωστή λειτουργία της εφαρμογής. Στην ουσία το Web-based κομμάτι της εφαρμογής αφορά τρία μικρότερα μέρη, καθώς πρέπει να υπάρχει ένα υποσύστημα για κάθε κατηγορία χρηστών. Αυτοί είναι οι φοιτητές, οι καθηγητές καθώς και οι διαχειριστές.

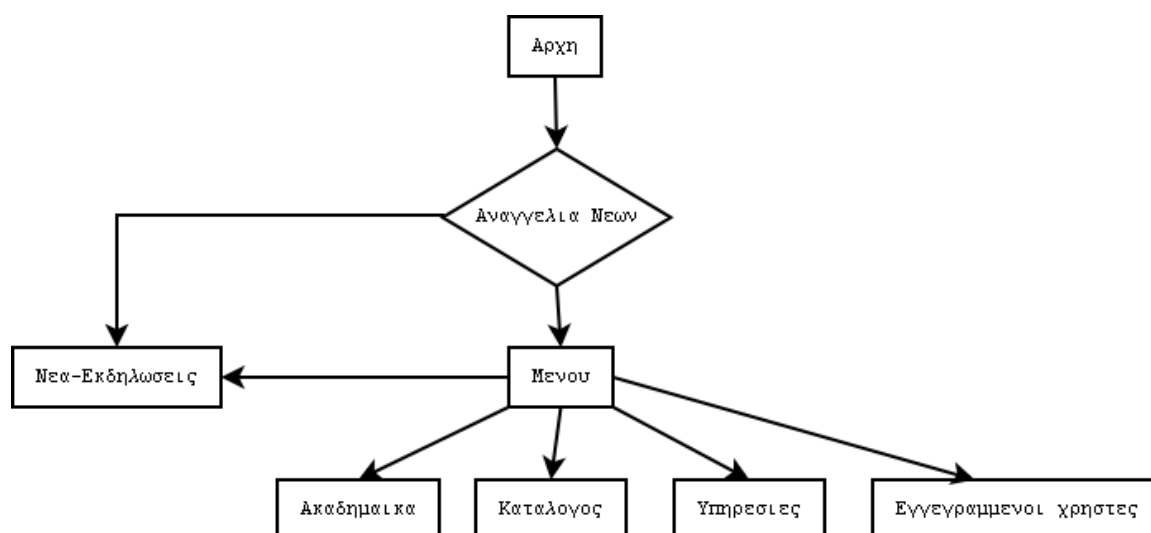
- Οι φοιτητές χρειάζονται μια γραφική διεπαφή για να μπορούν να εγγράφονται στα μαθήματα και εργαστήρια για τα οποία θέλουν να έχουν προσωποποιημένη πληροφορία, δηλαδή να λαμβάνουν φωνητικά μηνύματα και να ενημερώνονται για τα αποτελέσματα βαθμολογιών σχετικά με αυτά. Έχουν επίσης την ευχέρεια να δηλώνουν όποια μαθήματα ή εργαστήρια θέλουν, ενώ μπορούν επίσης να έχουν πρόσβαση σε μια λίστα με τις δηλώσεις που έχουν κάνει και να την τροποποιούν προσθέτοντας ή αφαιρώντας μαθήματα ή εργαστήρια. Επίσης δίνεται η δυνατότητα στους χρήστες, μέσα από από το υποσύστημα, να έχουν μια γραφική παρουσίαση του προσωποποιημένου προγράμματος μαθημάτων και εργαστηρίων που έχουν δηλώσει.
- Οι καθηγητές μπορούν χρησιμοποιώντας τον κωδικό, που τους παρέχεται από τους διαχειριστές του συστήματος, να εισέρχονται στο σύστημα και να ενημερώνουν βαθμολογίες μαθημάτων και εργαστηρίων στα οποία διδάσκουν.
- Τέλος οι διαχειριστές του συστήματος έχουν την ευθύνη για όλες τις άλλες εργασίες που είναι απαραίτητες για την σωστή λειτουργία του συστήματος. Οι εργασίες αυτές είναι εισαγωγές, ενημερώσεις και διαγραφές στις διάφορες ενότητες πληροφορίας της βάσης οι οποίες είναι οι φοιτητές, οι καθηγητές, οι ανακοινώσεις, οι καταχωρήσεις του τηλεφωνικού καταλόγου καθώς και των γιατρών, τα μαθήματα και τα εργαστήρια, το εβδομαδιαίο πρόγραμμα και το πρόγραμμα εξεταστικής.

1.2 Απαιτήσεις Φωνητικού Συστήματος

Το σύστημα ανήκει στην κατηγορία των συστημάτων πελάτη – εξυπηρετητή (client server systems). Ο χρήστης της υπηρεσίας συνδέεται με αυτήν χρησιμοποιώντας το σταθερό ή το κινητό του τηλέφωνο. Η αλληλεπίδραση με την υπηρεσία γίνεται τόσο με τη χρήση φωνής και σε κάποιες περιπτώσεις (εισαγωγή κωδικού) με χρήση DTMF εισόδου. Οι λειτουργικότητες που παρέχονται σε ένα ανώτερο επίπεδο είναι η ενημέρωση του χρήστη για ανακοινώσεις γενικού ενδιαφέροντος, ακαδημαϊκά θέματα, κατάλογος υπηρεσιών και προσώπων καθώς και διάφορες υπηρεσίες του πολυτεχνείου. Επίσης δίνεται η δυνατότητα στον χρήστη να μπει στην εφαρμογή ως εγγεγραμμένος χρήστης, ο οποίος μπορεί να είναι απλός user(π.χ. φοιτητής) ή και super user(π.χ. καθηγητής).

Οι πληροφορίες που δίνονται στον χρήστη εξάγονται από το σύστημα μετά από μία σειρά ερωταποκρίσεων μεταξύ χρήστη και συστήματος, κατά την οποία καθορίζονται παράμετροι ανάλογα με το είδος της πληροφορίας που πρέπει να εξαχθεί από τη βάση.

Δομή Βασικού Κορμού.



Μόλις ο χρήστης εισέρχεται στην υπηρεσία υπάρχει ανακοίνωση χαιρετισμού του χρηστή και κατατοπιστικό μήνυμα σχετικά με την λειτουργία της εφαρμογής. Στη συνέχεια ο χρήστης ενημερώνεται για την ύπαρξη ανακοινώσεων γενικού ενδιαφέροντος που απευθύνονται σε όλους τους φορείς του πολυτεχνείου και τους φοιτητές. Οι προηχογραφημένες αυτές ανακοινώσεις εισάγονται στο σύστημα από τον διαχειριστή του συστήματος, και αφορούν νέα και διάφορες εκδηλώσεις στα πλαίσια των δραστηριοτήτων του πολυτεχνείου. Ο χρήστης μπορεί να επιλέξει είτε να ακούσει τις ανακοινώσεις ή να προχωρήσει στην εφαρμογή. Αν επιλέξει να ακούσει τις ανακοινώσεις, ενημερώνεται για τα θέματά τους και έχει τη δυνατότητα να ακούσει ή να προσπεράσει τις ανακοινώσεις.

Στο επόμενο στάδιο της εφαρμογής το σύστημα παρουσιάζει στον χρήστη τις κύριες ενότητες που υποστηρίζει η εφαρμογή οι οποίες είναι οι εξής:

- Νέα – Εκδηλώσεις
- Ακαδημαϊκά θέματα
- Κατάλογος υπηρεσιών και προσώπων
- Διάφορες Υπηρεσίες του Πολυτεχνείου
- Εγγεγραμμένοι Χρήστες

Στην ενότητα Ακαδημαϊκά ο χρήστης μπορεί να βρει πληροφορίες για ανακοινώσεις πάνω σε:

- Πρόγραμμα εξεταστικής –μαθημάτων –εργαστηρίων
- Αποτελέσματα των εξετάσεων στα διάφορα μαθήματα και εργαστήρια

Ο χρήστης μπορεί να δώσει διάφορα στοιχεία για αναζήτηση στο σύστημα, όπως ημέρα ή ημερομηνία (σχετική ή απόλυτη), τμήμα και έτος φοίτησης, ώρα ή και όνομα μαθήματος ή εργαστηρίου. Τέλος δίνεται η δυνατότητα στον χρήστη να ενημερώνεται για τα αποτελέσματα των εξετάσεων στα διάφορα μαθήματα και εργαστήρια. Σε αυτήν την περίπτωση είναι υποχρεωτική ταυτοποίηση του χρήστη η οποία γίνεται με κωδικό πρόσβασης τον αριθμό μητρώου του φοιτητή (προμηθεύεται ο χρήστης από τον διαχειριστή του συστήματος).

Στην ενότητα τηλεφωνικός κατάλογος υπηρεσιών και προσώπων υπάρχει πλήρης κατάλογος με όλους τους φορείς του πολυτεχνείου περιλαμβάνοντας:

- Πρόσωπα, όπως καθηγητές, βοηθούς μαθημάτων, υπεύθυνους εργαστηρίων κ. λ. π
- Διάφορες υπηρεσίες του πολυτεχνείου, όπως γραμματείες, βιβλιοθήκη, εργαστήρια κτλ.

Το σύστημα θα έχει την δυνατότητα, για κάθε πρόσωπο που αναζητείται από τον χρήστη, να παρέχει το τηλέφωνο, (στο οποίο ο χρήστης θα μπορεί να προωθεί την κλήση του αν το επιθυμεί). Για κάθε υπηρεσία που αναζητείται, ο χρήστης θα έχει πρόσβαση σε κάποιο τηλέφωνο στο οποίο ο χρήστης θα μπορεί επίσης να προωθηθεί. Στην αναζήτηση προσώπων ο χρήστης έχει την δυνατότητα να αναζητεί με βάση το όνομα και το επίθετο ή και μόνο βάσει του επιθέτου.

Όταν ο χρήστης επιλέγει από το κεντρικό μενού την κατηγορία υπηρεσίες, το σύστημα του εμφανίζει ένα καινούργιο υπό-μενού, το μενού υπηρεσιών, όπως φαίνεται παρακάτω:

- Υπηρεσία Φωνητικού Ταχυδρομείου, και
- Υπηρεσία Αναζήτησης Γιατρών (συμβεβλημένοι με το πολυτεχνείο).

Μέσα από την υπηρεσία φωνητικού ταχυδρομείου (Voice Mail), οι απλοί εγγεγραμμένοι χρήστες της υπηρεσίας, οι οποίοι είναι οι φοιτητές του ιδρύματος μπορούν να ακούν τα ηχογραφημένα μηνύματα που στέλνουν οι super users, τους οποίους αποτελούν οι καθηγητές. Έτσι, κάποιος καθηγητής μπορεί να επικοινωνεί με τους φοιτητές του μέσω της φωνητικής πύλης.

Στην υπηρεσία αναζήτησης γιατρών, ο χρήστης θα μπορεί να αναζητήσει κάποιον γιατρό με βάση την ειδικότητά του ή και το όνομά του. Το σύστημα θα επιστρέφει στον χρήστη πληροφορίες σχετικά με την τοποθεσία – διεύθυνση του ιατρείου καθώς και τηλέφωνα, ώρες λειτουργίας, ενώ θα μπορεί να γίνεται και προώθηση της κλήσης για την επικοινωνία του χρήστη με το ιατρείο.

Τέλος ο χρήστης μπορεί να κάνει την πλοήγηση του στο σύστημα ως εγγεγραμμένος χρήστης έτσι ώστε το σύστημα να μπορεί να κάνει πιο γρήγορα (δεν χρειάζεται να εισάγει κάθε φορά εξάμηνο και τμήμα) και με ασφάλεια ανάκτηση πληροφορίας που αφορούν τον χρήστη. Η εγγραφή των χρηστών στο σύστημα πρέπει να έχει γίνει προηγουμένως από τον διαχειριστή του συστήματος (και έτσι οι χρήστες προμηθεύονται κάποιο όνομα χρήστη και κωδικό πρόσβασης.) Χρησιμοποιώντας αυτά τα στοιχεία ο χρήστης εισάγεται στο σύστημα. Έτσι το σύστημα μπορεί να ενημερώσει τον χρήστη αμέσως για πράγματα που τον αφορούν, όπως για παράδειγμα την ύπαρξη νέου μηνύματος στο voice mail, οπότε τον καλεί να μεταβεί στην υπηρεσία Voice Mail για να διαβάσει τα μηνύματα.

Κεφάλαιο 3:

2. Τεχνολογίες Επεξεργασίας Φωνής

Οι βασικές τεχνολογίες φωνής που χρησιμοποιήσαμε για την ανάπτυξη της φωνητικής εφαρμογής ήταν οι τεχνολογίες αναγνώρισης και σύνθεσης φωνής καθώς και τεχνολογίες διαχείρισης διαλόγων. Επίσης περιγράφουμε και τις βασικές αρχές πάνω στις οποίες βασίστηκε η σχεδίαση της διαδραστικής επικοινωνίας του συστήματος.

2.1 Αναγνώριση Φωνής

Όπως αναφέραμε και στην εισαγωγή της εργασίας, τα παλιότερα συστήματα αναγνώρισης φωνής επέτρεπαν την αναγνώριση μόνο ενός μικρού αριθμού ομιλητών, οι οποίοι έπρεπε να μιλούν χρησιμοποιώντας εκτεταμένες παύσεις ανάμεσα στις λέξεις. Πλέον, στα σύγχρονα συστήματα αναγνώρισης φωνής επιτρέπεται η συνεχόμενη και με φυσικό τρόπο ομιλία, ενώ υπάρχει η δυνατότητα αναγνώρισης ενός μεγάλου φάσματος φωνών. Τα συστήματα αυτά χρησιμοποιούν στατιστικές μεθόδους αναγνώρισης και μια μεγάλη ποικιλία τεχνικών.

Η αναγνώριση ξεκινάει με την front-end επεξεργασία του ψηφιακοποιημένου σήματος. Οι περισσότερες μέθοδοι χρησιμοποιούν φασματική ανάλυση και οι πιο διαδεδομένες είναι η Ανάλυση Γραμμικής Πρόβλεψης (LPC) η Εξαγωγή Mel-Frequency Cepstral Coefficients (MFCC) και η μοντελοποίηση κοχλίας. Στην συνέχεια λαμβάνει χώρα η αναγνώριση φωνημάτων, ακολουθιών φωνημάτων και λέξεων. Για την λειτουργία αυτή υπάρχουν διάφορες προσεγγίσεις όπως η χρήση Dynamic Time Warping (DTW), Artificial Neural Networks (ANN), expert systems η και συνδυασμό αυτών. Ωστόσο, σήμερα, οι πιο διαδεδομένες και αποτελεσματικές μέθοδοι αναγνώρισης είναι βασισμένες σε Hidden Markov Models (HMMs). [8]

Το σύστημα της Nuance Communications που χρησιμοποιήσαμε στην υλοποίηση της φωνητικής εφαρμογής, χρησιμοποιεί εξελιγμένα HMMs, που αναπτύχθηκαν στο ινστιτούτο SRI και παρέχουν υψηλές επιδόσεις σε εφαρμογές που

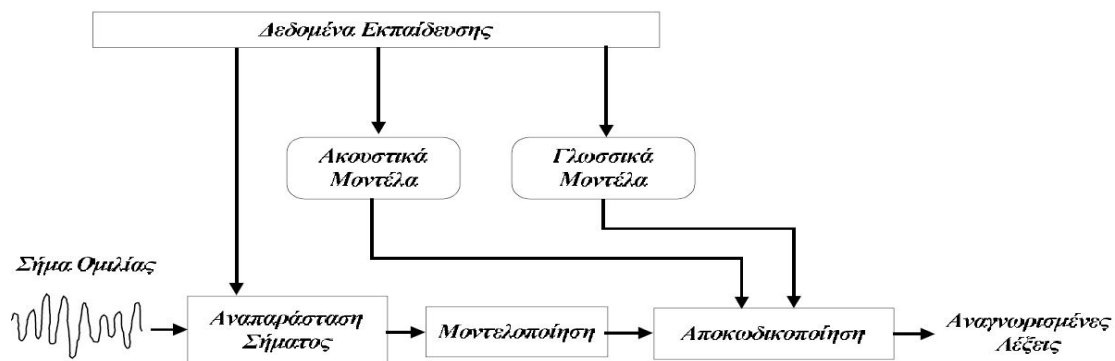
λειτουργούν ανεξάρτητα από τον ομιλητή, χωρίς εκπαίδευση, συνεχούς ομιλίας και μεγάλων λεξιλογίων.

Για να υλοποιηθεί ένα σύστημα αναγνώρισης φωνής με την χρήση HMMs πρέπει να επιλύσουμε τα παρακάτω προβλήματα:

- *Παραμετροποίηση front-end.* Σε αυτό το στάδιο είναι δυνατή η εξαγωγή της απαραίτητης πληροφορίας από την κυματομορφή ομιλίας, με τέτοιο τρόπο που να είναι συμβατά με τα μοντέλα που είναι βασισμένα σε HMMs. Βασική λειτουργία αυτής της φάσης της αναγνώρισης είναι να διαιρέσει το φωνητικό σήμα εισόδου σε frames κάποιων msec και να βγάλει από το καθένα ένα εξομαλυσμένο φάσμα, τα φασματικά χαρακτηριστικά του οποίου θα είναι σταθερά για το διάστημα αυτό (locally stationary). Για την λειτουργία αυτή συνήθως εφαρμόζεται μια συνάρτηση παραθύρου με απόσβεση στα άκρα (π.χ. ένα φίλτρο Hamming) σε κάθε τέτοιο frame. Επίσης, το σήμα φωνής πολλές φορές υφίσταται προενίσχυση υψηλής συχνότητας, προκειμένου να αντισταθμιστεί η εξασθένιση ακτινοβολίας που προκαλείται από τα χείλη του ομιλητή.
- *Σχεδιασμός ακουστικών μοντέλων.* Σε αυτό το στάδιο γίνεται η ακριβής αναπαράσταση των κατανομών του κάθε ήχου-φωνήματος από τα HMMs, έτσι ώστε να καλύπτεται καθένα από τα συμφραζόμενα, που μπορεί να συνοδεύουν τον ήχο. Οι παράμετροι των HMMs υπολογίζονται από δεδομένα εκπαίδευσης. Για πρακτικούς λόγους προτιμάται η αποσύνθεση των ακολουθιών λέξεων σε ακολουθίες βασικών ήχων, τα φωνήματα. Δεν πρέπει όμως να θεωρούμε ότι αρκεί να εκπαιδεύσουμε έναν αριθμό HMMs ίσο με τον αριθμό των φωνημάτων για κάθε γλώσσα. Και αυτό γιατί η επίδραση από τα συμφραζόμενα φωνήματα μπορεί να κάνει τα φωνήματα να ηχήσουν με πολύ διαφορετικό τρόπο σε κάθε περίπτωση. Άρα, θα πρέπει να εκπαιδευτούν διαφορετικά HMMs για κάθε φορά που έχουμε διαφορετικά συμφραζόμενα στο ίδιο φώνημα. Η πιο διαδεδομένη προσέγγιση είναι η χρήση των triphones όπου κάθε φώνημα έχει ένα διαφορετικό HMM μοντέλο ανάλογα με το ζευγάρι φωνημάτων που έχει δεξιά και αριστερά του.

- *Σχεδιασμός του γλωσσικού μοντέλου.* Το τελευταίο στάδιο επεξεργασίας ενός συστήματος αναγνώρισης είναι η αντιστοίχιση των ακολουθιών φωνημάτων σε ομιλούμενες λέξεις. Στην ουσία παράγουμε ένα μηχανισμό υπολογισμού της πιθανότητας εμφάνισης κάποιας λέξης σε μια έκφραση, δεδομένων των προηγούμενων λέξεων. Το γλωσσικό μοντέλο περιορίζει την αναγνώριση στα πλαίσια των λέξεων και εκφράσεων που μπορεί να δεχτεί η εφαρμογή. Αυτό επιτυγχάνεται με την χρήση των Μαρκοβιανών γραμματικών N-οστής τάξης (N-gramms) στις οποίες θεωρούμε ότι η κάθε λέξη σε μια φράση εξαρτάται μόνο από τις προηγούμενες N-1 λέξεις.
- *Αποκωδικοποίηση.* Μετά την επεξεργασία που περιγράψαμε προηγουμένως, το σύστημα προχωρά στην επιλογή της πιο πιθανής ακολουθίας λέξεων, που αποτελεί και το τελικό αποτέλεσμα αναγνώρισης.

Η διαδικασία αναγνώρισης μπορεί να αναπαρασταθεί γραφικά από το Σχήμα 1. Διαδικασία Αναγνώρισης. [8]



Σχήμα 1. Διαδικασία Αναγνώρισης

2.2 Σύνθεση Φωνής

Η σύνθεση φωνής ή σύνθεση ομιλίας (text-to-speech synthesis), αποτελεί το κομμάτι αυτό των τεχνολογιών φωνής που ασχολείται με την αυτόματη παραγωγή ομιλίας από κείμενο. Είναι γεγονός, ότι τα τελευταία χρόνια η εξέλιξη στον τομέα αυτόν ήταν ραγδαία, με αποτέλεσμα η ποιότητα της παραγόμενης φωνής να πλησιάζει αρκετά με αυτήν της φυσικής ομιλίας.

Μέχρι πρόσφατα οι επιστήμονες επικεντρώνονταν στην υλοποίηση συστημάτων, τα οποία προσπαθούσαν με διάφορους αλγόριθμους και τεχνικές να παράγουν φωνή βασιζόμενοι στους φυσικούς μηχανισμούς που χρησιμοποιεί ο άνθρωπος στην ομιλία του. Δυστυχώς όμως οι γνώσεις μας πάνω σε αυτούς τους φυσικούς μηχανισμούς της ανθρώπινης ομιλίας ήταν και παραμένουν αρκετά περιορισμένες, με αποτέλεσμα η παραγόμενη ομιλία, από τα συστήματα αυτά, να είναι αρκετά χαμηλή ποιοτικά.

Μια νέα προσέγγιση έμελλε να φέρει την επανάσταση στις τεχνολογίες σύνθεσης φωνής, και βασιζόταν στην ιδέα της αντιγραφής της ανθρώπινης ομιλίας αντί της προσπάθειας μίμησης των αρχών παραγωγής φωνής από τον άνθρωπο που κυριαρχούσε μέχρι εκείνη την στιγμή. Αυτό σημαίνει ότι, αντί να γίνεται προσπάθεια παραγωγής φωνητικών σημάτων μέσα από την τροποποίηση βασικών σημάτων με την χρήση εξισώσεων που προσομοιώνουν το ανθρώπινο σύστημα παραγωγής φωνής, θα μπορούσαμε να φτάσουμε στο επιθυμητό αποτέλεσμα μέσα από την χρήση προηχογραφημένων πραγματικών φωνητικών δειγμάτων των ήχων που θέλουμε να παράγουμε. Οπότε η διαδικασία σύνθεσης κάποιας λέξης ή πρότασης οδηγείται στην συγκριτικά πιο απλή διαδικασία της επιλογής της κατάλληλης αλληλουχίας ήχων και διασύνδεσης τους σε μια τελική κυματομορφή.

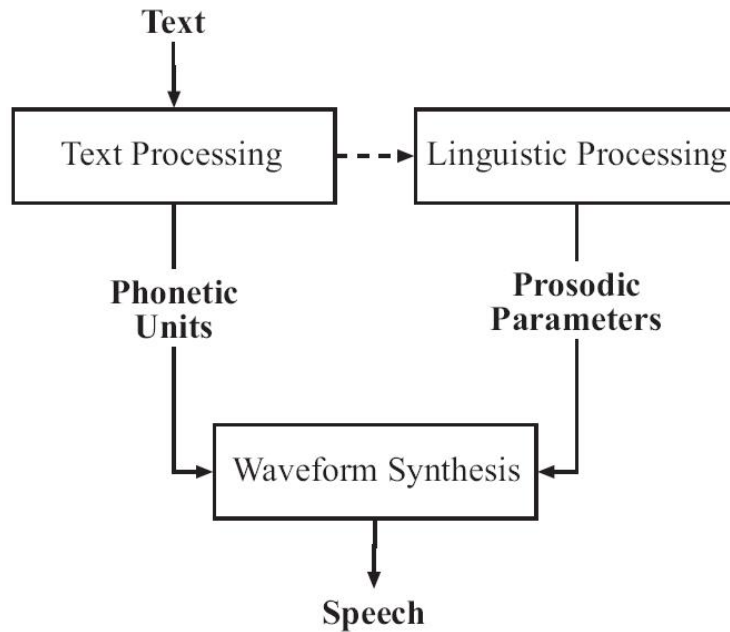
Η υψηλή ποιότητα, στην οποία έχουν φτάσει τα σημερινά συστήματα σύνθεσης φωνής (speech synthesizers), επιτρέπει την χρήση τους σε διάφορες εφαρμογές φωνής όπως είναι τα τηλεφωνικά κέντρα και οι φωνητικές εφαρμογές ανάκτησης πληροφοριών (Voice Portals). Παρόλα αυτά, η έρευνα στον τομέα της σύνθεσης φωνής συνεχίζεται με εντατικούς ρυθμούς. Η ποιότητα της συνθετικής

φωνής σήμερα μπορεί να έχει φτάσει σε υψηλά επίπεδα, ωστόσο είναι πολύ μακριά από το σημείο όπου θα είναι δυνατή η σύγκρισή της με την ανθρώπινη ομιλία.

Η διαδικασία σύνθεσης φωνής, έτσι όπως εμφανίζεται στα περισσότερα συστήματα σύνθεσης φωνής, αποτελείται από τα παρακάτω τρία σκέλη [9]:

- *Επεξεργασία κειμένου (text processing)*, το οποίο περιλαμβάνει ανάλυση των προτάσεων(sentence parsing), για να καταλάβουμε τον τρόπο με τον οποίο θα πρέπει να διαβαστεί η κάθε πρόταση- π.χ. αν είναι ερώτηση ή κατάφαση-, κανονικοποίηση κειμένου (text normalization), που έχει να κάνει με τις διαφοροποιήσεις που έχει ο γραπτός με τον προφορικό λόγο- π.χ. ενώ γράφουμε κ. Διγαλάκης προφέρουμε κύριος Διγαλάκης-, παραγωγή εκφράσεων και προφοράς, (phrasing and pronunciation generation), όπου μετατρέπουμε κάθε λέξη που πρόκειται να συντεθεί σε μια ακολουθία φωνημάτων, μια διαδικασία που γίνεται με την χρήση κάποιου λεξικού προφορών καθώς σε πολλές γλώσσες μπορεί να εμφανιστούν συγχύσεις κατά την διάρκεια αντιστοίχισης λέξης σε φώνημα.
- *Γλωσσολογική επεξεργασία (linguistic processing)*, όπου το σύστημα χρησιμοποιεί προσωδιακά μοντέλα (prosodic models) για την πρόβλεψη εκφοράς, τονισμού και χρονισμού ανάμεσα στα διάφορα τμήματα του λόγου (segmental timing). Η σωστή προσωδία δίνει μεγαλύτερη φυσικότητα στον παραγόμενο λόγο, βοηθώντας στην καλύτερη κατανόηση από την πλευρά του ακροατή, πράγμα που σημαίνει και λιγότερη κόπωση.
- *Σύνθεση τελικής κυματομορφής (waveform synthesis)*, όπου έχουμε την παραγωγή ομιλίας ύστερα από την προεπεξεργασία των δυο προηγούμενων καταστάσεων.

Η διαδικασία αυτή φαίνεται στο Σχήμα 2. Η διαδικασία σύνθεσης φωνής:



Σχήμα 2. Η διαδικασία σύνθεσης φωνής

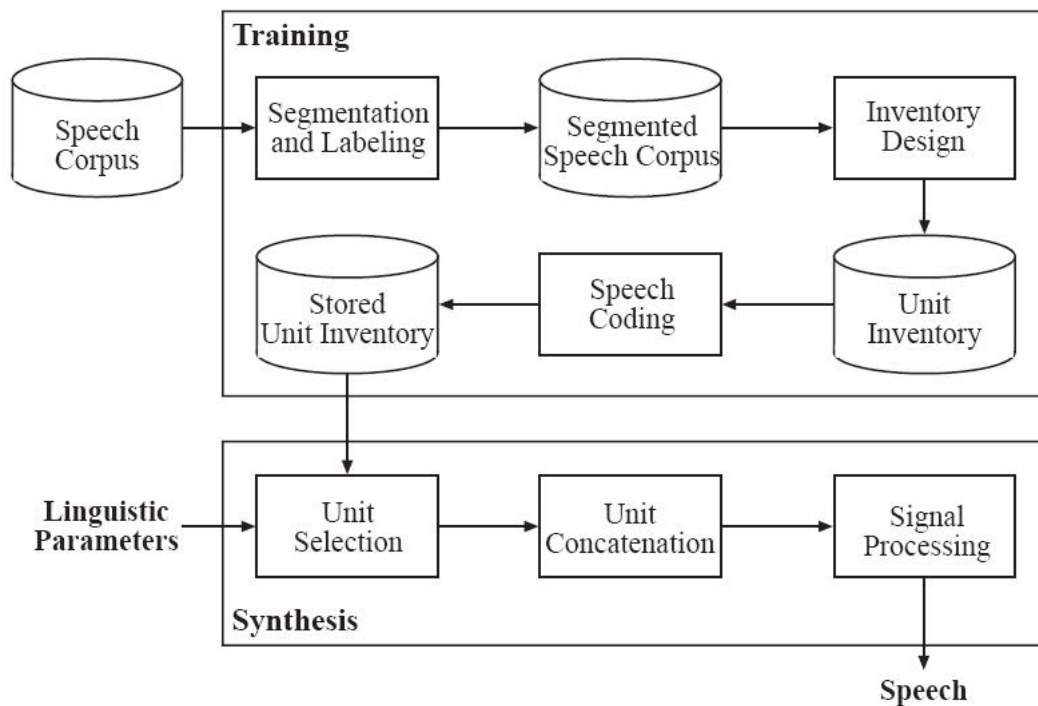
Η μηχανή σύνθεσης που χρησιμοποιεί το σύστημα, με το οποίο υλοποιήσαμε την εφαρμογή, χρησιμοποιεί concatenative synthesis, οπότε θα δούμε πως γίνεται η παραγωγή της τελικής κυματομορφής με αυτή την προσέγγιση. Αυτή αποτελείται από τα παρακάτω κομμάτια [9]:

- **Speech Corpus.** Εκεί βρίσκονται αποθηκευμένα τα synthesis units και αποτελείται από κομμάτια ομιλίας που έχουν ηχογραφηθεί από κάποιον συγκεκριμένο ομιλητή. Αποτελείται από κυματομορφές ομιλίας και τις αντίστοιχες ορθογραφικές εγγραφές.
- **Τμηματοποίηση και Ονοματοποίηση Ομιλίας (Speech Segmentation and Labeling).** Το speech corpus τμηματοποιείται σε βασικές μονάδες ομιλίας (speech units) και καθένα από αυτά παίρνει κάποια ονομασία, τόσο όσον αφορά τα φωνητικά χαρακτηριστικά όσο και τα προσωδιακά.
- **Σχεδιασμός του Ευρετηρίου Βασικών Μονάδων (Unit Inventory Design).** Στο κομμάτι αυτό, σκοπός είναι η εξαγωγή ενός συνόλου τμημάτων ομιλίας που να καλύπτουν όλες τις φωνητικές παραλλαγές μιας γλώσσας, έχοντας τις ελάχιστες ασυνέχειες στα σημεία συνένωσης και έχοντας και το μικρότερο

δυνατό μέγεθος. Το κομμάτι αυτό είναι πιο σημαντικό στην υλοποίηση του συστήματος σύνθεσης, καθώς επηρεάζει όλα τα υπόλοιπα κομμάτια.

- Κωδικοποίηση Ομιλίας και Αποθήκευση (Speech Coding and Storage). Το ευρετήριο τακτοποιείται και αποθηκεύεται με τέτοιο τρόπο ώστε να είναι προσβάσιμο κατά την διάρκεια της σύνθεσης. Μπορούμε να αποθηκεύσουμε είτε τις κυματομορφές ως έχουν είτε να κάνουμε κάποια επεξεργασία και να αποθηκεύσουμε τις παραμέτρους τους.
- Επιλογή Βασικών Μονάδων (Unit Selection). Σε αυτό το κομμάτι επιλέγονται τα καλύτερα units για την παραγωγή του ήχου που επιθυμούμε. Η επιλογή γίνεται με φωνητικά και προσωδιακά κριτήρια.
- Συνένωση Βασικών Μονάδων (Unit Concatenation). Αφού επιλέξουμε τις καλύτερες βασικές μονάδες για την σύνθεση της λέξης ή φράσης που επιθυμούμε, γίνεται η συνένωσή τους. Η μετάβαση από την μια βασική μονάδα στην άλλη πρέπει να είναι όσο γίνεται πιο ομαλή, κάτι το οποίο επιτυγχάνεται ελαχιστοποιώντας τις ασυνέχειες στα άκρα των μονάδων.
- Προσωδιακή Επεξεργασία Σήματος (Prosodic Signal Processing). Πέρα από την σωστή ακολουθία των φωνημάτων, πρέπει να έχουμε και την κατάλληλη προσωδία. Πολλά συστήματα προχωρούν σε διάφορες επεξεργασίες σήματος σχετικά με την διάρκεια ή την ενέργεια, με αποτέλεσμα πολλές φορές να υπάρχει πρόβλημα με την φυσικότητα του λόγου. Μια καλή πρακτική είναι να έχουμε ένα αρκετά πλούσιο ευρετήριο μονάδων που θα προβλέπουν την αλλαγή στην προσωδία.

Η σειρά με την οποία αυτά τα κομμάτια συμμετέχουν στην σύνθεση φαίνεται στο Σχήμα 3. Σύνθεση Φωνής:



Σχήμα 3. Σύνθεση Φωνής

2.3 Συστήματα Διαλόγου

Παρά την αλματώδη πρόοδο των τεχνολογιών φωνής και την γενική αποδοχή των αντίστοιχων εφαρμογών, είναι δύσκολο ακόμα για τα φωνητικά συστήματα να παράγουν πραγματικούς διαδραστικούς διαλόγους. Δηλαδή συνομιλίες που κάνουν οι άνθρωποι σε καθημερινές συζητήσεις, ή όταν πρόκειται να διαπραγματευτούν για κάποιο θέμα ή προϊόν. Συνομιλία ονομάζουμε την διαδικασία κατά την οποία δύο άνθρωποι, ή όπως στην περίπτωση μας, ένας άνθρωπος και μια μηχανή, μιλούν σε σειρά ο ένας μετά τον άλλον πάνω σε κάποιο θέμα. Υπάρχουν τρία διαδορετικά μοντέλα συνομιλίας μεταξύ ανθρώπου και μηχανής:

- *Διάλογοι ομιλίας καθοδηγούμενοι από τον χρήστη* (human-driven conversational dialogs). Σε αυτήν την περίπτωση ο χρήστης κάνει

διαδοχικές ερωτήσεις ή δίνει εντολές και το σύστημα ανταποκρίνεται ανάλογα.

- *Διάλογοι ομιλίας καθοδηγούμενοι από το σύστημα* (application-driven conversational dialogs). Το σύστημα κάνει διαδοχικές ερωτήσεις για να εκμαιεύσει τις απαντήσεις και τις εντολές από τον χρήστη.
- *Διάλογοι mixed-initiative*. Σε αυτή την περίπτωση έχουμε συνδυασμό των δύο προηγούμενων περιπτώσεων. Ο χρήστης και ο υπολογιστής μπορούν να παίρνουν την πρωτοβουλία να καθοδηγήσουν την συζήτηση κάνοντας ερωτήσεις.

Σήμερα η χρήση των εφαρμογών που κάνουν χρήση των διαλόγων application-driven είναι αρκετά διαδεδομένη, και ιδιαίτερα στις φωνητικές εφαρμογές όπου ζητείται από τον χρήστη να απαντήσει σε διαδοχικές ερωτήσεις του συστήματος χρησιμοποιώντας μια απλή λέξη ή φράση. Αυτό δίνει αρκετά πλεονεκτήματα στην υλοποίηση. Αφήνει τον χρήστη να επικεντρωθεί σε μια συγκεκριμένη απάντηση, αντί να μπει στην διαδικασία να μαντέψει πως θα πρέπει να ενημερώσει το σύστημα για την αίτηση που θέλει να προωθήσει. Επίσης επιτρέπει στον αναγνωριστή να αναγνωρίσει ένα μικρό σύνολο λέξεων κάθε φορά, και όχι όλες τις απαιτούμενες λέξεις μέσα σε κάποιο μεγάλο σύνολο λέξεων.

Ο διαχειριστής διαλόγων (dialog manager) έχει την ευθύνη να διαχειρίζεται τους διαλόγους και την πληροφορία που ανταλλάσσεται ανάμεσα στον χρήστη και στην φωνητική εφαρμογή. Αυτή την στιγμή η ανάπτυξη εφαρμογών γίνεται χρησιμοποιώντας VoiceXML browsers για την διαχείριση των διαλόγων. Ωστόσο, γίνεται έρευνα για την ανάπτυξη εξεζητημένων διαχειριστών διαλόγων, που θα επιτρέπουν την ανάπτυξη, τροποποίηση και εκτέλεση σχεδίων για την επίτευξη των στόχων της συνομιλίας.

Voice Dialog Interpreters

Οι **Voice Dialog Interpreters** είναι διαχειριστές διαλόγων που έχουν τα εξής χαρακτηριστικά:

- Ένα σύστημα ηχογράφησης για την ηχογράφηση των φωνητικών εισόδων του χρήστη.
- Έναν DTMF interpreter για την αποκωδικοποίηση των σημάτων που έρχονται από τα πλήκτρα της συσκευής του χρήστη.
- Μια μηχανή αναγνώρισης ομιλίας για την είσοδο φωνητικών εντολών του χρήστη.
- Ένα σύστημα που θα παράγει προηχογραφημένη ομιλία ή και μουσικά σήματα.
- Μια μηχανή σύνθεσης φωνής για την μετατροπή κειμένου σε συνθετική φωνή για έξοδο.
- Έναν dialog manager που ελέγχει την ροή των διαλόγων.

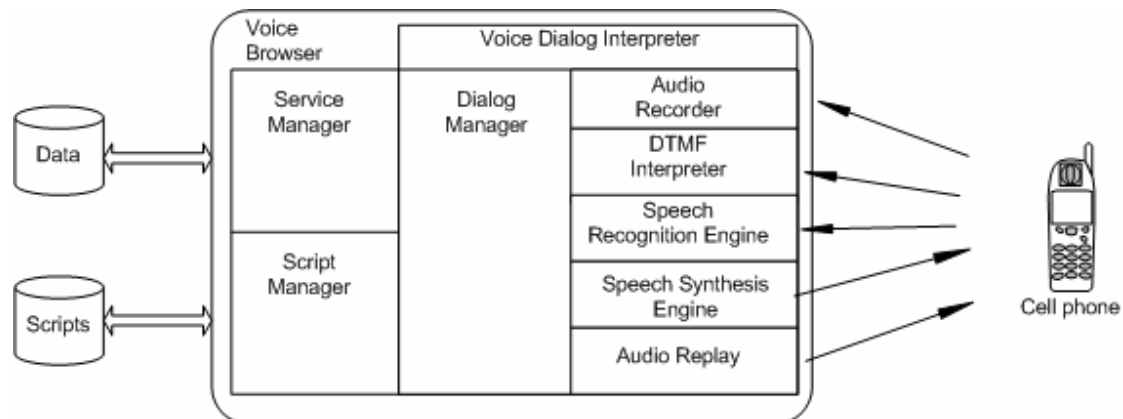
Ένας dialog interpreter εκτελεί μια συγκεκριμένη σειρά λειτουργιών η οποία αποτελείται από τα παρακάτω βήματα:

1. Παρακίνηση του χρήστη να απαντήσει σε κάποια ερώτηση ή αίτηση του συστήματος.
2. Ερμηνεία της εισόδου του χρήστη, φωνητική ή DTMF
3. Εκτέλεση των απαραίτητων ενεργειών ανάλογα με την είσοδο του χρήστη
4. Καθορισμός του επόμενου βήματος εκτέλεσης, συμπεριλαμβανομένης και της κλήσης ενός νέου αρχείου διαλόγου.

Voice Browser

Ένας voice browser [4], όπως φαίνεται και στο Σχήμα 4. Η αρχιτεκτονική του Voice Browser, αποτελείται από έναν voice dialog interpreter σε συνδυασμό με έναν document manager και έναν service manager. Ο document manager αποθηκεύει και ανακτά τα αρχεία που πρόκειται να επεξεργαστούν από τον voice dialog interpreter. Για να βελτιωθεί ο χρόνος απόκρισης του συστήματος, ο service manager αναλαμβάνει να φορτώσει όλα τα αρχεία που μπορεί να χρειαστεί ο voice dialog interpreter, έτσι ώστε να είναι διαθέσιμα όταν αυτός τα χρειαστεί. Οι voice browsers που δέχονται ως είσοδο VoiceXML αρχεία λέγονται VoiceXML browsers. Πρέπει

εδώ να παρατηρήσουμε ότι όλα αυτά τα κομμάτια που περιγράφονται στο Σχήμα 4 πρέπει να είναι εγκατεστημένα σε έναν κεντρικό υπολογιστή (server) μια και οι τηλεφωνικές συσκευές είναι κυρίως I/O συσκευές και δεν διαθέτουν το απαραίτητο hardware για να υποστηρίξουν μια τέτοια δομή. Κάθε τηλεφωνική συσκευή μπορεί να επικοινωνεί με τον κεντρικό υπολογιστή όπως θα δείξουμε παρακάτω.



Σχήμα 4. Η αρχιτεκτονική του Voice Browser

2.4 Αρχές Ανάπτυξης Εφαρμογών Φωνής

Οι εφαρμογές επεξεργασίας φωνής μοιάζουν με συζητήσεις μεταξύ ανθρώπων και ηλεκτρονικού υπολογιστή. Οι συζητήσεις μεταξύ των ανθρώπων έχουν πολυποίκιλο χαρακτήρα. Μερικές από τις ιδιότητες των συζητήσεων είναι η αναμονή σειράς ομιλίας, που σημαίνει κάθε φορά πρέπει να μιλάει μόνο ένας και όχι δύο ή περισσότεροι ταυτόχρονα, η δυνατότητα όσων συμμετέχουν στη συζήτηση να κάνουν ερωτήσεις και να κατευθύνουν την συζήτηση (shifts in initiative) και η φραστική ή όχι απόκριση για να υποδεικνύεται η κατανόηση.

Ένα μεγάλο πλεονέκτημα των εφαρμογών επεξεργασίας φωνής είναι ότι η ομιλία, είναι ένα πολύ φυσικό μέσο επικοινωνίας για τον άνθρωπο, καθώς είναι μια ιδιότητα που εξασκούν από πολύ νωρίς στη ζωή τους.

Μια αποτελεσματική εφαρμογή επεξεργασίας φωνής προσομοιώνει μερικές από αυτές τις βασικές αρχές της άνθρωπο με άνθρωπο επικοινωνίας. Επειδή η ομιλία είναι βαθιά ριζωμένη στην ανθρώπινη συμπεριφορά και δραστηριότητα, καταλαβαίνουμε ότι για την υλοποίηση αποτελεσματικών εφαρμογών φωνής είναι

απαραίτητη η μελέτη και η κατανόηση των διαφόρων τρόπων, με τους οποίους χρησιμοποιούμε την γλώσσα όταν επικοινωνούμε. Έτσι η εφαρμογή πρέπει να υιοθετήσει γλωσσικές συμβάσεις, οι οποίες βοηθούν τους ανθρώπους να ξέρουν τι θα πουν ανά πάσα στιγμή, και να αποφεύγει γλωσσικά πρότυπα τα οποία οδηγούν σε παραβιάσεις των αρχών της ευγένειας και της συνεργατικής συμπεριφοράς. Εδώ θα δούμε πότε είναι ενδεδειγμένη η χρήση φωνητικής διεπαφής H/Y(speech interface) και πότε πρέπει να αναζητήσουμε άλλο μέσο επικοινωνίας [10].

2.3.1 Φωνητική Είσοδος

Ένας καθοριστικός παράγοντας για να πετύχει μια φωνητική εφαρμογή είναι η ύπαρξη πραγματικού κέρδους από την χρήση τεχνολογιών φωνής. Επειδή η ομιλία, όπως αναφέραμε παραπάνω, είναι ένα φυσικό μέσω επικοινωνίας, οι προσδοκίες των χρηστών της εφαρμογής είναι λογικό να είναι πολύ αυξημένες. Έτσι η χρήση τεχνολογιών φωνής πρέπει να γίνεται μόνο όταν η ανάγκη είναι εμφανής, πχ τα χέρια και τα μάτια του χρήστη είναι απασχολημένα ή υπάρχουν δουλειές που δεν μπορούν να γίνουν αλλιώς, όπως είναι η πρόσβαση σε κάποιο ηλεκτρονικό ταχυδρομείο ή κάποια βάση δεδομένων όπως συμβαίνει στην περίπτωση μας.

Οι εφαρμογές φωνής μπορούν να είναι επιτυχείς μόνο όταν οι χρήστες είναι πρόθυμοι και έχουν κάποιο κίνητρο να συνεργαστούν. Ένα κλασικό παράδειγμα είναι οι αυτόματοι τηλεφωνητές. Οι χρήστες θέλουν οι κλήσεις τους να προωθηθούν και για αυτό είναι πρόθυμοι να απαντήσουν προσεχτικά σε αυτά που τους ζητάει το σύστημα. Η ομιλία ταιριάζει, και πρέπει να εφαρμοστεί σε κάποιες εργασίες, αλλά σε άλλες η χρήση της δεν είναι ενδεδειγμένη. Ο παρακάτω πίνακας δείχνει κάποια χαρακτηριστικά σύμφωνα με τα οποία μπορούμε να ορίσουμε πότε ενδείκνυται η χρήση φωνητικής εισόδου και πότε όχι.

Ενδείκνυται	Δεν Ενδείκνυται
Δεν υπάρχει πληκτρολόγιο (πχ τηλέφωνο ή κάποια άλλη φορητή συσκευή) Η εργασία απαιτεί τα χέρια του χρήστη	Η εργασία που πρέπει να εκτελεστεί απαιτεί ότι ο χρήστης θα μιλάει ταυτόχρονα και σε άλλους ανθρώπους.

<p>να είναι απασχολημένα, έτσι ώστε να μην μπορούν να χρησιμοποιήσουν το πληκτρολόγιο ή το ποντίκι</p> <p>Οι εντολές είναι ενσωματωμένες σε μια αρκετά βαθιά δομή μενού.</p> <p>Οι χρήστες αδυνατούν να πληκτρολογήσουν ή δεν αισθάνονται άνετα με αυτό.</p> <p>Αν υπάρχει κάποια μορφής αναπηρίας που δεν τους επιτρέπει είτε να χρησιμοποιήσουν τα χέρια τους ή τα μάτια τους.</p>	<p>Οι χρήστες δουλεύουν σε ένα περιβάλλον όπου υπάρχει παρουσία δυνατού θορύβου.</p> <p>Η εργασία μπορεί να εκτελεστεί ευκολότερα χρησιμοποιώντας πληκτρολόγιο και ποντίκι.</p>
--	---

Πίνακας 1 Χρήση φωνητικής εισόδου

Ενδείκνυται	Δεν ενδείκνυται
<p>Η εργασία που εκτελείται απαιτεί τα μάτια του χρήστη να είναι επικεντρωμένα σε διαφορετικό σημείο από την οθόνη του υπολογιστή. (π.χ. οδήγηση)</p> <p>Η περίσταση απαιτεί να επιστήσουμε την προσοχή του χρήστη σε κάτι.</p> <p>Οι χρήστες έχουν κάποια αναπηρία (αναπηρία όρασης)</p> <p>Όταν προσπαθούμε να δώσουμε κάποια συγκεκριμένη προσωπικότητα στην</p>	<p>Μεγάλες ποσότητες δεδομένων πρέπει να παρουσιαστούν στον χρήστη.</p> <p>Η εργασία απαιτεί από τον χρήστη να συγκρίνει τα δεδομένα</p> <p>Η πληροφορία είναι προσωπική ή απόρρητη</p>

εφαρμογή.	
-----------	--

Πίνακας 2 Χρήση φωνητικής εξόδου

Η χρήση τεχνολογιών επεξεργασίας φωνής μόνο και μόνο επειδή είναι κάτι το καινούργιο δεν εγγυάται και την επιτυχία του εγχειρήματος. Η ύπαρξη πραγματικής ανάγκης αυξάνει τις πιθανότητες επιτυχίας του.

2.3.2 Σχεδιασμός Για Φωνή – Χαρακτηριστικά και απαιτήσεις

Γενικά μια επιτυχημένη φωνητική εφαρμογή πρέπει να σχεδιάζεται έχοντας στο μυαλό την ανθρώπινη ομιλία και τις αρχές που την διέπουν. Είναι σπάνια αποτελεσματική η πρόσθεση φωνής σε εφαρμογές που είναι σχεδιασμένες για γραφικό περιβάλλον ή η μεταφορά του ήδη υπάρχοντος συστήματος σε μια εφαρμογή με χρήση φωνής μόνο. Το τελικό πρόγραμμα μπορεί να δουλέψει, αλλά είναι σίγουρο, ότι εφόσον το πρόγραμμα αυτό σχεδιάστηκε με το γραφικό περιβάλλον στο μυαλό, δεν θα είναι το ίδιο αποτελεσματικό με ένα πρόγραμμα που θα σχεδιαζόταν εξ αρχής με την ομιλία στο μυαλό.

Η οργάνωση και παρουσίαση της πληροφορίας στον χρήστη είναι ένα ακόμα σημαντικό σημείο μελέτης. Μια τελική παρουσίαση πληροφορίας που ανταποκρίνεται επαρκώς στο γραφικό περιβάλλον, μπορεί να αποτύχει εντελώς σε ένα φωνητικό περιβάλλον. Ή απλή και ακριβής ανάγνωση όλων όσων εμφανίζονται στην οθόνη σπάνια οδηγεί σε ικανοποιητικά αποτελέσματα.

Για παράδειγμα, ας πούμε ότι αποφασίζουμε να διαβάσουμε τα μηνύματα ενός απλού ηλεκτρονικού ταχυδρομείου (email box), ακριβώς όπως εμφανίζονται σε ένα γραφικό περιβάλλον, δηλαδή το θέμα του, το κείμενο, τον αποστολέα, την ημερομηνία, την ώρα, ακόμα και το μέγεθος. Όπως καταλαβαίνουμε διαβάζοντας ένα ένα όλα αυτά τα στοιχεία για κάθε αντικείμενο στο mail box και χρονοβόρο είναι, αλλά και δύσκολο για τον χρήστη να συγκρατήσει τόση πολλή πληροφορία. Επίσης, δεν συμβαδίζει και με τον φυσικό τρόπο με τον οποίο θα ενημερώναμε τον χρήστη φωνητικά. Μία πιθανή παρουσίαση της πληροφορίας σε μια φωνητική εφαρμογή δεν θα περιλάμβανε πληροφορίες όπως η ημερομηνία, η ώρα ή το μέγεθος του μηνύματος

παρά μόνο μετά από αίτηση του χρήστη. Ή ακόμα καλύτερα, θα οργάνωνε την πληροφορία σε ομάδες, όπως μηνύματα από τον ίδιο χρήστη πχ Έχετε 3 μηνύματα από τον Χρήστο, ή με το ίδιο θέμα, όπως έχετε δύο μηνύματα με θέμα «ανάπτυξη φωνητικής πύλης», ή ακόμα με βάση την ημερομηνία άφιξης των μηνυμάτων, όπως έχετε 4 νέα μηνύματα. Επίσης θα μπορούσε να ανακοινωθούν τα αποτελέσματα με έναν πιο φυσικό τρόπο. Αντί για «1 (ένα) Φωτιάδης Δημήτρης. Αποστολή δεμάτων» θα μπορούσε να λέει « μήνυμα πρώτο από τον Φωτιάδη Δημήτρη με θέμα την αποστολή δεμάτων»

Από την άλλη και στην είσοδο του χρήστη υπάρχει διαφορά. Θα ήταν πολύ παράξενο και αφύσικο για αυτόν αν έλεγε «πήγαινε Φωτιάδης Δημήτρης» για να ακούσει τα μηνύματα του συγκεκριμένου προσώπου. Θα μπορούσε όμως να λέει με έναν πιο φυσικό τρόπο «δώσε μου τα μηνύματα από τον κύριο Φωτιάδη». Η φυσική μορφή εντολών είναι κάτι, το οποίο ο χρήστης θα μπορούσε να θυμάται να πει χωρίς να κοιτάει στην οθόνη του υπολογιστή.

Ακόμα όμως και αν προχωρήσουμε στον σχεδιασμό της εφαρμογής έχοντας την φυσική ομιλία στο μυαλό, υπάρχουν πραγματικές και ουσιαστικές προκλήσεις που πρέπει να αντιμετωπιστούν μέχρι να φτάσουμε στη στιγμή που η εφαρμογή θα είναι ολοκληρωμένη και έτοιμη προς χρήση. Μόνο καταλαβαίνοντας τις προκλήσεις αυτές και εκτιμώντας τα διάφορα trade-offs, που πρέπει να κάνουμε κατά τη διάρκεια της διαδικασίας σχεδιασμού και υλοποίησης, θα καταφέρουμε να έχουμε το πλέον επιθυμητό αποτέλεσμα. Παρακάτω αναφέρουμε σε γενικές γραμμές τις προκλήσεις που πρέπει να λάβουμε υπόψη μας κατά την διαδικασία σχεδιασμού εφαρμογών επεξεργασίας φωνής.

Παροδικότητα

Ο λόγος χαρακτηρίζεται από το στοιχείο της παροδικότητας. Μόλις ακούς κάτι, την επόμενη στιγμή δεν υπάρχει. Αυτό το στοιχείο δεν υπάρχει στο γραφικό περιβάλλον όπου εκεί υπάρχει το στοιχείο της μονιμότητας. Μια επιφάνεια εργασίας σε ένα γραφικό περιβάλλον παραμένει σταθερή έως ότου ο χρήστης επιλέξει να εκτελέσει μία άλλη εργασία.

Χρησιμοποιώντας τον λόγο, οι χρήστες περιορίζονται να έχουν μνήμη μικρής διάρκειας. Επειδή ο λόγος είναι παροδικός, οι χρήστες μπορούν να θυμούνται περιορισμένο αριθμό αντικειμένων σε μια λίστα ή μπορεί να ξεχάσουν κάποια

σημαντική πληροφορία που μπορεί να τους δόθηκε στην αρχή μιας μεγάλης πρότασης. Παρομοίως, μιλώντας σε ένα σύστημα υπαγόρευσης κειμένου, οι χρήστες συχνά ξεχνούν τις ακριβείς λέξεις που έχουν ήδη πει.

Η περιορισμένη αυτή δυνατότητα των χρηστών να θυμούνται παροδική πληροφορία έχει σημαντικές επιπτώσεις στον τρόπο που σχεδιάζουμε εφαρμογές επεξεργασίας φωνής. Γενικά η παροδικότητα σημαίνει ότι η χρήση φωνής δεν είναι το καλύτερο μέσο για να παρουσιάσουμε μεγάλους όγκους δεδομένων.

Η ιδιότητα όμως αυτή του λόγου μπορεί να μας προσφέρει και κάποια πλεονεκτήματα. Επειδή ο χρήστης μπορεί να ακούει και να μιλάει ταυτόχρονα, η ομιλία είναι ο ιδανικός τρόπος για να τραβήξουμε την προσοχή του ενώ παρέχει και ένα εναλλακτικό τρόπο για να πάρουμε είσοδο από τον χρήστη. Ας πούμε ότι ο χρήστης λαμβάνει ειδοποίηση για την άφιξη κάποιου μηνύματος ηλεκτρονικού ταχυδρομείου την ίδια ώρα που δουλεύει πάνω σε κάποιο περιβάλλον ανάπτυξης λογισμικού ή απλής συγγραφής. Η χρήση λόγου θα μπορούσε να δώσει στον χρήστη την ευκαιρία να ρωτήσει για το όνομα του αποστολέα ή το θέμα του μηνύματος. Ο διάλογος αυτός θα μπορούσε να λάβει χώρα χωρίς ο χρήστης να είναι αναγκασμένος να αλλάξει το περιβάλλον εργασίας στο οποίο δουλεύει.

Invisibility

Ο λόγος είναι αόρατος. Η έλλειψη αυτή ορατότητας κάνει πολύ δύσκολη την παρουσίαση των λειτουργικοτήτων της εφαρμογής στον χρήστη. Για παράδειγμα, σε μια εφαρμογή γραφικού περιβάλλοντος όλη η λειτουργικότητα, δηλαδή τα μενού και άλλα στοιχεία που δείχνουν στον χρήστη τι μπορεί να κάνει, είναι συνήθως αρκετά φανερά στον χρήστη. Αντίθετα, σε μια φωνητική εφαρμογή, είναι πολύ πιο δύσκολο να δώσεις στον χρήστη να καταλάβει ποιες εργασίες μπορεί να εκτελέσει, και πολύ περισσότερο, πώς να τις εκτελέσει, δηλαδή ποιες λέξεις και φράσεις πρέπει να χρησιμοποιήσει για να μπορέσει να εκτελέσει τις εργασίες αυτές.

Ασυμμετρία

Η ομιλία έχει και ένα άλλο χαρακτηριστικό, το στοιχείο της ασυμμετρίας. Οι άνθρωποι τείνουν να μιλούν εύκολα και γρήγορα, αλλά ωστόσο έχουν δυσκολία στον να ακούν το ίδιο γρήγορα και εύκολα. Αυτή η ασυμμετρία έχει ως αποτέλεσμα οι

άνθρωποι να μπορούν να μιλούν πιο γρήγορα από ότι μπορούν να πληκτρολογούν, αλλά από την άλλη, μπορούν να ακούν πολύ πιο αργά από ότι μπορούν να διαβάζουν.

Αυτή η ασυμμετρία έχει σχεδιαστικές επιπτώσεις όσον αφορά στο είδος της πληροφορίας που θα μεταφραστεί σε ομιλία προς τον χρήστη, καθώς και στον όγκο της. Ένας σχεδιαστής μιας φωνητικής εφαρμογής πρέπει να λάβει υπόψη του το στοιχείο αυτό. Θα πρέπει να εξισορροπήσει την ανάγκη να προσαρμοστούν πολλές από τις εντολές, από και προς τον χρήστη, στην περιορισμένη δυνατότητα του χρήστη να δεχθεί πληροφορίες που προέρχονται από την χρήση φωνής.

Συνθετική Φωνή vs Προηχογραφημένα Μηνύματα

Λόγω του γεγονότος ότι η παραγόμενη φωνή από τις υπάρχουσες μηχανές σύνθεσης φωνής ακόμα δεν ακούγεται απόλυτα φυσική, η επιλογή μεταξύ φωνής από σύνθεση, προηχογραφημένων μηνυμάτων, ή της μη χρήσης φωνητικής εξόδου είναι συχνά πολύ δύσκολη. Παρόλο που η μέθοδος των προηχογραφημένων μηνυμάτων είναι πολύ πιο εύκολη, και ευχάριστη στην ακοή για τους χρήστες, είναι πολύ δύσκολο να εφαρμοστεί όταν τα δεδομένα που πρόκειται να παρουσιαστούν είναι δυναμικά. Για παράδειγμα, η μέθοδος που χρησιμοποιεί προηχογραφημένη φωνή δεν θα μπορούσε να καλύψει τις ανάγκες για την ανάγνωση μηνυμάτων ηλεκτρονικού ταχυδρομείου μέσω του τηλεφώνου. Και αυτό για τον προφανή λόγο ότι τα θέματα αλλάζουν από μήνυμα σε μήνυμα, και δεν μπορούμε να ξέρουμε εκ των προτέρων το κείμενο που περιέχουν. Έτσι, ή χρήση προηχογραφημένων μηνυμάτων είναι ενδεδειγμένη όταν έχουμε κείμενα που δεν αλλάζουν, ενώ αντίθετα η φωνή από σύνθεση είναι αυτή που προτιμάται για κείμενα που αλλάζουν δυναμικά.

Από την άλλη, ο συνδυασμός προηχογραφημένης ομιλίας και σύνθεσης φωνής δεν είναι η ενδεδειγμένη λύση. Παρά το γεγονός ότι οι χρήστες αναφέρουν ότι δεν τους είναι ευχάριστη η φωνή από σύνθεση, είναι στην ουσία σε θέση να συνηθίσουν σ' αυτή πιο εύκολα από όταν είναι συνδυασμένη με προηχογραφημένα μηνύματα. Η ικανότητα κατανόησης κατά την διαδικασία ακοής είναι σημαντικά μεγαλύτερη όταν η φωνή είναι αμετάβλητη.

Σε γενικές γραμμές, είναι καλύτερο να χρησιμοποιούμε προηχογραφημένα μηνύματα όταν είναι γνωστά τα κείμενα εκ των προτέρων, ή όταν πρέπει να δώσουμε ένα συγκεκριμένο χαρακτήρα ή μια ιδιαίτερη προσωπικότητα στον εκφωνητή των μηνυμάτων. Ενώ όταν τα κείμενα δεν μας είναι γνωστά από πριν ή η δυνατότητα

αποθήκευσης του συστήματος είναι περιορισμένη, τότε είναι καλύτερη η χρήση σύνθεσης φωνής.

Περιορισμοί Λόγω Απόδοσης της Αναγνώρισης Φωνής

Οι αναγνωριστές φωνής δεν είναι, σε καμία, περίπτωση τέλει ακροατές. Οποσδήποτε είναι επιρρεπείς σε λάθη. Οπότε μια από τις μεγάλες προκλήσεις στον σχεδιασμό των εφαρμογών επεξεργασίας φωνής είναι η ανάπτυξη εφαρμογών με μια ατελή τεχνολογία αναγνώρισης φωνής. Παρόλη την πρόοδο που παρατηρείται στις τεχνολογίες αυτές, είναι μάλλον απίθανο, στο άμεσο μέλλον τουλάχιστον, να έχουμε φτάσει στο επίπεδο των συστημάτων που συναντάμε στις ταινίες επιστημονικής φαντασίας. Όπως ο HAL στην ταινία «Η Οδύσσεια του Διαστήματος» του Stanley Kubrick.

Ένας σχεδιαστής τέτοιων εφαρμογών είναι υποχρεωμένος να καταλαβαίνει τους τύπους των λαθών που κάνουν οι αναγνωριστές φωνής και τους συνήθεις λόγους που προκαλούν τα λάθη αυτά.

Δυστυχώς τα λάθη, τα οποία κάνει ο αναγνωριστής οδηγούν, πολλές φορές τους χρήστες, να σχηματίζουν λανθασμένη άποψη για την λειτουργικότητα του συστήματος. Για παράδειγμα, αν ο χρήστης πει «διάβασε το επόμενο μήνυμα» και ο αναγνωριστής καταλάβει «ξαναπές το ίδιο μήνυμα» τότε το σύστημα θα επαναλάβει το ίδιο μήνυμα και ο χρήστης θα νομίζει ότι αυτό δεν είναι ο σωστός τρόπος για να δώσει την εντολή αυτή. Αν στη συνέχεια ο χρήστης πει «επόμενο» και το σύστημα απορρίψει το αίτημά του ως λάθος, τότε ο χρήστης θα αποκλείσει την επιλογή αυτή ως έγκυρη για να εκτελέσει την εργασία αυτή. Οι χρήστες δεν μπορούν να ξέρουν αν οι λέξεις που είπαν θα έπρεπε να δουλέψουν, εκτός και αν υπάρχει μια λίστα με όλες έγκυρες εντολές. Για αυτό αν οι εντολές δεν δουλεύουν, τότε οι χρήστες υποθέτουν ότι δεν είναι και έγκυρες.

Μερικά συστήματα αναγνώρισης προσαρμόζονται με τον καιρό στους χρήστες. Ωστόσο, για να έχουμε μια καλή απόδοση της αναγνώρισης απαιτείται να έχουμε συνεργάσιμους χρήστες, οι οποίοι να είναι διατεθειμένοι να προσαρμόσουν την ομιλία τους στις ανάγκες του συστήματος αναγνώρισης. Για αυτό είναι απαραίτητο οι χρήστες που θα κληθούν να χρησιμοποιήσουν μια εφαρμογή επεξεργασίας φωνής, να έχουν ένα ισχυρό και ξεκάθαρο κίνητρο για να κάνουν την εφαρμογή να δουλέψει για αυτούς.

Ευελιξία vs. Ακρίβεια

Ένα ευέλικτο σύστημα επιτρέπει στους χρήστες να καλούν τις ίδιες εντολές με πολλούς και διάφορους τρόπους. Όση περισσότερη ευελιξία παρέχει μια εφαρμογή στους χρήστες για είσοδο, τόσο αυξάνεται η πιθανότητα εμφάνισης λαθών. Για αυτό κατά την διαδικασία σχεδιασμού μια εφαρμογής κατευθυνόμενης και ελεγχόμενης από εντολές, ο σχεδιαστής πρέπει να βρει την ισορροπία μεταξύ της ευελιξίας και της ακρίβειας που θέλει να έχει το σύστημα αναγνώρισης του συστήματός του. Για παράδειγμα, σε μια εφαρμογή ημερολογίου-ατζέντας, μπορεί να επιτραπεί στον χρήστη να ρωτήσει για τα αυριανά ραντεβού με διάφορους τρόπους όπως:

- Πες μου για αύριο
- Τι έχω αύριο;
- Τι περιλαμβάνει η ατζέντα μου για αύριο;
- Διάβασέ μου το αυριανό πρόγραμμα
- Πες μου για τα ραντεβού που έχω στην ατζέντα μου για αύριο

Αυτό μπορεί να φαίνεται αρκετά φυσικό στην θεωρία, αλλά, αν η απόδοση του συστήματος αναγνώρισης είναι χαμηλή, τότε αναπόφευκτα οι χρήστες δεν θα το αποδεχτούν. Από την άλλη μεριά, εφαρμογές που προσφέρουν μόνο ένα μικρό, συγκεκριμένο σύνολο εντολών πάλι μπορεί να μην γίνουν αποδεκτές, ακόμα και αν οι φράσεις που χρησιμοποιούνται είναι σχεδιασμένες να ακούγονται φυσικά. Οι χρήστες συνήθως ξεχνούν τις ακριβείς λέξεις που αποτελούν τις συγκεκριμένες αυτές εντολές. Επιπλέον, κάτι το οποίο μπορεί να φαίνεται φυσικό για κάποιον, κάποιος άλλος μπορεί να το βρίσκει περίεργο. Για αυτό τον λόγο χρησιμοποιούμε τους μελλοντικούς χρήστες, για να βρούμε τους πιο κοινούς τρόπους, με τους οποίους εκφράζονται πάνω σε κάποιο συγκεκριμένο θέμα. Κατα αυτόν τον τρόπο οι εφαρμογές θα μπορούν να διακρίνονται από ένα αρκετά μεγάλο βαθμό ευελιξίας χωρίς όμως να προκαλείται δραματική μείωση της απόδοσης της αναγνώρισης.

2.3.3 Θέματα που αφορούν Εφαρμογές Φωνής

Εφαρμογές φωνής είναι αυτές, κατά τις οποίες η είσοδος και η έξοδος μέσω φωνής είναι τα μόνα μέσα, τα οποία είναι διαθέσιμα στον χρήστη. Οι περισσότερες από αυτές τις εφαρμογές λειτουργούν μέσω τηλεφώνου.

Στην ομιλία μας δεν έχουμε κάποιο συγκεκριμένο ρυθμό. Πολλές φορές ανάμεσα στις λέξεις παρεμβάλλονται παύσεις. Αυτό δημιουργεί σφάλματα στον αναγνωριστή, καθώς πολλές φορές μπορεί να δεχθεί την είσοδο πριν ο χρήστης ολοκληρώσει την φράση-εντολή. Ένα άλλο σημείο σύγχυσης είναι όταν ο χρήστης δεν παίρνει άμεση απάντηση στο αίτημά του, πχ λόγω χρόνου επεξεργασίας της αίτησης. Τότε ο χρήστης ξαναμιλάει, νομίζοντας ότι ο αναγνωριστής δεν τον άκουσε. Γι' αυτό ανακύπτει ή ανάγκη να έχουμε όσο το δυνατό πιο σωστή απόκριση από τον χρήστη, έτσι ώστε να μπορέσουμε να πετύχουμε την πλήρη ικανοποίηση των αιτημάτων του.

Όταν σχεδιάζουμε τους διαλόγους μεταξύ χρήστη και υπολογιστή, πρέπει να έχουμε υπόψη, ότι η έξοδος του συστήματος αργεί. Αυτό θα πρέπει να συνυπολογιστεί και με άλλους παράγοντες, όπως η ανάγκη του χρήστη να ξέρει ανά πάσα στιγμή σε τι κατάσταση βρίσκεται το σύστημα, αν δηλαδή ο αναγνωριστής έχει δεχτεί την είσοδό του και βρίσκεται σε στάδιο επεξεργασίας, και αν την έχει δεχτεί θέλει να ξέρει αν η είσοδός του έχει ερμηνευτεί σωστά. Μια συνηθισμένη μέθοδος είναι να ενημερώνεται συχνά ο χρήστης για τις εισόδους που έχει δώσει στο σύστημα. Αυτό πρέπει να συμπεριληφθεί στο κόστος εκτέλεσης μιας εργασίας. Επιβεβαιώνουμε έμμεσα τις εντολές όταν πρόκειται για απλή παρουσίαση δεδομένων και ζητάμε ευθεία και άμεση επιβεβαίωση όταν καταστρέφουμε ή ενημερώνουμε δεδομένα καθώς και όταν πρόκειται να εκκινήσουμε κάποια εργασία. Για παράδειγμα, θα ήταν πολύ σημαντικό να ζητάμε για άμεση και συνεχή επιβεβαίωση από τον χρήστη όταν πρόκειται να μεταφέρουμε ένα μεγάλο ποσό από τον λογαριασμό του, ενώ δεν θα μας ενδιέφερε να ζητήσουμε επιβεβαίωση για την ημερομηνία, αν είναι απλά να δώσουμε μια πρόγνωση καιρού. Έτσι, για την μεταφορά χρημάτων θα χρησιμοποιούσαμε μια άμεση ερώτηση, όπως «θα θέλατε να μεταφέρετε €5000 στην Χ Α.Ε; παρακαλώ πείτε ναι ή όχι». Για την πρόγνωση καιρού θα είχαμε μια πρόταση της μορφής « αύριο ο καιρός...». Λέγοντας αύριο δίνουμε και την έμμεση επιβεβαίωση στον χρήστη ότι καταλάβαμε την είσοδο του και σε περίπτωση που έχει γίνει σωστή αναγνώριση, η εκτέλεση της εφαρμογής προχωράει μπροστά χωρίς την παραμικρή καθυστέρηση..

Δεχόμενοι είσοδο από τον χρήστη

Μηνύματα παρακίνησης εισόδου ή απλά παρακινήσεις (prompts), τα οποία είναι καλά σχεδιασμένα, έχουν ως αποτέλεσμα ο χρήστης να έχει μια πετυχημένη αλληλεπίδραση με την φωνητική εφαρμογή. Πολλοί παράγοντες πρέπει να ληφθούν υπόψη κατά την διαδικασία σχεδιασμού των κειμένων αλληλεπίδρασης, αλλά ο πιο σημαντικός είναι ο καθορισμός των ορίων μεταξύ της ευελιξίας και της απόδοσης. Όσο πιο πολύ περιορίζουμε τον χρήστη σε συγκεκριμένες φράσεις, τόσο λιγότερη πιθανότητα έχουμε να συναντήσουμε προβλήματα αναγνώρισης. Από την άλλη, η μεγαλύτερη ευελιξία στις δυνατότητες εισόδου του χρήστη, συχνά αυξάνει την ταχύτητα εκτέλεσης των εντολών, ενώ παρέχει μεγαλύτερη φυσικότητα, και επιπλέον, οι χρήστες δεν είναι αναγκασμένοι να θυμούνται τις εντολές.

Τα μηνύματα σχεδιάζονται ακολουθώντας κάποια βασικά πρότυπα που περιγράφονται παρακάτω.:

Σύντομες Παρακινήσεις (short prompts)

Μια καλή πρακτική είναι να κρατάμε τις παρακινήσεις προς τον χρήστη σύντομες και περιεκτικές. Είναι δύσκολο, σύμφωνα με τους ψυχολόγους, για έναν άνθρωπο να συγκρατήσει μεγάλα ποσά πληροφορίας, με μια απλή αναφορά στην πληροφορία, όπως γίνεται στην ομιλία. Αρκεί να απευθυνθούμε σε κάποιον χρησιμοποιώντας μια μακριά και πολύπλοκη συστοιχία λέξεων και να τον ρωτήσουμε αν θυμάται ακριβώς αυτά που του είπαμε. Μπορούμε εύκολα να φανταστούμε ότι η κατάσταση γίνεται ακόμα χειρότερη με την χρήση σύνθεσης φωνής (synthesized speech).

Για αυτό είναι απαραίτητη η χρήση σύντομων παρακινήσεων, που να περιέχουν μόνο την χρήσιμη, ως προς τον χρήστη, πληροφορία. Επιπλέον, όταν επιδιώκουμε να δώσουμε οδηγίες στον χρήστη, για την επόμενη εντολή του, καλό είναι οι οδηγίες αυτές να βρίσκονται σε κάποια παρακίνηση βοήθειας (help prompt).

Άμεσες Παρακινήσεις (explicit prompts)

Χρησιμοποιούνται όταν η είσοδος του χρήστη πρέπει να είναι σαφώς καθορισμένη και περιορισμένη. Για παράδειγμα, για την εξόφληση κάποιου λογαριασμού, η παρακίνηση θα μπορούσε να είναι «για ολοκλήρωση της συναλλαγής πείτε ναι, πείτε όχι για να την ακυρώσετε». Κατ' αυτόν τον τρόπο η παρακίνηση οδηγεί τον χρήστη να πει ακριβώς ναι ή όχι. [4].

Έμμεσες Παρακινήσεις (implicit prompts)

Χρησιμοποιούνται όταν η εργασία που πρόκειται να εκτελεστεί μπορεί να αποδεχτεί πιο χαλαρή είσοδο. Οι παρακινήσεις αυτές βασίζονται σε διάφορους κανόνες που χρησιμοποιούμε στην ομιλία μας, και έχουν σκοπό να περιορίσουν την είσοδο του χρήστη. Για παράδειγμα, αν ο χρήστης πει «δώσε μου τα μηνύματα του Γιώργου», και υπάρχουν μηνύματα από δύο χρήστες με το όνομα Γιώργος, τότε το κείμενο παρακίνησης εισόδου του συστήματος θα ήταν «θα θέλατε τα μηνύματα του Γιώργου Παπαδόπουλου ή του Γιώργου Βασιλειάδη?». Οι χρήστες είναι δυνατόν να απαντήσουν κάπως είτε «του Βασιλειάδη; » ή «τα μηνύματα του Γιώργου Βασιλειάδη». Οι χρήστες δύσκολα θα απαντούσαν με μια πρόταση της μορφής «Εννοούσα ότι θέλω τα μηνύματα του Γιώργου Βασιλειάδη», αν και αυτό δεν είναι απίθανο.

Συντομευμένες Παρακινήσεις (tapered prompts)

Συντομεύουμε τις παρακινήσεις όταν αυτό είναι δυνατό. Υπάρχουν δύο περιπτώσεις όπου μπορεί να συμβεί αυτό. Η μια είναι όταν παρουσιάζουμε μια σειρά από όμοια δεδομένα, οπότε όσο προχωράμε αποφεύγουμε τις περιττές λέξεις. Πχ

«Αύριο στις δέκα έχετε Φυσική Ι στην αίθουσα Β.1.1,στις έντεκα Λειτουργικά Συστήματα στο ξύλινο αμφιθέατρο ».

Αλλά συντόμευση μπορούμε να έχουμε και όταν δίνουμε στον χρήστη την ίδια πληροφορία παραπάνω από μια φορά. Κάθε φορά που επαναλαμβάνουμε την πληροφορία προσπαθούμε να βγάλουμε τις λέξεις που δεν περιέχουν την πληροφορία. Π.χ., την πρώτη φορά που θα δώσουμε πληροφορία σε έναν χρήστη για

το πώς να διασχίσει μια λίστα, η παρακίνηση μπορεί είναι λεπτομερής αλλά την επόμενη φορά θα είναι συντομότερη μέχρι που στο τέλος θα είναι τελείως επιγραμματική. Π.χ.

«Για να διατρέξετε τη λίστα χρησιμοποιείτε τις εξής εντολές. Επόμενη για να προχωρήσετε στο επόμενο στοιχείο της λίστας, προηγούμενη για να γυρίσετε στο προηγούμενο στοιχείο και τέλος για να βγείτε από τη λίστα.»

«Πείτε επόμενη για να πάτε στο επόμενο στοιχείο της λίστας, προηγούμενη για το προηγούμενο και τέλος για να προχωρήσετε παρακάτω.»

«Χρησιμοποιείτε τις εντολές επόμενη, προηγούμενη, τέλος για να πλοηγηθείτε»

Παρακινήσεις Αυξανόμενου Μεγέθους (incremental prompts)

Χρησιμοποιούνται για να αυξήσουμε την ταχύτητα αλληλεπίδρασης για τους έμπειρους χρήστες και για να παρέχουμε βοήθεια στους λιγότερο έμπειρους. Αυτό επιτυγχάνεται αρχίζοντας από μικρές παρακινήσεις και προχωρώντας σε πιο λεπτομερείς σε περίπτωση που ο χρήστης δεν μπορεί να αποκριθεί. Ας πούμε, το αρχικό κείμενο παρακίνησης θα μπορούσε να είναι «βρίσκεστε στο κεντρικό μενού, πείτε μου τι θα θέλατε να κάνετε» ενώ αν ο χρήστης δεν μπορέσει να ανταποκριθεί τότε η παρακίνηση θα γίνει πιο λεπτομερής «πείτε ένα από τα εξής: Ακαδημαϊκά Θέματα, Υπηρεσίες του Πολυτεχνείου...»

Διαχείριση Λαθών

Ένας σημαντικός παράγοντας για την επιτυχία μιας εφαρμογής είναι ο τρόπος διαχείρισης σφαλμάτων από το σύστημα. Αν παρουσιαστεί κάποιο σφάλμα, είτε από τη μεριά του χρήστη είτε από την ίδια την εφαρμογή, μια αποτελεσματική φωνητική διεπαφή (speech user interface) θα πρέπει να διαθέτει τους απαραίτητους μηχανισμούς για να διαχειριστεί το λάθος αυτό. Αυτό που προσπαθούμε να κάνουμε

είναι να αποφύγουμε να έχουμε την ίδια είσοδο από τον χρήστη μιας και υπάρχει μεγάλη πιθανότητα να εμφανιστεί το ίδιο σφάλμα [4] [6].

Όταν το σύστημα δεν μπορεί να κατανοήσει την είσοδο του χρήστη, το σφάλμα αυτό λέγεται σφάλμα αποδοχής εισόδου (rejection error). Μια καλή τεχνική, για να αποφύγουμε αυτού του είδους τα λάθη, είναι να μην επαναλαμβάνουμε το ίδιο μήνυμα λάθους όταν ο χρήστης δεν κατορθώνει να δώσει την εντολή του σε διαδοχικές προσπάθειες. Επιπλέον, η επανάληψη της ίδιας παρακίνησης συχνά λαμβάνεται από τους χρήστες ως αγένεια. Η λύση είναι να παρέχουμε στον χρήστη την απαιτούμενη βοήθεια σταδιακά, και με τη μορφή κλιμάκωσης (progressive assistance). Για παράδειγμα, μια αρχική φράση θα μπορούσε να ήταν «συγγνώμη, αλλά δεν σας κατάλαβα» και στην συνέχεια, αν έχουμε επανάληψη του σφάλματος, θα μπορούσαμε να πούμε στον χρήστη να προσπαθήσει ξανά με κάποια διαφορετική φράση, και, αν πάλι δεν έχουμε το επιθυμητό αποτέλεσμα, θα μπορούσαμε να τον συμβουλευόμαστε να χρησιμοποιήσει καλύτερη προφορά.

Μια άλλη τεχνική είναι να χρησιμοποιήσουμε μια άμεση παρακίνηση, όπως να εμφανίσουμε στον χρήστη τις πιθανές επιλογές του, όπως «πείτε φωνητικό ταχυδρομείο ή υπηρεσία αναζήτησης γιατρών».

Επίσης μια ενδεδειγμένη τεχνική είναι να υπάρχει ένας εναλλακτικός τρόπος παροχής εισόδου, όπως για παράδειγμα να παρακινήσουμε τον χρήστη να χρησιμοποιήσει τα πλήκτρα στην συσκευή του τηλεφώνου του σε περίπτωση που η ομιλία αποτύχει.

Σε περιπτώσεις λανθασμένης αναγνώρισης τα λάθη είναι πιο δύσκολο να ανιχνευτούν. Ένας τρόπος αντιμετώπισης είναι να φιλτράρονται όλες οι εισοδοί του χρήστη που δεν έχουν νόημα. Για παράδειγμα, δεν μπορεί παρά να θεωρήσουμε ότι κάποιο λάθος έχει συμβεί, αν κάποιος χρήστης ζητήσει να μάθει τι μάθημα έχει στις 10μμ!!

Επίσης είναι καλό να υπάρχουν ευέλικτοι μηχανισμοί που να δίνουν την δυνατότητα στον χρήστη να διορθώνει έγκαιρα ολόκληρη ή μέρος της εισόδου του. Για παράδειγμα αν ο χρήστης ζητήσει το τηλέφωνο του κυρίου Διγαλάκη και το σύστημα αποκριθεί «Ζητήσατε το τηλέφωνο του κ. Ζερβάκη;», ένας ευέλικτος μηχανισμός διόρθωσης λαθών θα έπρεπε να επιτρέπει στον χρήστη να διορθώσει το όνομα: «Διγαλάκης είπα»

2.3.4 Εμπλέκοντας τους χρήστες

Όπως γίνεται εύκολα κατανοητό, η εμπλοκή των μελλοντικών χρηστών της εφαρμογής, κατά την διαδικασία του σχεδιασμού της, είναι ένας σημαντικός παράγοντας για την επιτυχία της. Για να επιτευχθεί η φυσικότητα και η αποτελεσματικότητα της εφαρμογής χρειάζεται να καταλάβουμε με ποιους τρόπους, σε ποια σημεία της δραστηριότητάς τους, και για ποιο λόγο οι χρήστες στους οποίους απευθύνεται η εφαρμογή θα αλληλεπιδρούν μαζί της.

Μελέτη Φυσικών Διαλόγων

Στα πρωταρχικά στάδια της εφαρμογής, οι χρήστες μπορούν να βοηθήσουν στον καθορισμό της λειτουργικότητας της εφαρμογής, αλλά και να προσφέρουν πληροφορίες για το πώς οι άνθρωποι συμπεριφέρονται στις μεταξύ τους συζητήσεις πάνω στο θέμα που πρόκειται να υλοποιηθεί. Αυτό γίνεται μέσα από την μελέτη φυσικών διαλόγων. Αυτό σημαίνει ότι δυνητικοί χρήστες καλούνται να συζητήσουν μεταξύ τους πάνω σε κάποια συγκεκριμένα σενάρια που θέλουμε να υλοποιηθούν. Για παράδειγμα, ας θεωρήσουμε ότι θέλουμε να σχεδιάσουμε ένα σενάριο όπου ένας χρήστης εισέρχεται στο σύστημα για να μάθει το πρόγραμμα της εξεταστικής. Τότε μπορούμε να χρησιμοποιήσουμε δύο χρήστες, για να κάνουμε την προσομοίωση της εργασίας. Ο ένας θα έχει τον ρόλο του χρήστη που θέλει να ακούσει το πρόγραμμα. Ο άλλος θα έχει τον ρόλο του συστήματος, και έχοντας μπροστά του το πρόγραμμα της εξεταστικής, θα το ανακοινώνει στον πρώτο. Έτσι ακούγοντας την συζήτηση μεταξύ τους μπορούμε να έχουμε μια εικόνα των διαλόγων που πρόκειται να σχεδιάσουμε.

Όταν η εφαρμογή, που θέλουμε να υλοποιήσουμε, είναι πολύ εξειδικευμένη, μια καλή πρακτική είναι να επιστρατεύσουμε στην θέση του συστήματος ένα εξειδικευμένο πάνω στο θέμα άτομο. Για παράδειγμα, αν είναι να υλοποιήσουμε μια υπηρεσία μιας χρηματοοικονομικής εταιρείας, τότε οι συμμετέχοντες χρήστες θα μπορούσαν να εξυπηρετούνται από κάποιον ειδικά εξουσιοδοτημένο από την εταιρεία σε θέματα πελατών.

Μέσα από τις μελέτες φυσικών διαλόγων μπορούμε να αντλήσουμε το απαραίτητο λεξιλόγιο της εφαρμογής, να εμπεδώσουμε τα γραμματικά πρότυπα (grammatical patterns) που εμφανίζονται συχνά, και να πάρουμε ιδέες για τα κείμενα παρακινήσεων που θα χρησιμοποιήσουμε καθώς και το είδος απόκρισης που θα επιστρέφουμε στον χρήστη. Στην περίπτωση που υπάρχει συμμετοχή κάποιου ειδικού πάνω στο θέμα της υλοποίησης, όπως περιγράψαμε παραπάνω, μπορούμε να κάνουμε τον σχεδιασμό βασιζόμενοι στις φράσεις και απαντήσεις, που εκείνος χρησιμοποιεί στην επικοινωνία του με τους πελάτες.

Οι μελέτες φυσικού λόγου μπορούν να γίνουν με μικρή συμμετοχή των χρηστών και είναι μια γρήγορη και ανέξοδη διαδικασία.

Μελέτη τύπου Wizard-of-Oz.

Μόλις τελειώσει ένας πρωταρχικός σχεδιασμός και χωρίς να έχουμε φτάσει στην τελειοποίηση της εφαρμογής, μια μελέτη τύπου Wizard-of-Oz θα βοηθούσε στην δοκιμή και καλυτέρευση της φωνητικής διεπαφής. Σ' αυτή τη μελέτη κάποιος από τους συμμετέχοντες, ο λεγόμενος wizard (μάγος), χρησιμοποιώντας τα κατάλληλα εργαλεία λογισμικού, προσομοιώνει την φωνητική διεπαφή. Χρήστες που συμμετέχουν στην μελέτη, παίρνουν σε κάποιο νούμερο τηλεφώνου και ένας άνθρωπος ακούει τις εντολές τους, αλλά χρησιμοποιεί τον υπολογιστή για να απαντήσει με συνθετική φωνή. Καθώς λοιπόν ο χρήστης ζητάει να εκτελέσει τις διάφορες εργασίες –ανάγνωση φωνητικών μηνυμάτων, ανάγνωση προγράμματος μαθημάτων κτλ.-, ο wizard αναλαμβάνει να εκτελέσει τις εργασίες αυτές, χρησιμοποιώντας τα εργαλεία λογισμικού που έχει στην διάθεσή του, και παρουσιάζει τα αποτελέσματα χρησιμοποιώντας την φωνή που συνθέτει το σύστημα.

Επειδή όμως τα εργαλεία λογισμικού, όπως δείξαμε προηγουμένως, είναι απαραίτητα για την διεκπεραίωση της μελέτης αυτής, γίνεται φανερό ότι είναι πιο χρονοβόρα και περίπλοκα από ότι οι μελέτες φυσικών διαλόγων. Τέλος αν μια πιλοτική έκδοση της εφαρμογής μπορεί να ολοκληρωθεί σχετικά γρήγορα είναι προτιμότερο από πλευράς κόστους και χρόνου να προχωρήσουμε στην μελέτη χρηστικότητας.

Μελέτη Χρηστικότητας Εφαρμογής

Η μελέτη χρηστικότητα μιας εφαρμογής καθορίζει πόσο καλά μπορούν οι χρήστες να χρησιμοποιήσουν τις διάφορες λειτουργικότητες που υποστηρίζει η εφαρμογή. Για να φτάσουμε σ' αυτό το σημείο, δεν χρειάζεται να έχει υλοποιηθεί ολόκληρη η εφαρμογή. Θα πρέπει, όμως πρώτα, να έχει υλοποιηθεί κάποια ελάχιστη λειτουργικότητα. Στους χρήστες παρουσιάζονται κάποια σενάρια και αυτοί θα πρέπει να χρησιμοποιήσουν το σύστημα για να εκτελέσουν κάποιες εργασίες.

Στις φωνητικές εφαρμογές, η μελέτη χρηστικότητα είναι ιδιαίτερα σημαντική για την ανίχνευση λαθών αναγνώρισης, κάτι το οποίο δεν θα μπορούσε να γίνει επιτυχώς μέσω μιας μελέτης τύπου Wizard-of-Oz. Και όπως έχουμε προαναφέρει τα λάθη αυτά επηρεάζουν σημαντικά την ποιότητα της εφαρμογής. Μέσα από την μελέτη αυτή έχουμε την δυνατότητα να δοκιμάσουμε την ικανότητα του συστήματος να διαχειρίζεται λάθη, μέσα σε κάποιο αληθινό περιβάλλον, όπου συμμετέχουν πραγματικοί χρήστες.

Στην μελέτη χρηστικότητα που ακολουθήσαμε εμείς, η λειτουργικότητα της εφαρμογής ήταν σχεδόν ολοκληρωμένη. Η μελέτη έγινε σε ένα σύνολο είκοσι επτά χρηστών. Στους χρήστες δόθηκε το ερωτηματολόγιο του Appendix B – Usability Questionnaire, το οποίο ζητάει από τους χρήστες να εκτελέσουν κάποιες συγκεκριμένες λειτουργίες, όπως να μάθουν το πρόγραμμα μαθημάτων για κάποια συγκεκριμένη μέρα, ή να ψάξουν για κάποιον γιατρό.

Κεφάλαιο 3:

3. Υλοποίηση της Εφαρμογής Φωνητικής Πύλης Πολυτεχνείου Κρήτης

Η εφαρμογή όπως προαναφέραμε χωρίζεται σε τρία σκέλη. Το πρώτο σκέλος αφορά στην υλοποίηση βάσης δεδομένων της εφαρμογής που περιέχει τις απαραίτητες πληροφορίες σχετικά με όλα τα θέματα που διαχειρίζεται η εφαρμογή. Το δεύτερο σκέλος αφορά το Web-based σύστημα, που επιτρέπει την ενημέρωση και διατήρηση της βάσης δεδομένων. Και το τρίτο σκέλος είναι η φωνητική εφαρμογή, που επιτρέπει στους χρήστες να κάνουν ερωτήσεις προς την βάση, παίρνοντας τις ζητούμενες πληροφορίες.

Όπως γίνεται εύκολα αντιληπτό, τα δύο τελευταία επίπεδα της εφαρμογής που βρίσκονται πάνω από την βάση δεδομένων, το Web-based σύστημα και η φωνητική εφαρμογή, χρησιμοποιούν και τα δύο την βάση αυτή. Η στρατηγική υλοποίησης που χρησιμοποιήσαμε ήταν η ίδια και στις δύο περιπτώσεις.

Η υλοποίηση της κάθε υποεφαρμογής χωρίστηκε σε δύο κομμάτια. Πρώτα υλοποιήσαμε το Front End κομμάτι. Αυτό είναι το μέρος που αναλαμβάνει την άμεση επικοινωνία με τους χρήστες. Στην φωνητική εφαρμογή το κομμάτι αυτό αφορά τον σχεδιασμό των διαλόγων και των κατάλληλων γραμματικών που θα επιτρέψουν στον χρήστη να έχει μια αποτελεσματική πλοήγηση στις διάφορες ενότητες της εφαρμογής, όπου θα μπορεί να κάνει αναζήτηση για την επιθυμητή για αυτόν πληροφορία. Στο Web-based σύστημα η υλοποίηση του Front End περιλαμβάνει τον σχεδιασμό των στατικών σελίδων HTML [11] τις οποίες οι χρήστες θα χρησιμοποιούν για να έρθουν σε επικοινωνία με την βάση.

Το δεύτερο κομμάτι περιλαμβάνει την υλοποίηση του Back End. Αυτό αφορά στην υλοποίηση της λειτουργικότητας, η οποία δεν γίνεται αντιληπτή από τον χρήστη. Η αρχιτεκτονική και οι τεχνολογίες που χρησιμοποιήθηκαν σε αυτό το κομμάτι ήταν ίδιες και για τις δύο υποεφαρμογές που υλοποιήθηκαν. Το αρχιτεκτονικό μοντέλο που χρησιμοποιήσαμε για την διασύνδεση όλων αυτών των κομματιών υλοποίησης περιγράφεται παρακάτω.

3.1 Το Αρχιτεκτονικό Μοντέλο Model View Controller

Ο σκοπός των υπολογιστικών συστημάτων, όπως το δικό μας είναι η ανάκτηση δεδομένων από μια βάση αποθήκευσης και η παρουσίασή τους στον χρήστη. Αυτό όμως δεν πρέπει να σημαίνει ότι τα δύο αυτά κομμάτια της υλοποίησης (Front End και Back End αντίστοιχα) του υπολογιστικού συστήματος θα πρέπει να είναι άρρηκτα συνδεδεμένα για λόγους απλούστευσης, ή εξοικονόμησης κώδικα και βελτιστοποιημένης απόδοσης. Μια τέτοια προσέγγιση ονομάζεται view centric και ακολουθείται σε τεχνολογίες όπως, PHP [15] και CGI [16]. Μια τέτοια προσέγγιση υποκρύπτει πολλά και αδιέξοδα προβλήματα, ειδικά όταν έχουμε να κάνουμε με μεγάλες, περίπλοκες και επεκτάσιμες εφαρμογές. Θα προσπαθήσουμε να συνοψίσουμε τα προβλήματα αυτά.

1. Η διασυνδετική διάταξη υπολογιστή χρήστη τείνει να υφίσταται πολύ πιο συχνά αλλαγές από ότι το σύστημα αποθήκευσης δεδομένων.
2. Οι συνήθεις εμπορικές εφαρμογές τείνουν να ενσωματώνουν επιχειρηματική λογική (business logic) που ξεφεύγει από τα όρια της απλής μεταφοράς δεδομένων.
3. Πολλές φορές σε μια εφαρμογή μπορεί να χρειαστεί να παρουσιαστούν τα ίδια δεδομένα με διαφορετικούς τρόπους. Στην περίπτωση μας, για παράδειγμα έχουμε παρουσίαση των δεδομένων τόσο σε γραφικό περιβάλλον GUI(Graphic User Interface) όσο και σε φωνητικό περιβάλλον VUI (Voice User Interface)
4. Ο σχεδιασμός των διάφορων διεπαφών ηλεκτρονικών υπολογιστών, είτε αυτό είναι HTML είτε VoiceXML, απαιτεί διαφορετικό φάσμα ικανοτήτων, όσον αφορά αυτούς που ασχολούνται με την ανάπτυξη τέτοιων συστημάτων, από τον σχεδιασμό περίπλοκων επιχειρηματικών λογικών.
5. Πολλές από τις αιτήσεις από τον χρήστη (requests) οδηγούν στην ίδια τελική παρουσίαση. Για παράδειγμα, μετά από προσθήκη ή διαγραφή ενός στοιχείου από μια λίστα, και στις δυο περιπτώσεις, το σύστημα

παρουσιάζει στον χρήστη μια ενημερωμένη άποψη της λίστας. Οπότε για να αποφύγουμε επανάληψη του ίδιου κώδικα, καλό θα ήταν η τελική αυτή παρουσίαση να είναι διαχωρισμένη από την διαδικασία προσθήκης ή διαγραφής.

6. Ένα άλλο ζήτημα είναι ότι ο σχεδιασμός των διεπαφών ηλεκτρονικού υπολογιστή είναι σχετικά πιο εξαρτημένος, από την εκάστοτε χρησιμοποιούμενη συσκευή αλληλεπίδρασης (PC, PDA, Cell Phone), από ότι η εκάστοτε υποστηριζόμενη επιχειρηματική λογική.
7. Επίσης η δημιουργία αυτοματοποιημένων δοκιμών για εφαρμογές διεπαφών ηλεκτρονικού υπολογιστή είναι αρκετά πιο δύσκολη και χρονοβόρα από τις αντίστοιχες δοκιμές στην επιχειρηματική λογική.

Πρόταση Υλοποίησης:

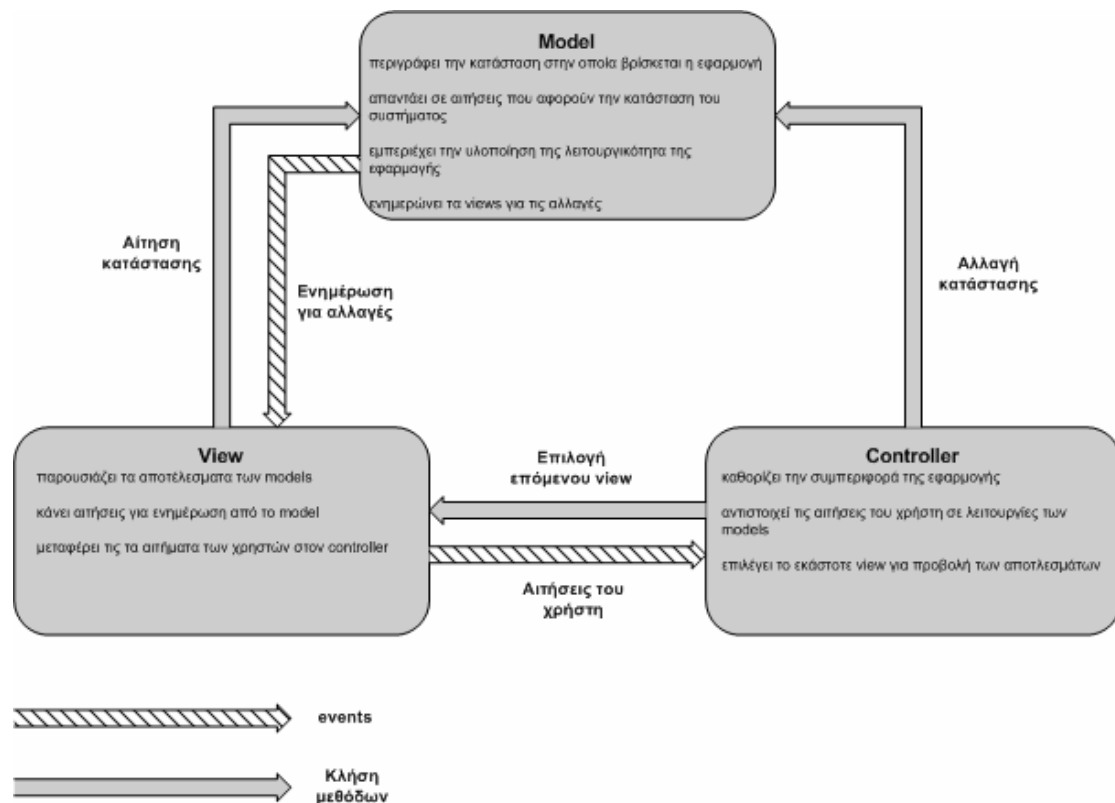
Το αρχιτεκτονικό μοντέλο Model-View-Controller (MVC) [12] [13] [14] είναι ένα ευρέως διαδεδομένο πρότυπο για την σχεδίαση διαδραστικών εφαρμογών. Διαχωρίζει την λειτουργικότητα της εφαρμογής ανάμεσα στα διάφορα αντικείμενα που είναι επιφορτισμένα με την διαχείριση και παρουσίαση της πληροφορίας, αποσκοπώντας στην μείωση του βαθμού συσχέτισης μεταξύ αυτών των αντικειμένων. Το μοντέλο διαχειρίζεται κλασικές εργασίες εφαρμογών όπως, είσοδο, επεξεργασία, και παρουσίαση

Αυτό που κάνει είναι να οργανώνει μια διαδραστική εφαρμογή σε τρία διαφορετικά επίπεδα. Το μοντέλο της εφαρμογής (Model), το οποίο είναι υπεύθυνο για την περιγραφή των δεδομένων και την επιχειρηματική λογική, την παρουσίαση των αποτελεσμάτων επεξεργασίας της αίτησης του χρήστη (View), και το επίπεδο που αφορά τις λειτουργίες που πρέπει να εκτελεστούν ανάλογα με την είσοδο του χρήστη -request dispatching and flow control-(Controller).

Model. Το **Model** διαχειρίζεται την συμπεριφορά και την ροή των δεδομένων στην εφαρμογή, απαντώντας σε αιτήσεις ανάκτησης πληροφορίας σχετικά με την κατάστασή του (συνήθως από τα Views) και απαντώντας επίσης σε οδηγίες αλλαγής κατάστασης (κυρίως από τον Controller).

View. Το View διαχειρίζεται την παρουσίαση των δεδομένων στον χρήστη.

Controller. Ο Controller ερμηνεύει τις εισόδους του χρήστη από το πληκτρολόγιο ή και το ποντίκι ή από κάποια άλλη μορφή εισόδου, όπως φωνή, ενημερώνοντας το Model ή/ και το View να αλλάξουν κατάσταση ανάλογα με την περίπτωση.



Σχήμα 5 Το αρχιτεκτονικό μοντέλο MVC

Τα πλεονεκτήματα:

1. Το MVC δίνει την δυνατότητα διαχωρισμού των σχεδιαστικών αναγκών (data persistence, presentation, control). Η σχεδιαστική αυτή επιλογή έχει διάφορα πλεονεκτήματα όσον αφορά την υλοποίηση πολύπλοκων και μεγάλων διαδραστικών εφαρμογών.
2. Μειώνεται η επανάληψη κώδικα.
3. Ομαδοποιείται ο έλεγχος και παράλληλα η εφαρμογή γίνεται εύκολα τροποποιήσιμη και επεκτάσιμη
4. Ορίζοντας καθαρά τις ευθύνες των κλάσεων που συμμετέχουν στην υλοποίηση, γίνεται πιο εύκολος ο εντοπισμός των λαθών και η εξουδετέρωσή τους.

5. Ελαχιστοποιείται η επίδραση που έχουν οι αλλαγές στο σύστημα, καθώς οι αλλαγές στο ένα κομμάτι του μοντέλου επιφέρει μικρή ή μηδενική ανάγκη για αλλαγές σε άλλα κομμάτια.
6. Αυξάνει την ικανότητα διατήρησης (maintainability) καθώς με τον διαχωρισμό των ρόλων, κάθε φορά που θα χρειαστεί να γίνουν αλλαγές σε κάποια λογική δεν θα χρειαστεί να κάνουμε τις αλλαγές αυτές σε πολλά σημεία του κώδικα που χρησιμοποιούν τη ίδια λογική.
7. Παρέχει ανεξαρτησία στους τύπους των χρηστών (client independence). Μπορούμε να έχουμε πολλές μορφές παρουσίασης των δεδομένων με την ίδια επιχειρηματική λογική.
8. Ο διαχωρισμός επιτρέπει και τον καταμερισμό της εργασίας σε άτομα εξειδικευμένα σε καθένα από τα κομμάτια αυτά.
9. Τέλος ο διαχωρισμός των επιπέδων και λειτουργιών της εφαρμογής επέτρεψε την χρήση των λεγόμενων προτύπων σχεδίασης στα διάφορα επίπεδα της εφαρμογής. Αυτά περιγράφονται παρακάτω.

3.1.1 Πρότυπα σχεδίασης

Τα πρότυπα σχεδίασης (design patterns) [14] περιγράφουν αποδεδειγμένες λύσεις σε προβλήματα που μπορεί να εμφανίζονται συχνά. Αυτά εκμεταλλεύονται τις γνώσεις και την εμπειρία των σχεδιαστών λογισμικού και τις περισσότερες φορές αναφέρονται σε συγκεκριμένα προβλήματα, αλλά μπορούν πολλές φορές να συνδυαστούν με διάφορους τρόπους για να δώσουν λύσεις σε προβλήματα που αφορούν τη σχεδίαση ολοκληρωμένων συστημάτων. Για να αναφέρονται σε αυτά οι σχεδιαστές χρησιμοποιούν μια συγκεκριμένη ορολογία που τα περιγράφει. Παρακάτω δίνονται μερικά πρότυπα σχεδίασης, τα οποία χρησιμοποιήσαμε στην υλοποίηση του συστήματός μας.

Στην υλοποίηση του επιπέδου παρουσίασης (View) της εφαρμογής χρησιμοποιήσαμε τα πρότυπα composite view και view helper. Το πρότυπο composite view έχει να κάνει με την σχεδίαση των σελίδων παρουσίασης αποτελεσμάτων στον χρήστη, είτε οι σελίδες αυτές είναι HTML, είτε είναι VoiceXML. Το πρότυπο κάνει πιο εύκολη την διαχείριση των σελίδων παρουσίασης εισάγοντας μια φόρμα (template) που περιέχει τα στοιχεία που είναι κοινά σε όλες τις

σελίδες. Οι σελίδες αυτές περιέχουν έναν συνδυασμό στατικού και δυναμικού περιεχομένου. Το στατικό περιεχόμενο αποτελούν στοιχεία όπως η επικεφαλίδα της σελίδας (header) και το background. Το δυναμικό περιεχόμενο εξαρτάται κάθε φορά από την σελίδα που παρουσιάζεται, ενώ το στατικό παραμένει σταθερό σε όλες τις σελίδες. Το composite view διαχωρίζει και οργανώνει αυτά ακριβώς τα κοινά στοιχεία. Το πρότυπο view helper ενθυλακώνει την παρουσίαση και την ανάκτηση δεδομένων, απλοποιώντας και εκλεπτύνοντας την δομή του view. Σύμφωνα με την λογική του προτύπου, διαχωρίζουμε την παρουσίαση των δεδομένων, η οποία επικεντρώνεται στην διαμόρφωση της σελίδας και των δεδομένων που παρουσιάζονται στον χρήστη, από την λογική που ασχολείται με την πρόσβαση και ανάκτηση των δεδομένων. Στην περίπτωση μας view helpers είναι τα JSP tags για την μορφοποίηση και παρουσίαση των δεδομένων και τα Java Servlets [32] για την ανάκτησή τους.

Στο model επίπεδο της εφαρμογής κάναμε χρήση του προτύπου session facade. Αυτό συντονίζει τις εργασίες μεταξύ των συνεργαζόμενων αντικειμένων – συναρτήσεων, συνενώνοντας τις λειτουργίες της εφαρμογής σε μια απλοποιημένη διεπαφή που χρησιμοποιείται για να παρουσιαστούν τα αποτελέσματα στην δομή που τα ζήτησε. Έχει την δυνατότητα να ενθυλακώνει και να κρύβει την πολυπλοκότητα των υλοποιούμενων κλάσεων που συνεργάζονται, αρκετές φορές με πολύπλοκους τρόπους, ενώ απομονώνει τις δομές που το καλούν από πιθανές αλλαγές στα αντικείμενα που υλοποιούν τις διάφορες εργασίες.

Στο κομμάτι που έχει να κάνει με την ανάκτηση δεδομένων χρησιμοποιήσαμε το πρότυπο value object. Το πρότυπο αυτό διευκολύνει την ανταλλαγή δεδομένων μεταξύ των διαφόρων επιπέδων της εφαρμογής, μειώνοντας έτσι το κόστος της κατανεμημένης επικοινωνίας. Σε μια κλήση από το ένα επίπεδο στο άλλο μπορεί να χρησιμοποιηθεί ένα μόνο value object για την ανάκτηση ενός συνόλου δεδομένων, το οποίο είναι ύστερα διαθέσιμο τοπικά στην δομή που το κάλεσε.

Τέλος, στο επίπεδο του controller, κάναμε χρήση του σχεδιαστικού προτύπου business delegate. Σύμφωνα με αυτό το πρότυπο, αποδεσμεύουμε το επίπεδο παρουσίασης από αυτό που εκτελεί το κομμάτι της επιχειρηματικής λογικής. Προσφέρει δηλαδή ένα φιλικό interface των μεθόδων που υλοποιεί το model επίπεδο της εφαρμογής προς το επίπεδο παρουσίασης έτσι ώστε το τελευταίο να μην ασχολείται με θέματα που αφορούν την ανάκτηση δεδομένων και διαχείριση λαθών.

Στο Appendix C – Technical Issues - **Error! Reference source not found.** δίνεται λεπτομερής περιγραφή της εφαρμογής του μοντέλου MVC στο σύστημά μας μαζί με κάποιο αντιπροσωπευτικό παράδειγμα λειτουργικότητας της υπηρεσίας..

3.2 *Υλοποίηση Front End*

3.2.1 Η γλώσσα VoiceXML

Με την υλοποίηση του Front End παράγουμε τον απαραίτητο κώδικα που είναι απαραίτητος για την επικοινωνία του υπολογιστή με τον εκάστοτε χρήστη. Η παραγωγή του κώδικα γίνεται με οδηγό τις απαιτήσεις που παρουσιάσαμε στον ορισμό του προβλήματος. Το κομμάτι της εφαρμογής που υλοποιεί την λογική των διαλόγων μπορεί να αναπτυχθεί σε μια μεγάλη ποικιλία γλωσσών προγραμματισμού όπως C, C++, ή ακόμα και Java [17]. Πολλές φορές, το κομμάτι αυτό είναι κωδικοποιημένο σε μια ειδική γλώσσα που σχετίζεται με κάποια ειδική πλατφόρμα διαδραστικής φωνητικής ανάδρασης (IVR). Πολλές τέτοιες πλατφόρμες έχουν δικές τους γλώσσες και εργαλεία, τα οποία οι σχεδιαστές πρέπει να χρησιμοποιούν για την υλοποίηση φωνητικών εφαρμογών. Η ισχύουσα τάση, ωστόσο τα τελευταία χρόνια, είναι η χρήση της γλώσσας VoiceXML [3], η οποία είναι μια γλώσσα markup language, για την υλοποίηση φωνητικών εφαρμογών. Είναι αρκετά όμοια με την HTML, ενώ είναι μια γλώσσα με επίκεντρο το λόγο (speech-centric). Κύριος σκοπός της είναι να προσφέρει τα πλεονεκτήματα των εφαρμογών βασισμένων στο Web και του λεγόμενου Content Delivery στα IVR συστήματα. Επιτρέπει την ενσωμάτωση των εφαρμογών φωνητικών υπηρεσιών στις υπάρχουσες εφαρμογές ανάκτησης δεδομένων, χρησιμοποιώντας την γνωστό πρότυπο σχεδίασης, πελάτη – εξυπηρετητή (client-server). Ποιο αναλυτικά οι σκοποί της φαίνονται παρακάτω

- Ελαχιστοποιεί την αλληλεπίδραση μεταξύ πελάτη και εξυπηρετητή παρέχοντας πολλαπλές αλληλεπιδράσεις ανά αρχείο.
- Απαλλάσσει τους προγραμματιστές από λεπτομέρειες προγραμματισμού χαμηλού επιπέδου και από το είδος της πλατφόρμας που πρόκειται να υλοποιηθεί.

- Διαχωρίζει τον κώδικα αλληλεπίδρασης (VoiceXML) από τον κώδικα που υλοποιεί την λογική εξυπηρέτησης των αιτήσεων του χρήστη.
- Προωθεί την συμβατότητα των υπηρεσιών μεταξύ των πλατφόρμων εφαρμογής.
- Είναι εύκολη η χρήση της για την υλοποίηση απλών αλληλεπιδράσεων, ενώ παρέχει και εργαλεία για την υποστήριξη πιο πολύπλοκων δομών διαλόγων.

Η VoiceXML είναι μια γλώσσα σχεδιασμένη για την δημιουργία ακουστικών διαλόγων (audio dialogs) με χρήση διάφορων τεχνολογιών όπως:

- σύνθεσης φωνής(synthesized speech)
- ψηφιακών ακουστικών σημάτων (digitized audio)
- αναγνώρισης φωνής-ομιλίας και DTMF εισόδου
- ηχογράφησης ομιλίας, τεχνολογιών τηλεφωνίας (μεταφορά κλήσης)
- και mixed initiative διαλόγων

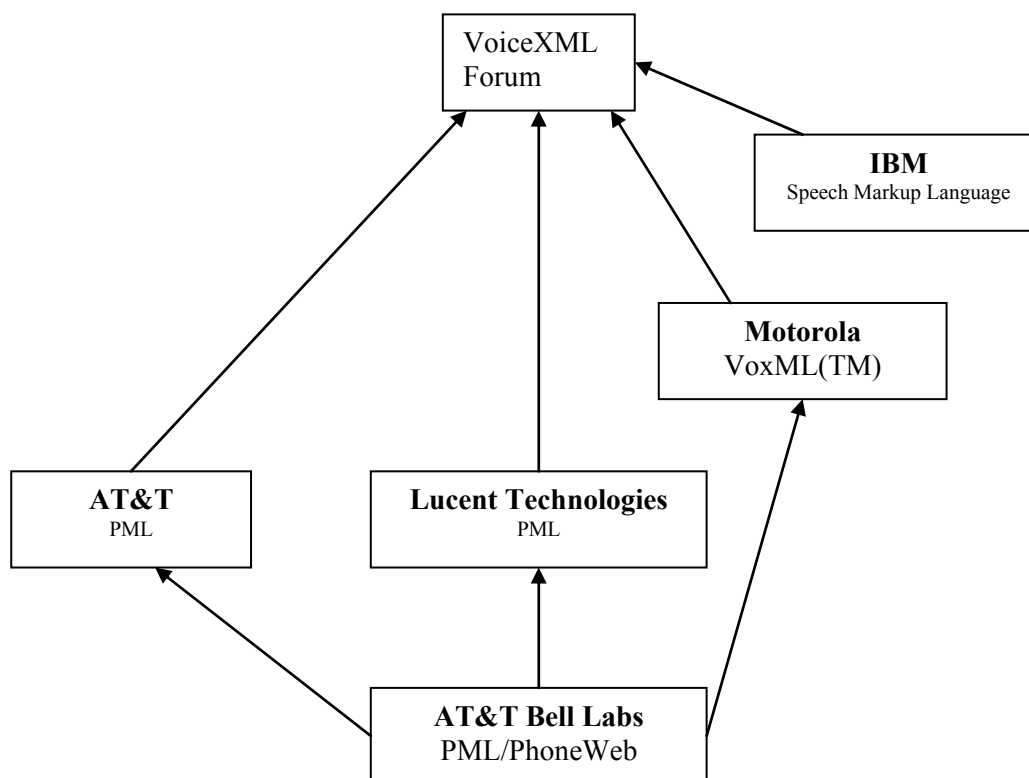
Χαρακτηριστικό της VoiceXML είναι ότι προσφέρει διάφορες ευκολίες σχεδίασης, όπως είναι τα menus (για παρουσίαση των διάφορων επιλογών στον χρήστη) ή τα forms (για την συλλογή στοιχείων από τον χρήστη, με ταυτόχρονη διαχείριση πολλών στοιχείων δεδομένων σε μια μόνο form).

Η VoiceXML είναι βασισμένη στην γλωσσική δομή της XML, και για αυτό είναι αρκετά οικεία η χρήση της σε πολλούς προγραμματιστές. Ξεκίνησε το 1995 ως μια γλώσσα προγραμματισμού για την σχεδίαση διαλόγων βασισμένη στην XML [18] [19] και με σκοπό την απλοποίηση της διαδικασίας αναγνώρισης της ομιλίας, κατά την διαδικασία ανάπτυξης φωνητικών εφαρμογών. Την έρευνα αυτή ανέλαβε η American Telephone and Telegraph Company (AT&T) στα πλαίσια ερευνητικού έργου με την ονομασία Phone Markup Language (PML). Σε σύντομο χρονικό διάστημα ερευνητικές ομάδες στην AT&T, την Lucent [21] και Motorola [22] άρχισαν να δουλεύουν πάνω σε δικές τους γλώσσες όμοιες με την PML

Το 1998, το World Wide Web Consortium (W3C) [23] διοργάνωσε ένα συνέδριο σχετικά με τους voice browsers. Την περίοδο εκείνη η AT&T και η Lucent

είχαν ήδη προχωρήσει στην υλοποίηση εντελώς διαφορετικών παραλλαγών της PML, ενώ η Motorola είχε ήδη παρουσιάσει την VoxML και η IBM [25] υλοποιούσε την SpeechML. Αλλά και πολλοί άλλοι, συμμετέχοντες στο συνέδριο, είχαν προχωρήσει στην υλοποίηση παρόμοιων γλωσσών σχεδιασμού διαλόγων, όπως για παράδειγμα η TalkML της HP και η Voice HTML της PipeBeach.

Το VoiceXML Forum δημιουργήθηκε τότε από τις AT&T, IBM, Lucent, και Motorola με την μορφή κοινοπραξίας, που σκοπός της ήταν ο καθορισμός μια κοινής γλώσσας προγραμματισμού για τον σχεδιασμό διαλόγων στις φωνητικές εφαρμογές. Η επιλογή της XML ως βάση της προσπάθειάς αυτής έγινε λόγω της πεποίθησης των συντελεστών ότι αυτή ήταν η κατεύθυνση την οποία ακολουθούσε η τεχνολογία.



Σχήμα 6. History of VoiceXML

Το 2000, το VoiceXML Forum ανακοίνωσε την έκδοση της VoiceXML 1.0. Αμέσως μετά, η VoiceXML 1.0 έγινε αποδεκτή από την W3C ως η βάση για την δημιουργία ενός νέου διεθνούς στάνταρ για την υλοποίηση φωνητικών εφαρμογών. Η VoiceXML 2.0 είναι το αποτέλεσμα των εργασιών πάνω σε αυτή τη βάση σε συνδυασμό με την συνεισφορά των εταιρειών που συμμετέχουν στο W3C, άλλων ομάδων που ανήκουν στην W3C καθώς και του απλού κοινού.

Βασική Αρχή της VoiceXML

Ένα VoiceXML αρχείο (ή ένα σύνολο από διασυνδεδεμένα αρχεία που συγκροτούν μια εφαρμογή) αποτελεί μια μηχανή πεπερασμένων καταστάσεων, που υλοποιεί έναν διάλογο. Ο χρήστης βρίσκεται κάθε φορά σε μια τέτοια κατάσταση, ή αλλιώς διάλογο. Κάθε διάλογος καθορίζει και τον επόμενο διάλογο, στον οποίο θα μεταβούμε. Οι μεταβάσεις ορίζονται μέσω των Universal Resource Identifiers (URIs), που προσδιορίζουν το επόμενο αρχείο ή διάλογο που θα χρησιμοποιηθεί. Αν το URI δεν ορίζει κάποιο συγκεκριμένο αρχείο, τότε λαμβάνεται το παρόν αρχείο. Αν δεν αναφέρεται σε κάποιο διάλογο, τότε λαμβάνουμε τον πρώτο διάλογο μέσα στο αρχείο. Η εκτέλεση της εφαρμογής τερματίζεται όταν δεν ορίζεται άλλο αρχείο προς μετάβαση ή όταν δηλώνεται ρητά ο τερματισμός της εφαρμογής.

Dialogs and Subdialogs

Υπάρχουν δύο είδη διαλόγων, τα forms και τα menus. Τα forms περιγράφουν μια αλληλεπίδραση που συλλέγει δεδομένα και τιμές για ένα σύνολο μεταβλητών που ανήκουν στο form item. Κάθε πεδίο (field) μπορεί να χαρακτηρίζεται από μια γραμματική, η οποία καθορίζει τις επιτρεπτές εισόδους για το πεδίο αυτό. Αν υπάρχει κάποια γραμματική στο επίπεδο της form τότε μπορεί να χρησιμοποιηθεί για να γεμίσουν διάφορα πεδία (fields) με μία μόνο προσπάθεια ομιλίας. Ένα menu παρουσιάζει στον χρήστη ένα σύνολο εντολών και στην συνέχεια μεταβαίνει σε διάφορους διαλόγους ανάλογα με την είσοδο του χρήστη.

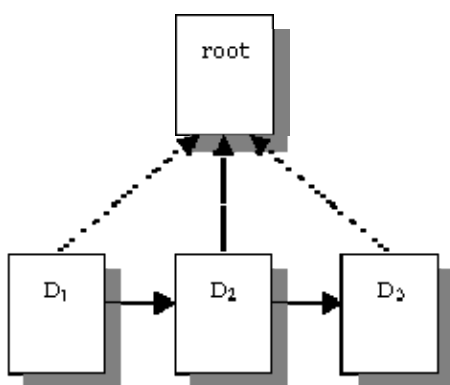
Ένας υποδιάλογος είναι μια μορφή κλήσης συνάρτησης, επειδή παρέχει την δυνατότητα της εισαγωγής μιας νέας αλληλεπίδρασης και της επιστροφής στον αρχικό διάλογο μετά την συλλογή κάποιων δεδομένων. Οι τιμές των μεταβλητών, οι γραμματικές και πληροφορίες σχετικά με την κατάσταση του αρχικού διαλόγου σώζονται και είναι διαθέσιμα μετά την επιστροφή από τον υποδιάλογο. Χρήση υποδιαλόγων μπορούμε να έχουμε, για παράδειγμα, για την εκτέλεση μιας εργασίας που απαιτεί να κάνουμε κάποιο query στη βάση ή για να κάνουμε κάποια εργασία που θα μπορούσε να επαναληφθεί σε διάφορα άλλα σημεία ή αρχεία-διαλόγους μέσα στην ίδια εφαρμογή.

Sessions

Μια σύνοδος (session) ξεκινάει όταν ο χρήστης ξεκινάει την αλληλεπίδραση με το VoiceXML Interpreter Context, συνεχίζει καθώς τα αρχεία φορτώνονται και επεξεργάζονται και τελειώνει όταν το ζητήσει ο χρήστης, ή όταν το ορίζει το αρχείο ή το Interpreter Context.

Εφαρμογή (Application)

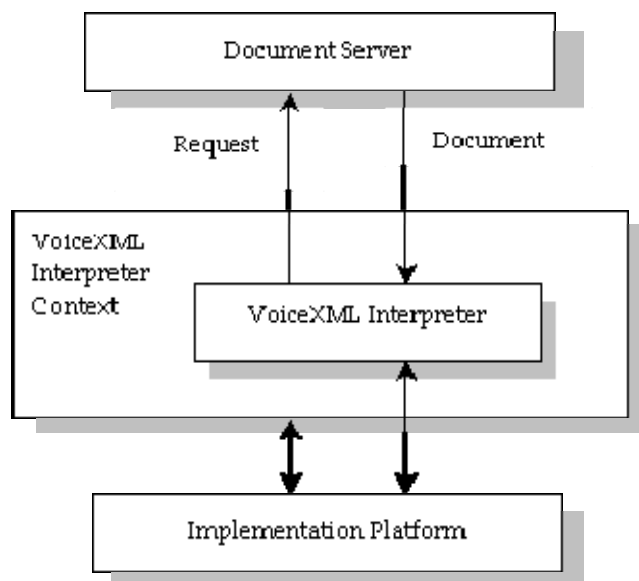
Μια εφαρμογή αποτελεί ένα σύνολο αρχείων που μοιράζονται το ίδιο αρχείο αρχής εφαρμογής (application root document). Όποτε ο χρήστης αρχίζει να αλληλεπιδρά με την εφαρμογή, το αρχείο αρχής εφαρμογής είναι και αυτό ενεργό. Αυτό παραμένει ενεργό καθώς ο χρήστης μεταβαίνει στα αρχεία της ίδιας εφαρμογής, και γίνεται ανενεργό όταν αυτό μεταβεί σε ένα αρχείο που δεν ανήκει στην εφαρμογή. Όσο το αρχείο αρχής εφαρμογής είναι ενεργό, οι μεταβλητές του είναι διαθέσιμες στα αρχεία της εφαρμογής, ενώ και οι γραμματικές του είναι ενεργές καθ' όλη την διάρκεια της εφαρμογής. Μια ενδεικτική παρουσίαση μιας εφαρμογής δίνεται παρακάτω.



Σχήμα 7 Μια ενδεικτική εφαρμογή σε VoiceXML

Αρχιτεκτονικό Μοντέλο

Το αρχιτεκτονικό μοντέλο δείχνει την μορφή με τη οποία είναι δομημένη η επεξεργασία των αρχείων VoiceXml που υλοποιούν μια φωνητική εφαρμογή.



Σχήμα 8 Αρχιτεκτονικό μοντέλο της VoiceXML

Ο Document Server επεξεργάζεται αιτήσεις (requests) από την εφαρμογή πελάτη, τον VoiceXML Interpreter, μέσω του VoiceXML Interpreter Context. Μετά την απαιτούμενη επεξεργασία, ο Document Server απαντάει παράγοντας νέα VoiceXML αρχεία, τα οποία επεξεργάζονται από τον VoiceXML Interpreter. Το VoiceXML Interpreter Context μπορεί να ελέγχει τις εισόδους του χρήστη παράλληλα με τον VoiceXML Interpreter. Για παράδειγμα, ένας VoiceXML Interpreter μπορεί να ακούει συνεχώς για κάποια ειδική φράση διαφυγής του χρήστη που τον οδηγεί σε κάποιο ανώτερο επίπεδο στην εφαρμογή, ενώ ένας άλλος μπορεί να ακούει για κάποιες για φράσεις διαφυγής που αλλάζουν τις προτιμήσεις του χρήστη όπως ένταση της φωνής ή τα χαρακτηριστικά της συνθετικής φωνής (text-to-speech TTS).

Η πλατφόρμα εφαρμογής (Implementation Platform) ελέγχεται από το VoiceXML Interpreter Context και από το VoiceXML Interpreter. Για παράδειγμα, σε ένα IVR σύστημα, το VoiceXML Interpreter Context έχει την ευθύνη για την αποδοχή μιας κλήσης και την ανεύρεση του αρχικού αρχείου της εφαρμογής, ενώ ο VoiceXML Interpreter διεξάγει τον διάλογο από εκείνη την στιγμή και μετά. Η πλατφόρμα εφαρμογής εκτελεί εργασίες ανάλογα με τις εισόδους του χρήστη (φωνητικά ή μέσω πληκτρολογίου) και τις αντιδράσεις του συστήματος (timer expiration). Μερικές από τις εργασίες αυτές εκτελούνται μέσω του ίδιου του

VoiceXML Interpreter, έτσι όπως καθορίζεται στο VoiceXML αρχείο, ενώ άλλες μέσω του VoiceXML Interpreter Context.

Στο Appendix C – Technical Issues δίνεται ένας πίνακας με όλα τα στοιχεία που αποτελούν την προγραμματιστική γλώσσα VoiceXML μαζί με μια σύντομη περιγραφή τους, όπως παρουσιάζεται στην επίσημη σελίδα της W3C.

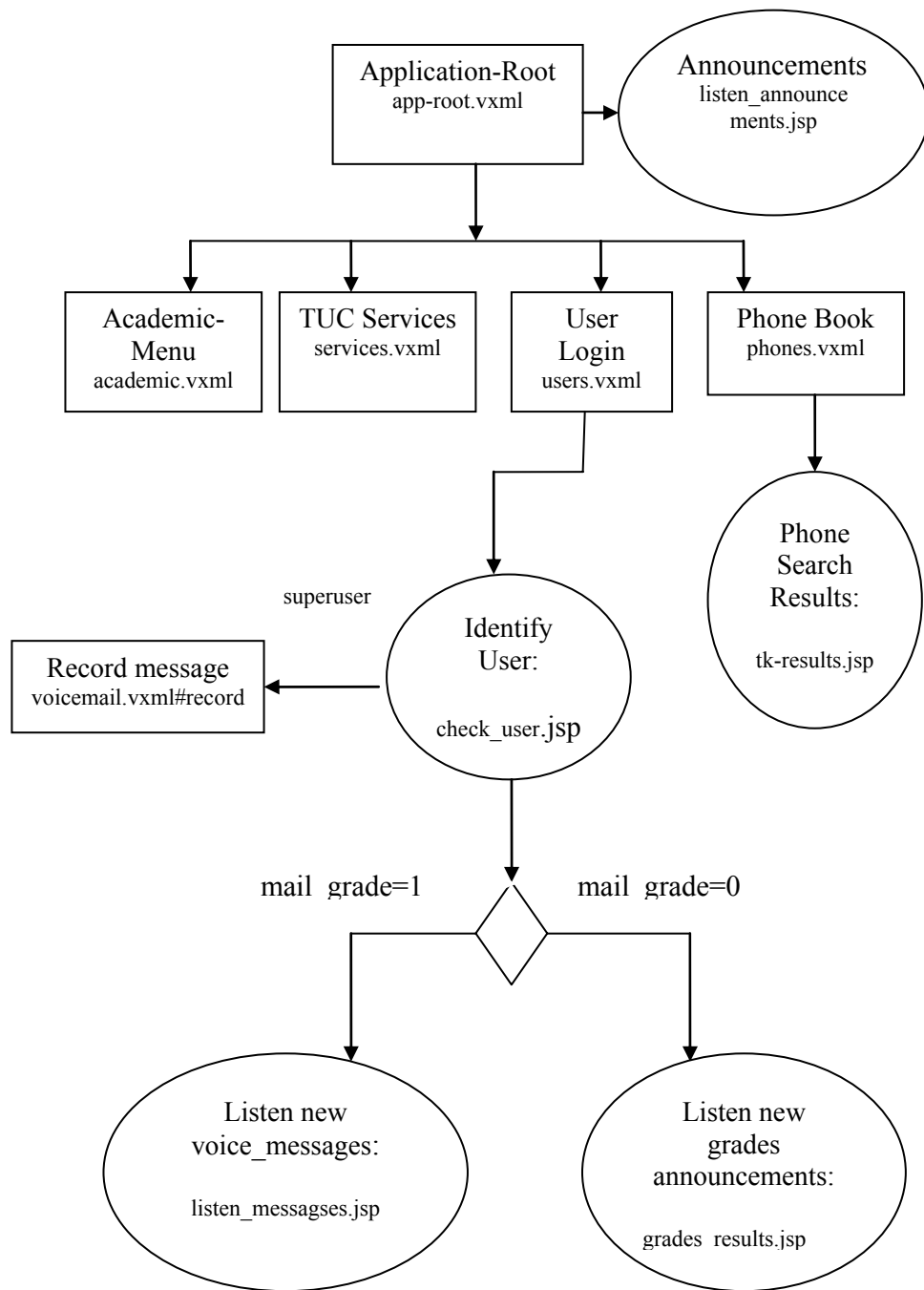
3.2.2 Η Φωνητική Εφαρμογή

Η φωνητική εφαρμογή είναι το μέρος εκείνο της εφαρμογής, το οποίο θα είναι διαθέσιμο στους χρήστες μέσω τηλεφώνου. Η υλοποίηση της φωνητικής εφαρμογής περιλαμβάνει την σχεδίαση και υλοποίηση των διαλόγων καθώς και των σχετικών γραμματικών.

3.2.2.1 Σχεδίαση - Υλοποίηση Διαλόγων

Για τις ανάγκες της εφαρμογής κάναμε, όπως αναφέραμε, χρήση της VoiceXML, καθώς υπάρχει διδασκαλία της συγκεκριμένης γλώσσας παίρνοντας υπόψη τα πλεονεκτήματα που προσφέρει. Η έκδοση που χρησιμοποιήσαμε ήταν η VoiceXML 2.1, αν και δεν παρεκκλίνει κατά πολύ από την VoiceXML 2.0, παρέχει ωστόσο, κάποιες αρκετά χρήσιμες επιλογές σε σύγκριση με την προηγούμενη έκδοση. Αυτές αφορούν κυρίως στην κλήση των δυναμικών γραμματικών, χρησιμοποιώντας JavaScript expressions για την περιγραφή URIs, δίνοντας την δυνατότητα να περιλαμβάνουμε όλες τις επιθυμητές μεταβλητές.

Η δομή της εφαρμογής και παίρνοντας ως βάση τον σχεδιασμό που είχαμε κάνει στην προηγούμενη ενότητα φαίνεται στο Σχήμα 9:

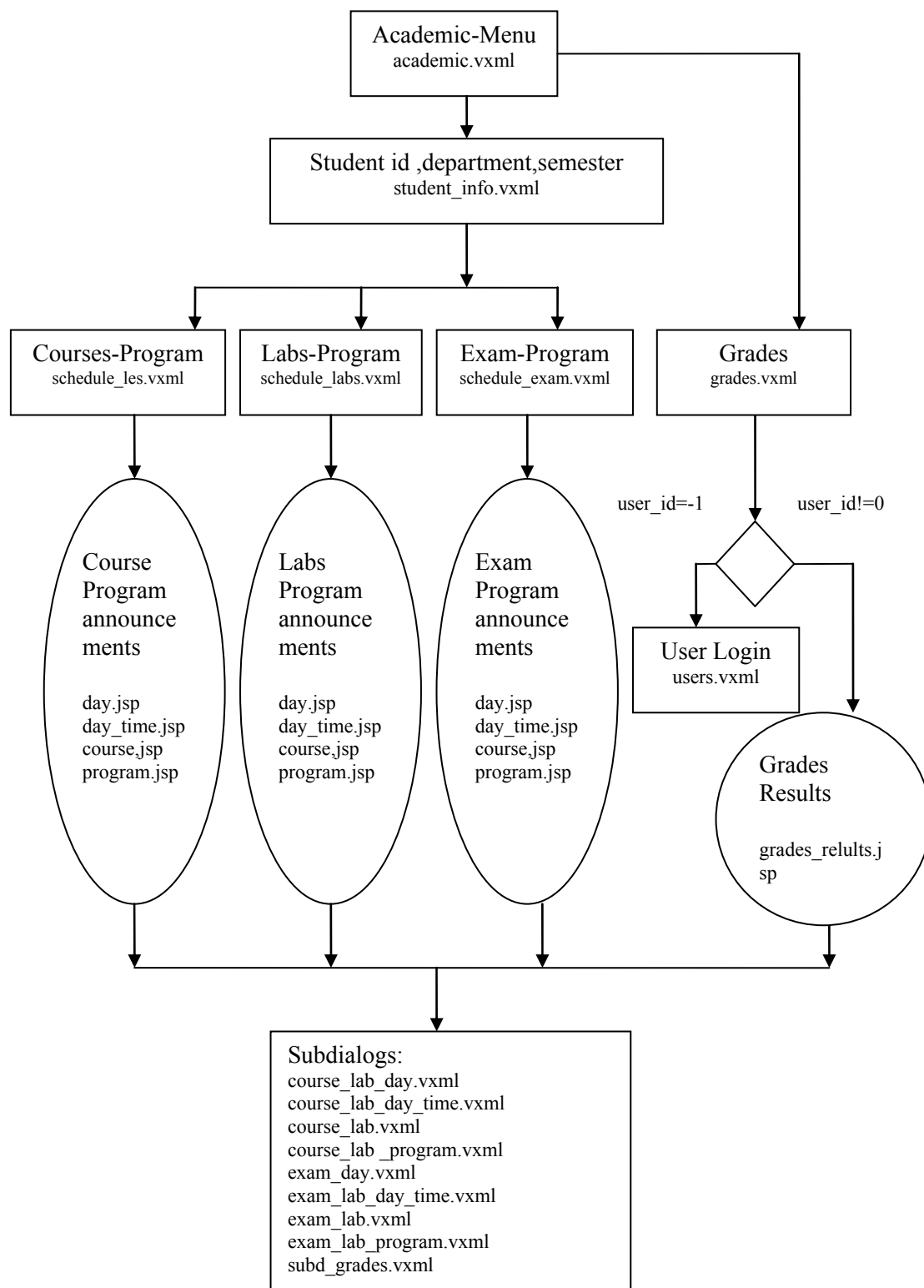


Σχήμα 9 Λειτουργικότητα Εφαρμογής

Το αρχείο app-root.vxml αποτελεί το αρχείο αρχής της εφαρμογής (application document –ή root document) της εφαρμογής και εκεί βρίσκονται όλες οι μεταβλητές και γραμματικές καθολικής συμπεριφοράς (global variables & global grammars). Αυτές ενεργοποιούνται όταν ξεκινάει η εφαρμογή και γίνονται ανενεργές όταν σταματάει η εκτέλεσή της. Για παράδειγμα, η μεταβλητή user_id, κρατάει την τιμή του διακριτικού id που έχει κάθε χρήστης στην βάση. Παίρνει τιμή όταν ο χρήστης κάνει login στην εφαρμογή και παραμένει ενεργή καθ' όλη την διάρκεια

εκτέλεσης των διαλόγων της εφαρμογής. Αντίστοιχο παράδειγμα για τις καθολικές γραμματικές αποτελούν οι γραμματικές που αναφέρονται στις παρακινήσεις που πυροδοτούν τις επιλογές «επανάληψη» και «βοήθεια», οι οποίες βρίσκονται στο `app-root.vxml` και είναι διαθέσιμες στον χρήστη σε κάθε σημείο της εφαρμογής. Σε αυτόν τον διάλογο, μετά από έναν σύντομο χαιρετισμό προς τον χρήστη, η εφαρμογή ελέγχει τη βάση για νέες ανακοινώσεις και αν υπάρχουν τις παρουσιάζει στον χρήστη. Σε διαφορετική περίπτωση, ανακοινώνει στον χρήστη τις επιλογές που έχει, δηλαδή κάνει μια σύντομη παρουσίαση της λειτουργικότητας της εφαρμογής. Από αυτό το σημείο, ο χρήστης ανάλογα με την είσοδό του μπορεί να οδηγηθεί σε μια από τις ενότητες Ακαδημαϊκά Θέματα (`academic.vxml`), Υπηρεσίες του Πολυτεχνείου Κρήτης (`services.vxml`), Εγγραφή στο σύστημα (`users.vxml`) και Τηλεφωνικός Κατάλογος (`phones.vxml`), όπως φαίνεται και στο Σχήμα 99.

Σε περίπτωση που ο χρήστης επιλέξει να κάνει login, μεταβαίνει στο `users.vxml`, όπου υπάρχει διάλογος που προτρέπει τον χρήστη να εισάγει έναν κωδικό πρόσβασης. Στην συνέχεια ελέγχουμε αν αυτός είναι `super_user` ή απλός `user`. Στην πρώτη περίπτωση, ο χρήστης είναι κάποιος καθηγητής και για αυτό μεταβαίνουμε στον υπο-διάλογο `record` του `voicemail.vxml` αρχείου, για να ξεκινήσουμε την διαδικασία ηχογράφησης κάποιου νέου φωνητικού μηνύματος. Στην δεύτερη περίπτωση, ο χρήστης είναι κάποιος φοιτητής και για αυτό ελέγχουμε αν έχει κάποιο νέο φωνητικό μήνυμα ή κάποια ανακοίνωση νέας βαθμολογίας μαθήματος ή εργαστηρίου. Αν ο χρήστης επιλέξει να χρησιμοποιήσει τον τηλεφωνικό κατάλογο που προσφέρει η εφαρμογή τότε το επόμενο αρχείο που καλείται από την εφαρμογή είναι το `phones.vxml`, στο οποίο ο χρήστης καλείται να εισάγει κάποιο όνομα προσώπου, το οποίο δραστηριοποιείται στο χώρο του πολυτεχνείου ή ακόμα και εργαστηρίων και υπηρεσιών. Τότε το σύστημα, αφού ζητήσει από τον χρήστη άμεση επιβεβαίωση για το όνομα που εισήγαγε, του παρουσιάζει τα αποτελέσματα της αναζήτησης στην βάση δεδομένων. Δυστυχώς επειδή η γραμματική που αντιστοιχεί σε αυτό το σημείο της εφαρμογής δημιουργείται δυναμικά με την χρήση των καταχωρήσεων που υπάρχουν στην βάση δεδομένων, σε περίπτωση που ο χρήστης κάνει αναζήτηση με κάποιο όνομα που δεν υπάρχει στην βάση ή δεν εισάγει σωστά την είσοδο, τα λάθη αναγνώρισης είναι μεγάλα. Για αυτό στην τρίτη αποτυχημένη προσπάθεια αναγνώρισης και αν το σύστημα δεν έχει καταλάβει την είσοδο του χρήστη θεωρούμε ότι η καταχώρηση που αυτός ψάχνει δεν υπάρχει.



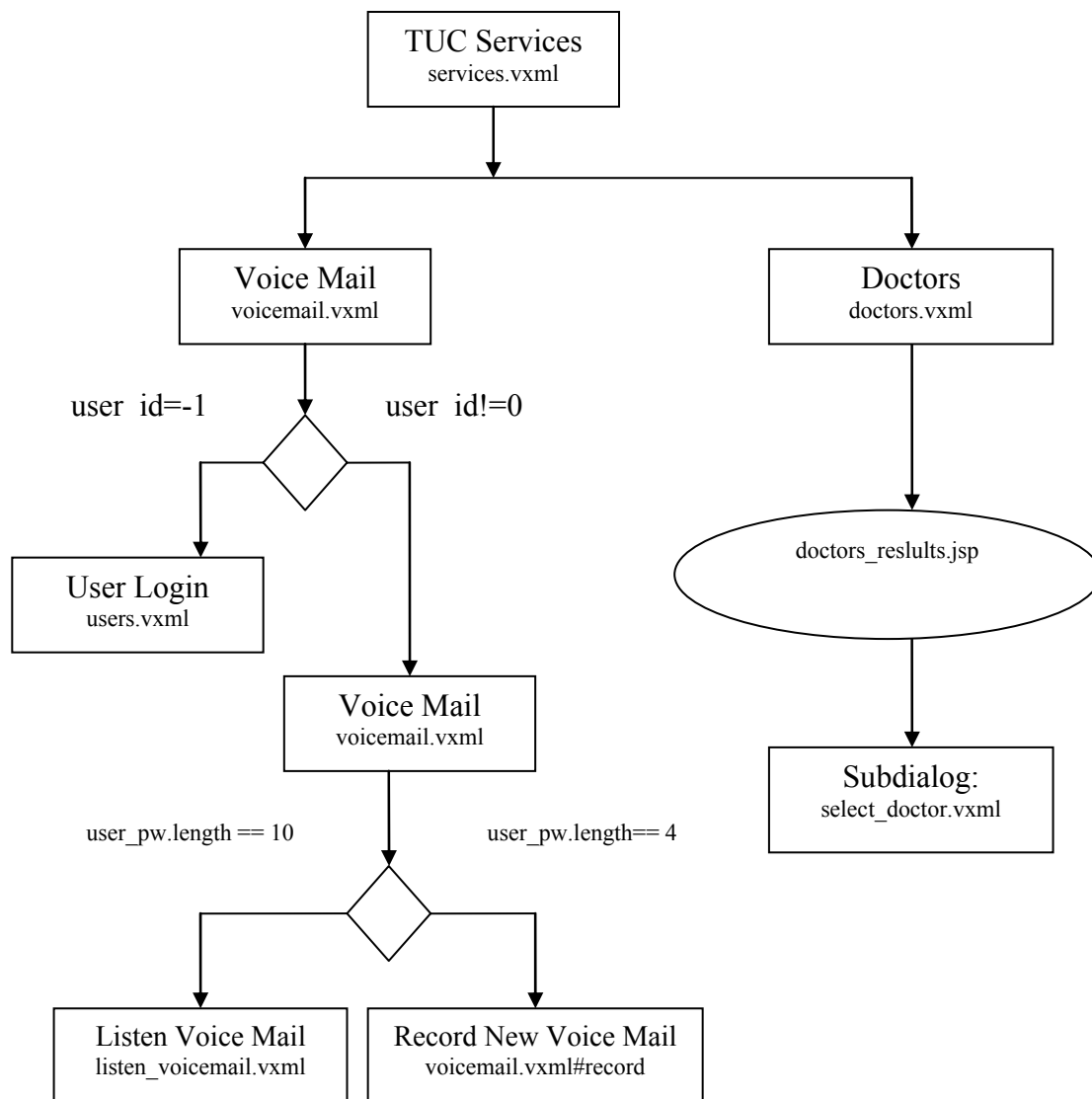
Σχήμα 10 Λειτουργικότητα Εφαρμογής

Το academic.vxml, το οποίο περιέχει την υλοποίηση της ενότητας Ακαδημαϊκά Θέματα, αποτελεί ένα υπο-μενού και βρίσκεται στο τρίτο επίπεδο της δομής της

εφαρμογής. Εδώ ο χρήστης έχει την δυνατότητα να ακούσει τα προγράμματα μαθημάτων, εργαστηρίων, και εξεταστικής όλων των τμημάτων του πολυτεχνείου και των ακαδημαϊκών εξαμήνων, καθώς και βαθμολογίες μαθημάτων.

Αφού έχουμε συλλέξει τα απαραίτητα στοιχεία που αφορούν τον χρήστη, όπως εξάμηνο, τμήμα ή αν έχει κάνει login, μεταφερόμαστε σε ένα από τα `schedule_les.vxml`, `schedule_les.vxml`, και `schedule_les.vxml`. Εκεί ο χρήστης εισάγει στοιχεία αναζήτησης που αφορούν το πρόγραμμα που θέλει να ακούσει. Τέτοια στοιχεία είναι η μέρα, η ημερομηνία, η ώρα, η μέρα ή η ημερομηνία και η ώρα, το συνολικό πρόγραμμα ή και το όνομα του μαθήματος ή του εργαστηρίου. Αφού γίνεται η επιβεβαίωση των στοιχείων που εισήγαγε ο χρήστης, το σύστημα ανατρέχει στην βάση και κάνει αναζήτηση με τα στοιχεία αυτά και προχωράει στην παρουσίαση των αποτελεσμάτων.

Αν ο χρήστης επιλέξει να ακούσει τις βαθμολογίες μαθημάτων και εργαστηρίων, τότε το επόμενο αρχείο που πρόκειται να επεξεργαστεί είναι το `grades.vxml`. Πρώτα ελέγχουμε αν ο χρήστης έχει κάνει login. Αν αυτό δεν συμβαίνει τότε τον παροτρύνουμε να πάει στον αντίστοιχο διάλογο για να μπορέσει να εγγραφεί στο σύστημα για να έχει την δυνατότητα να ακούσει τις βαθμολογίες. Αφού βεβαιωθούμε ότι ο χρήστης έχει κάνει login και διαθέτει κάποιο έγκυρο `user_id`, προχωρούμε στην ανακοίνωση των νέων βαθμολογιών που έχουν καταχωρηθεί στην βάση δεδομένων. Η ανακοίνωση των αποτελεσμάτων γίνεται μέσω υπο-διαλόγων, όπως φαίνεται στο Σχήμα 10 Λειτουργικότητα Εφαρμογής. Η αναζήτηση στην βάση δεδομένων επιστρέφει κάθε φορά μια λίστα με μαθήματα μαζί με όλες τις σχετικές πληροφορίες ανάλογα με την ερώτηση του χρήστη. Στους διαλόγους αυτούς, υπάρχει μια υλοποιημένη δομή που επιτρέπει στον χρήστη να διατρέχει την λίστα αυτή ακούγοντας τα επιθυμητά αποτελέσματα αναζήτησης.



Σχήμα 11 Λειτουργικότητα Εφαρμογής

Το αρχείο services.vxml αναφέρεται στο κομμάτι που έχει να κάνει με υπηρεσίες του Πολυτεχνείου Κρήτης που προσφέρονται μέσα από την εφαρμογή. Αυτές είναι το Φωνητικό Ταχυδρομείο (voicemail.vxml) και η Υπηρεσία Αναζήτησης Γιατρών (doctors.vxml).

Αν ο χρήστης επιλέξει το φωνητικό ταχυδρομείο, τότε πάλι ελέγχουμε αν έχει κάνει login, όπως κάναμε στην περίπτωση των βαθμολογιών. Αν αυτό ισχύει, και έχουμε κάποιο έγκυρο user_id, τότε το επόμενο βήμα είναι να ελέγξουμε, αν ο χρήστης είναι super_user ή απλός user. Την περίπτωση του super_user την αναφέραμε παραπάνω. Αν ο χρήστης είναι user τότε προχωράμε στην ανακοίνωση των νέων φωνητικών μηνυμάτων του. Στην συνέχεια, και αν αυτός το επιθυμεί, το σύστημα του ανακοινώνει παλιότερα φωνητικά μηνύματα που υπάρχουν στην βάση

με βάση το όνομα του μαθήματος ή του εργαστηρίου. Ο χρήστης δεν μπορεί να ακούσει φωνητικά μηνύματα για μαθήματα, στα οποία δεν έχει δηλωθεί.

Στην Υπηρεσία Αναζήτησης Γιατρών ο χρήστης μπορεί να ψάξει για κάποιον γιατρό, ο οποίος έχει σύμβαση με το Πολυτεχνείο. Η αναζήτηση μπορεί να γίνει με βάση το πλήρες όνομα του γιατρού ή την ειδικότητά του. Το σύστημα, αφού αναγνωρίσει την είσοδο του χρήστη, ζητάει να επιβεβαιώσει ότι αυτό που έχει καταλάβει είναι σωστό, και στη συνέχεια του παρουσιάζει τα αποτελέσματα. Ο χρήστης, καθώς διατρέχει την λίστα αποτελεσμάτων, μπορεί να ζητήσει λεπτομέρειες και να μάθει κατά αυτόν τον τρόπο, εκτός από το τηλέφωνο του γιατρού, και την διεύθυνση του ιατρείου καθώς και ώρες λειτουργίας

3.2.2.2 Υλοποίηση Γραμματικών

Σκοπός των γραμματικών αναγνωριστών ομιλίας είναι να δίνεται η δυνατότητα στις φωνητικές εφαρμογές να υποδεικνύουν σε ένα αναγνωριστή ομιλίας τις λέξεις και τις εκφράσεις που θα πρέπει να ακούσει. Δηλαδή :

- Λέξεις που μπορεί να πει ο χρήστης
- Εκφράσεις στις οποίες μπορεί να εμφανίζονται οι λέξεις αυτές
- Τρόποι με τους οποίους προφέρουμε την κάθε λέξη

Η σωστή ανάπτυξη των γραμματικών αποτελεί βασικό κομμάτι της δημιουργίας αποτελεσματικών εφαρμογών φωνής. Είναι δε άμεσα συνδεδεμένο με τη στρατηγική σχεδίασης των διαλόγων, την οποία μελετήσαμε στην προηγούμενη ενότητα, αλλά και με τη σχεδίαση αποτελεσματικών κειμένων παρακίνησης εισόδων (prompts). Οι στόχοι διαδραστικής επικοινωνίας (interactive communication), της υλοποίησης των γραμματικών και της λεπτομερούς σχεδίασης της εφαρμογής, είναι οι εξής:

1. *Η δημιουργία γραμματικών που κατά ένα μεγάλο ποσοστό θα προβλέπουν τις εισόδους των χρηστών.*

2. *Η σωστή σχεδίαση της ροής των διαλόγων, χρησιμοποιώντας τις κατάλληλες στρατηγικές και εκφράσεις, έτσι ώστε ο χρήστης να καθοδηγείται και να παραμένει μέσα στα πλαίσια της εκάστοτε γραμματικής.*

Όπως γίνεται κατανοητό, οι στόχοι είναι άμεσα συνδεδεμένοι μεταξύ τους. Η διαδικασία υλοποίησης των διαλόγων είναι στενά συνδεδεμένη με την υλοποίηση των γραμματικών. Αυτό που προσπαθούμε να κάνουμε είναι να προβλέψουμε όλες τις πιθανές εκφράσεις που μπορεί να εισάγει ο χρήστης, κάτι το οποίο απαιτεί εμπειρία και καλή γνώση γλωσσολογικών στοιχείων. Ωστόσο, ούτε η τέλεια γνώση, αλλά ούτε και η εμπειρία μπορούν να μας εξασφαλίσουν πλήρη κάλυψη. Για αυτό η ανάπτυξη των γραμματικών πρέπει να αφήνει την δυνατότητα της επέκτασής τους όταν αυτό κριθεί αναγκαίο. Είναι όμως σημαντικό να σημειώσουμε, ότι η προσθήκη νέων στοιχείων αναμενόμενων εκφράσεων σε μια γραμματική μπορεί να οδηγήσει σε νέες καταστάσεις συναγωνισμού, καθώς αυξάνεται η πιθανότητα να υπάρχουν συγχύσεις και σημεία αμφισημίας κατά την διαδικασία αναγνώρισης. Οπότε σκοπός είναι να μεγιστοποιήσουμε την κάλυψη των εκφράσεων του χρήστη ελαχιστοποιώντας παράλληλα τις υπερκαλύψεις.

Υπάρχουν δύο προσεγγίσεις για την υλοποίηση των γραμματικών. Οι βασισμένες σε κανόνες (rule-based) και οι γραμματικές βασισμένες στην στατιστική προσέγγιση, ή αλλιώς στοχαστικά γλωσσικά μοντέλα (Stochastic Language Models (N-Gram)). Οι πρώτες εμπεριέχουν μια ρητή δήλωση όλων των πιθανών ακολουθιών λέξεων και η δεύτερη βασίζεται στην εκμάθηση των πιθανοτήτων των διάφορων πιθανών ακολουθιών λέξεων μέσα από κάποια δεδομένα. Στην εφαρμογή μας χρησιμοποιήσαμε την ανάπτυξη των γραμματικών βασισμένων σε κανόνες (rule-based grammars) και για αυτό θα κάνουμε μια μικρή ανάλυση της μεθόδου αυτής και πως αυτή εφαρμόστηκε στην περίπτωση μας.

Παρακάτω γίνεται μια παρουσίαση των γραμματικών που χρησιμοποιούνται στην αναγνώριση ομιλίας, με σκοπό τον καθορισμό των λέξεων και εκφράσεων που πρόκειται να ακουστούν από τον αναγνωριστή ομιλίας. Οι γραμματικές αυτές μπορούν να χρησιμοποιηθούν, όπως είπαμε, από τους αναγνωριστές ομιλίας (speech recognizers) αλλά και από άλλους επεξεργαστές γραμματικών (grammar processors). Η σύνταξη των γραμματικών μπορεί να παρουσιαστεί με δύο μορφές, την Augmented BNF (ABNF) μορφή και την XML μορφή, τα οποία έχουν και τα δύο την εκφραστική

ισχύ μιας Context-Free Grammar (CFG) [26]. Εμείς θα ασχοληθούμε με την XML μορφή, επειδή αυτή χρησιμοποιήσαμε για την υλοποίηση της εφαρμογής. Η μορφή αυτή χρησιμοποιεί στοιχεία της γλώσσας XML για να εκφράσει τις γραμματικές δομές, ενώ δανείζεται στοιχεία σχεδιασμού από τις PipeBeach grammar, TalkML και ερευνητικές εκδοχές της JSpeech Grammar Format [JSGF] [26].

Επεξεργαστές Γραμματικών (Grammar Processors) και User Agents

Ένας επεξεργαστής γραμματικών είναι μια οντότητα, η οποία δέχεται ως είσοδο τις γραμματικές που περιγράφουμε. Ένας User Agent είναι ένας επεξεργαστής γραμματικών που δέχεται ως είσοδο από τον χρήστη μια ακολουθία λέξεων και την αντιστοιχίζει με βάση κάποια γραμματική, με σκοπό την παραγωγή ενός αποτελέσματος αναγνώρισης που ανταποκρίνεται στην είσοδο του χρήστη.

Οι αναγνωριστές ομιλίας είναι ένα σημαντικό κομμάτι των επεξεργαστών γραμματικών. Ένα άλλο κομμάτι αφορά τους Dual-Tone Multi-Frequency (DTMF) αναγνωριστές. Το είδος εισόδων που μπορεί να δεχτεί ένας User Agent καθορίζεται από τα είδη των γραμματικών που μπορεί να επεξεργαστεί. Για παράδειγμα, μπορεί να δεχτεί φωνητική είσοδο για γραμματικές που αφορούν φωνή και DTMF είσοδο για γραμματικές που χειρίζονται dtmf τύπου γραμματικές.

Ένας αναγνωριστής ομιλίας είναι ένας user agent που έχει τις παρακάτω εισόδους και εξόδους:

- Input : Μια ή πολλαπλές γραμματικές, που να ενημερώνουν τον αναγνωριστή για τις λέξεις και τις εκφράσεις που πρέπει να ακούσει.
- Input : Ένα ηχητικό σήμα που μπορεί να εμπεριέχει κάποιο κομμάτι ομιλίας που να ταιριάζει με την γραμματική που αναφέραμε.
- Output : Περιγραφές των αποτελεσμάτων της αναγνώρισης, οι οποίες περιέχουν λεπτομέρειες σχετικά με το περιεχόμενο της ομιλίας που αναγνωρίστηκε.
- Output : Πληροφορίες σχετικές με λάθη κατά την αναγνώριση ή με την απόδοση του συστήματος αναγνώρισης.

Σημασιολογική Προσέγγιση (Semantic Interpretation)

Ένας αναγνωριστής ομιλίας αντιστοιχίζει το ηχητικό σήμα της εισόδου για να παράγει ένα απλό κείμενο (raw text transcription or literal text) που αντιπροσωπεύει την είσοδο. Ο αναγνωριστής, έχει την δυνατότητα να επεξεργαστεί το κείμενο αυτό παραπέρα, παράγοντας μια σημασιολογική προσέγγιση της εισόδου. Η έξοδος μιας τέτοιας σημασιολογικής επεξεργασίας της εισόδου μπορεί να παρουσιαστεί χρησιμοποιώντας Natural Language Semantics Markup Language [NLSML].

Τερματικά Σύμβολα (Tokens) και επιστρεφόμενη πληροφορία (Slot)

Το τερματικό σύμβολο (terminal symbol - token) είναι εκείνο το μέρος της γραμματικής, το οποίο ορίζει ποιες λέξεις ή άλλες οντότητες μπορούν να γίνουν δεκτές. Στην αναγνώριση ομιλίας, το token είναι ένα δόκιμο ορθογραφικό λήμμα της γλώσσας που αναφέρεται η αναγνώριση. Ωστόσο token μπορεί να θεωρηθεί και κάθε άλλη συμβολοσειρά, την οποία ο αναγνωριστής μπορεί να την αναπαραστήσει φωνητικά.

Για να μπορέσει να κάνει την αντιστοίχιση μεταξύ φωνητικής εισόδου και γραμματικής, η αναγνώριση ομιλίας πρέπει να έχει την δυνατότητα να κάνει μοντελοποίηση των εκφράσεων σε κάθε token της γραμματικής. Οι αναγνωριστές ομιλίας διαθέτουν ποικιλία διαφορετικών τεχνικών για την εκτέλεση της εργασίας αυτής.

Ένας αναγνωριστής ομιλίας μεγάλου λεξιλογίου, μετατρέπει κάθε token σε μια ακολουθία φωνημάτων ή σε μια σειρά από ακολουθίες φωνημάτων. Η μετατροπή ενός ολόκληρου δόκιμου λήμματος (token) στην ομιλούμενη μορφή (phonemes) είναι πολύ διαφορετική ανάλογα με την γλώσσα, ή πολλές φορές, και ανάλογα με τα ιδιώματα της γλώσσας από περιοχή σε περιοχή.

Η μετατροπή κειμένου σε φώνημα από έναν αναγνωριστή μεγάλου λεξιλογίου μπορεί να περιλαμβάνει μερικές από τις παρακάτω υπο-διαδικασίες ή και όλες.

- *Αναζήτηση σε λεξικό προφορών.* Οι αναγνωριστές συνήθως διαθέτουν ένα λεξικό για κάθε υποστηριζόμενη γλώσσα, το οποίο παρέχει την ακολουθία φωνημάτων για κάθε token. Ο αλγόριθμος που κάνει την αντιστοίχιση μεταξύ token και προφοράς ορίζεται από το format του λεξικού ή/και από τον αναγνωριστή ομιλίας, ενώ μπορεί να διαφέρει ανάλογα με την γλώσσα.

- *Μορφολογική Ανάλυση (Morphological analysis)* Ο αναγνωριστής μπορεί να καθορίσει την μετασχηματισμένη μορφή του θέματος ενός token και του αντίστοιχου phoneme και να παράγει την μορφολογική παραλλαγή και την προφορά της. Αν, για παράδειγμα, είχαμε το token “Hyundai”, τότε ο αναγνωριστής θα μπορούσε να έχει έναν κανόνα που να καθορίζει την προφορά του λήμματος στον πληθυντικό αριθμό “Hyundai’s”.
- *Αυτόματη μετατροπή κειμένου σε phoneme (Automatic text-to-phoneme conversion)* Για πολλές γλώσσες υπάρχουν κανόνες που μπορούν να μετατρέψουν αυτόματα το κείμενο σε μια ακολουθία φωνημάτων. Για παράδειγμα, στην αγγλική γλώσσα οι περισσότερες (μα όχι όλες) λέξεις που έχουν την κατάληξη “ise” έχουν την κατάληξη “ai z” στην αντίστοιχη ακολουθία φωνημάτων τους. Κατά αυτόν τον τρόπο οι αναγνωριστές μπορούν να χρησιμοποιήσουν την αυτόματη μετατροπή για tokens τα οποία δεν μπορούν να βρεθούν σε κάποιο λεξικό φωνημάτων.

Κάθε γλώσσα ενδέχεται να έχει και άλλες πιο ειδικές μεθόδους για τον καθορισμό της προφοράς για κάθε token.

Όταν ένας επεξεργαστής γραμματικών δέχεται ως είσοδο μια γραμματική, η οποία περιέχει tokens που δεν μπορεί να αντιστοιχίσει σε κάποια μορφή φωνήματος, αλλά ούτε και να την χρησιμοποιήσει στην διαδικασία επεξεργασίας αναγνώρισης του ηχητικού σήματος, θα έπρεπε να ενημερώνεται το περιβάλλον, μέσα στο οποίο τρέχει ο επεξεργαστής γραμματικών. Παρακάτω δίνονται διάφορες τεχνικές που ακολουθήσαμε για να αποφύγουμε προβλήματα κατά την αναγνώριση.

- *Αποφυγή χρήσης Ακρωνύμιων.* Πρέπει όλοι οι χαρακτήρες που πρόκειται ακουστούν να είναι διαθέσιμοι στον αναγνωριστή. Για παράδειγμα, για ΜΠΔ χρησιμοποιήσαμε “μ π δ” ενός για το κ. Παπαδόπουλος χρησιμοποιήσαμε “κύριος παπαδόπουλος”.
- *Χρήση πεζών χαρακτήρων (lowercase).* Ο αναγνωριστής δεχόταν μόνο πεζούς χαρακτήρες. Για παράδειγμα αντί για “Πολυτεχνείο Κρήτης” χρησιμοποιήσαμε “πολυτεχνείο κρήτης”.

- *Ρητή έκφραση των σημείων στίξης.* Για παράδειγμα, αντικαθιστούμε το “&” με “ampersand” ή “και”, το “+” με το “συν”.
- *Περιγραφική έκφραση αριθμών.* Παρόλο που ο αναγνωριστής συνήθως υποστηρίζει ψηφία (0-9) οι υπόλοιποι αριθμοί δηλώνονται περιγραφικά όπως “δέκα” για “10” ή “εκατό” για “100”.

Σύνταξη XML Γραμματικών

Κατά την διαδικασία σχεδίασης των κειμένων παρακίνησης εισόδου, πρώτα ορίζουμε ποια είναι τα κομμάτια πληροφορίας που θα επιστρέφει η εκάστοτε γραμματική (slots), καθώς και μια σειρά από ενδεικτικές προτάσεις. Στην ουσία τα slots καθορίζουν και ποιος θα είναι ο πυρήνας (core), ποια είναι δηλαδή, τα κομμάτια αυτά που θα έχουν την πληροφορία που χρειαζόμαστε, και ποια θα είναι τα κομμάτια, τα οποία απλά θα συμπληρώνουν τις προτάσεις του χρήστη και θα περιβάλλουν τον πυρήνα (fillers). Αν για παράδειγμα θέλουμε να σχεδιάσουμε μια επιλογή του χρήστη για να ζητήσει το αυριανό πρόγραμμα μαθημάτων, θα χρησιμοποιούσαμε μια πρόταση της μορφής “θέλω να μάθω το πρόγραμμα μαθημάτων για **αύριο**”. Αυτή η πρόταση αποτελεί ένα slot, ενώ το core είναι η λέξη “αύριο” και ενώ όλα τα υπόλοιπα αποτελούν τους fillers.

Για την υλοποίηση των γραμματικών χρησιμοποιήσαμε την ανάπτυξη σε XML, όπως προαναφέραμε, κυρίως με το σκεπτικό ότι δεν είναι γλώσσα ανεπτυγμένη από κάποια συγκεκριμένη εταιρεία, όπως συμβαίνει για παράδειγμα με την Grammar Specification Language (GSL), η οποία είναι δημιούργημα της Nuance Communications.

Στον Πίνακα 4 φαίνονται οι τελεστές που χαρακτηρίζουν τις XML γραμματικές:

Τελεστής	Έκφραση	Ερμηνεία
<code><item/></code> Concatenation	<code><item></code> A B C...D <code></item></code>	A ακολουθούμενο από B ακολουθούμενο από C ακολουθούμενο από... D
<code><one-of/></code> Disjunction	<code><one-of></code> <code><item>A <item></code>	Ένα από τα A,B,C,D

	<code><item>B <item></code> <code><item>C <item></code> ... <code><item>D <item></code> <code></one-of></code>	
<code><item repeat = "0-1"/></code> Optional	<code><item repeat = "0-1"></code> A <code></item></code>	Το A είναι προαιρετικό
<code><item repeat = "1-"/></code> Positive Closure	<code><item repeat = "1-"></code> A <code></item></code>	Το A μπορεί να εμφανιστεί μια ή περισσότερες φορές
<code><item repeat = "0-"/></code> Kleene Closure	<code><item repeat = "0-"></code> A <code></item></code>	Το A μπορεί να εμφανιστεί καμία ή περισσότερες φορές

Πίνακας 3. Οι τελεστές των XML γραμματικών

Παρακάτω φαίνεται ένα παράδειγμα χρήσης των τελεστών αυτών για την δημιουργία μιας γραμματικής που χρησιμοποιήσαμε στην εφαρμογή.

```

<item>
  <one-of>
    <item>
      <item repeat="0-1">
        <one-of>
          <item> θα ήθελα </item>
          <item> χρειάζομαι </item>
          <item> θέλω </item>
        </one-of>
      </item>
    </one-of>
    <item> να με βοηθήσετε </item>
    <item> βοήθεια </item>
  </one-of>
  <item> βοηθείστε με </item>
</one-of>
<item>
  <item repeat="0-1">σας </item>
  <item> παρακαλώ </item>
</item>
</item>

```

Όπως μπορούμε εύκολα να δούμε, αυτό το μικρό κομμάτι γραμματικής αποτελείται από δύο μικρότερα κομμάτια. Η μια δηλώνει την έκφραση του χρήστη που έχει να κάνει με έκφραση της επιθυμίας του που μπορεί να είναι είτε «θα ήθελα» είτε

«θέλω» είτε ακόμα «χρειάζομαι» και η δεύτερη έχει να κάνει με το τι ακριβώς επιθυμεί ο χρήστης που στην προκειμένη περίπτωση είναι «βοήθεια» ή «να με βοηθήσετε».

Σε γενικές γραμμές οι γραμματικές δημιουργούνται μέσω επαναλήψεων των βασικών αυτών κανόνων, καθώς και μικρότερων ή και μεγάλων κομματιών γραμματικών. Για αυτό το λόγο θα πρέπει να είναι οργανωμένα για εύκολη διατήρηση και χρήση.

Αυτό γίνεται ακολουθώντας τα εξής βήματα:

1. Διάσπαση της γραμματικής σε μια λογικά διαρθρωμένη δομή μικρότερων τμημάτων γραμματικών.
2. Ο πυρήνας (core) και τα περιβάλλοντα στοιχεία (fillers) πρέπει να βρίσκονται σε ξεχωριστές υπο-γραμματικές.
3. Επιλογή περιγραφικών ονομάτων για τις υπο-γραμματικές και τις μεταβλητές που χρησιμοποιούμε.
4. Οι γραμματικές γράφονται έτσι ώστε να είναι ξεκάθαρη η δομή που χρησιμοποιείται.

3.2.3 Εφαρμογή Web-based (HTML)

Η Web-based εφαρμογή αποτελεί το κομμάτι εκείνο του συστήματος μέσω του οποίου γίνεται η διατήρηση και ενημέρωση της βάσης δεδομένων καθώς και διάφορες άλλες λειτουργίες για την σωστή λειτουργία της εφαρμογής. Αυτό θα είναι διαθέσιμο στους χρήστες από το διαδίκτυο. Το front end κομμάτι έχει να κάνει με τον σχεδιασμό των σεναρίων σε περιβάλλον HTML.

Η βασικές λειτουργικότητες που έπρεπε να καλύπτει η εφαρμογή υποστήριξης του συστήματος ήταν να εισάγει τα απαραίτητα δεδομένα στη βάση, και να υπάρχει η δυνατότητα τα δεδομένα αυτά να ενημερώνονται και όποτε κριθεί αναγκαίο να διαγράφονται. Στο Appendix C – Technical Issues δίνονται όλες οι περιπτώσεις που υλοποιήσαμε και καλύπτουν όλη την λειτουργικότητα που μας είναι αναγκαία για να κρατάμε την βάση της εφαρμογής μας ενήμερη.

3.3 Υλοποίηση Back End

Το μέρος αυτό αναφέρεται στις εργασίες της εφαρμογής οι οποίες είναι αόρατες από τον χρήστη και οι οποίες αναλαμβάνουν όλη την διαδικασία ανεύρεσης και επεξεργασίας πληροφοριών, πριν αυτά παρουσιαστούν μέσα από το front end κομμάτι της εφαρμογής που περιγράψαμε προηγουμένως

3.3.1 Παρουσίαση της Βάσης Δεδομένων

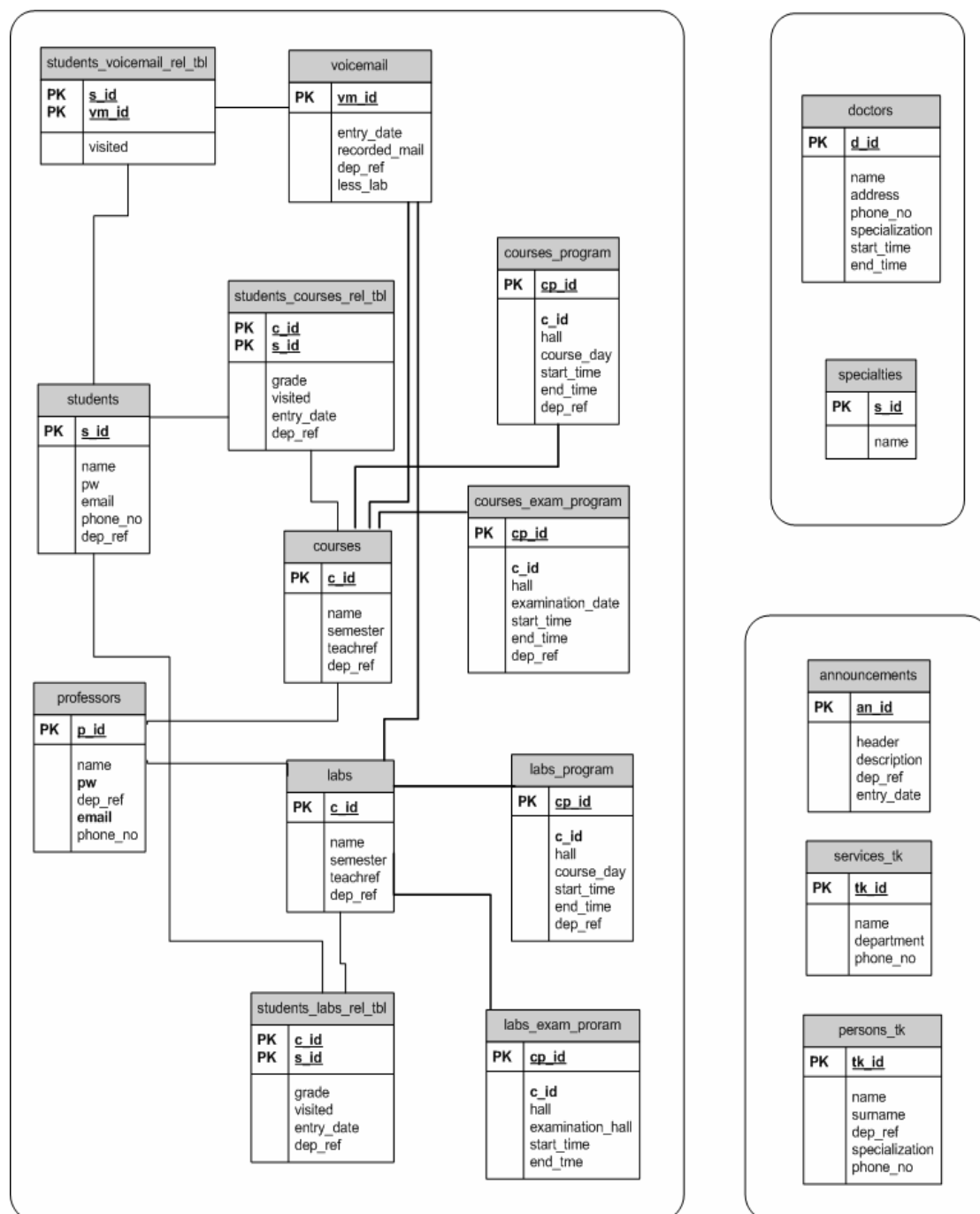
Η βάση δεδομένων που χρησιμοποιήσαμε για την εφαρμογή μας υλοποιήθηκε στον My-SQL Server 4.0.11-nt [27], το οποίο είναι ένα ελεύθερο λογισμικό με άφθονη βοήθεια στο διαδίκτυο. Η βάση υποστηρίζει πληροφορίες σχετικά με τους φοιτητές, τα μαθήματα, τους καθηγητές και όλες τις άλλες πληροφορίες που είναι απαραίτητες για την πλήρη κάλυψη όλων των λειτουργικοτήτων της εφαρμογής. Χρησιμοποιώντας τεχνικές ER [28], οργανώσαμε την πληροφορία σε δομικές ενότητες (tables) που περιγράφουν λεπτομερώς την ανακτώμενη πληροφορία. Η βάση δεδομένων, έτσι όπως υλοποιήθηκε, φαίνεται στο Σχήμα 12, το οποίο εξαγάγαμε μέσα από το MySQL-Front. Το MySQL-Front αποτελεί ένα αξιόπιστο εργαλείο ανοιχτού λογισμικού, το οποίο προσφέρει μια γραφική διεπαφή με τον MySQL server και την βάση δεδομένων που υλοποιήσαμε.

Ανακτώμενη Πληροφορία

Όσον αφορά το κεντρικό μενού η πληροφορία που μπορεί να ανακτηθεί άμεσα είναι οι τυχόν ανακοινώσεις προς all users που αναρτώνται στο web, ανά τακτά χρονικά διαστήματα, από το NOC. Από αυτές τις ανακοινώσεις η φωνητική πύλη θα μπορούσε σε πρώτη φάση να ανακοινώσει τον αριθμό και τις επικεφαλίδες τους, και στην συνέχεια και για όποια ανακοίνωση ο χρήστης επιθυμεί, να κάνει και

ανάγνωση του θέματος. Επίσης πρέπει να δημιουργηθεί ένα σύστημα ταυτοποίησης του χρήστη, μέσα από user name και password.

Τα δεδομένα που θα εξαχθούν από τη βάση πρέπει να χαρακτηρίζονται πλήρως από τις ιδιότητές τους. Αυτές κατηγοριοποιούνται σε κάποιες οντότητες που θα υπάρχουν στην υπηρεσία και θα εμπεριέχουν όλη την ανακτώμενη πληροφορία. Αυτές περιγράφονται στο Σχήμα 12:



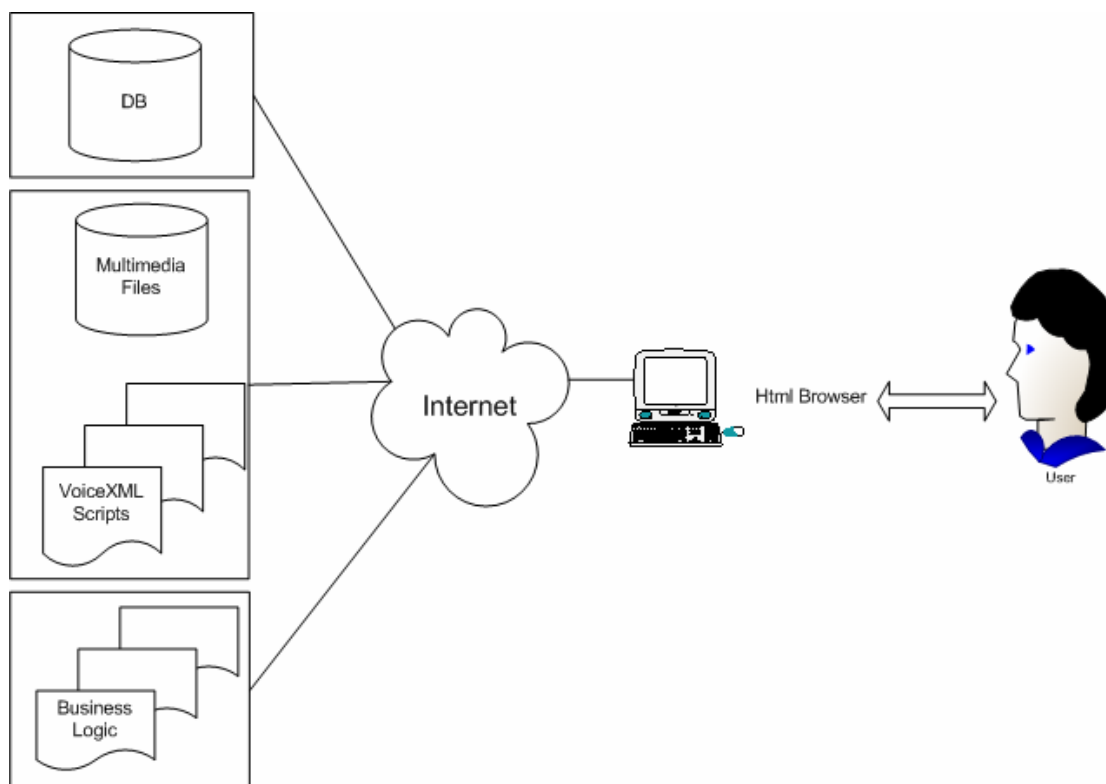
Σχήμα 12. Η Βάση Δεδομένων

Αναλυτική περιγραφή της πληροφορίας που υπάρχει στην βάση δεδομένων υπάρχει στο Appendix C – Technical Issues.

3.3.2 Ανάπτυξη Δυναμικών Εφαρμογών

Ανάπτυξη Δυναμικών Web-based εφαρμογών γραφικού περιβάλλοντος

Για την υλοποίηση του Web-based συστήματος της εφαρμογής χρησιμοποιήσαμε την αρχιτεκτονική που περιγράφεται στο Σχήμα 13. Visual Browser. Στο σχήμα αυτό περιγράφεται πως γίνεται η χρήση του web server για την περίπτωση των visual browsers (HTML browser). Ο HTML browser δέχεται HTML σελίδες από τον Web server και αφού τις επεξεργάζεται παρουσιάζει τα αποτελέσματα στην οθόνη του χρήστη. Δέχεται εντολές από τον χρήστη μέσω του πληκτρολογίου ή και του ποντικιού και μπορεί μέσα από τις επιλογές του χρήστη στις HTML σελίδες να έχει πρόσβαση στην βάση δεδομένων και να προκαλέσει διάφορες λειτουργίες στον application server [4].



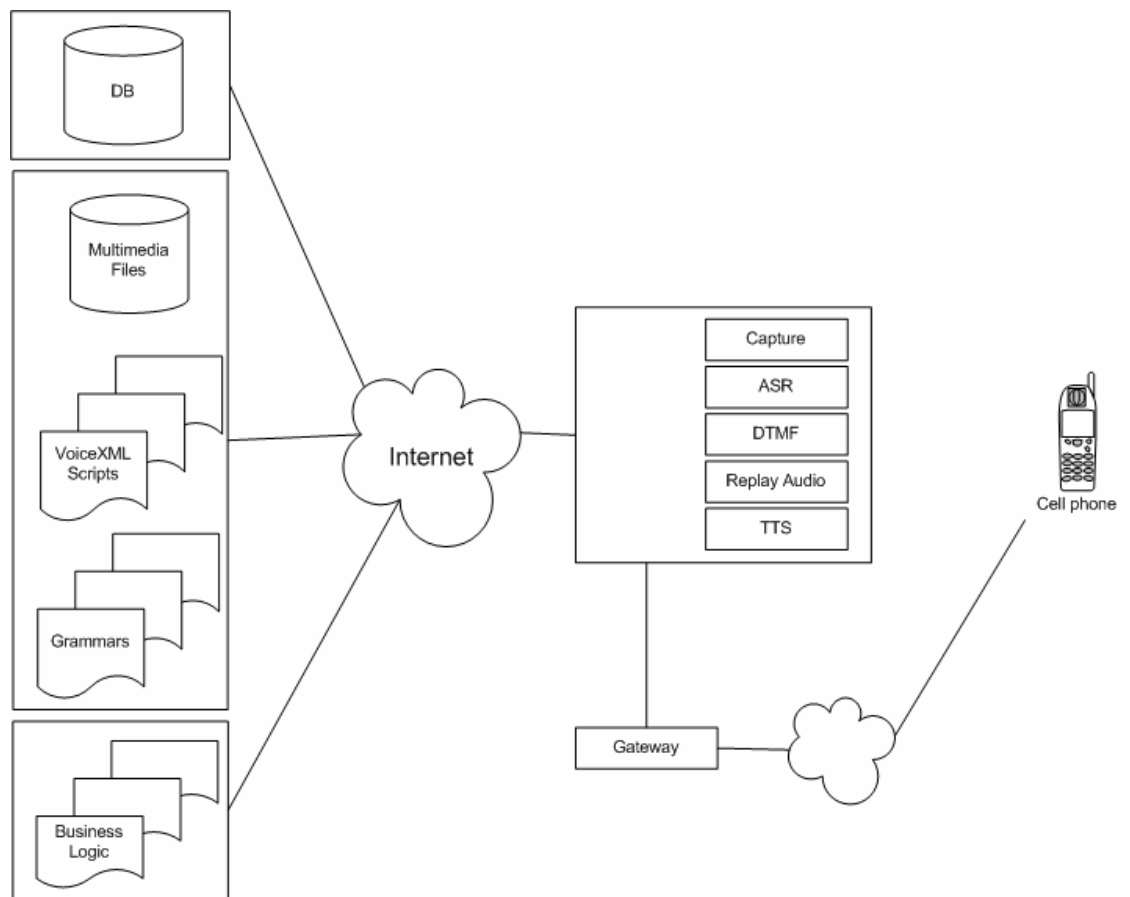
Σχήμα 13. Visual Browser

Ανάπτυξη Δυναμικών Φωνητικών Εφαρμογών

Ένας Voice Browser δουλεύει κατά τον ίδιο τρόπο με τον Visual Browser, με την διαφορά ότι δέχεται για είσοδο VoiceXML αρχεία για επεξεργασία και παράγει τους κατάλληλους ακουστικούς διαλόγους, οι οποίοι παρουσιάζονται στον χρήστη, ενώ δέχεται φωνητική είσοδο από τον χρήστη. Ο χρήστης μπορεί να μέσω των αρχείων που επεξεργάζεται ο browser να έχει πρόσβαση στην βάση δεδομένων και να προκαλέσει διάφορες λειτουργίες στον application server. Ωστόσο υπάρχει μια σημαντική διαφορά σε σύγκριση με τον Visual Browser. Το τηλέφωνο, το οποίο ο χρήστης χρησιμοποιεί για να επικοινωνήσει με το σύστημα, δεν έχει καμία υπολογιστική ισχύ, με αποτέλεσμα να μην μπορεί να γίνει η εκτέλεση του VoiceXML Browser στην συγκεκριμένη συσκευή.

Στο Σχήμα 14 Voice Browser, όπου παρουσιάζεται η χρήση ενός φωνητικού browser, φαίνεται πως ένα telephony gateway αναλαμβάνει να φέρει σε επικοινωνία τις τηλεφωνικές γραμμές με τον κόσμο του διαδικτύου και τα πρωτόκολλα που

χρησιμοποιούνται εκεί. Η λειτουργία του φωνητικού browser είναι παρόμοια με αυτή του Visual Browser, μόνο που όπως είπαμε, δεν μπορεί να εκτελεστεί στα όρια της τηλεφωνικής συσκευής του χρήστη, λόγω της έλλειψης του απαραίτητου hardware για την υποστήριξη της επεξεργασίας φωνής και ομιλίας. Για αυτό, ο browser εκτελείται σε ένα άλλο μηχάνημα το οποίο λέγεται voice server. Ο voice server παρέχει διάφορες υπηρεσίες όπως είναι η λήψη πληροφοριών από τον χρήστη, ηχογραφώντας την φωνή και αναγνωρίζοντας την είσοδό του με την χρήση της αναγνώρισης φωνής, ενώ μπορεί να ερμηνεύει την είσοδο του χρήστη μέσω του πληκτρολογίου. Επίσης έχει την δυνατότητα να μιλάει στον χρήστη χρησιμοποιώντας προηχογραφημένα μηνύματα και συνθετική φωνή. Οι είσοδοι του χρήστη και η έξοδος του συστήματος ελέγχονται από ένα VoiceXML αρχείο το οποίο ο voice browser λαμβάνει από τον web server. Καθώς τα VoiceXML αρχεία επεξεργάζονται και ερμηνεύονται, επιτυγχάνεται η πρόσβαση στην βάση δεδομένων, ενώ μπορούν να πυροδοτούνται διάφορες εργασίες στον application server. Η βάση δεδομένων είναι η ίδια με αυτήν που χρησιμοποιεί και ο HTML browser, ενώ πολλές από τις εργασίες του application server χρησιμοποιούνται, επίσης, από τον HTML server.[4]



Σχήμα 14 Voice Browser

Κεφάλαιο 4:

4. Αξιολόγηση

4.1 Δοκιμές Χρηστικότητας

Στο κομμάτι αυτό της υλοποίησης θέσαμε για πρώτη φορά το σύστημα στην διάθεση των μελλοντικών χρηστών. Η άποψή τους και οι παρατηρήσεις τους σχετικά με την ποιότητα της εφαρμογής έπαιξε καθοριστικό ρόλο στην τελική μορφή που είχε το σύστημα. Ενδεικτικά αναφέρουμε ποια ήταν τα βασικά προβλήματα που αντιμετώπισαν οι χρήστες κατά τις δοκιμές αυτές και πως αντιμετωπίστηκαν. Αυτά αφορούσαν προβλήματα στην ροή των διαλόγων, λάθη στην λογική υλοποίησης, λάθη αναγνώρισης αλλά και σε λάθη που οφείλονταν σε αναποτελεσματικές γραμματικές που έπρεπε να επανεξεταστούν.

Πολλοί χρήστες είχαν πρόβλημα με την πλοήγηση καθώς δεν είχαν την δυνατότητα να έχουν άμεση πρόσβαση σε κάποια αρκετά δημοφιλή υπομενού όπως ήταν το πρόγραμμα των μαθημάτων ή και τα αποτελέσματα βαθμολογιών. Με την τροποποίηση που κάναμε, ο χρήστης δεν χρειάζεται να καταβάλει μεγάλη προσπάθεια για να φτάσει στα υπομενού αυτά καθώς μπορεί πλέον από οποιοδήποτε σημείο της εφαρμογής να ζητήσει τις συγκεκριμένες λειτουργικότητες.

Ένα άλλο σημαντικό πρόβλημα ήταν οι γραμματικές που αφορούσαν τις Boolean αποκρίσεις των χρηστών. Γενικότερα το default που παρείχε η πλατφόρμα για την αναγνώριση προτάσεων τύπου Boolean αποδείχτηκε ανεπαρκής. Αυτή κάλυπτε ένα πολύ μικρό φάσμα εκφράσεων, όπως «ναι», «μάλιστα», «βέβαια» ή «όχι», «λάθος». Οι χρήστες έτειναν να απαντούν με πιο φυσικές εκφράσεις όπως «ναι αυτό θέλω» ή «ναι πες αυτό» ή «όχι δεν είπα αυτό». Παρόλο που έπρεπε να ξανασχεδιάσουμε τις γραμματικές για τις εκφράσεις αυτές, πρέπει να σημειώσουμε ότι ήταν αρκετά ενθαρρυντικό το γεγονός ότι οι χρήστες αισθάνονταν αρκετά άνετα με την εφαρμογή και δεν απαντούσαν με ψυχρές τυποποιημένες εκφράσεις αλλά με μια αρκετά φυσική γλώσσα.

Παρόλο, όμως, που ξανασχεδιάσαμε τις εκφράσεις για την αποδοχή Boolean αποκρίσεων, υπάρχει ένα βασικό πρόβλημα αναγνώρισης το οποίο δεν έχει να κάνει με την υλοποίηση. Αυτό έχει να κάνει με εκφράσεις που αποτελούνται από λίγα

«φωνήματα» όπως το «νέα», «ναι», «όχι». Σε αυτές τις περιπτώσεις το σύστημα δυσκολεύεται να αναγνωρίσει την είσοδο του χρήστη με αποτέλεσμα πολλές φορές να δυσχεραίνεται η αλληλεπίδραση με την εφαρμογή.

Ένα άλλο πρόβλημα, το οποίο είχαν οι χρήστες, και είναι επίσης δύσκολο να επιλυθεί, ήταν το γεγονός ότι η γραμματική που αφορούσε τις ειδικότητες των γιατρών που έχουν σύμβαση με το πολυτεχνείο υλοποιείται δυναμικά, χρησιμοποιώντας τις καταχωρήσεις ειδικοτήτων στην βάση δεδομένων. Οπότε αν ο χρήστης λείπει μια ειδικότητα που δεν υπάρχει στη βάση, όπως «οδοντίατρος», το σύστημα θα κάνει ενημέρωση ότι έγινε λάθος αναγνώριση και ο χρήστης δεν θα ξέρει, ότι απλά η ειδικότητα που ζητάει δεν υπάρχει. Εμείς καταφύγαμε στην λύση να ενημερώνουμε τον χρήστη, μετά την δεύτερη φορά που προσπαθεί να κάνει αναζήτηση, ότι μπορεί η ειδικότητα που ζητάει να μην υποστηρίζεται από την ασφάλεια του Πολυτεχνείου.

Γενικότερα υπήρχαν και λάθη στην λογική σε αρκετά σημεία της εφαρμογής, που αν δεν είχε εκτεθεί σε ενδελεχή έλεγχο από τους χρήστες θα ήταν αδύνατο να εντοπιστούν. Ένα τέτοιο λάθος αφορούσε τον υπολογισμό της ημερομηνίας και ώρας κάποιου μαθήματος, όπου αποκαλύφθηκε ένα σοβαρό λάθος στην λογική. Όταν ο χρήστης έλεγε «στις δεκαεπτά του μήνα» το σύστημα καταλάβαινε «στις δέκα ή ώρα στις επτά του μήνα», πράγμα πολύ δύσκολο να εντοπιστεί, από τη στιγμή που αυτό το λάθος συνέβαινε μόνο με ημερομηνίες του τύπου δεκαεπτά, δεκαέξι κτλ.

Σημαντικές ήταν, τέλος, οι παρατηρήσεις πάνω στην εκφορά των εκφράσεων παρακίνησης του συστήματος. Είναι γεγονός ότι λόγω του ότι δεν χρησιμοποιήσαμε προηχογραφημένα μηνύματα για τις παρακινήσεις προς τον χρήστη, αλλά συνθετική φωνή, έπρεπε να σχεδιάσουμε με προσεχτικό τρόπο τα κείμενα των παρακινήσεων για να ακούγονται όσο το δυνατόν πιο φυσικά.

4.2 Εκτίμηση Επιτυχούς Αναγνώρισης

Η διαδικασία της αξιολόγησης ήταν ένα σημαντικό στάδιο της υλοποίησης του συστήματος. Σε αυτό το στάδιο το σύστημα ετοιμάστηκε στην καλύτερη δυνατή του μορφή, ενώ έγιναν και οι κατάλληλες ρυθμίσεις για να μπορούμε να έχουμε όσο γίνεται καλύτερο feedback από τον αναγνωριστή, πράγμα το οποίο είναι απαραίτητο για την αξιολόγηση του συστήματος. Αυτά που παίρνουμε από τον αναγνωριστή είναι

οι ηχογραφημένες εκφράσεις των χρηστών καθώς και ένα LOG αρχείο το οποίο καταγράφει διάφορα στοιχεία, όπως το αποτέλεσμα της αναγνώρισης, το confidence level καθώς και το NL Interpretation. Στην συνέχεια δόθηκε σε δοκιμαστική χρήση στους πραγματικούς μελλοντικούς χρήστες, δηλαδή τα μέλη της πολυτεχνικής κοινότητας. Έπειτα από την κατάλληλη ενημέρωση μέσω της υπηρεσίας ηλεκτρονικού ταχυδρομείου all_users του πολυτεχνείου, περιμέναμε τις κλήσεις φοιτητών καθηγητών καθώς και άλλων φορέων του Πολυτεχνείου Κρήτης. Η ανταπόκριση που είχαμε ήταν κάτι παραπάνω από ικανοποιητική. Μέσα σε διάστημα τριών εβδομάδων το σύστημα δέχτηκε πάνω από εκατόν σαράντα (140) κλήσεις. Σε πρώτη φάση προχωρήσαμε στον υπολογισμό του Word Error Rate (WER) και του Sentence Error Rate (SER). Ενώ σε δεύτερη φάση υπολογίσαμε και το NL-Error Rate (NLER).

Υπολογισμός WER

Για να υπολογίσουμε το WER ακολουθήσαμε κάποια συγκεκριμένα βήματα, τα οποία φαίνονται παρακάτω. Το εργαλείο που χρησιμοποιήσαμε για την εξαγωγή αποτελεσμάτων ήταν το Sclite-1.4.2 της NIST [29] [30]. Για τον λόγο αυτό τα αρχεία που έπρεπε να ετοιμάσουμε για επεξεργασία έπρεπε να έχουν την απαιτούμενη δομή (format).

Βήμα 1) Δημιουργία των hypothesis files

Τα hypothesis files εξάγονται από το Log αρχείο του συστήματος αναγνώρισης, βλέποντας τι αναγνωρίζεται τελικά σε κάθε στάδιο αναγνώρισης. Τέτοιο παράδειγμα είναι το hyp, το οποίο το εξάγαμε από το LOG και φαίνεται παρακάτω..

```
πρόγραμμα μαθημάτων (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt01.wav)
το όγδοο ήμι (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt02.wav)
ναί (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt03.wav)
(2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt04.wav)
την τρίτη (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt05.wav)
σωστά (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt06.wav)
τέλος (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt07.wav)
όχι (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt08.wav)
```

Βήμα 2) Δημιουργία των transcription files

Τα transcription files δεν είναι τίποτα άλλο από ένα αρχείο με την μορφή του hypothesis file, όπου αντί για τις υποθέσεις του αναγνωριστή βάζουμε δίπλα σε κάθε utterance αυτό ακριβώς που είπε ο χρήστης. Για να ακούσουμε τα αρχεία wav που περιέχουν όσα έχουν πει οι χρήστες (utterances), χρησιμοποιούμε την εντολή playwav.exe στο command line ως εξής:

```
playwav.exe utt01.wav
```

Για την δημιουργία των transcription files ακολουθήσαμε τις οδηγίες που δίνονται για την μορφή των αρχείων αυτών για να έχουμε συμβατότητα με τα εργαλεία που χρησιμοποιήσαμε. Επίσης, ό,τι έχει ειπωθεί και δεν προβλέπεται από κάποια γραμματική το σημειώνουμε με την ένδειξη [OOG] για να μην το λάβουμε υπόψη μας κατά την διαδικασία της αξιολόγησης. Χαρακτηριστικό παράδειγμα ενός transcription file είναι το transc που αναφέρεται στο ίδιο LOG με το προαναφερθέν hypothesis file.

```
πρόγραμμα μαθημάτων (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt01.wav)
το όγδοο ήμι (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt02.wav)
ναι (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt03.wav)
την τρίτη (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt04.wav)
την τρίτη (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt05.wav)
σωστά (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt06.wav)
τέλος (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt07.wav)
όχι (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt08.wav)
έξοδος (2006\04April\07\16-34-05-beware-Analog-VOX-SC-line-4\utt09.wav)
```

Στην συνέχεια τρέξαμε το εργαλείο sclite σε Linux λειτουργικό σύστημα έχοντας ενεργή την επιλογή dtl (detailed) ή οποία μας δίνει λεπτομερή αποτελέσματα για τα αποτελέσματα αναγνώρισης

Υπολογισμός NLER

Στην συνέχεια προχωρήσαμε στον υπολογισμό του NLER. Για να μπορέσουμε να κάνουμε τον έλεγχο για τον NL Interpretation χρειαζόμαστε τα αντίστοιχα hypothesis και transcription files [31]. Οπότε, χρησιμοποιώντας τις γραμματικές του συστήματος καθώς και τα hypothesis που είχαμε φτιάξει προηγουμένως, προχωρούμε στην δημιουργία των NL hypothesis files ως εξής:

Βήμα 1) Προετοιμασία των γραμματικών

Σε αυτό το βήμα θα πρέπει να χρησιμοποιηθούν οι γραμματικές του συστήματος. Θα πρέπει μέσα από το Log να δούμε ποιες γραμματικές είναι ενεργές σε κάθε στάδιο αναγνώρισης. Αυτές ομαδοποιούνται κάθε φορά σε μια ανώτερη γραμματική που είναι ενεργή σε κάθε state. Οι γραμματικές είναι της μορφής:

RECOGNIZE

Nuance-Grammar-Label:ConfirmSizeAndToppings

Nuance-Package-Name:en-US

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE grammar PUBLIC "-//W3C//DTD GRAMMAR 1.0//EN"
    "http://www.w3.org/TR/speech-grammar/grammar.dtd">
<grammar version="1.0"
    xml:lang="el-GR"
    mode="voice"
    root="Back"
    xmlns="http://www.w3.org/2001/06/grammar"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org/2001/06/grammar
        http://www.w3.org/TR/speech-grammar/grammar.xsd">
    <rule id="Back" scope="public">
        <one-of>
            <item> βοήθεια <tag> &lt;NL "hlp"&gt; </tag> </item>
            <item> πίσω <tag> &lt;NL "bck"&gt; </tag></item>
        </one-of>
    </rule>
```


</grammar>

Στην παραπάνω γραμματική αν γίνει match η λέξη “βοήθεια” θα γεμίσει το slot NL με την τιμή “hlp” αν γίνει match η λέξη “πίσω” θα γεμίσει με την τιμή “bck”. Αν δεν γίνει τίποτα από τα δύο τότε θα σημαίνει ότι είναι Out Of Grammar (OOG).

Βήμα 2) Δημιουργία αρχείων με NL Interpretations

Έπειτα δημιουργήσαμε ένα dummy δυναμικό πακέτο γραμματικής. Αυτό το κάναμε δημιουργώντας πρώτα ένα άδειο αρχείο πχ foo.grammar και στη συνέχεια καλώντας την εντολή nuance-compile με ενεργή την επιλογή -enable_jit που δηλώνει ότι γίνεται compile μια just in time grammar ως εξής:

```
nuance-compile foo Greek -enable_jit
```

Με αυτόν το τρόπο δημιουργήσαμε το πακέτο foo το οποίο χρησιμοποιήσαμε για να βγάλουμε εκφράσεις μαζί με τα NL Interpretations. Αυτό το κάναμε εκτελώντας της εντολή:

```
generate.exe -package foo -grammar top.gsl -nl
```

Τελικά, για να γεμίσουμε με NL Interpretations τα hypothesis files και να φτιάξουμε τελικά τα hypothesis nrefs χρησιμοποιήσαμε την εντολή:

```
generate-nlref.exe -package foo -refs hypfoo.txt -nlref hypfoo.nlref -grammar top.gsl
```

όπου το hypfoo.txt είναι το hypothesis file, και το hypfoo.nlref το αρχείο με τα NL Interpretations. Στην συνέχεια, ακούγοντας τις εκφράσεις των χρηστών, δημιουργούμε τα αντίστοιχα transcription nrefs όπου υπάρχουν τα σωστά NL Interpretations. Τέλος χρησιμοποιώντας ένα μικρό πρόγραμμα που υλοποιήσαμε σε perl [34] που σύγκρινε τα δύο αρχεία γραμμή προς γραμμή, προχωρήσαμε στην εξαγωγή αποτελεσμάτων για την σημασιολογική προσέγγιση -NL Interpretation.

Αποτελέσματα

Οι κλήσεις που δεχτήκαμε ήταν συνολικά εκατόν σαράντα μια (141). Από αυτές εξάγαμε δύο χιλιάδες εξακόσιες τριάντα πέντε (2635) λέξεις των χρηστών και 1661 προτάσεις. Οι προτάσεις που εμπεριείχαν λάθη ήταν τριακόσιες εβδομήντα επτά (377) που αντιστοιχεί σε ένα ποσοστό 22,69 %. Αυτό το ποσοστό δεν ήταν ιδιαίτερα υψηλό από την στιγμή που οι χρήστες χρησιμοποιούσαν προτάσεις που αποτελούνταν από περιορισμένο αριθμό λέξεων. Οι προτάσεις που περιείχαν deletions είναι σε σημαντικό βαθμό περισσότερες από αυτές που περιείχαν insertions και substitutions και ήταν τριακόσιες πενήντα δύο (352), ποσοστό 21,19. Τα πολλά deletions που είχαμε οφείλονται τις περισσότερες φορές σε rejections από τον αναγνωριστή λόγω των κακών συνθηκών κατά την κλήση, όπως Background Noise, Hesitations, False Starts, Mispronunciations, Side Speech καθώς και χαμηλή ένταση στην φωνή των χρηστών. Πρέπει να σημειωθεί εδώ ότι οι χρήστες ήταν κατά κύριο λόγο φοιτητές του Πολυτεχνείου και για αυτό πολλές φορές παρατηρήσαμε κάποιες αυθόρμητες συμπεριφορές και αναγκαστήκαμε να μην λάβουμε καν υπόψη μας κάποιες κλήσεις, λόγω της κακής ποιότητας των ηχογραφημένων μηνυμάτων. Τέλος αυτές που είχαν substitutions είναι δεκαέξι (16), ποσοστό 0,96 % και insertions δεκατέσσερις (14), ποσοστό 0,84 %.

Οι λάθος αναγνωρισμένες λέξεις ήταν πεντακόσιες πενήντα τρεις (553), και άρα είχαμε ένα ποσοστό λανθασμένων αναγνωρίσεων στο σύνολο των λέξεων που είπαν οι χρήστες της τάξεως του 20,98 %, το οποίο σημαίνει ότι το ποσοστό επιτυχίας που πετύχαμε στην αναγνώριση λέξεων κυμαίνονταν στο 79,02 %. Όπως παρατηρήσαμε και πιο πριν το μεγαλύτερο μέρος των λάθος αναγνωρισμένων λέξεων αναφέρεται στα deletions τα οποία ήταν πεντακόσιες είκοσι δύο (522), που αντιστοιχεί σε ποσοστό 19,81 %. Ενώ είχαμε δεκαεφτά (17) substitutions με ποσοστό 0,64 % και insertions 14 με ποσοστό 0,53 %. Τα αποτελέσματα αυτά φαίνονται και στους παρακάτω πίνακες.

Sentences	Sentences with Errors	With Deletions	With Substitutions	With Insertions
1661	377 22,69 %	352 21,19 %	16 0,96 %	14 0,84 %

Table 1. Αποτελέσματα αξιολόγησης σε επίπεδο προτάσεων

Words	Errors	Deletions	Substitutions	Insertions
2635	553	522	17	14
	20,98 %	19,81 %	0,64 %	0,84 %

Table 2. Αποτελέσματα αξιολόγησης σε επίπεδο λέξεων

Η μελέτη για τον υπολογισμό του NLER που ακολούθησε, σε σύνολο χιλίων εφτακοσίων είκοσι εννέα (1729) NL Interpretations, από τα οποία τα 422 αναγνωρίστηκαν λάθος, μας έδωσε Natural Language Error Rate (NLER) της τάξεως του 24,40 % που σημαίνει ένα ποσοστό επιτυχίας για την σημασιολογική ανάλυση της τάξεως του 75,60 %.

Αποτελέσματα μετά από τις κατάλληλες τροποποιήσεις

Εξετάζοντας λεπτομερώς τις εκφράσεις των χρηστών και την αντίστοιχη κάθε φορά συμπεριφορά του συστήματος παρατηρήσαμε ότι μπορούσαμε να βελτιώσουμε την απόδοση αναγνώρισης αν προχωρούσαμε σε κάποιες μικρές αλλά ουσιαστικές αλλαγές τόσο στις χρησιμοποιούμενες γραμματικές όσο και στην δομή των διαλόγων σε κάποια σημεία της εφαρμογής. Έπειτα λοιπόν από τις απαραίτητες διορθώσεις στις γραμματικές, ανάλογα με τα συχνότερα λάθη στις εισόδους του χρήστη, καθώς και κάποιες αλλαγές στις ρυθμίσεις των VoiceXML αρχείων, τα αντίστοιχα ποσοστά επιτυχίας αυξήθηκαν για την αναγνώριση λέξεων στο 91 % με δραματική μείωση των deletions, ενώ τα insertions και substitutions μειώθηκαν ελαφρά. Τέλος στην σημασιολογική προσέγγιση το ποσοστό επιτυχίας έφτασε στο 88,15 %.

Κεφάλαιο 5:

5. Συμπεράσματα και Επεκτασιμότητα

Η φωνητική υπηρεσία του πολυτεχνείου Κρήτης αποτελεί μια πρότυπη εφαρμογή για την ανάκτηση πληροφοριών μέσω ομιλίας. Η χρήση μιας τέτοιας υπηρεσίας αποτελεί ένα πρωτοποριακό βήμα προς την καλύτερη εξυπηρέτηση των αναγκών της ακαδημαϊκής κοινότητας. Επίσης, πρέπει να σημειωθεί, ότι τα αποτελέσματα της αναγνώρισης απέδειξαν ότι ένα τέτοιο σύστημα θα μπορούσε να συνεισφέρει αποτελεσματικά στην καλύτερη ενημέρωση των φοιτητών σχετικά με ζητήματα που αφορούν την φοιτητική τους ζωή. Άλλωστε, η ανταπόκριση που είχαμε, κατά το στάδιο της αξιολόγησης, ήταν κάτι παραπάνω από ενθαρρυντική μιας και ακόμα μετά το τέλος της περιόδου αξιολόγησης, το σύστημα εξακολουθούσε να δέχεται κλήσεις από φοιτητές και άλλα μέλη της πολυτεχνικής κοινότητας που ζητούσαν να πάρουν διάφορες πληροφορίες. Είναι χαρακτηριστικό ότι πολλοί φοιτητές προσπαθούσαν επανειλημμένα να εγγραφούν στο σύστημα και να ακούσουν βαθμολογίες μαθημάτων ή προσωποποιημένα προγράμματα μαθημάτων παρόλο που είχαμε διασαφηνίσει ότι αυτό το μέρος της εφαρμογής δεν μπορεί να λειτουργήσει ακόμα λόγω τεχνικών δυσκολιών.

Είναι χαρακτηριστικό ότι λόγω του τρόπου σχεδίασης και υλοποίησης που χρησιμοποιήσαμε είναι πολύ εύκολη η επέκταση της εφαρμογής σε διάφορες άλλες λειτουργικότητες, έτσι ώστε να καλύπτονται όσο το δυνατόν καλύτερα οι ανάγκες των φοιτητών για πληροφόρηση.

Μερικές ιδέες που θα μπορούσαν να υλοποιηθούν μελλοντικά είναι οι εξής:

- Διασύνδεση με την βάση της βιβλιοθήκης για αναζήτηση καταχωρήσεων ή ανανέωση δανεισμού βιβλίου, ή οποία τώρα γίνεται μέσω τηλεφώνου.
- Διασύνδεση με την διανομή βιβλίου για ενημέρωση των φοιτητών για την άφιξη βιβλίων σχετικών με τα μαθήματα που έχουν δηλώσει..

- ενημέρωση σχετικά με την πόλη των Χανίων και τις δραστηριότητες των πολιτιστικών ομάδων.
- Ενημέρωση για το μενού της λέσχης ή και τις υπάρχουσες τιμές, μέσω μιας βάσης που θα ενημερώνεται καθημερινώς.
- Στην υπηρεσία τηλεφωνικού καταλόγου και αναζήτησης γιατρών, θα μπορούσε να υπάρχει απευθείας εκτροπή κλήσης προς το τηλέφωνο που αναζητήθηκε.

Appendix A – Dialog Specification

Σχεδιασμός διαλόγων(Dialog Specification)

Αναλυτική περιγραφή της λειτουργικότητας της εφαρμογής και των διαλόγων use cases.

Τυπογραφικές αναφορές

Τα κείμενα που αναφέρονται στα prompts βρίσκονται μέσα σε διπλά εισαγωγικά και είναι σε bold, italic “***Καλωσορίσατε στο σύστημα φωνητικής πύλης. ..***”

Τα κείμενα που αναφέρονται στις αποκρίσεις του χρήστη της εφαρμογής βρίσκονται σε διπλά εισαγωγικά και italic.

Όταν αναφερόμαστε σε κείμενα φυσικής γλώσσας NL αυτά είναι σε <> πχ <όνομα>. Αυτά αποτελούν τα slots και παίρνουν κάθε φορά τιμή ανάλογα με την εκάστοτε γραμματική και τις αποκρίσεις του χρήστη

Στην εφαρμογή χρησιμοποιούνται μεταβλητές συστήματος, τα οποία χρησιμοποιούνται για να αποθηκεύονται οι τιμές των slots, για τα οποία μιλήσαμε προηγουμένως. Αυτά βρίσκονται σε [] πχ. [όνομα]

Περιγραφή διαλόγων

Κατά την διάρκεια εκτέλεσης των διαλόγων, ο χρήστης μπορεί να καταφύγει ανά πάσα στιγμή σε κάποιες καταστάσεις και ανεξάρτητα σε ποιο σημείο των διαλόγων βρίσκεται. Οι καταστάσεις αυτές λέγονται καθολικές καταστάσεις.

Καθολική Συμπεριφορά:

Κάθε φορά που στην εφαρμογή λαμβάνει χώρα μια αναγνώριση η γραμματική που αφορά την μετάβαση στις καθολικές καταστάσεις είναι ενεργή. Έτσι οι χρήστες μπορούν ανά πάσα στιγμή να καταφεύγουν στις καταστάσεις 'βοήθεια', 'επανάληψη της υπαγόρευσης' ή 'έξοδος', οι οποίες είναι κάποιες καθολικές καταστάσεις..

Παραδείγματα τέτοιων περιπτώσεων μπορούν να περιγραφούν ως εξής:

Φράση χρήστη	Αντίστοιχη κατάσταση μετάβασης NL
"Βοήθεια" "Τι να κάνω"	<μετάβαση βοήθεια>
"έξοδος" "Γεια σας"	<μετάβαση έξοδος>
"επανάληψη" "απ 'την αρχή"	<μετάβαση επανάληψη>

Παρακάτω δίνεται ψευδο-κώδικας με την αναμενόμενη λειτουργία του συστήματος.

Action

if <μετάβαση βοήθεια> then

Αναπαραγωγή της συγκεκριμένης υπαγόρευσης που αντιστοιχεί στην κατάσταση που βρίσκεται ο χρήσης.

else if < μετάβαση επανάληψη> then

Επιστροφή στην αρχή της αντίστοιχης κατάστασης.

else if < μετάβαση έξοδος> then

Go to: Έξοδος

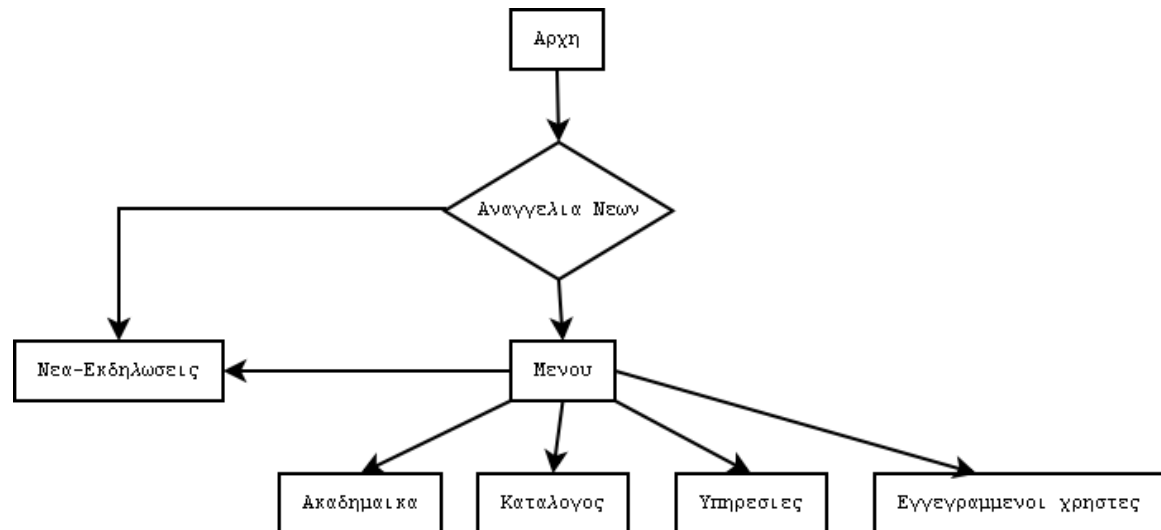
end if

Έξοδος

Σ' αυτήν την κατάσταση μεταβαίνουμε όταν ο χρήστης έχει επιλέξει να τερματίσει την εφαρμογή. Στις οδηγίες προς τον χρήστη, οι οποίες είναι πάντα στιγμή διαθέσιμες στον χρήστη, δίνεται η κατάλληλη ενημέρωση στον χρήστη ότι μπορεί να τερματίσει την εφαρμογή ανά πάσα στιγμή:

“Μπορείτε να τερματίσετε την εφαρμογή ανά πάσα στιγμή όταν τελειώσετε την εργασία σας”

Αρχή:



Είναι η κατάσταση όπου ο χρήστης μόλις έχει μπει στο σύστημα. Σ' αυτή τη φάση της εφαρμογής ο χρήστης μπορεί να ακούσει προ-ηχογραφημένες ανακοινώσεις γενικού ενδιαφέροντος, ενώ έχει την δυνατότητα να ζητήσει οδηγίες σε περίπτωση που είναι άπειρος στην χρήση του συστήματος.

Νέα εκδηλώσεις

Αρχική Ανακοίνωση

“Υπάρχουν [αρ ανακοινώσεων] ανακοινώσεις γενικού ενδιαφέροντος. Θα θέλατε να τις ακούσετε;

Πιθανές μεταβάσεις:

Ανάγνωση Ανακοινώσεων

Κεντρικό Μενού

<τιμές>

Ναι

Όχι

Παραδείγματα Εκφράσεων:

Έκφραση Χρήστη	Μετάβαση, τιμές
Ναι Βέβαια	<μετάβαση <u>Ανάγνωση Ανακοινώσεων</u> > <Ναι>
Όχι Δεν θέλω	<μετάβαση <u>Κεντρικό Μενού</u> > <Όχι>

Action

if <Ναι> then

go to: Ανάγνωση Ανακοινώσεων

else go to Κεντρικό Μενού

Διαχείριση λαθών

Λάθος 1-ανακοίνωση

“Πείτε ναι για να ακούσετε τις ανακοινώσεις ή όχι για να προχωρήσετε παρακάτω”

Λάθος 2-ανακοίνωση

“Για να ακούσετε τις ανακοινώσεις πείτε ναι. Γεια να προχωρήσετε στο κεντρικό μενού πείτε όχι”

Ανακοίνωση βοήθειας:

“Είναι απαραίτητο να πείτε ναι αν θέλετε να ακούσετε τις ανακοινώσεις γενικού ενδιαφέροντος, ή όχι αν δεν θέλετε να τις ακούσετε”

Μετά από τρεις προσπάθειες και αν ο χρήστης δεν έχει καταφέρει να δώσει έγκυρη απάντηση τότε το σύστημα προχωράει αυτόματα στο κεντρικό μενού.

Ανάγνωση Ανακοινώσεων:

Αρχική ανακοίνωση:

“Ανακοινώσεις. Χρησιμοποιήστε τις εντολές επόμενη, προηγούμενη για να προηγηθείτε και την εντολή τέλος για να επιστρέψετε στο κεντρικό μενού.”

Πιθανές μεταβάσεις:

Πλοήγηση στη λίστα αποτελεσμάτων

< τιμές >

Επόμενη

Προηγούμενη

Τέλος

Παραδείγματα Εκφράσεων:

Έκφραση Χρήστη	Μετάβαση, τιμές
Επόμενη ανακοίνωση	<μετάβαση Πλοήγηση >
Μετά	< Επόμενη>
Προηγούμενη ανακοίνωση	<μετάβαση Πλοήγηση >
Πριν	< Προηγούμενη>

Action

if < Επόμενη> then

 prompt: Επόμενη Ανακοίνωση

else if < Προηγούμενη> then

 prompt: Προηγούμενη Ανακοίνωση

else if < > then

 prompt: Επόμενη Ανακοίνωση

else if <Τέλος>

 go to Κεντρικό Μενού

end if

Ανακοίνωση

“Θέμα <Ανακοίνωση πρώτη>, ανακοίνωση <Ανακοίνωση> δημοσιεύτηκε <ημέρα εισαγωγής>. Θέμα <Ανακοίνωση δεύτερη>, ανακοίνωση <Ανακοίνωση>. ..”

Ανακοίνωση Βοήθειας

“Μπορείτε να ακούστε τις ανακοινώσεις ακολουθώντας τα εξής βήματα. Για να προχωρήσετε στην επόμενη ανακοίνωση, πείτε επόμενη, για να ακούσετε ξανά την προηγούμενη ανακοίνωση, πείτε προηγούμενη, και για να προχωρήσετε στο κεντρικό μενού πείτε τέλος. Για να ακούσετε την ανακοίνωση πείτε λεπτομέρειες ”

Κεντρικό μενού

Εδώ ο χρήστης αρχίζει την πλοήγησή του στο σύστημα καθώς το σύστημα ανακοινώνει τις βασικές λειτουργίες που μπορεί να εκτελέσει.

Αρχική ανακοίνωση

“Επιλέξτε μια από τις παρακάτω ενότητες, Νέα – Εκδηλώσεις, Ακαδημαϊκά, Κατάλογος υπηρεσιών και προσώπων, Υπηρεσίες, ενώ για να προηγηθείτε ως εγγεγραμμένοι χρήστες πείτε Εγγεγραμμένοι Χρήστες”

Ανακοίνωση μετά από επιστροφή από κάποια κατάσταση

“Βρίσκεστε στο κεντρικό μενού. Παρακαλώ πείτε σε ποια κατάσταση θέλετε να μεταβείτε.”

Ανακοίνωση Βοήθειας

“Παρακαλώ, πείτε μου την εργασία την οποία θέλετε να εκτελέσετε. Οι επιλογές σας είναι οι εξής :Νέα – Εκδηλώσεις, Ακαδημαϊκά για θέματα όπως πρόγραμμα μαθημάτων, Κατάλογος υπηρεσιών και προσώπων για πληροφορίες τηλεφώνων, Υπηρεσίες για εύρεση γιατρού ή για πρόσβαση στο φωνητικό ταχυδρομείο και Εγγεγραμμένοι Χρήστες”

Πιθανές μεταβάσεις :

Κεντρικό Μενού

Νέα – Εκδηλώσεις

Ακαδημαϊκά Θέματα

Τηλεφωνικός Κατάλογος

Υπηρεσίες του Πολυτεχνείου

Εγγεγραμμένοι Χρήστες

Εκφράσεις φυσικής γλώσσας NL - <τιμές>:

Κεντρικό Μενού

Νέα – Εκδηλώσεις

Ακαδημαϊκά θέματα

Τηλεφωνικός Κατάλογος

Υπηρεσίες του Πολυτεχνείου

Εγγεγραμμένοι Χρήστες

Παραδείγματα εκφράσεων

Έκφραση του χρήστη	Μεταβάσεις, τιμές μεταβλητών
Νέα Εκδηλώσεις	<μετάβαση <u>Νέα Εκδηλώσεις</u> > < Νέα-Εκδηλώσεις >
Ακαδημαϊκά Ακαδημαϊκά θέματα	<μετάβαση <u>Ακαδημαϊκά Θέματα</u> > < Ακαδημαϊκά θέματα >
Εγγεγραμμένοι Χρήστες Εγγεγραμμένος	<μετάβαση <u>Εγγεγραμμένοι Χρήστες</u> > < Εγγεγραμμένοι Χρήστες >

Action

if < Νέα Εκδηλώσεις >

Go to: Νέα εκδηλώσεις

else if < Ακαδημαϊκά Θέματα >

Go to: Ακαδημαϊκά Θέματα
else if <Τηλεφωνικός Κατάλογος >
Go to: Τηλεφωνικός Κατάλογος
else if < Υπηρεσίες του Πολυτεχνείου >
Go to: Υπηρεσίες του Πολυτεχνείου
else if <Κεντρικό Μενού>
Επιστροφή στην κορυφή της τρέχουσας κατάστασης
else if < Εγγεγραμμένοι Χρήστες >
Go to: Εγγεγραμμένοι Χρήστες
end if.

Διαχείριση λάθους.

Πρώτο:

“Παρακαλώ πείτε ένα από τα εξής: Νέα – Εκδηλώσεις, Ακαδημαϊκά θέματα, Κατάλογος υπηρεσιών και προσώπων, Υπηρεσίες του Πολυτεχνείου και Εγγεγραμμένοι Χρήστες. ”

Δεύτερο:

“Παρακαλώ πείτε τι θα θέλατε να κάνετε. Αν έχετε δυσκολίες πείτε βοήθεια, ενώ μπορείτε να πείτε και επανάληψη για να ξανακούσετε τις επιλογές σας.”

Τρίτο:

“Για να αντιμετωπίσετε το πρόβλημα σας δίνονται οι παρακάτω οδηγίες. ”

Οδηγίες:

Εδώ οδηγείται ο χρήστης όταν ζητήσει οδηγίες ή όταν το σύστημα δεν έχει καταφέρει να καταλάβει την είσοδο του χρήστη μετά από επανειλημμένες προσπάθειες. Ο χρήστης μπορεί να επιστρέψει στην προηγούμενη κατάσταση ανά πάσα στιγμή διακόπτοντας την ανακοίνωση.

Ανακοίνωση

“Οδηγίες. Η Φωνητική Πύλη του Πολυτεχνείου Κρήτης αποτελεί ένα σύστημα βασισμένο στην επεξεργασία φωνής, το οποίο σας επιτρέπει να πάρετε πληροφορίες σχετικά με διάφορα θέματα όπως πρόγραμμα μαθημάτων και εργαστηρίων, σας επιτρέπει να ακούσετε φωνητικά μηνύματα που αφορούν μαθήματα στα οποία είστε εγγεγραμμένοι, ενώ μπορείτε να βρείτε και διάφορες άλλες πληροφορίες όπως τηλέφωνα υπηρεσιών και ανθρώπων ή ειδικευμένους γιατρούς. Αν εγγραφείτε στο σύστημα το σύστημα μπορεί να κάνει πιο εύκολη τη πλοήγηση σας καθώς αν κάνετε login μπορεί να ελέγξει άμεσα για θέματα που σας αφορούν όπως η ύπαρξη νέου φωνητικού μηνύματος. Κατά την πλοήγηση σας θα έχετε άφθονη βοήθεια από τις αντίστοιχες ανακοινώσεις ενώ μπορείτε πάντα να πείτε βοήθεια και επανάληψη.”

Action

Επιστροφή στο κεντρικό μενού.

Ακαδημαϊκά

Εδώ οδηγείται ο χρήστης όταν ενδιαφέρεται να μάθει να πληροφορίες για το πρόγραμμα μαθημάτων και εργαστηρίων, το πρόγραμμα εξεταστικής και για βαθμολογίες μαθημάτων και εργαστηρίων.

Αρχική Ανακοίνωση

“Για να ακούσετε το πρόγραμμα μαθημάτων πείτε πρόγραμμα μαθημάτων, για το πρόγραμμα εργαστηρίων πείτε πρόγραμμα εργαστηρίων, ενώ για το πρόγραμμα της εξεταστικής πείτε πρόγραμμα εξεταστικής. Για να ακούσετε βαθμολογίες μαθημάτων και εργαστηρίων πείτε βαθμολογίες μαθημάτων και εργαστηρίων.”

Ανακοίνωση Βοήθειας

“Παρακαλώ, πείτε μου την εργασία την οποία θέλετε να εκτελέσετε. Οι επιλογές σας είναι οι εξής: πρόγραμμα μαθημάτων, πρόγραμμα εργαστηρίων και πρόγραμμα εξεταστικής, βαθμολογίες μαθημάτων και εργαστηρίων”

Πιθανές Μεταβάσεις:

Μενού Ακαδημαϊκών

Πρόγραμμα Μαθημάτων

Πρόγραμμα Εργαστηρίων

Πρόγραμμα Εξεταστικής

Εκφράσεις φυσικής γλώσσας NL - <τιμές>:

Μενού Ακαδημαϊκών

Πρόγραμμα Μαθημάτων

Πρόγραμμα Εργαστηρίων

Πρόγραμμα Εξεταστικής

Παραδείγματα εκφράσεων

Εκφραση Χρήστη	Μετάβαση, τιμές
Πρόγραμμα εργαστηρίων Εργαστήρια	<μετάβαση <u>Πρόγραμμα Εργαστηρίων</u> > < Πρόγραμμα Εργαστηρίων>
Πρόγραμμα μαθημάτων Μαθήματα	<μετάβαση <u>Πρόγραμμα Μαθημάτων</u> > < Πρόγραμμα Μαθημάτων>
Πρόγραμμα εξεταστικής Εξεταστική	<μετάβαση <u>Πρόγραμμα Εξεταστικής</u> > < Πρόγραμμα Εξεταστικής >

Action

if <Πρόγραμμα Εργαστηρίων> then

go to: Πρόγραμμα Εργαστηρίων

else if <Πρόγραμμα Μαθημάτων >

go to: Πρόγραμμα Μαθημάτων

else if < Πρόγραμμα Εξεταστικής >

go to: Πρόγραμμα Εξεταστικής

end if

Διαχείριση λάθους.

Πρώτο:

“Παρακαλώ πείτε ένα από τα εξής: πρόγραμμα μαθημάτων, πρόγραμμα εργαστηρίων, ή πρόγραμμα της εξεταστικής, βαθμολογίες μαθημάτων και εργαστηρίων”

Δεύτερο:

“Παρακαλώ πείτε τι θα θέλατε να κάνετε. Αν έχετε δυσκολίες πείτε βοήθεια, ενώ μπορείτε να πείτε και επανάληψη για να ξανακούσετε τις επιλογές σας.”

Τρίτο:

“Για να αντιμετωπίσετε το πρόβλημα σας δίνονται οι παρακάτω οδηγίες.”

Πρόγραμμα Εργαστηρίων

Αρχική ανακοίνωση

“Πρέπει να μου πείτε το τμήμα και το εξάμηνο για το οποίο ενδιαφέρεστε να μάθετε. Πχ «HMMY, 4^ο»”

Παραδείγματα εκφράσεων:

Εκφραση Χρήστη	Μετάβαση τιμές
Hμυ(HMMY),4ο	<μετάβαση Ανακοίνωση Προγράμματος > <τμήμα HMMY><εξάμηνο 4>

Action

if [τμήμα] = valid, [εξάμηνο]=valid then

go to: Ανακοίνωση Προγράμματος

else if only [τμήμα]= valid

prompt:”***Παρακαλώ πείτε το εξάμηνο για το οποίο θέλετε να μάθετε το πρόγραμμα.***”

set [εξάμηνο]=<εξάμηνο>

go to: **Ανακοίνωση Προγράμματος**

else if only [εξάμηνο]= valid

prompt:”***Πείτε μου και το τμήμα..***”

set [τμήμα]=<τμήμα>

go to: **Ανακοίνωση Προγράμματος**

end if

Πρέπει να επισημάνουμε και ότι αν ο χρήστης έχει κάνει login πιο πριν τότε το σύστημα αναγνωρίζει τον χρήστη και του επιστρέφει το πρόγραμμα μαθημάτων στο οποίο έχει προηγουμένως εγγραφεί ο χρήστης μέσω της on-line σελίδας εφαρμογής και μπορεί απλά να του κάνει μια αναγνωριστική ερώτηση όπως:

“Θέλετε το πρόγραμμα εργαστηρίων για για τα μαθήματα που έχετε δηλώσει μέσω του συστήματος;”.

Επίσης αν ο χρήστης έχει δηλώσει κάποια άλλη στιγμή κατά την πλοήγηση του στο σύστημα εξάμηνο και τμήμα τότε γίνεται η ακόλουθη ερώτηση για να βεβαιωθούμε ότι ο χρήστης θέλει τα μαθήματα που αντιστοιχούν στα στοιχεία που έχει δηλώσει.

“θέλετε το πρόγραμμα που αντιστοιχεί στο 3ο εξάμηνο και στο τμήμα ηλεκτρονικών μηχανικών και μηχανικών υπολογιστών, σωστά?”

Action

if < Ναι> then

go to: **Ανακοίνωση Προγράμματος**

else if <Όχι>

go to: **Πρόγραμμα Εργαστηρίων**

end if

Ανακοίνωση Προγράμματος

Αρχική ανακοίνωση

“Για να ακούσετε το πρόγραμμα κάποιας μέρας απλά πείτε την μέρα, πχ Τρίτη. Για κάποια μέρα και ώρα πείτε πρώτα την μέρα και μετά την ώρα, πχ σήμερα 9 η ώρα. Αν θέλετε να μάθετε για κάποιο συγκεκριμένο μάθημα απλά πείτε το μάθημα, πχ Διαφορικός Λογισμός 1. Τέλος μπορείτε να ακούσετε όλο το πρόγραμμα λέγοντας απλά, πρόγραμμα. ”

Παραδείγματα εκφράσεων

Εκφραση Χρήστη	Μεταβάσεις, τιμές
Σήμερα Αύριο Τρίτη Την Τρίτη	<μετάβαση <u>ΑνακοίνωσηΜέρας</u> > <μέρα X>
Σήμερα, 11 η ώρα Αύριο. στις 11 η ώρα Τρίτη, 11 Την Τρίτη, στις 11	<μετάβαση <u>ΑνακοίνωσηςΜέραςΩρας</u> > <μέρα X> <ώρα Y>
Διαφορικός Λογισμός 1	<μετάβαση <u>ΑνακοίνωσηΜαθήματος</u> > <μάθημα X>
Πρόγραμμα Όλο το πρόγραμμα Το πρόγραμμα	<μετάβαση <u>ΑνακοίνωσηΠρογράμματος</u> > <πρόγραμμα>

Action

if [μέρα]= valid,[ώρα]=NULL

Ανακοίνωση προγράμματος χρησιμοποιώντας την επιθυμητή Μέρα.

else if [μέρα]= valid,[ώρα]=valid

```
Ανακοίνωση προγράμματος χρησιμοποιώντας την επιθυμητή Μέρα και Ώρα.  
else if [μέρα]= NULL,[ώρα]=valid  
    Ανακοίνωση προγράμματος χρησιμοποιώντας την επιθυμητή Ώρα.  
else if [μάθημα]=valid  
    Ανακοίνωση προγράμματος χρησιμοποιώντας το επιθυμητό μάθημα.  
else if <πρόγραμμα>  
    Αναλυτική ανακοίνωση όλου του προγράμματος.  
end if.
```

Η ανακοίνωση των αποτελεσμάτων γίνεται κατά τον ίδιο τρόπο με την **Ανάγνωση Ανακοινώσεων**. Επίσης ανάλογη δράση έχουμε και για το πρόγραμμα εργαστηρίων και της εξεταστικής.

Βαθμολογίες Μαθημάτων και Εργατηρίων

if [κωδ_προσβ] != -1

if [grade_info > != 1 (αν έχει κάνει login και έχει ήδη ακούσει βαθμολογίες)

Go to: Ανάκτηση βαθμολογιών

else go to: Ανακοίνωση νέων καταχωρήσεων για βαθμολογίες μαθημάτων

else if <pin> is not valid

Play Prompt: “*Για να ακούσετε τις βαθμολογίες των μαθημάτων πρέπει πρώτα να κάνετε λογκίν. Θέλετε να εγγραφείτε στο σύστημα*”

if <Ναι> then

go to: Εγγεγραμμένοι Χρήστες

else go to: Ακαδημαϊκά

end if

disconnect

end if

Ανάκτηση βαθμολογιών:

Αρχική ανακοίνωση

“*Παρακαλώ πείτε μου το μάθημα ή το εργαστήριο για το οποίο θέλετε να μάθετε τον βαθμό σας.*”

Παραδείγματα εκφράσεων

Εκφραση Χρήστη	Μετάβαση τιμές
Οντοκεντρικός Προγραμματισμός	<μετάβαση <u>Ανάκτηση Βαθμολογιών</u> > <όνομα Οντοκεντρικός Προγραμματισμός > <course_lab μάθημα>
Εργαστήριο βασική θεωρία κυκλωμάτων	<μετάβαση <u>Ανάκτηση Βαθμολογιών</u> > <όνομα Εργαστήριο βασική θεωρία κυκλωμάτων ><course_lab εργαστήριο>

Διαχείριση λάθους.

Πρώτο:

“ Παρακαλώ πείτε μου το όνομα του μαθήματος ή του εργαστηρίου”

Δεύτερο:

“Παρακαλώ πείτε τι θα θέλατε να κάνετε. Αν έχετε δυσκολίες πείτε βοήθεια, ενώ μπορείτε να πείτε και επανάληψη για να ξανακούσετε τις επιλογές σας.”

Τρίτο:

“Για να αντιμετωπίσετε το πρόβλημα σας δίνονται οι παρακάτω οδηγίες. ”

Ανακοίνωση Βοήθειας

“ Μπορείτε να ακούσετε την βαθμολογία σε κάποιο μάθημα ή εργαστήριο που έχετε δηλώσει λέγοντας απλά το όνομά του”

Τηλεφωνικός Κατάλογος Υπηρεσιών και Προσώπων

Εδώ ο χρήστης οδηγείται για να πάρει πληροφορίες τηλεφωνικού καταλόγου σχετικά με πρόσωπα που δραστηριοποιούνται στον χώρο του Πολυτεχνείου Κρήτης, καθώς και υπηρεσίες και εργαστήρια του πολυτεχνείου.

Αρχική ανακοίνωση

“Πείτε μου το όνομα της υπηρεσίας ή του προσώπου για το οποίο ψάχνετε πληροφορίες.”

Ανακοίνωση Βοηθείας

“Παρακαλώ πείτε την ακριβή ονομασία της υπηρεσίας ή του προσώπου για το οποίο ζητάτε πληροφορίες. πχ Βασίλειος Διγαλάκης ή Εργαστήριο Τηλεπικοινωνιών. ”

Παραδείγματα Εκφράσεων

Έκφραση Χρήστη	Μεταβάσεις, τιμές
Βασίλειος Διγαλάκης	<μετάβαση <u>Τηλεφωνικός Κατάλογος</u> > <όνομα Βασίλειος Διγαλάκης>
Διγαλάκης	<μετάβαση <u>Τηλεφωνικός Κατάλογος</u> > <όνομα Διγαλάκης>
Εργαστήριο Τηλεπικοινωνιών	<μετάβαση <u>Τηλεφωνικός Κατάλογος</u> > <όνομα Εργαστήριο Τηλεπικοινωνιών>
Διγαλάκης Βασίλειος	<μετάβαση <u>Τηλεφωνικός Κατάλογος</u> > <όνομα Βασίλειος Διγαλάκης>

Action

if [όνομα]=valid then

Ανακοίνωση: “[όνομα] το τηλέφωνο είναι [αρ τηλεφώνου]”.

else if <action τίποτα> then

Ανακοίνωση: “*παρακαλώ πείτε προσεχτικά όλη την ονομασία της υπηρεσίας ή του προσώπου.*”

else if <action ΚενΜεν >

go to: Κεντρικό Μενού
else
 Διαχείριση λάθους
end if.

Διαχείριση λάθους.

Πρώτο:

“Αν ψάχνετε κάποιο συγκεκριμένο πρόσωπο πείτε μου το όνομά του ακολουθούμενο από το επίθετό του, αν ψάχνετε για κάποια υπηρεσία πείτε μου ολόκληρη την ονομασία της.”

Δεύτερο:

“Παρακαλώ πείτε μου το πρόσωπο ή την υπηρεσία που σας ενδιαφέρει. Αν έχετε δυσκολίες πείτε βοήθεια, ενώ μπορείτε να πείτε και επανάληψη για να ξανακούσετε τις επιλογές σας.”

Τρίτο:

“Για να αντιμετωπίσετε το πρόβλημα σας δίνονται οι παρακάτω οδηγίες.”

<goto: Οδηγίες>

Υπηρεσίες του πολυτεχνείου

Σε αυτό το κομμάτι της εφαρμογής υπάρχουν διαθέσιμες δυο επιλογές, οι οποίες αφορούν τις υπηρεσίες που προσφέρει το πολυτεχνείο μέσω της εφαρμογής. Αυτές είναι, η υπηρεσία Φωνητικού Ταχυδρομείου και η υπηρεσία Αναζήτησης Γιατρών.

Αρχική ανακοίνωση:

“Παρακαλώ πείτε «γιατροί», ή φωνητικό ταχυδρομείο ”

Ανακοίνωση Βοήθειας

“Μπορείτε να αναζητήσετε κάποιον γιατρό που έχει σύμβαση με το πολυτεχνείο Κρήτης λέγοντας υπηρεσία αναζήτησης γαιτρών, ενώ μπορείτε να ακούσετε τα φωνητικά σας μηνύματα λέγοντας υπηρεσία φωνητικού ταχυδρομείου”

Παραδείγματα Εκφράσεων

Εκφραση Χρήστη	Μεταβάσεις, τιμές
Παρακαλώ δώστε μου /θα ήθελα Γιατροί, Γιατρούς	<μετάβαση <u>Γιατροί</u> > <Γιατροί>
Φωνητικό Ταχυδρομείο Ταχυδρομείο, voicemail	<μετάβαση <u>Φωνητικό Ταχυδρομείο</u> > < Φωνητικό Ταχυδρομείο > (αν έχει κάνει ήδη login) αλλιώς, <μετάβαση <u>Εγγεγραμμένοι Χρήστες</u> > <εγγεγραμμένοι χρήστες>

Action:

```
if < γιατροί> then
    go to: Γιατροί
else if < Φωνητικό Ταχυδρομείο >
    if [user_id] = valid then
```

```
        go to: Φωνητικό Ταχυδρομείο
    else
        go to: Εγγεγραμμένοι Χρήστες
end if
```

Διαχείριση λάθους.

Πρώτο:

“Παρακαλώ πείτε ένα απο τα εξής: συμβεβλημένοι γιατροί για να αναζητήσετε κάποιον γαιτρό, ή πείτε φωνητικό ταχυδρομείο για να ακούσετε τα φωνητικά σας μηνύματα”

Δεύτερο:

“Παρακαλώ πείτε τι θα θέλατε να κάνετε.Αν έχετε δυσκολίες πείτε βοήθεια, ενώ μπορείτε να πείτε και επανάληψη για να ξανακούσετε τις επιλογές σας.”

Τρίτο:

“Για να αντιμετωπίσετε το πρόβλημα σας δίνονται οι παρακάτω οδηγίες.”

<goto: Οδηγίες>

Γιατροί

Αρχική ανακοίνωση:

“Παρακαλώ πείτε την ειδικότητα του γιατρού που ψάχνετε ή πείτε το όνομα του ”

Ανακοίνωση Βοήθειας

“Μπορείτε να αναζητήσετε κάποιον γιατρό που έχει σύμβαση με το πολυτεχνείο Κρήτης λέγοντας την ειδικότητά του, όπως παθολόγος, ενώ μπορείτε να πείτε και το όνομά του εφόσον το ξέρετε πχ Αλέξανδρος Λυκουρέζος ”

Παραδείγματα Εκφράσεων

Εκφραση Χρήστη	Μεταβάσεις, τιμές
Αλέξανδρος Λυκουρέζος	<μετάβαση <u>Γιατροί</u> > <όνομα Αλέξανδρος Λυκουρέζος >

Διαχείριση λάθους.

Πρώτο:

“Παρακαλώ πείτε το όνομα του γιατρού που ψάχνετε ή την ειδικότητά του.”

Δεύτερο:

“Παρακαλώ πείτε τι θα θέλατε να κάνετε. Αν έχετε δυσκολίες πείτε βοήθεια, ενώ μπορείτε να πείτε και επανάληψη για να ξανακούσετε τις επιλογές σας.”

Τρίτο:

“Για να αντιμετωπίσετε το πρόβλημα σας δίνονται οι παρακάτω οδηγίες.”

Φωνητικό ταχυδρομείο

```
if [user_id]= SUPERUSER then
    go to: Ηχογράφηση Μηνύματος
else if [user_id] =USER then
    if [mail_visited] != -1 then
        go to: Ανάκτηση Φωνητικών Μηνυμάτων
    else
        go to: Αναγγελία Νέων Φωνητικών Μηνυμάτων
else
    go to: Εγγεγραμμένοι Χρήστες
```

Ανάκτηση Φωνητικών Μηνυμάτων:

Αρχική ανακοίνωση

“ Για ποιο μάθημα ή εργαστήριο θέλετε να ακούσετε αν υπάρχει φωνητικό μήνυμα;”

Παραδείγματα εκφράσεων

Εκφραση Χρήστη	Μετάβαση τιμές
Οντοκεντρικός Προγραμματισμός	<μετάβαση Ανάγνωση Φωνητικών Μηνυμάτων><όνομα Οντοκεντρικός Προγραμματισμός ><course_lab μάθημα>
Εργαστήριο βασική θεωρία κυκλωμάτων	<μετάβαση Ανάγνωση Φωνητικών Μηνυμάτων ><όνομα Εργαστήριο βασική θεωρία κυκλωμάτων ><course_lab εργαστήριο>

Διαχείριση λάθους.

Πρώτο:

“ Παρακαλώ πείτε μου το όνομα του μαθήματος ή του εργαστηρίου”

Δεύτερο:

“Παρακαλώ πείτε τι θα θέλατε να κάνετε.Αν έχετε δυσκολίες πείτε βοήθεια, ενώ μπορείτε να πείτε και επανάληψη για να ξανακούσετε τις επιλογές σας.”

Τρίτο:

“Για να αντιμετωπίσετε το πρόβλημα σας δίνονται οι παρακάτω οδηγίες. ”

Ανακοίνωση Βοήθειας

“ Μπορείτε να ακούσετε τα φωνητικά μηνύματα σε σε κάποιο μάθημα ή εργαστήριο που έχετε δηλώσει λέγοντας απλά το όνομά του”

Ανάγνωση Φωνητικών Μηνυμάτων - Αναγγελία Νέων Φωνητικών Μηνυμάτων:

Παρομοίως με την Ανάγνωση Ανακοινώσεων.

Ηχογράφηση Μηνύματος:

Σε αυτήν την κατάσταση φτάνει κάποιος SUPERUSER, ο οποίος ανήκει στους καθηγητές. Αυτό γίνεται αφού έχει περάσει από την διαδικασία ταυτοποίησης.

Αρχική ανακοίνωση

“Θα θέλατε να στείλετε κάποιο φωνητικό μήνυμα;”

Γραμ/κη ΝαιΟχι-ανακ

Πιθανές μεταβάσεις: <μετάβαση> ΝαιΟχι

< ΝαιΟχι > Ναι

Οχι

Παραδείγματα Εκφράσεων:

Έκφραση Χρήστη	Μετάβαση, τιμές
Ναι	<μετάβαση <u>Ηχογράφηση</u> > <_Ηχογράφηση >
Οχι	<μετάβαση <u>Κεντρικό Μενού</u> > < Κεντρικό Μενού >

Action

if < Ναι > then

go to: Ηχογράφηση

else

go to: Κεντρικό Μενού

Ηχογράφηση 1:

Αρχική ανακοίνωση

“Για ποιο μάθημα προορίζεται το μήνυμα;”

Ανακοίνωση Βοήθειας

“Παρακαλώ πείτε μου το μάθημα, για τους φοιτητές του οποίου προορίζεται το μήνυμα.”

Παραδείγματα Εκφράσεων:

Εκφραση Χρήστη	Μετάβαση, τιμές
Για την Φυσική 2	<Φυσική 2>

Action

if [όνομα μαθήματος] = valid then

go to: **Ηχογράφηση 2**

Διαχείριση λάθους.

Πρώτο:

“Με τους μαθητές ποιου μαθήματος σχετίζεται το μήνυμα που θέλετε να ηχογραφήσετε;”

Δεύτερο:

“Παρακαλώ πείτε τι θα θέλατε να κάνετε. Αν έχετε δυσκολίες πείτε βοήθεια, ενώ μπορείτε να πείτε και επανάληψη αν δεν καταλάβατε τι σας ζητείται. ”

Τρίτο:

“Για να αντιμετωπίσετε το πρόβλημα σας δίνονται οι παρακάτω οδηγίες. ”

Ηχογράφηση 2:

Ανακοίνωση

“Πείτε το μήνυμά σας μετά τον χαρακτηριστικό ήχο”

-ηχογράφηση μηνύματος-

-αποστολή μηνύματος στους φοιτητές του επιθυμητού μαθήματος-

Εγγεγραμμένοι Χρήστες

Εδώ ο χρήστης οδηγείται σε περίπτωση που θέλει να κάνει login για να έχει μια πιο προσωποποιημένη πλοήγηση στο σύστημα ή αν έχει ζητήσει να έχει πρόσβαση στο φωνητικό του ταχυδρομείο χωρίς προηγουμένως να έχει κάνει login.

Αρχική ανακοίνωση

“Παρακαλώ πείτε μου η πληκτρολογήστε τον τετραψήφιο κωδικό πρόσβασης ”

Ανακοίνωση Βοηθείας

“Σας ζητείται να πείτε τον τετραψήφιο κωδικό πρόσβασης αν θέλετε να εγγραφείτε στο σύστημα για να ακούσετε τα φωνητικά σας μηνύματα ή να έχετε μια προσωποποιημένη πλοήγηση”

Παρδείγματα εκφράσεων:

Εκφραση Χρήστη	Μετάβαση, Τιμές
Ο κωδικός μου είναι ένα δύο τρία τέσσερα Ένα δύο τρία τέσσερα	<μετάβαση <u>Κωδικός Πρόσβασης</u> > <κωδικός 1234>

Action

if < **κωδικός** > is valid and confidence > 60

[**κωδικός**] = < **κωδικός** >

Go to: **Επιβεβαίωση**

else if <user_id> is not valid

if first or second attempt,

Play Prompt: ***“παρακαλώ προσπαθείστε ξανα πιο καθαρά..”***

Return to top of state.

end if

else if [attempt] = 3,

Play Prompt: ***“προμηθευτείτε τον κωδικό πρόσβασης και προσπαθήστε ξανά”***

disconnect

end if

Διαχείριση λάθους.

Πρώτο:

“πείτε μου τον κωδικό πρόσβασης για να κάνετε login. Ο ο αριθμός πρέπει να έχει ακριβώς τέσσερα ψηφία”

Δεύτερο:

“ Αν έχετε δυσκολίες πείτε βοήθεια, ενώ μπορείτε να πείτε και επανάληψη αν δεν καταλάβετε τι σας ζητείται. ”

Τρίτο:

“Για να αντιμετωπίσετε το πρόβλημα σας δίνονται οι παρακάτω οδηγίες. ”

Επιβεβαίωση

Αφού έχουμε κάνει αναζήτηση στην βάση με τον κωδικό πρόσβασης που εισήγαγε ο χρήστης και έχουμε βρει το όνομα στο οποίο αυτό αντιστοιχεί, ζητάμε από τον χρήστη να το επιβεβαιώσει για να προχωρήσουμε στις υπόλοιπες εργασίες..

Αρχική ανακοίνωση

“Ονομάξεστε [name], σωστά;”

Πιθανές μεταβάσεις :

Φωνητικό Ταχυδρομείο

Εγγεγραμμένοι Χρήστες

< τιμές >

Ναι

Όχι

Παραδείγματα Εκφράσεων:

Έκφραση Χρήστη	Μετάβαση, τιμές
Ναι	<μετάβαση <u>Φωνητικό Ταχυδρομείο</u> >

	<Ναι>
Όχι	<μετάβαση <u>Εγγεγραμμένοι Χρήστες</u> > <Όχι>

Action

if < Ναι> then

go to: Φωνητικό Ταχυδρομείο

else go to Εγγεγραμμένοι Χρήστες

Appendix B – Usability Questionnaire

Φωνητική Πύλη του Πολυτεχνείου Κρήτης

Η Φωνητική Πύλη του Πολυτεχνείου Κρήτης αποτελεί ένα σύστημα βασισμένο στην επεξεργασία φωνής, το οποίο σας επιτρέπει να πάρετε πληροφορίες σχετικά με διάφορα θέματα. Οι πληροφορίες που παρέχονται ανήκουν στις ακόλουθες κατηγορίες:

- ακαδημαϊκά θέματα
- τηλεφωνικός κατάλογος
- υπηρεσίες

Στα ακαδημαϊκά θέματα μπορείτε να βρείτε πληροφορίες όπως:

- Εβδομαδιαίο πρόγραμμα μαθημάτων
- Εβδομαδιαίο πρόγραμμα εργαστηρίων
- Πρόγραμμα εξεταστικής μαθημάτων και εργαστηρίων
- Βαθμολογίες μαθημάτων και εργαστηρίων

Στον τηλεφωνικό κατάλογο μπορείτε να βρείτε πληροφορίες όπως:

- τηλέφωνα υπηρεσιών
- τηλέφωνα προσώπων απασχολούμενων στο πολυτεχνείο

Στις υπηρεσίες θα βρείτε :

- την υπηρεσία αναζήτησης γιατρών, όπου μπορείτε να ψάξετε κάποιον γιατρό, ο οποίος έχει σύμβαση με το πολυτεχνείο, με βάση την ειδικότητά του ή το όνομά του
- την υπηρεσία φωνητικού ταχυδρομείου, όπου μπορείτε να ακούσετε τα φωνητικά μηνύματα που έχουν στείλει καθηγητές στους φοιτητές των μαθημάτων στα οποία έχετε δηλωθεί

Αν κάνετε login στο σύστημα, το σύστημα μπορεί να κάνει πιο εύκολη την πλοήγησή σας, καθώς μπορεί να ελέγξει άμεσα για θέματα που σας αφορούν, όπως η ύπαρξη νέου

φωνητικού μηνύματος, ενώ θα έχετε την δυνατότητα να ενημερώνεστε μόνο για το πρόγραμμα μαθημάτων και εργαστηρίων που έχετε δηλώσει.

Κατά την πλοήγηση σας θα έχετε άφθονη βοήθεια από τις αντίστοιχες ανακοινώσεις, ενώ μπορείτε πάντα να πείτε βοήθεια, επανάληψη και πίσω. Για να βγείτε μπορείτε να πείτε ανά πάσα στιγμή έξοδος ή να κλείσετε το τηλέφωνο.

1. Πρώτη αποστολή: μπείτε στο σύστημα για να μάθετε

- το πρόγραμμα των μαθημάτων κάποιας μέρας, π.χ. της δευτέρας.
- τι εργαστήριο έχετε σε κάποια συγκεκριμένη ώρα, π.χ. στις 9 ή την Παρασκευή στις 2
- το πρόγραμμα της εξεταστικής.
- πότε γράφετε κάποιο συγκεκριμένο μάθημα, π.χ. το μάθημα Οντοκεντρικός Προγραμματισμός

Παρακαλώ καταγράψτε τα προβλήματα που αντιμετωπίσατε:

2. Δεύτερη αποστολή: θα κάνετε αναζήτηση στον τηλεφωνικό κατάλογο.

Ενδεικτικά ονόματα:

Ζερβάκης Μιχάλης(ή απλά Ζερβάκης)

Παντινάκης Απόστολος

Αθηνά Παλιουδάκη

Μάρκος Παπαγεωργίου

Ενδεικτικές υπηρεσίες:

Εργαστήριο αυτοματισμού

Εργαστήριο βιομηχανικής παραγωγής με τη βοήθεια ηλεκτρονικού

υπολογιστή

εργαστήριο γεωστατιστικής

τμήμα μηχανικών ορυκτών πόρων

Η αναζήτηση στον κατάλογο γίνεται με βάση τον κατάλογο που είναι αναρτημένος στην σελίδα του πολυτεχνείου <http://www.tuc.gr/katalogoi/index.html>. Όσοι από σας είστε μέλη της πολυτεχνικής κοινότητας μπορείτε να δοκιμάσετε και άλλα ονόματα που έχετε κατά

νου... οι υπόλοιποι θα πρέπει να αρκεστείτε στα ενδεικτικά ονόματα ή να επισκεφτείτε την σελίδα με τον τηλεφωνικό κατάλογο.

Παρακαλώ καταγράψτε τα προβλήματα που αντιμετωπίσατε:

3. Τρίτη αποστολή: Θα κάνετε αναζήτηση γιατρού.

Παρακαλώ καταγράψτε τα προβλήματα που αντιμετωπίσατε:

4. Τέταρτη αποστολή: Θα εγγραφείτε στο σύστημα

Παρακαλώ καταγράψτε τα προβλήματα που αντιμετωπίσατε:

5. Πέμπτη αποστολή: Θα ακούσετε τα νέα σας φωνητικά μηνύματα και στη συνέχεια θα ζητήσετε να ακούσετε το φωνητικό μήνυμα για κάποιο συγκεκριμένο μάθημα ή εργαστήριο.

Ενδεικτικό μάθημα: Βασική Θεωρία Κυκλωμάτων.

Παρακαλώ καταγράψτε τα προβλήματα που αντιμετωπίσατε:

6. Έκτη αποστολή: Θα ακούσετε τις νέες ανακοινώσεις βαθμολογιών και στη συνέχεια θα ζητήσετε να ακούσετε τον βαθμό για κάποιο συγκεκριμένο μάθημα ή εργαστήριο. Ενδεικτικό μάθημα: Ψηφιακοί Υπολογιστές.

Παρακαλώ καταγράψτε τα προβλήματα που αντιμετωπίσατε :

Υποθέσεις του σεναρίου:

Λέγεστε: Θεόδωρος Γερμάνης

Το ΑΜ σας είναι: 2000030082

Φοιτείτε στο τμήμα ΗΜΜΥ(Ηλεκτρονικών Μηχανικών και Μηχανικών Υπολογιστών) και βρίσκεστε στο 3 εξάμηνο.

Τα ενδεικτικά ονόματα ημερών, μαθημάτων και ανθρώπων ισχύουν αλλά είστε ελεύθεροι να δοκιμάσετε και άλλες τιμές. Επίσης προσπαθήστε να είστε όσο το δυνατόν πιο χαλαροί στις εκφράσεις σας και να μιλάτε με σιγουριά. Μην περιμένετε να είναι τέλειο, (για αυτό άλλωστε γίνονται αυτές οι δοκιμές). Καταγράψτε εκφράσεις όπου το σύστημα δυσκολευόταν να σας καταλάβει...

Το τηλέφωνο κλήσης είναι 28210-37543

Ημερομηνία:

Ωρα:

Ευχαριστώ για τον χρόνο σας

Appendix C – Technical Issues

Αξίζει να σημειώσουμε εδώ κάποια τεχνικά χαρακτηριστικά της εργασίας. Όπως αναφέραμε και προηγουμένως η υλοποίηση της εργασίας στηρίχθηκε στην χρήση μιας σειράς γλωσσών προγραμματισμού και τεχνολογιών. Παίρνοντας τις με τη σειρά με την οποία χρησιμοποιήθηκαν έχουμε:

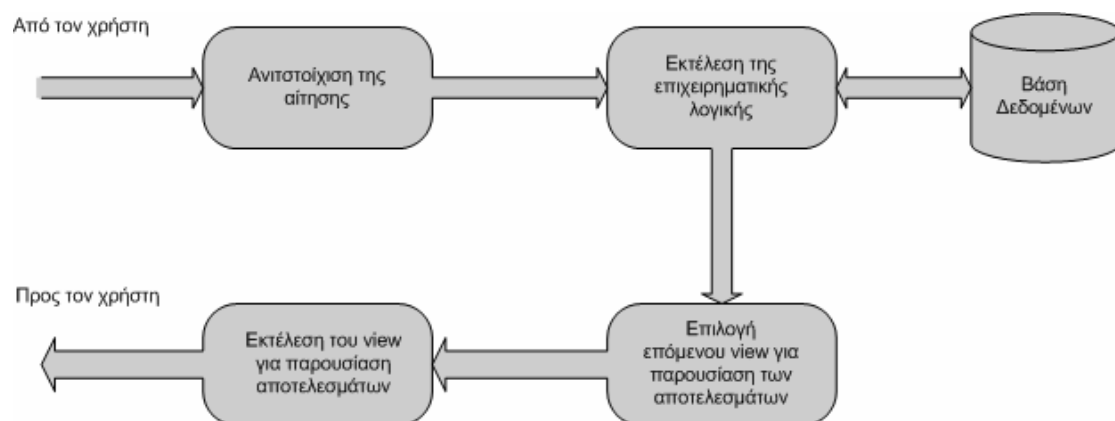
VoiceXML:	6000	programming lines
XML :	9500	programming lines
GSL :	450	programming lines
Java :	10395	programming lines
ECMA Script	2500	programming lines
Perl :	450	programming lines

Κατά την διάρκεια της εργασίας αντιμετωπίσαμε διάφορα προβλήματα που είχαν να κάνουν, είτε, πολλές φορές, με τα εργαλεία που χρησιμοποιούσαμε είτε και με την φύση αυτή καθαυτή του project. Ένα πρόβλημα που μας ταλαιπώρησε καθ' όλη την διάρκεια της υλοποίησης ήταν αυτό της χρήσης των ελληνικών. Ξεκινώντας από την βάση δεδομένων και συνεχίζοντας με την διασύνδεσή της μέσω των εργαλείων που είχαμε η διαδρομή που έπρεπε να διανύσουν οι ελληνικοί χαρακτήρες μέχρι να φτάσουν στα αυτιά ή τα μάτια του χρήστη ήταν μια ολόκληρη οδύσσεια. Άρα στην βάση δεδομένων οι ελληνικοί χαρακτήρες σώζονται ως VARCHAR και οι αντίστοιχοι πίνακες έχουν character encoding UTF-8. Στην συνέχεια στην σύνδεση του ODBC Driver ορίζουμε την επιλογή `useUnicode=true&characterEncoding=UTF-8` για να μεταφέρει άθικτα τα δεδομένα. Και τέλος για να εμφανίσουμε στον χρήστη τους χαρακτήρες σε UTF-8 ορίζουμε στην επικεφαλίδα κάθε σελίδας jsp την επιλογή `charset=utf-8" pageEncoding="utf-8` όπως και την εντολή `request.setCharacterEncoding(ut);`. Και βουαλά! ακούμε το σύστημα να μας χαιρετάει. Ένα μικρό βήμα για την εφαρμογή, ένα μεγάλο για τον προγραμματιστή που δεν χρειάζεται να παιδεύεται άλλο. Ένα άλλο πρόβλημα που αντιμετωπίσαμε με τα ελληνικά ήταν στον υπολογισμό του Word Error Rate μέσω του sclite. Το πρόβλημα είχε να κάνει με το γεγονός ότι το συγκεκριμένο εργαλείο δεν μπορούσε να επεξεργαστεί ελληνικούς χαρακτήρες.

Οπότε αναγκαστήκαμε να χρησιμοποιήσουμε ένα πρόγραμμα υλοποιημένο σε Perl που μετέτρεπε τα ελληνικά των hypothesis και transcription files σε greeklish, για να μπορέσουμε να κάνουμε τις απαραίτητες συγκρίσεις. Επίσης για την διασύνδεση της βάσης με το τα Servlets χρησιμοποιήσαμε connection pooling, μια μέθοδο που αυτοματοποιεί την διασύνδεση και βολεύει πολύ όταν έχουμε να κάνουμε με πολλές διασυνδέσεις.

i. Η εφαρμογή του MVC στο σύστημά μας

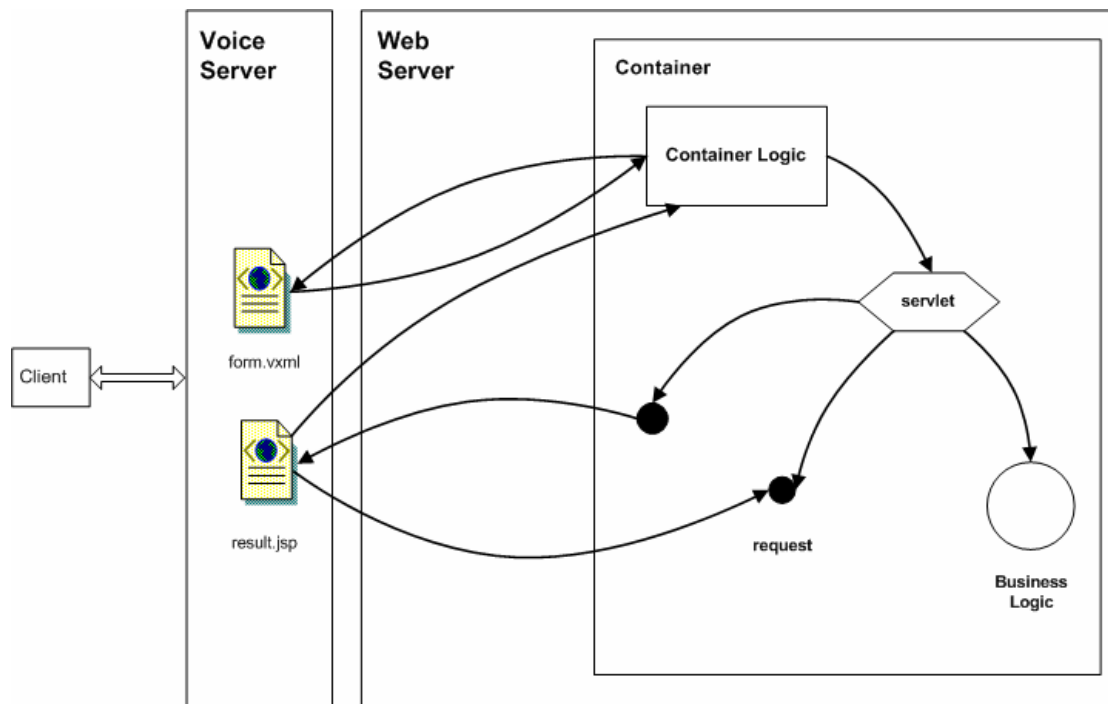
Παρακάτω θα δούμε πως υλοποιήσαμε το MVC στην εφαρμογή μας. Η βασική λειτουργία του φαίνεται στο Σχήμα 15:



Σχήμα 15 Περιγραφή εκτέλεσης αίτησης του χρήστη

Ο controller δέχεται μια αίτηση από τον χρήστη μέσω HTTP και καλεί την αντίστοιχη μέθοδο της επιχειρηματικής λογικής που περιλαμβάνεται στο μοντέλο. Ανάλογα με τα αποτελέσματα που επιστρέφει η μέθοδος, ο ελεγκτής επιλέγει την κατάλληλη όψη που θα κληθεί. Τέλος, ο ελεγκτής παράγει την όψη αυτή και αυτή παρουσιάζεται στον χρήστη.

Το μοντέλο που χρησιμοποιήσαμε στην υλοποίηση της εφαρμογής μας φαίνεται στην Σχήμα 16:



Σχήμα 16 Δομή της αρχιτεκτονικής της εφαρμογής χρησιμοποιώντας το μοντέλο MVC

Όπως φαίνεται και στο παραπάνω σχήμα, μέσω του servlet container (Tomcat) [33] χρησιμοποιείται ένα servlet που έχει τον ρόλο του controller ανάμεσα στον voice browser και στην σελίδα JSP που πρόκειται να παραχθεί. [32] Ο controller αυτός περιέχει όλη την λογική που αφορά την δρομολόγηση των αιτήσεων(request dispatching) βάσει των URL των αιτήσεων, την διαχείριση των εισόδων και την κατάσταση της εφαρμογής, ενώ χειρίζεται και την επιλογή των JSP σελίδων που θα παρουσιαστούν στο χρήστη διαχωρίζοντας, κατά αυτόν τον τρόπο τα servlets και τα JSP.

Όλες οι αιτήσεις οδηγούνται σε ένα και μοναδικό servlet μέσω ενός αρχείου XML που λειτουργεί ως servlet mapping. Το servlet mapping είναι μια δομή όπου αποθηκεύονται όλα τα servlet τα οποία έχουν γίνει deploy στον servlet container, καθώς και οι παράμετροί τους. Δίνεται το παρακάτω mapping:

```
<servlet>
  <servlet-name>ActionController</servlet-name>
  <servlet-class>tucvoiceportal.web.ActionController</servlet-class>
  <init-param>
    <param-name>Announcements</param-name>
    <param-value>tucvoiceportal.web.Announcements</param-value>
  </init-param>
```

```

....
....
....
</servlet>
<servlet-mapping>
  <servlet-name>ActionController</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>

```

Μια δομή τύπου if then else, για την επιλογή των servlets που πρέπει να εκτελεστούν κάθε φορά, θα οδηγούσε σε αρκετά μεγάλη σε μέγεθος και πολυπλοκότητα μέθοδο. Εναλλακτικά, λοιπόν, χρησιμοποιήσαμε μια διαφορετική προσέγγιση. Ορίσαμε, όπως φαίνεται και από το servlet mapping που δώσαμε παραπάνω μόνο ένα κεντρικό servlet, το ActionController. Στην συνέχεια δηλώσαμε ένα interface με το όνομα IAction, που υλοποιεί μια και μόνο λειτουργία την invoke().

```

public interface IAction {
    void invoke(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException;
}

```

Ο ActionController μπορεί να ψάξει για καθορισμένες υλοποιήσεις της IAction, βάσει του ονόματός τους. Αν η διεύθυνση του servlet στον δίσκο είναι η <http://.../tucVoicePortal/WEB-INF>, τότε η αίτηση θα είχε την μορφή <http://.../tucVoicePortal/Announcements.do>. Άρα ο controller θα ψάξει μια υλοποίηση της IAction με το όνομα Announcements. Όλες οι αιτήσεις οδηγούνται στον κεντρικό controller ActionController ο οποίος και κάνει με την σειρά του την δρομολόγησή τους στις υπηρεσίες που εξυπηρετούν τις εκάστοτε αιτήσεις:

```

public class ActionController extends HttpServlet {

    //Initialize global variables
    public void init() throws ServletException {
        _controllers = new Hashtable();
    }
}

```

```

Enumeration enum = getServletConfig().getInitParameterNames();

while (enum.hasMoreElements())
{
    String controller_name = enum.nextElement().toString();
    String class_name = getServletConfig().getInitParameter(controller_name);
    try
    {
        Class toRun = Class.forName(class_name);
        IAction controller = (IAction)toRun.newInstance();
        _controllers.put(controller_name, controller);
    }
    catch (Exception class_exception)
    {
        //Smth
    }
}

//Process the HTTP Get request
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
    throws ServletException, IOException
{

    String uri = request.getRequestURI();
    int con_start = uri.lastIndexOf("/");
    int con_end = uri.lastIndexOf(".do");
    String req_name = uri.substring(con_start+1, con_end);

    if (_controllers.containsKey(req_name))
    {
        IAction controller = (IAction)_controllers.get(req_name);
        controller.invoke(request, response);
    }

}

public void doPost(HttpServletRequest request,
                  HttpServletResponse response)

```



```

        throws ServletException, IOException
    {

        String uri = request.getRequestURI();
        int con_start = uri.lastIndexOf("/");
        int con_end = uri.lastIndexOf(".do");
        String req_name = uri.substring(con_start+1, con_end);

        if (_controllers.containsKey(req_name))
        {
            IAction controller = (IAction)_controllers.get(req_name);
            controller.invoke(request, response);
        }

    }

    //Clean up resources
    public void destroy() {
    }

    private Hashtable _controllers;
}

```

Όπως βλέπουμε, ο ελεγκτής εξυπηρετεί HTTP requests GET και POST. Ο controller περιέχει έναν πίνακα κατακερματισμού (Hashtable) με αντικείμενα τύπου IAction, που το καθένα έχει για δείκτη το όνομά του. Όταν φορτώνεται το servlet (ActionController) καλείται η μέθοδος init, η οποία γεμίζει τον πίνακα αυτόν με αντικείμενα τύπου IAction που υλοποιούν λειτουργίες του μοντέλου. Κατά αυτόν τον τρόπο κάθε φορά που το servlet δέχεται μια αίτηση για κάποια λειτουργία, ψάχνει στον πίνακα για μια καταχώρηση με το όνομά της και εκτελεί την κλάση αυτή καλώντας την μέθοδό της invoke με τις παραμέτρους της αίτησης. Στη συνέχεια ο controller (servlet) χρησιμοποιεί τα αποτελέσματα που επιστρέφει το μοντέλο για να καθορίσει το πια όψη θα κληθεί στη συνέχεια. Παρακάτω δίνεται και ένας ενδεικτικός κώδικας για την υποκλάση της IAction που χρησιμοποιεί το μοντέλο και τελικά αποφασίζει ποια όψη (JSP) θα κληθεί στη συνέχεια.

```

public Announcements() {
}

public void invoke(HttpServletRequest request,
                    HttpServletResponse response)
    throws ServletException, IOException
{
    TucVpData td = new TucVpData();
    ArrayList Announcements = new ArrayList();
    //Calling the model method that executes business logic
    Announcements = td.getAnnouncements();

    request.setAttribute("Announcements", Announcements);

    // We initiate a request dispatcher for the JSP
    RequestDispatcher view =
    request.getRequestDispatcher("listen_announcements.jsp");
    // Use the request dispatcher to crank up the JSP, sending it
    // the request and the response
    view.forward(request, response);

    return;
}

```

Παρακάτω δίνεται ενδεικτικά και ο κώδικας της μεθόδου του μοντέλου που υλοποιεί την επιχειρηματική λογική της εφαρμογής.

```

public ArrayList getAnnouncements()
{
    ArrayList all_announcements = new ArrayList();
    Hashtable announcement = new Hashtable();

    try
    {
        // Load the JDBC driver
        Class.forName("com.mysql.jdbc.Driver").newInstance();
    }
    catch (Exception e)

```

```

{
    //      Let's hope we never reach this point
    System.err.println("Driver '"+ e +"' not found:" + e.getMessage() );
}
try
{
    //      Set up the connection to the database
    Connection conn = DriverManager.getConnection(
"jdbc:mysql://localhost/tucvp?user=nuance&password=nuance");
    Statement stmt = conn.createStatement();

    String query = "";
    Calendar now = new GregorianCalendar();
    int today= now.get(Calendar.DAY_OF_MONTH)-1;
    if (today==0)today=1;
    int query_day= now.get(Calendar.DAY_OF_MONTH)-1;
    int month = now.get(Calendar.MONTH)+1;
    int query_month = now.get(Calendar.MONTH)+1;
    int year = now.get(Calendar.YEAR);
    int query_year = now.get(Calendar.YEAR);
    java.util.Date last_date = null;
    query_day=today-15;
    query_month=month-1;
    if(query_day <= 0 ){
        query_day=30+query_day;
        if(query_month==0){
            query_year=year-1;
            query_month=12;
        }
    }
    String entry_date=year+"-"+query_month+"-"+query_day;
    try
    {
        SimpleDateFormat sql_date_format = new SimpleDateFormat("yyyy-MM-dd");
        last_date = sql_date_format.parse(entry_date);
    }
    catch (Exception e)
    {
        System.out.println(e.getMessage());
    }
}

```

```

    }

    query = "select header,description,dep_ref,entry_date from announcements"
        + " where (DAY(entry_date) > "+query_day+ " and MONTH(entry_date) = "+month+" ) "
        + " or (DAY(entry_date) > "+query_day+ " and MONTH(entry_date) >= "+query_month+" ) "
        + " or (DAY(entry_date) < "+query_day+ " and MONTH(entry_date) = "+month+" and "
MONTH(entry_date) <= 15)"
        + " order by entry_date DESC";

    ResultSet st = stmt.executeQuery(query);

    while (st.next()) {
        announcement=new Hashtable();
        announcement.put("header_announcements",st.getString("header"));
        announcement.put("description_announcements", st.getString("description"));
        announcement.put("entry_date_announcements", st.getDate("entry_date").toString());
        all_announcements.add(announcement);
    }

    if (stmt != null) {
        stmt.close();
    }

    if (conn != null) {
        conn.close();
    }
}
catch (Exception e) {
    e.printStackTrace();
}
return all_announcements;
}

```

ii. VoiceXML Specification

Πίνακας με όλα τα στοιχεία που αποτελούν την προγραμματιστική γλώσσα VoiceXML και μια σύντομη περιγραφή τους.

Element	Purpose
<assign>	Assign a variable a value
<audio>	Play an audio clip within a prompt
<block>	A container of (non-interactive) executable code
<catch>	Catch an event
<choice>	Define a menu item
<clear>	Clear one or more form item variables
<disconnect>	Disconnect a session
<else>	Used in <if> elements
<elseif>	Used in <if> elements
<enumerate>	Shorthand for enumerating the choices in a menu
<error>	Catch an error event
<exit>	Exit a session
<field>	Declares an input field in a form
<filled>	An action executed when fields are filled
<form>	A dialog for presenting information and collecting data
<goto>	Go to another dialog in the same or different document
<grammar>	Specify a speech recognition or DTMF grammar
<help>	Catch a help event
<if>	Simple conditional logic

Element	Purpose
<initial>	Declares initial logic upon entry into a (mixed initiative) form
<link>	Specify a transition common to all dialogs in the link's scope
<log>	Generate a debug message
<menu>	A dialog for choosing amongst alternative destinations
<meta>	Define a metadata item as a name/value pair
<metadata>	Define metadata information using a metadata schema
<noinput>	Catch a noinput event
<nomatch>	Catch a nomatch event
<object>	Interact with a custom extension
<option>	Specify an option in a <field>
<param>	Parameter in <object> or <subdialog>
<prompt>	Queue speech synthesis and audio output to the user
<property>	Control implementation platform settings.
<record>	Record an audio sample
<reprompt>	Play a field prompt when a field is re-visited after an event
<return>	Return from a subdialog.
<script>	Specify a block of ECMAScript client-side scripting logic
<subdialog>	Invoke another dialog as a subdialog of the

Element	Purpose
	current one
<submit>	Submit values to a document server
<throw>	Throw an event.
<transfer>	Transfer the caller to another destination
<value>	Insert the value of an expression in a prompt
<var>	Declare a variable
<vxml>	Top-level element in each VoiceXML document

iii. Web Based system Specification

Παρακάτω δίνονται όλες οι περιπτώσεις που υλοποιήσαμε και καλύπτουν όλη την λειτουργικότητα που μας είναι αναγκαία για να κρατάμε την βάση της εφαρμογής μας ενήμερη

Εισαγωγή Φοιτητών

Διαγραφή Φοιτητών

Ενημέρωση Στοιχείων Φοιτητών

Εισαγωγή Καθηγητών

Διαγραφή Καθηγητών

Ενημέρωση Στοιχείων Καθηγητών

Εισαγωγή Μαθημάτων

Διαγραφή Μαθημάτων

Ενημέρωση Στοιχείων Μαθημάτων

Εισαγωγή Ανακοινώσεων

Διαγραφή Μαθημάτων

Εισαγωγή Προσώπου στον Τηλεφωνικό Κατάλογο

Διαγραφή Προσώπου από τον Τηλεφωνικό Κατάλογο

Ενημέρωση Στοιχείων Προσώπου στον Τηλεφωνικό Κατάλογο

Εισαγωγή Υπηρεσίας στον Τηλεφωνικό Κατάλογο

Διαγραφή Υπηρεσίας από τον Τηλεφωνικό Κατάλογο

Ενημέρωση Στοιχείων Υπηρεσίας στον Τηλεφωνικό Κατάλογο

Εισαγωγή Γιατρού στην Λίστα Γιατρών

Διαγραφή Γιατρού από την Λίστα Γιατρών

Ενημέρωση Στοιχείων Γιατρού στην Λίστα Γιατρών

Εισαγωγή Ειδικότητας στην Λίστα Ειδικοτήτων Γιατρών

Διαγραφή Ειδικότητας από την Λίστα Ειδικοτήτων Γιατρών

Εισαγωγή στο πρόγραμμα Μαθημάτων

Διαγραφή από το πρόγραμμα Μαθημάτων

Ενημέρωση Στοιχείων καταχώρησης στο πρόγραμμα Μαθημάτων

Εισαγωγή στο πρόγραμμα Εργαστηρίων

Διαγραφή από το πρόγραμμα Εργαστηρίων

Ενημέρωση Στοιχείων καταχώρησης στο πρόγραμμα Εργαστηρίων

Εισαγωγή στο πρόγραμμα Εξεταστικής

Διαγραφή από το πρόγραμμα Εξεταστικής

Ενημέρωση Στοιχείων στο πρόγραμμα Εξεταστικής

Ενημέρωση βαθμολογιών Μαθημάτων

Ενημέρωση βαθμολογιών Εργαστηρίων

Δήλωση Μαθημάτων και Εργαστηρίων

Διαγραφή δηλωμένων Μαθημάτων και Εργαστηρίων

Παρουσίαση προσωποποιημένου προγράμματος Μαθημάτων

Παρουσίαση προσωποποιημένου προγράμματος Εργαστηρίων

Παρουσίαση προσωποποιημένου προγράμματος Εξεταστικής

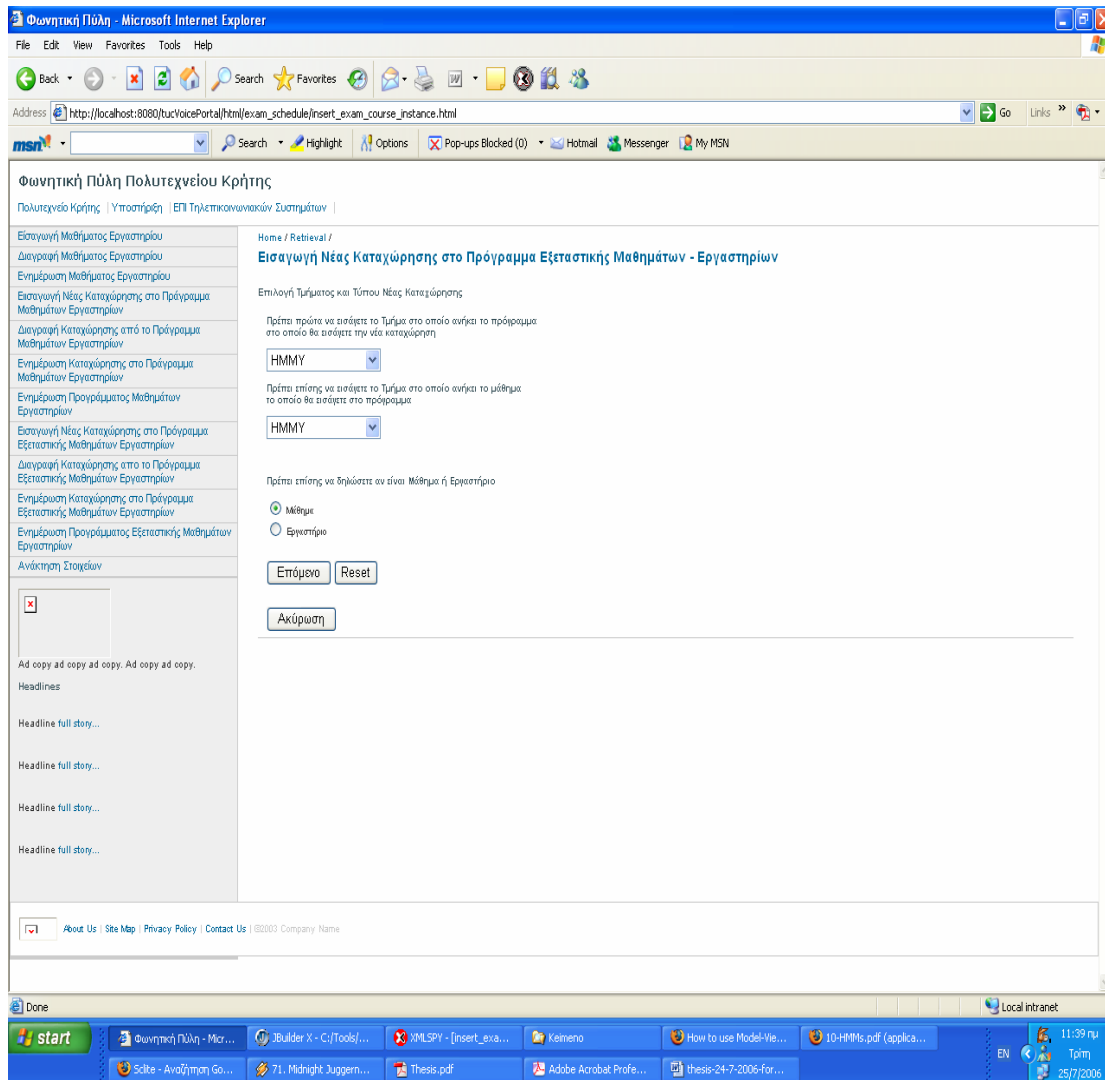
Παρουσίαση προγράμματος Μαθημάτων βάσει εξαμήνου και τμήματος

Παρουσίαση προγράμματος Εργαστηρίων Εξεταστικής βάσει εξαμήνου και τμήματος

Παρουσίαση προγράμματος Εξεταστικής βάσει εξαμήνου και τμήματος

Υλοποίηση Στατικών Σελίδων

Στην σχεδίαση των στατικών σελίδων HTML προσπαθήσαμε να κρατήσουμε την δομή των σελίδων όσο πιο απλή μπορούσαμε καθώς έπρεπε να είναι όσο το δυνατόν πιο ξεκούραστη η πλοήγηση του εκάστοτε χρήστη στο σύστημα. Για αυτό όπως φαίνεται και στην παρακάτω εικόνα που δείχνει μια ενδεικτική σελίδα προτιμήσαμε τις πολύ απλές δομές σε ένα άσπρο φόντο με τους διάφορους πίνακες και γραμμές εισόδου δεδομένων να είναι σε αποχρώσεις του μπλε και του γκρι.



Εικόνα 17 Ενδεικτική HTML σελίδας της Web Based εφαρμογής

Υπάρχουν τρία βασικά μενού τα οποία παρουσιάζονται σε διαφορετικούς εμπλεκόμενους με την εφαρμογή. Κατά αυτόν τον τρόπο υπάρχει ένα μενού για τον κατεξοχήν διαχειριστή του συστήματος και οι δυνατότητες και λειτουργίες που μπορεί να εκτελέσει είναι οι εξής:

1. Διαχείριση μαθημάτων και εργαστηρίων.
2. Διαχείριση φοιτητών και καθηγητών
3. Διαχείριση ανακοινώσεων.
4. Διαχείριση προγράμματος μαθημάτων, εργαστηρίων και εξεταστικής.
5. Διαχείριση της λίστας γιατρών και των ειδικοτήτων.
6. Διαχείριση τηλεφωνικού καταλόγου

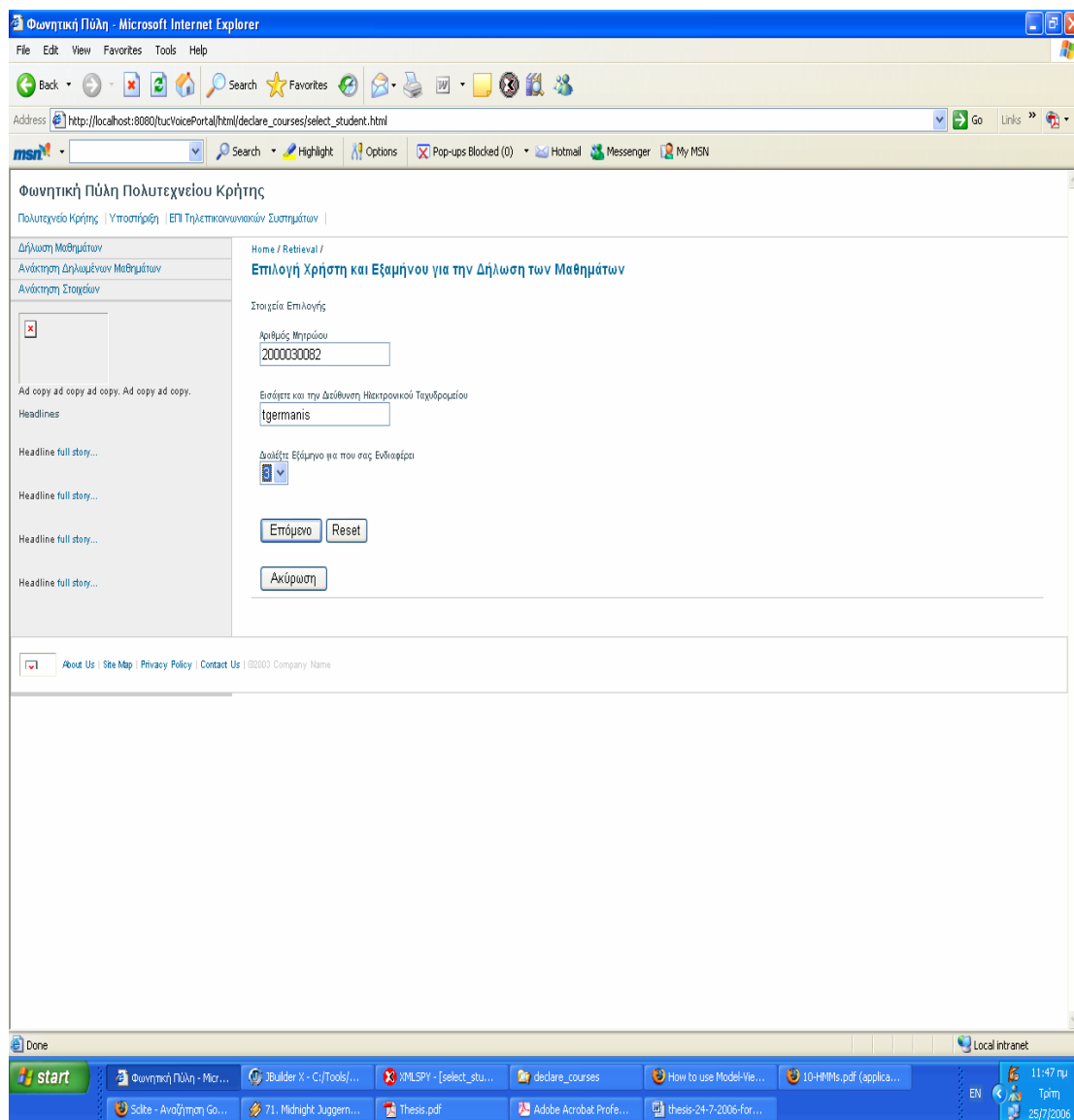
Σε κάθε ένα από αυτά τα αντιστοιχούν υπό-μενού που έχουν να κάνουν με την εισαγωγή, διαγραφή και ενημέρωση στοιχείων στη βάση δεδομένων της εφαρμογής.

Ένα άλλο μενού είναι αυτό στο οποίο έχουν πρόσβαση οι καθηγητές και οι δυνατότητες και λειτουργίες που μπορεί να εκτελέσει είναι οι εξής:

1. Ενημέρωση βαθμολογιών μαθημάτων και εργαστηρίων που διδάσκει.
2. Διαχείριση των απεσταλμένων, από των ίδιο, φωνητικών μηνυμάτων.

Τέλος υπάρχει και ένα μενού το οποίο απευθύνεται στους φοιτητές και στο οποίο θα μπορούν να έχουν πρόσβαση μέσω του διαδικτύου χρησιμοποιώντας κάποιους κωδικούς που θα βοηθούν στην ταυτοποίησή τους. Για της ανάγκες της εφαρμογής χρησιμοποιούμε το ΑΜ των φοιτητών σε συνδυασμό με κάποιο email που μας έχουν δώσει. Οι δυνατότητες και λειτουργίες που μπορεί να εκτελέσει είναι οι εξής:

1. Δήλωση μαθημάτων και εργαστηρίων καθώς και διαγραφή δηλώσεων



Εικόνα 18 Δήλωση στοιχείων φοιτητή

2. Ανάκτηση προγράμματος μαθημάτων, εργαστηρίων και εξεταστικής

Η Δραση Beware - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://localhost:8080/huc/VoicePortal/html/HtmlGetCoursesSchedule.do?dep_id=1&semester=6&ins_del_up=1&lessLab=0&goButton=%CE%95%CF%80%CF%8C%CE%85%CE%8D%CE%BF

msn Search Highlight Options Pop-ups Blocked (0) Hotmail Messenger My MSN

Επιλογή Μεθόδων - Εργαστηρίων

A test table with merged cells

	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή	Σάββατο
09:00:00	ηλεκτρονικά συστήματα 2 αθανάσιος λιόβας M. 4 0 0 2 09:00:00	εισαγωγή στην επεξεργασία φωνής βασιλης δημαλίκης B 1 0 0 5 09:00:00				
10:00:00						
11:00:00				βίσεις δεδομένων σταύρος χριστοδουλίκης B 1 0 0 1 11:00:00		
12:00:00						
13:00:00	κωντ νικόλαος μπαλάνης M. 5 0 0 2 13:00:00	βίσεις δεδομένων σταύρος χριστοδουλίκης B 1 0 0 1 13:00:00				
14:00:00				ψηφιακή επεξεργασία εικόνας μιχάλης ξεροβάκης B 1 0 0 1 14:00:00	ψηφιακή επεξεργασία εικόνας μιχάλης ξεροβάκης B 1 0 0 8 14:00:00	
15:00:00	ηλεκτρονική 2 απόστολος σιμέλης B 1 0 0 8 15:00:00					
16:00:00	κωντ νικόλαος μπαλάνης B 1 0 0 1 16:00:00	ηλεκτρονικά συστήματα 2 αθανάσιος λιόβας B 1 0 0 8 16:00:00			οργάνωση ηλεκτρονικών υπολογιστών ιωάννης παπαευσταθίου M. 4 0 0 2 16:00:00	
17:00:00		γραμματικά συστήματα εμμανουήλ χριστοδουλού A. 2 1. N. O. 17:00:00		οργάνωση ηλεκτρονικών υπολογιστών ιωάννης παπαευσταθίου B 1 0 0 8 17:00:00		
18:00:00					γραμματικά συστήματα εμμανουήλ χριστοδουλού B 1 0 0 1 18:00:00	
19:00:00						

Done

start

verhagen henricus N... html 17. 05 Diana Kral - S... thesis-17-7-2006-for... MSN Messenger Η Δραση Beware - M...

Builder X - C:/Tools/... M.LSPY - [insert_exe...

Local intranet

7:32 μμ Δευτέρα 17/7/2006

Εικόνα 19 Ενδεικτικό προσωποποιημένο πρόγραμμα μαθημάτων φοιτητή

iv. Data Base

Αναλυτική περιγραφή της πληροφορίας που υπάρχει στην βάση δεδομένων:

Ανακοινώσεις: -Όπως για ακαδημαϊκά θέματα – πληροφορίες εκπαίδευσης ή για τα Χανιά καθώς και για τρόπους πρόσβασης στο πολυτεχνείο -Θα πρέπει να υπάρχουν

1. Ημερομηνία εισαγωγής τους

2. Επικεφαλίδα
3. Το κυρίως κείμενο.
4. Τμήμα

Για την αναγγελία-αναζήτηση του προγράμματος μαθημάτων και των εργαστηρίων θα πρέπει αυτά να συνοδεύονται από την εξής πληροφορία:

Μαθήματα :

1. Ονομασία
2. Καθηγητής του μαθήματος.
3. Το τμήμα στο οποίο ανήκει
4. Το έτος – εξάμηνο

Για την καταχώρηση του προγράμματος της εξεταστικής των μαθημάτων χρησιμοποιείται ο παρακάτω πίνακας:

Πρόγραμμα Εξεταστικής Μαθημάτων :

1. Κωδικός Μαθήματος
2. Αίθουσα διεξαγωγής της εξέτασης
3. Ημερομηνία διεξαγωγής της εξέτασης
4. Ώρα διεξαγωγής της εξέτασης
5. Ώρα λήξης της εξέτασης
6. Το τμήμα στο οποίο ανήκει

Παρομοίως για την καταχώρηση του προγράμματος της των μαθημάτων χρησιμοποιείται ο παρακάτω πίνακας:

Πρόγραμμα Μαθημάτων :

7. Κωδικός Μαθήματος
8. Αίθουσα διεξαγωγής του μαθήματος

9. Ημερομηνία διεξαγωγής του μαθήματος
10. Ώρα διεξαγωγής του μαθήματος
11. Ώρα λήξης του μαθήματος
12. Το τμήμα στο οποίο ανήκει

Εργαστήρια :

1. Ονομασία
2. Καθηγητής του εργαστηρίου
3. Το τμήμα στο οποίο ανήκει
4. Το έτος – εξάμηνο

Για την καταχώρηση του προγράμματος της εξεταστικής των εργαστηρίων χρησιμοποιείται ο παρακάτω πίνακας:

Πρόγραμμα Εξεταστικής Εργαστηρίων :

13. Κωδικός Μαθήματος
14. Αίθουσα διεξαγωγής της εξέτασης
15. Ημερομηνία διεξαγωγής της εξέτασης
16. Ώρα διεξαγωγής της εξέτασης
17. Ώρα λήξης της εξέτασης
18. Το τμήμα στο οποίο ανήκει

Παρομοίως για την καταχώρηση του προγράμματος των εργαστηρίων χρησιμοποιείται ο παρακάτω πίνακας:

Πρόγραμμα Εργαστηρίων :

19. Κωδικός Μαθήματος
20. Αίθουσα διεξαγωγής του εργαστηρίου
21. Ημερομηνία διεξαγωγής του εργαστηρίου
22. Ώρα διεξαγωγής του εργαστηρίου

23. Ώρα λήξης του εργαστηρίου

24. Το τμήμα στο οποίο ανήκει

Επίσης στο σύστημα πρέπει να υπάρχουν καταχωρημένοι όλοι οι καθηγητές των τμημάτων, οι οποίοι χαρακτηρίζονται ως εξής :

Καθηγητές :

1. Το όνομα τους
2. Τηλέφωνο
3. E-mail
4. Κωδικός πρόσβασης για την χρήση του συστήματος
5. Το τμήμα στο οποίο ανήκουν

Για τους φοιτητές οι οποίοι θα είναι μαζί με τους καθηγητές οι εν δυνάμει εγγεγραμμένοι χρήστες του συστήματος θα πρέπει να έχουμε αντίστοιχη πληροφορία:

Φοιτητές:

1. Το όνομα τους
2. Κωδικός χρήσης
3. Διεύθυνση e-mail
4. Τηλέφωνο
5. Τμήμα

Για την καταχώρηση των, συνδεδεμένων με τους φοιτητές μαθημάτων και εργαστηρίων, και τους αντίστοιχους βαθμούς χρησιμοποιούμε τους παρακάτω πίνακες:

Φοιτητές - Μαθήματα:

1. Κωδικός μαθήματος

2. Κωδικός φοιτητή
3. Βαθμός μαθήματος
4. Ένδειξη αναγγελίας βαθμού
5. Ημερομηνία δήλωσης μαθήματος
6. Τμήμα

Φοιτητές - Εργαστήρια:

1. Κωδικός Εργαστηρίου
2. Κωδικός φοιτητή
3. Βαθμός Εργαστηρίου
4. Ένδειξη αναγγελίας βαθμού
5. Ημερομηνία δήλωσης Εργαστηρίου
6. Τμήμα

Για τις καταχωρήσεις των φωνητικών μηνυμάτων που οι καθηγητές απευθύνουν στους φοιτητές χρησιμοποιούνται οι παρακάτω πίνακες:

Φωνητικά μηνύματα:

1. Κωδικός μηνύματος
2. Κωδικός σχετιζόμενου μαθήματος- εργαστηρίου
3. Τμήμα
4. Αναφορά στον τύπο (αναφέρεται σε μάθημα ή εργαστήριο)
5. Ημερομηνία εισαγωγής
6. Ηχογραφημένο μήνυμα

Φοιτητές – Φωνητικά Μηνύματα:

1. Κωδικός Φωνητικού Μηνύματος

2. Κωδικός φοιτητή
3. Ένδειξη αναγγελίας μηνύματος.

Τα πρόσωπα που φυλάσσονται στη βάση για την υπηρεσία τηλεφωνικού καταλόγου χαρακτηρίζονται ως εξής:

Πρόσωπα ΤΚ:

1. Το όνομα τους
2. Τμήμα στο οποίο απασχολούνται
3. Τηλέφωνο
4. Χαρακτηρισμός (όπως, καθηγητής ή βοηθός εργαστηρίου)

Επίσης οι υπηρεσίες χαρακτηρίζονται αντίστοιχα ως εξής :

Υπηρεσίες ΤΚ:

1. Ονομασία τους
2. Τηλέφωνο
3. Τμήμα στο οποίο απασχολούνται

Θα πρέπει να υπάρχει ένας κατάλογος όλων των γιατρών που έχουν σύμβαση με το πολυτεχνείο

Γιατροί:

1. Το όνομα τους
2. Ειδικότητα
3. Το τηλέφωνο
4. Διεύθυνση του ιατρείου
5. Ωρες λειτουργίας

Ειδικότητες Γιατρών:

1. Κωδικός
2. Όνομα Ειδικότητας

Βιβλιογραφία

- [1] Conversational Interfaces: Advances And Challenges (1997) (28 citations)Victor Zue Proc. Eurospeech '97.
- [2] Dream Tech Software India Inc, “VoiceXML 2.0 Developer's Guide”.
- [3] <http://www.w3.org/TR/voicexml20/> “Voice Extensible Markup Language (VoiceXML) Version 2.0”.
- [4] “[VoiceXML Introduction to Developing Speech Applications](#)”, James A. Larson Prentice Hall (June 2002).
- [5] Adam Hocek, David Cuddihy: “Definitive VoiceXML”, Prentice Hall, PTR, 2003
- [6] Michael H. Cohen, James P. Giangola, Jennifer Balogh “Voice User Interface Design”. Addison-Wesley, 2004.
- [7] <http://www.auditmype.com/acronym/ARPA.asp>
- [8] Βασίλης Διγαλάκης, Σημειώσεις του μαθήματος: Εισαγωγή στην Επεξεργασία Φωνής
- [9] Christos C. Vosnidis “Robust Speech Synthesis”, Technical University of Crete, June 2004
- [10] Sharma, Chetan /Kunins, Jeff “Voicexml : Strategies and Techniques for Effective Voice Application Development with Voicexml 2.0 (Professional Developer's Guide Series)” Publisher:John Wiley & Sons Inc Published 2001
- [11] <http://www.w3.org/MarkUp>
- [12] <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html> “How to use Model-View-Controller (MVC)”

- [13] <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpatterns/html/DesMVC.asp> “Model-View-Controller”.
- [14] http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/app-arch/app-arch2.html
- [15] <http://www.php.net/>
- [16] <http://www.w3.org/CGI/>
- [17] <http://java.sun.com/>
- [18] Dave Mercer “XML: A Beginner's Guide”.
- [19] <http://www.w3.org/XML/>
- [20] <http://xml.coverpages.org/vxml.html>
- [21] <http://www.lucent.com/>
- [22] <http://www.motorola.com/>
- [23] www.w3c.com
- [25] www.ibm.com
- [26] <http://www.w3.org/TR/speech-grammar/> “Speech Recognition Grammar Specification Version 1.0”
- [27] <http://www.mysql.com/>

- [28] [Database Modeling and Design: Logical Design, 4th Edition \(The Morgan Kaufmann Series in Data Management Systems\)](#) by Toby J. Teorey, Sam Lightstone, and Tom Nadeau, Sep 6, 2005)
- [29] <http://www.nist.gov/>
- [30] <http://www.nist.gov/speech/tools/index.htm>
- [31] Nuance recognition system documentation. Grammar - Chapter 8 System Evaluation – Testing Recognition Performance.
- [32] Bryan Basham, Kathy Sierra & Bert Bates, “Head First Servlets & JSP”, O’Reilly, 2004
- [33] <http://tomcat.apache.org/>
- [34] [Learning Perl, Second Edition](#) by Randal L. Schwartz, Tom Christiansen, and Larry Wall, Jul 1997.