



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

ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΗΟΥ-CS-UGP-2010-05

ΑΝΑΠΤΥΞΗ ΚΑΙ ΕΞΟΜΟΙΩΣΗ ΡΟΜΠΟΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

ΕΛΕΓΧΟΜΕΝΟΥ ΑΠΟ ΦΩΝΗ

ΠΑΠΑΚΩΝΣΤΑΝΤΙΝΟΥ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΦΛΩΡΟΣ ΑΝΔΡΕΑΣ



**HELLENIC OPEN UNIVERSITY
SCHOOL OF SCIENCES AND TECHNOLOGY**





Παπακωνσταντίνου Κωνσταντίνος,

‘Ανάπτυξη και εξομοίωση ρομποτικού συστήματος ελεγχόμενου από φωνή’



Πτυχιακή Εργασία *HOU-CS-UGP-2010-05*

**ΑΝΑΠΤΥΞΗ ΚΑΙ ΕΞΟΜΟΙΩΣΗ ΡΟΜΠΟΤΙΚΟΥ
ΣΥΣΤΗΜΑΤΟΣ ΕΛΕΓΧΟΜΕΝΟΥ ΑΠΟ ΦΩΝΗ**

ΠΑΠΑΚΩΝΣΤΑΝΤΙΝΟΥ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ : ΦΛΩΡΟΣ ΑΝΔΡΕΑΣ



© ΕΑΠ, 2010

Η παρούσα διατριβή, η οποία εκπονήθηκε στα πλαίσια της ΘΕ ΠΛΗ40, και τα λοιπά αποτελέσματα της αντίστοιχης Πτυχιακής Εργασίας (ΠΕ) αποτελούν συνιδιοκτησία του ΕΑΠ και του φοιτητή, ο καθένας από τους οποίους έχει το δικαίωμα ανεξάρτητης χρήσης και αναπαραγωγής τους (στο σύνολο ή τμηματικά) για διδακτικούς και ερευνητικούς σκοπούς, σε κάθε περίπτωση αναφέροντας τον τίτλο και το συγγραφέα και το ΕΑΠ όπου εκπονήθηκε η ΠΕ καθώς και τον επιβλέποντα και την επιτροπή κρίσης.



Ανάπτυξη και Εξομοίωση Ρομποτικού Συστήματος Ελεγχόμενου από Φωνή

ΠΑΠΑΚΩΝΣΤΑΝΤΙΝΟΥ ΚΩΝΣΤΑΝΤΙΝΟΣ

Επιβλέπων	Μέλος	Μέλος
Φλώρος Ανδρέας	Καμέας Αχιλλέας	Ζαχαράκης Ιωάννης
Επίκουρος Καθηγητής, Τμήμα Τεχνών Ήχου και Εικόνας, Ιόνιο Πανεπιστήμιο	Επίκουρος Καθηγητής, Σχολή Θετικών Επιστημών & Τεχνολογίας, Ελληνικό Ανοικτό Πανεπιστήμιο	Επίκουρος Καθηγητής, Τμήμα Εφαρμογών Πληροφορικής στη Διοίκηση & Οικονομία ΑΤΕΙ Πάτρας

Περίληψη

Οι τεχνολογικές εξελίξεις στον τομέα της ρομποτικής επιτρέπουν πλέον την υλοποίηση εκπαιδευτικών ρομποτικών συστημάτων σχετικά μικρού κόστους και πολυπλοκότητας, ενώ παράλληλα στην αγορά διατίθενται αξιόλογα προϊόντα που αφορούν στις υπηρεσίες φωνής, αλλά και υψηλού επιπέδου λογισμικό ανάπτυξης ρομποτικών εφαρμογών.

Η παρούσα εργασία έχει σαν στόχο την σχεδίαση ενός εκπαιδευτικού ρομποτικού συστήματος, το οποίο μπορεί να ελέγχεται απομακρυσμένα με τη χρήση ελληνικών και αγγλικών φωνητικών εντολών. Ο έλεγχος αυτός αφορά τόσο στο φυσικό ρομποτικό σύστημα όσο και στην εξομοίωσή του σε εικονικό περιβάλλον.

Λέξεις-κλειδιά: Ρομποτικό σύστημα, ρομπότ, Microsoft Robotics Studio, MRDS, τεχνολογίες φωνής, αναγνώριση φωνής, σύνθεση φωνής, Lego NXT



Development and Simulation of a Voice Controlled Robotic System

PAPAKONSTANTINOU KONSTANTINOS

Επιβλέπων	Kameas	Μέλος
Floros Andreas	Kameas Achilleas	Zaxarakis Ioannis
Assistant Professor, Ionian University	Assistant Professor, Hellenic Open University	Assistant Professor, T.E.I of Patras

Περίληψη

Now days, recent technological advances in the field of robotics allow the implementation of educational robotic systems of low cost and complexity while on the other hand, quality products relevant to voice services are released to the market along with quality software for the development of robotic applications.

The purpose of writing this thesis is the design of an educational robotic system, which can be remotely controlled through the use of Greek and English voice commands. This type of voice control has to do with both the physical robot and also a virtual representation of the robot in a simulation environment.

Λέξεις-κλειδιά: Robotic System, robot, Microsoft Robotics Studio, MRDS, voice technology, voice recognition, speech synthesis, Lego NXT



Παπακωνσταντίνου Κωνσταντίνος,

‘Ανάπτυξη και εξομοίωση ρομποτικού συστήματος ελεγχόμενου από φωνή’



Πίνακας Περιεχομένων

1. Εισαγωγή	11
2. Συστήματα επεξεργασίας φωνής	12
2.1 Ο μηχανισμός παραγωγής της ανθρώπινης ομιλίας	12
2.2 Βασικές έννοιες	13
2.3 Αναγνώριση φωνής (Speech Recognition)	15
2.3.1 Δομή συστημάτων SR	15
2.3.2 Η σημασία της γλώσσας	16
2.4 Σύνθεση φωνής από κείμενο (Text to Speech Synthesis)	18
2.4.1 Πρώτες προσπάθειες παραγωγής τεχνητής ομιλίας	18
2.4.2 Δομή συστημάτων TTS	19
2.5 Κατανόηση προφορικού λόγου (Spoken Language Understanding)	20
2.5.1 Χρήση συστημάτων SLU	20
2.5.2 Δομή συστημάτων SLU	21
2.6 Προϊόντα λογισμικού στον τομέα επεξεργασίας φωνής	22
2.6 Επεξεργασία φωνής σε περιβάλλον Windows	23
2.6.1 Speech Recognition API (SAPI)	24
3. Ρομποτικά συστήματα	28
3.1 Γενικά για τα ρομπότ	28
3.1.1 Ο ρόλος της τέχνης	28
3.1.2 Πεδία εφαρμογών της ρομποτικής	28
3.1.3 Ο ορισμός του ρομπότ	29
3.1.4 Σύγχρονα ευφυή ρομπότ : το ευφρές όχημα	30
3.2 Το εκπαιδευτικό ρομπότ Lego MINDSTORMS NXT	32
3.2.1 Σύντομο ιστορικό	32
3.2.2 Lego NXT brick	33
3.2.3 Επενεργητές (actuators) και αισθητήρες (sensors)	37
4. Εφαρμοσμένο λογισμικό ανάπτυξης ρομποτικών εφαρμογών	41
4.1 Διαθέσιμα προϊόντα ρομποτικού λογισμικού	41
4.2 Microsoft Robotics Developer Studio	42
4.2.1 Σύντομο ιστορικό	42
4.2.2 Επισκόπηση του Microsoft RDS	43
4.2.3 Χαρακτηριστικά πλεονεκτήματα	44
4.2.4 Concurrency and Coordination Runtime (CCR)	47



4.2.5 Decentralized Software Services (DSS)	49
4.2.6 Visual Programming Language (VPL)	51
5. Ανάπτυξη λογισμικού, κατασκευή και εξομοίωση συστήματος	53
5.1 Η εύκαμπτη μεθοδολογία ανάπτυξης	53
5.1.1 Το πλαίσιο μεθοδολογιών MSF	53
5.1.2 Επιλογή πλαισίου μεθοδολογίας	54
5.2 Βασικές υποδομές και συνιστώσες λογισμικού ανάπτυξης	57
5.2.1 Το .NET framework	57
5.2.2 Η γλώσσα C#	57
5.2.3 Το περιβάλλον ανάπτυξης Visual Studio	58
5.2.4 Η μηχανή αναγνώρισης Julius	59
5.2.5 Η συνιστώσα Conjura	60
5.2.6 Simon : ανοικτό λογισμικό αναγνώρισης φωνής	60
5.2 Απαιτήσεις συστήματος	62
5.2.1 Λειτουργικές απαιτήσεις	62
5.2.2 Μη λειτουργικές απαιτήσεις	64
5.3 Αρχιτεκτονική αποτύπωση συστήματος	66
5.3.1 Λειτουργική απεικόνιση (functional view)	66
5.3.2 Λογική απεικόνιση (logical view)	77
5.3.3 Απεικόνιση της λειτουργικής υποστήριξης (maintenance view)	84
5.3.4 Φυσική απεικόνιση (physical view)	91
6. Αποτελέσματα	99
6.1 Παρουσίαση εφαρμογής	99
6.2 Δοκιμές αναγνώρισης φωνητικών εντολών	101
6.2.1 Σενάριο 1 : Περιβάλλον ήσυχου δωματίου	101
6.2.2 Σενάριο 2 : Τυπικό περιβάλλον δωματίου	102
6.2.2 Σενάριο 3 : Εκφορά μη-προδιεγραμμένων λέξεων	103
6.2.3 Συνοπτική ανάλυση αποτελεσμάτων	105
7. Συμπεράσματα, περιορισμοί και επεκτάσεις	106
7.1 Ανασκόπηση	106
7.2 Επίτευξη στόχων	107
7.3 Περιορισμοί	107
7.4 Πιθανές επεκτάσεις	107
7.5 Προοπτικές τεχνολογικών πεδίων	108



Βιβλιογραφικές αναφορές	109
Βιβλία	109
Διαδίκτυο	110
Παραρτήματα	114
Παράρτημα Α: Ακρωνύμια / Συντομογραφίες	114
Παράρτημα Β: Ευρετήριο Σχημάτων	115
Παράρτημα Γ: Ευρετήριο Εικόνων	115
Παράρτημα Δ: Ευρετήριο Πινάκων	116



1. Εισαγωγή

Ιδιαίτερο ενδιαφέρον της αυτοματοποίησης διαδικασιών που αφορούν στην ανθρώπινη ομιλία παρουσιάζει ο τομέας της επικοινωνίας μεταξύ ανθρώπου και μηχανής. Σε αυτόν τον τομέα διακρίνουμε δύο βασικές υποκατηγορίες :

- την αναγνώριση της ανθρώπινης ομιλίας από μία μηχανή και την μετατροπή της σε κείμενο
- την μετατροπή κειμένου σε συνθετική ομιλία

Από την άλλη πλευρά, στον τομέα της ρομποτικής ένα από τα εύκολα διακρινόμενα αποτελέσματα της αλματώδους εξέλιξης, είναι η διάθεση στην αγορά εξελιγμένων εκπαιδευτικών ρομποτικών συστημάτων χαμηλού κόστους και η παράλληλη διάθεση λογισμικού που αφορά στην πλήρη διαχείρισή τους αλλά και στην ανάπτυξη ρομποτικών εφαρμογών γενικότερα.

Στόχος της παρούσας εργασίας είναι ο δημιουργικός συνδυασμός των τεχνολογιών φωνής και εκπαιδευτικής ρομποτικής, με τελικό εφαρμοσμένο αποτέλεσμα τον απομακρυσμένο έλεγχο ενός ρομποτικού συστήματος μέσω φωνητικών εντολών που εκφέρονται στην ελληνική ή στην αγγλική γλώσσα.

Η προσέγγιση αυτή πραγματοποιείται μέσα από σύγχρονες υποδομές ρομποτικού λογισμικού και πλεονεκτήματά της αποτελούν τόσο η απλότητα της διεπαφής όσο και η ευκολία χειρισμού από μη ειδικούς χρήστες, καθώς η ομιλία αποτελεί τον πλέον φυσικό τρόπο επικοινωνίας.

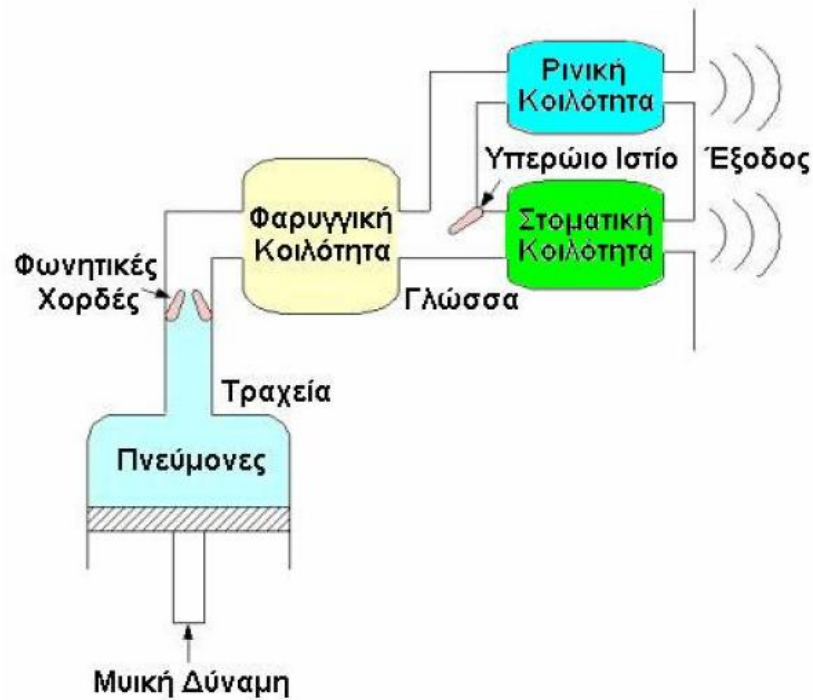


2. Συστήματα επεξεργασίας φωνής

Η επεξεργασία φωνής αναφέρεται σε τεχνολογίες που σχετίζονται με τους τομείς της αναγνώρισης φωνής (*Speech Recognition*), της σύνθεσης φωνής από κείμενο (*Text to Speech*) και της κατανόησης του προφορικού λόγου (*Spoken Language Understanding*). Στην ενότητα αυτή, καταρχήν περιγράφεται συνοπτικά ο μηχανισμός παραγωγής της ανθρώπινης ομιλίας. Έπειτα δίνονται κάποιοι βασικοί ορισμοί που έχουν συνεκτικό ρόλο μεταξύ των τεχνολογιών και στην συνέχεια περιγράφονται οι αρχές των συστημάτων αναγνώρισης και σύνθεσης φωνής. Ακολούθως παρατίθενται κάποια χαρακτηριστικά προϊόντα της τεχνολογίας φωνής και γίνεται εκτενέστερη αναφορά στην προγραμματιστική υποδομή SAPI (*Speech Recognition API*) που παρέχει το λειτουργικό σύστημα Windows καθώς είναι αυτή που χρησιμοποιείται στην ανάπτυξη της τελικής εφαρμογής.

2.1 Ο μηχανισμός παραγωγής της ανθρώπινης ομιλίας

Η λάρυγγική κοιλότητα περιλαμβάνει δύο πτυχές με ρυθμιζόμενη τάση που αποτελούν τις φωνητικές χορδές ενώ το μεταξύ τους μεταβαλλόμενο άνοιγμα αποτελεί τη γλωττίδα. Κατά την διαδικασία παραγωγής ομιλίας οι μύες των πνευμόνων δημιουργούν την απαιτούμενη πίεση αέρα, ώστε να δημιουργηθεί μια ροή αέρα προς την γλωττίδα [b3]. Η ροή αυτή του αέρα διέρχεται διαμέσου της γλωττίδας στο λοιπό φωνητικό κανάλι που είναι ο λάρυγγας. Εκεί διαμορφώνεται από τα χαρακτηριστικά του και εισάγεται στην στοματική κοιλότητα (ή και την ρινική, σε περίπτωση έρρινης ομιλίας), από όπου και παράγεται η ομιλία διαμέσου του στόματος [b3]. Στο Σχήμα 1 παρακάτω, παρουσιάζεται ο μηχανισμός παραγωγής της ανθρώπινης ομιλίας :



Σχήμα 1: Μηχανισμός παραγωγής ανθρώπινης ομιλίας [b3]

2.2 Βασικές έννοιες

Στις τεχνολογίες φωνής είναι συχνή η αναφορά σε κάποιες βασικές έννοιες όπως είναι τα *φωνήματα*, οι *φθόγγοι*, τα *εκφωνήματα* και η *προσωδία* και οι οποίες αποτελούν κλειδιά στην συνολική κατανόηση των συστημάτων επεξεργασίας φωνής. Οι έννοιες αυτές ορίζονται και εξηγούνται στη συνέχεια.

- **Φωνήματα (phonemes)**

Τα *φωνήματα* αποτελούν αφαιρετικές κατασκευές του κλάδου της γλωσσολογίας που καλείται *φωνολογία* και χρησιμοποιούνται για την συστηματοποίηση της μελέτης της γλώσσας και την περιγραφή φαινομένων που συνδέονται με την προφορά των λέξεων [i13]. Ο όρος δηλώνει τα ελάχιστα τμήματα ομιλίας που χρησιμοποιούνται για τον διαχωρισμό της μίας λέξης από μία άλλη [b6].

- **Φθόγγοι (phones)**

Οι *φθόγγοι* είναι οι στοιχειώδεις ήχοι από τους οποίους αποτελείται η ομιλία. Κάθε γλώσσα έχει ένα σύνολο φθόγγων από τους οποίους παρέχει τις λέξεις της. Τα χαρακτηριστικά των φθόγγων έχουν να κάνουν με την εκφορά τους η οποία περιλαμβάνει τη φωνή, την άρθρωση και την ακουστική τους σύσταση δηλαδή το είδος και τις ιδιότητες του ίδιου του ήχου. Οι



φθόγγοι χωρίζονται στα γνωστά φωνήεντα και σύμφωνα, αλλά υπάρχει και ενδιάμεση κατηγορία φθόγγων που καλούνται ημίφωνα [i13].

- **Σχέση φωνημάτων και φθόγγων**

Οι φθόγγοι αποτελούν την ακουστική πραγματοποίηση των φωνημάτων: έτσι π.χ. το αγγλικό φώνημα /t/ έχει δύο πολύ διαφορετικές ακουστικές πραγματοποιήσεις στις λέξεις ‘sat’ και ‘meter’, δηλαδή το ίδιο φώνημα μπορεί να αντιστοιχεί σε δύο διαφορετικούς φθόγγους [b6]. Το ίδιο μπορεί να συμβαίνει και για το σύνολο των φωνημάτων μίας γλώσσας που μπορεί να έχουν διαφορετικές ακουστικές πραγματοποιήσεις όταν εκφέρονται από διαφορετικούς ομιλητές – στην περίπτωση της Ελληνικής γλώσσας αυτό συμβαίνει όταν ακούμε έναν ομιλητή π.χ. από κάποια περιοχή της Ηπείρου και άλλον ομιλητή από κάποια περιοχή της Κρήτης να εκφέρουν τις ίδιες λέξεις.

Παρόλα αυτά, τα φωνήματα παραμένουν ένας συστηματικός τρόπος διαχωρισμού του νοήματος των λέξεων. Έτσι το αγγλικό φώνημα /r/ σηματοδοτεί την λέξη ‘rat’ σε αντίθεση με τη λέξη ‘bat’ που μεν ακούγεται παρόμοια αλλά διαχωρίζεται σαφώς λόγω του φωνήματος /b/.

- **Εκφωνήματα (utterances)**

Τα εκφωνήματα αποτελούν μονάδες γλωσσικής ανάλυσης της σημασίας του τομέα της πραγματολογίας. Αν και ο ορισμός ποικίλει, κοινός τόπος είναι ότι το εκφώνημα αποτελεί ένα κομμάτι γλώσσας, πριν και μετά το οποίο υπάρχει παύση και αντιστοιχεί στον λόγο που παράγει ένας συγκεκριμένος ομιλητής σε μία συγκεκριμένη περίπτωση. Το μήκος του εκφωνήματος μπορεί να είναι μεγαλύτερο, ίσο ή μικρότερο από μία πρόταση, δηλαδή μπορεί να είναι ακόμη και μία φράση ή μία λέξη [i25].

- **Προσωδία**

Η προσωδία αποτελεί μια πολύπλοκη σύνθεση τόσο φυσικών (άρθρωση, μηχανισμός ομιλίας) όσο και φωνητικών διεργασιών τα οποία υιοθετούνται από τον ομιλητή προκειμένου να εκφράσει την διάθεση του, να εκφέρει τις υποθέσεις και την προσοχή του. Αποτελεί ουσιαστικά ένα παράλληλο κανάλι πληροφορίας της καθημερινής λεκτικής επικοινωνίας του ομιλητή [b7].

Η προσωδία έχει σημαντικό ρόλο στη διαμόρφωση του μηνύματος που εκφέρεται, καθώς αποδίδει την εννοιολογική απόχρωση (connotation) του μηνύματος και μπορεί να εκφράσει τη συναισθηματική διάθεση του ομιλητή όπως έκπληξη, θυμό κ.τ.λ.



Από την πλευρά του ακροατή η προσωδία διακρίνεται από τα παρακάτω ερεθίσματα [b7]:

- τις *παύσεις (pauses)*
- την *ακουστότητα (loudness)* δηλαδή το υποκειμενικό μέγεθος που δείχνει το πόσο δυνατά αντιλαμβάνεται ένας άνθρωπος έναν ήχο συγκεκριμένης ηχητικής στάθμης (dB-SPL).
- την *σχετική διάρκεια (rate duration)* δηλαδή την διάρκεια των φωνημάτων αλλά αφορά και στον συγχρονισμό και στον ρυθμό τους
- την *τονικότητα (pitch)* που είναι η αντίληψη της *θεμελιώδους συχνότητας* από τον ακροατή. Η *θεμελιώδης συχνότητα* προκύπτει από τον βασικό ρυθμό ταλάντωσης των φωνητικών χορδών.
- την *χροιά* η οποία ορίζεται ως το πλήθος των αρμονικών και το σχετικό τους πλάτος ως προς την θεμελιώδη.

2.3 Αναγνώριση φωνής (Speech Recognition)

Η αναγνώριση φωνής (Speech Recognition - SR) είναι η διαδικασία της μετατροπής ενός ακουστικού σήματος, το οποίο λαμβάνεται από κάποια συσκευή, σε ένα σύνολο από λέξεις[b4].

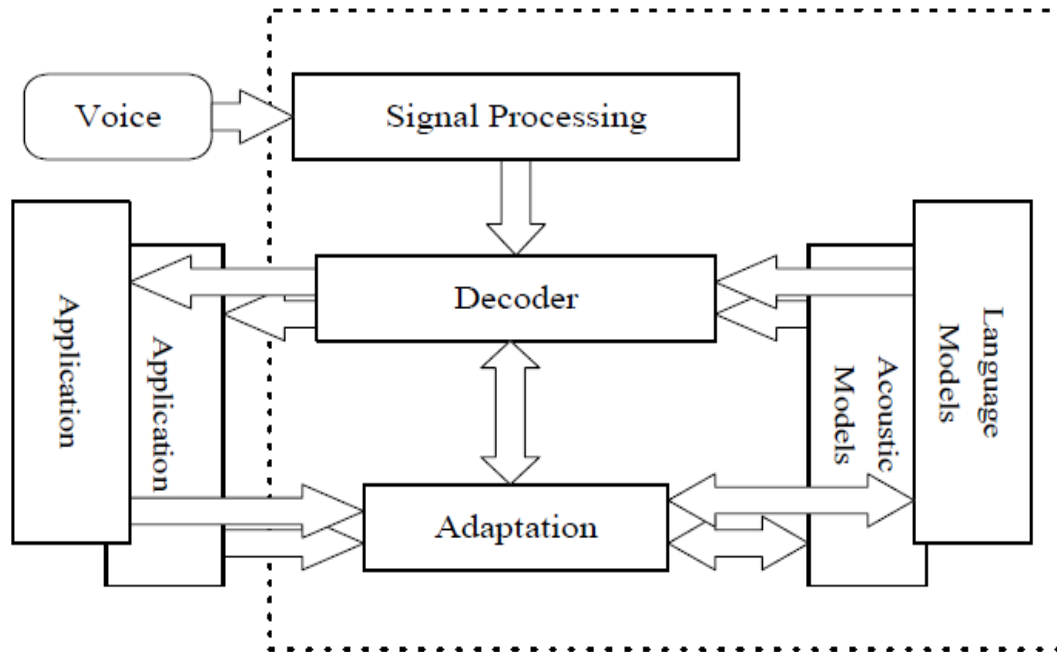
2.3.1 Δομή συστημάτων SR

Σε ένα SR σύστημα τις βασικότερες συνιστώσες (components) αποτελούν ο *αποκωδικοποιητής (decoder)*, το *ακουστικό μοντέλο* και το *γλωσσικό μοντέλο (language model)*. Οι εφαρμογές επικοινωνούν με τον *αποκωδικοποιητή (decoder)* ώστε να ληφθούν τα τελικά αποτελέσματα της αναγνώρισης. Το *ακουστικό μοντέλο (acoustic model)* περιλαμβάνει όλη την αναπαράσταση της γνώσης για την φωνητική και την ακουστική του συστήματος που αφορά στις συσκευές εισόδου (μικρόφωνο), στις διαφορές φύλλου και διαλέκτου μεταξύ των ομιλητών κ.τ.λ. Το *γλωσσικό μοντέλο* αναφέρεται στη γνώση που έχει το σύστημα στο τι αποτελεί μία λέξη, ποιες λέξεις είναι πιθανό να συνυπάρχουν και σε τι συχνότητα [b6].

Όπως φαίνεται και παρακάτω στο Σχήμα 2, η επεξεργασία ξεκινάει όταν το σήμα εισόδου της ομιλίας εισέρχεται στην συνιστώσα της *επεξεργασίας σήματος (signal processing)*, η οποία τροφοδοτεί με διανύσματα χαρακτηριστικών (feature vectors) τον *αποκωδικοποιητή*. Ο *αποκωδικοποιητής* χρησιμοποιεί τόσο το *ακουστικό* και το *γλωσσικό μοντέλο* ώστε να παράγει τις λέξεις εκείνες που έχουν τη μεγαλύτερη πιθανότητα με βάση την είσοδο των



προαναφερόμενων διανυσμάτων. Μπορεί επίσης να παρέχει δεδομένα στην συνιστώσα της προσαρμογής ώστε να τροποποιήσει το ακουστικό ή το γλωσσικό μοντέλο [b6].



Σχήμα 2: Βασική δομή ενός συστήματος SR [b6]

2.3.2 Η σημασία της γλώσσας

Η σχετική με την γλώσσα γνώση είναι σημαντική στην κατανόηση του προφορικού λόγου. Η γνώση αυτή περιλαμβάνει καταρχήν την λεξικολογική γνώση που έχει να κάνει με σημαντικά στοιχεία όπως ο ορισμός του λεξικού (dictionary) για όλες τις λέξεις που χρησιμοποιούνται και της αντίστοιχης προφοράς τους. Περιλαμβάνει επίσης γνώση για το συντακτικό και την σημασιολογία της γλώσσας που σχετίζονται με την γραμματική [b6].

- **Το λεξικό προφοράς (pronunciation dictionary)**

Το λεξικό προφοράς περιέχει σε αλφαβητική ταξινόμηση τις λέξεις που θα χρησιμοποιηθούν με τα αντίστοιχα φωνήματα που χαρακτηρίζουν και την προφορά τους. Για την αναπαράσταση των φωνημάτων υπάρχουν διάφορες τυποποιήσεις όπως π.χ. μπορούμε να δούμε αμέσως πιο κάτω στον Πίνακα 1 δύο απλουστευτικά παραδείγματα με βάση την τυποποίηση IPA :



Πίνακας 1: Παράδειγμα τυποποίησης κατά IPA

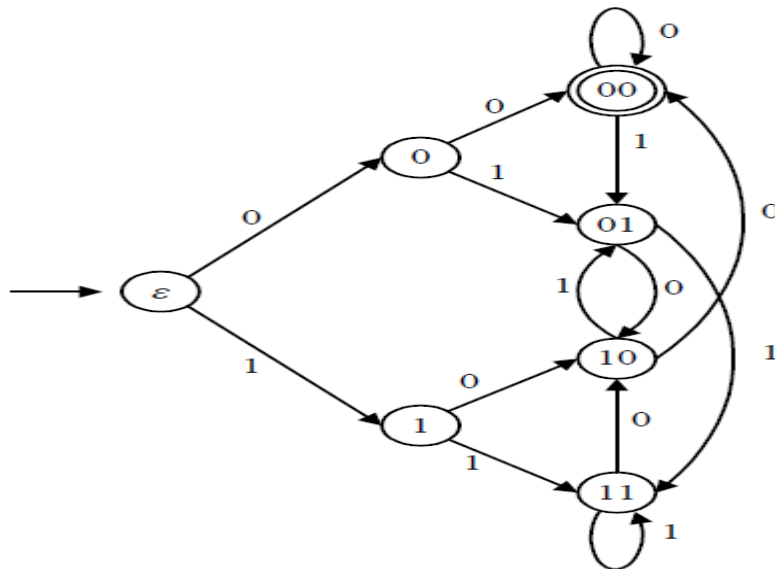
Λέξη	Φωνήματα και προφορά κατά IPA
ΣΤΡΙΨΕ	stri p se
ΑΡΙΣΤΕΡΑ	ari ste ra

- **Το διεθνές φωνητικό αλφάβητο IPA (International Phonetic Alphabet)**

Ο IPA (International Phonetic Association) αποτελεί τον παλαιότερο οργανισμό με αντικείμενο την φωνητική και ιδρύθηκε το 1886. Εκτός των άλλων του δραστηριοτήτων σκοπός του είναι να παρέχει στην παγκόσμια ακαδημαϊκή κοινότητα ένα πρότυπο για την φωνητική αναπαράσταση όλων των γλωσσών. Το πρότυπο αυτό είναι το διεθνές φωνητικό αλφάβητο που καλείται επίσης IPA (International Phonetic Alphabet) και η τελευταία του έκδοση πραγματοποιήθηκε το 2005 [i27].

- **Πεπερασμένα αυτόματα**

Στην δεκαετία του '50 οι McCulloch και Pitt εισήγαγαν την έννοια μίας αφηρημένης μηχανής πεπερασμένων καταστάσεων για την μοντελοποίηση των νευρώνων του εγκεφάλου η οποία έχει την εξής λειτουργία: κάθε φορά βρίσκεται σε μία από ένα σύνολο καταστάσεων, αλλάζοντας κατάσταση ανάλογα με την είσοδο που δέχεται το σύστημα [b14]. Η μηχανές αυτές ονομάζονται *πεπερασμένα αυτόματα*.



Σχήμα 3: Παράδειγμα αυτόματου που αναγνωρίζει μία γλώσσα



- **Ο ρόλος της γραμματικής**

Στο γλωσσικό μοντέλο (*language model*) δύο στοιχεία είναι θεμελιώδη : η *γραμματική* (*grammar*) και ο *αλγόριθμος ανάλυσης* (*parsing algorithm*) [b6]. Μία *γραμματική* αποτελείται από ένα σύνολο κανόνων οι οποίοι μοντελοποιούν τις ιδιότητες μίας γλώσσας [b14]. Ο *αλγόριθμος ανάλυσης* αφορά στην τεχνική ανάλυσης μίας πρότασης έτσι ώστε να φανεί εάν η δομή της είναι συμβατή με την *γραμματική* [b6].

Έχει αποδειχτεί ότι οι *γραμματικές ανεξάρτητες συμφραζομένων* (*CFG grammars*) είναι ισοδύναμες με ένα είδος αυτομάτου, το αυτόματο στοιβας και αυτό σημαίνει ότι για κάθε *γραμματική* υπάρχει ένα *αυτόματο στοιβας* που μπορεί να αναγνωρίσει όλες τις λέξεις που παράγονται από αυτή και αντίστροφα για κάθε *αυτόματο* υπάρχει μία *γραμματική* που παράγει τη γλώσσα που αναγνωρίζεται από το αυτόματο [b14]. Έτσι οι γλώσσες που παράγονται από γραμματικές μπορούν να αναγνωριστούν και από κάποιο αυτόματο στοιβας. Αυτός είναι και ένας από τους βασικούς λόγους που τα αυτόματα χρησιμοποιούνται στην προαναφερόμενη ανάλυση προτάσεων αλλά και σε πολλές άλλες εφαρμογές, όπως στη φάση της συντακτικής ανάλυσης που πραγματοποιούν οι μεταγλωττιστές, στον εντοπισμό λέξεων σε κείμενο κτλ. Οι *γραμματικές* χρησιμοποιούνται επίσης για τον περιορισμό των πιθανών συνδυασμών των λέξεων που μπορούν να παραχθούν [b4].

2.4 Σύνθεση φωνής από κείμενο (Text to Speech Synthesis)

Η σύνθεση φωνής από κείμενο (Text to Speech - TTS) είναι η διαδικασία παραγωγής προφορικού λόγου από μία μηχανή η οποία δέχεται ως είσοδο κάποιο κείμενο, με σκοπό την μίμηση ανθρώπινης ομιλίας [b6].

2.4.1 Πρώτες προσπάθειες παραγωγής τεχνητής ομιλίας

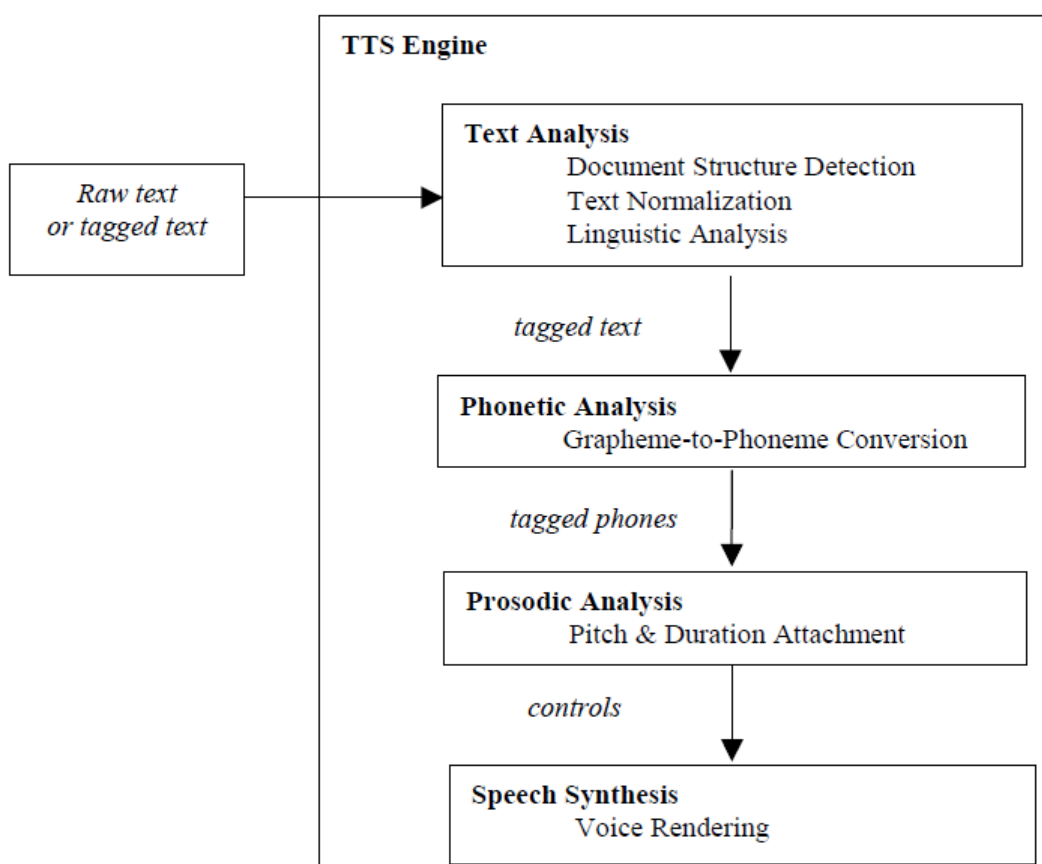
Οι πρώτες προσπάθειες για παραγωγή τεχνητής ομιλίας ξεκινούν πάνω από δύο αιώνες πριν όταν το 1779 ο Ρώσος καθηγητής Christian Kratzenstein επεξήγησε τις διαφορές μεταξύ των πέντε φωνηέντων και επιπλέον έφτιαξε μία συσκευή από ακουστικούς ταλαντωτές τους οποίους ενεργοποιούσε με αέρα, για να παράγει τα φωνήεντα αυτά [b7]. Το 1791 στη Βιέννη ο Wolfgang von Kempelen, ο οποίος ξεκίνησε την εργασία του πριν τον Kratzenstein, εισάγει την ακουστικο-μηχανική μηχανή φωνής που μπορεί να παράγει απλούς ήχους και κάποιους φωνητικούς συνδυασμούς [b7]. Στην σύγχρονη εποχή, ο πρώτος ηλεκτρικός συνθέτης δημιουργήθηκε από τον Stewart το 1922 ενώ η πρώτη συσκευή που θεωρείται ως συνθέτης φωνής ήταν ο VODER από τον Homer Dudley το 1939. Ο πρώτος συνθέτης συνάρθρωσης (*articulator synthesizer*) παρουσιάζεται το 1958 από τον George Rosen στο



MIT, ενώ το 1979 οι Allen, Hunnicut και Klatt παρουσιάζουν το MITalk ένα εργαστηριακό σύστημα σύνθεσης φωνής που αναπτύχθηκε στο MIT. Οι εξελίξεις από εκεί και πέρα είναι ραγδαίες.

2.4.2 Δομή συστημάτων TTS

Σε ένα σύστημα TTS τις βασικότερες συνιστώσες (components) αποτελούν αυτές που πραγματοποιούν *ανάλυση κειμένου (text analysis)*, *φωνητική ανάλυση (phonetic analysis)* ή αλλιώς *φωνητική μεταγραφή, προσωδιακή ανάλυση (prosodic analysis)* και τέλος *σύνθεση φωνής (speech synthesis)*. Αρχικά εισάγεται στο σύστημα κείμενο το οποίο μπορεί να περιέχει ετικέτες (tags) για την υποστήριξης των επόμενων διαδικασιών ανάλυσης. Η *ανάλυση κειμένου* κανονικοποιεί (normalizes) το κείμενο σε κατάλληλη μορφή ώστε να μπορεί να ομιληθεί από την μηχανή. Η *φωνητική ανάλυση* μετατρέπει το επεξεργασμένο κείμενο στην αντίστοιχη φωνητική ακολουθία. Έπειτα, η *προσωδιακή ανάλυση* προσθέτει στη φωνητική ακολουθία κατάλληλη πληροφορία *τονικότητας (pitch)* και *διάρκειας (duration)* και τέλος η *σύνθεση φωνής* αξιοποιεί τις παραμέτρους από τη φωνητική ακολουθία για να παράγει την τελική κυματομορφή ομιλίας (speech waveform) [b6].



Σχήμα 4: Βασική δομή ενός συστήματος TTS [b6]



2.5 Κατανόηση προφορικού λόγου (Spoken Language Understanding)

Η έννοια της μηχανικής κατανόησης μπορεί να δοθεί τόσο από λειτουργική όσο και από στοχαστική σκοπιά. Με βάση την λειτουργική οπτική, η κατανόηση επιτυγχάνεται όταν η μηχανή (π.χ. Η/Υ) με την οποία αλληλεπιδρούμε καταλαβαίνει τα αιτήματα και τις επιθυμίες μας και παράγει ένα επιθυμητό αποτέλεσμα. Με βάση την στοχαστική οπτική, για να μπορεί να κατανοήσει η μηχανή πρέπει να έχει αποκτήσει μία ακριβή και μονοσήμαντη άποψη που να αντιστοιχεί σε μία πραγματική κατάσταση [b6]. Πρακτικά η έννοια της κατανόησης είναι εξαρτώμενη από μία δεδομένη κατάσταση.

2.5.1 Χρήση συστημάτων SLU

Τα συστήματα αναγνώρισης και σύνθεσης φωνής παρόλη την λειτουργική τους αυτοτέλεια αποτελούν μόνο μέρος της συνολικής λύσης όταν μας ενδιαφέρει να συνάγουμε σημασιολογικό νόημα. Έτσι ένας αποκωδικοποιητής (*decoder*) μπορεί να μετατρέπει την ομιλία σε κείμενο με ορθό τρόπο και η σύνθεση φωνής από κείμενο επίσης να λειτουργεί, αλλά με μόνο αυτές τις δύο συνιστώσες (components) δεν υπάρχει καμία σύνδεση μεταξύ της ομιλίας και των λειτουργιών που επιθυμούμε να επιτελούνται.

Ένα σύστημα κατανόησης προφορικού λόγου (Spoken Language Understanding - SLU) χρησιμοποιείται όταν απαιτείται να ερμηνευτούν εκφωνήματα μέσα σε κάποιο σημασιολογικό πλαίσιο και στη συνέχεια να πραγματοποιηθούν κατάλληλες ενέργειες [b6].

Σε ένα τέτοιο σύστημα υπάρχουν τέσσερις κύριες θεματικές περιοχές από τις οποίες εξάγεται το νόημα [b6]:

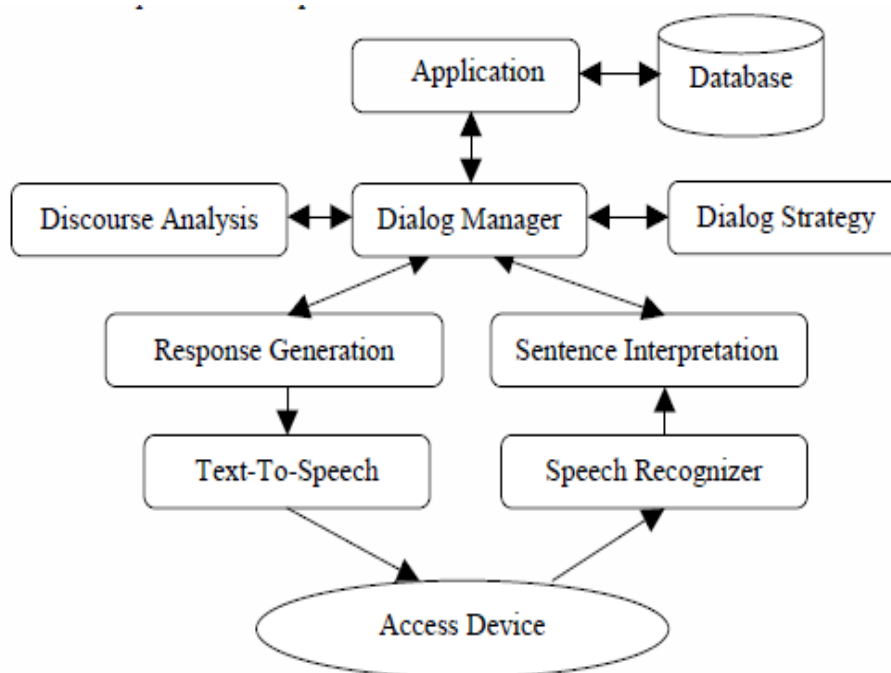
- Προθέσεις (Intents) : καταδεικνύει τους σκοπούς τόσο του ομιλητή όσο και του ακροατή σε μία συνομιλία.
- Πλαίσιο (Context) : στοιχεία όπως οι ευκαιρίες για ομιλία, οι διακοπές κ.τ.λ. σε μία συγκεκριμένη σκηνή αλληλεπίδρασης.
- Περιεχόμενο (Content) : εκφράζει το νοηματικό ή κυριολεκτικό περιεχόμενο κάθε εκφωνήματος (utterance) αλλά και ολόκληρης της συνομιλίας
- Υποθέσεις (assumptions) : το τι υποθέτει ο κάθε συνομιλητής για τους υπόλοιπους όσον αφορά δυνατότητες, πνευματική κατάσταση κ.τ.λ.



2.5.2 Δομή συστημάτων SLU

Οι βασικές συνιστώσες ενός συστήματος SLU είναι ο *διαχειριστής διαλόγου (dialog manager)*, η *ερμηνεία προτάσεων (sentence interpretation)*, η *ανάλυση της ομιλίας (discourse analysis)* και οι συνιστώσες της *αναγνώρισης φωνής (speech recognizer)* και *σύνθεσης φωνής από κείμενο (text-to-speech synthesis)* αντίστοιχα.

Όπως φαίνεται και στο Σχήμα 5 πιο κάτω, ο *διαχειριστής διαλόγου* παίζει κεντρικό ρόλο καθώς ελέγχει την ροή της συζήτησης και αποτελεί κομβικό σημείο επικοινωνίας με τις υπόλοιπες συνιστώσες. Παρέχει επίσης τα απαραίτητα δεδομένα για τον *σχηματισμό αποκρίσεων (response generation)* και έχει την ευθύνη του να συντηρεί την «εντύπωση» του συστήματος (*system’s idea*) για την *κατάσταση της ομιλίας (discourse state)* [b6]. Η *κατάσταση της ομιλίας* καταγράφει δεδομένα που αφορούν στην τρέχουσα συνομιλία, στα τρέχοντα αντικείμενα που φαίνεται να είναι στο επίκεντρο της προσοχής, στην ιστορικότητα αυτών των αντικειμένων και άλλα στοιχεία κατάστασης. Αυτού του είδους η πληροφορία είναι σημαντική για την σημασιολογική ερμηνεία (*semantic interpretation*) των εκφωνημάτων [b6]. Από τα υπόλοιπα μέρη, οι συνιστώσες της *αναγνώρισης φωνής* και *σύνθεσης φωνής από κείμενο* αναλαμβάνουν τις διαδικασίες εισόδου και εξόδου του συστήματος αντίστοιχα όπως αυτές έχουν περιγραφεί στις προηγούμενες ενότητες. Στην διαδικασία εισόδου κατά την *ερμηνεία προτάσεων (sentence interpretation)* τα αποτελέσματα της αναγνώρισης φωνής αναλύονται σε σημασιολογικές δομές (*semantic forms*), ενώ πολλές φορές απαιτείται και *ανάλυση του συνεχούς λόγου (discourse analysis)* για την παρακολούθηση του περιεχομένου και την εξάλειψη ανακολουθιών [b6].



Σχήμα 5: Βασική δομή ενός συστήματος SLU

Τα διάφορα υποσυστήματα έχουν την δυνατότητα να αλλάζουν την ροή της πληροφορίας όπως φαίνεται στο Σχήμα 5 με αποτέλεσμα διάφορες βελτιώσεις. Έτσι πχ ο *διαχειριστής διαλόγου* μπορεί να παρέχει πληροφορίες ώστε να καθοδηγεί την αναγνώριση φωνής σε πρώιμα στάδια της διαδικασίας [b6].

2.6 Προϊόντα λογισμικού στον τομέα επεξεργασίας φωνής

Τα τελευταία χρόνια ο τομέας ανάπτυξης και χρήσης εφαρμογών που αφορούν στην αναγνώριση και σύνθεση φωνής παρουσιάζει αυξημένη δραστηριότητα. Έτσι στον χώρο παρατηρούμε ότι υπάρχει πληθώρα προϊόντων (Πίνακας 2) που αποτελούν είτε αυτόνομες εμπορικές εφαρμογές είτε πλατφόρμες και εργαλεία ανάπτυξης λογισμικού. Παρουσιάζει δε ιδιαίτερο ενδιαφέρον η ανάπτυξη ιδιαίτερα αξιόλογων τέτοιων προϊόντων ανοιχτού κώδικα (open-source) που μπορούν να χρησιμοποιηθούν χωρίς περιορισμούς.

Πίνακας 2: Δείγμα διαθέσιμων προϊόντων στον τομέα επεξεργασίας φωνής

Προϊόν	Εταιρία - Φορέας	Σύντομη περιγραφή	Τύπος προϊόντος
WebSphere Voice	IBM	Οικογένεια προϊόντων φωνής που εκτείνεται από μικρές εφαρμογές γραφείου μέχρι ανάγκες μεγάλων επιχειρήσεων	Εμπορικά προϊόντα



		http://www-01.ibm.com/software/voice/	
Dragon Naturally Speaking Solutions	Nuance	Σουίτα προϊόντων που απευθύνεται από καταναλωτές με απλές ανάγκες μέχρι επιχειρήσεις και επαγγελματίες προγραμματιστές από μία εταιρία που εξειδικεύεται στον χώρο http://www.nuance.com/naturallyspeaking/	Εμπορικά προϊόντα
Microsoft Speech Technologies	Microsoft	Τεχνολογίες και υπηρεσίες φωνής για καταναλωτές επιχειρήσεις και επαγγελματίες προγραμματιστές http://www.microsoft.com/SPEECH/	Εμπορικά αλλά και ελεύθερα διαθέσιμα προϊόντα (freeware)
CMU Sphinx	Carnegie Mellon University	Λογισμικό αναγνώρισης φωνής υλοποιημένο εξολοκλήρου με τη γλώσσα Java http://cmusphinx.sourceforge.net/	Λογισμικό ανοικτού κώδικα
Simon	Simon listens	Εφαρμογή αναγνώρισης φωνής για άτομα με περιορισμένη κινητικότητα http://www.simon-listens.org	Λογισμικό ανοικτού κώδικα
eSpeak text to speech		Συνθέτης φωνής για πολλές γλώσσες συμπεριλαμβανομένων των αγγλικών και των ελληνικών http://java.sun.com/products/java-media/speech/	Λογισμικό ανοικτού κώδικα
Java Speech API	Oracle	Προγραμματιστική διεπαφή για ανάπτυξη εφαρμογών με βάση τη γλώσσα Java http://espeak.sourceforge.net/	Ελεύθερα διαθέσιμο λογισμικό (freeware)

2.6 Επεξεργασία φωνής σε περιβάλλον Windows

Η Microsoft συμμετέχει στην ανάπτυξη αυτόματων συστημάτων αναγνώρισης φωνής για περισσότερα από 15 χρόνια. Το 1993 προσλαμβάνονται τα ιδρυτικά στελέχη του Speech Research Group από το πανεπιστήμιο Carnegie Mellon και ταυτόχρονα ξεκινάει οι πρώτες προσπάθειες με βάση την άδεια του τότε πρωτοπόρου συστήματος αναγνώρισης Sphinx II[4].



Το 1999 γίνεται εξαγορά της εταιρίας Entropic η οποία εξέλιξε το σύνολο εργαλείων Hidden Markov Model Toolkit (HTK) αρχικά υλοποιημένο από το τμήμα μηχανικής του Cambridge University [i4]. Το 2001 με την πρώτη έκδοση των Windows XP συμπεριλαμβάνεται και η σημαντική 5.01 έκδοση του Speech Recognition API (SAPI) η οποία παρέχει δυνατότητες τόσο αναγνώρισης όσο και σύνθεσης φωνής. Το 2007 πραγματοποιείται εξαγορά της εταιρίας Tellme Networks προς ενίσχυση του τομέα της τεχνολογίας φωνής [i4].

Σήμερα, υποστηρίζονται από την Microsoft τεχνολογίες αναγνώρισης και σύνθεσης φωνής στις παρακάτω υποδομές :

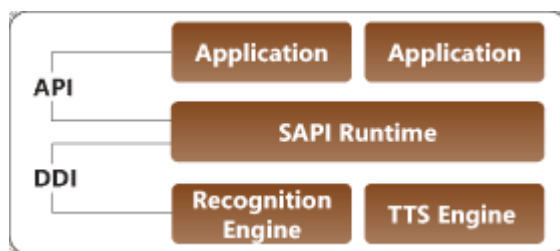
- Εφαρμογές γραφείου (desktop applications), με ενσωματωμένες δυνατότητες αναγνώρισης και σύνθεσης φωνής ως μέρος των λειτουργικών συστημάτων Windows Vista και Windows 7
- Εφαρμογές διακομιστών (server applications), με υποδομές σε προϊόντα όπως το Microsoft Exchange 2007 Unified Messaging, ο Microsoft Office Communication Server 2007 R2 Speech Server κτλ
- Ενσωματωμένα συστήματα (Embedded Systems), με ειδικές εκδόσεις του SAPI για το αντίστοιχο λειτουργικό σύστημα Microsoft’s Windows CE.

2.6.1 Speech Recognition API (SAPI)

Το Microsoft Speech Recognition API παρέχει μία αφαιρετική προγραμματιστική διεπαφή (high level interface) μεταξύ των εφαρμογών και των μονάδων φωνητικής επεξεργασίας (speech engines). Το SAPI μειώνει κατά πολύ την επιβάρυνση ανάπτυξης του κώδικα που απαιτείται για την αξιοποίηση τεχνολογιών φωνής από εφαρμογές. Διαχειρίζεται εσωτερικά θέματα όπως η χρήση συσκευών ήχου και ο χειρισμός νημάτων (thread handling) επιτρέποντας έτσι στην ανάπτυξη λογισμικού εφαρμογών φωνής να επικεντρώνεται κάθε φορά στην επιχειρησιακή λογική ελέγχου και όχι σε χαμηλού επιπέδου χαρακτηριστικά [i4]

- **Συνιστώσες SAPI (SAPI components)**

Μια αφαιρετική όψη της αρχιτεκτονικής των βασικών συνιστωσών (components) του SAPI κατά την επικοινωνία με εφαρμογές, φαίνεται στο παρακάτω σχήμα:



Σχήμα 6: Επικοινωνία εφαρμογών με το SAPI

Όπως παρατηρούμε και στο Σχήμα 6, στο SAPI περιλαμβάνονται οι δύο βασικοί τύποι μονάδων φωνητικής επεξεργασίας: η μηχανή αναγνώρισης φωνής (Recognition Engine - RE) και η μηχανή σύνθεσης φωνής (Text to Speech Engine - TTS). Η μηχανή αναγνώρισης φωνής δέχεται ως είσοδο ψηφιακά δεδομένα φωνής (voice audio stream) και παράγει το αντίστοιχο κείμενο. Η μηχανή σύνθεσης φωνής κάνει το αντίστροφο: παίρνει ως είσοδο κάποιο κείμενο και παράγει την αντίστοιχη φωνητική έκφραση [i8].

Το SAPI περιλαμβάνει εσωτερικά κατάλληλη προγραμματιστική διεπαφή (Application Programmatic Interface - API) για την επικοινωνία μεταξύ των εφαρμογών και των δύο προαναφερόμενων μονάδων RE και TTS. Επίσης περιλαμβάνει διεπαφή σε χαμηλότερο επίπεδο οδηγών (Device Driver Interface - DDI) μέσω του οποίου επικοινωνούν οι μονάδες RE και TTS με το SAPI. Η αρχιτεκτονική αυτή εγγυάται ότι οι εφαρμογές δεν επικοινωνούν απευθείας με τις μονάδες φωνητικής επεξεργασίας παρά μόνο μέσω του SAPI [i6].

- **Αρμοδιότητες SAPI**

Το SAPI έχει την ευθύνη των παρακάτω λειτουργιών [i6]:

- Να αξιοποιεί τις πηγές ήχου (μικρόφωνο, αρχείο φωνής κτλ) και να μετατρέπει την είσοδο σε κατάλληλη μορφή προς χρήση από τις μονάδες επεξεργασίας φωνής.
- Να αξιοποιεί αρχεία γραμματικής, είτε αυτά παρέχονται ως στατικά από διάφορες πηγές (πχ μνήμη, σκληρός δίσκος, διαδικτυακή διεύθυνση κτλ), είτε παρέχονται δυναμικά και να εξασφαλίζει την εγκυρότητα των απαιτούμενων γραμματικών κανόνων.
- Να μεταγλωττίζει γραμματικές που είναι αρχικά μορφοποιημένες σε XML (XML grammar format), αλλά και άλλες εξειδικευμένες μορφοποιήσεις γραμματικής, σε αποτελέσματα κατανοητά στις μηχανές.
- Να χειρίζεται τις κλήσεις για αναγνώριση μεταξύ πολλών εφαρμογών.
- Να αποθηκεύει φωνητικά δεδομένα ή αποτελέσματα για μεταγενέστερη χρήση.



- Να βεβαιώνει ότι οι εφαρμογές δεν προκαλούν λάθη στο λογισμικό των μονάδων: να προστατεύει από τροφοδότηση λανθασμένων παραμέτρων και να χειρίζεται περιπτώσεις μη-απόκρισης των εφαρμογών (application hanging or crashing)

- **Speech Recognition Engine (SRE)**

Το Speech Recognition Engine (SRE) αποτελεί τη μονάδα λογισμικού που είναι υπεύθυνη για την αναγνώριση φωνής. Έχει τις εξής αρμοδιότητες [i6]:

- Να πραγματοποιεί αναγνώριση φωνής
- Να εκτελεί ερωτήματα στο SAPI σε τακτά χρονικά διαστήματα (polling) σχετικά με πιθανές αλλαγές στη χρησιμοποιούμενη γραμματική ή γενικότερες αλλαγές κατάστασης που το αφορούν
- Να παράγει συμβάντα (programmatic events) για πληθώρα γεγονότων ώστε να παρέχει κατάλληλες ειδοποιήσεις (notifications) για σημαντικές πληροφορίες
- Να χρησιμοποιεί μέσω του DDI τις διεπαφές του SAPI που αφορούν στη γραμματική

- **Text to Speech Engine (TTS)**

Η μηχανή Text to Speech (TTS Engine) αποτελεί τη μονάδα λογισμικού που είναι υπεύθυνη για την παραγωγή συνθετικής φωνής από κείμενο. Η συνθετική φωνή που παράγεται είναι παρόμοια με την ανθρώπινη λόγω της εφαρμογής διαφόρων φίλτρων. Οι μηχανές TTS που είναι συμβατές με την έκδοση 5 ή μεταγενέστερες του SAPI μπορούν να επιτύχουν βελτιώσεις στην ποιότητα της παραγόμενης συνθετικής φωνής καθώς μπορούν να χρησιμοποιήσουν τεχνολογίες XML για να ελέγξουν την παραμετροποίηση χαρακτηριστικών φωνής. Έχει τις εξής αρμοδιότητες [i6]:

- Να παράγει συνθετική φωνή χρησιμοποιώντας πηγές κειμένου που έχουν Unicode κωδικοποίηση. Η παραγωγή αυτή μπορεί να γίνει σύγχρονα ή ασύγχρονα
- Να δίνει στα ανώτερα επίπεδα λογισμικού την δυνατότητα εξαγωγής πληροφοριών που αφορούν στην τρέχουσα κατάσταση (όπως π.χ. η τρέχουσα θέση μέσα στο κείμενο), είτε με χρήση ερωτημάτων σε τακτά χρονικά διαστήματα (polling), είτε με συμβάντα πραγματικού χρόνου (programmatic events)
- Να επιτρέπει τον έλεγχο της ροής δίνοντας την δυνατότητα προσωρινής παύσης και επαναφοράς της παραγωγής συνθετικής φωνής
- Να επιτρέπει την τροποποίηση των προκαθορισμένων χαρακτηριστικών (default settings) είτε χρησιμοποιώντας συγκεκριμένες προγραμματιστικές μεθόδους είτε με χρήση XML δομών



- Να υποστηρίζει σχεδόν οποιαδήποτε παραμετροποίηση μπορεί να αφορά στον προορισμό του παραγόμενου αποτελέσματος (κάρτα ήχου, ενδιάμεση μνήμη, ειδικό υλικό τηλεφωνίας), καθώς παρέχει πληθώρα μεθόδων ελέγχου της δρομολόγησης του ήχου (audio control methods)



3. Ρομποτικά συστήματα

Ο τομέας της ρομποτικής αποτελεί έναν πολλά υποσχόμενο χώρο που ήδη παίζει σημαντικό παραγωγικό ρόλο σε τομείς όπως οι αυτοκινητοβιομηχανίες. Η ενότητα αφού περιγράφει πολύ συνοπτικά τα πεδία εφαρμογών της ρομποτικής, ακολουθεί συγκεκριμένη προσέγγιση για τον ορισμό ενός ρομπότ για τον οποίο δεν υπάρχει κοινή τοποθέτηση στην επιστημονική κοινότητα. Έπειτα επικεντρώνεται στην περιγραφή του εκπαιδευτικού ρομποτικού μοντέλου Lego NXT.

3.1 Γενικά για τα ρομπότ

3.1.1 Ο ρόλος της τέχνης

Ο όρος ‘ρομπότ’ εισάγεται και αποκτά δημοσιότητα στην αντίληψη του κοινού το 1921 στην Πράγα με το θεατρικό έργο του Karel Capek ‘Rossum’s Universal Robots’, όπου τα ρομπότ αποτελούν μία κατώτερη φυλή εργατών φτιαγμένων από βιολογικά μέρη εξαναγκασμένων να υπηρετούν τους ανθρώπους. Η διαμόρφωση του όρου συνεχίζεται με τις κλασικές ταινίες επιστημονικής φαντασίας ‘Metropolis’ το 1926, ‘The Day of the Earth’ 1951 και ‘Forbidden Planet’ το 1956, όπου η έννοια του ρομπότ πλέον αλλάζει και του αποδίδονται μηχανικά μέρη [b9].

3.1.2 Πεδία εφαρμογών της ρομποτικής

Η σύγχρονη εξέλιξη στην ρομποτική καλύπτουν έναν ευρύ τομέα υποβοηθούμενες από το μελλοντικό όραμα ότι τα ρομπότ θα μπορούν να αναλάβουν όλες τις εργασίες οι οποίες είναι επικίνδυνες, ρυπαρές και πληκτικές (dangerous, dirty and dreary - DDD). Η σημερινή τεχνολογία δείχνει ότι αυτό μπορεί να καταστεί εφικτό καθώς παρατηρούμε ήδη ένα επίπεδο ωριμότητας στην ρομποτική που έχει οδηγήσει τα ρομπότ από τα εργοστάσια σε διάφορους εξειδικευμένους τομείς εφαρμογών και υπηρεσιών όπως πολύ συνοπτικά αναφέρονται οι ακόλουθοι [b13]:

- Βιομηχανική ρομποτική (Industrial Robotics)
- Υποβρύχια ρομποτική (Underwater Robotics)
- Εναέρια ρομποτική (Aerial Robotics)
- Διαστημική ρομποτική (Space Robots and Systems)
- Ρομποτική στην γεωργία και στην δασοπονία (Robotics in Agriculture and Forestry)
- Ρομποτική στον κατασκευαστικό τομέα (Robotics in Construction)



- Ρομποτική που αφορά σε εφαρμογές υψηλού κινδύνου (Robotics in Hazardous Applications)
- Μεταλλευτική ρομποτική (Mining Robotics)
- Ρομποτική που αφορά στην αναζήτηση και διάσωση ανθρώπων (Search and Rescue Robotics)
- Ευφυή οχήματα (Intelligent Vehicles)
- Ρομποτική στην Ιατρική (Medical Robots and Systems)
- Ρομποτική αποκατάστασης και φροντίδας για την υγεία (Rehabilitation and Health Care Robotics)
- Οικιακή Ρομποτική (Domestic Robotics)
- Εκπαιδευτική Ρομποτική (Robots for Education)

Στην Εικόνα 1 βλέπουμε ένα δείγμα ρομποτικής αναζήτησης και διάσωσης (Search and Rescue Robotics): το ρομπότ V2 χρησιμοποιήθηκε στην καταστροφή που επακολούθησε την έκρηξη στο ορυχείο Sago στην Δυτική Βιρτζίνια των ΗΠΑ το 2006 [b13].



Εικόνα 1: Ρομπότ διάσωσης V2 [b13]

3.1.3 Ο ορισμός του ρομπότ

Παρουσιάζει ενδιαφέρον το γεγονός ότι ακόμα και σήμερα δεν υπάρχει από τους ειδήμονες συμφωνία ως προς τον ορισμό [i18] [b13]. Αν και έχει δοθεί ορισμός από τον οργανισμό προτύπων ISO (ISO 8373) σύμφωνα με τον οποίο: «ρομπότ είναι ένας επαναπρογραμματιζόμενος βραχίονας πολλαπλών λειτουργιών σχεδιασμένος για να μεταφέρει υλικά, εξαρτήματα, εργαλεία ή άλλες ειδικές διατάξεις μέσω προγραμματισμένων κινήσεων για



την επίτευξη ενός αριθμού εργασιών», είναι φανερό ότι δεν καλύπτει πλήρως όλο το δυνατό εύρος εφαρμογών, καθώς αφορά αποκλειστικά σε βιομηχανικούς βραχίονες.

- **Το ευφύες ρομπότ**

Η παρούσα εργασία ακολουθεί συγκεκριμένη προσέγγιση [b9] στην οποία διαχωρίζεται η έννοια του *ρομπότ* από αυτή του *ευφυούς ρομπότ*, με βάση το ειδικό βάρος του ρόλου της αυτονομίας:

Το *ευφύες ρομπότ* είναι μία μηχανική οντότητα (mechanical creature) η οποία μπορεί να δράσει αυτόνομα [b9].

Ο όρος ‘ευφύες’ συνεπάγεται ότι το ρομπότ δεν πραγματοποιεί απλώς εργασίες με έναν μηχανικό και επαναλαμβανόμενο τρόπο και έτσι διαχωρίζεται από τις αυτοματοποιήσεις όπως ο προαναφερόμενος βιομηχανικός βραχίονας όσο εξελιγμένες και αν είναι αυτές. Ο όρος ‘μηχανικό δημιούργημα’ καταρχήν σημαίνει ότι ένα ρομπότ αποτελείται από μηχανικά και (προς το παρόν) όχι βιολογικά τμήματα. Επίσης επισημαίνει την ικανότητα του ρομπότ να αλληλεπιδρά με τον πραγματικό κόσμο (να κινείται, να αλλάζει την κατάσταση γύρω του κτλ.). Τέλος το ότι το ρομπότ μπορεί να ‘δράσει αυτόνομα’ σημαίνει ότι μπορεί καταρχήν να λειτουργεί με έναν επαρκή (self-contained) τρόπο χωρίς την συνεχή διαμεσολάβηση κάποιου ανθρώπου και επίσης μπορεί να προσαρμόζεται στις αλλαγές του περιβάλλοντος για την επίτευξη κάποιου στόχου [b9].

3.1.4 Σύγχρονα ευφύη ρομπότ : το ευφύες όχημα

Χαρακτηριστικό παράδειγμα ευφυούς ρομπότ αποτελεί το ευφύες όχημα. Η ιδέα των ευφυών οχημάτων ξεκινάει το 1960 αλλά τα πρώτα πρωτότυπα αναπτύσσονται από τον στρατιωτικό τομέα στα μέσα της δεκαετίας του ‘80 ενώ στην έχουμε και μεταφορά του ενδιαφέροντος στην ανάπτυξη οχημάτων για περιβάλλοντα πόλης. Στα μέσα της δεκαετίας του ‘90 κατασκευάζονται τα πρώτα πρωτοποριακά μοντέλα και το 1995 ομάδα από το πανεπιστήμιο Carnegie Mellon πραγματοποιεί το πείραμα που ονομάζεται ‘No Hands Across America’. Σε αυτό το πείραμα πραγματοποιήθηκε αυτόματα οδήγηση βασισμένη αποκλειστικά σε μηχανική όραση στο 98% του χρόνου μίας διαδρομής 2800 μιλίων [b13].

Οι εξελίξεις στον τομέα από εκεί και πέρα είναι ραγδαίες με κάποια σημαντικά ορόσημα: Παρακάτω στην Εικόνα 2 βλέπουμε πέντε οχήματα τα οποία τερμάτισαν στα πλαίσια του αγώνα ‘Grand Challenge’ που οργανώθηκε το 2003 από το τμήμα έρευνας DARPA (Defense Advanced Research Projects Agency) των ΗΠΑ. Η πρόκληση αφορούσε σε ευφύη



οχήματα τα οποία έπρεπε να καλύψουν μία απόσταση μεγαλύτερη των 200 χιλιομέτρων σε ένα σενάριο αυτόνομης οδήγησης σε περιβάλλον ερήμου και χωρίς να υπάρχει εκ των προτέρων γνώση ή χαρτογράφηση εμποδίων, παρά μόνο ελάχιστα βοηθητικά σημεία επισήμανσης (road markers) σε μία διαδρομή προκαθορισμένη από περίπου 3000 GPS σημεία [b13]. Στον διαγωνισμό συμμετείχε μεγάλος αριθμός ερευνητικών φορέων ενώ η διοργάνωση θεωρείται ότι πρόσφερε σημαντική αρωγή στην επιστημονική γνώση και εξέλιξη του συγκεκριμένου τομέα.



Εικόνα 2: Αυτόνομα, ευφυή οχήματα στον διαγωνισμό DARPA Grand Challenge [b13]

Αν και οι πρόσφατες και τρέχουσες ερευνητικές πρωτοβουλίες έχουν δείξει ότι η πλήρης αυτονομία τέτοιων οχημάτων είναι πλέον ρεαλιστική, κυρίως για νομικούς λόγους αυτό δεν αποτελεί πρωτεύοντα στόχο. Έτσι, αμεσότερος στόχος στη σύγχρονη αυτοκινητοβιομηχανία είναι η εξέλιξη οχημάτων με συστήματα επίβλεψης και γενικότερα η ανάπτυξη εξελιγμένων συστημάτων υποβοήθησης οδήγησης (advanced driving assistance systems - ADAS). Σε αυτά τα οχήματα παρόλο που ο οδηγός έχει τον κυρίως έλεγχο, το όχημα είναι εφοδιασμένο με ηλεκτρονικά και μηχανικά συστήματα που επιτρέπουν την ανίχνευση πιθανών κινδύνων και δρουν κατάλληλα είτε μέσω ειδοποιήσεων, είτε ακόμα και αναλαμβάνοντας τον πλήρη έλεγχο του οχήματος [b13].



3.2 Το εκπαιδευτικό ρομπότ Lego MINDSTORMS NXT

Το εκπαιδευτικό ρομποτικό μοντέλο Lego MINDSTORMS NXT της εταιρίας Lego απέκτησε μεγάλη επιτυχία [b8] από τις αρχικές εκδόσεις του. Η επιτυχημένη αυτή πορεία η οποία συνεχίζεται μέχρι και τις τελευταίες εκδόσεις, δεν αφορά απλώς σε κάποιον αριθμό πωλήσεων αλλά κυρίως στην υιοθέτηση του από μία ευρεία κοινότητα που συμπεριλαμβάνει ερασιτέχνες ενδιαφερόμενους, μαθητές, σπουδαστές και ακαδημαϊκούς φορείς με τελικό αποτέλεσμα την διάχυση και συνεισφορά στην επιστημονική γνώση με έναν ιδιαίτερα ελκυστικό τρόπο.



Εικόνα 3 : Το μοντέλο Lego NXT με συνδεδεμένες μονάδες αισθητήρων και επενεργητών

3.2.1 Σύντομο ιστορικό

Το 1998 η εταιρία Lego παρουσιάζει το πρώτο MINDSTORMS ως σύνολο σερβομηχανισμών. Το σύστημα καλείται Robotics Invention System (RIS) και παρέχει προγραμματιζόμενο αυτόνομο τμήμα που ονομάζεται Robotic Control Explorer (RCX). Το 2006 παρουσιάζει την εξέλιξη του συστήματος που αποτελεί το MINDSTORMS NXT, ενώ την τελευταία εξέλιξη αποτελεί η έκδοση του Lego NXT 2.0 (8547) το καλοκαίρι του 2009[b8], το οποίο και χρησιμοποιείται στην παρούσα εργασία.



3.2.2 Lego NXT brick

Η θεμελιώδης μονάδα του Lego NXT που περιλαμβάνει τον μικροεπεξεργαστή 32bit, έχει μορφή πλίνθου και για αυτό εύλογα καλείται *NXT intelligent brick*. Το *NXT brick* έχει την ευθύνη της επεξεργασίας των δεδομένων, τον συγχρονισμό των μηχανισμών του Lego NXT και συντονίζει την επικοινωνία με τρίτα συστήματα μέσω των πρωτοκόλλων Bluetooth και USB 2.0.

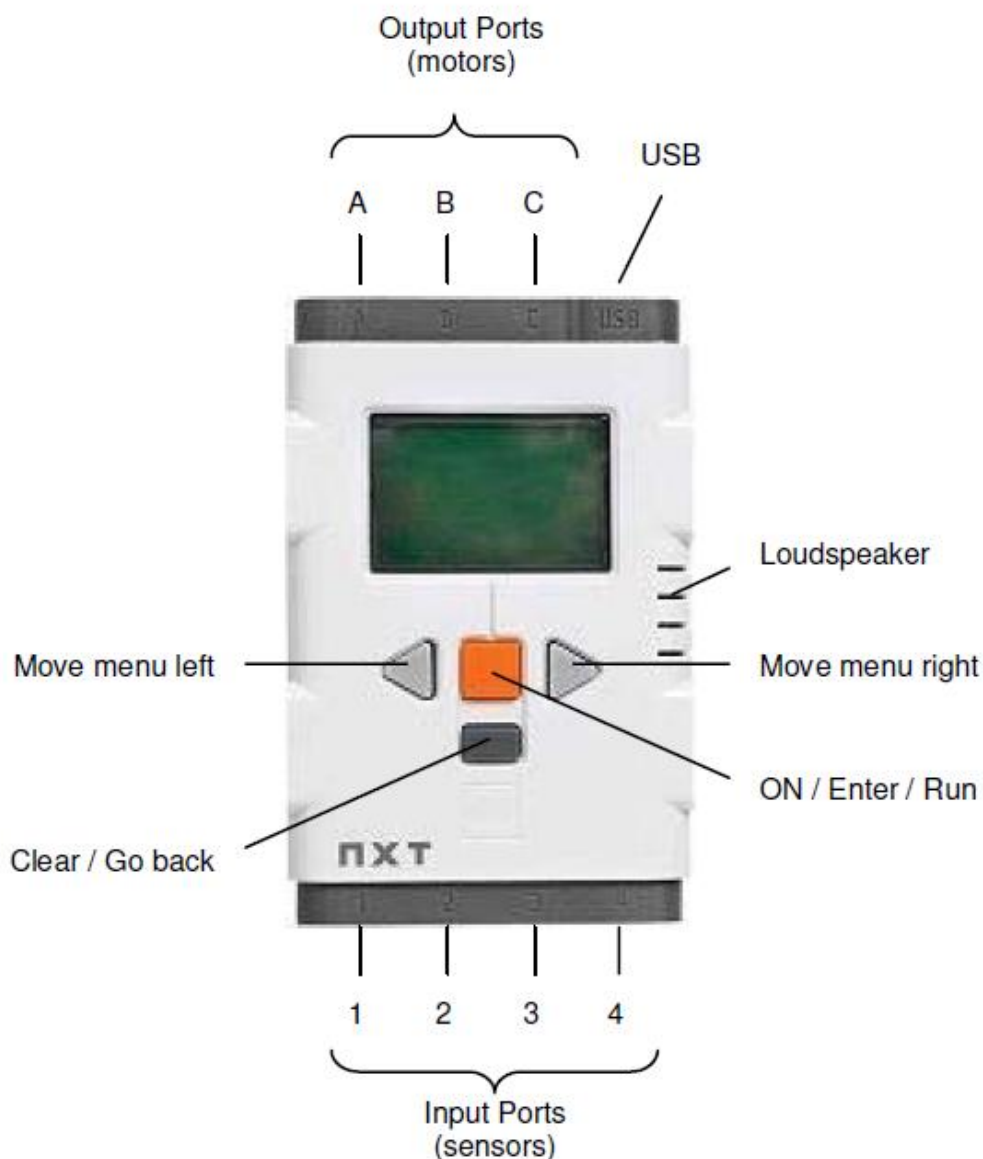
- **Διεπαφές υλικού NXT brick**

Επάνω στο *NXT brick* διακρίνονται οι διεπαφές υλικού που διασύνδεουν τον επεξεργαστή με διάφορες μονάδες, αλλά επιτρέπουν και την εύκολη παραμετροποίηση από τον άνθρωπο που χειρίζεται το ρομπότ [i14] [i16] [i17]:

- **Θύρες εξόδου (output ports):** υπάρχουν τρεις θύρες εξόδου που αναγράφονται A, B και C αντίστοιχα που χρησιμοποιούνται για την διασύνδεση *επενεργητών (actuators)*, όπως είναι οι σερβο-κινητήρες (*servo-motors*).
- **Θύρες εισόδου (input ports):** υπάρχουν τέσσερις θύρες εισόδου που αριθμούνται 1, 2, 3, 4 αντίστοιχα και επιτρέπουν στο *NXT brick* την μέτρηση παραμέτρων εισόδου, ανάλογα με τον *αισθητήρα (sensor)* που έχει συνδεθεί σε κάθε μία από αυτές.
- **Θύρα USB (USB port) :** επιτρέπουν την διασύνδεση με τρίτα συστήματα μέσω ενός απλού USB καλωδίου και την ανταλλαγή δεδομένων.
- **Μεγάφωνο (loudspeaker) :** μπορεί να αναπαράγει ήχους με βάση το πρόγραμμα που εκείνη τη στιγμή εκτελείται.
- **Οθόνη ενδείξεων LCD (LCD display):** παρέχει το μενού επιλογών και παρουσιάζει όλες τις απαραίτητες ενδείξεις στον χρήστη (π.χ. τρέχουσα συνδεσιμότητα με Bluetooth)
- **Κουμπιά ελέγχου (NXT buttons):** ελέγχουν λειτουργίες και την παραμετροποίηση του επεξεργαστή. Συγκεκριμένα, το κουμπί ελέγχου με πορτοκαλί χρωματισμό αντιστοιχεί στις εντολές ‘ON’, ‘Enter’ και ‘Run’ με τις οποίες ενεργοποιούμε το *NXT brick*, καταχωρούμε κάποια εντολή και εκτελούμε κάποιο πρόγραμμα αντίστοιχα. Τα δύο κουμπιά ελέγχου με ανοιχτό γκρι χρωματισμό χρησιμοποιούνται για την πλοήγηση μέσα στο μενού που παρουσιάζεται στην οθόνη, ενώ το κουμπί με τον σκούρο γκρι χρωματισμό αντιστοιχεί στις εντολές ‘Clear’, ‘Go Back’ και ‘OFF’ με τις οποίες καθαρίζουμε κάποια επιλογή, πλοηγούμαστε στην προηγούμενη επιλογή του μενού και απενεργοποιούμε το *NXT brick* αντίστοιχα.



Στην Εικόνα 4, βλέπουμε μία χαρακτηριστική εξωτερική όψη του *NXT brick* και τις προαναφερόμενες διεπαφές υλικού :



Εικόνα 4: Το NTX brick και οι εξωτερικές διεπαφές του [i17]

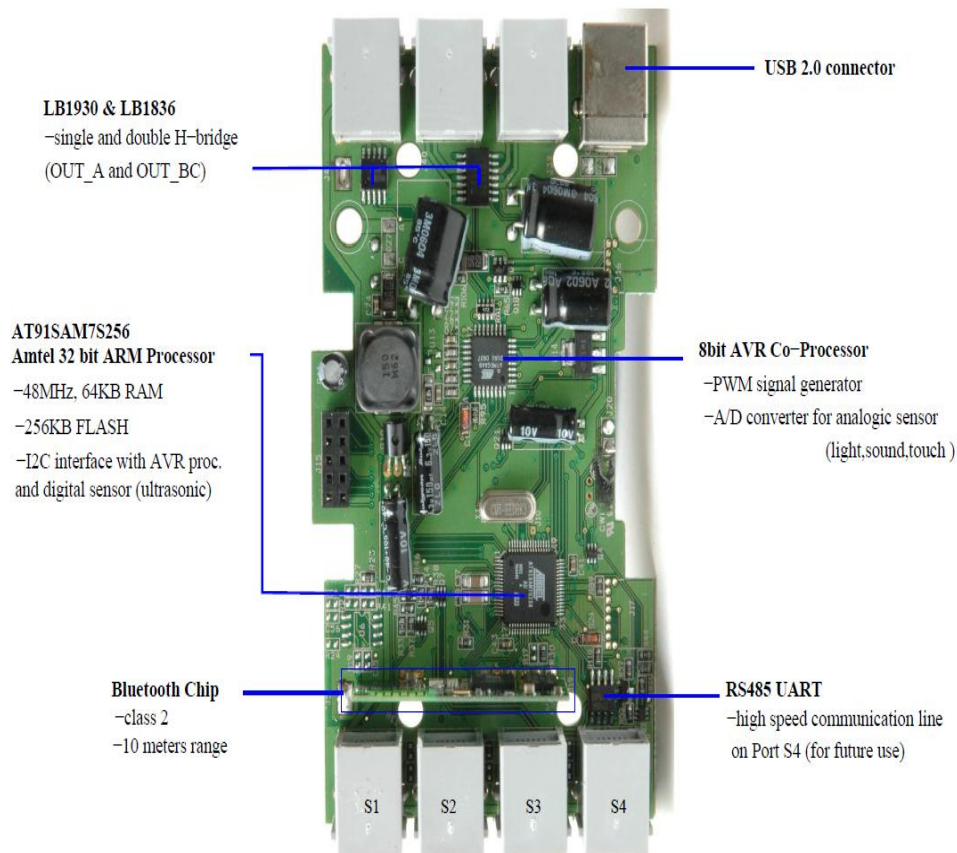
- **Προδιαγραφές υλικού στο NXT brick**

Το *NXT brick* για να παρέχει τις υπηρεσίες του, κάνει χρήση υψηλού επιπέδου ηλεκτρονικών κυκλωμάτων. Ο παρακάτω πίνακας αποτελεί μία σύνοψη των προδιαγραφών υλικού του επεξεργαστή [i14]:

**Πίνακας 3:** Προδιαγραφές υλικού NXT brick

Επεξεργαστής (main processor)	Atmel® 32-bit ARM® processor, AT91SAM7S256 - 256 KB FLASH - 64 KB RAM - 48 MHz
Συνεπεξεργαστής (co-processor)	Atmel® 8-bit AVR processor, ATmega48 - 4 KB FLASH - 512 Byte RAM - 8 MHz
Bluetooth επικοινωνία (Bluetooth wireless communication)	CSR BlueCore™ 4 v2.0 +EDR System - Supporting the Serial Port Profile (SPP) - Internal 47 KByte RAM - External 8 MBit FLASH - 26 MHz
USB 2.0 επικοινωνία (USB 2.0 communication)	Full speed port (12 Mbit/s)
4 θύρες εισόδου (4 input ports)	6-wire interface supporting both digital and analog interface - 1 high speed port, IEC 61158 Type 4/EN 50170 compliant
3 θύρες εξόδου 3 output ports	6-wire interface supporting input from encoders
Οθόνη ενδείξεων LCD (LCD Display)	100 x 64 pixel LCD black & white graphical display - View area: 26 X 40.6 mm
Μεγάφωνο (Loudspeaker)	Sound output channel with 8-bit resolution - Supporting a sample rate of 2-16 KHz
Διεπαφή τεσσάρων κουμπιών με τον χρήστη (4 button user-interface)	Rubber buttons
Πηγές ενέργειας (Power source)	6 AA batteries - Alkaline batteries are recommended - Rechargeable Lithium-Ion battery 1400 mAH is available
Κύκλωμα διασύνδεσης (Connector)	6-wire industry-standard connector, RJ12 Right side adjustment

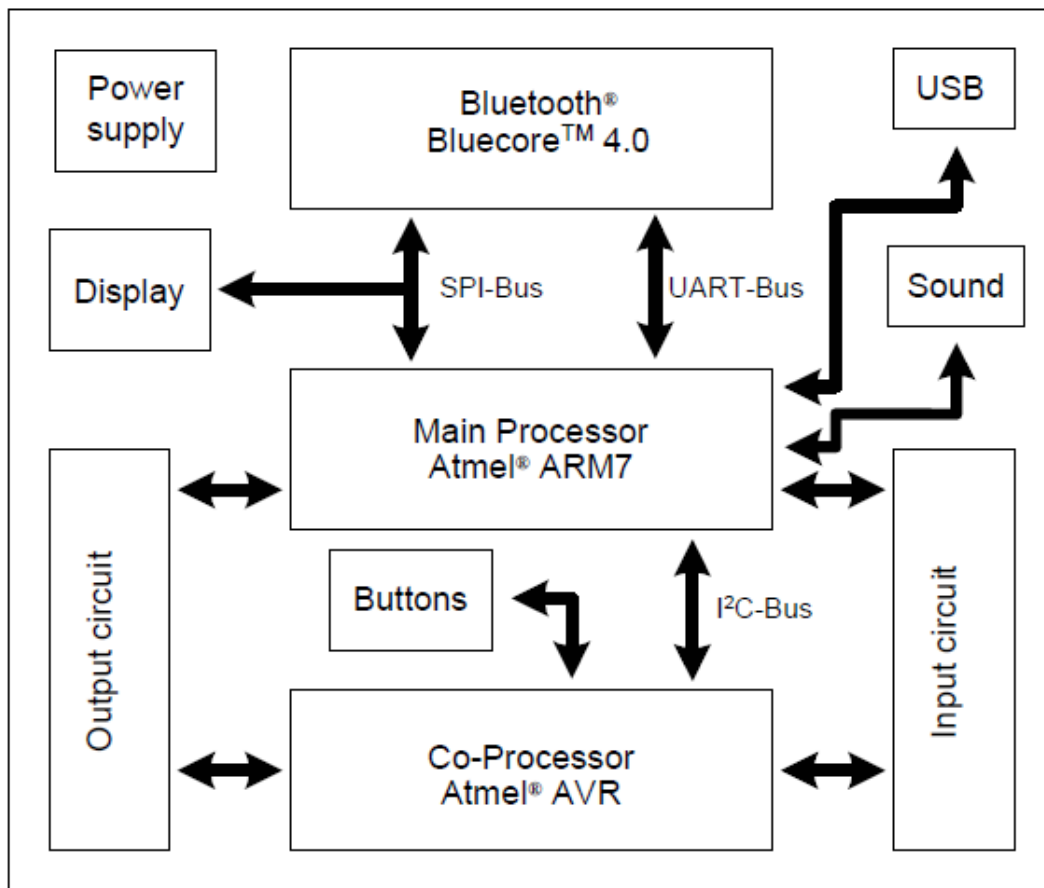
Αμέσως παρακάτω στην Εικόνα 5, έχουμε μία εσωτερική όψη του *NXT brick* όπου διακρίνονται οι πιο χαρακτηριστικές μονάδες υλικού :



Εικόνα 5: Εσωτερική όψη μονάδων υλικού στο NXT brick [i26]

- **Εσωτερική επικοινωνία μονάδων**

Η εσωτερική επικοινωνία των προαναφερόμενων μονάδων υλικού πραγματοποιείται με συντεταγμένο και συστηματικό τρόπο, όπως φαίνεται στο παρακάτω αφαιρετικό διάγραμμα στο Σχήμα 7 [i14]:



Σχήμα 7: Διάγραμμα μονάδων υλικού και εσωτερικής επικοινωνίας στο NXT brick

3.3.3 Επενεργητές (actuators) και αισθητήρες (sensors)

Το *Lego NXT* παρέχει ένα σύνολο από μηχανισμούς επενεργητών και αισθητήρων. Οι επενεργητές δέχονται ηλεκτρικά σήματα/μηνύματα από το *NXT brick* και τα μετατρέπουν σε κάποια φυσική δράση, ενώ οι αισθητήρες ανιχνεύουν το περιβάλλον και μετατρέπουν τις αντίστοιχες καταμετρήσεις σε ηλεκτρικά σήματα/μηνύματα προς το *NXT brick*.

Οι πιο σημαντικοί από τους μηχανισμούς αυτούς είναι οι εξής [i16] [i17] :

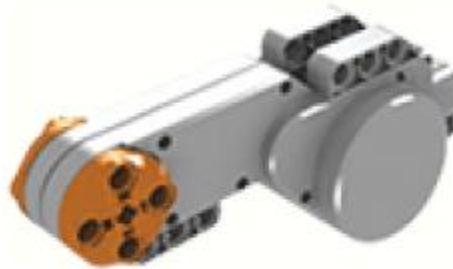
- **Σερβοκινητήρας (servo motor)**

Οι σερβοκινητήρες είναι οι επενεργητές (actuators) που δίνουν στο *Lego NXT* την δυνατότητα της κίνησης και μπορούν να συγχρονίζονται ώστε να αποδίδουν αυτή την κίνηση ομαλά.

Κάθε σερβοκινητήρας έχει εσωτερικά ενσωματωμένο έναν αισθητήρα περιστροφής (rotation sensor), που ανατροφοδοτεί με πληροφορίες το *NXT brick* και επιτρέπει τον έλεγχο της



κίνησης με μεγάλη ακρίβεια. Ο αισθητήρας περιστροφής καταμετρά τις περιστροφές του σερβοκινητήρα σε μοίρες ή σε ολόκληρες περιστροφές, με μία ολόκληρη περιστροφή να είναι 360 μοίρες. Η ταχύτητα του σερβοκινητήρα ελέγχεται από με διαμόρφωση εύρους παλμών (PWM) και εξαρτάται από την μέση τάση που εφαρμόζεται σε αυτόν [b8].



Εικόνα 6: Ο σερβοκινητήρας (servo motor) στο Lego NXT

- **Αισθητήρας επαφής (touch sensor)**

Ο αισθητήρας επαφής είναι ουσιαστικά ένας διακόπτης (switch) μπορεί να ανιχνεύσει πότε πιέζεται και πότε απελευθερώνεται. Έτσι το ρομπότ που διαθέτει αισθητήρα επαφής μπορεί να αντιδρά σε διάφορα σενάρια π.χ. το ρομπότ προγραμματίζεται να σταματάει την κίνησή του όταν ο αισθητήρας επαφής προσκρούει σε κάποιο εμπόδιο.



Εικόνα 7: Ο αισθητήρας επαφής (touch sensor) στο Lego NXT

- **Αισθητήρας υπερήχων (ultrasonic sensor)**

Ο αισθητήρας υπερήχων επιτρέπει την μέτρηση της απόστασης από ένα αντικείμενο και την ανίχνευση κίνησης. Η απόσταση μπορεί να υπολογιστεί σε εκατοστά ή ίντσες και υπάρχει δυνατότητα καταμέτρησης από 0 έως 255 εκατοστά με ακρίβεια +/- 3 εκατοστών. Ο αισθητήρας χρησιμοποιεί την επιστημονική αρχή βάσης της οποίας μία απόσταση καταμετράται υπολογίζοντας τον χρόνο που χρειάζεται ένα υπερηχητικό κύμα να προσκρούσει σε ένα εμπόδιο και να επιστρέψει.



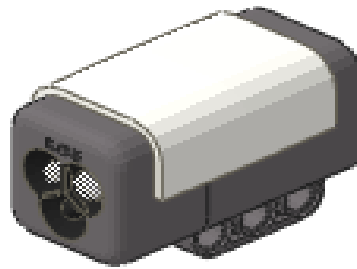
Ο αισθητήρας υπερήχων παρουσιάζει κάποιους λογικούς περιορισμούς: μπορεί να παρουσιάσει δυσκολία στο να ανιχνεύσει αντικείμενα που είναι φτιαγμένα από μαλακό υλικό, έχουν καμπυλώσεις (π.χ. μία μπάλα) ή είναι πολύ μικρά ή λεπτά. Επίσης όταν έχουμε δύο ή περισσότερους αισθητήρες που λειτουργούν στον ίδιο χώρο μπορεί να έχουμε παρεμβολές στην λειτουργία τους.



Εικόνα 8: Ο αισθητήρας υπερήχων (ultrasonic sensor) στο Lego NXT

- **Αισθητήρας χρώματος (color sensor)**

Ο αισθητήρας χρώματος έχει πολλαπλή λειτουργικότητα: μπορεί να διακρίνει το κόκκινο, πράσινο και το μπλε χρώμα (RGB), μπορεί να καταμετρά την ένταση του αντανακλώμενου φωτός επάνω σε επιφάνειες ή μέσα σε ένα δωμάτιο και τέλος μπορεί να λειτουργεί ως RGB Led δηλαδή να εκπέμπει τα τρία χρώματα.



Εικόνα 9: Ο αισθητήρας χρώματος (color sensor) στο Lego NXT

Ο αισθητήρας χρώματος είναι ένας αισθητήρας που αποτελεί εξέλιξη του παλαιότερου αισθητήρα φωτός (light sensor) και παρέχεται με την πιο πρόσφατη έκδοση του Lego MINDSTORMS NXT 2.0.

- **Αισθητήρας ήχου (sound sensor)**

Ο αισθητήρας ήχου ανιχνεύει το επίπεδο της πίεσης του ήχου (sound pressure level) σε μονάδες dB (decibels) και dBA (A-weighted decibels).



- Κατά την ανίχνευση σε dB, οι ήχοι διαφορετικών συχνοτήτων καταγράφονται από τον αισθητήρα με την ίδια ευαισθησία και κατά συνέπεια καταγράφονται ήχοι που μπορεί να έχουν πολύ υψηλές ή χαμηλές συχνότητες για να γίνουν αντιληπτοί από ένα ανθρώπινο αυτί, χωρίς να λαμβάνεται υπόψιν η ψυχοακουστική τους βαρύτητα (με βάση τις καμπύλες ίσης ακουστότητας της ανθρώπινης ακοής).
- Κατά την ανίχνευση σε dBA, ο αισθητήρας είναι προσαρμοσμένος στα επίπεδα ευαισθησίας που έχει το ανθρώπινο αυτί και συνεπώς καταγράφονται μόνο οι ηχητικές στάθμες όπως αυτές γίνονται υποκειμενικά αντιληπτές από το ανθρώπινο αυτί.

Ο αισθητήρας μπορεί να ανιχνεύσει επίπεδα πίεσης ήχου μέχρι 90 dB και οι καταμετρήσεις του αποδίδονται ως ποσοστό αυτού του άνω ορίου. Όσο μικρότερο είναι το ποσοστό που αποδίδεται, τόσο πιο χαμηλός σημαίνει ότι είναι ο προσλαμβανόμενος ήχος. Έτσι με βάση αυτή την αρχή μέτρησης τα παρακάτω ποσοστά αποδίδουν την αντιστοιχία των ποσοστών με γνωστούς ήχους :

- 4-5% : αντιστοιχεί όπως όταν βρισκόμαστε σε ένα ήσυχο δωμάτιο χωρίς θορύβους
- 5-10% : όπως όταν ακούμε κάποιον να μιλάει κανονικά και βρίσκεται σε απόσταση
- 10-30%: σαν να έχουμε κανονική συζήτηση κοντά στον αισθητήρα ή μουσική σε μέσο επίπεδο
- 30-100% : όπως όταν ακούμε ανθρώπους να φωνάζουν ή μουσική να παίζει δυνατά

Θα πρέπει να σημειώσουμε ότι ο αισθητήρας ήχου δεν συμπεριλαμβάνεται στο συνολικό πακέτο που παρέχεται με την αγορά του Lego NXT.



Εικόνα 10: Ο αισθητήρας ήχου (sound sensor) στο Lego NXT



4. Εφαρμοσμένο λογισμικό ανάπτυξης ρομποτικών εφαρμογών

Η όλο και αυξανόμενη διάθεση λογισμικού ανάπτυξης ρομποτικών εφαρμογών, υπογραμμίζει την κινητικότητα και τις ενδιαφέρουσες εξελίξεις που επιφυλάσσει ο χώρος. Υψηλού επιπέδου λογισμικό διατίθεται τόσο σε εμπορική μορφή με προσιτό κόστος, όσο και ελεύθερα σε μορφή ανοικτού κώδικα. Η παρούσα ενότητα πραγματοποιεί μία συνοπτική αναφορά σε συγκεκριμένες εκδόσεις σύγχρονου ρομποτικού λογισμικού και έπειτα αναλύει τα χαρακτηριστικά του Microsoft Robotics Studio το οποίο και αποτελεί την ρομποτική πλατφόρμα λογισμικού για την ανάπτυξη της εφαρμογής φωνητικού ελέγχου.

4.1 Διαθέσιμα προϊόντα ρομποτικού λογισμικού

Υπάρχουν διαθέσιμες αρκετές διαφορετικές εκδόσεις λογισμικού που αφορούν στην ανάπτυξη λογισμικού για ρομποτικά συστήματα οι οποίες έχουν συγκεκριμένη στόχευση και διατίθενται, είτε ως εμπορικά προϊόντα, είτε ως ελεύθερα διαθέσιμο λογισμικό και λογισμικό ανοικτού κώδικα. Στον Πίνακα 5 παραθέτονται κάποια χαρακτηριστικά τέτοια προϊόντα :

Πίνακας 4: Προϊόντα ρομποτικού λογισμικού

Λογισμικό	Τύπος	
Evolution Robotics ERSP	Πλατφόρμα ανάπτυξης	Εμπορικό προϊόν
Microsoft Robotics Studio	Πλατφόρμα ανάπτυξης	Ελεύθερα διαθέσιμο
OROCOS	Βιβλιοθήκες ελέγχου μηχανικών μερών	Λογισμικό ανοικτού κώδικα
URBI	Πλατφόρμα ανάπτυξης	Εμπορικό προϊόν
Webots	Περιβάλλον εξομοίωσης	Εμπορικό προϊόν



Player, Stage, Gazebo	Πλατφόρμα ανάπτυξης	Λογισμικό ανοικτού κώδικα
iRobot AWARE	Πλατφόρμα ανάπτυξης	Εμπορικό προϊόν
OpenJAUS	Πλατφόρμα ανάπτυξης	Λογισμικό ανοικτού κώδικα
CLARAty	Πλατφόρμα ανάπτυξης	Λογισμικό ανοικτού κώδικα

4.2 Microsoft Robotics Developer Studio

Το Microsoft Robotics Developer Studio (Microsoft RDS) αποτελεί την πλατφόρμα ανάπτυξης ρομποτικών εφαρμογών που έχει αναπτύξει η εταιρία Microsoft παρέχοντας δυνατότητα υποστήριξης πλήθους φυσικών ρομποτικών συστημάτων. Βασίζεται επάνω στο λειτουργικό σύστημα Windows και η τελευταία έκδοσή του [i2] απευθύνεται τόσο σε ερασιτέχνες προγραμματιστές, όσο και στην ακαδημαϊκή κοινότητα αλλά και σε επαγγελματίες που ασχολούνται με την ανάπτυξη ρομποτικών εφαρμογών [i1].

Το Microsoft RDS περιλαμβάνει το περιβάλλον εκτέλεσης ασύγχρονων υπηρεσιών, ένα σύνολο εργαλείων ανάπτυξης και προσομοίωσης εφαρμογών, καθώς επίσης και βοηθήματα (tutorials) αλλά και υποδείγματα κώδικα [i9].

Αξίζει να σημειωθεί ότι οι εφαρμογές που κατασκευάζονται με βάση το MRDS κατά κύριο λόγο εκτελούνται μέσω ενός H/Y που διαθέτει λειτουργικό σύστημα Windows και έχει τον έλεγχο του ρομπότ και όχι απευθείας επάνω στο ρομποτικό σύστημα. Για να γίνει απευθείας εκτέλεση θα πρέπει το ρομπότ να έχει αυτούσια επάνω του την δυνατότητα εγκατάστασης του λειτουργικού συστήματος Windows Embedded [b2][i10].

4.2.1 Σύντομο ιστορικό

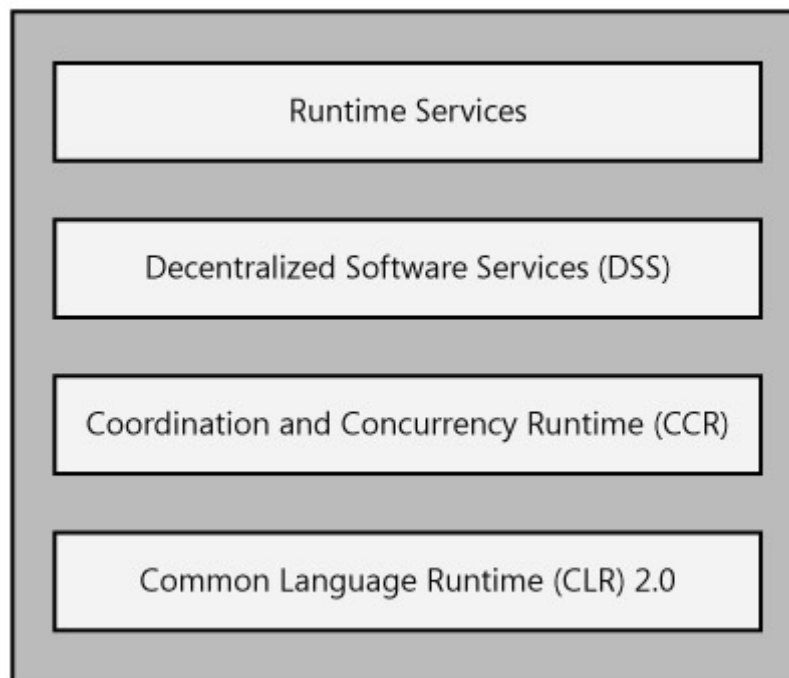
Η ενασχόληση της Microsoft με την ρομποτική ξεκινάει το 2003, καθώς είναι πλέον φανερό ότι υπάρχει διαρκώς αυξανόμενο ενδιαφέρον τόσο από ερευνητικά ιδρύματα και πανεπιστήμια όσο και από εταιρίες κατασκευής ρομποτικών συστημάτων. Δημιουργείται ομάδα μελέτης και ανάπτυξης και τον Δεκέμβριο του 2006 κυκλοφορεί η έκδοση 1.0. Τον



Ιούλιο του 2007 κυκλοφορεί η έκδοση 1.5 και μέχρι τότε η σύντηξη που χαρακτηρίζει το προϊόν είναι MSRS. Τον Απρίλιο του 2008 εκδίδεται μία προ-επισκόπηση (Community Technology Preview) της έκδοσης 2.0 και η σύντηξη του προϊόντος οριστικά γίνεται Microsoft RDS ή MRDS [b2]. Τέλος τον Ιούνιο του 2009 εκδίδεται η τελευταία έκδοση (Microsoft Robotics Developer Studio R2) η οποία και χρησιμοποιείται στην παρούσα εργασία.

4.2.2 Επισκόπηση του Microsoft RDS

Το Microsoft RDS ως πλατφόρμα ανάπτυξης ρομποτικών εφαρμογών χρησιμοποιεί τρεις δομικές μονάδες λογισμικού (software units) που καλούνται αντίστοιχα Decentralized Software Services (DSS), Concurrency and Coordination Runtime (CCR) και .NET Common Language Runtime (CLR), με συγκεκριμένη διαστρωμάτωση (layering) [b1], όπως αυτή εμφανίζεται στο Σχήμα 8 :



Σχήμα 8: Διαστρωμάτωση βασικών μονάδων στο MRDS [b1]

- **Σημασία του ταυτοχρονισμού (concurrency)**

Σε ένα ρομποτικό σύστημα είναι δεδομένο ότι διαφορετικά γεγονότα συμβαίνουν σε πραγματικό χρόνο και κατά την ίδια χρονική στιγμή [b2]. Για παράδειγμα καθώς ένα εκπαιδευτικό ρομπότ Lego NXT κινείται, ο αισθητήρας υπερήχων συλλέγει ταυτόχρονα δεδομένα για την απόσταση του ρομπότ από πιθανά εμπόδια.



Στο παρελθόν για τον χειρισμό αυτού του είδους πολύ-επεξεργασίας (multi-tasking) έπρεπε απαραίτητα να γραφτεί λογισμικό που να αξιοποιεί χαμηλού επιπέδου χαρακτηριστικά του λειτουργικού συστήματος όπως νήματα (OS threads), κρίσιμες περιοχές (critical sections) και σημαφόρους (semaphores). Η αποτύπωση πολύπλοκης λογικής και η αποσφαλμάτωση (debugging) τέτοιων εφαρμογών πραγματικού χρόνου, αποτελούσε πάντοτε μία πολύ χρονοβόρα και επίπονη διαδικασία.

Η μονάδα λογισμικού CCR απαλείφει πολλά από τα παραπάνω προβλήματα, καθώς αναλαμβάνει τις εργασίες του ταυτοχρονισμού και την ευθύνη του χειρισμού των πολύ-νηματικών (multi-threaded) αναγκών πραγματικού χρόνου που εισάγουν οι ρομποτικές εφαρμογές [b1].

- **Ο ρόλος των υπηρεσιών**

Στο MRDS ως υπηρεσία λογισμικού εννοείται η μονάδα υπολογισμού (computational unit) η οποία χαρακτηρίζεται από μία μοναδική ταυτότητα (identity) και έχει κατάσταση (state), συμπεριφορά (behavior) και ένα πλαίσιο λειτουργίας (context) [i11].

Ενώ το CCR αναλαμβάνει την ευθύνη της ταυτόχρονης εκτέλεσης εργασιών επάνω σε έναν μόνο Η/Υ, το DSS προσθέτει ένα ακόμα επίπεδο το οποίο επιτρέπει την κατανεμημένη αξιοποίηση υπηρεσιών μεταξύ πολλών κόμβων(nodes) διαμέσου ενός δικτύου. Έτσι το DSS αναλαμβάνει τόσο την επικοινωνία μεταξύ των υπηρεσιών μέσω μηνυμάτων, αλλά και έχει κομβικό ρόλο στην σύνθεση (partnering) και την ενορχήστρωση τους (orchestration) [b2].

4.2.3 Χαρακτηριστικά πλεονεκτήματα

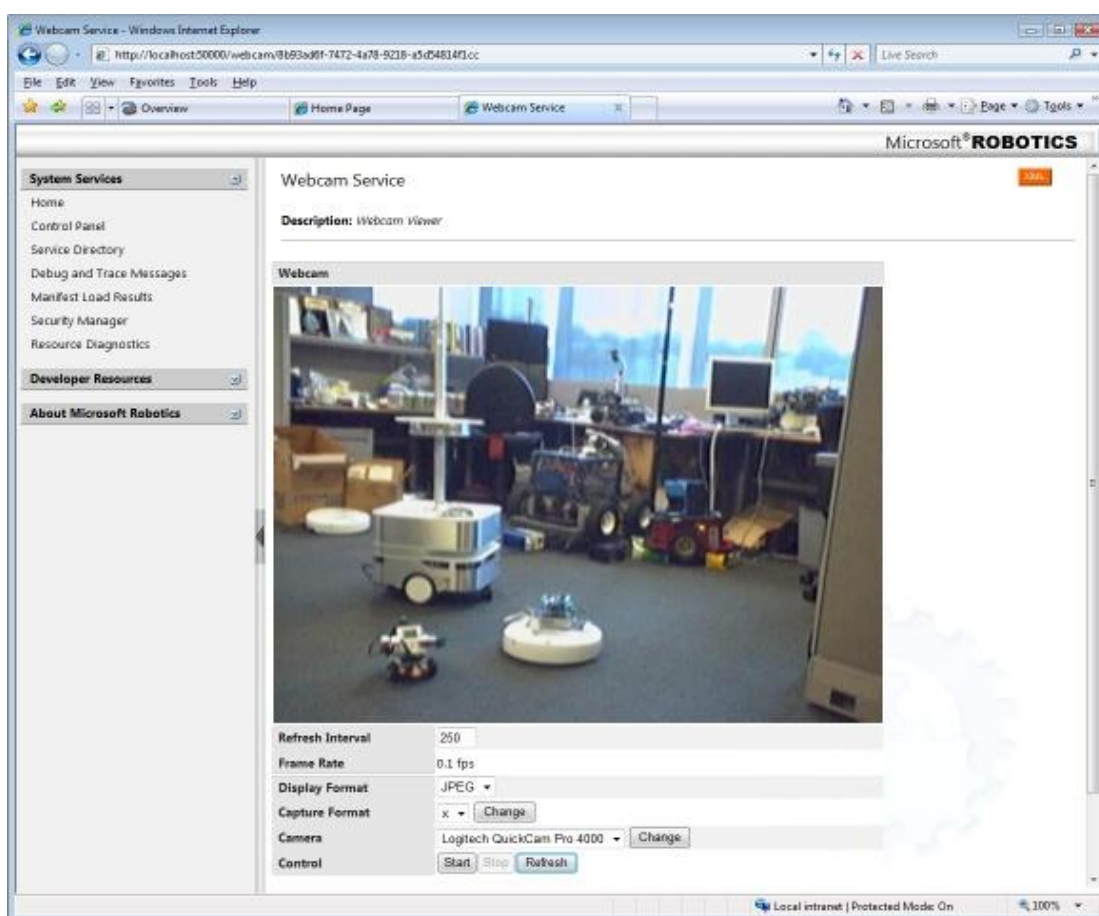
Τα βασικότερα πλεονεκτήματα του Microsoft RDS συνοψίζονται στα εξής [i9]:

- Υλοποιεί το μοτίβο της αρχιτεκτονικής REST και παρέχει ένα περιβάλλον εκτέλεσης βασισμένο στις υπηρεσίες (service-oriented runtime). Κομβικό ρόλο παίζουν οι προαναφερόμενες μονάδες CCR και DSS.
- Το CCR διευκολύνει την συγγραφή προγραμμάτων που αφορούν στον χειρισμό της ασύγχρονης επικοινωνίας και του ταυτοχρονισμού (concurrency) δεδομένων εισόδου/εξόδου σε πραγματικό χρόνο μεταξύ των μηχανισμών ενός ρομποτικού συστήματος.
- Το DSS κάνει ευκολότερη τόσο την παρακολούθηση σε πραγματικό χρόνο των δεδομένων που παρέχονται από τους αισθητήρες ενός ρομποτικού συστήματος,



όσο και τον απομακρυσμένο έλεγχο του συστήματος χρησιμοποιώντας έναν απλό φυλλομετρητή (web browser).

Στην παρακάτω εικόνα φαίνονται από έναν φυλλομετρητή οι εικόνες σε πραγματικό χρόνο που παρέχει μία υπηρεσία web-κάμερας (webcam service) δημιουργημένη με το Microsoft RDS. Το περιβάλλον παρακολούθησης και απομακρυσμένου ελέγχου μέσω web παρέχεται εγγενώς στο Microsoft RDS και δεν χρειάζεται κανενός είδους επέμβαση ή ανάπτυξη από τους τελικούς χρήστες



Εικόνα 11: Παρακολούθηση περιβάλλοντος σε πραγματικό χρόνο μέσω browser

- Έχει κλιμακωτή αρχιτεκτονική (modular architecture) η οποία ενθαρρύνει την δημιουργία σύνθεσης λειτουργιών που αποτελούνται από απλούστερες. (πχ μία υψηλότερου επιπέδου υπηρεσία πλοήγησης ενός ρομπότ, μπορεί να ενσωματώνει μία ή περισσότερες υπηρεσίες χαμηλότερου επιπέδου που επεξεργάζονται δεδομένα από αισθητήρες).



- Παρέχει προγραμματιστική γλώσσα σε γραφικό περιβάλλον (Visual Programming Language, VPL) στην οποία τα συστατικά ενός ρομποτικού συστήματος παρέχονται με σχήματα και συνδέσμους. Έτσι, ακόμα και χρήστες χωρίς υπόβαθρο γνώσεων προγραμματισμού μπορούν να ασχοληθούν με την δημιουργία απλούστερων ή ακόμα και αρκετά σύνθετων εφαρμογών.
- Παρέχει τρισδιάστατο περιβάλλον εξομοίωσης (Visual Simulation Environment - VSE) που βασίζεται επάνω σε προηγμένη μηχανή φυσικών υπολογισμών (physics engine) της εταιρείας AGEIA και δίνει την δυνατότητα για ρεαλιστικές εξομοιώσεις ,τόσο του περιβάλλοντος χώρου, όσο και των ρομποτικών μοντέλων και των αλληλεπιδράσεων μεταξύ τους (Σχήμα 14).



Εικόνα 12: Εικονικό περιβάλλον πόλης δημιουργημένο με το VSE

- Παρέχει ενσωματωμένη υποστήριξη (build-in support) για διάφορους τύπους φυσικών ρομποτικών συστημάτων (πχ LEGO Mindstorms NXT, iRobot Create κτλ). Γενικότερα όμως, αποτελεί μία πλατφόρμα ανάπτυξης λογισμικού για οποιοδήποτε ρομποτικό σύστημα.
- Υποστηρίζει την επικοινωνία μεταξύ H/Y και ρομποτικού συστήματος μέσω διαφόρων καναλιών επικοινωνίας (USB, Bluetooth, WiFi κτλ). Υποστηρίζει επίσης και την απευθείας εκτέλεση των προγραμμάτων σε όσα ρομποτικά



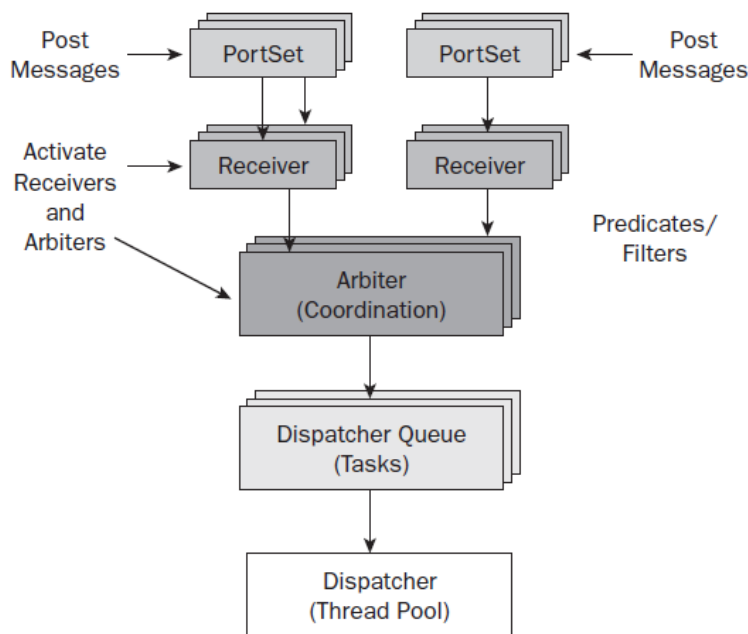
μοντέλα μπορούν να έχουν εγκατεστημένη κάποια από τις ειδικές εκδόσεις του λειτουργικού Windows Embedded.

- Επεκτείνεται εύκολα με προσθήκες βιβλιοθηκών και υπηρεσιών
- Είναι βασισμένο επάνω στην πλατφόρμα .NET η οποία μεταξύ των άλλων δίνει την επιλογή περισσότερων από μίας προγραμματιστικών γλωσσών για τις προς ανάπτυξη ρομποτικές εφαρμογές (Microsoft C++, C#, VB.NET, Microsoft Iron Python)

4.2.4 Concurrency and Coordination Runtime (CCR)

Το CCR είναι μία βιβλιοθήκη που έχει δημιουργηθεί με χρήση του .NET framework (managed code library) και καλύπτει ανάγκες εφαρμογών βασισμένων στις υπηρεσίες (service-oriented applications) που αφορούν στη διαχείριση ασύγχρονων λειτουργιών, στο χειρισμό του ταυτοχρονισμού (concurrency), στην αξιοποίηση παράλληλης επεξεργαστικής ισχύος (parallel hardware) και στο χειρισμό των μερικών αποτυχιών (partial failure) [i9]. Το CCR επιτελεί τις παραπάνω εργασίες μέσα στα πλαίσια μίας διεργασίας (OS process) [b2].

Το παρακάτω σχήμα συνοψίζει την αρχιτεκτονική δομή του CCR :



Σχήμα 9: Αρχιτεκτονική CCR

Οι σημαντικότερες συνιστώσες (components) του CCR είναι οι εξής [b2] :



- Τα *ports* αποτελούν τις δομές δεδομένων που χρησιμοποιεί το CCR για την προσωρινή αποθήκευση μηνυμάτων. Κάθε *port* είναι ουσιαστικά μία ουρά μηνυμάτων (message queue) που βασίζεται στην κλασική αρχή Last-in-First-Out (LIFO). Κάθε μήνυμα στο CCR μπορεί να είναι ένας οποιοσδήποτε απλός ή σύνθετος τύπος δεδομένων που υποστηρίζεται από το .NET CLR (π.χ. integer, string, float αλλά ακόμα και κλάσεις). Κάθε *port* δέχεται μηνύματα ενός και μόνο τύπου. Τα μηνύματα εισάγονται (en-queuing) σε ένα *port* χρησιμοποιώντας προγραμματιστικά την μέθοδο αποστολής που καλείται *post* (posting messages to a port).
- Σε αντίθεση με τα *ports*, τα *portsets* μπορούν να αποθηκεύσουν πολλούς διαφορετικούς τύπους μηνυμάτων. Εσωτερικά κάθε *portset* αποτελείται από ένα σύνολο από διαφορετικές ουρές μηνυμάτων (message queues), δηλαδή από ένα σύνολο από *ports* τα οποία μπορούμε να τα χειριστούμε συνολικά ως μία οντότητα.
- Οι *receivers* είναι υπεύθυνοι για την ανάκτηση και επεξεργασία των μηνυμάτων (message de-queuing) που υπάρχουν στα *ports*. Κάθε *port* μπορεί να έχει πολλούς *receivers* οι οποίοι έχουν την δυνατότητα να ενεργοποιούνται ασύγχρονα κάθε φορά που φτάνει ένα ή περισσότερα μηνύματα. Οι *receivers* επιδέχονται συγκεκριμένες συνθήκες ενεργοποίησης (activation conditions) και έτσι μπορούμε να δημιουργήσουμε σύνθετες λογικές εκφράσεις. Έτσι, όταν χρησιμοποιείται η συνθήκη *Join* μεταξύ δύο *ports*, ο *receiver* περιμένει μέχρι να φτάσουν απαραίτητα δύο μηνύματα, ένα σε κάθε *port*, για να ξεκινήσει την επεξεργασία (κάτι αντίστοιχο με την λογική εντολή AND). Όταν χρησιμοποιείται η συνθήκη *Choice* μεταξύ δύο *ports*, ο *receiver* περιμένει μέχρι να φτάσει ένα μήνυμα σε οποιαδήποτε από τα δύο *ports* (κάτι αντίστοιχο με την λογική εντολή OR).
- Οι *arbiters* είναι καταρχήν επιφορτισμένοι με την εργασία αποτίμησης των προαναφερόμενων συνθηκών ενεργοποίησης. Επιπλέον, υλοποιούν λογική συντονισμού (coordination) και ελέγχου των *tasks*.
- Κάθε *task* περιέχει μία αναφορά (reference) στο κομμάτι του κώδικα που θέλουμε να εκτελεστεί στις διάφορες περιπτώσεις ελέγχου (π.χ. όταν φτάνει ένα μήνυμα σε ένα *port*). Το κομμάτι αυτό του κώδικα στην ορολογία του CCR καλείται *handler*.
- Κάθε φορά που οι συνθήκες ενεργοποίησης ικανοποιούνται, ένα *task* τοποθετείται σε μία ουρά που καλείται *dispatcher queue* και προωθείται τελικά στον *dispatcher* για εκτέλεση. Ο *dispatcher* δηλαδή αναθέτει τα *tasks* στα



νήματα του λειτουργικού συστήματος (IO threads) και φροντίζει για την ισομερή κατανομή τους [b1]. Όταν εκτελείται το *task*, εκτελείται ασύγχρονα και ο αντίστοιχος *handler*. Είναι συνηθισμένο ένας *handler* εκτός των άλλων να στέλνει και καινούργια μηνύματα οπότε ξεκινάει ένας καινούργιος κύκλος επεξεργασίας μηνυμάτων.

- Για τον χειρισμό των αποτυχιών (exceptions) το CCR χρησιμοποιεί ειδικούς δομημένους μηχανισμούς που καλούνται *causalities*.

4.2.5 Decentralized Software Services (DSS)

Το DSS αποτελεί ένα περιβάλλον εκτέλεσης υπηρεσιών (runtime environment) το οποίο έχει δημιουργηθεί με βάση την πλατφόρμα .NET και τοποθετείται αρχιτεκτονικά επάνω από το επίπεδο του CCR. Παρέχει ένα μοντέλο υπηρεσιών το οποίο καταναλώνει λίγους υπολογιστικούς πόρους (lightweight) και συνδυάζει έννοιες από την αρχιτεκτονική REST για την δημιουργία εφαρμογών υψηλής κλιμάκωσης. Ενσωματώνει χαρακτηριστικά όπως την απομόνωση των υπηρεσιών, τον χειρισμό των καταστάσεών τους με δομημένο τρόπο, χειρισμούς συμβάντων και τη δυνατότητα σύνθεσης υπηρεσιών από απλούστερες [i9]. Το DSS επεκτείνει τις δυνατότητες του CCR για παράλληλη επεξεργασία μηνυμάτων καθώς επιτρέπει ασύγχρονη ανταλλαγή μηνυμάτων τόσο μεταξύ των διεργασιών (OS processes) ενός Η/Υ, όσο και μεταξύ κόμβων μέσα σε ένα δίκτυο [b2].

- **Το πρωτόκολλο DSSP**

Το DSSP είναι ένα απλό πρωτόκολλο βασισμένο στο SOAP το οποίο ορίζει ένα μοντέλο υπηρεσιών με κοινές αρχές που αφορούν στην ταυτότητα (identity), την κατάσταση (state) και τις συσχετίσεις (relationships) μεταξύ των υπηρεσιών. Το DSSP ορίζει ένα σύνολο λειτουργιών (operations) βασισμένων σε μηνύματα που εκφράζουν την κατάσταση των υπηρεσιών. Οι λειτουργίες αυτές παρέχουν υποστήριξη για ανάκτηση, χειρισμό και ειδοποιήσεις συμβάντων (event notification) [i11].

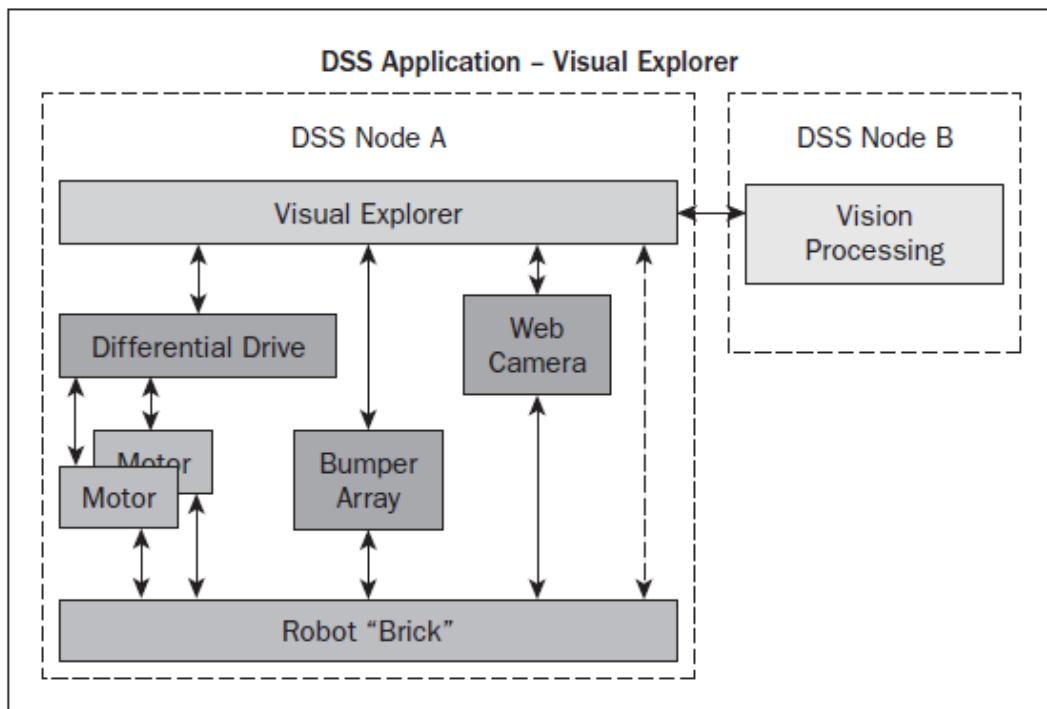
Το DSSP διαφέρει από το πρωτόκολλο SOAP στο ότι διαχωρίζει την κατάσταση μίας υπηρεσίας από την συμπεριφορά (behavior) της και αναπαριστά την πρόσβαση σε μία υπηρεσία σαν ως σύνολο λειτουργιών που επιδρούν στην κατάστασή της [b2].

- **Υπηρεσίες στο DSS**

Η έννοια της υπηρεσίας είναι κομβική τόσο στο ίδιο DSS όσο και γενικότερα στο Microsoft Robotics Studio, καθώς αποτελεί την βασική δομική και λειτουργική μονάδα. Βασικό

χαρακτηριστικό αποτελεί η δυνατότητα της σύνθεσης (partnering) και ενορχήστρωσης (orchestration) υπηρεσιών με σκοπό την δημιουργία μεγαλύτερων εφαρμογών.

Στο παρακάτω σχήμα παρουσιάζεται η διαστρωμάτωση μίας ρομποτικής εφαρμογή η οποία αποτελείται από διάφορων ειδών υπηρεσίες οι οποίες επικοινωνούν μεταξύ τους μέσω μηνυμάτων. Παρατηρούμε ότι η εφαρμογή εκτελείται σε δύο κόμβους (nodes) - τους DSS Node A και DSS Node B αντίστοιχα- οι οποίοι πρακτικά θα μπορούσαν να είναι δύο Η/Υ μέσα σε ένα δίκτυο. Το DSS αναλαμβάνει τόσο την επικοινωνία μεταξύ των υπηρεσιών τόσο στον κόμβο A, όσο και μεταξύ των κόμβων.



Σχήμα 10: Παράδειγμα σύνθεσης και επικοινωνίας υπηρεσιών

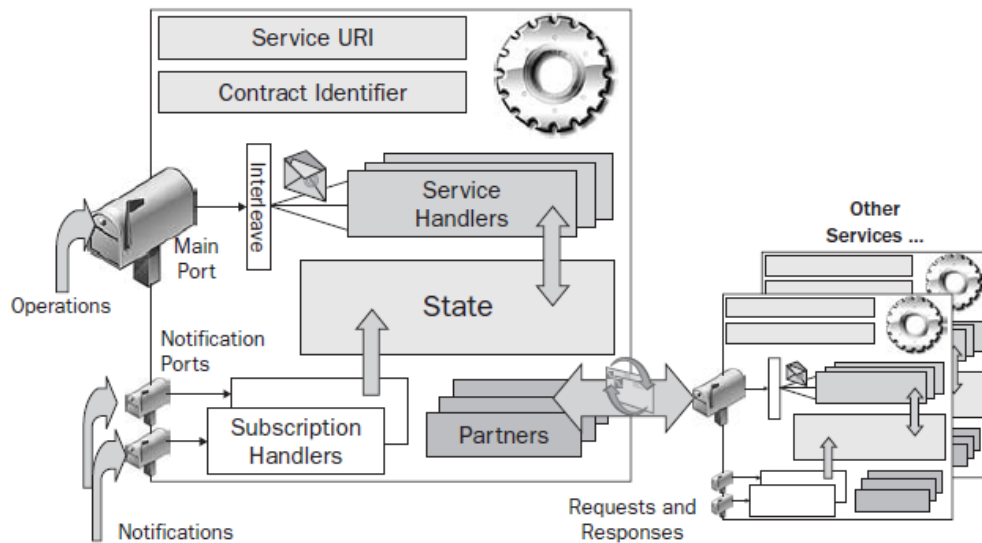
Μία υπηρεσία αποτελείται από τις εξής συνιστώσες (components) :

- *Κατάσταση (state)* : Τα δεδομένα που αναπαριστούν μία υπηρεσία σε μία δεδομένη χρονική στιγμή [i11].
- *Συμπεριφορά (behavior)* : Το σύνολο των λειτουργιών (operations) που μία υπηρεσία μπορεί να επιτελέσει και οι οποίες υλοποιούνται από συγκεκριμένα τμήματα κώδικα (*handlers*) [b2].
- *Πλαίσιο λειτουργίας (context)* : Οι *συσχετισμοί (partnerships)* με άλλες υπηρεσίες και η αρχική κατάσταση της ίδιας της υπηρεσίας [b2].
- *Σύμβαση (contract)* : Περιλαμβάνει τόσο τα μηνύματα που μπορούν να αποσταλούν σε μία υπηρεσία όσο και την μοναδική ταυτότητα της υπηρεσίας (*contract*



identifier) που εκφράζεται με την μορφή ενός Universal Resource Identifier (URI) [b2]

Το σχήμα 11 που ακολουθεί δείχνει τόσο τις συνιστώσες όσο και τον τρόπο πραγματοποίησης της εσωτερικής και εξωτερικής επικοινωνίας μεταξύ των υπηρεσιών.

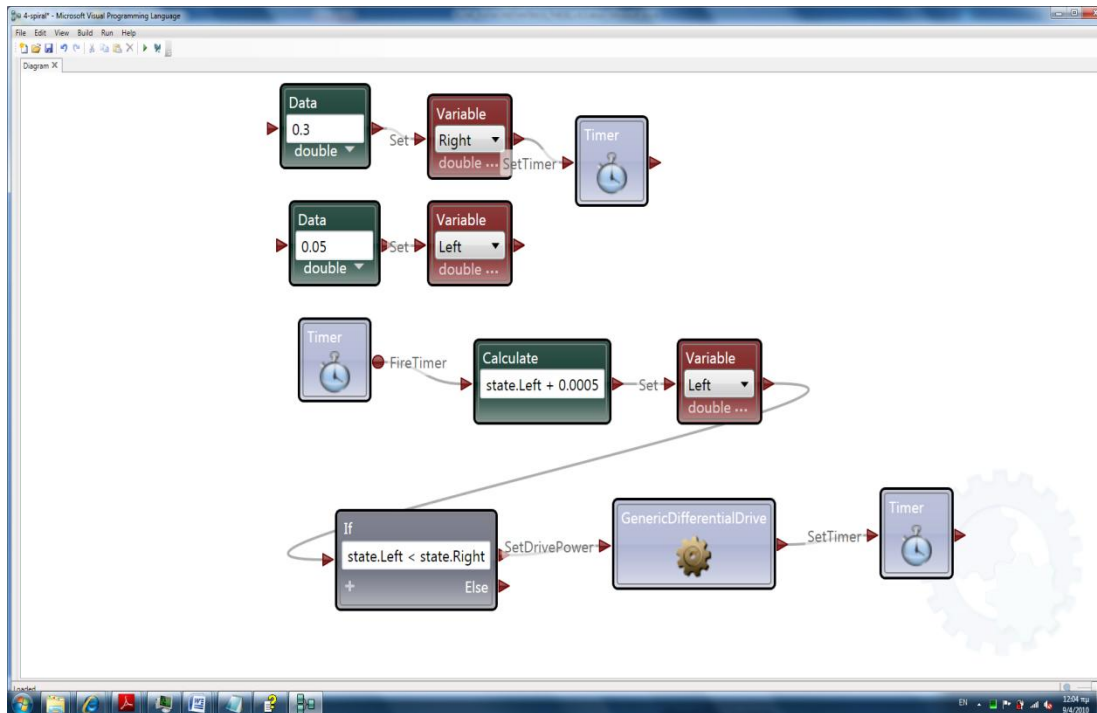


Σχήμα 11: Συνιστώσες μίας υπηρεσίας και τρόποι επικοινωνίας

4.2.6 Visual Programming Language (VPL)

Η VPL αποτελεί ένα περιβάλλον ανάπτυξης εφαρμογών βασισμένο σε ένα γραφικό προγραμματιστικό μοντέλο που επικεντρώνεται σε ροές δεδομένων (dataflow-based programming model). Η VPL είναι έτσι σχεδιασμένη έτσι ώστε να ανταποκρίνεται με ευκολία σε ανάγκες ταυτόχρονης ή και καταμερισμένης επεξεργασίας (concurrent or distributed processing) [19].

Κάθε διάγραμμα ροής σε VPL αποτελείται από δραστηριότητες (activities) οι οποίες παριστάνονται γραφικά ως ορθογώνια τμήματα με συγκεκριμένους χρωματισμούς ανάλογα με τον τύπο της λειτουργίας που επιτελούν. Κάθε δραστηριότητα έχει εισόδους και εξόδους ώστε να μπορεί να λαμβάνει και να αποστέλλει δεδομένα όταν διασυνδέεται με άλλες δραστηριότητες. Οι δραστηριότητες μπορεί πρακτικά να είναι, είτε εντολές ελέγχου ροής και δομές που παρέχονται από την ίδια την VPL, είτε προκατασκευασμένες (pre-built) DSS υπηρεσίες που υπάρχουν στο MRDS, είτε κάποια δραστηριότητα που έχει φτιάξει ο ίδιος ο χρήστης.



Εικόνα 13: Διάγραμμα δραστηριοτήτων σε VPL



5. Ανάπτυξη λογισμικού, κατασκευή και εξομοίωση συστήματος

Το συνολικό σύστημα όπως έχει οριστεί, συνδυάζει την ανάπτυξη λογισμικού (εφαρμογή προσομοίωσης και ελέγχου) και την κατασκευή υλικού (ρομπότ) για την πραγματοποίηση των στόχων της. Στην ενότητα αυτή καταρχήν αιτιολογείται η επιλογή του μοντέλου της εύκαμπτης ανάπτυξης και έπειτα περιγράφονται οι βασικές υποδομές και συνιστώσες λογισμικού που παίρνουν μέρος στην ανάπτυξη της εφαρμογής. Έπειτα, καθορίζονται οι λειτουργικές και μη λειτουργικές απαιτήσεις του όλου συστήματος και δίνεται η αρχιτεκτονική του αποτύπωση μέσα από μία σειρά διαφορετικών όψεων. Οι όψεις αυτές εκφράζουν τις πλέον σημαντικές αποφάσεις που λήφθηκαν σε όλες τις φάσεις της εξέλιξης του και για όλα τα διαφορετικά μέρη του.

5.1 Η εύκαμπτη μεθοδολογία ανάπτυξης

Ο όρος *εύκαμπτη ανάπτυξη* (*agile development*) έχει προέλθει από το μανιφέστο που γράφτηκε το 2001 από μία ομάδα που περιλάμβανε τους δημιουργούς διάφορων ευέλικτων μεθόδων ή συστάσεων ανάπτυξης λογισμικού (όπως Scrum, Extreme Programming κτλ) και είναι γνωστό ως ‘Manifesto for Agile Software Development’ [i31] [i32]. Το μανιφέστο δεν αποτελεί μία μεθοδολογία αλλά έναν όρο - πλαίσιο (*umbrella term*) [i31] που δίνει έμφαση σε τέσσερις βασικές αξίες έναντι κάποιων συμβατικών. Έτσι με βάση το μανιφέστο σημασία έχουν [b16] :

- Πρόσωπα και αλληλεπιδράσεις (*individuals and interactions*) αντί για διεργασίες και εργαλεία (*processes and tools*)
- Λειτουργικό λογισμικό (*working software*) αντί για κατανοητή τεκμηρίωση (*comprehensive documentation*)
- Συνεργασία με τον πελάτη (*customer collaboration*) αντί για διαπραγμάτευση με βάση συμβόλαια (*contract negotiation*)
- Ανταπόκριση στην αλλαγή (*responding to change*) αντί για την παρακολούθηση ενός πλάνου (*following a plan*)

5.1.1 Το πλαίσιο μεθοδολογιών MSF

Το Microsoft Solutions Framework (MSF) αναπτύχθηκε το 1993 ενώ χρησιμοποιήθηκε πρώτη φορά το 1994. Το 2005 εισάγεται η έκδοση 4.0 σε συνεργασία με την αντίστοιχη έκδοση του Visual Studio 2005 [b15]. Το MSF παρέχει ένα αφαιρετικό ‘πλαίσιο’ (*high level framework*) αρχών και οδηγιών που βοηθούν τους κατασκευαστές λογισμικού και τους



επαγγελματίες του χώρου (developers) να αναπτύξουν επιτυχημένο λογισμικό μειώνοντας τα σχετικά ρίσκα και κόστη και αυξάνοντας την ταχύτητα ανάπτυξης.

Το Microsoft Solutions Framework for Agile Software Development (MSF4ASD) είναι μία εκδοχή του MSF η οποία προσεγγίζει την εύκαμπτη ανάπτυξη και καθοδηγείται από σενάρια [b15]. Αποτελείται από τρία βασικά στοιχεία που είναι οι *διαδρομές (tracks)*, οι *ομάδες (advocacy groups)* και οι *κύκλοι (cycles)* [b15] όπου :

- Οι *διαδρομές* ορίζουν τις χρονικές φάσεις που ορίζεται η συνολική διαδικασία
- Οι *ομάδες* ορίζουν συγκεκριμένες αρμοδιότητες και το MSF4ASD ορίζει έξι συγκεκριμένους ρόλους για αυτές τις αρμοδιότητες
- Οι *κύκλοι* ορίζουν την συχνότητα με την οποία συμβαίνουν δραστηριότητες από τις *ομάδες* στα πλαίσια των *διαδρομών*.

Βασικό στοιχείο του MSF4ASD είναι ότι καθοδηγείται κάθε φορά από το πλαίσιο (context-driven) του συστήματος που αναπτύσσεται [b15] και άρα δεν αποτελεί κάποιου είδους φορμαλιστική αποτύπωση αυστηρά καθορισμένων διαδικασιών, παρά προσαρμόζεται ανάλογα με τις αλλαγές του περιβάλλοντος.

5.1.2 Επιλογή πλαισίου μεθοδολογίας

Η εφαρμοσμένη κατεύθυνση που έχει η παρούσα εργασία, οι ανάγκες που προέκυψαν για έρευνα και συνδυασμό διαφόρων συνιστωσών (components) λογισμικού, η εγκαθίδρυση συνεργασιών μεταξύ τους μέσα στο πλαίσιο της εργασίας, ο επανακαθορισμός διαφόρων λογικών, λειτουργικών και τεχνικών χαρακτηριστικών με βάση νέα δεδομένα, αποτέλεσαν παράγοντες ενός πλαισίου στο οποίο οι αλλαγές ήταν ο κανόνας και όχι η εξαίρεση.

Το MSF4ASD πλαισιώνει τις παραπάνω ανάγκες με ικανοποιητικό τρόπο. Λόγω ακριβώς της ευελιξίας που παρέχει το MSF4ASD, η ενσωμάτωση μίας διακριτής διαδικασίας που δεν έχει να κάνει αμιγώς με λογισμικό, όπως ήταν η κατασκευή του φυσικού ρομποτικού συστήματος, έγινε απρόσκοπτα όπως θα δειχθεί παρακάτω.

Με οδηγούς τους έξι ρόλους του MSF4ASD ορίσαμε τις εξής αντιστοιχίες που αφορούν στην συνολική εφαρμογή :

Πίνακας 5: Ρόλοι εργασιών στην ανάπτυξη και κατασκευή του συστήματος

Ρόλος	Πρόσωπο
-------	---------



Διαχειριστής έργου (Project Manager)	Επιβλέπων καθηγητής
Αρχιτέκτονας έργου (Architect)	Φοιτητής
Αναλυτής (Business Analyst)	Φοιτητής
Προγραμματιστής (Developer)	Φοιτητής
Tester	Φοιτητής

Με βάση το MSF4ASD σε ένα πρόσωπο μπορεί να δοθούν περισσότεροι του ενός ρόλοι που αντιστοιχούν σε αρμοδιότητες. Στον Πίνακα 6 φαίνεται ακριβώς αυτή η αντιστοίχιση.

Αμέσως μετά, στον Πίνακα 7, δίνεται η αντιστοίχιση των όρων εισάγει το MSF4ASD στην τρέχουσα εργασία : παρατηρούμε ότι ουσιαστικά υπήρξαν πέντε διαδρομές (φάσεις), σε κάθε μία από τις οποίες εμπλέκονταν περισσότεροι του ενός ρόλοι για την περάτωση κύκλων από δραστηριότητες. Παρατηρούμε ότι όλη αυτή η διαδικασία είναι επαναληπτική (iterative), και κάθε νέα διαδρομή εμπλουτίζεται σε διάφορα επίπεδα από τα δεδομένα της προηγούμενης διαδρομής. Για παράδειγμα, ήδη στην 1^η Διαδρομή που αφορούσε στην ανάλυση της εργασίας πήραν μέρος διάφοροι ρόλοι σε διάφορα επίπεδα. Στην 2^η Διαδρομή που αφορούσε την σχεδίαση εμπλουτίστηκαν και βελτιώθηκαν τα παράγωγα της 1^{ης} Διαδρομής κοκ

Πίνακας 6: Ρόλοι και φάσεις εφαρμογής

	Διαδρομή 1 ^η	Διαδρομή 2 ^η	Διαδρομή 3 ^η	Διαδρομή 4 ^η	Διαδρομή 5 ^η
Ρόλοι					
Διαχειριστής έργου Αναλυτής Προγρ/στής Tester	1^η Εργασία (Ανάλυση) - Διερεύνηση τομέα (domain research) -Ανάλυση - Τεχνική διερεύνηση -Πρωτότυπο κώδικα				



Διαχειριστής έργου Αρχιτέκτονας Προγρ/στής Tester		2^η Εργασία (Σχεδίαση) -Τεχνική διερεύνηση -Αρχιτεκτονική σχεδίαση -Εμπλουτισμός εφαρμογής			
Διαχειριστής έργου Προγρ/στής Tester Αρχιτέκτονας Tester			3^η Εργασία (Ανάπτυξη) - Τεχνική διερεύνηση - Λειτουργική έκδοση εφαρμογής - Σχεδιαστικές βελτιώσεις		
Διαχειριστής έργου Αναλυτής Αρχιτέκτονας Προγρ/στής Tester				4^η Εργασία (Ολοκλήρωση) - Πρόσθετη διερεύνηση τομέα (domain research) - Τεχνική διερεύνηση - Εμπλουτισμός ανάλυσης - Εμπλουτισμός λειτουργιών εφαρμογής	
Διαχειριστής έργου Αναλυτής Αρχιτέκτονας Προγρ/στής Tester					Παράδοση - Ολοκλήρωση λειτουργιών - Επισκόπηση και έλεγχοι - Λογισμικό εγκατάστασης

Τέλος, καθώς το MSF4ASD δεν υπαγορεύει κάποιες συγκεκριμένες αρχιτεκτονικές όψεις αποφασίσαμε μία αρχιτεκτονική αποτύπωση η οποία περιλαμβάνει τέσσερις βασικές απεικονίσεις του συστήματος : την λειτουργική, την λογική, την απεικόνιση της



λειτουργικής του υποστήριξης και τέλος την φυσική απεικόνιση όπως αυτές αναπτύσσονται σε επόμενη ενότητα. Οι όψεις αυτές θεωρούμε ότι είναι ικανές για την τεχνική τεκμηρίωση της εφαρμογής, χωρίς η συντήρησή τους να αποτελεί μία γραφειοκρατική αναγκαιότητα που καθυστερεί και ουσιαστικά εμποδίζει την ανάπτυξη εφαρμογής. Εντάσσονται δηλαδή σε μία εύκαμπτη (agile) κατεύθυνση ανάπτυξης.

5.2 Βασικές υποδομές και συνιστώσες λογισμικού ανάπτυξης

Στην ανάπτυξη της εφαρμογής ελέγχου εκτός από την πλατφόρμα του Microsoft Robotics Studio συμμετέχουν υποδομές αλλά και κάποιες σημαντικές συνιστώσες (components) λογισμικού, χωρίς τις οποίες η περάτωσή της είτε δεν θα ήταν καν δυνατή είτε θα παρουσίαζε πολύ μεγάλες καθυστερήσεις.

5.2.1 Το .NET framework

Το .NET framework είναι ένα αφαιρετικό ‘πλαίσιο’ λογισμικού (software framework) που μπορεί να εγκατασταθεί σε Η/Υ που βασίζονται στο λειτουργικό σύστημα Windows [i29]. Η έκδοση 1.0 πραγματοποιήθηκε το 2003 ενώ η τελευταία έκδοση είναι η 4.0 και πραγματοποιήθηκε το 2010. Περιλαμβάνει δύο βασικές συνιστώσες : μία εκτενή βιβλιοθήκη με βάση την οποία διευθετούνται προγραμματιστικά προβλήματα και ονομάζεται Base Class Library καθώς και μία ιδεατή μηχανή που ονομάζεται Common Language Runtime (CLR) και χειρίζεται την εκτέλεση των εφαρμογών που έχουν γραφτεί για το .NET [i29].

Το .NET framework είναι μία θεμελιώδης υποδομή για την ανάπτυξη σύγχρονων εφαρμογών στα Windows και ως τέτοια δεν αποτέλεσε προϊόν επιλογής για την εφαρμογή μας. Αυτό ωστόσο κάθε άλλο παρά μειονέκτημα αποτέλεσε, καθώς επικέντρωσε την προγραμματιστική προσπάθεια στην διαχείριση της λογικής και όχι στην διαχείριση του υλικού ή σε χαμηλού επιπέδου διαδικασίες, με προφανές τεράστιο κέρδος όσον αφορά τους παράγοντες του χρόνου και της πολυπλοκότητας.

5.2.2 Η γλώσσα C#

Η αντικειμενοστρεφής γλώσσα C# είναι βασισμένη στα μαθήματα που διδάχτηκε ο τομέας ανάπτυξης λογισμικού από διάφορες γλώσσες όπως η C (υψηλή απόδοση), C++ (αντικειμενοστρεφής δομή), Java (αντικειμενοστρεφής δομή, υψηλή ασφάλεια) και Visual Basic (ταχεία ανάπτυξη) [b15]. Το αποτέλεσμα είναι μία εξελισσόμενη γλώσσα προγραμματισμού κατάλληλη για την ανάπτυξη πάσης φύσεως εφαρμογών που έχουν ως κοινό παρονομαστή την υποδομή του λειτουργικού συστήματος Windows.



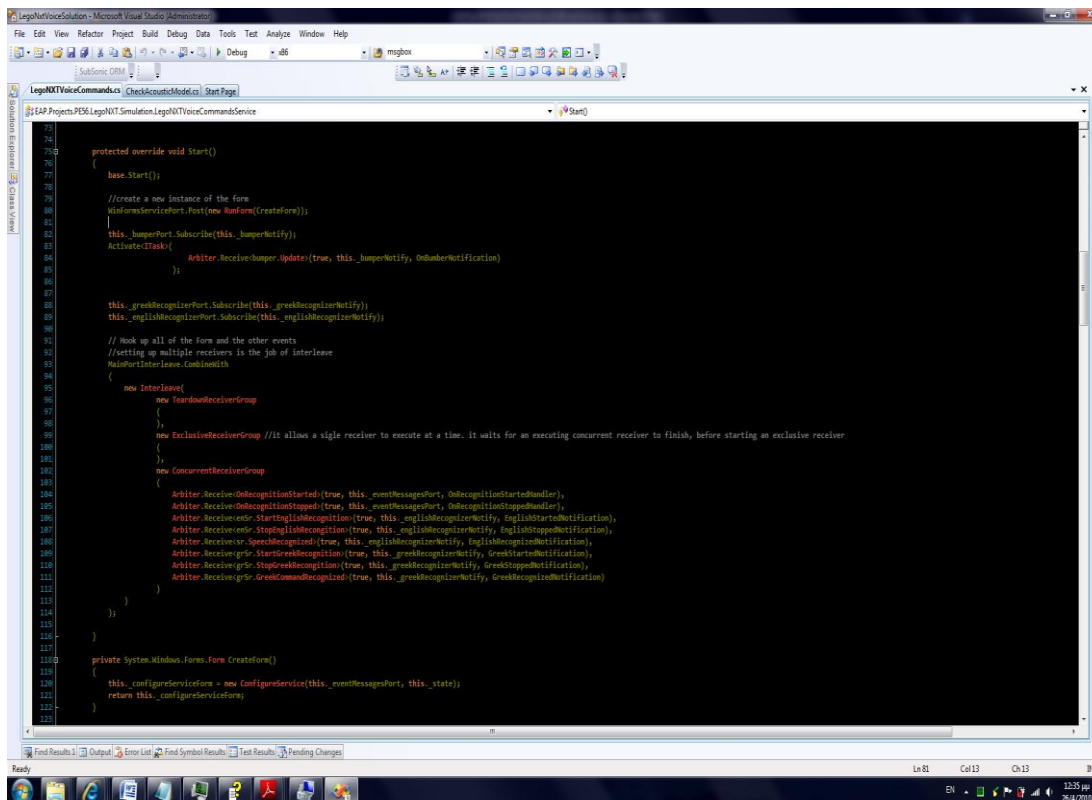
Η γλώσσα C# αρχικά αναπτύχθηκε από μία μικρή ομάδα διακεκριμένων μηχανικών της Microsoft των Anders Hejlsberg and Scott Wiltamuth. Ο πρώτος είναι αρκετά γνωστός για την ανάπτυξη της παλαιότερης επιτυχημένης γλώσσας Turbo Pascal καθώς και για την καθοδήγηση της ομάδας μηχανικών που ανέπτυξε την επίσης επιτυχημένη Borland Delphi [b15].

Η θεμελιώδης C# είναι πολύ απλή με ένα σύνολο με λιγότερες από 100 εντολές και περίπου 12 βασικούς τύπους δεδομένων, αλλά έχει υψηλό βαθμό εκφραστικότητας και υποστηρίζει πλήρως την δομημένη, αντικειμενοστρεφή προγραμματιστική προσέγγιση [b15].

Η ανάπτυξη της εφαρμογής φωνητικού ελέγχου έχει γίνει εξολοκλήρου με την γλώσσα C#.

5.2.3 Το περιβάλλον ανάπτυξης Visual Studio

Το Visual Studio είναι το ενσωματωμένο περιβάλλον ανάπτυξης (Integrated Development Environment) που διαθέτει η εταιρία Microsoft [i30]. Η πρώτη έκδοσή του πραγματοποιήθηκε το 2003, ενώ τη πιο πρόσφατη έκδοση αποτελεί το ‘Visual Studio 2010’ της αντίστοιχης χρονιάς. Μπορεί να χρησιμοποιηθεί για την ανάπτυξη οποιασδήποτε μορφής εφαρμογών λογισμικού που αφορά στις διάφορες εκδόσεις των λειτουργικών συστημάτων Windows όπως εφαρμογές με διεπαφές που εγκαθίστανται απευθείας στο λειτουργικό (windows forms), ιστότοπους και web εφαρμογές (web sites and web applications), εφαρμογές υπηρεσιών web (web services), ρομποτικές εφαρμογές κτλ. Διαθέτει εγγενώς (build in) εξελιγμένα εργαλεία σχεδίασης, συγγραφής και αναδιοργάνωσης (refactoring) κώδικα καθώς και εργαλεία αποσφαλμάτωσης (debugging tools) και μπορεί να υποστηρίξει διάφορα συστήματα ελέγχου και εκδόσεων κώδικα (source control systems)[i30]. Τέλος, μπορεί να υποστηρίξει την ανάπτυξη σε διάφορες γλώσσες προγραμματισμού όπως η Visual C++, C#, Visual Basic, F# κτλ. Στην Εικόνα 15 πιο κάτω, βλέπουμε μία άποψη της έκδοσης με την οποία έχει γίνει η ανάπτυξη της εφαρμογής φωνητικού ελέγχου :



Εικόνα 14 : Ανάπτυξη κώδικα μέσω του Visual Studio 2008 R2

Εάν το Visual Studio είναι ήδη εγκατεστημένο, τότε με την εγκατάσταση της ρομποτικής πλατφόρμας του MRDS ενσωματώνονται στο Studio και οι κατάλληλες συνιστώσες (υπό μορφή plug-ins) που επιτρέπουν την ανάπτυξη ρομποτικών εφαρμογών.

Με βάση τα παραπάνω, το Visual Studio αποτέλεσε μία ‘φυσική’ επιλογή για την ανάπτυξη της εφαρμογής φωνητικού ελέγχου με βάση την έκδοση ‘Visual Studio 2008 R2’.

5.2.4 Η μηχανή αναγνώρισης Julius

Το λογισμικό ανοικτού κώδικα (open source) Julius [i20] περιγράφεται ως ένας υψηλής απόδοσης και μεγάλων λεξιλογίων αποκωδικοποιητής αναγνώρισης φωνής (Large Vocabulary CSR Engine). Βασίζεται σε υλοποιήσεις του μαθηματικού μοντέλου κρυμμένων αλυσίδων Markov (Hidden Markov Models - HMM) που είναι εξαρτημένες από το περιεχόμενο (context dependent HMM) και μπορεί να πραγματοποιήσει αποκωδικοποίηση σε πραγματικό χρόνο [i20].

Αποτελεί μία από τις πλέον σημαντικές συνιστώσες ολόκληρης της εφαρμογής φωνητικού ελέγχου, καθώς είναι η μηχανή αναγνώρισης φωνής που χρησιμοποιείται στο ακουστικό μοντέλο των ελληνικών.



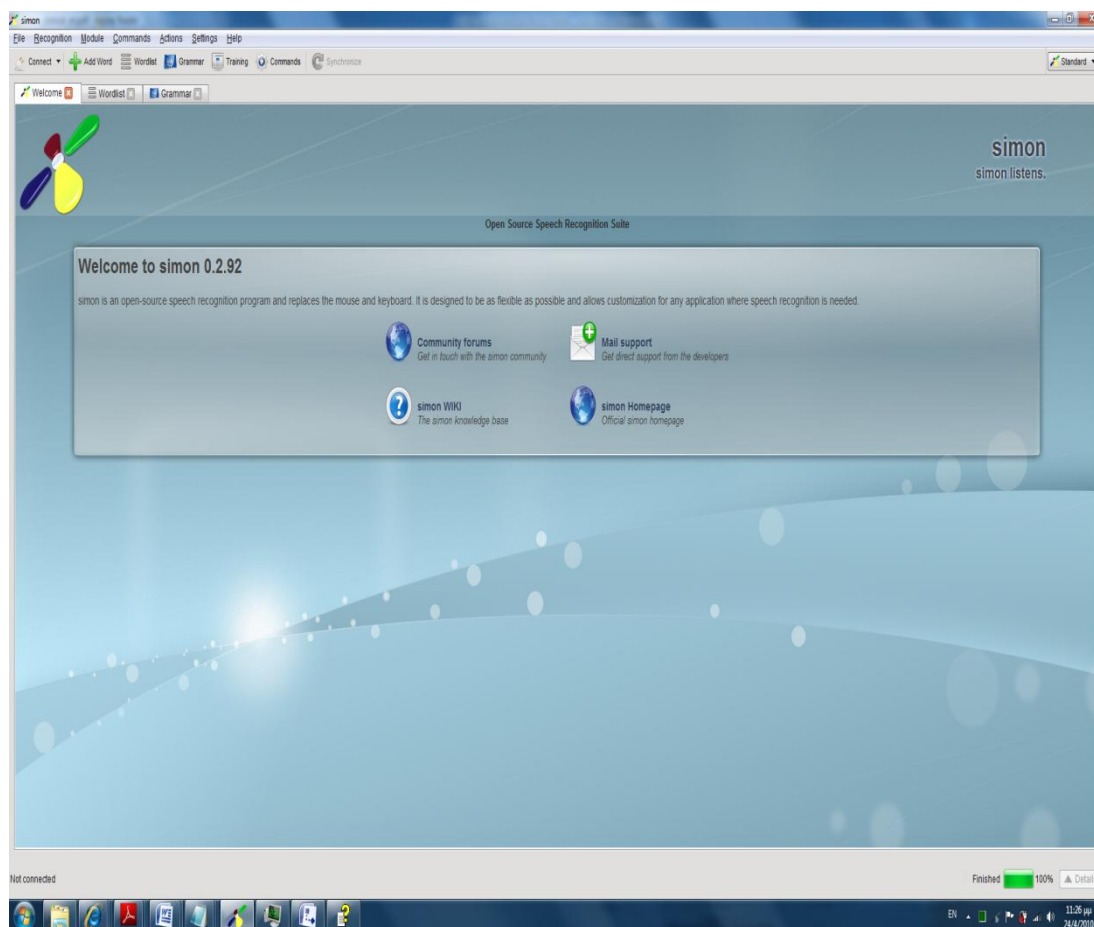
5.2.5 Η συνιστώσα Conjura

Η συγκεκριμένη συνιστώσα αποτελεί ένα API γραμμένο σε Visual C++ και αποτελεί μέρος του ερευνητικού έργου ανοικτού κώδικα Conjura [i19] με αρχικό σκοπό την αναγνώριση φωνής στα πορτογαλικά όπως ομιλούνται στην Βραζιλία, χωρίς ωστόσο να υπάρχει δέσμευση με συγκεκριμένη γλώσσα. Χρησιμοποιείται ως προγραμματιστικό ‘κάλυμμα’ (wrapper) του αποκωδικοποιητή Julius ώστε αυτός να μπορεί να κληθεί με σχετικά μικρή προγραμματιστική προσπάθεια από εφαρμογές βασισμένες σε .NET. Κάτω από αυτό ακριβώς το πλαίσιο χρησιμοποιείται και από την εφαρμογή φωνητικής καθοδήγησης.

5.2.6 Simon : ανοικτό λογισμικό αναγνώρισης φωνής

Το λογισμικό ανοικτού κώδικα Simon αποτελεί ένα πρόγραμμα αναγνώρισης φωνής, με κύριο στόχο την παροχή υπηρεσιών φωνητικής αναγνώρισης σε ανθρώπους με μειωμένες κινητικές δυνατότητες [i28]. Το λογισμικό είναι μεν αξιόλογο και λειτουργικό, αλλά τα χαρακτηριστικά του είναι ακόμα υπό διαμόρφωση ενώ υπάρχουν και θέματα προς επίλυση (beta release).

Ειδοποιός διαφορά από άλλες εφαρμογές της αγοράς, εκτός από την δυνατότητα ελεύθερης και ανοιχτής διάθεσης του, αποτελεί η εξατομικευμένη προσέγγιση στην αναγνώριση φωνής. Το λογισμικό Simon δεν έρχεται με κανενός είδους προ-εγκατεστημένο ή προ-εκπαιδευμένο ακουστικό μοντέλο [i28]. Αντίθετα παρέχει μία φιλική και αρκετά διαισθητική διεπαφή χρήσης (user interface), με την οποία ο κάθε ενδιαφερόμενος μπορεί να δημιουργήσει εύκολα τόσο το γλωσσικό όσο και το ακουστικό μοντέλο με βάση την θεματολογία που τον ενδιαφέρει. Έτσι, υπάρχει μία ενσωματωμένη υποδομή δημιουργίας σεναρίων όπου οι χρήστες και ιδιαίτερα τα άτομα με κινητικά προβλήματα μπορούν να παραμετροποιούν σύνολα φωνητικών εντολών που έχουν δημιουργήσει οι ίδιοι ανάλογα με τις ανάγκες τους.



Εικόνα 15: Το ανοιχτό λογισμικό αναγνώρισης φωνής Simon

Το πρόγραμμα Simon παραπέμπει σε αρχιτεκτονική πελάτη-διακομιστή (client-server) όπου εξ' ορισμού και τα δύο μέρη εγκαθίστανται στον ίδιο Η/Υ, ενώ ο τελικός χρήστης χρησιμοποιεί την διεπαφή της εφαρμογής πελάτη όπως φαίνεται και στην Εικόνα 16. Η τρέχουσα έκδοση είναι η 0.2.92.

- **Χρήση στην εφαρμογή**

Το πρόγραμμα Simon αποτελεί το λογισμικό με βάση το οποίο γίνεται η δημιουργία και η εκπαίδευση του ακουστικού μοντέλου της εφαρμογής αναγνώρισης εντολών για τα ελληνικά.

- **Περιορισμοί**

Το ότι το πρόγραμμα Simon στην τρέχουσα έκδοση βρίσκεται σε λειτουργικό αλλά σχετικά πρόωρο στάδιο ανάπτυξης (beta release), αποτέλεσε πηγή περιορισμών αλλά και προβλημάτων. Για την δημιουργία και τις πρώτες εκδόσεις του ακουστικού μοντέλου της εφαρμογής χρησιμοποιήθηκε η έκδοση 0.2 της οποίας η σταθερότητα δεν αποτελούσε



δεδομένο χαρακτηριστικό. Επιπλέον, ένα από τα (φαινομενικά αξεπέραστα) προβλήματα που παρουσιάστηκαν ήταν η αδυναμία συνεργασίας του λογισμικού με τους ελληνικούς χαρακτήρες. Έτσι, ανεξάρτητα με την κωδικοποίηση των χαρακτήρων (UTF-8, Windows 1253 κτλ) όπου υπήρχαν στην *γραμματική* ή στο *λεξικό προφοράς* (*pronunciation dictionary*) ελληνικοί χαρακτήρες, το λογισμικό Simon παρουσίαζε προγραμματιστικά λάθη (exceptions) και η διαδικασία δημιουργίας του ακουστικού μοντέλου δεν μπορούσε να ξεκινήσει.

- **Υπερνίκηση περιορισμών και συμβάσεις**

Για να ξεπεραστεί το παραπάνω πρόβλημα, οι ελληνικές καταχωρήσεις τόσο στο *λεξικό προφοράς* όσο και στην *γραμματική* διαγράφηκαν και έγινε επανα-εισαγωγή τους με χρήση λατινικών χαρακτήρων (greeklish). Αν και κάτι τέτοιο μπορεί αρχικά να φαίνεται παράδοξο, γίνεται γρήγορα αντιληπτό ότι δεν έχει καμία σημασία. Κάθε *μηχανή αναγνώρισης* επί της ουσίας δεν δίνει καμία σημασία στους χαρακτήρες, παρά στους διακριτούς ήχους που αποτελούν κάθε λέξη δηλαδή στα φωνήματα αλλά και την προσωδιακή πληροφορία που αυτά εισάγουν. Έτσι στην περίπτωση μας, επειδή τα ελληνικά αποτελούν την μητρική γλώσσα των ομιλητών που έλαβαν μέρος στην δημιουργία του ακουστικού μοντέλου, στοιχεία όπως η προφορά των φωνημάτων, η εκφορά των λέξεων, η ακολουθία των ήχων κτλ, εγγυώνται την σωστή αποτύπωση των ηχητικών πληροφοριών και άρα την αναγνώριση των εντολών που δίνονται στα ελληνικά.

Συνέπεια του παραπάνω περιορισμού φυσικά είναι ότι όταν οι φωνητικές εντολές αναγνωρίζονται, η οπτική απόδοσή τους στους χρήστες γίνεται με λατινικούς χαρακτήρες (greeklish).

5.2 Απαιτήσεις συστήματος

Στην ενότητα αυτή παραθέτονται οι λειτουργικές και μη-λειτουργικές απαιτήσεις του συστήματος :

5.2.1 Λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις σχετίζονται με τις εργασίες οι οποίες πρέπει να εκτελούνται και καθορίζουν πλήρως την συμπεριφορά του συστήματος. Εκφράζουν έτσι τα επιθυμητά αποτελέσματα που το σύστημα πρέπει να παράγει και την απόκριση που πρέπει να εμφανίζει όταν ισχύουν συγκεκριμένες συνθήκες [b10].



- **Σύνοψη απαιτήσεων**

Οι γενικές απαιτήσεις αφορούν στην κατασκευή ενός ρομποτικού συστήματος το οποίο ελέγχεται απομακρυσμένα μέσω συγκεκριμένων φωνητικών εντολών που μπορούν να δοθούν στα ελληνικά ή στα αγγλικά. Συνεπακόλουθα απαιτείται αφενός η ανάπτυξη της κατάλληλης εφαρμογής φωνητικού ελέγχου του ρομπότ στο πραγματικό περιβάλλον, αλλά και η δυνατότητα πλήρους εξομοίωσης και ελέγχου του σε εικονικό περιβάλλον.

- **Αναλυτική περιγραφή**

Η ανάλυση των απαιτήσεων της συνοπτικής περιγραφής δίνεται στον παρακάτω πίνακα:

Πίνακας 7: Περιγραφή απαιτήσεων

A/A	Περιγραφή
R1	Το σύστημα αναγνώρισης φωνής περιλαμβάνει όλες τις υποδομές για την αναγνώριση συγκεκριμένων προτάσεων τόσο στα αγγλικά όσο και στα ελληνικά. Όσον αφορά τα αγγλικά οι υποδομές αυτές είναι ενσωματωμένες στα Windows και χρησιμοποιούν ως προγραμματιστική διεπαφή το <i>SAPI</i> . Τα Windows, δεν παρέχουν φωνητική υποστήριξη για τα ελληνικά και οι υποδομές αυτές πρέπει να κατασκευαστούν. Συγκεκριμένα, πρέπει με την κατάλληλη επιλογή εργαλείων να κατασκευαστεί το <i>λεξικό</i> , η <i>γραμματική</i> και το <i>ακουστικό μοντέλο</i> για την αναγνώριση συγκεκριμένων φωνητικών εντολών στα ελληνικά. Ειδικά για το <i>ακουστικό μοντέλο</i> απαιτείται και η επαναλαμβανόμενη εκπαίδευσή του με βάση μία ή περισσότερες φωνές ώστε να υπάρχει ικανοποιητική φωνητική αναγνώριση στα ελληνικά.
R2	Η κατασκευή του φυσικού ρομπότ πρέπει να γίνει με χρήση συνδυαζόμενου υλικού του μοντέλου Lego NXT 2.0. Η κατασκευή πρέπει χαρακτηρίζεται από σχετική απλότητα, συγκεκριμένη δομή που να υπόκειται σε τυποποίηση ενός αριθμού βημάτων και ευκολία στην επανασυναρμολόγηση ώστε να είναι δυνατόν σε κάποιον τρίτο μελετητή να την επαναλάβει σε σύντομο χρονικό διάστημα.
R3	Κατά την εκκίνηση της εφαρμογής ο χρήστης επιλέγει εάν θα εργαστεί στο περιβάλλον προσομοίωσης ή στο πραγματικό περιβάλλον με το φυσικό μοντέλο Lego NXT.
R4	Ο χρήστης εκφωνεί τις εντολές ελέγχου μέσω ενός απλού μικροφώνου που συνδέεται με τον H/Y.
R5	Το περιβάλλον προσομοίωσης είναι κάποιος εσωτερικός ή εξωτερικός τρισδιάστατος χώρος που αναπαριστάται στον H/Y, όπου το εικονικό ρομπότ Lego



	NXT είναι ευδιάκριτο και η φορά κινήσεων σαφής κατά την εκτέλεση των φωνητικών εντολών.
R6	Τόσο κατά την προσομοίωση όσο και κατά την εκτέλεση στο πραγματικό περιβάλλον, παρέχεται φιλική διεπαφή στον χρήστη (friendly user interface) ώστε να μπορεί να ξεκινήσει και να παύσει την αναγνώριση φωνητικών εντολών.
R7	Τόσο κατά την προσομοίωση όσο και κατά την εκτέλεση στο πραγματικό περιβάλλον, μέσω της διεπαφής που περιγράφει η απαίτηση R6 ο χρήστης επιλέγει την γλώσσα που θα εκφωνήσει τις εντολές (ελληνικά ή αγγλικά).
R8	Οι φωνητικές εντολές που δίνονται τόσο στα ελληνικά όσο και στα αγγλικά αποτελούν απλές προτάσεις που περιγράφουν σύντομα πως πρέπει να κινηθεί το ρομπότ (πχ ‘Προχώρα μπροστά’, ‘Στρίψε δεξιά’ κ.τ.λ.)
R9	Το ποσοστό επιτυχίας s μίας φωνητικής εντολής θεωρείται μέσο όταν ισχύει $0,4 \leq s \leq 0,7$. Σε αυτή την περίπτωση όσον αφορά στα αγγλικά και μόνο, επιστρέφεται στον χρήστη φωνητική απάντηση από την εφαρμογή ότι η εντολή δεν αναγνωρίστηκε. Όταν ισχύει ότι $s < 0,4$ η φωνητική εντολή αγνοείται. Για τα ελληνικά δεν υποστηρίζεται Text-to-Speech υπηρεσία.

5.2.2 Μη λειτουργικές απαιτήσεις

Τα χαρακτηριστικά του συστήματος που δεν αφορούν στην εκτέλεση λειτουργιών από αυτό, αποτελούν τις μη λειτουργικές απαιτήσεις του [b10].

- **Απαιτήσεις περιβάλλοντος λογισμικού**

Οι απαιτήσεις που αφορούν το περιβάλλον του λογισμικού περιγράφουν όλες τις υποδομές ή τις συνιστώσες (components) που απαιτούνται για την ομαλή του λειτουργία.

Πίνακας 8: Μη λειτουργικές απαιτήσεις περιβάλλοντος λογισμικού

Λειτουργικό σύστημα (OS)	Windows Vista 32/64bit Windows 7 32/64bit
Προαπαιτούμενο λογισμικό	
Ρομποτική πλατφόρμα	Microsoft Robotics Studio Express Edition 2008 R2 ή
Πλατφόρμα υποδομής	.NET framework 3.5 SP1
Λογισμικό δημιουργίας και εκπαίδευσης ακουστικού μοντέλου	Simon 0.2.92
Προγραμματιστική διεπαφή	SAPI (ενσωματωμένο λογισμικό στο OS)



(API) για τα αγγλικά	
Αποκωδικοποιητής αναγνώρισης φωνής για τα αγγλικά	Windows Speech Recognition (ενσωματωμένο λογισμικό στο OS)
Προγραμματιστική διεπαφή (API) για τα ελληνικά	Coruja [i19] (λογισμικό ανοικτού κώδικα)
Αποκωδικοποιητής αναγνώρισης φωνής για τα ελληνικά	Julius [i20] (λογισμικό ανοικτού κώδικα)
Φυλλομετρητής (Web Browser)	Mozilla Firefox 2.0 η ανώτερος
	Microsoft Internet Explorer 7.0 ή ανώτερος

- **Απαιτήσεις επιδόσεων**

Ο χρόνος απόκρισης τόσο του εξομοιωμένου όσο και του πραγματικού ρομπότ στις φωνητικές εντολές πρέπει να είναι μικρότερος του ενός δευτερολέπτου (<1 sec).

- **Απαιτήσεις ανάπτυξης λογισμικού**

Η ανάπτυξη του λογισμικού για την εφαρμογή ελέγχου απαιτείται να γίνει με βάση τα εξής :

Πίνακας 9: Μη λειτουργικές απαιτήσεις ανάπτυξης

Περιβάλλον ανάπτυξης	Microsoft Visual Studio 2008 SP1
Αντικειμενοστρεφής γλώσσα προγραμματισμού	C#

- **Απαιτήσεις επικοινωνίας**

Ο φωνητικός έλεγχος πραγματοποιείται μέσω καναλιού ασύρματης επικοινωνίας Bluetooth μεταξύ του Η/Υ και του ρομπότ. Δηλαδή ο χρήστης εκφωνεί τις εντολές από το μικρόφωνο που συνδέεται με τον Η/Υ και ο Η/Υ αποστέλλει τα κατάλληλα σήματα ελέγχου μέσω του πρωτοκόλλου Bluetooth στο ρομπότ. Αν και η απόσταση ελέγχου η οποία υποστηρίζεται από το πρωτόκολλο Bluetooth είναι σχετικά μικρής κλίμακας, ωστόσο κρίνεται ικανή για την περίπτωση των σεναρίων που θα υλοποιηθούν

- **Συστάσεις υλικού**

Το περιβάλλον προσομοίωσης του MRDS χρησιμοποιεί εσωτερικά μηχανή αναπαράστασης φυσικών φαινομένων (physics engine) η οποία έχει κάποιες απαιτήσεις πόρων. Ο παρακάτω πίνακας περιγράφει κάποιες συστάσεις και όχι απαιτήσεις, με βάση τις οποίες εγγυημένα μπορούμε να έχουμε καλή απόδοση της εφαρμογής σε περιβάλλον προσομοίωσης. Παρόλα



αυτά και με υλικό που δεν πληροί αυτές τις συστάσεις η εφαρμογή είναι πλήρως λειτουργική.

Πίνακας 10: Συστάσεις υλικού

Επεξεργαστής (processor)	Intel Core 2 Duo 2.53 GHz ή ανώτερος
Μνήμη (RAM)	2 GB Ram ή ανώτερη
Κάρτα γραφικών	NVIDIA GeForce 9600 ή ανώτερη

5.3 Αρχιτεκτονική αποτύπωση συστήματος

Η αρχιτεκτονική εφαρμογών είναι η διαδικασία του καθορισμού μίας δομημένης λύσης η οποία ικανοποιεί όλες τις τεχνικές και λειτουργικές απαιτήσεις, ενώ παράλληλα βελτιστοποιεί συνήθη ποιοτικά χαρακτηριστικά όπως η απόδοση, η ασφάλεια και η διαχειρισσιμότητα. Περιλαμβάνει όλες τις σημαντικές αποφάσεις που παίρνονται σχετικά με μία ευρεία περιοχή παραγόντων όπως η συμπεριφορά, η δομή, η υλοποίηση και η εγκατάσταση μίας εφαρμογής, όπου κάθε τέτοια απόφαση έχει σημαντικό αντίκτυπο στην επιτυχία της τελικής εφαρμογής [i3].

Οι απαιτήσεις της εργασίας αφορούν στη συναρμολόγηση των ρομποτικών μερών, την κατασκευή των υποδομών για την αναγνώριση των εντολών στα ελληνικά και την ανάπτυξη του κατάλληλου λογισμικού φωνητικού ελέγχου. Προς ικανοποίηση των απαιτήσεων αυτών που αναφέρονται και σε υλικό και σε λογισμικό, έχει πραγματοποιηθεί μία σύνθεση χαρακτηριστικών από μεθοδολογίες αρχιτεκτονικής αποτύπωσης [b11] [i3] ώστε να φαίνονται οι περισσότερο επεξηγηματικές για το έργο αρχιτεκτονικές όψεις.

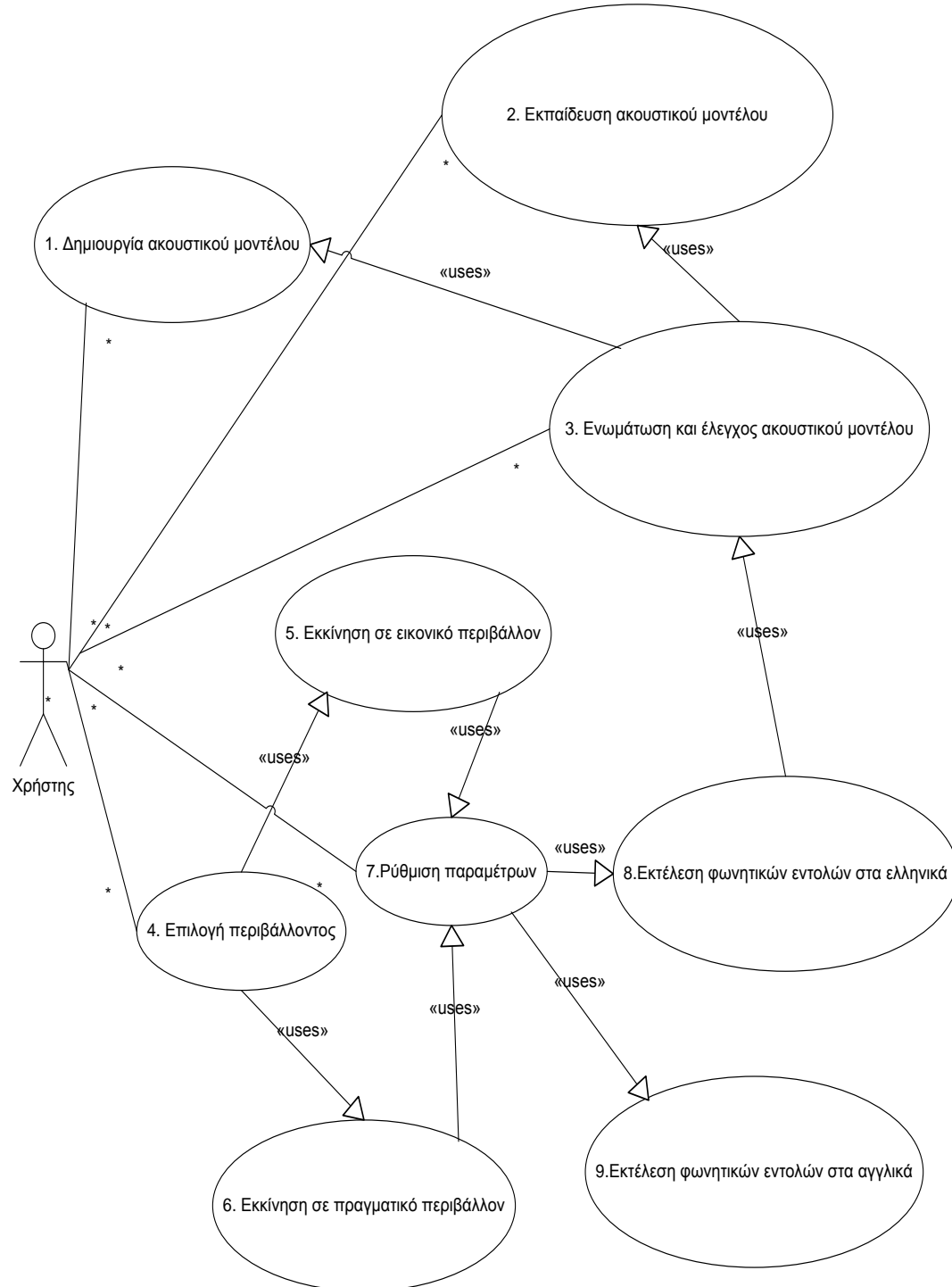
Για την παρουσίαση των πλέον σημαντικών αποφάσεων παρουσιάζεται μία σειρά από διαφορετικές αρχιτεκτονικές απεικονίσεις, που σκοπό έχουν την επισκόπηση του συστήματος κάθε φορά από διαφορετική οπτική και τελικά την πληρέστερη κατανόησή της δομής και των λειτουργιών του.

5.3.1 Λειτουργική απεικόνιση (*functional view*)

Η λειτουργική απεικόνιση παρουσιάζει την εφαρμογή από την σκοπιά των λειτουργιών που αυτή επιτελεί. Οι λειτουργίες αυτές δίνονται καταρχήν με την μορφή σεναρίων χρήσης (Use Cases) τα οποία και προσδιορίζουν την συμπεριφοράς της εφαρμογής μέσω ενός συνόλου διαδοχικών ενεργειών.

- **Διάγραμμα βασικών σεναρίων χρήσης**

Το παρακάτω διάγραμμα παρουσιάζει με αφαιρετικό τρόπο τα βασικά σεσάρια της εφαρμογής.



Σχήμα 12: Διάγραμμα βασικών σεναρίων χρήσης



- **Περιγραφή σεναρίων χρήσης**

Ακολουθεί αναλυτικότερη περιγραφή των σεναρίων χρήσης.

Σενάριο Χρήσης 1

1.Τίτλος

Δημιουργία ακουστικού μοντέλου

2.Συνοπτική Περιγραφή

Ο χρήστης μέσα από κατάλληλη εξωτερικό λογισμικό (third-party software), δημιουργεί το ακουστικό μοντέλο που αφορά στα ελληνικά ώστε να μπορεί να πραγματοποιηθεί αναγνώριση ελληνικών εντολών.

3.Προυποθέσεις

3.1 Έχει δημιουργηθεί κατάλληλο *λεξικό προφοράς (pronunciation dictionary)* για τις ελληνικές εντολές.

3.2 Έχει εγκατασταθεί το εξωτερικό λογισμικό (third-party software) με βάση το οποίο έχει γίνει η εισαγωγή (import) του *λεξικού προφοράς*, η κατασκευή της *γραμματικής (grammar)* και η εκπαίδευση του μοντέλου.

3.3 Έχει γίνει επιλογή ενός τυπικά κατάλληλου χώρου για ηχογράφηση (πχ ήσυχο δωμάτιο)

3.4 Έχει συνδεθεί στον Η/Υ κατάλληλο μικρόφωνο για την εκφώνηση των εντολών

4.Βασική Ροή Γεγονότων

Ο χρήστης πραγματοποιεί τις ακόλουθες ενέργειες :

4.1 Εκινεί το εξωτερικό λογισμικό (third-party software)

4.2 Εισάγει το *λεξικό προφοράς* στο εξωτερικό λογισμικό

4.3 Από το *λεξικό προφοράς* που έχει ήδη εισαχθεί, επιλέγει τις λέξεις για τις οποίες τον ενδιαφέρει να γίνει η αναγνώριση

4.4 Για κάθε μία λέξη που επιλέγει, εισάγει την κατηγορία στην οποία επιθυμεί να ανήκει η λέξη αυτή.

4.5 Δημιουργεί την γραμματική με βάση τις κατηγορίες του βήματος 4.4.

4.6 Επιλέγει τους συνδυασμούς λέξεων για τις οποίες επιθυμεί να εκπαιδεύσει για πρώτη φορά το σύστημα.

4.7 Ηχογραφεί τους συνδυασμούς λέξεων που επέλεξε στο βήμα 4.5

4.8 Το εξωτερικό λογισμικό εκτελεί κατάλληλες διαδικασίες και αλγόριθμους ώστε



μέσω των ηχογραφήσεων, του λεξικού προφοράς και της γραμματικής να δημιουργηθούν τα αρχεία που αντιστοιχούν στο ακουστικό μοντέλο.

4.9 Το εξωτερικό λογισμικό παρέχει στον χρήστη ενημερωμένους δείκτες αναγνώρισης (recognition rates) για τους συνδυασμούς λέξεων που χρησιμοποιήθηκαν.

5.Εναλλακτικές Ροές

5.1 Εναλλακτική Ροή 1

5.1.1.

6.Μετα-συνθήκες

Καμία

7.Επιπρόσθετη Πληροφορία

Καμία

Σενάριο Χρήσης 2

1.Τίτλος

Εκπαίδευση ακουστικού μοντέλου για τα ελληνικά

2.Συνοπτική Περιγραφή

Ο χρήστης εκπαιδεύει το ακουστικό μοντέλο που αφορά στα ελληνικά, ώστε να βελτιώσει τα ποσοστά επιτυχίας αναγνώρισης των ελληνικών εντολών.

3.Προϋποθέσεις

3.1 Έχει εγκατασταθεί εξωτερικό λογισμικό (third-party software) με βάση το οποίο έχει ήδη δημιουργηθεί ακουστικό μοντέλο.

3.3 Έχει γίνει επιλογή ενός τυπικά κατάλληλου χώρου για ηχογράφηση (πχ ήσυχο δωμάτιο)

3.4 Έχει συνδεθεί στον Η/Υ κατάλληλο μικρόφωνο για την εκφώνηση των εντολών

4.Βασική Ροή Γεγονότων

Ο χρήστης πραγματοποιεί τις ακόλουθες ενέργειες :

4.1 Εκκινεί το εξωτερικό λογισμικό (third-party software)

4.2 Επιλέγει από το λεξικό προφοράς συνδυασμούς λέξεων για τις οποίες επιθυμεί να εκπαιδεύσει επιπλέον το σύστημα.

4. Ηχογραφεί τους συνδυασμούς λέξεων που επέλεξε στο βήμα 4.2



4.4 Το εξωτερικό λογισμικό εκτελεί κατάλληλες διαδικασίες και αλγόριθμους ώστε μέσω των ηχογραφήσεων, του λεξικού προφοράς και της γραμματικής να ενημερωθεί (εκπαιδευτεί) το ακουστικό μοντέλο.

4.5 Το εξωτερικό λογισμικό παράγει τα καινούργια αρχεία που αντιστοιχούν στο ενημερωμένο ακουστικό μοντέλο.

4.6 Το εξωτερικό λογισμικό παρέχει στον χρήστη ενημερωμένους δείκτες αναγνώρισης (recognition rates) για τους συνδυασμούς λέξεων που χρησιμοποιήθηκαν.

5.Εναλλακτικές Ροές

5.1 Εναλλακτική Ροή 1

5.1.1.

6.Μετα-συνθήκες

Καμία

7.Επιπρόσθετη Πληροφορία

Καμία

Σενάριο Χρήσης 3

1.Τίτλος

Ενσωμάτωση και έλεγχος ακουστικού μοντέλου

2.Συνοπτική Περιγραφή

Ο χρήστης ενσωματώνει τα φυσικά αρχεία που αντιστοιχούν στο ενημερωμένο ακουστικό μοντέλο και ελέγχει εάν η αναγνώριση λειτουργεί ικανοποιητικά.

3.Προϋποθέσεις

3.1 Έχει εγκατασταθεί το κατάλληλο εξωτερικό λογισμικό εκπαίδευσης που περιγράφεται στο Σενάριο 1.

3.2 Έχει συνδεθεί στον Η/Υ κατάλληλο μικρόφωνο για την εκφώνηση των εντολών

4.Βασική Ροή Γεγονότων

Ο χρήστης πραγματοποιεί τις ακόλουθες ενέργειες :

4.1 Μέσω κατάλληλης διεπαφής της εφαρμογής εισάγει το ενημερωμένο ακουστικό μοντέλο και γλωσσικό μοντέλο που δημιουργήθηκε ή εκπαδεύτηκε από το εξωτερικό



λογισμικό (Σενάρια Χρήσης 1 και 2)

4.2 Εκκινεί την αναγνώριση φωνής στα ελληνικά

4.2 Εκφέρει στα ελληνικά κάποιους από τους συνδυασμούς λέξεων που έχει ήδη εκπαιδεύσει. Εάν πρόκειται για την καθοδήγηση του ρομπότ, τυπικοί συνδυασμοί αποτελούν οι : ‘Προχώρα μπροστά’, ‘Πήγαινε πίσω’, ‘Στρίψε δεξιά’, ‘Στρίψε αριστερά’ και ‘Σταμάτα’

4.3 Η εφαρμογή αναγνωρίζει τις λέξεις που εκφωνούνται και παρουσιάζει ποια λέξη εκτιμάται ότι εκφωνήθηκε καθώς και έναν αριθμό μεταξύ 0 και 1 που αποτελεί την *βεβαιότητα (confidence)* αναγνώρισης

4.4 Ο χρήστης σταματάει την αναγνώριση

5.Εναλλακτικές Ροές

5.1 Εναλλακτική Ροή 1

5.1.1. Η εφαρμογή δεν αναγνώρισε τις εκφερόμενες λέξεις

5.1.2. Η εφαρμογή δεν παρουσιάζει καμία πληροφορία για την αποτυχία της αναγνώρισης

5.2 Εναλλακτική Ροή 2

5.2.1. Τα αρχεία που αποτελούν το ακουστικό μοντέλο δεν βρέθηκαν

5.2.2. Η εφαρμογή παρουσιάζει κατάλληλο μήνυμα

5.3 Εναλλακτική Ροή 3

5.3.1. Υπήρξε κάποιο λάθος κατά την διαδικασία της αναγνώρισης

5.3.2. Η εφαρμογή παρουσιάζει το μήνυμα λάθους που προέκυψε

6.Μετα-συνθήκες

Καμία

7.Επιπρόσθετη Πληροφορία

Καμία

Σενάριο Χρήσης 4

1.Τίτλος

Επιλογή περιβάλλοντος

2.Συνοπτική Περιγραφή

Ο χρήστης επιλέγει εάν θα δώσει εντολές στο εικονικό ή στο πραγματικό περιβάλλον



3.Προϋποθέσεις

- 3.1. Κατάλληλο μικρόφωνο είναι συνδεδεμένο με τον Η/Υ του χρήστη
- 3.2. Έχει πραγματοποιηθεί η σύνδεση μεταξύ του Η/Υ και του φυσικού ρομποτικού συστήματος Lego NXT 2.0 μέσω Bluetooth (pairing)

4.Βασική Ροή Γεγονότων

- 4.1. Ο χρήστης μέσα από κατάλληλο μενού που παρέχει η εφαρμογή επιλέγει το περιβάλλον στο οποίο θα δώσει εντολές

5.Εναλλακτικές Ροές

5.1 Εναλλακτική Ροή 1

- 5.1.1. Ο χρήστης επιλέγει να δώσει εντολές στο εικονικό περιβάλλον (Σενάριο Χρήσης 5)

5.2 Εναλλακτική Ροή 2

- 5.2.1. Ο χρήστης επιλέγει να δώσει εντολές στο πραγματικό περιβάλλον (Σενάριο Χρήσης 6)

6.Μετα-συνθήκες

Καμία

7.Επιπρόσθετη Πληροφορία

Καμία

Σενάριο Χρήσης 5

1.Τίτλος

Εκκίνηση σε εικονικό περιβάλλον

2.Συνοπτική Περιγραφή

Η εφαρμογή εκκινείται σε συγκεκριμένο εικονικό περιβάλλον

3.Προϋποθέσεις

- 3.1. Έχει οριστεί η μορφή του εικονικού περιβάλλοντος (π.χ. εξωτερικός χώρος) και η θέση του εικονικού ρομπότ Lego NXT μέσα σε αυτό
- 3.2. Έχει φορτωθεί από την εφαρμογή το αρχείο με την κατάλληλη γραμματική που περιέχει τις εντολές ως κείμενο σε συγκεκριμένη xml μορφή (xml grammar file)



4.Βασική Ροή Γεγονότων

Ο χρήστης πραγματοποιεί τις ακόλουθες ενέργειες :

4.1 Μπορεί να πλοηγήσει το εικονικό Lego NXT ρομπότ στο εικονικό περιβάλλον μέσω μίας επιπλέον υπάρχουσας γραφικής διεπαφής.

4.2 Ρυθμίζει παραμέτρους του συστήματος (Σενάριο Χρήσης 7)

5.Εναλλακτικές Ροές

Καμία

6.Μετα-συνθήκες

Καμία

7.Επιπρόσθετη Πληροφορία

Καμία

Σενάριο Χρήσης 6

1.Τίτλος

Εκκίνηση σε πραγματικό περιβάλλον

2.Συνοπτική Περιγραφή

Η εφαρμογή εκκινείται και ο χρήστης μπορεί να εργαστεί στο πραγματικό περιβάλλον

3.Προϋποθέσεις

3.1. Έχει κατασκευαστεί συγκεκριμένη μορφή του φυσικού ρομπότ Lego NXT 2.0

3.2 . Έχει πραγματοποιηθεί η σύνδεση μεταξύ του H/Y και του φυσικού ρομπότ μέσω του πρωτοκόλλου Bluetooth (pairing)

3.3. Έχει φορτωθεί από την εφαρμογή το αρχείο με την κατάλληλη γραμματική που περιέχει τις εντολές ως κείμενο σε συγκεκριμένη xml μορφή (xml grammar file)

4.Βασική Ροή Γεγονότων

4.1. Ο χρήστης ρυθμίζει παραμέτρους του συστήματος (Σενάριο Χρήσης 7)

5.Εναλλακτικές Ροές

Καμία



6.Μετα-συνθήκες

Καμία

7.Επιπρόσθετη Πληροφορία

Καμία

Σενάριο Χρήσης 7

1.Τίτλος

Ρύθμιση παραμέτρων

2.Συνοπτική Περιγραφή

Ο χρήστης ρυθμίζει τις παραμέτρους που αφορούν την γλώσσα και την έναρξη, παύση της αναγνώρισης

3.Προϋποθέσεις

3.1. Έχει συνδεθεί στον Η/Υ κατάλληλο μικρόφωνο για την εκφώνηση των εντολών

4.Βασική Ροή Γεγονότων

4.1. Ο χρήστης εργάζεται στο εικονικό σύστημα και επιλέγει να ξεκινήσει η αναγνώριση ελληνικών

4.2. Ο χρήστης εκφέρει στα ελληνικά κάποια από τις εντολές ‘Προχώρα μπροστά’, ‘Πήγαινε πίσω’, ‘Στρίψε δεξιά’, ‘Στρίψε αριστερά’ και ‘Σταμάτα’

4.3. Το εικονικό ρομπότ Lego NXT εκτελεί τις ελληνικές εντολές (Σενάριο 8)

4.4. Ο χρήστης επιλέγει να σταματήσει η αναγνώριση των ελληνικών

4.5. Το εικονικό ρομπότ σταματάει να εκτελεί εντολές

5.Εναλλακτικές Ροές

5.1 Εναλλακτική Ροή 1

5.1.1. Ο χρήστης εργάζεται στο εικονικό σύστημα και επιλέγει να ξεκινήσει η αναγνώριση αγγλικών

5.1.2. Ο χρήστης εκφέρει στα αγγλικά κάποια από τις εντολές ‘Move forwards’, ‘Move Backwards’, ‘Turn Left’, ‘Turn Right’ και ‘Stop’

5.1.3. Το εικονικό ρομπότ Lego NXT εκτελεί τις εντολές στα αγγλικά (Σενάριο 9)

5.1.4. Ο χρήστης επιλέγει να σταματήσει η αναγνώριση των αγγλικών

5.1.5. Το εικονικό ρομπότ σταματάει να εκτελεί εντολές



5.2 Εναλλακτική Ροή 2

5.2.1. Ο χρήσης εργάζεται στο πραγματικό σύστημα και επιλέγει να ξεκινήσει η αναγνώριση ελληνικών

5.2.2. Ο χρήστης εκφέρει στα ελληνικά κάποια από τις εντολές του βήματος 4.2

5.2.3. Το πραγματικό ρομπότ Lego NXT ξεκινάει να εκτελεί τις εντολές στα ελληνικά (Σενάριο 8)

5.2.4. Ο χρήστης επιλέγει να σταματήσει η αναγνώριση των ελληνικών

5.2.5. Το πραγματικό ρομπότ σταματάει να εκτελεί εντολές

5.3 Εναλλακτική Ροή 3

5.3.1. Ο χρήσης εργάζεται στο πραγματικό σύστημα και επιλέγει να ξεκινήσει η αναγνώριση αγγλικών

5.3.2. Ο χρήστης εκφέρει στα αγγλικά κάποια από τις εντολές του βήματος 5.1.2

5.3.3. Το πραγματικό ρομπότ Lego NXT εκτελεί τις αγγλικές εντολές (Σενάριο 9)

5.3.4. Ο χρήστης επιλέγει να σταματήσει η αναγνώριση των αγγλικών

5.3.5. Το πραγματικό ρομπότ σταματάει να εκτελεί εντολές

6.Μετα-συνθήκες

Καμία

7.Επιπρόσθετη Πληροφορία

Καμία

Σενάριο Χρήσης 8

1.Τίτλος

Εκτέλεση φωνητικών εντολών στα ελληνικά

2.Συνοπτική Περιγραφή

Το ρομπότ κινείται με βάση συγκεκριμένες εντολές στα ελληνικά

3.Προϋποθέσεις

3.1. Έχει γίνει η αναγνώριση κάποιας συγκεκριμένης εντολής

4.Βασική Ροή Γεγονότων

4.1. Το εικονικό ρομποτικό σύστημα εκτελεί συγκεκριμένες κινήσεις. Κατά αντιστοιχία των ελληνικών φωνητικών εντολών της βασικής ροής γεγονότων του σεναρίου 7:



κινείται μπροστά, κινείται πίσω, γυρνάει αριστερά (90 μοίρες), γυρνάει δεξιά (90 μοίρες) ή σταματάει κάθε είδους κίνηση

5.Εναλλακτικές Ροές

5.1 Εναλλακτική Ροή 1

5.1.1. Το πραγματικό ρομποτικό σύστημα εκτελεί συγκεκριμένες κινήσεις. Οι κινήσεις έχουν την συμπεριφορά των κινήσεων όπως περιγράφονται στη βασική ροή γεγονότων, αλλά εκτελούνται στο πραγματικό περιβάλλον.

6.Μετα-συνθήκες

Καμία

7.Επιπρόσθετη Πληροφορία

Καμία

Σενάριο Χρήσης 9

1.Τίτλος

Εκτέλεση φωνητικών εντολών στα αγγλικά

2.Συνοπτική Περιγραφή

Το ρομπότ κινείται με βάση συγκεκριμένες εντολές στα αγγλικά

3.Προϋποθέσεις

3.1. Έχει γίνει η αναγνώριση κάποιας συγκεκριμένης εντολής

4.Βασική Ροή Γεγονότων

4.1. Το εικονικό ρομποτικό σύστημα εκτελεί συγκεκριμένες κινήσεις. Κατά αντιστοιχία των αγγλικών φωνητικών εντολών της εναλλακτικής ροής γεγονότων του σεναρίου 7: κινείται μπροστά, κινείται πίσω, γυρνάει αριστερά (90 μοίρες), γυρνάει δεξιά (90 μοίρες) ή σταματάει κάθε είδους κίνηση .

5.Εναλλακτικές Ροές

5.1 Εναλλακτική Ροή 1

5.1.1. Το πραγματικό ρομποτικό σύστημα εκτελεί συγκεκριμένες κινήσεις. Οι κινήσεις έχουν την συμπεριφορά των κινήσεων όπως περιγράφονται στη βασική ροή γεγονότων, αλλά εκτελούνται στο πραγματικό περιβάλλον.

**6.Μετα-συνθήκες**

Καμία

7.Επιπρόσθετη Πληροφορία

Καμία

5.3.2 Λογική απεικόνιση (logical view)

Η λογική απεικόνιση αποτυπώνει και αναλύει τη δομή, τη σημασία και τις αλληλεπιδράσεις σημαντικών λογικών δομικών στοιχείων της εφαρμογής όπως οι οργανωτικές ιεραρχίες και τα δομικά στοιχεία λογισμικού.

- **Λογικές οντότητες ακουστικού μοντέλου**

Η κατασκευή του ακουστικού μοντέλου προϋποθέτει την αξιοποίηση τόσο του *λεξικού προφοράς (pronunciation dictionary)* όσο και της *γραμματικής (grammar)* οι οποίες δίνονται στην συγκεκριμένη ενότητα μέσω μίας λογικής αποτύπωσης.

Όσον αφορά στο λεξικό της εφαρμογής, παρατίθεται στον Πίνακα 12 και περιλαμβάνει λέξεις που δεν αφορούν αποκλειστικά και μόνο στις εντολές καθοδήγησης. Οι ελληνικές λέξεις αποτυπώνονται με λατινικούς χαρακτήρες (greeklish) λόγω των περιορισμών του λογισμικού εκπαίδευσης όπως περιγράφονται στην ενότητα 5.2.6. Σε κάθε μία λέξη έχει αντιστοιχηθεί το σύνολο των φωνημάτων που καταδεικνύουν την προφορά της, ενώ το σύνολο αυτών των προφορών έχουν εξαχθεί μέσω συγκεκριμένου [i25] δικτυακού τόπου που χρησιμοποιεί λήμματα από λεξικά της ελληνικής γλώσσας.

Πίνακας 11: Το λεξικό προφοράς της εφαρμογής

Λέξη	Προφορά
AMESOS	a m e s o s
ARISTA	a r i s t a
ARISTERA	a r i s t e r a
ARISTEVO	a r i s t e v o
BROSTA	b r o s t a
DEKSIA	d e k s i a
DEKTIKOS	d e k t i k o s
DIXNO	d i x n o



PIJENE	pijene
PISO	piso
PROXORA	proxora
STAMATA	stamata
STAMATIMA	stamatima
STAMNA	stamna
STATIKOS	statikos
STOMA	stoma
STOXOS	stoxos
STRIMENOS	strimenos
STRIPSE	stripse
STRIVO	strivo
TORA	tora
XORA	xora
XOREVO	xorevo
XOROS	xoros

Όσον αφορά στη γραμματική, σε αυτό το σημείο δεν απαιτείται ο ορισμός κάποιου είδους φορμαλισμού (όπως πχ ο ορισμός με τερματικά και μη-τερματικά σύμβολα μίας *γραμματικής ανεξάρτητης συμφοραζομένων*) – άλλωστε αυτή την διαδικασία θα την αναλάβει σε επόμενο στάδιο το εξωτερικό λογισμικό εκπαίδευσης Simon. Μπορούμε όμως να έχουμε μία αφαιρετική λογική απεικόνιση όπου αντιστοιχούνται επιθυμητοί κανόνες σε συνδυασμούς λέξεων :

Πίνακας 12: Λογικοί κανόνες αντιστοίχισης

Κανόνας	Αντιστοίχιση σε ομάδες λέξεων του λεξικού
TURN LEFT	STRIPSE ARISTERA
TURN RIGHT	STRIPSE DEXIA
MOVE FORWARDS	PIJENE BROSTA PROXORA BROSTA
MOVE BACKWARDS	PIJENE PISO PROXORA PISO
STOP	STAMATA



Οι λογικοί κανόνες που φαίνονται στον Πίνακα 13 δεν αποτελούν γραμματική. Αποτελούν όμως την βάση της κατασκευής της σε επόμενο στάδιο, καθώς αντιστοιχούν ένα σύνολο κανόνων σε συγκεκριμένους επιτρεπόμενους συνδυασμούς λέξεων. Οι λέξεις αυτές ανήκουν φυσικά στο λεξικό της εφαρμογής.

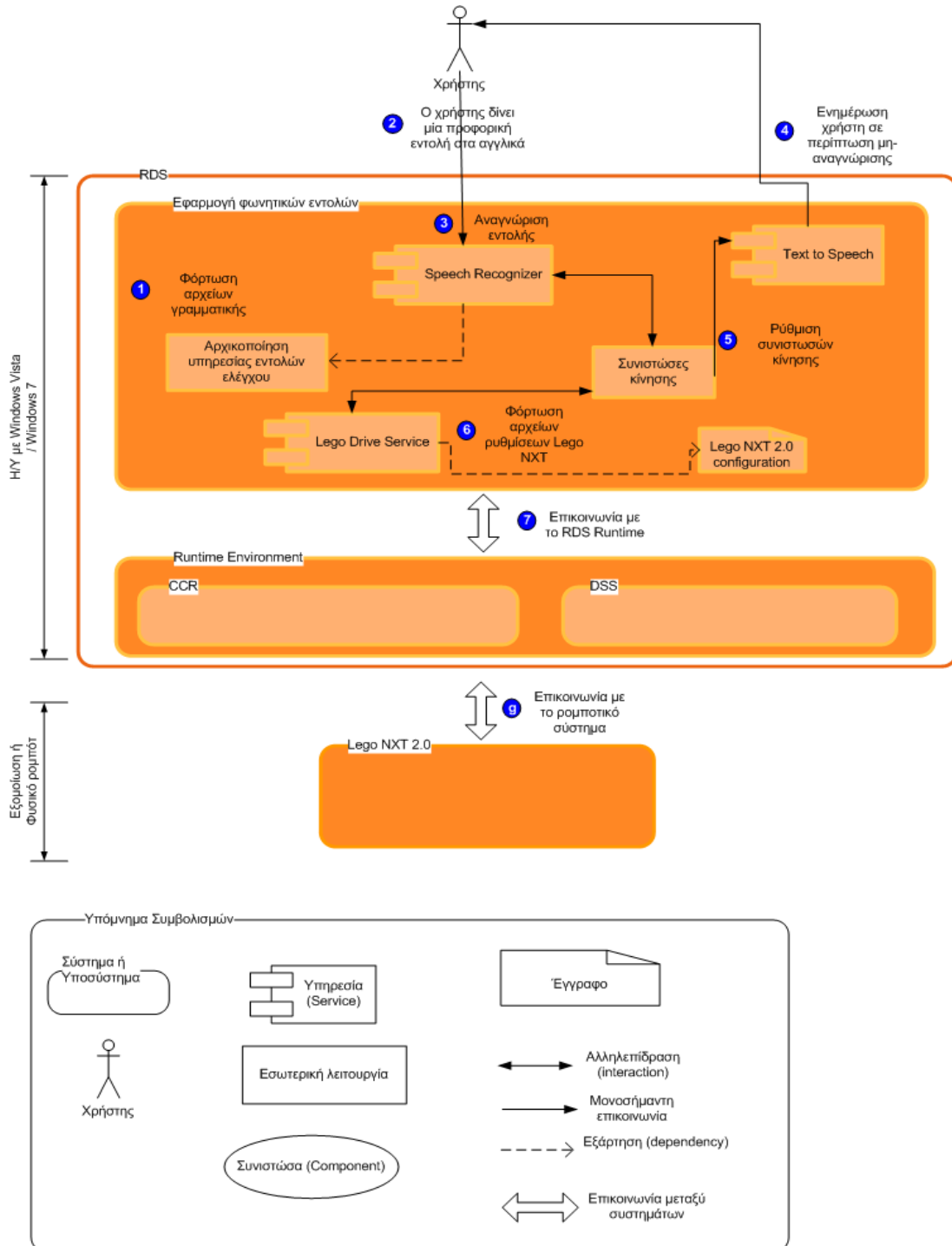
- **Πλαίσιο λειτουργίας**

Το πλαίσιο λειτουργίας με βάση τα συγκεκριμένα σενάρια χρήσης όπως εμφανίστηκαν στην λειτουργική άποψη, παρουσιάζει με έναν αρκετά παραστατικό τρόπο τον βασικό μηχανισμό της εφαρμογής, ο οποίος περιλαμβάνει τις πλέον σημαντικές λειτουργίες της.

Για τον σκοπό αυτό χρησιμοποιείται το διάγραμμα πλαισίου λειτουργίας (system context diagram), το οποίο δεν ακολουθεί κάποιο τυπικό φορμαλισμό (πχ UML) αλλά συνδυάζει τα παρακάτω διαγράμματα :

- Fundamental modelling concepts block diagram : περιγράφει τις συνιστώσες (components) και τα συστήματα τα οποία παίρνουν μέρος στον μηχανισμό που περιγράφεται
- UML collaboration diagram : περιγράφει την ακολουθία των αλληλεπιδράσεων που αποτελούν την ροή ελέγχου ενός μηχανισμού
- UML activity diagram : περιγράφει τις οντότητες και την κατάστασή τους, παράλληλα με την ροή ελέγχου (object flow)

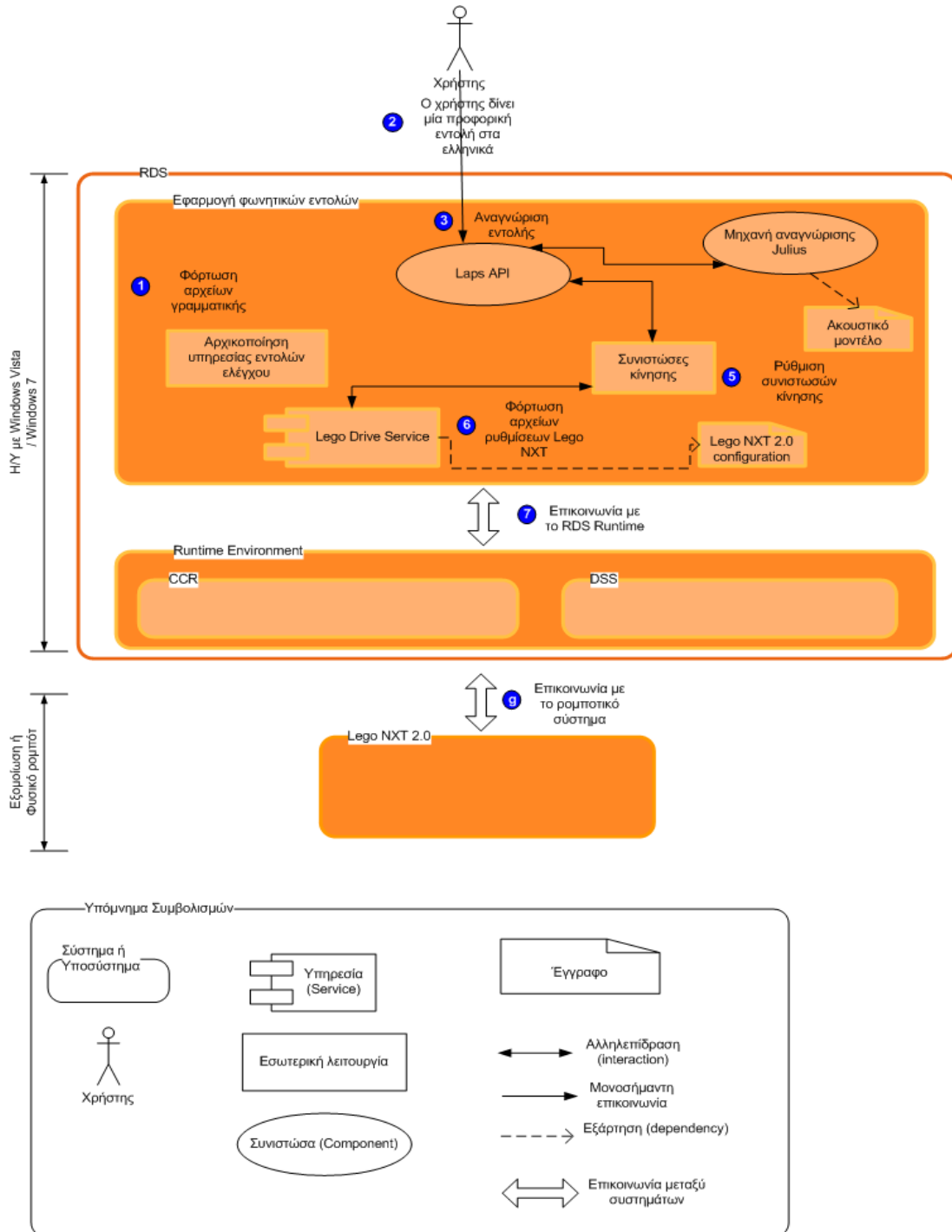
Στο παρακάτω διάγραμμα πλαισίου λειτουργίας του Σχήματος 13 παρουσιάζονται μόνο οι πολύ βασικές υπηρεσίες, συνιστώσες και εσωτερικές λειτουργίες που λαμβάνουν μέρος στην αναγνώριση των αγγλικών εντολών - θεωρούμε ότι ο χρήστης ήδη έχει επιλέξει να εκφωνεί στα αγγλικά. Στο διάγραμμα δεν εξετάζουμε εάν πρόκειται για το εικονικό ή το πραγματικό ρομπότ καθώς η λογική αποτύπωσης είναι πανομοιότυπη. Τα σημεία 1-8 καταδεικνύουν την χρονική ακολουθία των βασικών δράσεων που λαμβάνουν μέρος, καθώς επίσης και τις αλληλεπιδράσεις μεταξύ των συνιστωσών και των υπηρεσιών (services). Βλέπουμε ότι αρχικά φορτώνονται τα απαραίτητα αρχεία γραμματικής και έπειτα ο χρήστης δίνει μία εντολή στα αγγλικά που αναγνωρίζεται από την υπηρεσία του Speech Recognizer των Windows. Εάν η εντολή έχει μικρή βεβαιότητα ή δεν αναγνωριστεί καθόλου, τότε χρησιμοποιείται η υπηρεσία Text-to-Speech για να επιστρέψει στον χρήστη μία απλή φωνητική ειδοποίηση μη-αναγνώρισης. Εάν η εντολή αναγνωριστεί, τότε εσωτερικές λειτουργίες με βάση το κείμενο της αναγνωρισμένης εντολής ρυθμίζουν τις συνιστώσες κίνησης της υπηρεσίας Lego Drive (όπως είναι το διάλυσμα κίνησης, η ισχύς κτλ). Έπειτα η υπηρεσία Lego Drive φορτώνει τα κατάλληλα αρχεία ρυθμίσεων και μέσω της επικοινωνίας με το MRDS runtime δίνονται οι κατάλληλες εντολές στο ρομποτικό σύστημα.



Σχήμα 13: Διάγραμμα πλαισίου λειτουργίας της αναγνώρισης των αγγλικών εντολών



Στο επόμενο διάγραμμα πλαισίου λειτουργίας του Σχήματος 14 αφορά στην αναγνώριση των ελληνικών εντολών. Η ροή είναι παρόμοια με το προηγούμενο διάγραμμα, ωστόσο υπάρχουν οι εξής ουσιαστικές διαφοροποιήσεις : καταρχήν η φωνητική αναγνώριση γίνεται μέσω της SRE μηχανής Julius ή οποία καλείται από συγκεκριμένο API (Laps API). Το Laps Api είναι ο wrapper όπως περιγράφεται στην ενότητα 5.2.5 μέσω του οποίου η εφαρμογή καλεί το SRE Julius. Η αναγνώριση που επιτελεί το SRE εξαρτάται από το ακουστικό μοντέλο το οποίο εσωτερικά αξιοποιεί. Η δεύτερη σημαντική διαφοροποίηση αφορά στο ότι στα ελληνικά δεν συμπεριλαμβάνεται υπηρεσία ή συνιστώσα Text-to-Speech για να απαντήσει στα ελληνικά όπως αυτό διευκρινίζεται και στην απαίτηση R9.

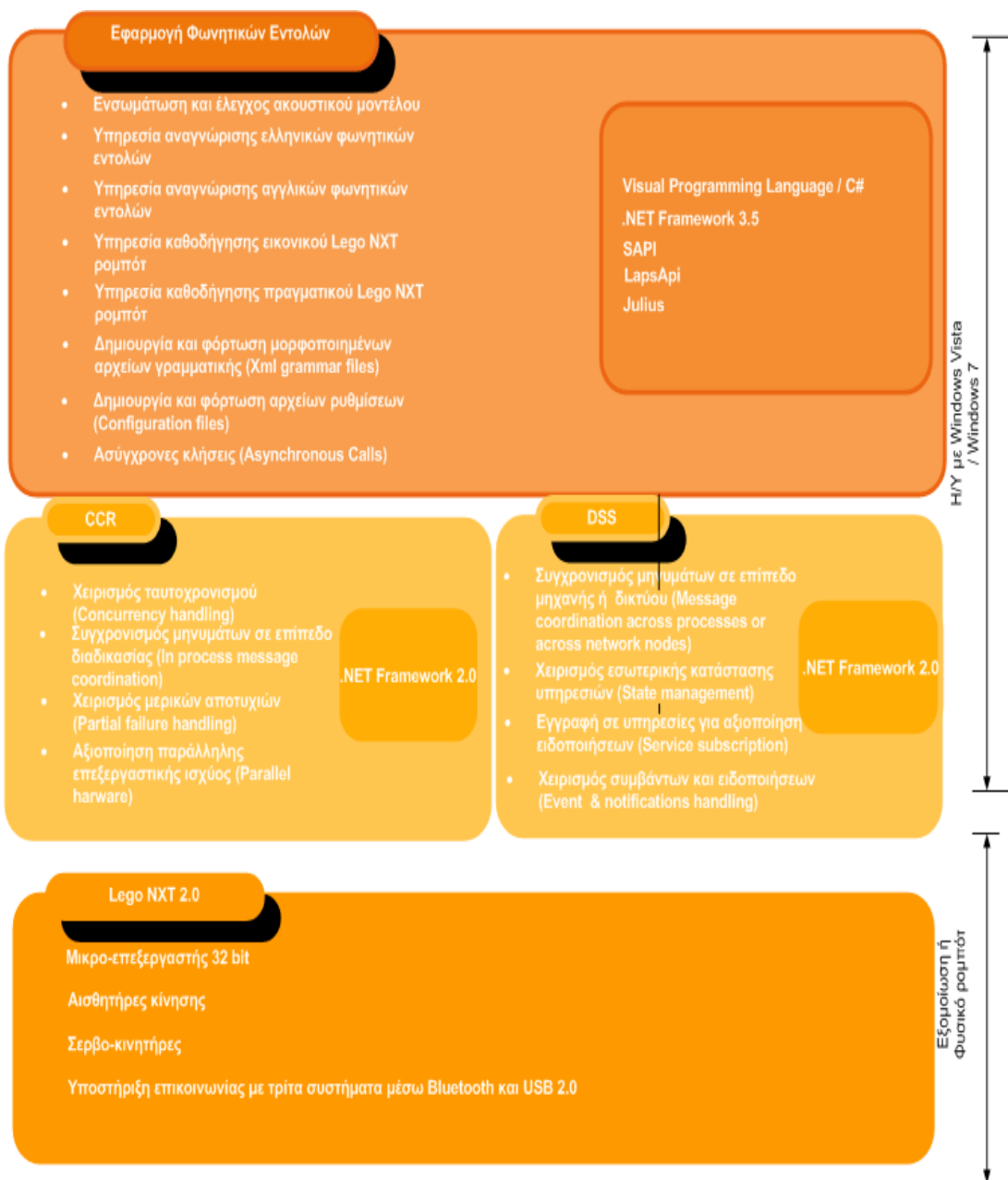


Σχήμα 14: Διάγραμμα πλαισίου λειτουργίας της αναγνώρισης των ελληνικών εντολών



• Διαστρωμάτωση συστήματος

Η διαστρωμάτωση του συστήματος αφορά στα σημαντικότερα επίπεδα και τις που λαμβάνουν μέρος σε μία αποτύπωση η οποία είναι στατική. Στο αντίστοιχο διάγραμμα παρακάτω απεικονίζονται τα βασικά δομικά στοιχεία αλλά και κάποιοι από τους μηχανισμούς του κάθε επιπέδου. Στα εσωτερικά ένθετα παρουσιάζονται καλούμενες εξωτερικές συνιστώσες και εμπλεκόμενες υποδομές ή τεχνολογίες υλοποίησης.

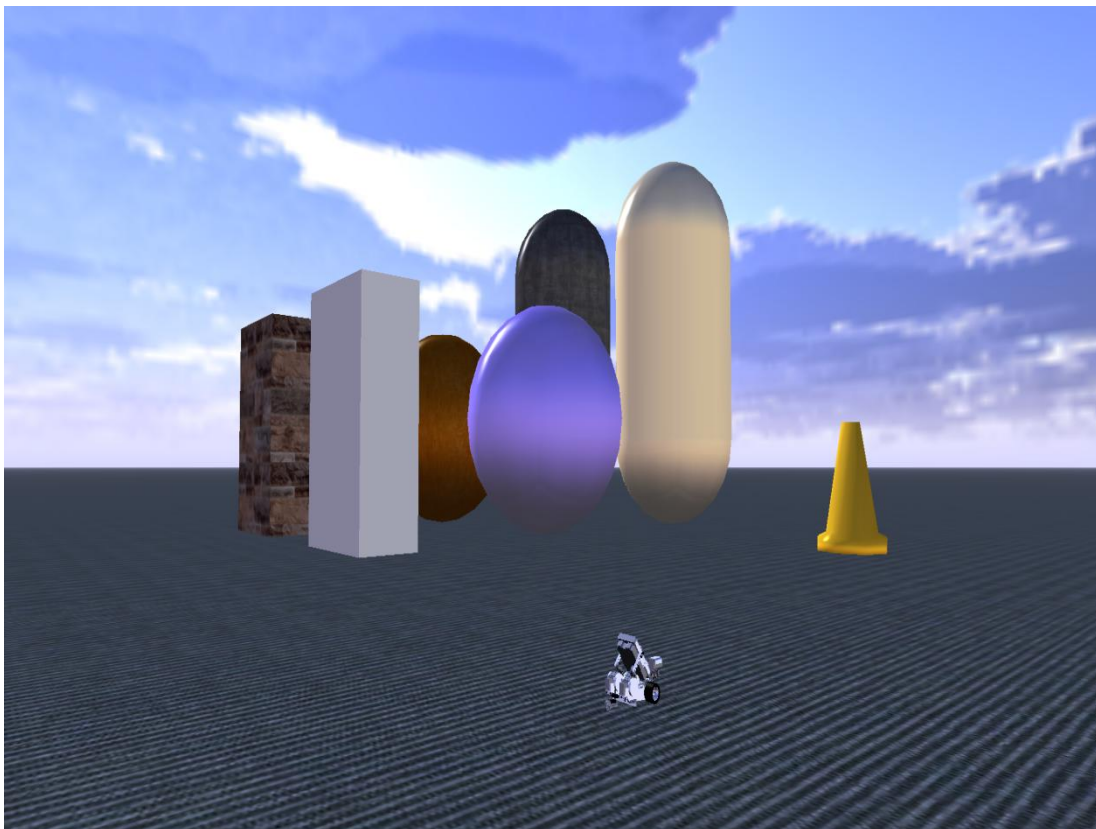


Σχήμα 15: Διαστρωμάτωση επιπέδων συστήματος



- **Εφαρμογή περιβάλλοντος προσομοίωσης**

Το τρισδιάστατο περιβάλλον που χρησιμοποιείται στα σενάρια φωνητικού ελέγχου για την καθοδήγηση του εικονικού Lego NXT, μπορούμε να το δούμε σαν μία εξειδικευμένη μορφή λογικής απεικόνισης. Η καθοδήγηση του εικονικού ρομπότ γίνεται σε αναπαράσταση εξωτερικού χώρου. Ο χρήστης εάν διαθέτει κατάλληλη έκδοση του Microsoft Robotics Studio (πχ Academic Edition) μπορεί να αλλάξει την θέση των αντικειμένων, να προσθέσει στον χώρο τρισδιάστατα αντικείμενα και γενικότερα να μεταβάλλει το περιβάλλον κατά βούληση. Στην Εικόνα 17 βλέπουμε την αρχική όψη του εικονικού περιβάλλοντος και τη θέση εικονικού του Lego NXT όταν ο χρήστης επιλέγει να εργαστεί σε συνθήκες προσομοίωσης.



Εικόνα 16: Αναπαράσταση εσωτερικού χώρου

5.3.3 Απεικόνιση της λειτουργικής υποστήριξης (maintenance view)

Η απεικόνιση της λειτουργικής υποστήριξης ή συντήρησης (maintenance view) της εφαρμογής, αναφέρεται στην ανάλυση των τακτικών εκείνων διαδικασιών που είναι σημαντικές για την υποστήριξη της λειτουργίας της και οι οποίες στο μεγαλύτερο μέρος τους δεν μπορούν να παρέχονται από την ίδια την εφαρμογή.



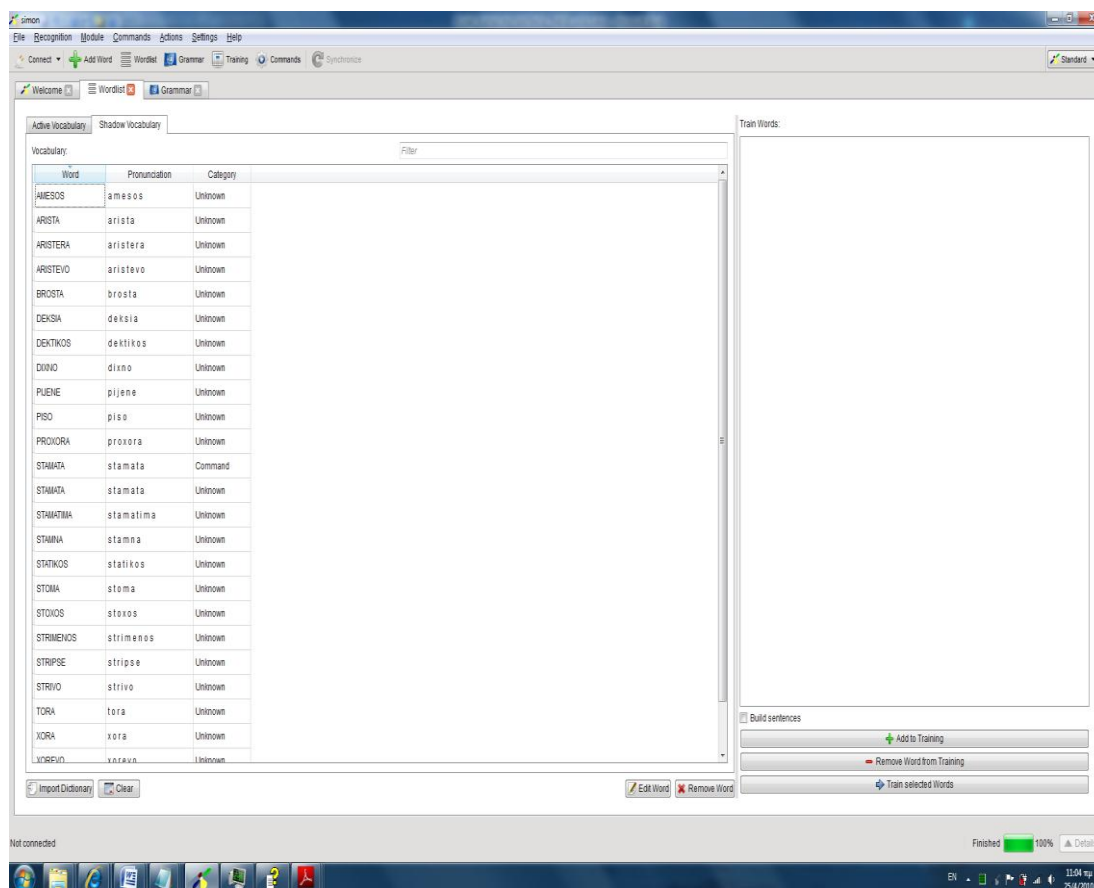
Στην ενότητα 5.3.1 εκτός των άλλων περιγράφονται τα εξής σενάρια :

1. ‘1.Δημιουργία ακουστικού μοντέλου’
2. ‘2.Εκπαίδευση ακουστικού μοντέλου’
3. ‘3.Ενσωμάτωση και έλεγχος ακουστικού μοντέλου’

Τα σενάρια αυτά ενσωματώνουν ζωτικές διαδικασίες συντήρησης για την λειτουργία της εφαρμογής που αφορούν στο *γλωσσικό και ακουστικό μοντέλο* χωρίς τα οποία δεν μπορούμε να έχουμε αναγνώριση στα ελληνικά. Επιπλέον οι διαδικασίες αυτές παρέχονται από το εξωτερικό λογισμικό Simon, όπως αυτό έχει ήδη περιγραφεί και η πραγματοποίησή τους δίνεται αμέσως παρακάτω :

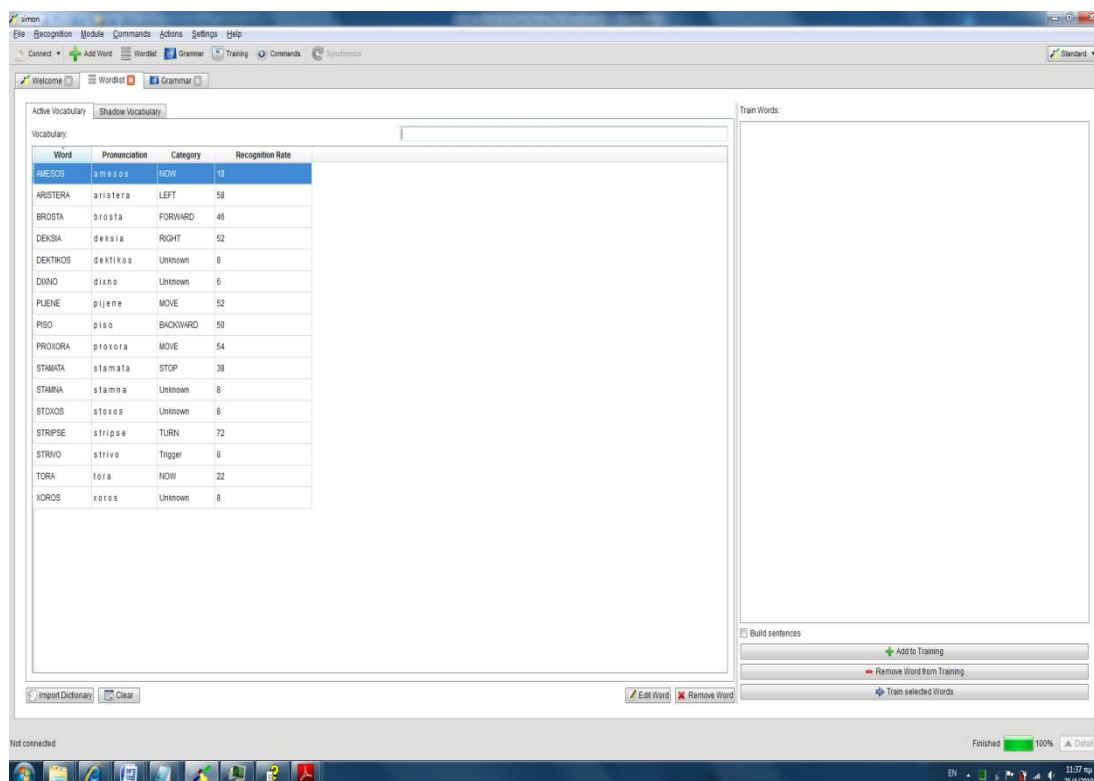
- **Πραγματοποίηση δημιουργίας ακουστικού μοντέλου**

Το *λεξικό προφοράς (pronunciation dictionary)* με την μορφή που έχει στην ενότητα 5.3.2 πρέπει πρώτα να μεταφερθεί σε ένα απλό αρχείο κειμένου. Η ενσωμάτωση του στο λογισμικό Simon πραγματοποιείται μέσα από εύκολες διαδικασίες εισαγωγής (dictionary import), καθώς υποστηρίζει διάφορες διαμορφώσεις λεξικών (dictionary formats). Το λεξικό αυτό στο Simon ονομάζεται *αφανές λεξικό (shadow dictionary)* και χρησιμοποιείται για να παρέχει τις προφορές των λέξεων, ενώ μπορεί να είναι οσοδήποτε μεγάλο και να εμπλουτίζεται απεριόριστα. Στην Εικόνα 18 αμέσως πιο κάτω φαίνεται το *λεξικό προφοράς* μετά την ενσωμάτωσή του :



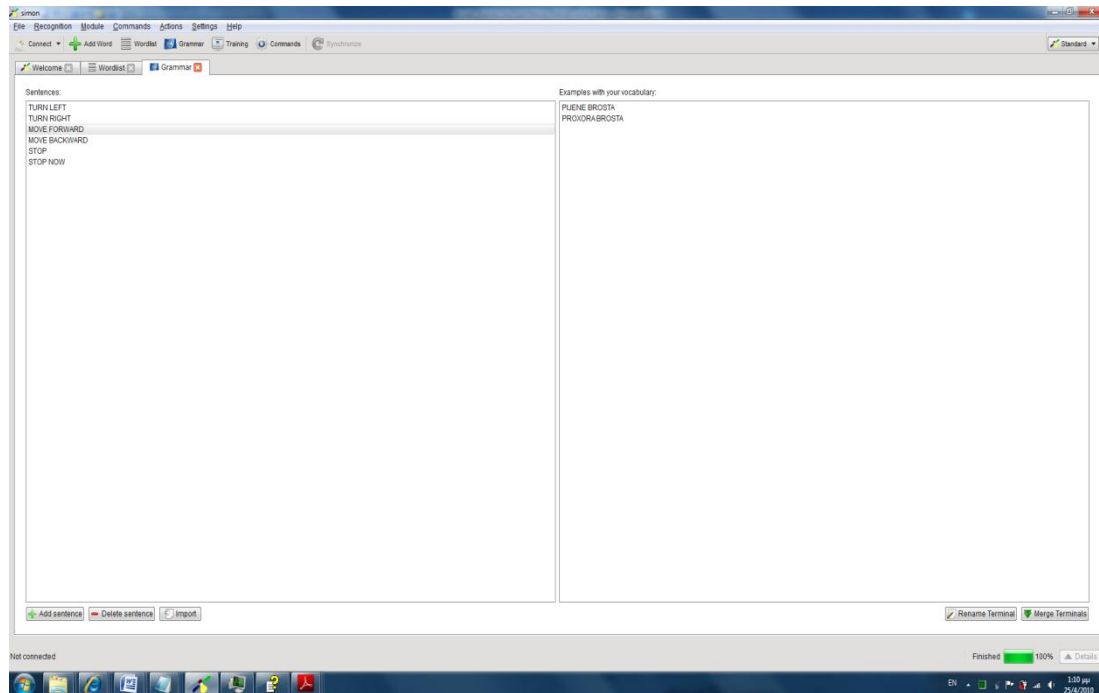
Εικόνα 17: Εισαγωγή του λεξικού της εφαρμογής στο Simon

Επόμενο βήμα αποτελεί η επιλογή των επιθυμητών λέξεων που θα χρησιμοποιηθούν. Στο Simon οι λέξεις που επιλέγει ο χρήστης προστίθενται σε μία καινούργια λίστα που αποτελεί το ενεργό λεξικό (active dictionary). Το λεξικό αυτό, σε αντίθεση με το αφανές λεξικό, πρέπει να είναι πολύ μικρότερο και τα περιεχόμενα του χρησιμοποιούνται στην διαδικασία της αναγνώρισης. Κάθε λέξη που προστίθεται στο ενεργό λεξικό μπορεί να υπαχθεί σε μία επιθυμητή κατηγορία. Στην παρακάτω Εικόνα 19 μπορούμε να δούμε το ενεργό λεξικό της εφαρμογής που περιέχει τόσο τις χαρακτηριστικές λέξεις που θα χρησιμοποιηθούν στην πλοήγηση του ρομπότ όπως “ARISTERA”, “BROSTA” κτλ, όσο και κάποιες άλλες. Κάθε λέξη που αφορά στην πλοήγηση υπάγεται κάτω από μία κατηγορία πχ οι λέξεις “PIJENE” και “PROXORA” ανήκουν στην λογική κατηγορία “MOVE”.



Εικόνα 18: Το ενεργό λεξικό της εφαρμογής στο Simon

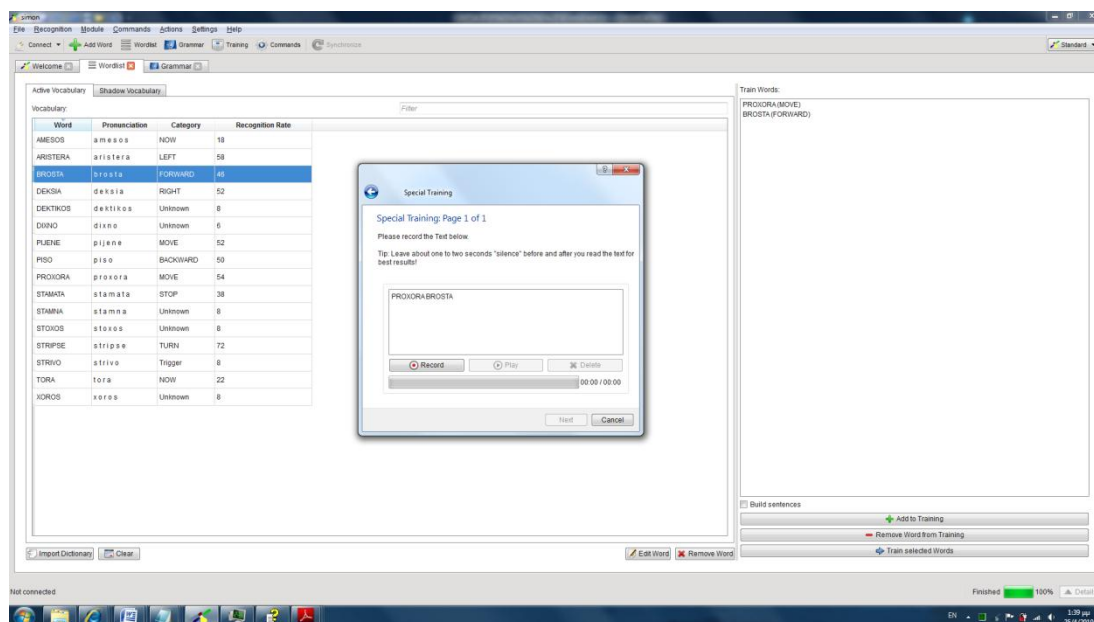
Για το λογισμικό Simon οι κατηγορίες στις οποίες υπάγονται οι λέξεις έχουν ειδική σημασία καθώς αποτελούν ουσιαστικά τα τερματικά σύμβολα (terminals) με βάση τα οποία δημιουργούνται από το Simon εσωτερικά οι κανόνες παραγωγής της γραμματικής σε επόμενο στάδιο. Από την οπτική του χρήστη, ο ορισμός της γραμματικής μέσα από το Simon είναι μία απλή υπόθεση, καθώς το μόνο που πρέπει να γίνει είναι η επιλογή των επιθυμητών κατηγοριών ή συνδυασμών τους. Έτσι με βάση τα παραπάνω, ενώ ο χρήστης επιλέγει μόνο τις κατηγορίες που επιθυμεί, ουσιαστικά επιλέγει τα τερματικά σύμβολα με τα οποία το λογισμικό θα κατασκευάσει την απαιτούμενη γραμματική.



Εικόνα 19: Η γραμματική της εφαρμογής στο Simon

Όπως παρατηρούμε στο αριστερό μέρος της Εικόνας 20, έχουν επιλεγεί οι συνδυασμοί των κατηγοριών που μας ενδιαφέρουν οι οποίοι ουσιαστικά αποτελούν συνδυασμούς τερματικών συμβόλων. Στο δεξί μέρος της οθόνης φαίνονται οι συνδυασμοί των λέξεων του ενεργού λεξικού που αντιστοιχούν σε αυτές τις κατηγορίες και εισάγονται αυτόματα από το Simon. Έτσι για παράδειγμα, βλέπουμε ότι στον συνδυασμό κατηγοριών που επιλέξαμε ως “MOVE FORWARD” αυτόματα αντιστοιχούν οι προτάσεις “PIJENE BROSTA” και “PROXORA BROSTA”, στον συνδυασμό “TURN LEFT” αντιστοιχεί η πρόταση “STRIPSE ARISTERA” κ.ο.κ.

Τελικό βήμα μετά τον ορισμό της γραμματικής αποτελεί η ηχογράφηση των επιθυμητών προτάσεων για την δημιουργία της βάσης ήχου που θα χρησιμοποιηθεί στην αναγνώριση. Στην περίπτωση της εφαρμογής μας, τυπικές τέτοιες προτάσεις αποτελούν οι φωνητικές εντολές : “PROXORA BROSTA”, “PIJENE PISO”, “STRIPSE DEKSIA”, “STRIPSE ARISTERA” και “STAMATA”. Το λογισμικό SIMON παρέχει τις λειτουργίες ώστε να πραγματοποιούνται οι ηχογραφήσεις των επιθυμητών προτάσεων με απλό τρόπο.



Εικόνα 20: Ηχογραφήσεις προτάσεων για την δημιουργία βάσης ήχου στο Simon

Στην Εικόνα 21 φαίνεται η εκκίνηση της λειτουργία ηχογράφησης για την πρόταση ‘PROXORA BROSTA’. Μετά την ηχογράφηση ενημερώνονται οι δείκτες αναγνώρισης (recognition rates) του ενεργού λεξικού οι οποίοι στο Simon σηματοδοτούν το πόσες φορές έχει αναγνωριστεί επιτυχημένα μία λέξη.

Μετά την ηχογράφηση των επιθυμητών προτάσεων, την διαδικασία δημιουργίας του ακουστικού μοντέλου την αναλαμβάνει εξολοκλήρου το λογισμικό Simon. Για να γίνει αυτό ο χρήστης ενεργοποιεί μία διαδικασία συγχρονισμού με το πάτημα ενός κουμπιού.

- **Πραγματοποίηση εκπαίδευσης ακουστικού μοντέλου**

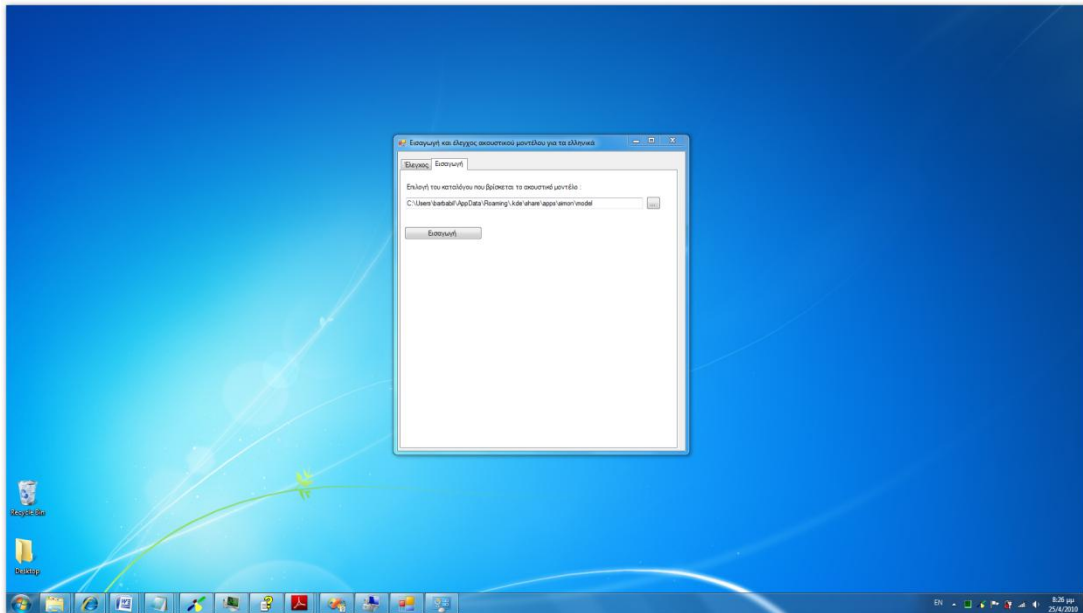
Η εκπαίδευση του ακουστικού μοντέλου στο λογισμικό Simon δεν συμπεριλαμβάνει τίποτε άλλο παρά την επαναληπτική ηχογράφηση των εντολών που μας ενδιαφέρουν (ή παραλλαγών τους) και την ενεργοποίηση της διαδικασίας συγχρονισμού με το πάτημα ενός κουμπιού, όπως περιγράφηκε στις προηγούμενες παραγράφους.

- **Πραγματοποίηση ενσωμάτωσης και ελέγχου ακουστικού μοντέλου**

Τα φυσικά αρχεία που αποτελούν το ακουστικό αλλά και το γλωσσικό μοντέλο το οποίο δημιουργήθηκε ή ενημερώθηκε με τις προηγούμενες διαδικασίες, βρίσκονται σε τελείως διαφορετικό κατάλογο από αυτόν που γνωρίζει η εφαρμογή μας. Η ενσωμάτωσή του αποτελεί μία διαδικασία που παρέχεται από μία συνιστώσα (component) της εφαρμογής η οποία απλώς αντιγράφει τα κατάλληλα αρχεία που αποτελούν τα μοντέλα στους καταλόγους

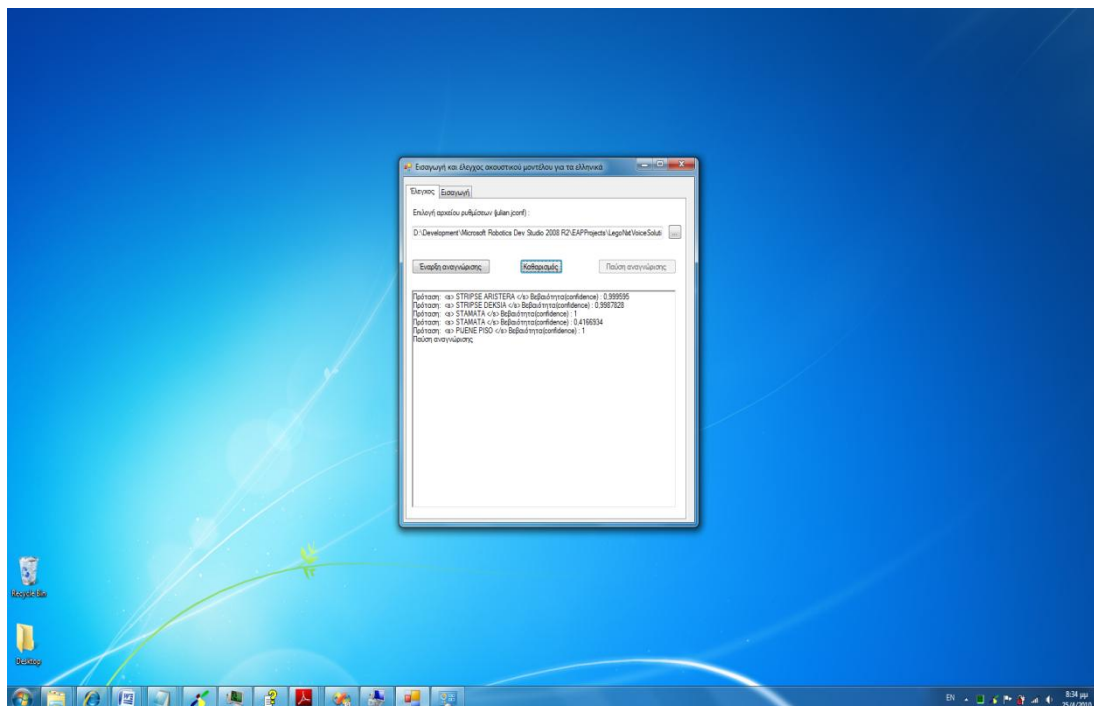


που αναγνωρίζει η εφαρμογή. Στην Εικόνα 22 βλέπουμε την διεπαφή της εφαρμογής που παρέχει αυτές τις λειτουργίες :



Εικόνα 21: Ενσωμάτωση ακουστικού και γλωσσικού μοντέλου στην εφαρμογή

Ο έλεγχος του ακουστικού μοντέλου αποτελεί την τελευταία διαδικασία που έχει να κάνει με την επιβεβαίωση όλων των προηγούμενων λειτουργιών συντήρησης. Παρέχεται από την ίδια συνιστώσα της εφαρμογής που πραγματοποιεί και την ενσωμάτωση. Αφού συνδεθεί στον Η/Υ κατάλληλο μικρόφωνο για την εκφώνηση προτάσεων, ο χρήστης ξεκινάει την διαδικασία της αναγνώρισης και εκφωνεί τις λέξεις ή προτάσεις που επιθυμεί. Η εφαρμογή ανταποκρίνεται εμφανίζοντας τόσο τις λέξεις που αναγνωρίστηκαν όσο και έναν αριθμό c ($0 < c \leq 1$) που αποτελεί την βεβαιότητα (confidence) της αναγνώρισης. Στην παρακάτω Εικόνα 23 ο χρήστης ξεκινάει την αναγνώριση και εκφωνεί τις εντολές : “Στρίψε Αριστερά”, “Στρίψε δεξιά”, “Σταμάτα”, “Σταμάτα” (πολύ χαμηλή φωνή) και “Πήγαινε πίσω”. Η εφαρμογή, λόγω των περιορισμών όπως έχουν περιγραφεί στην ενότητα 5.2.6, παρουσιάζει με λατινικούς χαρακτήρες τις προτάσεις που εκφωνήθηκαν και την αντίστοιχη βεβαιότητα. Ο χρήστης όποτε επιθυμεί μπορεί να παύσει την αναγνώριση.



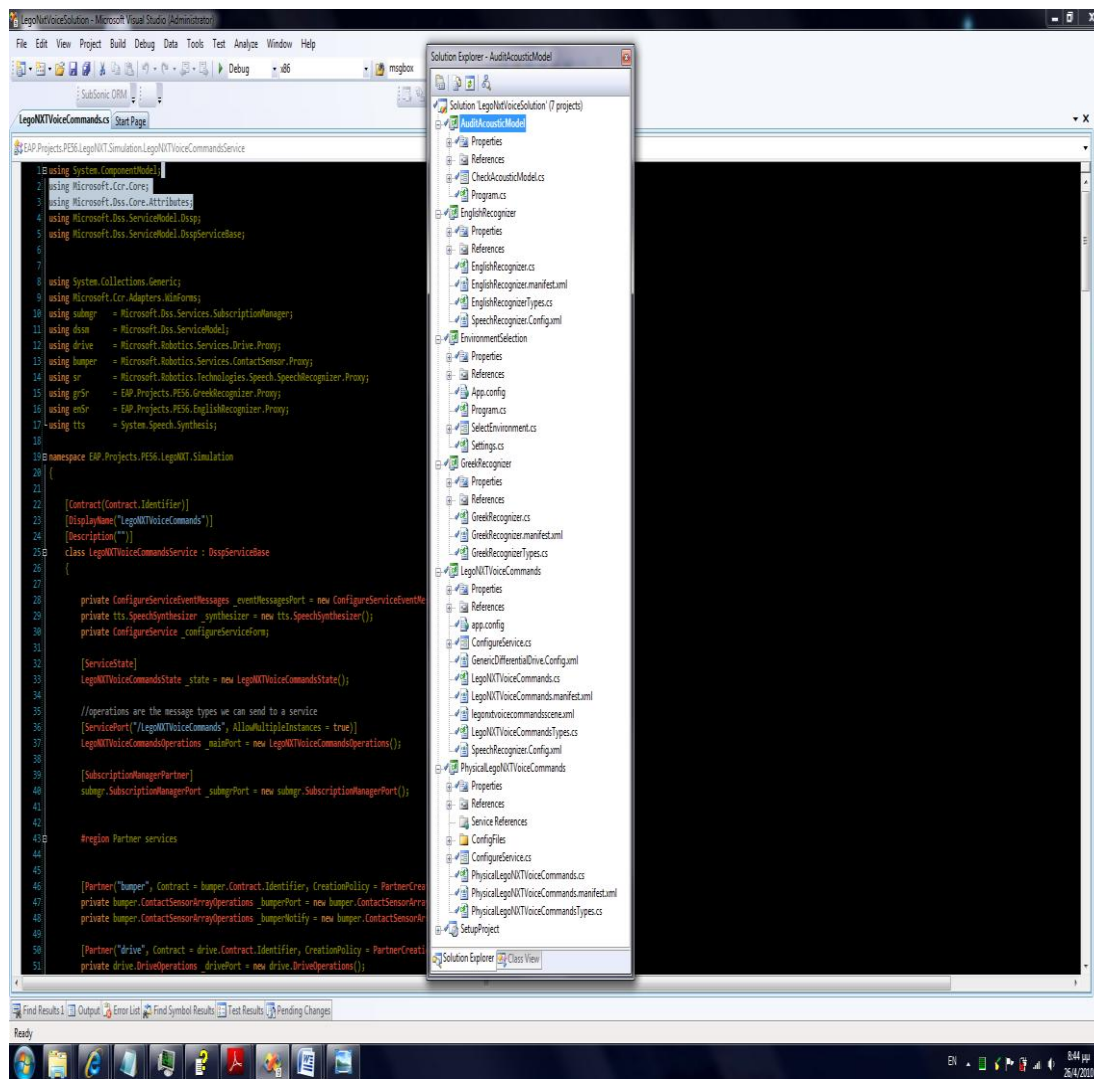
Εικόνα 22: Έλεγχος ακουστικού μοντέλου από την εφαρμογή

5.4.3 Φυσική απεικόνιση (*physical view*)

Στην φυσική απεικόνιση συμπεριλαμβάνεται τόσο η στατική ιεραρχία των αρχείων της φάσης ανάπτυξης, όσο και η οργάνωση και οι σχετικές τοποθεσίες των τελικών παραγόμενων φυσικών αρχείων (πχ εκτελέσιμα αρχεία, xml αρχεία κτλ). Στην φυσική άποψη συμπεριλαμβάνουμε και την κατασκευή του φυσικού ρομποτικού συστήματος.

- **Απεικόνιση ιεραρχικής οργάνωσης των αρχείων ανάπτυξης**

Στο επόμενη εικόνα (Εικόνα 24) βλέπουμε την οργάνωση όλων των αρχείων ανάπτυξης της εφαρμογής μέσα στο Visual Studio 2008. Κάθε ένα από αυτά τα αρχεία κώδικα βρίσκεται κάτω από ένα *Visual Studio project*, ενώ κάθε project αποτελεί είτε μία συνιστώσα ή υπηρεσία της συνολικής εφαρμογής. Όλα τα projects βρίσκονται κάτω από ένα solution που τα οργανώνει λογικά, βρίσκεται στην κορυφή της ιεραρχίας και έχει ονομαστεί ‘LegoNXTVoiceSolution’.



Εικόνα 23: Ιεραρχική οργάνωση των αρχείων ανάπτυξης στο Visual Studio 2008

Κάποια από τα projects που φαίνονται στην Εικόνα 24 αποτελούν απευθείας υλοποιήσεις των σεναρίων χρήσης της ενότητας 5.3.1. Παρακάτω δίνεται μία συνοπτική περιγραφή των λειτουργιών των projects αυτών, με την σειρά που βρίσκονται στην ιεραρχία οργάνωσης στο Visual Studio:

- AuditAcousticModel : Ελέγχει το ακουστικό μοντέλο και εισάγει τις ενημερωμένες εκδόσεις του ώστε να μπορούν να αξιοποιηθούν από την εφαρμογή. Αποτελεί υλοποίηση του σεναρίου χρήσης ‘1. Ενσωμάτωση και έλεγχος ακουστικού μοντέλου’
- EnglishRecognizer : Επιτελεί την αναγνώριση των αγγλικών φωνητικών εντολών μέσω της επικοινωνίας με την κατάλληλη μηχανή SRE. Αποτελεί υλοποίηση του σεναρίου χρήσης ‘9.Εκτέλεση φωνητικών εντολών στα αγγλικά’.



- EnvironmentSelection : Ρυθμίζει κατάλληλα όλες τις παραμέτρους που αφορούν την επιλογή περιβάλλοντος εργασίας από τον χρήστη (εικονικό ή πραγματικό περιβάλλον). Αποτελεί υλοποίηση του σεναρίου χρήσης ‘4.Επιλογή περιβάλλοντος’.
- GreekRecognizer : Επιτελεί την αναγνώριση στα ελληνικά μέσω της επικοινωνίας με την κατάλληλη μηχανή SRE. Αποτελεί υλοποίηση του σεναρίου χρήσης ‘8.Εκτέλεση φωνητικών εντολών στα ελληνικά’.
- LegoNXTVoiceCommands : Ελέγχει το εικονικό Lego NXT ρομπότ στο εικονικό περιβάλλον. Επιτελεί την αναγνώριση στα ελληνικά μέσω της επικοινωνίας με την κατάλληλη μηχανή SRE. Αποτελεί υλοποίηση των σεναρίων χρήσης ‘5.Εκκίνηση σε εικονικό περιβάλλον’ και ‘7.Ρύθμιση Παραμέτρων’
- PhysicalLegoNXTVoiceCommands : Ελέγχει το φυσικό Lego NXT ρομπότ στο πραγματικό περιβάλλον. Επιτελεί την αναγνώριση στα ελληνικά μέσω της επικοινωνίας με την κατάλληλη μηχανή SRE. Αποτελεί υλοποίηση των σεναρίων χρήσης ‘6.Εκκίνηση σε πραγματικό περιβάλλον’ και ‘7.Ρύθμιση Παραμέτρων’
- SetupProject : Περιλαμβάνει τις τελικές εκδόσεις (release builds) όλων των απαραίτητων αρχείων για την τελική εγκατάσταση της εφαρμογής. Το ίδιο αποτελεί το πρόγραμμα που διαχειρίζεται την εγκατάσταση του λογισμικού.

• Απεικόνιση τελικών παραγόμενων αρχείων

Παρακάτω παραθέτονται μόνο τα πιο σημαντικά αρχεία. Όλες οι τοποθεσίες (paths) των φυσικών αρχείων είναι σχετικές με βάση την τοποθεσία που βρίσκεται εγκατεστημένο το Microsoft Robotics Studio. Αυτό συμβαίνει για να βρίσκει η εφαρμογή όλα τα απαραίτητα αρχεία που αφορούν δομικά στοιχεία του MRDS ανεξάρτητα με την τοποθεσία της εγκατάστασής του σε κάθε H/Y.

Πίνακας 13: Στοιχεία εκτελέσιμου εκκίνησης εφαρμογής

Σχετική τοποθεσία (relative path)	EAPPProjects\LegoNxtVoiceSolution\Binaries\
Ονόματα αρχείων	EnvironmentSelection.exe
Περιγραφή	Το συγκεκριμένο εκτελέσιμο εκκινεί την κυρίως εφαρμογή που καθοδηγεί με φωνητικές εντολές το εκπαιδευτικό ρομπότ Lego NXT μέσα στο εικονικό ή το πραγματικό περιβάλλον.

**Πίνακας 14:** Στοιχεία κύριων εκτελέσιμων αρχείων εφαρμογής

Σχετική τοποθεσία (relative path)	EAPPProjects\LegoNxtVoiceSolution\Binaries\
Ονόματα αρχείων	EnvironmentSelection.exe
	EnglishRecognizer.Y2010.M03.dll
	EnglishRecognizer.Y2010.M03.Proxy.dll
	EnglishRecognizer.Y2010.M03.Transform.dll
	GreekRecognizer.Y2010.M03.dll
	GreekRecognizer.Y2010.M03.Proxy.dll
	GreekRecognizer.Y2010.M03.Transform.dll
	LegoNXTVoiceCommands.Y2010.M03.dll
	LegoNXTVoiceCommands.Y2010.M03.Proxy.dll
	LegoNXTVoiceCommands.Y2010.M03.Transform.dll
	PhysicalLegoNXTVoiceCommands.Y2010.M04.dll
	PhysicalLegoNXTVoiceCommands.Y2010.M04.Proxy.dll
	PhysicalLegoNXTVoiceCommands.Y2010.M04.Transform.dll
Περιγραφή	Τα εκτελέσιμα αποτελούν την κυρίως ρομποτική εφαρμογή.

Πίνακας 15: Στοιχεία εκτελέσιμων αρχείων μηχανής Julius

Σχετική τοποθεσία (relative path)	EAPPProjects\LegoNxtVoiceSolution\Binaries\Julius\Binaries\ forWindows_folder
Ονόματα αρχείων	Julius.dll
	sent.dll
Περιγραφή	Αποτελεί την υλοποίηση της μηχανής αναγνώρισης Julius [i20] για τα ελληνικά

Πίνακας 16: Στοιχεία εκτελέσιμου διεπαφής με τη μηχανή Julius

Σχετική τοποθεσία (relative path)	EAPPProjects\LegoNxtVoiceSolution\Binaries\
Ονόματα αρχείων	LapsAPI.dll
Περιγραφή	Αποτελεί την υλοποίηση του API επικοινωνίας με τη μηχανή Julius. Είναι η μέρος της υλοποίησης του του ερευνητικού



	project ανοικτού κώδικα Coruja [i19]
--	--------------------------------------

Πίνακας 17: Στοιχεία αρχείων παραμετροποίησης Julius

Σχετική τοποθεσία (relative path)	EAPPProjects\LegoNxtVoiceSolution\VoxForge - Julius Configuration\
Ονόματα αρχείων	julian.jconf model.grammar, model.voca
Περιγραφή	<p>4. Το αρχείο ‘julian.jconf’ είναι ένα αρχείο παραμετροποίησης (configuration file) μέσω του οποίου ρυθμίζονται όλες οι παράμετροι για να λειτουργήσει η μηχανή Julius</p> <p>5. Στον υποκατάλογο ‘Grammar’ έχει καθοριστεί η γραμματική (αρχεία model.grammar, model.voca) που αποτελεί το σύνολο των κανόνων που οριοθετούν τις εντολές μας και οι αντίστοιχες μετατροπές τους (αρχεία model.dfa, model.dict) στην μορφή που κατανοεί η μηχανή Julius [i20]</p> <p>6. Στον υποκατάλογο ‘acoustic_model_files_build726’ υπάρχει το τελικό παραγόμενο ακουστικό μοντέλο με την μορφή του αρχείου “hmmdefs” (και του βοηθητικού “tiedlist”)</p> <p>Η συγκεκριμένη δομή παρέχεται με την μορφή ανοικτού κώδικα από τον δικτυακό τόπο Voxforge [i21]</p>

Πίνακας 18: Στοιχεία εκτελέσιμου ελέγχου του ακουστικού μοντέλου

Σχετική τοποθεσία (relative path)	LegoNxtVoiceSolution\Binaries\
Ονόματα αρχείων	AuditAcousticModel.exe
Περιγραφή	Αποτελεί μέρος της εφαρμογής για τον έλεγχο και την ενσωμάτωση του ακουστικού μοντέλου



- **Κατασκευή φυσικού Lego NXT μοντέλου**

Για την ικανοποίηση της απαίτησης R2 για την κατασκευή του φυσικού ρομπότ, μπορούμε να έχουμε διάφορες προσεγγίσεις καθώς υπάρχουν αξιόλογες αναφορές στην βιβλιογραφία [b12] αλλά και στο διαδίκτυο που αφορούν αποκλειστικά στην συναρμολόγηση μοντέλων με βάση το Lego MINDSTORMS NXT.

- **Αρχική προσέγγιση κατασκευής μοντέλου**

Το πρώτο φυσικό ρομπότ που κατασκευάσαμε αποτέλεσε μία παραλλαγή του μοντέλου ExploroBot [b12] του οποίου το σύστημα οδήγησης αποτελείται δύο βασικούς τροχούς με αντίστοιχους μηχανισμούς σερβοκινητήρων και έναν βοηθητικό τροχό που απλώς υποστηρίζει την συνολική κίνηση. Ωστόσο έγινε αντιληπτό ότι, αν και το μοντέλο που κατασκευάστηκε ήταν πλήρως λειτουργικό, ήταν δύσκολο να παρέχουμε τον απαιτούμενο βαθμό τυποποίησης με την έννοια των επακριβώς καθορισμένων βημάτων συναρμολόγησης από τα απλά μέρη στα πιο σύνθετα. Κάτι τέτοιο σε επαρκή λεπτομέρεια θα απαιτούσε πολύ χρόνο και την χρήση συγκεκριμένου λογισμικού μοντελοποίησης (virtual modeling) – όπως τα LDraw [i22], MLCAD [i23] κτλ. Το πρόβλημα αυτό ενώ δεν είναι καθόλου απαγορευτικό για την λειτουργία του ίδιου του ρομπότ ή του συνολικού συστήματος, όμως θέτει την κατασκευή του στην υποκειμενική κρίση του κάθε ενδιαφερόμενου, πράγμα όχι επιθυμητό για τους συνολικούς στόχους της εργασίας.

- **Τελική προσέγγιση κατασκευής μοντέλου**

Έτσι, η νέα προσέγγιση για την κατασκευή του τελικού μοντέλου, βασίστηκε στις παρεχόμενες εκπαιδευτικές υποδομές της ίδιας της εταιρίας Lego η οποία σε συνεργασία με άλλους φορείς [i16] [i24], έχει σαν σκοπό την εισαγωγή τόσο των μαθητών αλλά και του οποιουδήποτε ενδιαφερόμενου στους επιστημονικούς τομείς της ρομποτικής, της μηχανικής και των μαθηματικών με έναν παιδαγωγικό, πρακτικό και ταυτόχρονα ελκυστικό τρόπο [i16].

Το τελικό μοντέλο ακολουθεί τις πλήρως τυποποιημένες και εύκολες οδηγίες συγκεκριμένου εκπαιδευτικού υλικού της Lego [i16], επανασυναρμολογείται εύκολα, έχει την επιζητούμενη απλότητα και είναι εγγυημένα λειτουργικό είτε ως αυτόνομη μονάδα είτε στα πλαίσια των απαιτήσεων της εργασίας.

Στην Εικόνα 24 έχουμε την εμπρόσθια άποψη του τελικού μοντέλου :



Εικόνα 24: Εμπρόσθια άποψη του τελικού Lego NXT μοντέλου

Στην αμέσως επόμενη εικόνα (Εικόνα 25) έχουμε μία πλάγια άποψη του τελικού ρομποτικού μοντέλου :



Εικόνα 25: Πλάγια άποψη του τελικού Lego NXT μοντέλου

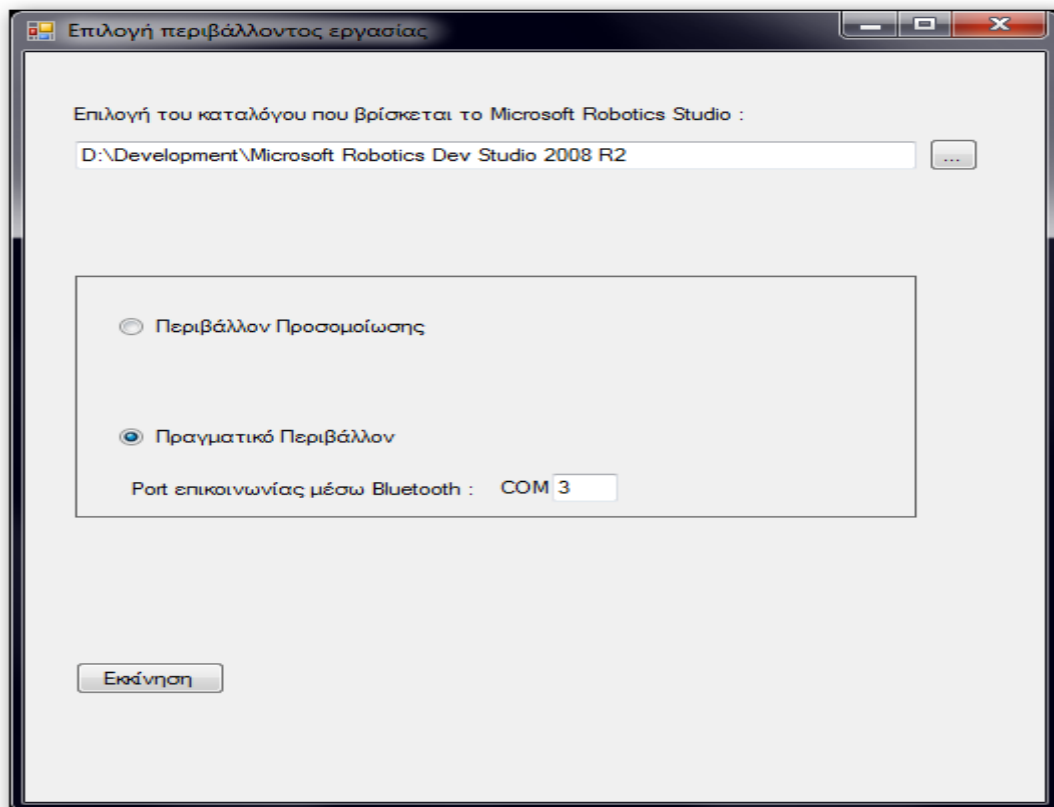


6. Αποτελέσματα

Σε αυτή την ενότητα παρακολουθούμε τα εφαρμοσμένα αποτελέσματα της εργασίας, δηλαδή την έμπρακτη λειτουργία της εφαρμογής φωνητικού ελέγχου. Σε αυτά τα πλαίσια πρώτα γίνεται μία πολύ σύντομη παρουσίαση της εφαρμογής και έπειτα παρουσιάζονται τα αποτελέσματα κάποιων εμπειρικών δοκιμών ως πρώτη αποκόμιση ενδεικτικών στοιχείων για τα ποσοστά επιτυχίας της αναγνώρισης των φωνητικών εντολών με βάση τα υπάρχοντα φωνητικά και γλωσσικά μοντέλα.

6.1 Παρουσίαση εφαρμογής

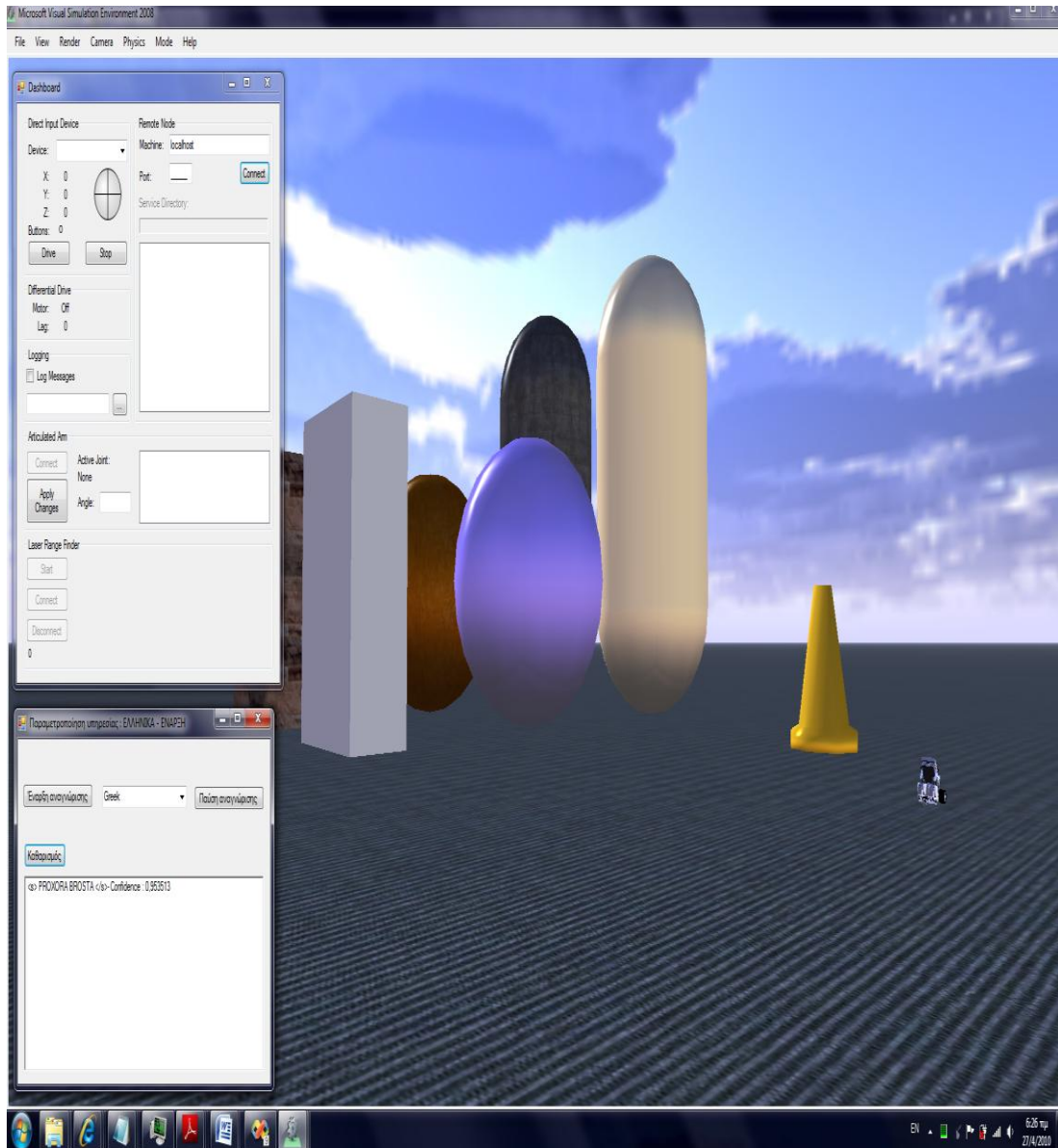
Αρχικά η εφαρμογή παρουσιάζει μία πολύ απλή διεπαφή στην οποία ο χρήστης επιλέγει εάν θα εργαστεί στο εικονικό ή στο πραγματικό περιβάλλον και άρα εάν θα δώσει φωνητικές εντολές στο εικονικό ή στο πραγματικό ρομπότ. Στην Εικόνα 26 βλέπουμε ότι εάν ο χρήστης επιλέξει να δώσει εντολές στο πραγματικό Lego NXT ρομπότ, η εφαρμογή πρέπει να γνωρίζει και την θύρα επικοινωνίας μέσω Bluetooth :



Εικόνα 26: Αρχική οθόνη εφαρμογής για την επιλογή περιβάλλοντος



Έστω ότι ο χρήστης όμως επιλέγει να δώσει εντολές στο εικονικό ρομπότ. Τότε εκκινείται το περιβάλλον προσομοίωσης μαζί με κάποια προαιρετικά χειριστήρια και την διεπαφή μέσω της οποίας ο χρήστης επιλέγει εάν θα εκφωνήσει εντολές στα ελληνικά ή στα αγγλικά.



Εικόνα 27: Εκτέλεση φωνητικών εντολών σε εικονικό περιβάλλον

Ο χρήστης έχοντας συνδέσει ένα απλό μικρόφωνο μπορεί να αρχίσει να εκφωνεί τις εντολές καθοδήγησης στα ελληνικά ή στα αγγλικά και το εικονικό ρομπότ εκτελεί τις αντίστοιχές κινήσεις. Όταν πρόκειται για ελληνικές εντολές η διεπαφή τις εμφανίζει με λατινικούς χαρακτήρες λόγω του περιορισμού που περιγράψαμε στην ενότητα 5.2.6. Το ίδιο ακριβώς αποτέλεσμα όσον αφορά στην πλοήγηση έχουμε και με το πραγματικό Lego NXT ρομπότ



6.2 Δοκιμές αναγνώρισης φωνητικών εντολών

Οι δοκιμές που πραγματοποιήθηκαν υλοποιούν τρία σενάρια αναγνώρισης φωνητικών εντολών : στα δύο πρώτα σενάρια εκφωνούνται οι γνωστές εντολές πλοήγησης στο περιβάλλον ενός ήσυχου δωματίου και στο τυπικό περιβάλλον ενός δωματίου. Στο τρίτο σενάριο συμμετέχει μόνο ο κύριος εκπαιδευτής του μοντέλου και οι λέξεις που εκφωνούνται είναι μη-προδιεγραμμένες δηλαδή δεν έχουν καμία σχέση ούτε με τις γραμματικές για τα ελληνικά και στα αγγλικά, ούτε με το πλαίσιο της εφαρμογής. Οι ομιλητές, ο ‘ΟΜΙΛΗΤΗΣ Α’ και ‘ΟΜΙΛΗΤΗΣ Β’ αντίστοιχα, είναι άτομα διαφορετικού φύλλου. Ο ‘ΟΜΙΛΗΤΗΣ Α’ είναι ο κύριος εκπαιδευτής του ελληνικού και αγγλικού ακουστικού μοντέλου, ενώ ο ‘ΟΜΙΛΗΤΗΣ Β’ έχει πολύ μικρή συνεισφορά στην εκπαίδευσή τους. Επιπλέον ιδιαιτερότητα αποτελεί το ότι ο ‘ΟΜΙΛΗΤΗΣ Β’ δεν έχει συμμετάσχει καθόλου στην εκπαίδευση της τελευταίας εντολής (εντολή ‘Σταμάτα’).

Κάθε ένας από τους ομιλητές επαναλάμβανε τις περιγραφόμενες εντολές συγκεκριμένο αριθμό φορών τόσο στα ελληνικά όσο και στα αγγλικά. Όσον αφορά στις ελληνικές εντολές, αυτό πραγματοποιήθηκε μέσα από την διεπαφή που παρέχει η εφαρμογή και χρησιμοποιείται για τον έλεγχο του ακουστικού μοντέλου. Όσον αφορά στις αγγλικές εντολές, χρησιμοποιήθηκε η ίδια συνιστώσα (component) που καλεί η εφαρμογή για την αναγνώριση των αγγλικών δηλαδή ο Speech Recognizer. Ο Speech Recognizer είναι ενσωματωμένος στο λειτουργικό σύστημα των Windows 7 και εσωτερικά επικοινωνεί με το SAPI.

Τέλος τα ποσοστά επιτυχίας αποτελούν τους μέσους όρους που προέκυψαν από τα αθροίσματα της βεβαιότητας (confidence) αναγνώρισης, όπως αυτά επιστρέφονταν από τις αντίστοιχες μηχανές αναγνώρισης (SREs) για τα ελληνικά και τα αγγλικά αντίστοιχα. Όταν υπάρχει κάποιο ποσοστό αναγνώρισης αλλά η εντολή που αναγνωρίζεται είναι διαφορετική από αυτή που εκφωνήθηκε, θεωρούμε το ποσοστό αυτό ως μηδενικό.

6.2.1 Σενάριο 1 : Περιβάλλον ήσυχου δωματίου

Στο πρώτο σενάριο οι ομιλητές εκφωνούν τις τυπικές εντολές πλοήγησης του ρομπότ σε ένα ήσυχο δωμάτιο.

Πίνακας 19: Ποσοστά επιτυχίας πρώτου ομιλητή σε περιβάλλον ήσυχου δωματίου

ΟΜΙΛΗΤΗΣ Α



Ελληνικά			Αγγλικά		
Εντολή	Επαναλήψεις	Ποσοστό επιτυχίας	Εντολή	Επαναλήψεις	Ποσοστό επιτυχίας
Προχώρα μπροστά	10	0,99999997	Move forwards	10	0,854179
Πήγαινε πίσω	10	0,99698761	Move backwards	10	0,947367
Στρίψε δεξιά	10	0,699997	Turn right	10	0,740631
Στρίψε αριστερά	10	0,49999999	Turn left	10	0,911437
Σταμάτα	10	0,6999985	Stop	10	0,925674
		3,896983			4,379288

Πίνακας 20: Ποσοστά επιτυχία δεύτερου ομιλητή σε περιβάλλον ήσυχου δωματίου

ΟΜΙΑΗΤΗΣ Β					
Ελληνικά			Αγγλικά		
Εντολή	Επαναλήψεις	Ποσοστό επιτυχίας	Εντολή	Επαναλήψεις	Ποσοστό επιτυχίας
Προχώρα μπροστά	10	0,99999517	Move forwards	10	0,673645
Πήγαινε πίσω	10	0,9790831	Move backwards	10	0,518139
Στρίψε δεξιά	10	0,37397784	Turn right	10	0,090981
Στρίψε αριστερά	10	0,26031384	Turn left	10	0,730202
Σταμάτα	10	0,09823356	Stop	10	0,935964
		2,711604			2,948931

6.2.2 Σενάριο 2 : Τυπικό περιβάλλον δωματίου

Στο δεύτερο σενάριο οι ομιλητές εκφωνούν τις τυπικές εντολές πλοήγησης του ρομπότ σε ένα δωμάτιο όπου υπάρχουν κάποιες συνηθισμένες ηχητικές πηγές εκτός από τους ομιλητές (π.χ. ραδιόφωνο ή τηλεόραση ανοιχτή σε κανονικά επίπεδα έντασης).

Πίνακας 21: Ποσοστά επιτυχίας πρώτου ομιλητή σε τυπικό περιβάλλον δωματίου

ΟΜΙΑΗΤΗΣ Α					
Ελληνικά			Αγγλικά		



Εντολή	Επαναλήψεις	Ποσοστό επιτυχίας		Εντολή	Επαναλήψεις	Ποσοστό επιτυχίας
Προχώρα μπροστά	10	0,478159		Move forwards	10	0,947888
Πήγαινε πίσω	10	0,712868		Move backwards	10	0,946572
Στρίψε δεξιά	10	0		Turn right	10	0,625394
Στρίψε αριστερά	10	0,096627		Turn left	10	0,921918
Σταμάτα	10	0,783547		Stop	10	0,945806
		2,071201				4,387578

Πίνακας 22: Ποσοστά επιτυχίας δεύτερου ομιλητή σε τυπικό περιβάλλον δωματίου

ΟΜΙΛΗΤΗΣ Β					
Ελληνικά			Αγγλικά		
Εντολή	Επαναλήψεις	Ποσοστό επιτυχίας	Εντολή	Επαναλήψεις	Ποσοστό επιτυχίας
Προχώρα μπροστά	10	0,999994	Move forwards	10	0
Πήγαινε πίσω	10	0,572243	Move backwards	10	0,090877
Στρίψε δεξιά	10	0,994112	Turn right	10	0
Στρίψε αριστερά	10	0,691648	Turn left	10	0,086951
Σταμάτα	10	0,29776	Stop	10	0,269217
		3,555757			0,447045

6.2.2 Σενάριο 3 : Εκφορά μη-προδιεγραμμένων λέξεων

Στο τρίτο σενάριο ο κύριος εκπαιδευτής εκφωνεί τις εντολές σε περιβάλλον ενός ήσυχου δωματίου. Ισχύουν όλα όσα προαναφέραμε για τα ποσοστά εκτός από μία διαφοροποίηση : όταν η μηχανή δεν αναγνωρίσει την λέξη αυτό θεωρείται επιτυχία, καθώς καμία λέξη δεν ανήκει στην ορισμένη γραμματική.



Πίνακας 23: Ποσοστά επιτυχίας στα ελληνικά με μη-προδιεγραμμένες λέξεις

ΟΜΙΛΙΤΗΣ Α			
Ελληνικά			
Εκφερόμενη Λέξη	Επαναλήψεις	Λέξη που αναγνωρίστηκε	Ποσοστό επιτυχίας
Δομή	5	Σταμάτα Προχώρα πίσω Σταμάτα Προχώρα πίσω Προχώρα πίσω Σταμάτα	0
Κέλυφος	5	Πήγαινε μπροστά Σταμάτα Σταμάτα Σταμάτα Πήγαινε πίσω	0
Βήμα	5	Σταμάτα Σταμάτα Σταμάτα Σταμάτα Σταμάτα	0
Ανεβαίνω	5	Σταμάτα Σταμάτα Σταμάτα Σταμάτα Σταμάτα	0

Πίνακας 24: Ποσοστά επιτυχίας στα αγγλικά με μη-προδιεγραμμένες λέξεις

ΟΜΙΛΙΤΗΣ Α			
Αγγλικά			
Εκφερόμενη Λέξη	Επαναλήψεις	Τι αναγνωρίστηκε	Ποσοστό επιτυχίας
Dimension	5	- - - - -	1
Shell	5	Stop Stop Stop Stop Stop	0
Country	5	- - - - -	1
Test	5	- Stop Stop - - -	0,666667



6.2.3 Συνοπτική ανάλυση αποτελεσμάτων

Οι διαφορές μεταξύ των δύο ακουστικών μοντέλων, δηλαδή αυτού που κατασκευάσαμε για τα ελληνικά και αυτού που υπάρχει ενσωματωμένου στο λειτουργικό σύστημα Windows 7, όσον αφορά στις ώρες εκπαίδευσης και στον αριθμό των διαφορετικών χρηστών που έχουν συνεισφέρει με φωνητικές εγγραφές είναι δεδομένα τεράστιες. Το δεδομένο αυτό, καθώς και το γεγονός ότι οι παραπάνω μετρήσεις δεν έγιναν κάτω από ένα ελεγχόμενο περιβάλλον δεν μας επιτρέπει να κάνουμε μία εμπειριστατωμένη συγκριτική ανάλυση. Ωστόσο μπορούμε να διαπιστώσουμε τα εξής :

- Όσον αφορά το Σενάριο 1 του ήσυχου δωματίου βλέπουμε ότι τα ποσοστά επιτυχίας του ομιλητή A που είναι και ο εκπαιδευτής του ακουστικού μοντέλου για τα ελληνικά, είναι μεγαλύτερα. Το ίδιο ισχύει και για τα αγγλικά καθώς τα Windows 7 υποστηρίζουν λειτουργίες εκπαίδευσης του ενσωματωμένου σε αυτά ακουστικού μοντέλου και άρα πάλι ο ομιλητής A έχει το πλεονέκτημα που αποτυπώνεται.
- Στο Σενάριο 2 που αφορά τυπικό περιβάλλον δωματίου ωστόσο τα πράγματα αλλάζουν. Βλέπουμε ότι ο θόρυβος μπορεί να επηρεάσει κατά πολύ τα αποτελέσματα, καθώς στα ελληνικά ο ομιλητής B έχει καλύτερα αποτελέσματα ενώ και στα αγγλικά τα αποτελέσματα μεταξύ των δύο ομιλητών είναι παραπλήσια
- Τέλος, στο Σενάριο 3 που εκφωνούνται μη προδιαγραμμένες λέξεις βλέπουμε ότι η μηχανή Julius που χρησιμοποιούμε για το ακουστικό μοντέλο των ελληνικών, αποτυγχάνει τελείως καθώς δεν απορρίπτει καμία εντολή αλλά προσπαθεί να ταιριάξει την οποιαδήποτε εκφερόμενη εντολή με κάποια από αυτές που γνωρίζει. Η μηχανή των Windows φαίνεται ότι υποστηρίζει τις διαδικασίες απόρριψης λέξεων όταν η εκφορά τους διαφέρει σημαντικά. Όταν η εκφορά όμως των λέξεων είναι παραπλήσια και η μηχανή αναγνώρισης των Windows αποτυγχάνει τελείως.



7. Συμπεράσματα, περιορισμοί και επεκτάσεις

7.1 Ανασκόπηση

Η παρούσα εργασία αποτέλεσε μία προσπάθεια προς την κατεύθυνση του συνδυασμού ρομποτικών εκπαιδευτικών συστημάτων και τεχνολογιών φωνής με χρήση σύγχρονου λογισμικού, με σκοπό την ανάπτυξη εφαρμογής φωνητικού ελέγχου στα ελληνικά και στα αγγλικά. Για την θεμελίωση της προσπάθειας αυτής, αναφερθήκαμε καταρχήν στα συστήματα επεξεργασίας φωνής περιγράφοντας βασικές συνιστώσες των συστημάτων αναγνώρισης φωνής (Speech Recognition Systems), των συστημάτων σύνθεσης φωνής από κείμενο (Text-to-Speech Systems) αλλά και του συνδυασμού τους σε ποιά ολοκληρωμένα συστήματα (Spoken Language Understanding Systems) όταν απαιτείται η κατανόηση του προφορικού λόγου. Έπειτα, συνεχίσαμε με την περιγραφή της λειτουργίας επεξεργασίας φωνής όπως υλοποιείται σε περιβάλλον Windows με κύριο συστατικό την προγραμματιστική διεπαφή Speech Recognition API (SAPI). Μετά αναφερθήκαμε στα ρομποτικά συστήματα, τόσο στα γενικότερα πεδία εφαρμογών τους όσο και στον ορισμό του ρομπότ, καθώς στην επιστημονική κοινότητα δεν υπάρχει ακόμα συμφωνία ως προς το θέμα αυτό και ακολουθήσαμε την προσέγγιση του ορισμού του ευφυούς ρομπότ. Συνεχίσαμε με την περιγραφή του ιστορικού αλλά κυρίως των βασικών συστατικών του εκπαιδευτικού ρομπότ Lego Mindstorms NXT. Στην επόμενη ενότητα περιγράψαμε τα βασικά χαρακτηριστικά αλλά και εσωτερικές λειτουργίες της ρομποτικής πλατφόρμας ανάπτυξης λογισμικού Microsoft Robotics Studio η οποία και αποτέλεσε την επιλογή για την ανάπτυξη της εφαρμογής μας. Αμέσως μετά περιγράψαμε την διαδικασία ανάπτυξης, κατασκευής και εξομοίωσης του συστήματος, ξεκινώντας από την επιλογή ενός πλαισίου εύκαμπτης (agile) μεθοδολογίας όπως είναι το Microsoft Solutions Framework for Agile. Περιγράψαμε ακολούθως τις απαιτήσεις του συστήματος και καταλήξαμε στην αρχιτεκτονική του αποτύπωση μέσα από τέσσερις συγκεκριμένες όψεις του συστήματος. Οι όψεις αυτές αφορούσαν στην ίδια την εφαρμογή φωνητικού ελέγχου σε περιβάλλον προσομοίωσης και σε πραγματικό περιβάλλον, στην κατασκευή και στην εκπαίδευση του ακουστικού μοντέλου και τέλος στην κατασκευή του φυσικού ρομπότ. Έπειτα δείξαμε τα αποτελέσματα της συνολικής προσπάθειας παρουσιάζοντας πολύ σύντομα την εφαρμογή κατά την εκτέλεσής της και τέλος πραγματοποιήσαμε κάποιες εμπειρικές μετρήσεις της ποσοστιαίας επιτυχίας των φωνητικών εντολών με βάση τυπικά σενάρια.



7.2 Επίτευξη στόχων

Οι εφαρμοσμένοι στόχοι της εργασίας εκτιμούμε ότι πραγματοποιήθηκαν πλήρως. Ξεπεράσαμε την αδυναμία αναγνώρισης ελληνικής ομιλίας που έχουν τα Windows, δημιουργώντας ένα καινούργιο ακουστικό μοντέλο και στη συνέχεια το εκπαιδεύσαμε με το λογισμικό ανοικτού κώδικα Simon, διαπιστώνοντας το υψηλό επίπεδο υπηρεσιών και λειτουργιών που μπορεί να παρέχει το ανοικτό λογισμικό στην σημερινή εποχή. Χρειάστηκε να συνδυάσουμε ένα πλήθος από ετερογενείς τεχνολογίες και συνιστώσες και με βάση τις δυνατότητες που παρέχει το Microsoft Robotics Studio αναπτύξαμε μία λειτουργική εφαρμογή καθοδήγησης του εκπαιδευτικού ρομπότ Lego NXT τόσο σε εικονικό όσο και σε πραγματικό περιβάλλον, ενώ παράλληλα κατασκευάσαμε και συγκεκριμένο φυσικό ρομποτικό μοντέλο του Lego NXT. Δείξαμε τέλος, ότι η εξέλιξη της τεχνολογίας καθιστά δυνατή την ανάπτυξη συνεργιών μεταξύ ρομποτικών συστημάτων και τεχνολογιών φωνής με σχετικά μικρό κόστος και μη-τετριμμένα αποτελέσματα.

7.3 Περιορισμοί

Ωστόσο όπως σε κάθε προσπάθεια, υπήρξαν και περιοριστικοί παράγοντες. Το εξωτερικό λογισμικό δημιουργίας και εκπαίδευσης κώδικα Simon δημιούργησε προβλήματα με την αδυναμία συνεργασίας με τους ελληνικούς χαρακτήρες. Η πλατφόρμα ανάπτυξης του MRDS παράλληλα την ισχυρή παρουσία και δυναμική της έδειξε ότι έχει αρκετά μεγάλη καμπύλη εκμάθησης λόγω μίας διαφορετικής φιλοσοφίας και προγραμματιστικής προσέγγισης που ακολουθεί. Τέλος, η μηχανή αναζήτησης Julius που χρησιμοποιήσαμε για τα ελληνικά, έδειξε ότι σε τυπικά περιβάλλοντα με συνηθισμένα επίπεδα θορύβου δεν λειτουργεί πολύ ικανοποιητικά.

7.4 Πιθανές επεκτάσεις

Επεκτάσεις της εργασίας θα μπορούσαν να αποτελέσουν εφαρμογές οι οποίες δίνουν έμφαση στην ανάπτυξη και εξέλιξη συμπληρωματικής λογικής ελέγχου. Καθώς δείξαμε ότι το ακουστικό μοντέλο μπορεί να εμπλουτιστεί και να εκπαιδευτεί από διαφορετικούς ομιλητές, μπορούμε να έχουμε ανάπτυξη πολύπλοκων εφαρμογών φωνητικού ελέγχου. Για παράδειγμα, θα μπορούσαμε να αναπτύξουμε εφαρμογή μηχανικής μάθησης ώστε το ρομπότ να ‘μαθαίνει’ τις διαδρομές οι οποίες του υπαγορεύονται με τις φωνητικές εντολές. Θα μπορούσαμε επίσης να έχουμε εφαρμογές διαλόγου μεταξύ ανθρώπου και ρομπότ



έχοντας ως βάση ήδη υπάρχοντες μηχανισμούς λογισμικού ‘chatbox’ και την δυνατότητα ανταλλαγής αρχείων ήχου μεταξύ Η/Υ και του φυσικού ρομπότ κτλ

7.5 Προοπτικές τεχνολογικών πεδίων

Ενώ η σημερινή αξιοποίηση των τεχνολογιών αναγνώρισης, σύνθεσης και κατανόησης ήδη δίνει αποτελέσματα με εφαρμογές και υπηρεσίες, η εξέλιξή της φαίνεται ότι μπορεί να έχει σημαντική επίδραση σε πληθώρα πεδίων όπως η εκπαίδευση (πχ εκμάθηση γλωσσών), η οικιακή διαχείριση (πχ απομακρυσμένος έλεγχος έξυπνων συσκευών), τα συστήματα ασφαλείας κτλ

Τα ρομποτικά συστήματα από την άλλη, έχοντας ήδη δείξει αναμφισβήτητα αποτελέσματα σε τομείς όπως η βιομηχανία, η διαστημική έρευνα και η ιατρική, πραγματοποιούν τα τελευταία χρόνια τεχνολογικά άλματα δίνοντας δείγματα ενός επιστημονικού πεδίου που πραγματικά μπορεί να αλλάξει το τοπίο της σύγχρονης ζωής.

Με τον κατάλληλο συνδυασμό των παραπάνω τεχνολογικών τομέων μπορούμε να έχουμε όχι μόνο τα αποτελέσματα κάποιων ευφυών εφαρμογών και συστημάτων, αλλά και μια εξαιρετική δυναμική για δημιουργία πρωτοποριακών και ταυτόχρονα ουσιαστικών υπηρεσιών περισσότερο κοντά στις ανάγκες του σύγχρονου ανθρώπου στο πολύ άμεσο μέλλον.



Βιβλιογραφικές αναφορές

Βιβλία

- [b1] S. Morgan, ‘Programming Microsoft Robotics Studio’, Microsoft Press, 2008
- [b2] K. Jones, T. Taylor, ‘Professional Microsoft Robotics Developer Studio’, Wrox Programmer to Programmer, 2008
- [b3] Ι. Μπόρας, ‘Αυτόματος Τεμαχισμός Ψηφιακών Σημάτων Ομιλίας και Εφαρμογή στην Σύνθεση Ομιλίας, Αναγνώριση Ομιλίας και Αναγνώριση Γλώσσας’, Πανεπιστήμιο Πατρών, 2009
- [b4] R. Cole, V. Zue, W. Ward+, ‘Survey of the State of the Art in Human Language Technology’, Cambridge University Press, 1996
- [b5] Ε. Γιούρα, ‘Ανάπτυξη Συστήματος Αναγνώρισης Φωνημάτων για τις Ανάγκες της Αυτόματης Αναγνώρισης Γλώσσας’, Πανεπιστήμιο Πατρών, 2007
- [b6] Huang X., Acero A., Hon H-W., “Spoken Language Processing: A Guide to Theory, Algorithm and System Development”, Pearson Publication, New Jersey, 2001
- [b7] Π. Ζέρβας, ‘Μοντελοποίηση και Ψηφιακή Επεξεργασία Προσωδιακών Φαινομένων της Ελληνικής Γλώσσας με Εφαρμογή στην Σύνθεση Ομιλίας’, Πανεπιστήμιο Πατρών, 2007
- [b8] M.Gasperì, P.Hurbain+, ‘Extreme NXT – Extending the LEGO MINDSTORMS NXT to the Next Level’, Apress, 2007
- [b9] R. Murphy, ‘Introduction to AI Robotics’, The MIT Press, Cambridge, Massachusetts London, England, 2000
- [b10] Β. Βεσκούκης, ‘Τεχνολογία Λογισμικού Ι’, Ελληνικό Ανοικτό Πανεπιστήμιο, 2000
- [b11] P. Kruchten, ‘Architectural Blueprints—The “4+1” View Model of Software Architecture’, Rational Software Corp., 1995
- [b12] J.Kelly, ‘LEGO Mindstorms NXT – The Mayan Adventure’, Apress, 2006



[b13] B. Siciliano(Editor), Oussama Khatib(Editor), ‘Springer Handbook of Robotics’, Springer, 2008

[b14] A. Δημητρίου, ‘Αυτόματα και Τυπικές Γλώσσες’, Ελληνικό Ανοικτό Πανεπιστήμιο, 2002

[b15] J. Liberty, D. Xie, ‘Programming C# 3.0’ 5th edition, O’ Reilly, 2008

[b16] J. W.A. Traa, ‘Rational Unified Process vs. Microsoft Solutions Framework: A Comparative Study’, Erasmus University, Rotterdam, The Netherlands, 2006

Λιαδίκτυο

[i1] Microsoft Robotics, <http://www.microsoft.com/robotics/> (τελευταία ημερομηνία προσπέλασης : 06/04/2010)

[i2] MRDS 2008 R2 Datasheet, <http://go.microsoft.com/fwlink/?LinkID=133306> (τελευταία ημερομηνία προσπέλασης : 06/04/2010)

[i3] J.D. Meier, D. Hill, A. Horner+, Application Architecture Guide 2nd Edition, Microsoft, <http://www.microsoft.com/downloads/details.aspx?FamilyID=ce40e4e1-9838-4c89-a197-a373b2a60df2&displaylang=en> (τελευταία ημερομηνία προσπέλασης: 24/12/2009)

[i4] Microsoft Speech Technologies, <http://www.microsoft.com/speech/developers.aspx> (τελευταία ημερομηνία προσπέλασης : 06/04/2010)

[i5] Cambridge University and Microsoft Offer Speech Recognition Toolkit for Free, <http://www.microsoft.com/Presspass/press/2000/sept00/cambridgepr.msp> (τελευταία ημερομηνία προσπέλασης: 06/04/2010)

[i6] Speech API 5.3 – SR Engine Vendor Porting Guide, [http://msdn.microsoft.com/en-us/library/ms717034\(VS.85\).aspx#_Toc503606865](http://msdn.microsoft.com/en-us/library/ms717034(VS.85).aspx#_Toc503606865) (τελευταία ημερομηνία προσπέλασης : 06/04/2010)

[i7] Microsoft Speech API, http://en.wikipedia.org/wiki/Microsoft_Speech_API (τελευταία ημερομηνία προσπέλασης : 06/04/2010)



- [i8] Robert Brown, Exploring New Speech Recognition And Synthesis APIs in Windows Vista, <http://msdn.microsoft.com/en-us/magazine/cc163663.aspx> (τελευταία ημερομηνία προσπέλασης : 06/04/2010)
- [i9] Microsoft Robotics Developer Center, <http://msdn.microsoft.com/en-us/robotics/default.aspx>, (τελευταία ημερομηνία προσπέλασης : 07/04/2010)
- [i10] Windows Embedded, <http://www.microsoft.com/windowseembedded/en-us/default.mspx> (τελευταία ημερομηνία προσπέλασης : 07/04/2010)
- [i11] H. Nielsen, G. Chrysanthakopoulos, ‘Decentralized Software Services Protocol – DSSP/1.0’, Microsoft, <http://download.microsoft.com/download/5/6/B/56B49917-65E8-494A-BB8C-3D49850DAAC1/DSSP.pdf> (τελευταία ημερομηνία προσπέλασης: 08/04/2010)
- [i12] M.Cowling, R.Sitte, ‘Analysis of Speech Recognition Techniques for use in a Non-Speech Sound Recognition System’
<http://www.elec.uow.edu.au/staff/wysocki/dspcs/papers/004.pdf>, (τελευταία ημερομηνία προσπέλασης : 08/04/2010)
- [i13] A. Πρωτόπαπας, ‘Εισαγωγή στη Φωνητική’,
http://www.ilsp.gr/homepages/protopapas/pdf/Protopapas_phon-notes_2003.pdf
(τελευταία ημερομηνία προσπέλασης : 08/04/2010)
- [i14] Lego NXT Developer Kits, <http://mindstorms.lego.com/en-us/support/files/default.aspx> (τελευταία ημερομηνία προσπέλασης : 10/04/2010)
- [i15] Lego MINDSTORMS NXT Official Site, <http://mindstorms.lego.com/en-us/default.aspx> (τελευταία ημερομηνία προσπέλασης : 10/04/2010)
- [i16] NXT Downloads - Education NXT User Guide and Building Instructions,
http://www.lego.com/education/school/default.asp?locale=2057&pagename=dsnxt&l2id=3_2&l3id=3_2_5&l4id=3_2_5_3 (τελευταία ημερομηνία προσπέλασης : 10/04/2010)



[i17] University of Zurich - Department of Informatics, ‘Tutorial for Programming the Lego MINDSTORMS NXT’

http://legoengineering.com/images/stories/news/artificial_intelligence/TutorialMindstormsNXT.pdf (τελευταία ημερομηνία προσπέλασης : 10/04/2010)

[i18] ‘What is a robot ?’, CBC News,

<http://www.cbc.ca/news/background/tech/robotics/definition.html>, (τελευταία ημερομηνία προσπέλασης : 10/04/2010)

[i19] Research software for speech recognition in Brazilian Portuguese, <http://www.laps.ufpa.br/falabrasil/downloads.php>, (τελευταία ημερομηνία προσπέλασης: 17/03/2010)

[i20] Julius - Open-Source Large Vocabulary CSR Engine, http://julius.sourceforge.jp/en_index.php?q=index-en.html, (τελευταία ημερομηνία προσπέλασης : 17/03/2010)

[i21] Voxforge - Collection of transcribed speech for use with free and open-source recognition engines, <http://www.voxforge.org/>, (τελευταία ημερομηνία προσπέλασης: 16/03/2010)

[i22] LDraw.org – Centralized LDraw Resources, <http://www.ldraw.org/index.html>, (τελευταία ημερομηνία προσπέλασης: 12/04/2010)

[i23] MLCAD, http://www.lm-software.com/mlcad/e_default.htm, (τελευταία ημερομηνία προσπέλασης: 12/04/2010)

[i24] LEGO engineering, <http://www.legoengineering.com>, (τελευταία ημερομηνία προσπέλασης: 12/04/2010)

[i25] Πύλη για την Ελληνική γλώσσα, <http://www.greek-language.gr/greekLang/index.html>, (τελευταία ημερομηνία προσπέλασης: 21/04/2010)

[i26] L.Cuvillon, B. Bayle, ‘Introduction to Mechatronics – Discover robots with the Lego Mindstorms’, http://eavr.u-strasbg.fr/~bernard/education/ensps_1a/slides_mecatro_1a.pdf, (τελευταία ημερομηνία προσπέλασης: 23/04/2010)



- [i27] The International Phonetic Association, <http://www.langsci.ucl.ac.uk/ipa/>,
(τελευταία ημερομηνία προσπέλασης: 23/04/2010)
- [i28] Simon wiki, http://www.cyber-byte.at/wiki/index.php/Main_Page, (τελευταία
ημερομηνία προσπέλασης: 24/04/2010)
- [i29] .NET framework, http://en.wikipedia.org/wiki/.NET_Framework, (τελευταία
ημερομηνία προσπέλασης: 24/04/2010)
- [i30] Microsoft Visual Studio, http://en.wikipedia.org/wiki/Microsoft_Visual_Studio,
(τελευταία ημερομηνία προσπέλασης: 24/04/2010)
- [i31] J. Sutherland, ‘Agile Principles and Values’, [http://msdn.microsoft.com/en-us/library/dd997578\(v=VS.100\).aspx](http://msdn.microsoft.com/en-us/library/dd997578(v=VS.100).aspx), (τελευταία ημερομηνία προσπέλασης: 24/04/2010)
- [i32] ‘Manifesto for Agile Software Development’, <http://agilemanifesto.org/>, (τελευταία
ημερομηνία προσπέλασης: 24/04/2010)



Παραρτήματα

Παράρτημα Α: Ακρωνύμια / Συντομογραφίες

Application Programmatic Interface	API
Automatic Speech Recognition	ASR
Concurrency & Coordination Runtime	CCR
Decentralized Software Services	DSS
Decentralized Software Services Protocol	DSSP
Dialog Management	DM
Hidden Markov Model Toolkit	HTK
Hidden Markov Models	HMM
International Phonetic Alphabet	IPA
Microsoft Robotics Developer Studio	Microsoft RDS ή MRDS
Representational State Transfer	REST
Speech Recognition API	SAPI
Speech Recognition Engine	SR
Spoken Language Generation	SLG
Spoken Language Understanding	SLU
Text to Speech Synthesis	TTS
Visual Programming Language	VPL
Visual Simulation Environment	VSE
Ελληνικό Ανοικτό Πανεπιστήμιο	ΕΑΠ
Ηλεκτρονικός Υπολογιστής	Η/Υ



Παράρτημα Β: Ευρετήριο Σχημάτων

Σχήμα 1: Μηχανισμός παραγωγής ανθρώπινης ομιλίας [b3]	13
Σχήμα 2: Βασική δομή ενός συστήματος SR [b6]	16
Σχήμα 3: Παράδειγμα αυτόματου που αναγνωρίζει μία γλώσσα	17
Σχήμα 4: Βασική δομή ενός συστήματος TTS [b6]	19
Σχήμα 5: Βασική δομή ενός συστήματος SLU	22
Σχήμα 6: Επικοινωνία εφαρμογών με το SAPI	25
Σχήμα 7: Διάγραμμα μονάδων υλικού και εσωτερικής επικοινωνίας στο NXT brick	37
Σχήμα 8: Διαστρωμάτωση βασικών μονάδων στο MRDS [b1]	43
Σχήμα 9: Αρχιτεκτονική CCR	47
Σχήμα 10: Παράδειγμα σύνθεσης και επικοινωνίας υπηρεσιών	50
Σχήμα 11: Συνιστώσες μίας υπηρεσίας και τρόποι επικοινωνίας	51
Σχήμα 12: Διάγραμμα βασικών σεναρίων χρήσης	67
Σχήμα 13: Διάγραμμα πλαισίου λειτουργίας της αναγνώρισης των αγγλικών εντολών	80
Σχήμα 14: Διάγραμμα πλαισίου λειτουργίας της αναγνώρισης των ελληνικών εντολών	82
Σχήμα 15: Διαστρωμάτωση επιπέδων συστήματος	83

Παράρτημα Γ: Ευρετήριο Εικόνων

Εικόνα 1: Ρομπότ διάσωσης V2 [b13]	29
Εικόνα 2: Αυτόνομα, ευφυή οχήματα στον διαγωνισμό DARPA Grand Challenge [b13]....	31
Εικόνα 3 : Το μοντέλο Lego NXT με συνδεδεμένες μονάδες αισθητήρων και επενεργητών 32	
Εικόνα 4: Το NTX brick και οι εξωτερικές διεπαφές του [i17]	34
Εικόνα 5: Εσωτερική όψη μονάδων υλικού στο NXT brick [i26]	36
Εικόνα 6: Ο σερβοκινητήρας (servo motor) στο Lego NXT	38
Εικόνα 7: Ο αισθητήρας επαφής (touch sensor) στο Lego NXT	38
Εικόνα 8: Ο αισθητήρας υπερήχων (ultrasonic sensor) στο Lego NXT	39
Εικόνα 9: Ο αισθητήρας χρώματος (color sensor) στο Lego NXT	39
Εικόνα 10: Ο αισθητήρας ήχου (sound sensor) στο Lego NXT	40
Εικόνα 11: Παρακολούθηση περιβάλλοντος σε πραγματικό χρόνο μέσω browser	45
Εικόνα 12: Εικονικό περιβάλλον πόλης δημιουργημένο με το VSE	46
Εικόνα 13: Διάγραμμα δραστηριοτήτων σε VPL	52
Εικόνα 14 : Ανάπτυξη κώδικα μέσω του Visual Studio 2008 R2	59
Εικόνα 15: Το ανοιχτό λογισμικό αναγνώρισης φωνής Simon	61
Εικόνα 16: Αναπαράσταση εσωτερικού χώρου	84



Εικόνα 17: Εισαγωγή του λεξικού της εφαρμογής στο Simon	86
Εικόνα 18: Το ενεργό λεξικό της εφαρμογής στο Simon	87
Εικόνα 19: Η γραμματική της εφαρμογής στο Simon	88
Εικόνα 20: Ηχογραφήσεις προτάσεων για την δημιουργία βάσης ήχου στο Simon	89
Εικόνα 21: Ενσωμάτωση ακουστικού και γλωσσικού μοντέλου στην εφαρμογή	90
Εικόνα 22: Έλεγχος ακουστικού μοντέλου από την εφαρμογή	91
Εικόνα 23: Ιεραρχική οργάνωση των αρχείων ανάπτυξης στο Visual Studio 2008	92
Εικόνα 24: Εμπρόσθια άποψη του τελικού Lego NXT μοντέλου	97
Εικόνα 25: Πλάγια άποψη του τελικού Lego NXT μοντέλου	98
Εικόνα 26: Αρχική οθόνη εφαρμογής για την επιλογή περιβάλλοντος	99
Εικόνα 27: Εκτέλεση φωνητικών εντολών σε εικονικό περιβάλλον	100

Παράρτημα Δ: Ευρετήριο Πινάκων

Πίνακας 1: Παράδειγμα τυποποίησης κατά IPA	17
Πίνακας 2: Δείγμα διαθέσιμων προϊόντων στον τομέα επεξεργασίας φωνής	22
Πίνακας 3: Προδιαγραφές υλικού NXT brick	35
Πίνακας 4: Προϊόντα ρομποτικού λογισμικού	41
Πίνακας 5: Ρόλοι εργασιών στην ανάπτυξη και κατασκευή του συστήματος	54
Πίνακας 6: Ρόλοι και φάσεις εφαρμογής	55
Πίνακας 7: Περιγραφή απαιτήσεων	63
Πίνακας 8: Μη λειτουργικές απαιτήσεις περιβάλλοντος λογισμικού	64
Πίνακας 9: Μη λειτουργικές απαιτήσεις ανάπτυξης	65
Πίνακας 10: Συστάσεις υλικού	66
Πίνακας 11: Το λεξικό προφοράς της εφαρμογής	77
Πίνακας 12: Λογικοί κανόνες αντιστοίχισης	78
Πίνακας 13: Στοιχεία εκτελέσιμου εκκίνησης εφαρμογής	93
Πίνακας 14: Στοιχεία κύριων εκτελέσιμων αρχείων εφαρμογής	94
Πίνακας 15: Στοιχεία εκτελέσιμων αρχείων μηχανής Julius	94
Πίνακας 16: Στοιχεία εκτελέσιμου διεπαφής με τη μηχανή Julius	94
Πίνακας 17: Στοιχεία αρχείων παραμετροποίησης Julius	95
Πίνακας 18: Στοιχεία εκτελέσιμου ελέγχου του ακουστικού μοντέλου	95
Πίνακας 19: Ποσοστά επιτυχίας πρώτου ομιλητή σε περιβάλλον ήσυχου δωματίου	101
Πίνακας 20: Ποσοστά επιτυχία δεύτερου ομιλητή σε περιβάλλον ήσυχου δωματίου	102
Πίνακας 21: Ποσοστά επιτυχίας πρώτου ομιλητή σε τυπικό περιβάλλον δωματίου	102
Πίνακας 22: Ποσοστά επιτυχίας δεύτερου ομιλητή σε τυπικό περιβάλλον δωματίου	103



Πίνακας 23: Ποσοστά επιτυχίας στα ελληνικά με μη-προδιεγραμμένες λέξεις _____ 104

Πίνακας 24: Ποσοστά επιτυχίας στα αγγλικά με μη-προδιεγραμμένες λέξεις _____ 104

