

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και πολυμέσων



**ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΘΕΜΑ: “Οπτικοακουστική αναπαράσταση σελίδων
απο το internet”**

**Σπουδαστής:
Καμαράτος Εμμανουήλ**

**Εισηγητής:
Μαλάμος Αθανάσιος**

**Ηράκλειο
Φεβρουάριος 2008**

“ Αφιερώνω αυτή την εργασία στην οικογένεια μου που ήταν πάντα στο πλάι μου και με στήριξε σε όλη τη διάρκεια των σπουδών μου. ”

Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν να καταφέρω να ολοκληρώσω την κατασκευή αυτής της εφαρμογής.

Ιδιαίτερες ευχαριστίες θα ήθελα να δώσω στον κύριο Μαλάμο Αθανάσιο ο οποίος ήταν ο υπεύθυνος του project και με βοήθησε σε κάθε φάση της πορείας για τελειώσω αυτή της εργασία.

Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου που μου στάθηκαν και με βοήθησαν ψυχικά, ηθικά και υλικά σε όλη αυτή την προετοιμασία και τη διαδικασία.

Περιεχόμενα

1.ΠΡΟΛΟΓΟΣ	7
2.ΕΙΣΑΓΩΓΗ.....	9
2.1 Ποιος είναι ο σκοπός της εργασία μου.....	9
2.2 Τα εργαλεία που χρησιμοποίησα για να φτιάξω.....	9
2.3 Οι προϋποθέσεις της λειτουργίας του προγράμματος ..	11
3.ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ.....	13
3.1 Αριθμηση και ονομασία πλήκτρων.....	14
3.2 Speak.....	15
3.3 Surf.....	16
3.4 Stop.....	17
3.5 speech recognition.....	17
3.6 Όταν λέμε εκπαιδεύσουμε το πρόγραμμα	20
3.7 Exit.....	21
4.ΕΓΧΕΙΡΙΔΙΟ ΓΟΑ ΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΤΗ.....	22
4.1 Surf.....	23
4.2.....	25
4.3 EXIT.....	26
4.5 Speak.....	30
5.ΘΟΡΥΒΟΣ.....	33
6. ΚΩΔΙΚΑΣ.....	34
7.ΚΔΙΚΑΣ.....	42
8.ΦΙΛΤΡΑΡΙΣΜΑ.....	44
MICROSOFT SPEECH SDK5.1.....	49
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	50

1. ΠΡΟΛΟΓΟΣ

Σε αυτή εδώ την εργασία έχουμε φτιάξει μια web Based (διαδικτυακή) εφαρμογή που σαν κύριο στόχο έχει να στηρίζει τ' άτομα με ειδικές ανάγκες.

Αυτό επιτυγχάνεται με τη βοήθεια της εφαρμογής μας και κάποιων add-on που της έχουμε προσθέσει και για τα οποία θα μιλήσουμε αργότερα.

Η εφαρμογή μας χρειάστηκε αρκετές ώρες ενασχόλησης από τη μεριά μας και από όσους ήταν υπεύθυνοι για το όλο project.

Αυτό που έχουμε φτιάξει είναι ένας webBrowser που εκτός από το να δείχνει τις σελίδες του internet έχει άλλη δυο επιπλέον δυνατότητες: να “πιάνει” τα κείμενα που περιέχονται σε μια σελίδα , να τ' απαγγέλει και να τα εμφανίζει στην οθόνη.

Τα κείμενα εμφανίζονται στην οθόνη μας με δυο τρόπους.

α) Ο πρώτος τρόπος είναι να γίνεται με αναπαράσταση τους στη νοηματική γλώσσα.

β) Ο δεύτερος είναι μέσα σ' ένα rich Text Box να αναπαρίστανται τα κείμενα, τα οποία έχουν υποστεί φιλτράρισμα και τους έχουμε αφαιρέσει όλα τα html tags που περιέχονται σε μια σελίδα. Μερικά από αυτά μπορεί να είναι tags δηλαδή /t /b [b] [/b],html κώδικας,java scripts και διάφορα άλλα.

Πέρα όμως από αυτές τις δυο έχουμε προσθέσει μια τρίτη που ίσως είναι και η πιο σημαντική δυνατότητα.

Αυτή η τρίτη δυνατότητα που έχουμε προσθέσει στην εφαρμογής μας είναι ότι **μπορεί να δεχθεί φωνητικές εντολές από τον χρήστη.**

Δηλαδή έχουμε φτιάξει ένα κομμάτι του προγράμματος που μετά από κάποια εκπαίδευση μπορεί να καταλάβει τι του ζητάει ο χρήστης να κάνει και να το πραγματοποιήσει με απόλυτη επιτυχία.

Να τονίσουμε ότι είναι η πιο σημαντική ιδιότητα από όλες τις προηγούμενες και είναι αυτή που μας παίδεψε περισσότερο από όλες τις άλλες.

2. ΕΙΣΑΓΩΓΗ.

2.1. Ποιος είναι ο σκοπός της πτυχιακής εργασίας μου.

Ο σκοπός της εργασίας είναι να φτιάξουμε ένα φωνητικό browser για το διαδίκτυο. Μ' αυτή μας την εφαρμογή ανοίγουμε το δρόμο σε άτομα που έχουν σοβαρά προβλήματα υγείας όπως τυφλοί, ανάπηροι, κουφοί και λοιπά έτσι ώστε να μην στερούνται την δυνατότητα να μπορούν να κάνουν και αυτοί μια περιήγηση στο internet καθώς επίσης να έχουν και όλα τα ωφέλεια που προσφέρει το internet, δηλαδή να μπορούν να εξυπηρετούνται σε διάφορα θέματα όπως υγεία ,εργασία ,δημοσίων υπηρεσιών και τα λοιπά.

Με απώτερο σκοπό να μπορούν να εξυπηρετούνται αποκλειστικά και μόνο μέσω του διαδικτύου.

2.2. Τα εργαλεία που χρησιμοποίησα για να κατασκευάσω το πρόγραμμα.

Η εφαρμογή μας είναι γραμμένη εξολοκλήρου σε visual studio .net 2005 και της έχουμε προσθέσει κάποια επιπλέον χαρακτηριστικά ώστε να καταφέρουμε να επιτύχουμε το συγκεκριμένο αποτέλεσμα.

Κάποια από αυτά είναι ο **web Browser** και το **Microsoft Speech** που είναι στην έκδοση 5.1 .Το speech είναι πάρα πολύ βασικό add-on γιατί μας βοηθάει στο να καταφέρουμε να δημιουργήσουμε το κομμάτι της εφαρμογής που έχει ως κύριο στόχο την αναγνώριση της φωνής ή της ομιλίας.

Να μην ξεχάσω ν' αναφέρω ότι το συγκεκριμένο plug – in μας βοηθάει και στο να καταφέρουμε να κάνουμε την εφαρμογή ν' απαγγείλει τις λέξεις που είναι αποθηκευμένες στο buffer (προσωρινή αποθήκευση μνήμης).

Το buffer είναι προσωρινή αποθήκευση των δεδομένων μας μέσα σ' ένα αρχείο που βρίσκετε μέσα στο δίσκο του υπολογιστή μας.

Ο web Browser είναι ένα component το οποίο μας βοηθάει στο ν' απεικονίζουμε τις σελίδες από το internet που έχει ζητήσει ο χρήστης από την εφαρμογή μας. Δηλαδή να τις φέρουμε στον υπολογιστή του, να τις φορτώσουμε στη μνήμη και σαν τελική κίνηση να τις εμφανίσουμε στην οθόνη του.

Οι σελίδες αυτές είναι στο internet και εμφανίζονται μέσα απο την εφαρμογή μας.

Το component αυτό περιέχεται μέσα στην κεντρική φόρμα της εφαρμογής μας και είναι το πάνω παράθυρο μέσα απο το οποίο μπορούμε να βλέπουμε τις σελίδες απο το internet.

Το speech μας βοηθάει στο να καταφέρουμε να κάνουμε το πρόγραμμα να δέχεται φωνητικές εντολές απο το χρήστη. Επίσης μας βοηθάει να μπορούμε να δώσουμε στο πρόγραμμα τη δυνατότητα ν' απαγγέλει τα γράμματα που διαβάζει απο το buffer αφού πρώτα τα έχει αποθηκεύσει εκεί που του έχουμε ορίσει και φυσικά τα έχει φιλτράρει ώστε να φύγουν ή ν' αφαιρεθούν όλα τα περιττά html tags που περιέχονται σε μια σελίδα.

Τέλος έχουμε βάλει μια μπάρα που απεικονίζει την λέξη που περνάει εκείνη τη στιγμή από την οθόνη μας.

Αυτό μας βοηθάει στο να ελέγχουμε καλύτερα τα αποτελέσματα που μας δίνει η εφαρμογή μας. Επίσης βοηθάει τ' άτομα με προβλήματα στην ακοή να είναι σε θέση να διαβάζουν κανονικά ολόκληρες τις λέξεις μέσα από τα κείμενα της σελίδας και γιατί όχι αργότερα και ολόκληρα τα κείμενα.

Έτσι μπορούν να διαβάσουν ολόκληρη τη σελίδα να βγάλουν το νόημα απο τα κείμενα της και με αυτόν τον τρόπο μπορούν να καταλάβουν τι τους ζητάμε να κάνουν ώστε να είναι σε θέση να πράξουν τ' αντίστοιχα.

2.3 Οι προϋποθέσεις της λειτουργίας του προγράμματος.

ΣΗΜΑΝΤΙΚΟ:

Για να δουλέψει σωστά η εφαρμογή μας πρέπει να έχουμε εγκαταστήσει το Microsoft Framework 2. Το framework είναι ένα layer των windows που μας βοηθάει να φτιάξουμε κάποιες εφαρμογές πιο εύκολα και πιο γρήγορα σε σχέση με μια συνηθισμένη μέθοδο ανάπτυξης.

Αυτό γίνεται γιατί το framework αναλαμβάνει να εμφανίσει και να χειριστεί την εφαρμογή όπως ακριβώς την έχουμε σχεδιάσει ανεξάρτητος της έκδοσης των windows που χρησιμοποιούμε.

Αυτό σημαίνει ότι εφόσον έχουμε το framework στο υπολογιστή μας δεν έχει καμία σημασία αν τρέχουμε την έκδοση των windows Xp είτε των windows Vista άσχετα με την όποια από τις εκδόσεις τους, είτε έχουμε τα windows 98. Η εφαρμογή μας θα εμφανίζεται και θα δουλεύει ακριβώς με τον ίδιο τρόπο.

Βέβαια να μην ξεχάσουμε ότι το framework διαθέτει και κάποιες έτοιμες συναρτήσεις οι οποίες θα μας βοηθήσουν παρά πολύ. Αυτό το λέμε γιατί αυτές έχουν τη δυνατότητα να κάνουν κάποια πράγματα από μόνες τους.

Ένα από αυτά είναι ότι υπάρχει ενσωματωμένος ο garbage collector του οποίου η δουλειά είναι ν' αναλάβει να καθαρίσει τη μνήμη του υπολογιστή μας δηλαδή τη RAM από όποια δεδομένα δεν χρησιμοποιούνται ή που δεν χρειάζονται από τα προγράμματα που είναι φορτωμένα εκείνη τη χρονική περίοδο από το χρήστη.

Επίσης ένα ακόμα add-on που πρέπει να εγκαταστήσουμε είναι το Microsoft speech SDK 5.1. Αυτό το add on μας βοηθάει στο να φτιάξουμε το κομμάτι της εφαρμογής που αναγνωρίζει την ανθρώπινη ομιλία και την μετατρέπει σ' εντολές για τον υπολογιστή.

Αυτές οι εντολές μπορεί να έχουν τη μορφή κειμένου δηλαδή ο υπολογιστής να μεταφράζει την φωνή μας σε γραπτό κείμενο το οποίο θα το εμφανίζει στην οθόνη, να μεταφράζει τις εντολές μας και να εκτελεί κάποια συγκεκριμένη λειτουργία και διάφορα άλλα.

Στο σημείο αυτό να τονίσουμε ότι η c και η c# δεν έχουν garbage collector μιας και αυτό είναι χαρακτηριστικό της java. Βέβαια επειδή τα framework τα έχει κατασκευάσει η Microsoft κάθισα οι προγραμματιστές της και έφτιαξαν από μόνοι τους ένα δικό τους garbage collector.

Δεν του έδωσαν αυτό το όνομα απλά κάνει ακριβώς την ίδια δουλειά με τον αντίστοιχο της JAVA οπότε δανείζομαι την ονοματολογία από εκεί. Ο garbage collector

της Microsoft με τη σειρά του αντιγράφει την αντίστοιχη λειτουργία από το κομμάτι του προγράμματος της java.

Δηλαδή και αυτός κοιτάει ποια πακέτα δεν χρειάζονται και φροντίζει ώστε να τα κλείσει ή να τα απομακρύνει για να καταφέρει να απελευθερώσει όσο περισσότερη μνήμη ram γίνεται.

Το .net framework 2 μπορούμε να το βρούμε στην εξής διεύθυνση:

<http://www.microsoft.com/downloads/details.aspx?familyid=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en>

3. ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ.

Σ' αυτό εδώ το εγχειρίδιο θα δώσουμε όλες τις απαραίτητες οδηγίες, διευκρινήσεις και κατευθύνσεις που χρειάζονται έτσι ώστε να μπορέσει κάποιος να χρησιμοποιήσει την εφαρμογή μας.

Έχουμε φτιάξει μια web-based (διαδικτυακή)εφαρμογή με σκοπό την αναπαράσταση σελίδων από το internet.

Επίσης της έχουμε προσθέσει και κάποιες extra δυνατότητες ώστε να βοηθήσουμε τ' άτομα με ειδικές ανάγκες.

Κάποιες από τις επιπλέον λειτουργίες που έχουμε προσθέσει είναι :

α) Η φωνητική αναπαράσταση των σελίδων.

Αυτό είναι πολύ χρήσιμο σε άτομα με προβλήματα στην όραση διότι έχουν τώρα τη δυνατότητα να κάνουν μια περιήγηση στο διαδίκτυο με τη βοήθεια της εφαρμογής μας ή κάποιας παρόμοιας.

β) Η απεικόνιση των κειμένων που περιέχονται σε μια σελίδα. Να μην ξεχνάμε ότι αυτό γίνεται με βάση τη νοηματική γλώσσα με την βοήθεια του δακτυλικού αλφαβήτου. Έτσι πετυχαίνουμε να δώσουμε τη δυνατότητα στα άτομα με πρόβλημα στην ακοή να μπορούν να πραγματοποιήσουν κάποιες εργασίες μέσω του διαδικτύου.

γ) Η αναγνώριση της φωνής.

Είναι η πιο βασική λειτουργία του προγράμματος. Αφού έχουμε εκπαιδεύσει το πρόγραμμα στον τόνο της φωνής μας μπορούμε χρησιμοποιώντας ένα μικρόφωνο να δίνουμε εντολές στην εφαρμογή μας και αυτή να τις εκτελεί.

Στο σημείο αυτό να τονίσω ότι πριν ξεκινήσει η διαδικασία της αναγνώρισης φωνής πρέπει πρώτα να κάνουμε ένα μικρό training δηλαδή μια εκπαίδευση με τη βοήθεια του μικροφώνου ώστε το πρόγραμμα να καταφέρει να προσαρμοστεί στη φωνή μας και να έχει την κατά δύναμη μικρότερη απόκλιση από τις εντολές μας.

3.1 Αρίθμηση και ονομασία των πλήκτρων μας.

Παράλληλα με την ονομασία θα δίνουμε και κάποιες διευκρινήσεις για τον τρόπο λειτουργίας τους.

Μ' αυτή εδώ τη μέθοδο δίνουμε στο πρόγραμμα τη δυνατότητα να χρησιμοποιηθεί από όλους.

Η εφαρμογή μας έχει πέντε κουμπιά:

α) Το πρώτο κουμπί είναι το **speak**, μόλις το πατήσουμε λέμε στην εφαρμογή μας να πάει να πάρει τη διεύθυνση που του έχουμε γράψει στο text box μας.

Να πάει να πάρει τη σελίδα να τη φέρει στον υπολογιστή μας και να την εμφανίσει στην οθόνη μας.

Μετά σαν δεύτερη εντολή να πάρει τα κείμενα της σελίδας ,να τα φιλτράρει και να μας τα εμφανίσει μέσα στο text box που του έχουμε ορίσει.

Να μην ξεχάσουμε οτι όταν πατήσουμε το κουμπί speak ανοίγει ένα νέο παράθυρο pop-up μέσα στο οποίο αναπαρίστανται ολόκληρες οι λέξεις. Μέσα σε αυτό το παράθυρο έχουμε τοποθετήσει ένα ακόμα κουμπί ή καλύτερα ένα **track Bar** το οποίο είναι κάτι παρόμοιο σαν το **scroll bar** και με το οποίο μπορούμε να ρυθμίζουμε την ταχύτητα που θα εμφανίζονται οι λέξεις στην οθόνη μας.

Δηλαδή πόσο χρόνο σε δευτερόλεπτα πάντα θα περιμένει το πρόγραμμα μέχρι να απαγγείλει την επόμενη λέξη.

β) Το δεύτερο κουμπί μας είναι το **surf** που πολύ απλά πάει παίρνει τη σελίδα που του έχουμε γράψει στο text box μας και μας την εμφανίζει στην οθόνη μας. Προσοχή αυτή η εντολή δεν φέρνει τα κείμενα!

γ) Το τρίτο κουμπί μας είναι η αναγνώριση φωνής ή αλλιώς **speech recognition**. Μόλις το πατήσουμε λέμε στο πρόγραμμα να πάει και να καλέσει τη συνάρτηση που πραγματοποιεί αυτή τη συγκεκριμένη εργασία στην εφαρμογή μας.

δ) Το τέταρτο κουμπί είναι το **stop** που μας δίνει την δυνατότητα να σταματήσουμε όλες τις λειτουργίες που εκτελεί η εφαρμογή μας εκείνη τη χρονική στιγμή.

ε) Το τελευταίο κουμπί είναι το **exit** το οποίο μας βγάζει απο την εφαρμογή.

3.2 Speak:

Όταν το πατάμε δίνουμε στην εφαρμογή μας την εντολή να πάρει τη διεύθυνση που της έχουμε γράψει στο textbox μας. Να πάει να φέρει τη σελίδα ,να φέρει τα κείμενα της ,να τα καθαρίσει απο όλα τα περιττά στοιχεία όπως html, htm ,java script και άλλα Html tags.

Αφού γίνουν όλα αυτά ν' ανοίξει ένα νέο παράθυρο δηλαδή ένα pop-up στο οποίο βλέπουμε ν' αναπαρίστανται τα κείμενα της σελίδας λέξη προς λέξη .Επίσης μέσα στο ίδιο παράθυρο βλέπουμε τα κείμενα της σελίδας, δηλαδή τις λέξεις ν' αναπαρίστανται στη νοηματική γλώσσα. Όταν οι λέξεις αναπαρίστανται, ταυτόχρονα απαγγέλλονται για να τις ακούει ο χρήστης.

Μέσα στο pop-up παράθυρο που ανοίγει όταν πατάμε το κουμπί speak είναι τοποθετημένη μια μπάρα. Η δουλειά της είναι να καθορίζει την ταχύτητα με την οποία θα προφέρονται οι λέξεις του κειμένου.

Η μονάδα μέτρησης που χρησιμοποιούμε είναι σε δευτερόλεπτα.

Η διαβάθμιση της είναι από το 2 μέχρι και το 10.

Ένα παράδειγμα είναι:

αν επιλέξουμε το 2 που είναι και η μικρότερη τιμή που μπορούμε να δώσουμε, όταν έχει διαβάσει τη μια λέξη ο χρόνος που πρέπει να περάσει μέχρι να διαβάσει την επόμενη λέξη είναι δυο δευτερόλεπτα.

Κάτι αντίστοιχο γίνετε για όποια τιμή και αν έχουμε επιλέξει μεταξύ του δυο (2) και του δέκα (10) που είναι οι δυο ακραίες τιμές μας. Να τονίσουμε ότι το **track Bar** παίρνει μόνο ακέραιες τιμές μεταξύ των επιτρεπόμενων τιμών.

Πιστεύω οτι τώρα είναι το κατάλληλο σημείο για να ξεκαθαρίσουμε τι ακριβώς είναι τα δυο frames που χρησιμοποιούμε στην εφαρμογή μας.

Στην πραγματικότητα το πάνω frame είναι ένα component που λέγεται web Browser και μας είναι απαραίτητο για να μπορέσουμε να απεικονίσουμε τη σελίδα που ζητάμε από το διαδίκτυο.

Το κάτω frame είναι ένα άλλο component που λέγεται rich text box και μας χρησιμεύει στο να εμφανίσει όλα τα κείμενα που περιέχονται σε αυτή τη σελίδα.

Επίσης να σημειώσουμε ότι όλοι οι μη χρήσιμοι χαρακτήρες φιλτράρονται και δεν εμφανίζονται.

Στην εφαρμογή μας έχουμε προσθέσει μια επιπλέον ικανότητα η οποία μας βοηθάει να κάνουμε την εφαρμογή να είναι σε θέση ν' απεικονίζει και ελληνικούς και λατινικούς χαρακτήρες.

Τα κείμενα της σελίδας αποθηκεύονται σ' ένα buffer (προσωρινή θέση μνήμης) και εφόσον εμείς έχουμε πατήσει το πλήκτρο speak ανοίγει ένα pop up παράθυρο, όπου οι χαρακτήρες που είναι αποθηκευμένοι στο buffer μας, μετατρέπονται σε λέξεις και μετά από αυτή τη μετατροπή αρχίζουν ν' αναπαρίστανται λέξη προς λέξη.

Η αναπαράσταση γίνεται στη νοηματική γλώσσα με τη βοήθεια του δακτυλικού αλφαβήτου. Επίσης σ' αυτό το παράθυρο έχουμε προσθέσει μια μπάρα όπου αναπαρίστανται ολόκληρες οι λέξεις που πρέπει να εμφανιστούν στην οθόνη μας.

Αυτό γίνεται για να μπορεί κάποιος με πρόβλημα στην ακοή να διαβάζει ολόκληρες λέξεις και να είναι σε θέση να καταλαβαίνει το νόημα της σελίδας. Αφού πια έχει διαβάσει το κείμενο μπορεί να πράξει τ' αντίστοιχα.

Βέβαια αυτή η προσθήκη έχει διπλό νόημα, γιατί βοηθάει και μας στο να ελέγχουμε την ορθότητα των αποτελεσμάτων που έρχονται σαν output από την εφαρμογή μας.

3.3 Surf:

Όταν πατάμε το κουμπί Surf δίνουμε στο πρόγραμμα μας την εντολή να πάει να πάρει τη διεύθυνση που του έχουμε δώσει. Αυτή την διαβάζει από το textbox που του έχουμε ορίσει.

Μετά πάει στη σελίδα, τη φέρνει στον υπολογιστή μας και στο τέλος μας την εμφανίζει μέσα από το component που λέγεται web Browser.

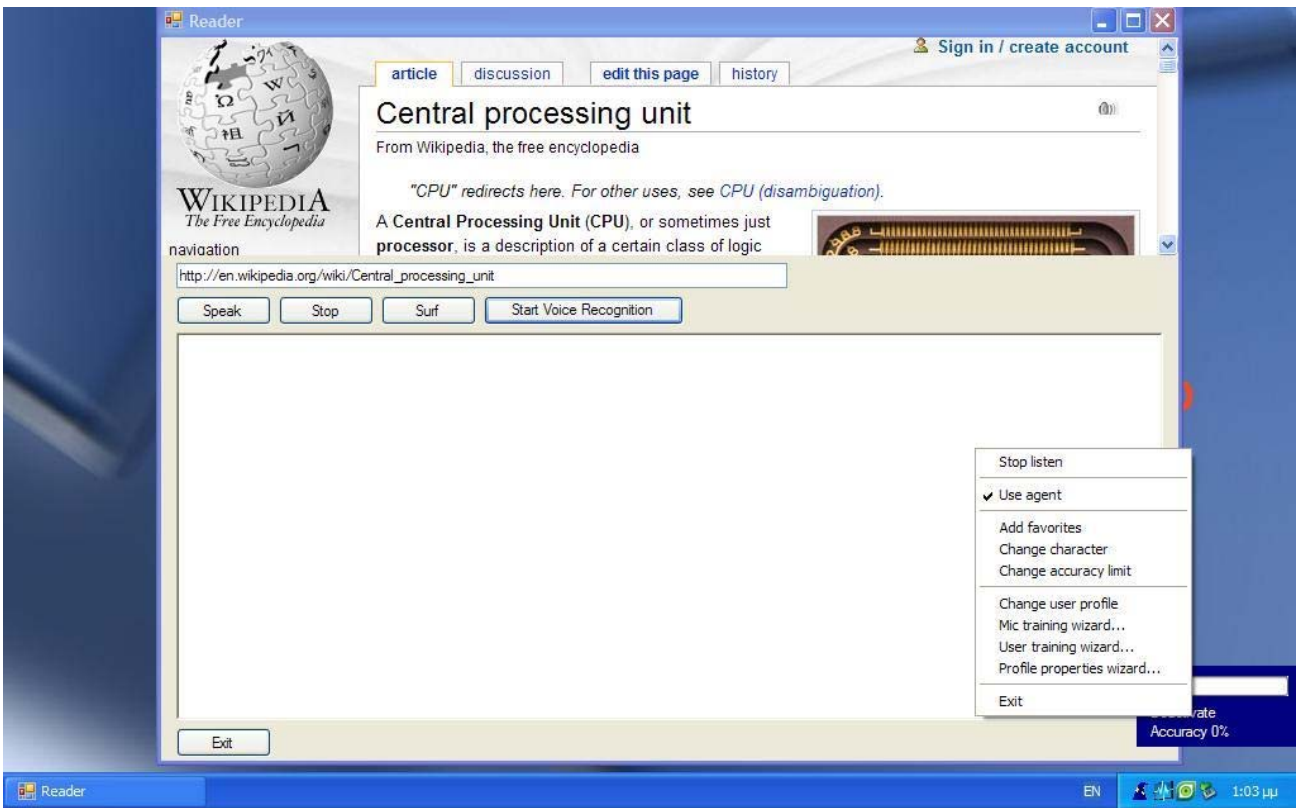
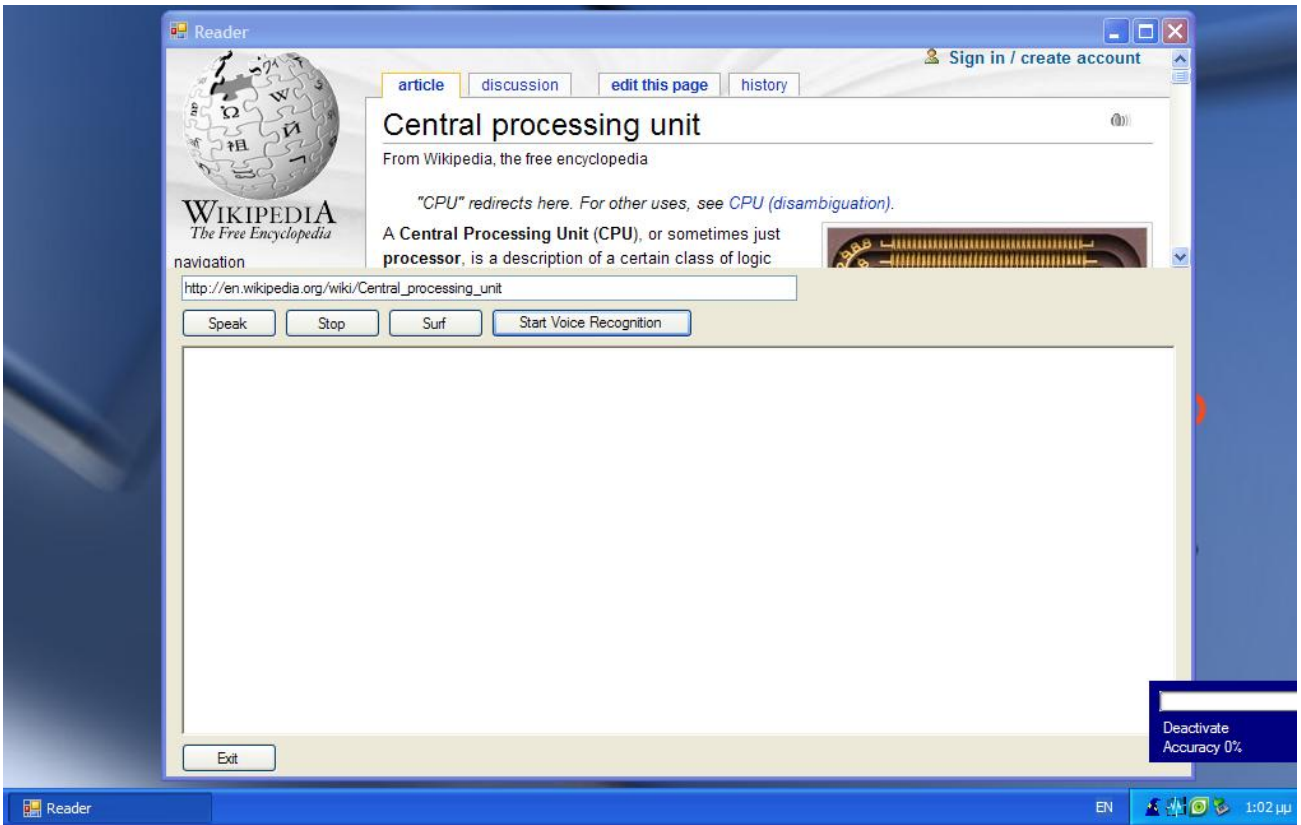
3.4 Stop:

Όταν πατάμε το κουμπί stop η εφαρμογή μας σταματάει όλες τις λειτουργίες της. Το χρησιμοποιούμε όταν θέλουμε να δούμε κάποια άλλη σελίδα ή αν χρειάζεται να κάνουμε κάποια άλλη δουλειά. Για να είμαστε σε θέση να επανέλθουμε αργότερα ώστε να μπορούμε να συνεχίσουμε απο εκεί που είχαμε μείνει.

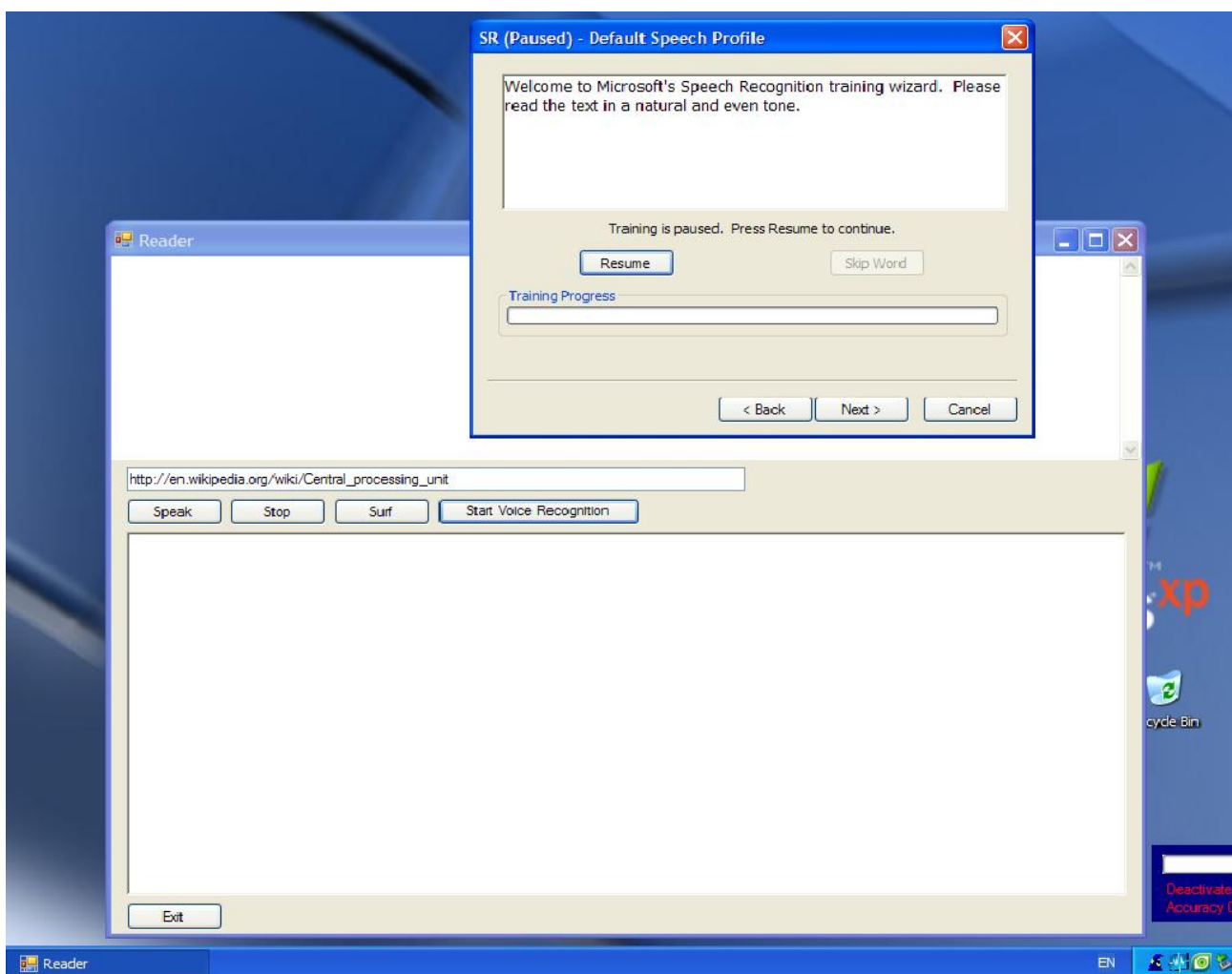
3.5 Speech recognition:

Το Speech recognition είναι βέβαια το πιο σημαντικό κομμάτι όλης της εφαρμογής. Με τη βοήθεια του το πρόγραμμα μπορεί μέσω ενός μικροφώνου να δεχθεί φωνητικές εντολές απο το χρήστη και να τις πραγματοποιήσει.

Μόλις πατήσουμε το κουμπί speech recognition θα δούμε ότι στη οθόνη ότι θα εμφανιστεί ένα καινούργιο εικονίδιο.



Όταν κάνουμε δεξί click πάνω του θα δούμε μια σειρά απο επιλογές. Για να πραγματοποιηθεί η αναγνώριση της ομιλίας μας πρέπει να πατήσουμε το κουμπί mic training wizard. Αμέσως θα εμφανιστεί ένα κουτάκι που θα περιέχει κάποιες προτεινόμενες λέξεις τις οποίες εμείς καλούμαστε να διαβάσουμε έτσι ώστε να εκπαιδύσουμε το πρόγραμμα στον τόνο και στον τρόπο ομιλίας μας. Όταν γίνει αυτό η εφαρμογή θα είναι σε θέση ν' αναγνωρίζει τη φωνή μας και να καταλαβαίνει τι της ζητάμε να κάνει.



3.6 Όταν λέμε “εκπαιδεύσουμε” το πρόγραμμα τι εννοούμε?

Η εφαρμογή μας ζητάει να απαγγείλουμε κάποιες προτεινόμενες λέξεις οι οποίες εμφανίζονται στην οθόνη μας μέσα σ' ένα κουτάκι. Όσο εμείς κάνουμε αυτή τη διαδικασία το πρόγραμμα καταγράφει τον τόνο της φωνής μας ,τη προφορά μας και τον τρόπο ομιλίας μας γενικότερα.

Όπως μπορούμε να δούμε στη από πάνω φωτογραφία έχουμε τις λέξεις μέσα στο άσπρο πλαίσιο. Κάτω από αυτό έχουμε δυο κουμπιά, το resume και το skip word. Αν θελήσουμε να σταματήσουμε την εκπαίδευση του προγράμματος για κάποιο λόγο και να συνεχίσουμε λίγο αργότερα πατάμε το κουμπί pause,μόλις το κάνουμε αλλάζει η ετικέτα (label) του κουμπιού σε resume. Όταν τελειώσουμε ότι κάνουμε πατάμε πάλι το κουμπί resume και συνεχίζουμε από εκεί που είχαμε μείνει.

Το δεύτερο κουμπί που έχει αυτή η φόρμα είναι το skip word.

Αν κατά τη διάρκεια της εκπαίδευση το πρόγραμμα δεν μπορεί να βρει μια λέξη τότε μπορούμε πατώντας το να προχωρήσουμε στην επόμενη.

Με βάση τα δείγματα που έχει απο τη εκπαίδευση που έχει υποστεί ξεκινάει την ανάλυση του σήματος. Με τη βοήθεια των μαθηματικών μοντέλων στατιστικής και πιθανοτήτων αναλύει το σήμα που παίρνει για είσοδο .Να μην ξεχνάμε οτι για τον υπολογιστή η φωνή μας είναι ένα ηχητικό σήμα. Όταν πια το πρόγραμμα έχει αρκετά μεγάλο δείγμα απο το σήμα της φωνής μας, είναι πια πλήρως ικανό να αναγνωρίσει τις φωνητικές εντολές μας και στην μετέπειτα διαδικασία.

Κάποιες ενδεικτικές εντολές είναι activate, deactivate κ.τ.λ.

Όλα αυτά γίνονται με τη βοήθεια των speech object's για τα οποία θα κάνουμε λόγο λίγο αργότερα στον τομέα programming manual.

Μια σύντομη περιγραφή όμως μπορούμε να τη δώσουμε.

Το Microsoft speech SDK 5.1 είναι το εργαλείο για να φτιάξουμε την αναγνώριση ομιλίας. Το κανονικό του όνομα είναι Speech Recognition object's και μαζί με το speech της Microsoft είναι τα μόνα εργαλεία που χρειαζόμαστε για να φτιάξουμε το κομμάτι που αναφέρεται στο speech recognition.

3.7 Exit:

Τέλος έχουμε το κουμπί exit νομίζω ότι δεν χρειάζεται πολλές συστάσεις.

Όταν το πατάμε λέμε στην εφαρμογή να σταματήσει ότι κάνει, ν' αδειάσει όλα τα buffer και να κλείσει όλα τα connection που έχει ανοίξει από το surfing, να σταματήσει το service που τρέχει από πίσω ώστε να είναι ικανά τα windows να την χρησιμοποιήσουν. Σαν δεύτερη κίνηση να κλείσει και το δεύτερο service που είναι για το speech SDK της Microsoft και σαν τελική εντολή και η ίδια η εφαρμογή να τερματιστεί.

Τώρα θα πούμε δυο λόγια για το πως δουλεύει η εφαρμογή μας.

Πρέπει να τονίσουμε ότι πέρα από το γραφικό περιβάλλον το οποίο είναι αυτό που βλέπουμε στην οθόνη μας, υπάρχει και ένα άλλο κομμάτι του προγράμματος που δεν έχει γραφικό περιβάλλον και δεν εμφανίζεται στην οθόνη μας.

Αυτό βεβαίως είναι και το πιο σημαντικό κομμάτι σε όλη την εφαρμογή μας.

Στο σημείο αυτό πρέπει να πούμε ότι το service που τρέχει από πίσω χρησιμοποιεί κάποιες βιβλιοθήκες που είναι κατάλληλες στο να μας βοηθήσουν να πραγματοποιήσουμε το κομμάτι της εφαρμογής μας που δέχεται φωνητικές εντολές.

Οι εντολές αυτές έρχονται αποκλειστικά και μόνο από το χρήστη φυσικά με τη βοήθεια ενός μικροφώνου.

4. ΕΓΧΕΙΡΙΔΙΟ ΓΙΑ ΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΤΗ.

Η εφαρμογή μας είναι μια web base application. Για να την κάνουμε να δείχνει τις σελίδες απο το internet της έχουμε κάνει intergraded (ενσωματώσει) τη μηχανή του internet explorer με τη μορφή ενός component που λέγεται web Browser1. Επίσης έχουμε προσθέσει τα πέντε προαναφερθέντα κουμπιά, ένα frame που η χρησιμότητα του είναι ν' απεικονίζει τις σελίδες του internet και ένα δεύτερο frame που η δουλειά του είναι να απεικονίζει τα κείμενα απο τις σελίδες του internet. Έχουμε βάλει ένα buffer ώστε να μπορούμε να αποθηκεύουμε τα κείμενα μιας σελίδας και να τα διαβάζουμε από εκεί όταν χρειαστεί να τα εμφανίσουμε στην οθόνη μας. Τέλος έχουμε προσθέσει ένα textbox στο οποίο βάζουμε τη διεύθυνση της σελίδας που θέλουμε να δούμε.

Η εφαρμογή πάει και διαβάζει απο το textbox τη διεύθυνση που έχουμε δώσει με τη βοήθεια μιας συνάρτησης που έχουμε φτιάξει.

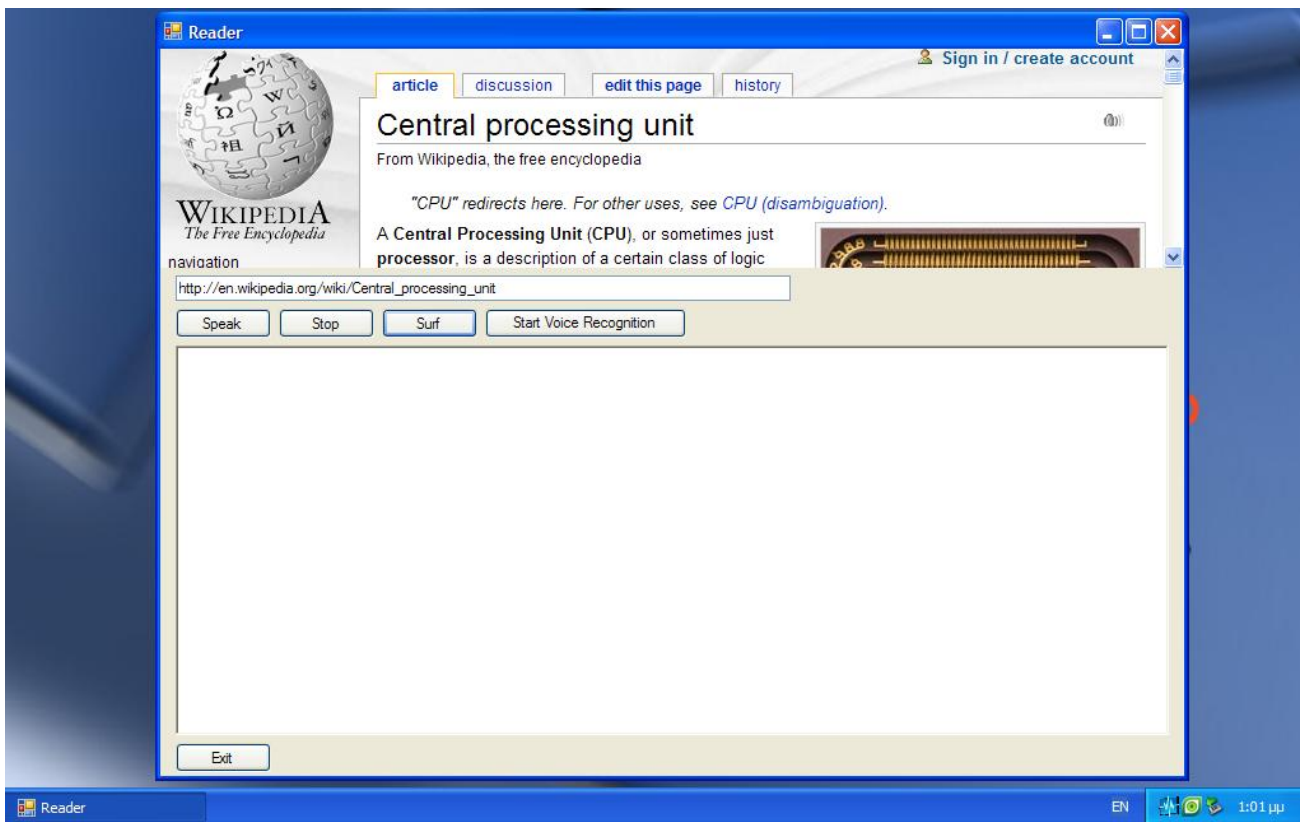
Με αυτό το τρόπο μας εμφανίζει τη σελίδα και τα κείμενα της.

Για να καταφέρει η εφαρμογή να απαγγέλει τα γράμματα που είναι αποθηκευμένα στο buffer και να είναι ικανή να καταλαβαίνει τις φωνητικές εντολές που δέχεται απο το χρήστη έχουμε κάνει ακόμα μια μικρή προσθήκη.

Αυτή δεν είναι άλλη από ένα ακόμα component που είναι το Microsoft speech το οποίο βρίσκετε στην έκδοση 5.1.

Η εφαρμογή μας είναι κατασκευασμένη με τέτοιο τρόπο ώστε κάθε κομμάτι της να είναι εντελώς αυτόνομο απο τα υπόλοιπα. Αυτό μας βοηθάει στη διόρθωση του προγράμματος γιατί δεν χρειάζεται να ελέγχουμε μεγάλα κομμάτια από τον κώδικα του.

Τώρα θα παραθέσω μια εικόνα από την κεντρική φόρμα της εφαρμογής.



4.1 Surf

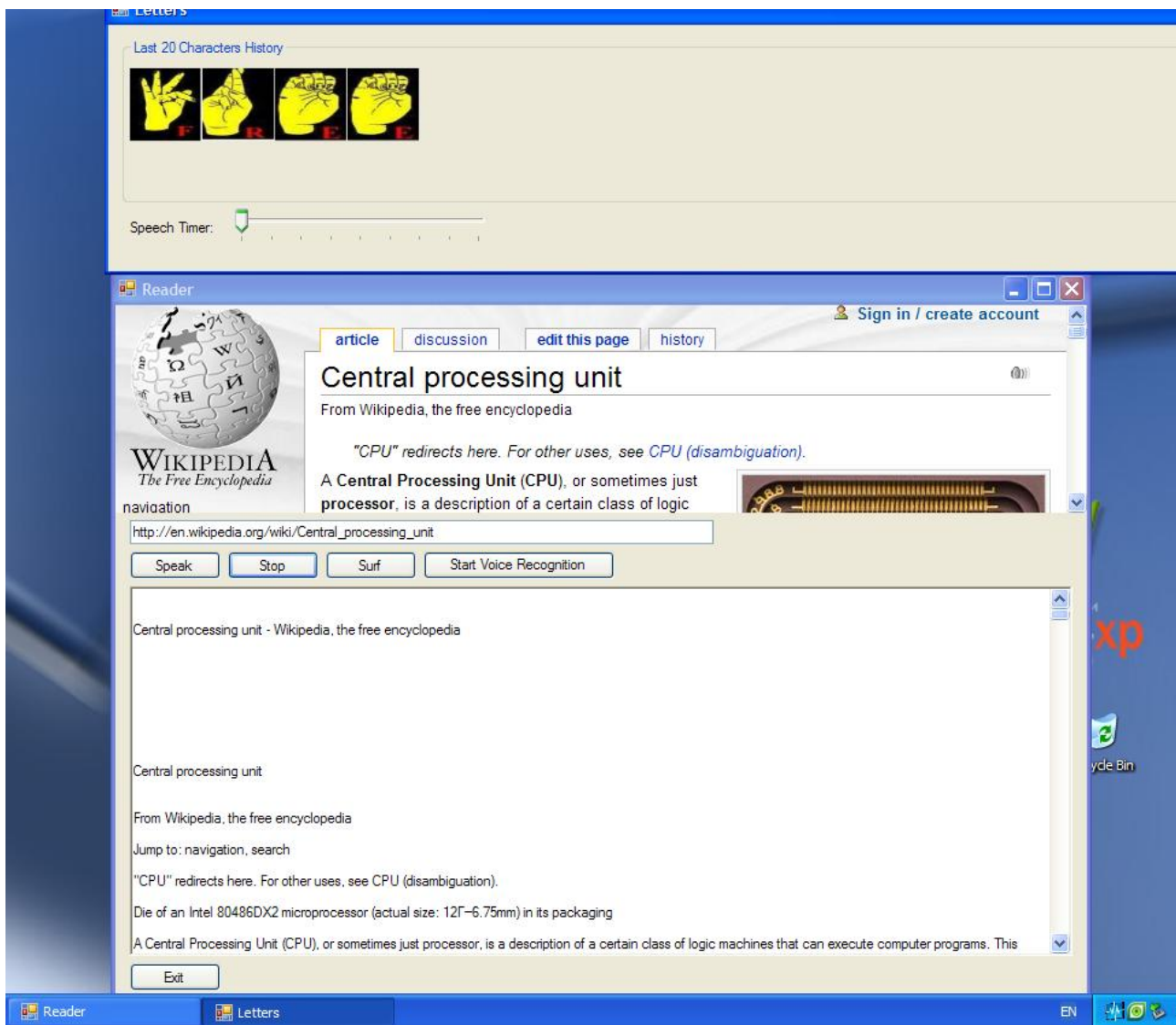
Όταν ο χρήστης πατήσει το κουμπί **surf** ενεργοποιεί το κομμάτι του κώδικα που είναι υπεύθυνο να πάει να φέρει τη σελίδα και να τη δείξει στο πάνω frame που στην πραγματικότητα είναι ο web Browser1.

Αυτό που γίνεται στην πράξη είναι ότι ενεργοποιείτε το component web Browse το οποίο είναι ένα object που έχει μια ιδιότητα που λέγεται Navigate δηλαδή πλοήγηση. Αυτό παίρνει τη διεύθυνση (url) από το textbox1 και μας φέρνει τη σελίδα στον υπολογιστή μας.

Εκείνη τη στιγμή ενεργοποιείτε το κομμάτι του κώδικα που λέει:

Κώδικας :

```
webBroser1.Navigate(textbox1.Text);
```



4.2 STOP

Όταν ο χρήστης πατήσει το κουμπί **stop** ενεργοποιεί το κομμάτι του κώδικα που είναι για να σταματάει όλες τις λειτουργίες του προγράμματος μας.

Δηλαδή να αδειάζει το buffer και να σταματήσει τον timer1 να κλείσει όλες τις συνδέσεις που έχει ανοίξει με το Internet και σαν τελική εντολή να σταματήσει να φέρνει δεδομένα από το διαδίκτυο και να κλείσει όλα τα connections που έχει ανοίξει με όλες τις διευθύνσεις που έχει συνδεθεί.

Κώδικας :

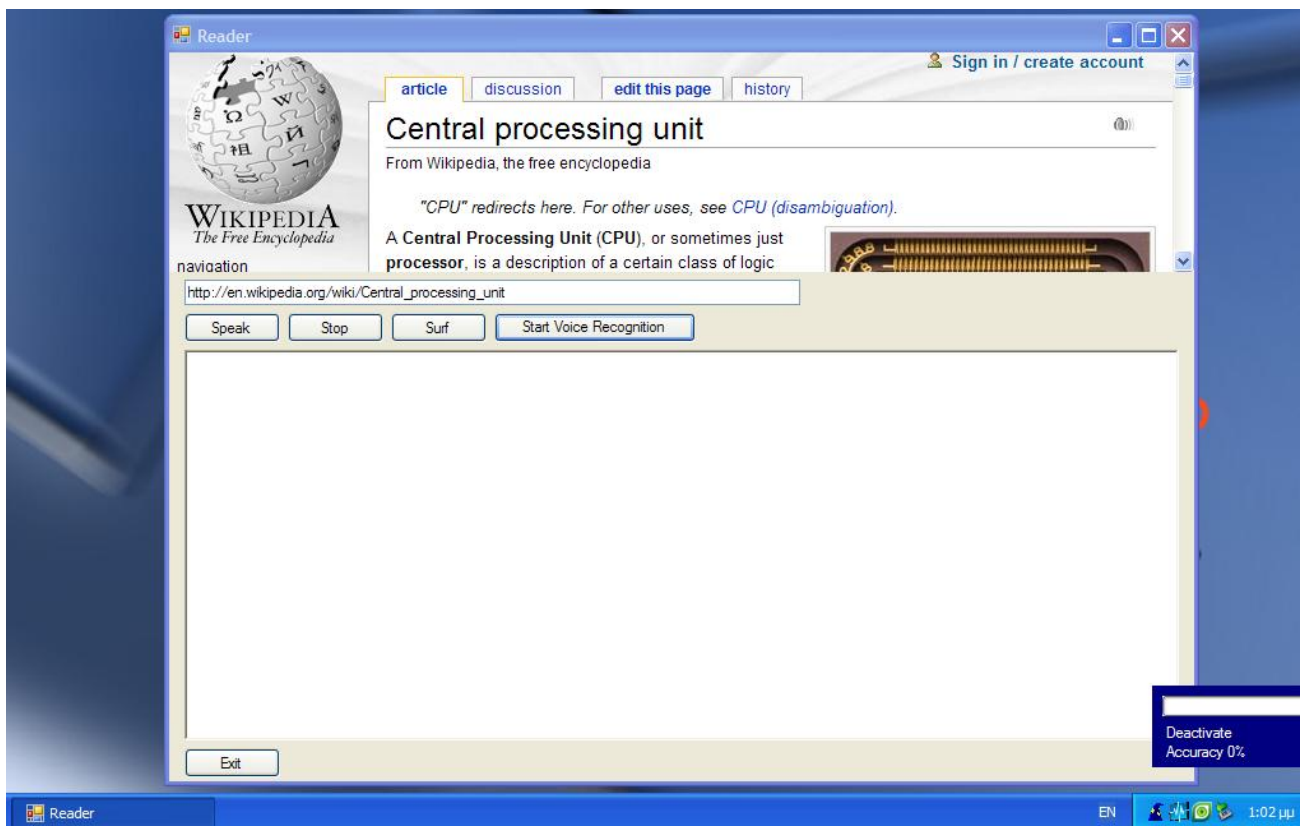
```
fReadBuffer.clear();  
timer1.Enable=false;
```

4.3 EXIT

Όταν ο χρήστης πατήσει το κουμπί **exit** τότε ενεργοποιεί το κομμάτι του κώδικα που είναι για να αδειάζουν όλα τα buffer, να σταματάει ο timer , να κλείσει όλες τις συνδέσεις που έχει ανοίξει με τις διάφορες σελίδες απο το διαδίκτυο, να σταματήσει το service που τρέχει ώστε η εφαρμογή ώστε να είναι προσπελάσιμη από τα windows , να σταματήσει το service που είναι για το speech sdk 5.1 μιας και το χρησιμοποιεί και αυτό η εφαρμογή και σαν τελευταία κίνηση πρέπει να κλείσει η εφαρμογή και να εξαφανίζεται η φόρμα.

Κώδικας:

```
fReadBuffer.clear();  
timer1.Enable=false;  
this.Close();  
this.Dispose();
```

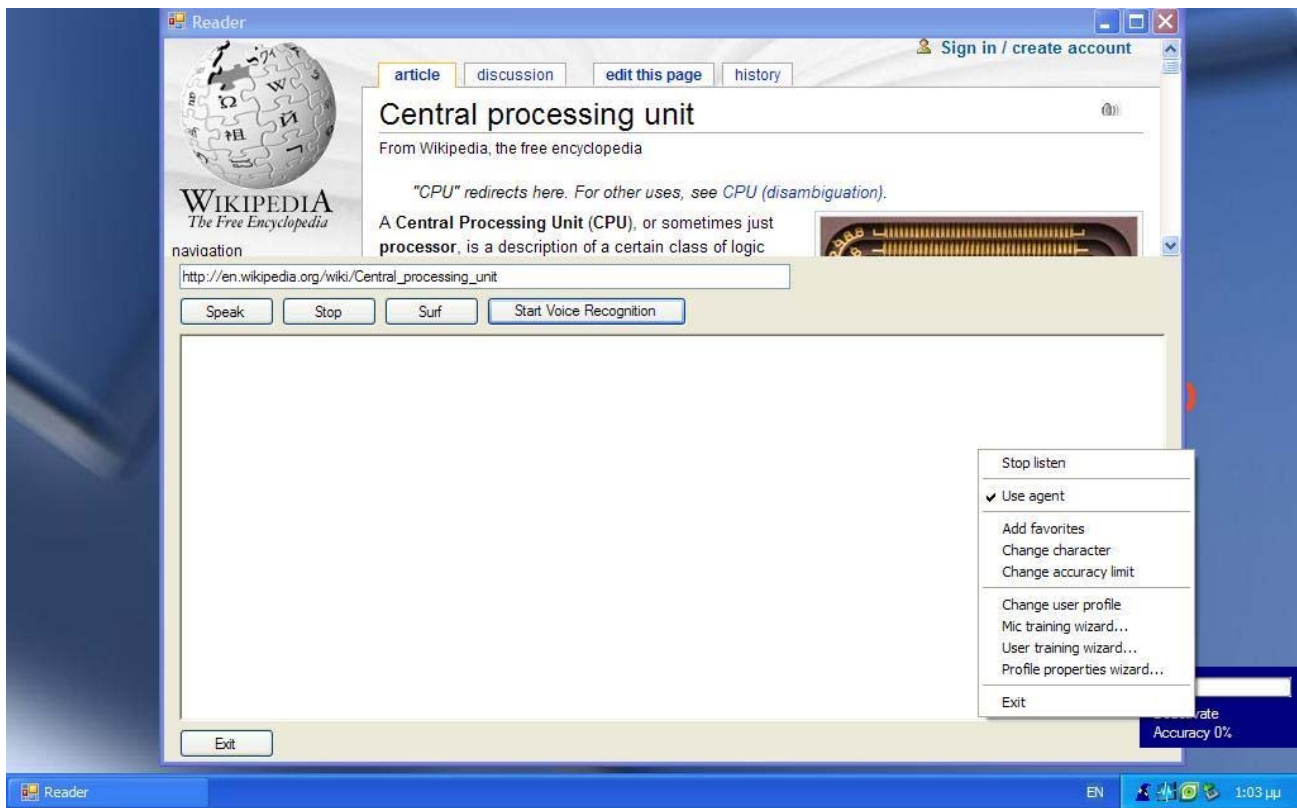


4.4 Speech recognition ή Αναγνώριση φωνής.

Όταν ο χρήστης πατήσει το κουμπί **speech recognition** ενεργοποιεί το κομμάτι του κώδικα που είναι για να ξεκινήσει την αναγνώριση ομιλίας. Αυτό το βλέπουμε στο από πάνω σχήμα.

Κώδικας :

```
frmMain frm= new frmMain();  
frm.Show();
```



Για να ξεκινήσουμε τη διαδικασία εκμάθησης του προγράμματος στη φωνή μας πρέπει να πατήσουμε δεξί click πάνω στη νέα φόρμα και να πάμε στην επιλογή Mic training wizard.

Για να καταφέρνουμε μετά την ολοκλήρωση της εκπαίδευση του να μπορούμε να δίνουμε στην εφαρμογή μας φωνητικές εντολές.

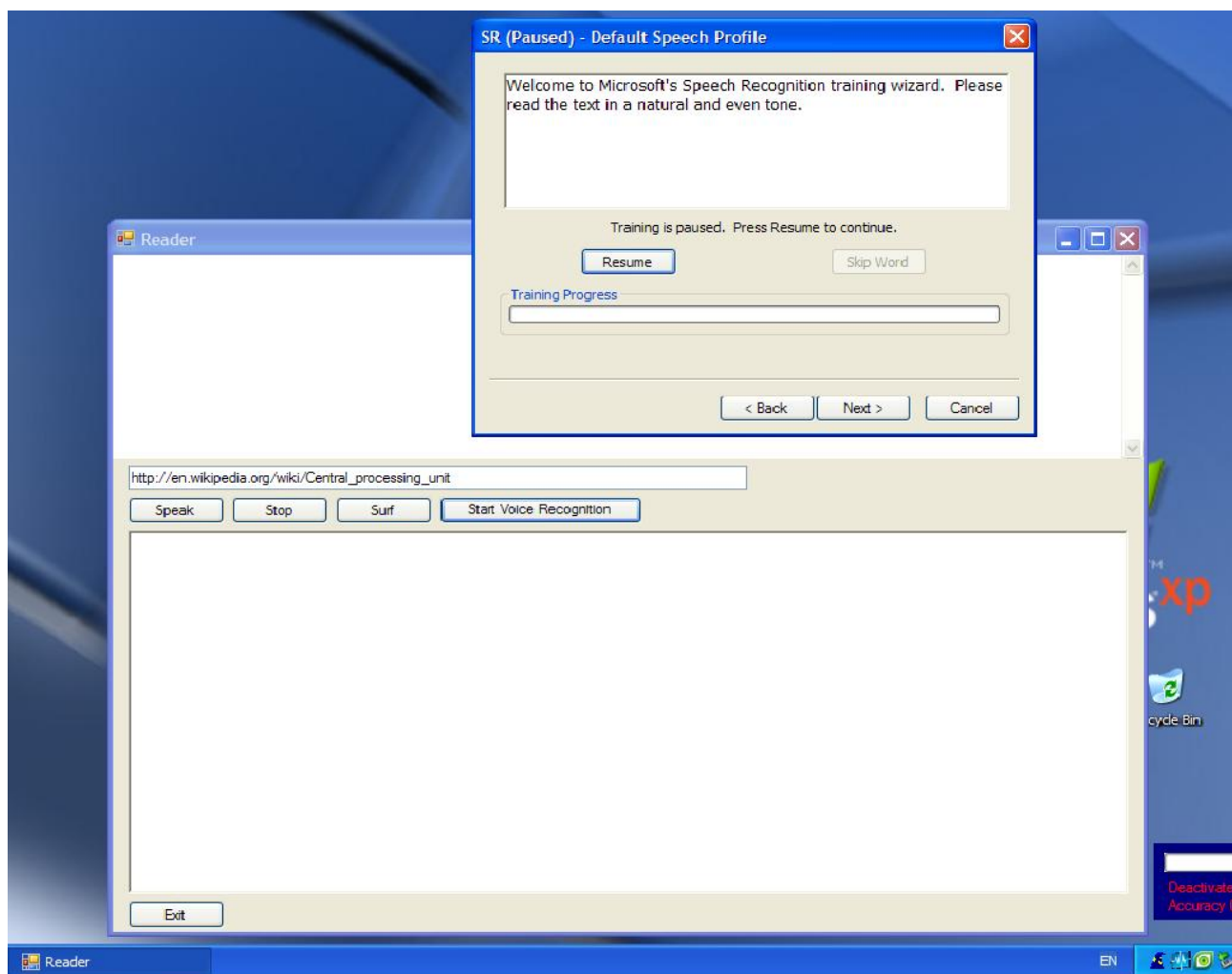
Στο σημείο αυτό να μην ξεχάσω ότι η καινούργια main χρησιμοποιεί τα δικά της dll όπως το user32. Φυσικά μπορούμε μέσω του προγράμματος να επιλέξουμε και την ακρίβεια που θέλουμε να έχουμε στο μικρόφωνο.

Με το που ενεργοποιηθεί το κομμάτι του κώδικα που καλεί τη νέα main βλέπουμε ότι εμφανίζει μια νέα φόρμα. Αυτή η νέα φόρμα είναι υπεύθυνη να κάνει όλη τη δουλειά ώστε να δουλέψει το κομμάτι του προγράμματος που είναι για την αναγνώριση ομιλίας. Μόλις πατήσουμε το κουμπί βγαίνει ένα μήνυμα που μας ζητάει να εκπαιδύσουμε το πρόγραμμα στη φωνή μας.

Πως θα γίνει κάτι τέτοιο?

Αυτό το βλέπουμε διαβάζοντας τις οδηγίες που μας εμφανίζει το ίδιο το πρόγραμμα στην οθόνη.

Δηλαδή:



Πρώτα θα μας δώσει ένα μικρό κείμενο με κάποιες λέξεις τις οποίες πρέπει να διαβάσουμε. Το πρόγραμμα σιγά σιγά αρχίζει να καταλαβαίνει τη φωνή μας και ν' αναγνωρίζει τις λέξεις απο το κείμενο. Μόλις καταλάβει μια λέξη απο το κείμενο την φωτίζει με μπλε χρώμα. Όταν έχει πια βρει όλες τις λέξεις απο το κουτάκι μας πηγαίνει

στο επόμενο. Μόλις έχουμε τελειώσει από όλα τα κουτάκια τότε το πρόγραμμα είναι έτοιμο να δεχθεί φωνητικές εντολές από το χρήστη.

Όταν ο χρήστης δίνει στην εφαρμογή ένα input το οποίο προέρχεται από το μικρόφωνο πρέπει το πρόγραμμα να μπορεί να καταλαβαίνει τις εντολές που του δίνουμε και να είναι σε θέση να τις εκτελεί.

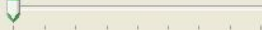
Αυτό είναι το κομμάτι που ονομάζουμε φωνητικές εντολές στο πρόγραμμα και δίνονται αποκλειστικά και μόνο από το χρήστη με τη βοήθεια ενός μικροφώνου.

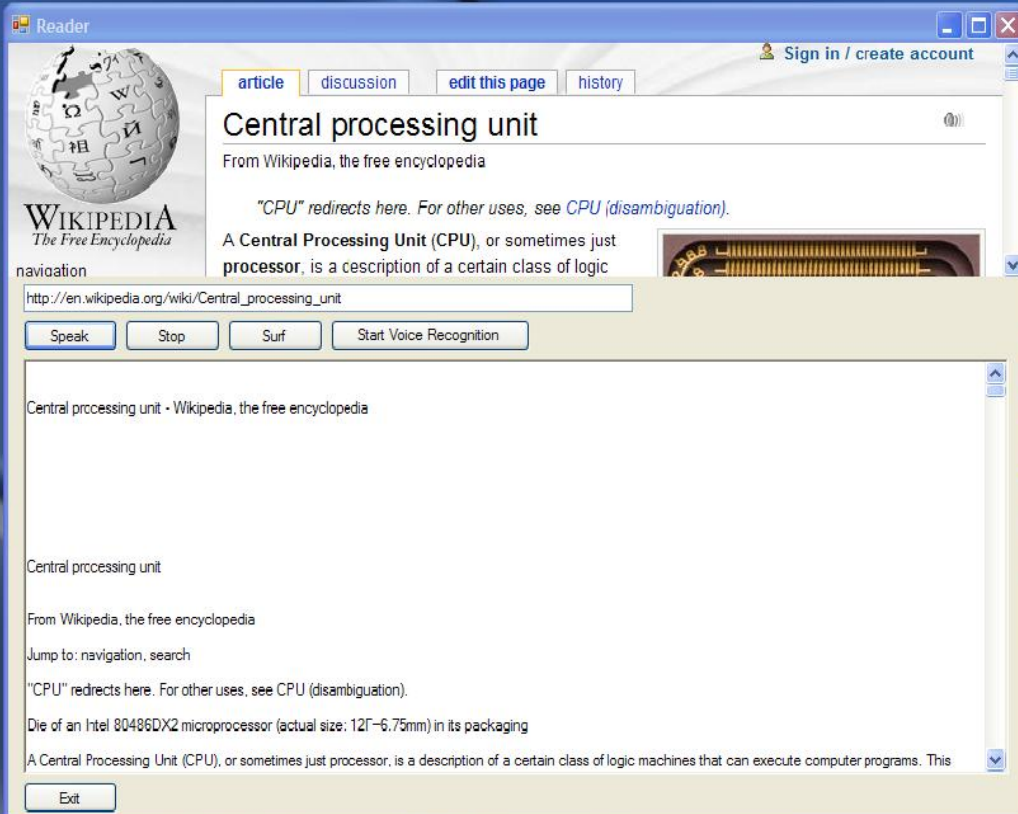
Κατά τη διάρκεια της εκπαίδευσης το πρόγραμμα καταγράφει τον τρόπο ομιλίας μας, τον τόνο της φωνής μας και το πως προφέρουμε κάποια γράμματα. Μετά πάει και συγκρίνει αυτά τα αποτελέσματα με αυτά που έχει αποθηκευμένα στη βάση δεδομένων του. Μετά τη σύγκριση κάνει τις απαραίτητες τροποποιήσεις και ρυθμίσεις ώστε να είναι έτοιμο να αναγνωρίσει τη φωνή μας. Να πούμε στο σημείο αυτό ότι η σύγκριση γίνεται με τη βοήθεια στατιστικών μοντέλων και με κάποια εφαρμογή πιθανοτήτων πάνω στο σήμα μας. Να μην ξεχνάμε ότι για τον ηλεκτρονικό υπολογιστή η φωνή μας είναι ένα ηχητικό σήμα.

Εφόσον ο υπολογιστής αποθηκεύει τη φωνή μας με μορφή σήματος μπορούμε να εφαρμόσουμε διάφορες τεχνικές για να ελέγχουμε την ορθότητα και την καθαρότητα του σήματος.

Αυτό γίνεται γιατί δεν πρέπει να ξεχνάμε ότι όταν μιλάμε για σήματα τα οποία προέρχονται ειδικά από μικρόφωνο, είναι πολύ πιθανό ότι θα υπάρχει ένα μέρος τους που θα έχει αλλοιωθεί και σίγουρα θα έχουμε και κάποιο ποσοστό θορύβου.



Speech Timer: 



Reader

Sign in / create account

article discussion edit this page history

Central processing unit

From Wikipedia, the free encyclopedia

"CPU" redirects here. For other uses, see CPU (disambiguation).

A **Central Processing Unit (CPU)**, or sometimes just **processor**, is a description of a certain class of logic

navigation

http://en.wikipedia.org/wiki/Central_processing_unit

Speak Stop Surf Start Voice Recognition

Central processing unit - Wikipedia, the free encyclopedia

Central processing unit

From Wikipedia, the free encyclopedia

Jump to: navigation, search

"CPU" redirects here. For other uses, see CPU (disambiguation).

Die of an Intel 80486DX2 microprocessor (actual size: 12Γ-6.75mm) in its packaging

A Central Processing Unit (CPU), or sometimes just processor, is a description of a certain class of logic machines that can execute computer programs. This

Exit

4.5 Speak

Τελευταίο και πιο σημαντικό έχουμε το κουμπί **speak**.

Όταν το πατήσει ο χρήστης ενεργοποιεί το κομμάτι του κώδικα που πάει και παίρνει τη διεύθυνση που του έχουμε γράψει μέσα στο `textbox1` το οποίο περιέχεται μέσα στην εφαρμογή μας. Αμέσως μετά πάει και δεσμεύει ένα δυναμικό χώρο στη μνήμη του υπολογιστή ώστε όταν έρθουν τα κείμενα να τα αποθηκεύσει εκεί. Σαν δεύτερη κίνηση ανοίγει μια σύνδεση με το διαδίκτυο φέρνει τη σελίδα στο υπολογιστή μας και ξεκινάει να παίρνει όλα τα κείμενα της σελίδας.

Σαν τρίτη κίνηση ενεργοποιεί το component που λέγεται web Browser του οποίου η λειτουργία είναι να εμφανίζει τη σελίδα που του έχουμε ζητήσει. Έτσι μέσα σε αυτό το παράθυρο βλέπουμε τη σελίδα μας.

Εφόσον έχουμε αποθηκεύσει τα κείμενα μας μέσα σε ένα buffer είμαστε πλέον έτοιμοι μας ξεκινάμε τη διαδικασία του φιλτραρίσματος.

Όταν όλα αυτά έχουν τελειώσει, ανοίγει ένα καινούργιο (pop-up) παράθυρο μέσα στο οποίο αρχίζουν να απεικονίζονται οι λέξεις και εκεί είναι όπου ενεργοποιείται το κομμάτι του κώδικα για την φωνητική αναπαράσταση των λέξεων. Δηλαδή πάει παίρνει μια προς μια τις λέξεις απο την προσωρινή αποθήκη (buffer) που του έχουμε ορίσει και ξεκινάει να τις απαγγέλει με τον τρόπο που είπαμε πιο πάνω δηλαδή λέξη προς λέξη.

Μέσα στο ίδιο παράθυρο έχουμε βάλει ένα track Bar το οποίο είναι κάτι παρόμοιο με το scroll bar. Η λειτουργία του είναι να μπορεί ο χρήστης να ρυθμίζει την ταχύτητα με την οποία εμφανίζονται οι λέξεις στην οθόνη του.

Η μονάδα μέτρησης είναι σε δευτερόλεπτα και οι ακραίες τιμές που μπορεί να πάρει είναι από το 2 μέχρι και το 10.

Αυτό γίνεται γιατί κάτω από 2 δευτερόλεπτα είναι πάρα μα πάρα πολύ μικρό το χρονικό διάστημα και ο χρήστης δεν θα προλαβαίνει να το διαβάσει.

Από την άλλη πάνω από 10 δευτερόλεπτα θα είναι πάρα πολλή αργή η ανάγνωση και ο χρήστης θα καθυστερεί πάρα πολύ να διαβάσει το κείμενο.

Με αυτά τα κριτήρια επιλέξαμε τις δυο ακραίες τιμές.

Η εντολή που πραγματοποιεί αυτό το κομμάτι του κώδικα είναι:

```
private void trackBar1_Scroll(object sender, EventArgs e)
{ timer1.Interval = trackBar1.Value * 1000; }
```

Ο timer μετράει το χρόνο σε mSeconds αλλά εμείς τον θέλουμε σε δευτερόλεπτα γιατί είναι η πιο μικρή μονάδα μέτρησης που καταλαβαίνει ο άνθρωπος.

Για το λόγο αυτό και πολλαπλασιάζουμε την τιμή του timer1 με το 1000.

5. ΘΟΡΥΒΟΣ

Τι εννοούμε όταν λέμε θόρυβος?

Ο θόρυβος είναι ένα τυχαίο φαινόμενο και πολλές φορές εμφανίζεται χωρίς να υπάρχει απαραίτητα κάποιο γεγονός ή κάποια αιτία που να το προκαλεί.

Κάποιες απο τις συνηθισμένες αιτίες που προκαλούν θόρυβο μπορεί να είναι:

α) εάν το σήμα απο την αρχή δε είναι πολύ καθαρό,

β) κατά τη διάρκεια της δειγματοληψίας αν προξενήσει κάποια μικρή παρεμβολή ή ακόμα και να υπάρξει κάποια έστω και μικρή παραμόρφωση στο σήμα μας,

γ) για κάποιο λόγο ας πούμε μ' ένα τυχαίο φαινόμενο το σήμα μπορεί να έχει υποστεί κάποια παραμόρφωση,

δ) τέλος εάν απο την αρχή κάτι δεν πήγε καλά ή κάναμε κάτι λάθος και αυτό είναι η αιτία να προξενήσει θόρυβος στο σήμα μας.

Αυτή η παραμόρφωση που έχει υποστεί το σήμα ονομάζεται θόρυβος.

Ένα μικρό μέρος απο τον κώδικα που πραγματοποιεί τις λειτουργίες της εφαρμογής μας είναι:

Για αρχή ανοίγει το μικρόφωνο και μετά φορτώνει το speech API που λέγετε για συντομία SAPI. Όταν τελειώσει αυτή η διαδικασία ξεκινάει να φορτώνει τη γραμματική. Η γραμματική περιέχει διάφορες λέξεις που είναι αποθηκευμένες μέσα σ' ένα xml αρχείο.

Τώρα θα παραθέσω ένα μικρό κομμάτι του κώδικα από το Speech recognition.

6. ΚΩΔΙΚΑΣ

```
public void RecoContext_Recognition(int StreamNumber, object
StreamPosition, SpeechRecognitionType RecognitionType, ISpeechRecoResult e)

    //calculate accuracy
    float accuracy=
(float)e.PhraseInfo.Elements.Item(0).EngineConfidence;

    //change accuracyMax dynamicly
    if (accuracyMax<accuracy)
        accuracyMax=accuracy;

    if (accuracy<0)
        accuracy=0;

    accuracy=(int)((float)accuracy/accuracyMax*100);
    label2.Text="Accuracy "+accuracy.ToString()+"%";

    //get phrase
    string phrase=e.PhraseInfo.GetText(0,-1,true);
    //make sure it's in lower case (for safer use only)
    phrase=phrase.ToLower();

    //if recognized any ...

    if (phrase!="" && accuracy>=accuracyLimit)
        {

    //Only if agent enabled
        if (menuItem14.Checked==true)
            {
                agent1.StopAll("");
                agent1.Speak(phrase,"");
            }
        }
```

```
switch (e.PhraseInfo.Rule.Name)
```

```
    //rule name (not the phrase !)  
    {
```

```
        case "Activate":  
        {
```

```
            //Only if agent enabled
```

```
            if (menuItem14.Checked==true)
```

```
            {
```

```
                //show character
```

```
                agent1.Show(false);
```

```
            }
```

```
            //load grammar
```

```
SAPIGrammarFromFile("XMLDeactivate.xml");
```

```
            //notify user
```

```
            label1.Text="Activate";
```

```
        //Only if agent enabled
```

```
        if (menuItem14.Checked==true)
```

```
        {
```

```
            //animate character
```

```
            agent1.Play("StartListening");
```

```
            agent1.Speak("I'm listening","");
```

```
        }
```

```
        break;
```

```
    }
```

```

case "Deactivate":
    {
        //load grammar
        SAPIGrammarFromFile("XMLActivate.xml");

        //notify user
        label1.Text="Deactivate";

        //Only if agent enabled
        if (menuItem14.Checked==true)
        {
            //animate character
            agent1.Play("Wave");
            agent1.Hide(false);
        }
        break;
    }

case "Start":
    {
        keybd_event((byte)Keys.LWin,0,0,0); //key
        keybd_event((byte)Keys.LWin,0,2,0); //key

        //load grammar
        SAPIGrammarFromFile("XMLStart.xml");

        //notify user
        label1.Text="Start";
        break;
    }

```

down

up

case "Right":

```
    {  
    keybd_event((byte)Keys.Right,0,0,0); //key down  
    keybd_event((byte)Keys.Right,0,2,0); //key up  
    break;  
    }
```

case "Left":

```
    {  
    keybd_event((byte)Keys.Left,0,0,0); //key down  
    keybd_event((byte)Keys.Left,0,2,0); //key up  
    break;  
    }
```

case "Up":

```
    {  
keybd_event((byte)Keys.Up,0,0,0); //key down  
    keybd_event((byte)Keys.Up,0,2,0); //key up  
    break;  
    }
```

case "Down":

```
    {  
    keybd_event((byte)Keys.Down,0,0,0); //key  
    keybd_event((byte)Keys.Down,0,2,0); //key  
    break;  
    }
```

down

up

```
case "Enter":
```

```
    {  
        keybd_event((byte)Keys.Enter,0,0,0);    //key down  
        keybd_event((byte)Keys.Enter,0,2,0);    //key up  
        break;  
    }
```

```
case "Escape":
```

```
    {  
        keybd_event((byte)Keys.Escape,0,0,0);    //key down  
        keybd_event((byte)Keys.Escape,0,2,0);    //key up  
        keybd_event((byte)Keys.LMenu,0,2,0);    //key up
```

```
//load grammar (used to reset grammar in case it contains menu stuff ...)
```

```
SAPIGrammarFromFile("XMLDeactivate.xml");
```

```
        //notify user  
        label1.Text="Activate";  
        break;  
    }
```

```
case "PureEscape":
```

```
    {  
        keybd_event((byte)Keys.Escape,0,0,0);    //key down  
        keybd_event((byte)Keys.Escape,0,2,0);    //key up  
        break;  
    }
```

```

case "Alt":
    {
    keybd_event((byte)Keys.LMenu,0,0,0); //key down
    keybd_event((byte)Keys.LMenu,0,2,0); //key up
    //check if there is any menu and hook it
    IntPtr hWnd=GetForegroundWindow();
    IntPtr hMnu=GetMenu(hWnd);
    int mnuCnt=GetMenuItemCount(hMnu);

    if (mnuCnt!=0)
    {
        //Only if agent enabled
        if (menuItem14.Checked==true)
        {
            //animate character
            agent1.Play("DoMagic1");
            agent1.Think("Hooking menu
...");

            //add menu to grammar
            hookMenu(hMnu);

            //Only if agent enabled
            if (menuItem14.Checked==true)
            {
                //animate character
                agent1.Play("Idle1_1");
            }
        }
    }
    else
    {
        //load grammar

```

```
SAPIGrammarFromFile("XMLDeactivate.xml");
```

```
    //notify user
```

```
    label1.Text="Activate";
```

```
    }
```

```
    break;
```

```
}
```

```
case "Tab":
```

```
{
```

```
    keybd_event((byte)Keys.Tab,0,0,0); //key down
```

```
    keybd_event((byte)Keys.Tab,0,2,0); //key up
```

```
    break;
```

```
}
```

```
case "ShiftTab":
```

```
{
```

```
    keybd_event((byte)Keys.LShiftKey,0,0,0); //key down
```

```
    keybd_event((byte)Keys.Tab,0,0,0); //key down
```

```
    keybd_event((byte)Keys.Tab,0,2,0); //key up
```

```
    keybd_event((byte)Keys.LShiftKey,0,2,0); //key up
```

```
    break;
```

```
}
```

```
case "CloseProgram":
```

```
{
```

```
    Close();
```

```
    break;
```

```
}
```

```
case "ShowAbout":
```

```
{
```

```
    if (frmAbout1==null)
```

```
    {
```

```
        //show frmAbout
```

```
        frmAbout1=new frmAbout();
```



```
frmAbout1.Closed+=new  
EventHandler(frmAbout1_Closed);  
//send user profile
```

```
frmAbout1.Tag=(string)objRecoContext.Recognizer.Profile.GetDescription(  
0);  
frmAbout1.Show();  
  
//load grammar  
SAPIGrammarFromFile("XMLAbout.xml");  
  
//notify user  
label1.Text="About Speech Recognition";  
break;  
}  
  
case "CloseAbout":  
{  
//close frmAbout  
if (frmAbout1!=null)  
{  
frmAbout1.Close();  
frmAbout1=null;  
}  
break;
```

7. ΣΤΟΙΧΕΙΑ ΑΝΑΠΤΥΞΗΣ ΚΑΙ ΣΧΕΔΙΑΣΜΟΥ ΤΟΥ ΚΩΔΙΚΑ

Στο σημείο αυτό να τονίσουμε ότι η εφαρμογή είναι κατασκευασμένη με τέτοιο τρόπο ώστε κάθε κομμάτι που την απαρτίζει να είναι ανεξάρτητο από όλα τα άλλα.

Αυτό που θέλω να πω είναι ότι κάθε συνάρτηση κάνει χρήση των δικών της μεταβλητών και έχει τα δικά της ορίσματα.

Με αυτόν τον τρόπο διασφαλίζουμε ότι ακόμα και αν κάποιο κομμάτι της εφαρμογής έχει κάποιο πρόβλημα ή αν για κάποιο λόγο στο τέλος παρουσιάσει κάποιο δεν πρόκειται να επηρεάσει τις λειτουργίες των υπόλοιπων κομματιών ή τέλος πάντων όχι σε τέτοιο βαθμό που να έχουμε τρομερά λάθη στα αποτελέσματα που θα μας δώσει το πρόγραμμα σαν έξοδο.

Ένα ακόμα μεγάλο καλό που προκύπτει από αυτό τον τρόπο ανάπτυξης είναι ότι ακόμα και αν υπάρξει κάποια παρανόηση κατά την καταγραφή των απαιτήσεων ή στις λειτουργικές προδιαγραφές της εφαρμογής τότε η διόρθωση θα γίνει πολύ πιο εύκολα μιας και χρειάζεται να κάνουμε κάποιες πολύ μικρές διορθώσεις στο συγκεκριμένο κομμάτι του κώδικα και μόνο.

Με αυτόν τον τρόπο καταφέρνουμε να κάνουμε εύκολα τις όποιες αλλαγές χρειάζονται στην εφαρμογή μας και φυσικά αυτές ισοδυναμούν μ' ένα μόνο μικρό και αυτοδύναμο κομμάτι του κώδικα συνολικά και φυσικά δεν συγκρίνονται με τις αλλαγές που θα χρειαζόταν να κάνουμε αν είχαμε υιοθέτηση ένα κλασικό μοντέλο ανάπτυξης για την κατασκευή της εφαρμογής μας.

Με τον τρόπο αυτό πετυχαίνουμε να έχουμε καλύτερο έλεγχο της εφαρμογής μας συνολικά. Αν όμως όπως προείπαμε για κάποιο λόγο έχουν γίνει λάθη είτε από δική μας παρερμηνευση είτε από τη μεριά του πελάτη, το κομμάτι της διόρθωσης θα είναι σαφώς πιο εύκολο, πιο δομημένο και σαφώς πιο γρήγορο στην προσπέλαση του. Σαν συνέπεια αυτού, η διόρθωση γίνεται πιο γρήγορα με λιγότερο κόπο, με σαφώς μικρότερο κόστος και με λιγότερα έξοδα.

Ένα ακόμα καλό που προκύπτει από αυτόν τον τρόπο ανάπτυξης μιας εφαρμογής είναι και το reusability!

Δηλαδή όταν ο κώδικας είναι σωστά γραμμένος και χωρισμένος σε μικρά αυτόνομα κομμάτια τότε πιθανόν κάποια απο αυτά τα κομμάτια να μπορούν να ξαναχρησιμοποιηθούν σε κάποια άλλη εφαρμογή.

Στο σημείο αυτό να τονίσουμε ότι επειδή τα αποτελέσματα απο τη μια συνάρτηση χρησιμοποιούνται σαν είσοδο για τις άλλες συναρτήσεις αν προκύψει κάποιο πρόβλημα στα αποτελέσματα της μιας τότε και τ' αποτελέσματα των άλλων θα είναι λάθος σαν λογικό επακόλουθο.

Βέβαια εμείς επειδή ήδη το προβλέψαμε για το λόγο αυτό έχουμε κατασκευάσει έτσι την εφαρμογή ώστε να είναι δομημένη σε ανεξάρτητα κομμάτια. Μόλις λοιπόν προκύψει κάποιο πρόβλημα να είναι όσο το δυνατόν πιο εύκολο και γρήγορο να γίνει η διόρθωση.

Με τον τρόπο αυτό κερδίζουμε αρκετό χρόνο και γλυτώνουμε πολλά λάθη. Αυτό γίνεται γιατί είναι πιο εύκολο να διαβάσεις ένα μικρό κομμάτι απο τον κώδικα της εφαρμογής και να το διορθώσεις ή να αλλάξεις τυχόν λάθη παρά να κάθεσαι και να έχεις μπροστά σου ένα σωρό απο γραμμές κώδικα και στο τέλος να μην θυμάσαι ούτε ο ίδιος τι ακριβώς γίνεται πιο πάνω ή πιο κάτω.

Τέλος για το κομμάτι της εφαρμογής που είναι για την αναγνώριση ομιλίας.

Να πούμε ότι όταν καλούμε τη νέα main φορτώνουμε ένα καινούργιο πρόγραμμα το οποίο κάνει χρήση δικών του αυτόνομων συναρτήσεων και μεταβλητών που λέγονται `SpeechRecognitionTypeSpeechObjectRecognition`. Αυτές οι συναρτήσεις είναι που μας βοηθάνε στο να μπορέσουμε να ολοκληρώσουμε το κομμάτι του κώδικα το οποίο πραγματοποιεί την αναγνώριση της φωνής και να είναι πλήρως λειτουργικό.

Οι `Speech Recognition` συναρτήσεις είναι υπεύθυνες για να πάρουν το εισερχόμενο σήμα απο το μικρόφωνο να εφαρμόσουν πάνω του διάφορα μοντέλα πιθανοτήτων και στατιστικής ώστε να είναι ικανά να το συγκρίνουν με αυτά που έχουν αποθηκευμένα στη βάση δεδομένων τους. Έτσι ποια είμαστε σε θέση να πραγματοποιήσουμε αυτό το δύσκολο κομμάτι της εφαρμογής μας.

8. ΦΙΛΤΡΑΡΙΣΜΑ

Όταν το πρόγραμμα μας έχει τελειώσει με την αποθήκευση των γραμμάτων στο buffer ξεκινάει να τα διαβάζει απο αυτό. Τα γράμματα διαβάζονται με έναν πολύ συγκεκριμένο τρόπο δηλαδή σαν λέξη προς λέξη. Στο σημείο αυτό να τονίσουμε ότι ξεκινάμε απο το πρώτο γράμμα και διαβάζουμε μέχρι τον πρώτο κενό χαρακτήρα.

Μόλις το πρόγραμμα έχει βρει μια λέξη κοιτάει μήπως είναι στη λίστα με αυτές που πρέπει να φιλτραριστούν. Αυτό που θέλουμε να πούμε, είναι οτι πρέπει να κοπούν ή αλλιώς να γίνει αντικατάσταση τους με ένα κενό (null) χαρακτήρα.

Με τον τρόπο αυτό λοιπόν ξεκινάει η διαδικασία του φιλτραρίσματος.

8.1 Τι εννοούμε όταν λέμε φιλτράρισμα.

Το πρόγραμμα αρχίζει να διαβάζει μια προς μια τις λέξεις που είναι αποθηκευμένες στο buffer. Όταν δει οτι κάποια λέξη που περιέχεται μεν στο κείμενο αλλά στην πραγματικότητα είναι οδηγία για τον internet explorer τότε πρέπει να κάνουμε τις απαραίτητες ενέργειες έτσι ώστε να τη φιλτράρουμε δηλαδή να την αφαιρέσουμε.

Αυτό που θέλω να πω είναι οτι όταν θέλουμε ν' αλλάξουμε σειρά στο κείμενο μας, στο τέλος της σειράς γράφουμε την εντολή “ /n ” ή όταν θέλουμε να κάνουμε τα γράμματα πιο έντονα βάζουμε το κείμενο μας ανάμεσα σε [b] και [/b]. Αυτές οι εντολές περιέχονται στα κείμενα των σελίδων και έχουν τη μορφή λέξεων αλλά στην πραγματικότητα δεν είναι.

Εμείς πρέπει να πάρουμε τα κείμενα της σελίδας αλλά πρέπει να κόψουμε τις οδηγίες που συμπεριλαμβάνονται και οι οποίες είναι οδηγίες για τον εκάστοτε explorer που χρησιμοποιούμε για το διαδίκτυο.

Υπάρχουν διάφορα πράγματα που πρέπει να φιλτραριστούν.

Μεταξύ αυτών είναι και οι οδηγίες για τον internet explorer, κάποια απο τα διάφορα java script's που χρησιμοποιούνται γενικά απο τις σελίδες, κάποιες οδηγίες οι οποίες είναι μέσα σε αρχεία τύπου xml και διάφορα άλλα όπως link's για εικόνες ,για άλλες σελίδες και τα λοιπά.

Αυτό θα το κάνουμε με το να ελέγχουμε το κείμενο μας λέξη προς λέξη και όπου βρίσκουμε κάποια λέξη που να ταιριάζει με τα pattern που έχουμε ορίσει, θα πρέπει να τη φιλτράρουμε και να φροντίζουμε ώστε να αντικαθίστανται με κενούς χαρακτήρες.

Οπότε μόλις η συνάρτηση διαβάσει μια λέξη που δεν πρέπει να εμφανίζεται γιατί απλά είναι οδηγία για τον explorer που μας δείχνει τη σελίδα, τότε πολύ απλά το παίρνει και του κάνει μια αντικατάσταση μ' ένα κενό χαρακτήρα δηλαδή με αυτό ().

Πώς όμως κάνουμε την αντικατάσταση (replace)?

Η εντολή που μας βοηθάει δεν είναι άλλη απο τη regex. Τα ορίσματα της είναι πολύ απλά δηλαδή ο ορισμός γίνετε με τον εξής απλό τρόπο.

Η λέξη που πρέπει να αντικατασταθεί , η λέξη που θα την αντικαταστήσει.

Ένα παράδειγμα είναι αυτό: regex("target",replace char).

Εδώ να πούμε οτι λόγω της φύσεως της HTML πρέπει να μπουν μια σειρά απο regex γιατί έχουμε ένα σωρό απο tags και java scripts που πρέπει να κόψουμε.

Αυτό το κομμάτι του κώδικα μας πήρε πολύ χρόνο γιατί πρέπει να βρούμε και να κάνουμε replace σε όλα αυτά τα tags.Βέβαια είχαμε την μεγάλη βοήθεια η οποία δεν είναι άλλη απο το διαδίκτυο (internet). Εδώ χρειάστηκε να διαβάσω και κάποια βιβλία για τη δομή και τον τρόπο σύνταξης της HTML, ώστε να είμαι σε θέση να κατανοήσω τον τρόπο σύνταξης αυτής της γλώσσας και να καταφέρω να αφαιρέσω τα κομμάτια που δεν είναι χρήσιμα.

8.2 Η διαδικασία του φιλτραρίσματος

Όταν ο χρήστης πατήσει το κουμπί speak λέει στο πρόγραμμα να πάει να διαβάσει απο το textbox που του έχουμε ορίσει τη διεύθυνση της σελίδας που θέλουμε να μας εμφανίσει στην οθόνη μας. Μετά να πάει να πάρει τα κείμενα της σελίδας να τα φιλτράρει και να τα εμφανίσει στην οθόνη. Τα κείμενα όμως περιέχουν εκτός απο χρήσιμες πληροφορίες και διάφορα άλλα HTM ή HTML tags.Όπως είναι λογικό αυτά είναι τα στοιχεία πρέπει να φιλτραριστούν δηλαδή να κοπούν.

Αυτή η διαδικασία είναι αρκετά δύσκολη και αρκετά χρονοβόρα.

Έχει έρθει πλέον η ώρα να την περιγράψουμε αναλυτικά.

Ξεκινάμε λοιπόν απο τα απλά βήματα και σιγά σιγά θα πηγαίνουμε στα πιο σύνθετα.

Πρώτα απο όλα πρέπει να πάμε στη σελίδα που έχει ορίσει ο χρήστης και να την τραβήξουμε στον υπολογιστή μας. Αυτό το επιτυγχάνουμε μ' ένα component που λέγεται web Client και η μεταβλητή που χρησιμοποιούμε στην εφαρμογή μας λέγεται client.

Ο web Client μιας και είναι κλάση έχει μια ιδιότητα που είναι το download Data.Αυτή η ιδιότητα πάει παίρνει όλη τη σελίδα και την αποθηκεύει σε μια προσωρινή θέση στη μνήμη (buffer) που έχει τη μορφή byte.

Άρα σαν συνέπεια αυτού όταν θέλουμε να κατεβάσουμε τη σελίδα στον υπολογιστή μας με απώτερο σκοπό να φιλτράρουμε όλα τα αχρηστα κατ' εμάς στοιχεία γράφουμε τις εξής εντολές:

```
web Client client= new web Client() ;  
Byte page Data = client.downloadData(textBox1.text);
```

Αυτό που κάνουν αυτές οι δυο γραμμές είναι το εξής απλό ,στην πρώτη γραμμή φτιάχνουμε μια μεταβλητή που είναι τύπου webClient και της δίνουμε τ' όνομα client.Μετά το αντικείμενο client του λέμε να γίνει ένας νέος webClient ώστε να είμαστε σε θέση να το χρησιμοποιήσουμε.

Αφού γίνουν όλα αυτά προχωράμε πλέον στη δεύτερη γραμμή του κώδικα μας όπου μπορούμε να δούμε οτι εδώ φτιάχνουμε μια μεταβλητή τύπου Byte και την ονομάζουμε pageDate.Τώρα βλέπουμε οτι το αντικείμενο client έχει μια ιδιότητα (property) που λέγεται downloadData και σαν όρισμα του δίνουμε τη διεύθυνση (url) απο το textBox που με τη σειρά του έχει πάρει τη διεύθυνση που του έχει δώσει ο χρήστης .Άρα λοιπόν ο client πάει παίρνει τη διεύθυνση απο το χρήστη και τραβάει ολόκληρη τη σελίδα στον υπολογιστή μας. Αφού έχει κατεβάσει ολόκληρη τη σελίδα στον υπολογιστή μας την αποθηκεύει στη μεταβλητή pageData.

Με αυτόν τον τρόπο έχουμε καταφέρει να φέρουμε τη σελίδα απο το διαδίκτυο στον υπολογιστή μας. Βέβαια να μην ξεχάσουμε οτι η σελίδα είναι αποθηκευμένη σε μορφή Byte.Ο χρήστης όμως μπορεί να διαβάσει μόνο κείμενο άρα πρέπει να κάνουμε μια μετατροπή στα δεδομένα μας ώστε να γίνουν απο byte σε κείμενο. Αυτό γίνεται πολύ εύκολα με μια απλή εντολή.

Αυτή δεν είναι άλλη απο την :

```
String pageHtml=encoding.getEncoding(1253).getText();
```

Δηλαδή φτιάχνουμε ένα string και του ορίζουμε την κωδικοποίηση των Ελληνικών που είναι το 1253. Με αυτή την εντολή λέμε στο πρόγραμμα να πάει να πάρει τα δεδομένα που είναι σε μορφή byte και να τα μετατρέψει σε κείμενο (string).

Αυτή η μετατροπή γίνεται αυτόματα στα Αγγλικά αλλά για τα Ελληνικά πρέπει να του δώσουμε την κωδικοποίηση με το χέρι.

Μέχρι αυτό το βήμα έχουμε πάρει τη σελίδα την έχουμε φέρει στον υπολογιστή μας και την έχουμε μετατρέψει σε κείμενο. Τώρα πια μπορούμε να την υποβάλλουμε σε μια διαδικασία φιλτραρίσματος έτσι ώστε να αφαιρέσουμε όλα τα Html ,Htm ,java script tags που υπάρχουν στη σελίδα ώστε να μείνει μόνο το καθαρό κείμενο.

Το φιλτράρισμα γίνεται με μια σειρά απο `regeX()`. Η `regeX` είναι μια εντολή η οποία ψάχνει το κείμενο και όταν βρει τη λέξη που της έχουμε δώσει στο πρώτο όρισμα την αντικαθιστά με αυτή που έχει στο δεύτερο όρισμα. Εμείς σαν πρώτο όρισμα έχουμε δώσει τα Html tags και φυσικά σαν δεύτερο το κενό χαρακτήρα δηλαδή `null`.

Όλες οι `regeX()` είναι μέσα σε μια συνάρτηση που λέγεται **`removeScripts`**.

Μόλις αφαιρεθεί η πλειοψηφία απο τα `script` το κείμενο γίνεται `return` δηλαδή το επιστρέφουμε σε μια άλλη συνάρτηση που λέγεται **`getText`**. Αυτή με τη σειρά της αφαιρεί τα όσα εναπομείναντα tags όπως το `/n` ,`/t` ,`/b` . Όταν ποια έχουν φύγει όλα τα Html tags το κείμενο επιστρέφεται και είναι πλέον καθαρό απο όλα τα περιττά στοιχεία. Αφού έχει γίνει ποιά και αυτή η δουλειά τώρα αναλαμβάνει δουλειά ένα άλλο κομμάτι της εφαρμογής.

Αυτό το κομμάτι είναι που φροντίζει να φορτωθούν οι λέξεις σ' ένα `buffer` (προσωρινή θέση) ώστε να μπορέσει να τις απαγγείλει η εφαρμογή μας. Για να γίνει όμως αυτό πρέπει το κομμάτι του κώδικα που είναι για αυτή τη δουλειά να ενεργοποιηθεί δηλαδή να ελέγξει αν οι λέξεις είναι σωστές να τις καταχωρήσει και μετά να τις απαγγείλει.

Αυτό γίνεται με τη βοήθεια του Microsoft Speech SDK 5.1 .

9. MICROSOFT SPEECH SDK 5.1

Αμέσως μετά το φιλτράρισμα των γραμμάτων ξεκινάει η επόμενη διαδικασία.

Η δουλειά της είναι να παίρνει το φιλτραρισμένο κείμενο και να ξεκινάει την απαγγελία των γραμμάτων με τον ίδιο τρόπο όπως και πριν δηλαδή λέξη προς λέξη.

Οι λέξεις απεικονίζονται στη νοηματική γλώσσα. Παράλληλα με την απεικόνιση τους στην οθόνη γίνεται και απαγγελία των λέξεων.

Αυτή η απεικόνιση γίνεται στη νοηματική γλώσσα. Ευτυχώς έχουμε ένα εργαλείο που μας βοηθάει στο να το πραγματοποιήσουμε. Αυτό φυσικά δεν είναι άλλο από το δακτυλικό αλφάβητο.

Με αυτόν τον τρόπο μπορούμε να κάνουμε τ' άτομα με σοβαρά προβλήματα στην ακοή να μπορούν και αυτοί με τη σειρά τους να χρησιμοποιήσουν την εφαρμογή μας.

Έτσι είναι εφικτό και γι' αυτούς να μπορούν να πραγματοποιήσουν μια περιήγηση στο internet και να μπορούν να καλύψουν κάποιες ανάγκες τους.

10. ΒΙΒΛΙΟΓΡΑΦΙΑ

Χρησιμοποιήθηκε υλικό απο τις παρακάτω ιστοσελίδες :

- Tutorial της Microsoft για το visual studio .net 2005
- Tutorial της Microsoft για το C# .net 2005
- Tutorial της Microsoft για το speech Technologies.
- en.wikipedia.org/wiki/Speech_Application_Programming_Interface
- Tutorial της Microsoft για το speech recognition