

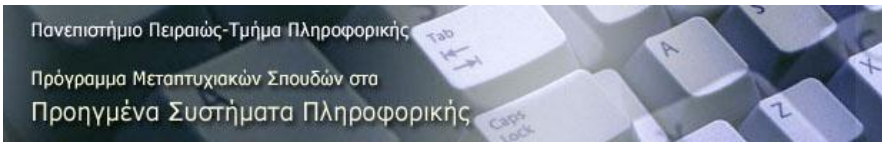


Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Επικοινωνία ευφυών πρακτόρων σε εικονικά περιβάλλοντα με χρήση φυσικής γλώσσας που μετατρέπεται σε ομιλία
Όνοματεπώνυμο Φοιτητή	Γεωργίτσης Δημήτριος του Δημητρίου
Αριθμός Μητρώου	ΜΠΣΠ07007
Κατεύθυνση	Ευφυείς Τεχνολογίες Αλληλεπίδρασης Ανθρώπου-Υπολογιστή
Επιβλέπων	Θεμιστοκλής Παναγιωτόπουλος, Καθηγητής



Πανεπιστήμιο Πειραιώς-Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών στα
Προηγμένα Συστήματα Πληροφορικής

Ημερομηνία Παράδοσης **Νοέμβριος 2010**



Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Περίληψη

Τα εικονικά περιβάλλοντα και οι ευφυείς πράκτορες που ενεργούν μέσα σε αυτά είναι ένας τομέας ο οποίος μπορεί να έχει πάρα πολλές εφαρμογές. Στα γνωσιακά χαρακτηριστικά των ευφυών πρακτόρων συγκαταλέγεται η ικανότητά τους να επικοινωνούν μεταξύ τους όσο και με χρήστες που εισέρχονται στα εικονικά περιβάλλοντα. Για να γίνει πιο επιτυχής η προσομοίωση αυτής της επικοινωνίας θα έπρεπε ο ευφυής πράκτορας να μπορεί να συνδιαλλάσσεται με τις υπόλοιπες οντότητες γύρω του με χρήση φυσικής γλώσσας. Στα πλαίσια αυτής της μεταπτυχιακής διατριβής γίνεται εισαγωγή με χρήση VERL ενός εικονικού περιβάλλοντος στο σύστημα REVE Worlds. Στο εικονικό περιβάλλον αυτό συνδέονται ευφυείς πράκτορες στους οποίους έχει δοθεί η ικανότητα να αντιλαμβάνονται και να δημιουργούν φυσική γλώσσα μέσω προγράμματος τύπου ELIZA. Τέλος για ακόμα πιο ρεαλιστικό αποτέλεσμα χρησιμοποιείται τεχνολογία σύνθεσης ομιλίας στον ευφυή πράκτορα ώστε η φυσική γλώσσα που παράγεται από την ELIZA να μετατρέπεται σε ήχο. Γίνεται βιβλιογραφική μελέτη των συστημάτων ομιλίας και κυρίως των συστημάτων μετατροπής κειμένου σε ομιλία, των βασικών τεχνολογιών και των αλγορίθμων που χρησιμοποιούν. Ιδιαίτερη έμφαση δίνεται στο σύστημα FreeTTS στο οποίο βασίζεται η υλοποίηση για τη δημιουργία ομιλίας. Τα παραπάνω συνδυάζονται σε ένα σενάριο ικανοποίησης ενός αιτήματος από ένα ευφυή πράκτορα με εμπλοκή άλλων δύο ευφυών πρακτόρων χωρίς παρέμβαση χρήστη με επικοινωνία μέσω φυσικής γλώσσας η οποία μετατρέπεται σε ήχο. Τέλος γίνεται μία αναφορά σε μελλοντικές υλοποιήσεις που θα είχαν ενδιαφέρον ώστε να εξελίξουν και να βελτιώσουν το μοντέλο επικοινωνίας που αποτυπώνεται στα πλαίσια αυτής της διατριβής.

Λέξεις Κλειδιά: ELIZA, FreeTTS, TTS Systems, Intelligent Agents , Σύνθεση Ομιλίας, chatter bots, virtual environments

Abstract

Virtual Environments and intelligent agents that interact within are a field that can have a lot of applications. One of the agents' knowledge characteristics is their ability to interact with each other as well as with users that enter the same virtual environments. To make the simulation of this communication better the intelligent agent should be able to interact using natural language. Within this postgraduate dissertation a virtual environment is imported in the REVE Worlds system using VERL. Intelligent Agents interact in it who have been given the ability to create and comprehend natural language through an ELIZA-type system. Finally to achieve an even more realistic simulation Speech generation technology is applied to the intelligent agent so that the natural language produced is converted to sound. There is a study of the various speech generation systems as well as systems that convert text to speech, the technology involved and the algorithms that are being used. The FreeTTS system is thoroughly analyzed as it is the system on which the text to speech generation capabilities of the intelligent agents are based. All of the above are combined in a request fulfillment scenario of an intelligent agent that includes the actions of two more. Their communication based solely on natural language that is converted to speech. Finally there is a short description of future projects that could be developed in order to expand and improve the communication model that is studied in this dissertation.

Keywords: ELIZA, FreeTTS, TTS Systems, Intelligent Agents , speech synthesis, chatter bots, virtual environments

Ευχαριστίες

Η παρούσα μεταπτυχιακή διατριβή εκπονήθηκε στα πλαίσια του μεταπτυχιακού προγράμματος «Προηγμένα Συστήματα Πληροφορικής» του Πανεπιστημίου Πειραιά υπό την επίβλεψη του καθηγητή Θ. Παναγιωτόπουλου τον οποίο και θα ήθελα να ευχαριστήσω για την καθοδήγηση, την υπομονή και το ενδιαφέρον που έδειξε για τη συγγραφή αυτής της διατριβής. Επίσης θα ήθελα να ευχαριστήσω το Δρ. Γ. Αναστασάκη για την πολύτιμη βοήθειά του και την άριστη συνεργασία μας. Τέλος θα ήθελα να ευχαριστήσω το συνάδελφο Ε. Γιακουμή καθώς το αποτέλεσμα που επιτυγχάνεται είναι απόρροια της συνεργασίας μας και του συνδυασμού των μεταπτυχιακών διατριβών μας.

Περιεχόμενα

Περίληψη	3
Abstract	4
Ευχαριστίες	5
Εισαγωγή – Σύντομη Περιγραφή Προβλήματος/Αντικειμένου	9
Εικονικά περιβάλλοντα	9
Εικονική πραγματικότητα και εικονικά περιβάλλοντα	9
Εφαρμογές εικονικών περιβαλλόντων	10
Αντικείμενα σε εικονικό περιβάλλον	14
Αναπαράσταση αντικειμένων	14
Item Aspect	15
Γλώσσα VERL	15
Αναπαράσταση εικονικού πράκτορα	20
Εισαγωγή	20
Ο εικονικός πράκτορας ως item	20
Perception aspect	21
Λήψη δεδομένων αντίληψης	22
Επεξεργασία δεδομένων	23
Αποθήκευση γνώσης	23
Activity aspect	23
Agent Action	24
Ευφυής Πράκτορας και ομιλία	25
Εισαγωγή	25
Εικονικό σώμα	25
Physical Aspect	26
Semantic aspect	26
Access aspect	26
Activity Aspect	27
Πλατφόρμα jReve	27
Εισαγωγή	27
Extensible 3D Graphics (X3D) και Xj3D	28
Συμβολική αναπαράσταση γνώσης	29
Κύκλος ζωής	29
Ομιλία και τεχνολογίες ομιλίας	31
Ομιλία ως μέσο επικοινωνίας	31
Υποκατηγορίες τεχνολογίας ομιλίας	34
Σύνθεση φωνής	36
Ιστορική αναδρομή	36

Τεχνολογία και Συστήματα Μετατροπής Κειμένου σε Ομιλία... 38	38
Ανάλυση Συστημάτων Μετατροπής Κειμένου σε Ομιλία..... 40	40
Υλοποίηση συστήματος ΜΚσΟ σε ευφυή πράκτορα..... 40	40
Java Speech Api..... 41	41
FreeTTS 41	41
VoiceGenerator 44	44
Φυσική γλώσσα 45	45
Επεξεργασία Φυσικής Γλώσσας 45	45
Ιστορική αναδρομή 45	45
Chatter bots..... 46	46
Πρόγραμμα Eliza 46	46
Υλοποίηση φυσικής γλώσσας σε ευφυή πράκτορα 46	46
Case Study 47	47
Τεχνολογίες που χρησιμοποιήθηκαν 47	47
Γλώσσα Java..... 47	47
Περιβάλλον Netbeans..... 48	48
Σενάριο 49	49
Virtual environment (bar)..... 50	50
Agents in bar..... 54	54
Πράκτορας Customer 54	54
Πράκτορας Waiter 55	55
Πράκτορας Barman 55	55
Συμπεράσματα..... 56	56
Voice Recognition..... 56	56
Επικοινωνία στα Ελληνικά..... 57	57
Interfaces..... 57	57
Βιβλιογραφία..... 59	59
Παράρτημα 62	62
Οδηγός για προσθήκη των δυνατοτήτων στον Ευφυή Πράκτορα ... 62	62
Προσθήκη effector Talk στο acces point Mouth στον Agent.... 62	62
Προσθήκη nodes στο vrml μοντέλο μας..... 64	64
Προσθήκη του SpeechModule στον κώδικα του Ε.Π..... 65	65
Επεξήγηση του SpeechModule..... 68	68
Χρήση της ομιλίας..... 70	70

Εικόνες

Στιγμιότυπο από τον Cosmo	11
Perception Aspect.....	22
Activity Aspect	24
Η αλυσίδα επικοινωνίας από άνθρωπο σε άνθρωπο μέσω ομιλίας.....	32
Μπλοκ διάγραμμα του μηχανισμού παραγωγής ομιλίας.....	33
Παράδειγμα ροής αέρα εξόδου από την γλωττίδα κατά την διάρκεια ηχηρής ομιλίας	33
Παράδειγμα κυματομορφής ομιλίας όπου φαίνονται τα αντίστοιχα pitchmarks.....	34
Σύστημα αμφίδρομης φωνητικής απόκρισης - ΑΦΑ, (Interactive Voice Response – IVR).....	34
Μπλοκ διάγραμμα ενός τυπικού διαλογικού συστήματος.....	35
Κατηγορίες της τεχνολογίας ομιλίας.....	36
Οι ακουστικοί ταλαντωτές του Christian Kratzenstein, 1779	36
Η μηχανές των (α) Wolfgang von Kempelen και (β) Charles Wheatstone	37
Ιστορικό χρονοδιάγραμμα της σύνθεσης φωνής	38
Λειτουργικό διάγραμμα συστήματος ΜΚσΟ	40
Οι δομές φωνής και έκφρασης.....	42
Βήματα της Λειτουργίας της Σύνθεσης του συστήματος FreeTTS.....	44
Java.....	47
Γενική απεικόνιση του εικονικού περιβάλλοντος.....	50
Εσωτερικό του εικονικού περιβάλλοντος.....	51
Παράδειγμα items και items groups.....	52
Πράκτορας Customer.....	54
Ο πράκτορας Waiter με τον Customer.....	55
Ο πράκτορας barman.....	56
Intelligent Agent Full Communication Interface	58
Intelligent Agent Communication Basic Interface	58

Εισαγωγή – Σύντομη Περιγραφή Προβλήματος/Αντικειμένου

Η συνεχής εξέλιξη στα εικονικά περιβάλλοντα σε συνδυασμό με τις αυξανόμενες δυνατότητες των ευφυών οντοτήτων που δρουν μέσα σε αυτά έχουν οδηγήσει σε πληθώρα εφαρμογών τα τελευταία χρόνια. Όσο πιο ρεαλιστικό είναι ένα εικονικό περιβάλλον και όσο πιο ρεαλιστικές είναι οι συμπεριφορές των ευφυών πρακτόρων που δρουν μέσα σε αυτό τόσο πιο επιτυχημένη είναι η προσομοίωση. Η παρούσα διατριβή ασχολείται με την επικοινωνία ανάμεσα στους ευφείς πράκτορες και την όσο το δυνατόν πλησιέστερη προσέγγισή της προς την ανθρώπινη επικοινωνία. Οι πράκτορες θα πρέπει να μπορούν να επικοινωνούν με χρήση φυσικής γλώσσας, να υπάρχει φωνητική επικοινωνία και όχι απλώς κείμενο και να μπορούν εκτός από την ανταλλαγή πληροφοριών να δώσουν παραγγέλματα σε άλλους πράκτορες. Αυτό μπορεί να επιτευχθεί καλύπτοντας τέσσερις βασικές δυνατότητες ως προς τον κάθε πράκτορα. Τη δυνατότητα κατανόησης φυσικής γλώσσας, τη δυνατότητα δημιουργίας φυσικής γλώσσας, τη δυνατότητα να μετατρέπει σε φωνή τις προτάσεις που δημιουργήσει και τη δυνατότητα να καταλαβαίνει φωνητικά σήματα και να τα επεξεργάζεται ως φυσική γλώσσα. Το αποτέλεσμα στο οποίο αποσκοπεί αυτή η ανάλυση και υλοποίηση είναι να δημιουργηθεί ένα πρότυπο το οποίο θα μπορούσε να παρέχει εύκολα τη δυνατότητα να προσαρμοστούν πάνω του εύκολα διαφορετικές υλοποιήσεις. Επίσης σε συνδυασμό με τη μεταπτυχιακή διατριβή του Ε.Γιακουμή θα δοθεί ένα πλήρες και λειτουργικό σενάριο που να επιτυγχάνει το στόχο της διατριβής.

Εικονικά περιβάλλοντα

Το παρόν κεφάλαιο δίνει μια συνοπτική περιγραφή των εικονικών περιβαλλόντων και των τεχνολογιών που σχετίζονται με αυτά. Εισαγωγικά δίνεται μια σειρά ορισμών, που αφορούν τα εικονικά περιβάλλοντα και την εικονική πραγματικότητα γενικότερα, και γίνεται μια διάκριση μεταξύ των διαφόρων ειδών, ανάλογα με τον τύπο υλικού που χρησιμοποιούν. Στη συνέχεια γίνεται λεπτομερής επισκόπηση διαφόρων σχεδιαστικών ζητημάτων, που αφορούν στο μοντέλο του περιβάλλοντος και στους διάφορους τρόπους αλληλεπίδρασης του χρήστη με αυτό, και παρατίθενται οι σχετικές τεχνολογίες τόσο από πλευράς υλικού όσο και λογισμικού. Τέλος, παρουσιάζεται ένας αριθμός από σημαντικές εφαρμογές σε διάφορους χώρους, επιστημονικούς και μη, που αναδεικνύουν τη χρησιμότητα των εικονικών περιβαλλόντων ως μέσο αλληλεπίδρασης με το χρήστη.

Εικονική πραγματικότητα και εικονικά περιβάλλοντα

Επειδή δεν υπάρχει κάποιος συγκεκριμένος και αυστηρός ορισμός για τον όρο Εικονική Πραγματικότητα, δίνονται παρακάτω κάποιοι από τους επικρατέστερους. Ο ίδιος ο όρος βέβαια είναι αντιφατικός και οδηγεί σε παρεξηγήσεις και σε πολύωρες φιλοσοφικές συζητήσεις.

Ο πατέρας του όρου Jaron Lanier, έδωσε τον εξής ορισμό το 1989: "Ένα αλληλεπιδραστικό, τρισδιάστατο περιβάλλον, φτιαγμένο από υπολογιστή, στο οποίο μπορεί κάποιος να εμβυθιστεί."

Απο εκεί και έπειτα, δόθηκαν ποικίλοι ορισμοί, μερικοί από τους οποίους δίνονται παρακάτω:

"Η Εικονική Πραγματικότητα, αποτελεί ένα όρο που έχει γίνει πρόσφατα γνωστός αλλά και από τους πλέον διαδεδομένους στο χώρο των υπολογιστών, ο οποίος μεταφέρει το χρήστη ή τους χρήστες, σε ένα συνθετικό, τεχνητό, εικονικό και φτιαγμένο από υπολογιστή περιβάλλον." M.Krueger (1991)

"Αλληλεπιδραστικά γραφικά πραγματικού χρόνου (real-time) με τρισδιάστατα μοντέλα, συνδυασμένα με μια τεχνολογία απεικόνισης η οποία δίνει τη δυνατότητα στο χρήστη για εμβύθιση στον μοντελοποιημένο κόσμο και τη δυνατότητα για απευθείας χειρισμό." Fuchs, H., Bishop, et al. (1992)

"Η ψευδαίσθηση της συμμετοχής σε ένα συνθετικό περιβάλλον αντί για την εξωτερική παρατήρηση ενός τέτοιου περιβάλλοντος. Η Εικονική Πραγματικότητα βασίζεται σε τρισδιάστατες, στερεοσκοπικές μονάδες απεικόνισης, με ανιχνευτή της κίνησης του κεφαλιού,

του χεριού ή του σώματος και στερεοσκοπικό ήχο. Η Εικονική Πραγματικότητα είναι μια εμπειρία εμπύθισης που χρησιμοποιεί όλες τις αισθήσεις." Gigante, M. (1993)

"Η Εικονική Πραγματικότητα αναφέρεται σε αλληλεπιδραστικά, πολυ-αισθητικά, βασισμένα στη όραση, τρισδιάστατα, περιβάλλοντα εμπύθισης, δημιουργημένα από υπολογιστή, καθώς και ο συνδυασμός των τεχνολογιών που απαιτούνται για την ανάπτυξη τέτοιων περιβαλλόντων." Cruz-Neira, C. (1993)

"Μπορεί να οριστεί σαν ένας νέος τρόπος επικοινωνίας μεταξύ ανθρώπου και μηχανής. Ένα από τα χαρακτηριστικά του είναι η υιοθέτηση συσκευών απεικόνισης και αλληλεπίδρασης των ανθρώπινων αισθήσεων. Στερεοσκοπικά συστήματα απεικόνισης, δίνουν τη εντύπωση πραγματικής χωρικής αντίληψης των τρισδιάστατων εικόνων οι οποίες παράγονται από τον υπολογιστή. Επιπλέον, η αίσθηση του ότι είσαι εμπυθισμένος σε ένα εικονικό περιβάλλον, δυναμώνει με τη χρήση συσκευών όπως το γάντι (data glove), το οποίο επιτρέπει πιο φυσική και ενστικτώδη απευθείας αλληλεπίδραση." Ellis, S. R. (1994)

"Ένα υπολογιστικό σύστημα το οποίο χρησιμοποιείται για τη δημιουργία εικονικών κόσμων, στους οποίους ο χρήστης έχει την εντύπωση της ύπαρξης του σε αυτούς και επιπλέον έχει την ικανότητα να πλοηγηθεί και να χειριστεί τα αντικείμενά τους." C.Manetta, & Blade R. (1995)

"Η Εικονική Πραγματικότητα είναι τα από τον υπολογιστή φτιαγμένα, τρισδιάστατα, εξομοιωμένα περιβάλλοντα τα οποία απαντώνται σε πραγματικό χρόνο (real-time), καθώς τα διαχειρίζεται ο χρήστης." Mills, S., Noyes, J. (1999)

"Βασικά είναι μία διεπαφή, η οποία συνδυάζει διαφορετικά τεχνικά συστήματα με σκοπό να δώσει τη δυνατότητα στον χρήστη να αλληλεπιδράσει σε πραγματικό χρόνο με μία εφαρμογή για την απεικόνιση (visualization), την περιγραφή της κίνησης (animation), την παραγωγή (generation) και την μεταβολή (modification) τρισδιάστατων δεδομένων, δημιουργημένων από υπολογιστή τα οποία βλέπει στερεοσκοπικά. Ένας όρος που περικλείει τα πάντα και περιγράφει την τεχνολογία και όλο το πεδίο γενικότερα." VIEW of the future Project (2001)

"Η εξομοίωση ενός πραγματικού ή φανταστικού περιβάλλοντος, το οποίο μπορεί να το βιώσει ο χρήστης οπτικά στις τρεις διαστάσεις του πλάτους, ύψους και βάθους και το οποίο μπορεί επιπροσθέτως να παρέχει μια αλληλεπιδραστική οπτική εμπειρία με κίνηση σε πραγματικό χρόνο (real-time) με ήχο και πιθανώς και απτικές ή άλλες μορφές ανάδρασης." Whatis.com full reference (2003)

"Ένα μέσο το οποίο αποτελείται από αλληλεπιδραστικές εξομοιώσεις με υπολογιστή, οι οποίες 'αισθάνονται' την θέση και τις ενέργειες του χρήστη, και αντικαθιστούν ή επαυξάνουν την ανάδραση σε μία ή παραπάνω αισθήσεις, δίνοντας το αίσθημα της πνευματικής εμπύθισης ή παρουσίας στην εξομοίωση (ένας εικονικός κόσμος)." Sherman, W. R., Craig, A., B. (2003)

Στην επιστημονική κοινότητα αποφεύγεται η χρήση του όρου Εικονική Πραγματικότητα λόγω της αντιφατικότητάς του και χρησιμοποιείται ο όρος **Εικονικό Περιβάλλον**, Virtual Environment στα αγγλικά, (αγγλική συντομογραφία VE). Τα εικονικά περιβάλλοντα, που αναφέρονται στη βιβλιογραφία και ως συστήματα εικονικών περιβαλλόντων (virtual environment systems) ή και ως εικονικοί κόσμοι (virtual worlds), είναι σε γενικές γραμμές τα συστήματα τα οποία στο παρελθόν αναφέρονταν με το γενικότερο όρο εικονική πραγματικότητα. Δυστυχώς, μέχρι σήμερα δεν έχουν υπάρξει γενικώς αποδεκτοί ορισμοί για τους παραπάνω όρους.

Εφαρμογές εικονικών περιβαλλόντων

Τα εικονικά περιβάλλοντα έχουν εφαρμογές σε τομείς όπου απαιτείται η εξομοίωση ενός κόσμου ή μέρος του, η εκδήλωση δυναμικής και συμπεριφοράς ευφυών οντοτήτων καθώς και η αλληλεπίδραση του χρήστη με τον εικονικό κόσμο και τους κατοίκους του με τέτοιο τρόπο ώστε να δεσμεύεται το ενδιαφέρον του χρήστη. Κάποια παραδείγματα είναι οι τομείς της εκπαίδευσης και κατάρτισης, της ψυχαγωγίας με χρήση υπολογιστή, της προσομοίωσης και αξιολόγησης όπως επίσης και της επιστημονικής έρευνας.

Στο χώρο της εκπαίδευσης και της κατάρτισης, η πλειοψηφία των συστημάτων είναι χτισμένα γύρω από έναν κεντρικό εικονικό εκπαιδευτή με δυνατότητες κίνησης και αλληλεπίδρασης με το

χρήστη. Η ειδική αυτή κατηγορία εικονικών πρακτόρων αναφέρεται συνολικά ως «κινούμενοι πράκτορες-παιδαγωγοί» («animated pedagogical agents») ή «εικονικοί εκπαιδευτές» («virtual tutors»). Ανήκει στο ευρύτερο πεδίο της εικονικής διήγησης (virtual storytelling), το οποίο αφορά τη χρήση εικονικών περιβαλλόντων για την δραματοποιημένη απεικόνιση πλοκής. Σε αρκετές περιπτώσεις δίνεται η δυνατότητα συμμετοχής του παρατηρητή, δημιουργώντας τις προϋποθέσεις για αφήγηση με αναδυόμενη δομή (emergent narrative). Ως έννοια, η εικονική διήγηση έχει αποτελέσει τη βάση για το σχεδιασμό πλήθους εφαρμογών εικονικών περιβαλλόντων στους χώρους της εκπαίδευσης και της ψυχαγωγίας.

Στην εφαρμογή Design-A-Plant παρουσιάζεται ο εικονικός πράκτορας Herman the Bug. Η εφαρμογή έχει σαν στόχο την ανάπτυξη των ικανοτήτων μαθητών για επίλυση προβλημάτων, μέσω μίας διαδικασίας σύνθεσης φανταστικών φυτών τα οποία ευδοκιμούν σε συγκεκριμένες συνθήκες. Ο ρόλος του Herman the Bug είναι να μετατρέπει τη διαδικασία αυτή σε μία ευχάριστη εμπειρία πραγματοποιώντας διάφορες ενέργειες καθαρά ψυχαγωγικού χαρακτήρα, αλλά και να βοηθά τον μαθητή παρέχοντας πληροφορίες και βοήθεια.

Αντίστοιχης φιλοσοφίας, η Adele είναι ένας εικονικός πράκτορας σε εφαρμογές εκπαιδευτικού χαρακτήρα μέσω του World-Wide Web, με χαρακτηριστικά όπως μία ανθρωπόμορφη και, άρα, προσφιλή στο χρήστη παρουσίαση, δυνατότητα παρακολούθησης και σχολιασμού των ενεργειών του, όπως επίσης και υποβοήθησή του μέσω στοιχείων και επεξηγήσεων. Η Adele έχει χρησιμοποιηθεί κυρίως σε εφαρμογές ιατρικού και κλινικού περιεχομένου.

Ο Cosmo σχεδιάστηκε για να μελετήσει την ικανότητα των πρακτόρων να συνδυάσουν δυναμικά τις δυνατότητες μορφασμών, κινήσεων και ομιλίας αναφερόμενοι σε αντικείμενα του εικονικού περιβάλλοντος ενώ παρέχουν συμβουλές. Ο ρόλος ενός μαθητή είναι να συνοδεύσει ένα πακέτο μέσα στο internet. Το κατευθύνει και το καθοδηγεί μέσα από δίκτυα με συνδεδεμένους routers. Έτσι ο μαθητής εξοικειώνεται με τους κανόνες της δικτυακής δρομολόγησης, της τοπολογίας δικτύων και μαθαίνει τους κανόνες για αποφυγή συμφόρησης και τη διαδικασία της διευθυνσιοδότησης. Όλα αυτά μέσω του Cosmo επιτυγχάνονται με ένα εύκολο και διασκεδαστικό τρόπο.



Στιγμιότυπο από τον Cosmo

Με γνωστικό αντικείμενο και στόχους που κινούνται σε περισσότερο τεχνικό επίπεδο, ο

WHIZLOW είναι ένας εικονικός πράκτορας με στόχο την εξοικείωση αρχαρίων με θέματα αρχιτεκτονικής υπολογιστών και συστημάτων. Σε αντίθεση με τον Herman the Bug και την Adele, ο WHIZLOW κατοικεί σε ένα πλήρως τρισδιάστατο εικονικό κόσμο, την CPU CITY, η οποία είναι μία διαισθητική μεταφορά του εσωτερικού ενός ηλεκτρονικού υπολογιστή και, έτσι, διαθέτει κτήρια όπως «RAM», «CPU» και «hard drive». Με στόχο την επεξήγηση της λειτουργίας του υπολογιστή, ο WHIZLOW μπορεί να εκτελεί ενέργειες οι οποίες αντιστοιχούν σε μεμονωμένες εντολές ή ολοκληρωμένα προγράμματα. Η απεικόνιση των ενεργειών περιλαμβάνει μετακίνηση μεταξύ κτηρίων και αντικείμενα που αντιστοιχούν σε συγκεκριμένες εντολές, γεγονός που οδηγεί στην δέσμευση της προσοχής του παρατηρητή και στην καλύτερη κατανόηση της απεικονιζόμενης διαδικασίας.

Ένας συνθετότερος και με ευρύτερη εφαρμογή εικονικός εκπαιδευτής είναι ο Steve (Soar Training Expert for Virtual Environments). Ο ρόλος του είναι να υποστηρίζει την κατάρτιση σπουδαστών στη διεκπεραίωση διαδικασιών και τη χρήση εξοπλισμού. Ο Steve διαθέτει ένα πλήρως τρισδιάστατο εικονικό σώμα (το οποίο μπορεί να πάρει διάφορες μορφές με συνηθέστερη αυτή του συνδυασμού κεφαλιού-κορμού-χεριού), ενώ κατοικεί σε ένα τρισδιάστατο εικονικό κόσμο κατάλληλα διαμορφωμένο για την κάθε εκπαιδευτική συνεδρία. Στον ίδιο εικονικό κόσμο εμβυθίζονται και οι σπουδαστές, σε αντίθεση με τα συστήματα που παρουσιάστηκαν παραπάνω. Ο Steve έχει τη δυνατότητα να εξηγεί διαδικασίες βηματικά, να παρέχει επεξηγήσεις, να παρακολουθεί τις κινήσεις των σπουδαστών και να τους διορθώνει. Επίσης, μπορεί να καταδεικνύει με τον δείκτη του εικονικού του χεριού συγκεκριμένα σημεία ελέγχου εξοπλισμού (χειριστήρια, όργανα μετρήσεων, κ.α.) και να κατευθύνει το βλέμμα του προς τον ή τους χρήστες, γεγονός με ιδιαίτερη σημασία για την επίτευξη πιστευτότητας στην αλληλεπίδραση μεταξύ εικονικών πρακτόρων και ανθρώπων-χρηστών .

Η συμπεριφορά του Steve παράγεται με χρήση του SOAR, ενός συστήματος συμπερασματολογίας βασισμένο σε κανόνες το οποίο έχει αποτελέσει τη βάση για πολλές εφαρμογές στο χώρο της Τεχνητής Νοημοσύνης γενικά, αλλά και με στόχο ευφυείς εικονικούς πράκτορες, ειδικότερα. Ο Steve έχει ενσωματωθεί σε ένα πλήθος εφαρμογών εκπαίδευσης, όπως το Mission Rehearsal Exercise System (MRE), όπου και τονίζεται η δυσκολία στην ολοκλήρωση των διαφόρων συστατικών τα οποία είναι απαραίτητα για τη λειτουργία ενός εικονικού περιβάλλοντος . Το MRE έχει ως αντικείμενο την εκπαίδευση στρατιωτών σε θέματα χειρισμού κρίσεων, όπως την αντιμετώπιση ατυχημάτων και την προστασία τραυματιών σε εμπόλεμη κατάσταση. Επιπλέον, έχει αξιοποιηθεί για την εκπαίδευση σε τεχνικές διάλογου και διαπραγμάτευσης σε καταστάσεις πίεσης , όπως επίσης και στην ενσωμάτωση μοντέλων συναισθημάτων σε εικονικούς πράκτορες για εφαρμογές εκπαίδευσης και διαλόγου .

Τα εικονικά περιβάλλοντα βρίσκουν εφαρμογές και στο χώρο της υγείας, όπως, για παράδειγμα, η πολυμεσική παραγωγή «Carmen's Bright Ideas» η οποία αξιοποιεί έναν εικονικό πράκτορα και την ιδέα του διαδραστικού παιδαγωγικού δράματος (interactive pedagogical drama) για να εκπαιδεύσει μητέρες παιδιών με καρκίνο σε θέματα επίλυσης προβλημάτων . Οι θεραπευτικές ιδιότητες της αίσθησης της παρουσίας σε εικονικά περιβάλλοντα εμβύθισης, με στόχο την αντιμετώπιση πόνου, ιδεοληψιών και καταστάσεων μετατραυματικού άγχους, όπως επίσης και η χρήση των εικονικών περιβαλλόντων ως ερευνητικών εργαλείων στο χώρο της Νευρολογίας, έχουν επίσης διερευνηθεί . Εντούτοις, η έννοια του εικονικού πράκτορα δεν είναι πάντα σχετική με την κάθε εφαρμογή, με αποτέλεσμα αρκετές προτάσεις να βασίζονται σε τεχνολογίες από το χώρο της Εικονικής Πραγματικότητας και να χαρακτηρίζονται από ευφυή και βασισμένη σε γνώση ανταλλαγή πληροφορίας με το χρήστη, αλλά να μην μπορούν να χαρακτηριστούν ως ολοκληρωμένα εικονικά περιβάλλοντα αλλά περισσότερο ως έμπειρα συστήματα (expert systems) με προηγμένες δυνατότητες απεικόνισης και αλληλεπίδρασης, όπως, για παράδειγμα, το σύστημα «Expert Surgical Assistant» .

Ο SAM είναι ένας εικονικός πράκτορας με αντικείμενο την εκπαίδευση και την ψυχαγωγία παιδιών χωρίς επίβλεψη. Σαν προσέγγιση, έχει την ιδιαιτερότητα της δυνατότητας αλληλεπίδρασης μεταξύ του δικού του εικονικού χώρου και του φυσικού περιβάλλοντος του παιδιού, χαρακτηριστικό που περιγράφεται ως «διαμοιραζόμενη πραγματικότητα» («shared reality»). Ο εικονικός κόσμος του SAM προβάλλεται σε μία επίπεδη οθόνη, μεταξύ της οποίας και του παιδιού βρίσκεται ένα συναρμολογούμενο κάστρο. Επίσης, μπροστά από το κάστρο

υπάρχει μία επιφάνεια ευαίσθητη στην πίεση, χάρη στην οποία ο SAM μπορεί να αντιλαμβάνεται τα κομμάτια τα οποία το παιδί μεταχειρίζεται. Σε συνδυασμό με ένα σύστημα αναγνώρισης του τόνου και της έντασης της φωνής του παιδιού, το σύστημα μπορεί να δίνει οδηγίες στο παιδί, να το κατευθύνει προς την συναρμολόγηση του κάστρου, να διατηρεί το ενδιαφέρον του και να δημιουργεί την ψευδαίσθηση της μετάβασης αντικειμένων, κατά την εξέλιξη του παιχνιδιού, από τον εικονικό κόσμο στο φυσικό και αντίστροφα.

Το σύστημα FearNot είναι ένα εικονικό περιβάλλον με στόχο την αντιμετώπιση των ψυχολογικών συνεπειών που έχει σε παιδιά ο ενδοσχολικός εκφοβισμός (bullying). Στην αρχή κάθε συνεδρίας, το σύστημα παρουσιάζει στο παιδί τον χώρο ενός σχολείου. Στη συνέχεια, απεικονίζεται ένα συμβάν εκφοβισμού μεταξύ δύο παιδιών τα οποία αναπαρίστανται από εικονικούς πράκτορες. Με το πέρας του συμβάντος, το θύμα απευθύνεται στο παιδί ζητώντας συμβουλές για την αντιμετώπιση παρόμοιων περιστατικών και, μέσα από επαναλήψεις της παραπάνω διαδικασίας (στις οποίες η εξέλιξη του περιστατικού επηρεάζεται από τις επιλογές του παιδιού) το σύστημα προσπαθεί να μεταδώσει ένα συγκεκριμένο διδακτικό μήνυμα. Με βάση το σύστημα FearNot έχει διερευνηθεί και τεκμηριωθεί η σχέση μεταξύ της ικανότητας διατήρησης ενός υψηλού επιπέδου συναισθηματικής εμπλοκής των συμμετεχόντων και της αποτελεσματικότητας εφαρμογών εικονικών περιβαλλόντων στην εκπαίδευση και την ψυχαγωγία .

Η δυνατότητα των εικονικών περιβαλλόντων να υποστηρίξουν τομείς με αμιγώς ψυχαγωγικό χαρακτήρα έχει επίσης διερευνηθεί, καθώς τα παιχνίδια με υπολογιστή αποτελούν, πλέον, ένα χώρο με τεράστια εμπορική δυναμική αλλά και σημαντικά τεχνολογικά επιτεύγματα. Χαρακτηριστικότερο παράδειγμα είναι τα μεγάλης κλίμακας δικτυακά παιχνίδια (massively-multiplayer online games, MMOs) τα οποία έχουν επικρατήσει χάρη στην παράλληλη εξέλιξη του Internet και την μεγάλη διείσδυση της ευρυζωνικότητας. Τα MMOs είναι συστήματα στα οποία ο χρήστης συνδέεται κυρίως μέσω του Internet και εισάγεται σε έναν εικονικό κόσμο στον οποίο συνυπάρχει με μεγάλο αριθμό άλλων παικτών. Τα διαθέσιμα παιχνίδια σήμερα χαρακτηρίζονται από μία ποικιλία στόχων, διακρίνονται όμως όλα από κάποια κοινά χαρακτηριστικά, τα οποία τα κατατάσσουν χωρίς διαπραγμάτευση στην κατηγορία των εικονικών περιβαλλόντων: ο εικονικός κόσμος διαθέτει αντικείμενα με δυνατότητες αλληλεπίδρασης ενώ, εκτός από τους παίκτες, κατοικείται και από εικονικούς πράκτορες (non-player characters, NPCs) οι οποίοι λειτουργούν αυτόνομα έχοντας συγκεκριμένους στόχους. Ο ρόλος των NPCs ποικίλει: μπορεί να είναι απλοί χαρακτήρες χωρίς δυνατότητα αλληλεπίδρασης με τους παίκτες, χαρακτήρες με συγκεκριμένο κοινωνικό ρόλο (π.χ. φρουροί, πωλητές, οδηγοί, κ.τ.λ.) ή, ακόμα, να έχουν κρίσιμο ρόλο στην εξέλιξη της πλοκής. Ιδιαίτερα η τελευταία κατηγορία διακρίνεται, συνήθως, από προηγμένες δυνατότητες παραγωγής συμπεριφοράς, δανειζόμενη τεχνικές και προσεγγίσεις από το χώρο της Τεχνητής Νοημοσύνης. Ενδεικτικά αναφέρονται τα World of Warcraft¹ και EVE Online² ως εκπρόσωποι του χώρου των MMOs. Λόγω της εμπορικότητάς τους, τα MMOs εκμεταλλεύονται, κατά κανόνα, την τελευταία λέξη της τεχνολογίας σε θέματα γραφικών, ήχου και αλληλεπίδρασης, ενώ συχνά υποστηρίζουν εξοπλισμό εμπύθισης και διεπαφής με βάση την αφή. Η δυνατότητά τους, όμως, να προσφέρουν στην επιστημονική έρευνα περιορίζεται ακριβώς από αυτόν τον εμπορικό τους χαρακτήρα, ο οποίος συνήθως τα δεσμεύει με κλειστά πνευματικά δικαιώματα και πατέντες. Επιπλέον, ο στόχος της ταχείας παραγωγής εμπορεύσιμων αποτελεσμάτων οδηγεί, όχι σπάνια, στην υιοθέτηση ad hoc και με αμφιλεγόμενη θεωρητική ισχύ προσεγγίσεων, γεγονός που περιορίζει ακόμα περισσότερο τη δυνατότητα προσφοράς του συγκεκριμένου χώρου στην επιστημονική έρευνα. Αξίζει να σημειωθεί πως, παρ' όλα αυτά, πολλά παιχνίδια - MMOs και μη - έχουν αποτελέσει τη βάση για ερευνητικές προσπάθειες σε διάφορα επιστημονικά πεδία συμπεριλαμβανομένων και των εικονικών περιβαλλόντων .

Ο χώρος των παιχνιδιών με υπολογιστή έχει να επιδείξει πολλά πρόσθετα παραδείγματα, με μικρότερη έως ανύπαρκτη υποστήριξη για συνύπαρξη πολλών παικτών αλλά με αξιοσημείωτες δυνατότητες ευφυούς συμπεριφοράς από τη μεριά των εικονικών πρακτόρων. Για παράδειγμα, το παιχνίδι Sims¹ δίνει στο χρήστη τον έλεγχο ενός ανθρωπόμορφου εικονικού χαρακτήρα στον τρισδιάστατο εικονικό κόσμο ενός σπιτιού . Πέραν του βαθμού στον οποίο ελέγχεται από τον χρήστη, ο εικονικός πράκτορας έχει και την ικανότητα αυτόνομης συμπεριφοράς. Ο παίκτης ορίζει επιλεκτικά τους στόχους του εικονικού πράκτορα ο οποίος, με τη σειρά του, δημιουργεί πλάνα

ενεργειών με στόχο την επίτευξη των στόχων που του ανατίθενται από τον χρήστη πέραν αυτών τους οποίους έχει θέσει ο ίδιος. Οι εικονικοί πράκτορες έχουν προτιμήσεις και ανάγκες, η συμβατότητα των οποίων με τους εκάστοτε στόχους τους επηρεάζει τη συναισθηματική τους κατάσταση βάσει ενός απλού μοντέλου συναισθημάτων. Επιπλέον, αξιολογούμενοι είναι και οι δυνατότητές τους για εκδήλωση κοινωνικότητας και πολυεπίπεδων σχέσεων μεταξύ τους.

Ένα άλλο παράδειγμα παιχνιδιού με χαρακτηριστικά εικονικού περιβάλλοντος είναι το Creatures2. Το συγκεκριμένο παιχνίδι εξελίσσεται σε ένα εικονικό κόσμο ο οποίος δεν είναι τρισδιάστατος αλλά οι χαρακτήρες που κατοικούν σ' αυτόν διαθέτουν εντυπωσιακά χαρακτηριστικά εξελιξιμότητας, αναπαραγωγής, αλληλεπίδρασης και ικανοτήτων μάθησης.

Κατά τη διάρκεια της τελευταίας δεκαετίας έχει σημειωθεί μεγάλη πρόοδος ως προς τον ρεαλισμό στην απεικόνιση και την φυσικότητα στην κίνηση στον τομέα της αναπαράστασης εικονικών πρακτόρων για σκοπούς προσομοίωσης και ψυχαγωγίας, ειδικά στην περίπτωση των εικονικών ανθρώπων (virtual humans). Το ACE (agent common environment) είναι ένα περιβάλλον ορισμού και εκτέλεσης εφαρμογών και σεναρίων με έναν ή περισσότερους εικονικούς ανθρώπους οι οποίοι διαθέτουν ικανότητα αντίληψης και αλληλεπίδρασης με αντικείμενα στο περιβάλλον τους. Χαρακτηριστικό του ACE είναι το ότι επικεντρώνεται σε θέματα ρεαλιστικής απεικόνισης, κίνησης και εκτέλεσης ενεργειών, ενώ μεταφέρει την ευθύνη της παραγωγής συμπεριφοράς σε κατάλληλα τμήματα λογισμικού τα οποία εισάγονται στο σύστημα ανάλογα με τις ανάγκες της κάθε εφαρμογής. Επιπλέον, το σύστημα παρέχει ισχυρά εργαλεία ορισμού του εικονικού κόσμου, της κίνησης των εικονικών ανθρώπων και της αλληλεπίδρασής τους με αντικείμενα, λειτουργώντας, έτσι, όχι απλά ως μία βάση εκτέλεσης σεναρίων και εφαρμογών, αλλά σαν ένα ολοκληρωμένο περιβάλλον ανάπτυξης. Παρόμοιας φιλοσοφίας είναι και το σύστημα SimHuman. Το σύστημα παρέχει τα απαραίτητα μέσα για τον ορισμό του εικονικού κόσμου και των αντικειμένων που περιέχονται σε αυτόν, όπως επίσης και της εμφάνισης και της κίνησης των εικονικών πρακτόρων. Επιπλέον, σε μία προσπάθεια κάλυψης του κενού που υπάρχει στο χώρο των ολοκληρωμένων εργαλείων ανάπτυξης εφαρμογών εικονικών περιβαλλόντων, το σύστημα ενσωματώνει μοντέλα αντίληψης, εντοπισμού και χειρισμού συγκρούσεων καθώς και ρεαλιστικής μορφοποίησης της σκελετικής δομής και του ρουχισμού των εικονικών πρακτόρων, ώστε να μη δημιουργούνται οπτικές ανωμαλίες κατά την κίνηση και τη μετάβαση μεταξύ στάσεων. Το σύστημα δίνει έμφαση στον κρίσιμο παράγοντα της εκτέλεσης σε πραγματικό χρόνο, ενώ υποστηρίζει εγγενώς την ενσωμάτωση και αξιοποίησή του σε σύνθετες εφαρμογές. Μία εξαντλητική παρουσίαση των τομέων εφαρμογής των εικονικών περιβαλλόντων θα ήταν πέρα από τους στόχους της παρούσας διατριβής. Ενδεικτικά, αναφέρονται εφαρμογές - με, κατά περίπτωση, μεγαλύτερη ή μικρότερη έμφαση σε συγκεκριμένα χαρακτηριστικά και λειτουργικά στοιχεία - στην οπτικοποίηση εξειδικευμένων επιστημονικών δεδομένων, την μοντελοποίηση μη-ανθρωπόμορφων εικονικών πρακτόρων, την ιστορική αναπαράσταση, την παρουσίαση, τον προσανατολισμό σε χώρους όπως θέατρα και μουσεία, την τέχνη και την κοινωνική δικτύωση.

Αντικείμενα σε εικονικό περιβάλλον

Αναπαράσταση αντικειμένων

Όπως είπαμε ο εικονικός μας κόσμος αποτελείται από ένα σύνολο εικονικών αντικειμένων, που έχει το item ως δομή δεδομένων για την αναπαράστασή του. Στο κεφάλαιο αυτό θα ορίσουμε σε αφηρημένο σχεδιαστικό επίπεδο το item aspect ως βασικό συστατικό αναπαράστασης πληροφορίας. Για το σκοπό αυτό θα ορίσουμε τρία εξειδικευμένα item aspects, το physical aspect που αντιστοιχεί σε αναπαράσταση πληροφορίας σχετικά με την φυσική απεικόνιση του item, το semantic aspect, για την σημασιολογία και την διαθέσιμη προς αντίληψη πληροφορία και τέλος το access aspect, για την διαθέσιμη προς αλληλεπίδραση λειτουργικότητα.

Item Aspect

Είναι ανάγκη το item aspect να οριστεί σε δύο επίπεδα, ένα ενοιολογικό και ένα ως δομή δεδομένων. Σε ενοιολογικό επίπεδο το item aspect αναπαριστά μια όψη του αντικειμένου είτε αυτό αφορά στην αναπαράστασή του μέσα στον εικονικό κόσμο, στην σημασιολογία του, ή ακόμη και στην λειτουργικότητα που θέλουμε να του προσδώσουμε. Ως δομή δεδομένων περιέχει όλες τις απαραίτητες πληροφορίες για τον ορισμό και τη λειτουργία του item σε τέτοιο επίπεδο αφαίρεσης και κατάλληλη μορφή ώστε να καλύπτει τις εκάστοτε ανάγκες του εικονικού κόσμου και των αλληλεπιδράσεων μέσα σε αυτόν.

Για να ορίσουμε ένα item στο σύνολο του και να του προσδώσουμε την φυσική, σημασιολογική και λειτουργική υπόσταση του είναι ανάγκη να ορίσουμε καθένα από τα αντίστοιχα item aspects. Έτσι στην ουσία καθένα από τα item aspects παρέχει μέρος της πληροφορίας που είναι απαραίτητη για την εξυπηρέτηση των υπολογιστικών αναγκών του εικονικού περιβάλλοντος.

Physical aspect : Αναπαριστά την φυσική υπόσταση του item, την ενσωμάτωση του δηλαδή μέσα στον εικονικό κόσμο. Για να το κάνει όμως αυτό πρέπει να πληρεί κάποιες απαιτήσεις. Τα περισσότερα αντικείμενα μέσα στον εικονικό κόσμο αναπαριστούν υλικά αντικείμενα από το φυσικό κόσμο. Ως εκ τούτου, τα αφορούν τόσο οι βασικές φυσικές ιδιότητες που απορρέουν από την έννοια της ενσωμάτωσης (θέση, προσανατολισμό, κάλυψη χώρου στον εικονικό κόσμο), όσο και άλλες που αφορούν στην οπτική, γεωμετρική, ακουστική κ.τ.λ. μορφοποίηση του καθενός, όπως αυτή πραγματοποιείται από τα αντίστοιχα υποσυστήματα του εικονικού περιβάλλοντος.

Semantic aspect : Αναπαριστά την πληροφορία σχετικά με το item η οποία είναι διαθέσιμη προς αντίληψη (π.χ. εικονικούς πράκτορες). Έτσι αποτελεί το σημείο πρόσβασης στην σημασιολογία του αντικειμένου σαν τμήμα ενός ευρύτερου εικονικού κόσμου. Η όψη αυτή του item είναι που δίνει την δυνατότητα σε έναν πράκτορα να αντιλαμβάνεται τόσο τις φυσικές ιδιότητες του αντικειμένου όσο και την λειτουργικότητα που έχει αυτό, ώστε να είναι εφικτή η αλληλεπίδραση του με αυτό.

Access aspect : Αναπαριστά τη διαθέσιμη από τη μεριά του item λειτουργικότητα, επιτρέποντας την αλληλεπίδραση μεταξύ items και της όποιας πλευράς είναι ικανή για αλληλεπίδραση (π.χ. από εικονικούς πράκτορες). Είναι το μέρος της πληροφορίας για το item που δηλώνει τις ιδιότητες που μπορεί να έχει τις συγκεκριμένες ενέργειες που μπορεί να κάνει ένας πράκτορας σε αυτό.

Γλώσσα VERL

Η γλώσσα VERL (Virtual Environment Representation Language) είναι μία γλώσσα ορισμού εικονικών κόσμων σύμφωνα με την αναπαράσταση REVE για χρήση σε εικονικά περιβάλλοντα. Βασισμένη στη μεταγλώσσα XML, η VERL παρέχει συντακτικές δομές οι οποίες απεικονίζουν το σύνολο των αφηρημένων εννοιών που ορίζονται στα προηγούμενα κεφάλαια.

Οι ορισμοί σε VERL αποτελούνται από ένα ανεξάρτητο από την εφαρμογή και την υλοποίηση, όπως επίσης και ένα εξειδικευμένο ως προς αυτές, μέρος. Το ανεξάρτητο μέρος αντιστοιχεί στο επίπεδο σύνταξης XML, ενώ το εξειδικευμένο στις τιμές συγκεκριμένων ορισμάτων. Για παράδειγμα, όπως θα φανεί παρακάτω, ένα στοιχείο XML τύπου «function» ορίζει ένα function (δηλαδή, κάτι το οποίο αφορά γενικά την αναπαράσταση, ανεξαρτήτως της εφαρμογής) ενώ η τιμή του ορίσματος «class» του ίδιου στοιχείου ορίζει την υλοποιημένη κλάση του function όπως αυτή παρέχεται από συγκεκριμένη εφαρμογή. Με τον τρόπο αυτό, η VERL επιτυγχάνει να γεφυρώσει την ενοιολογική, υψηλού επιπέδου περιγραφή ενός εικονικού κόσμου, με την τεχνική, χαμηλού επιπέδου υλοποίησή του, στα πλαίσια ενός συγκεκριμένου εικονικού περιβάλλοντος, διατηρώντας ξεκάθαρη τη διάκριση μεταξύ των δύο πλευρών και μη εμπλεκοντας στη βασική της σύνταξη στοιχεία προσανατολισμένα στην εφαρμογή. Η σύνταξη της VERL ορίζεται από το παρακάτω DTD (Document Type Definition).

```
<!-- DTD for ABRML version 1.0 -->

<!ELEMENT abr ( world ) >
<!ATTLIST abr version CDATA #REQUIRED >

<!ELEMENT world ( info? , itemGroup* ) >
<!ATTLIST world version CDATA #REQUIRED >
<!ATTLIST world name CDATA #REQUIRED >
<!ATTLIST world dimensions CDATA #REQUIRED >

<!ELEMENT info EMPTY >
<!ATTLIST info organization CDATA #IMPLIED >
<!ATTLIST info url CDATA #IMPLIED >
<!ATTLIST info email CDATA #IMPLIED >
<!ATTLIST info revision CDATA #IMPLIED >
<!ATTLIST info date CDATA #IMPLIED >
<!ATTLIST info author CDATA #IMPLIED >
<!ATTLIST info description CDATA #IMPLIED >

<!ELEMENT itemGroup ( item+ ) >
<!ATTLIST itemGroup name CDATA #REQUIRED >
<!ATTLIST itemGroup location CDATA #REQUIRED >
<!ATTLIST itemGroup behaviours CDATA "true" >

<!ELEMENT item ( virtualModel, semanticModel?, accessModel? ) >
<!ATTLIST item name CDATA #IMPLIED >
<!ATTLIST item class CDATA "real.general" >
<!ATTLIST item fit CDATA "true" >
<!ATTLIST item fitCentre CDATA "false" >

<!ELEMENT virtualModel ( transform? ) >
<!ATTLIST virtualModel source CDATA #REQUIRED >

<!ELEMENT transform EMPTY >
<!ATTLIST transform translation CDATA "0.0, 0.0, 0.0" >
<!ATTLIST transform rotation CDATA "0.0, 1.0, 0.0, 0.0" >
<!ATTLIST transform scale CDATA "1.0, 1.0, 1.0" >

<!ELEMENT semanticModel ( symbol* ) >

<!ELEMENT symbol ( argument* ) >
<!ATTLIST symbol name CDATA #REQUIRED >
```



```
<!ELEMENT argument EMPTY >
<!ATTLIST argument class CDATA #REQUIRED >
<!ATTLIST argument args CDATA #IMPLIED >

<!ELEMENT accessModel ( accessPoint* ) >

<!ELEMENT accessPoint ( function* ) >
<!ATTLIST accessPoint name CDATA #REQUIRED >
<!ATTLIST accessPoint node CDATA #REQUIRED >
<!ATTLIST accessPoint visible CDATA "true" >

<!ELEMENT function EMPTY >
<!ATTLIST function name CDATA #REQUIRED >
<!ATTLIST function class CDATA #REQUIRED >
<!ATTLIST function args CDATA "" >
```

Η ελάχιστη μορφή την οποία μπορεί να έχει ένας ορισμός εικονικού κόσμου σε VERL, μία μορφή η οποία δεν ορίζει κανένα ουσιαστικό περιεχόμενο παρά μόνο πληροφορίες όπως τίτλος, περιγραφή, έκδοση, στοιχεία δημιουργού, πνευματική ιδιοκτησία, κ.τ.λ., φαίνεται παρακάτω.

```
<!DOCTYPE reve SYSTEM "url/to/verl-1.x.dtd">

<reve version="1.0">
  <world version="1.1" name="MidWorld" dimensions="100.0, 261.8">
    <info author="George Anastassakis"
      date="20/1/2007"
      revision="1"
      email="anastas@unipi.gr"
      organization="Knowledge Engineering Lab, Department of Informatics,
        University of Piraeus"
      description="A sample REVE world."
      url="http://kelnet.cs.unipi.gr/reve"
    />

    <!-- item definitions -->

  </world>
</reve>
```

Στη γλώσσα VERL, ένα item ορίζεται από ένα στοιχείο «item» το οποίο βρίσκεται μέσα σε ένα στοιχείο «itemGroup», όπως παρακάτω. Όπως φαίνεται, για κάθε στοιχείο «itemGroup» ορίζεται

ένα μοναδικό ανά ορισμό VERL όνομα μέσω του ορίσματος «name». Επιπλέον, το όρισμα «location» δέχεται μία τιμή τύπου URL που προσδιορίζει την πηγή από την οποία αντλούνται δεδομένα φυσικής αναπαράστασης, σε δενδροειδή δόμηση, για τα physical aspects των items του itemGroup (π.χ., αρχείο, δικτυακός πόρος, κ.τ.λ.).

```
<itemGroup name="itemGroupName" location="location" behaviours="true">

  <item name="itemName" class="itemClass" fit="isFit"
    fitCentre="isCentreFit">
    <!-- item aspect definitions -->
  </item>
</itemGroup>
```

Στον ίδιο ορισμό VERL, είναι επιτρεπτό να ορίζουν το ίδιο location περισσότερα του ενός itemGroups. Τέλος, το όρισμα «behaviours» καθορίζει το κατά πόσο θα συμπεριληφθούν δυναμικά στοιχεία στα physical aspects των items του itemGroup, δηλαδή animation, scripts, συμπεριφορά, κ.τ.λ., σε περίπτωση όπου είναι διαθέσιμα μέσω του location. Τα αποσιωπητικά πριν και μετά το στοιχείο itemGroup υποδηλώνουν τη δυνατότητα ορισμού πολλαπλών itemGroups για τον ίδιο ορισμό VERL (είναι προφανές ότι πρέπει να οριστεί τουλάχιστον ένα itemGroup, με κατάλληλο location, για κάθε ξεχωριστή πηγή δεδομένων φυσικής αναπαράστασης).

Εξ' άλλου, για κάθε item ορίζεται, επίσης, ένα μοναδικό ανά ορισμό VERL όνομα. Επιπλέον, το όρισμα «class» δέχεται μία τιμή τύπου string η οποία αντιστοιχεί στο item class του item. Τέλος, τα όρισμα «fit» και «fitCentre» δέχονται τιμές τύπου boolean και καθοδηγούν τη διαδικασία προσαρμογής της φυσικής αναπαράστασης του item. Συγκεκριμένα, αν το fit είναι true, το item προσαρμόζεται. Το αποτέλεσμα της προσαρμογής εξαρτάται από το fitCentre, το οποίο, αν είναι false, προκαλεί προσαρμογή του item ως προς την αρχή του συστήματος συντεταγμένων του, ενώ αν, αντίθετα, είναι true, προκαλεί προσαρμογή του item ως προς το κέντρο της βάσης του περικλείοντα σχήματός του. Τα αποσιωπητικά πριν και μετά το στοιχείο item υποδηλώνουν τη δυνατότητα ορισμού πολλαπλών items για το ίδιο itemGroup.

Στο τμήμα του στοιχείου item που σημειώνεται ως «<!-- item aspect definitions -->», θα οριστούν τα item aspects του αντίστοιχου item. Στη γλώσσα VERL, τα τρία βασικά item aspects ορίζονται με τρία, αντιστοίχως, στοιχεία μέσα στο στοιχείο item, συγκεκριμένα, τα «physicalAspect», «semanticAspect» και «accessAspect», όπως παρακάτω.

```
<item name="itemName" class="itemClass" fit="isFit" fitCentre="isCentreFit">
<physicalAspect ...>
<!-- initial transformation definition --> </physicalAspect>
<semanticAspect ...>
<!-- symbol and argument definitions --> </semanticAspect>
<accessAspect ...>
<!-- access point and function definitions --> </accessAspect>
</item>
```

Στη γλώσσα VERL, ένα physical aspect ορίζεται από ένα στοιχείο «physicalAspect».

```
<physicalAspect source="physicalAspectSource">
  <transform
    translation="initialTranslation"
    rotation="initialRotation"
    scale="initialScale"
  />
</physicalAspect>
```

Όπως φαίνεται παραπάνω, για κάθε στοιχείο «physicalAspect» ορίζεται το όρισμα «source», η τιμή physicalAspectSource του οποίου αντιστοιχεί στον κόμβο ο οποίος διατίθεται μέσω του resource του γονικού itemGroup και θα χρησιμοποιηθεί ως κόμβος-ρίζα του physical aspect. Επιπλέον, για κάθε στοιχείο «physicalAspect» ορίζεται, προαιρετικά, ένα στοιχείο «transform», τα ορίσματα «translation», «rotation» και «scale» του οποίου καθορίζουν τις αρχικές τιμές θέσης, προσανατολισμού και κλιμάκωσης της φυσικής αναπαράστασης του item, αντίστοιχα. Εξ' άλλου, ένα semantic aspect ορίζεται από ένα στοιχείο «semanticAspect», όπως παρακάτω.

```
<semanticAspect>
  <symbol name="symbolName">
    <argument class="argumentClass" args="argumentCreationArgs" /> </symbol>
</semanticAspect>
```

Όπως φαίνεται παραπάνω, για κάθε στοιχείο «semanticAspect» δηλώνεται ένας αριθμός εμφωλευμένων στοιχείων «symbol» τα οποία ορίζουν τα σύμβολα του αντίστοιχου semantic aspect. Για κάθε στοιχείο «symbol» ορίζεται ένα μοναδικό ανά semantic aspect όνομα μέσω του ορίσματος «name». Τα αποσιωπητικά πριν και μετά το στοιχείο «symbol» υποδηλώνουν τη δυνατότητα ορισμού πολλαπλών συμβόλων για το ίδιο semantic aspect.

Επιπλέον, για κάθε στοιχείο «symbol» δηλώνεται ένας αριθμός εμφωλευμένων στοιχείων «argument» τα οποία ορίζουν τα ορίσματα του αντίστοιχου συμβόλου. Για κάθε στοιχείο «argument» δηλώνονται δύο ορίσματα, τα «class» και «args». Η τιμή argumentClass του ορίσματος «class» ορίζει την κλάση του αντικειμένου που θα αναπαραστήσει το αντίστοιχο σύμβολο, ενώ η τιμή argumentCreationArgs του ορίσματος «args» ορίζει τα ορίσματα αρχικοποίησης του αντικειμένου. Χάρη στο μηχανισμό αυτό, οι διαθέσιμες κλάσεις ορισμάτων συμβόλων δεν είναι προκαθορισμένες για το εικονικό περιβάλλον· αντίθετα, δηλώνονται δυναμικά για τον κάθε εικονικό κόσμο μέσω του VERL ορισμού του, καθιστώντας, ουσιαστικά, εφικτή την επέκταση των δυνατοτήτων αναπαράστασης σημασιολογίας items από τον ίδιο το χρήστη. Τα αποσιωπητικά πριν και μετά το στοιχείο «argument» υποδηλώνουν τη δυνατότητα ορισμού πολλαπλών ορισμάτων για το ίδιο σύμβολο.

Τέλος, ένα access aspect ορίζεται από ένα στοιχείο «accessAspect», όπως παρακάτω.

```
<accessAspect>
  <accessPoint name="accessPointName" node="accessPointNode" visible="isVisible">
    <function name="funcName" class="funcClass" args="funcCreateArgs" />
  </accessPoint>
</accessAspect>
```

Όπως φαίνεται παραπάνω, για κάθε στοιχείο «accessAspect» δηλώνεται ένας αριθμός εμφωλευμένων στοιχείων «accessPoint» τα οποία ορίζουν τα accesspoints του αντίστοιχου access aspect. Για κάθε στοιχείο «accessPoint» ορίζεται ένα μοναδικό ανά access aspect όνομα μέσω του ορίσματος «name». Ακόμα, ορίζεται ο κόμβος του physical aspect του ίδιου item με τον οποίο σχετίζεται το accesspoint, μέσω του ορίσματος «node». Το όρισμα «visible» αξιοποιείται σε επίπεδο εφαρμογής, καθορίζοντας το εάν είναι επιθυμητό να απεικονίζονται οπτικοί σχολιασμοί για το συγκεκριμένο accesspoint. Τα αποσιωπητικά πριν και μετά το στοιχείο «accessPoint» υποδηλώνουν τη δυνατότητα ορισμού πολλαπλών access points για το ίδιο access aspect.

Επιπλέον, για κάθε στοιχείο «accessPoint» δηλώνεται ένας αριθμός εμφωλευμένων στοιχείων «function» τα οποία ορίζουν τα functions του αντίστοιχου accesspoint. Για κάθε στοιχείο «function» δηλώνονται τρία ορίσματα, τα «name», «class» και «args». Η τιμή funcName του ορίσματος «name» ορίζει ένα μοναδικό ανά accesspoint όνομα για το function. Η τιμή «funcClass» ορίζει την κλάση του αντικειμένου που θα αναπαραστήσει το function, ενώ η τιμή funcCreateArgs του ορίσματος «args» ορίζει τα ορίσματα αρχικοποίησης του αντικειμένου. Χάρη στο μηχανισμό αυτό, οι διαθέσιμες κλάσεις functions δεν είναι προκαθορισμένες για το εικονικό περιβάλλον· αντίθετα, δηλώνονται δυναμικά για τον κάθε εικονικό κόσμο μέσω του VERL ορισμού του, καθιστώντας, ουσιαστικά, εφικτή την επέκταση των δυνατοτήτων αλληλεπίδρασης με items από τον ίδιο το χρήστη. Τα αποσιωπητικά πριν και μετά το στοιχείο «function» υποδηλώνουν τη δυνατότητα ορισμού πολλαπλών functions για το ίδιο accesspoint.

Αναπαράσταση εικονικού πράκτορα

Εισαγωγή

Σε αντιστοιχία με τη βασισμένη στην έννοια του item προσέγγιση αναπαράστασης αντικειμένων σε εικονικούς κόσμους, στο κεφάλαιο αυτό ορίζεται το εικονικό σώμα, σε αφηρημένο σχεδιαστικό επίπεδο, ως ειδικής κατηγορίας item. Με στόχο την κάλυψη των ιδιαίτερων αναγκών της κατηγορίας αυτής, ορίζονται δύο εξειδικευμένα item aspects: το perception aspect, για την αναπαράσταση δυνατοτήτων του εικονικού σώματος αντίληψης του περιβάλλοντός του, και το activity aspect, για την αναπαράσταση των δυνατοτήτων αλληλεπίδρασης του εικονικού σώματος με εικονικά αντικείμενα και, κατ' επέκταση, τον εικονικό κόσμο στο σύνολό του.

Ο εικονικός πράκτορας ως item

Ένας εικονικός πράκτορας μπορεί να θεωρηθεί, στα πλαίσια ενός τυπικού εικονικού περιβάλλοντος, ως ένα υποσύστημα που αποτελείται από (α) ένα συστατικό παραγωγής συμπεριφοράς και (β) ένα εικονικό σώμα. Το συστατικό παραγωγής συμπεριφοράς δεν αφορά την προτεινόμενη αναπαράσταση. Αντίθετα, ο σχεδιασμός, η υλοποίηση και η λειτουργία του εικονικού σώματος, από τη σκοπιά της αλληλεπίδρασής του τόσο με τον εικονικό κόσμο όσο και με τα υπόλοιπα συστατικά του εικονικού πράκτορα, αποτελούν βασικό της μέλημα: το εικονικό σώμα είναι το μόνο μέσο χάρη στο οποίο ο εικονικός πράκτορας, ως ολοκληρωμένη οντότητα με

δυνατότητες συμπεριφοράς, μπορεί να αποκτήσει γνώση σχετικά με το περιβάλλον του και να το επηρεάσει.

Ανεξαρτήτως του σχεδιασμού του, το συστατικό παραγωγής συμπεριφοράς του εικονικού πράκτορα έχει ως κύριο ρόλο την επεξεργασία γνώσης σχετικά με τον ίδιο και το περιβάλλον του, και την παραγωγή ενεργειών και πλάνων, κατά κανόνα βάσει προκαθορισμένων ή δυναμικών προτιμήσεων, επιθυμιών, επιδιώξεων, στόχων, συναισθημάτων, αναγκών και κινήτρων. Ως μέρος ενός υπολογιστικού συστήματος, το συστατικό παραγωγής συμπεριφοράς ανταλλάσσει γνώση με το υπόλοιπο σύστημα υπό τη μορφή δεδομένων. Τα δεδομένα αυτά προέρχονται από και κατευθύνονται προς το εικονικό σώμα, καθώς αυτό, όπως προαναφέρθηκε, αποτελεί το μόνο μέσο αλληλεπίδρασης του εικονικού πράκτορα με τον εικονικό κόσμο. Έτσι, προκειμένου το εικονικό σώμα να μπορεί να αποτελεί μέρος του εικονικού κόσμου κατά τα πρότυπα της προτεινόμενης αναπαράστασης - δηλαδή, ως item - είναι αναγκαίος ο ορισμός των όποιων πρόσθετων χαρακτηριστικών ως κατάλληλα εξειδικευμένα item aspects. Τα πρόσθετα αυτά χαρακτηριστικά πηγάζουν από (α) το γεγονός της συνεργασίας του εικονικού σώματος με το συστατικό παραγωγής συμπεριφοράς και (β) τις δυνατότητες του για αντίληψη και δράση. Τελικά, το εικονικό σώμα ορίζεται ως μία ειδική κατηγορία item με δύο πρόσθετα item aspects, πέραν των physical, semantic και access aspects τα οποία ισχύουν για όλα τα items.

Perception aspect: Ορίζει δυνατότητες αντίληψης του εικονικού κόσμου. Διαχειρίζεται τους αισθητήρες του εικονικού πράκτορα, λειτουργώντας ως μεσολαβητής μεταξύ του εικονικού κόσμου - ο οποίος γίνεται αντιληπτός υπό τη μορφή δεδομένων αντίληψης - και των μηχανισμών παραγωγής συμπεριφοράς του. Επίσης, εκτελεί χαμηλού επιπέδου διαδικασίες παραγωγής γνώσης σχετικά με τον εικονικό κόσμο βασιζόμενο σε άμεσα διαθέσιμα ή προαποθηκευμένα δεδομένα αντίληψης. Καλύπτει τις απαιτήσεις δυνατοτήτων εικονικών πρακτόρων για αντίληψη.

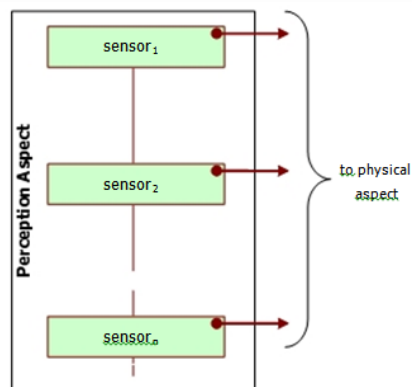
Activity aspect: Ορίζει δυνατότητες δράσης. Διαχειρίζεται τους επιδραστές του εικονικού πράκτορα, εφοδιάζοντας τους μηχανισμούς παραγωγής συμπεριφοράς του με τη δυνατότητα εκτέλεσης απλών ή σύνθετων ενεργειών. Επίσης, παρέχει και μηχανισμούς διαχείρισης και παρακολούθησης της διαδικασίας εκτέλεσης ενεργειών. Καλύπτει τις απαιτήσεις δυνατοτήτων εικονικών πρακτόρων για δράση.

Στο σημείο αυτό πρέπει να σημειωθεί ότι η ύπαρξη αισθητήρων και επιδραστών δεν είναι υποχρεωτική προκειμένου ένα item να χαρακτηρίζεται ως εικονικό σώμα και να αποτελεί μέρος ενός ολοκληρωμένου εικονικού πράκτορα. Από τη μία πλευρά, ένα εικονικό σώμα θα μπορούσε, για παράδειγμα, να μη διαθέτει καμία δυνατότητα αντίληψης του εικονικού κόσμου, εκτελώντας προκαθορισμένα, μη-προσαρμοσίμα πρότυπα συμπεριφοράς. Από την άλλη, θα μπορούσε να μη διαθέτει καμία δυνατότητα δράσης, παραμένοντας ολοκληρωτικά στατικό μέσα στο περιβάλλον, έχοντας απλά δυνατότητες παρατήρησής του. Σε μία ακραία περίπτωση, ένα item θα μπορούσε να μην διαθέτει ούτε αισθητήρες ούτε επιδραστές και, παρ' όλα αυτά, να ενέχει ρόλο εικονικού σώματος. Η διάκριση, σε κάθε περίπτωση, γίνεται με βάση το αν το item διαθέτει perception και activity aspect, η ταυτόχρονη ύπαρξη των οποίων είναι κατά την προτεινόμενη αναπαράσταση, αναγκαία και ικανή συνθήκη προκειμένου ένα item να είναι εικονικό σώμα, ενώ είναι καθ' όλα επιτρεπτό για κάποιον από τα δύο (ή και για τα δύο) να μην ορίζονται αισθητήρες και επιδραστές, αντίστοιχα.

Perception aspect

Το perception aspect ορίζει τις δυνατότητες αντίληψης ενός εικονικού πράκτορα. Συγκεκριμένα, περιγράφει το πώς ο εικονικός πράκτορας λαμβάνει πληροφορία σχετικά με το περιβάλλον του, πώς η πληροφορία αυτή υφίσταται χαμηλού επιπέδου επεξεργασία με στόχο την παραγωγή πρόσθετης γνώσης για τον εικονικό κόσμο, και πώς, τελικά, καθίσταται διαθέσιμη στα υπόλοιπα συστατικά του. Ως δομή δεδομένων, το perception aspect αποτελεί ένα σημείο πρόσβασης στο σύνολο των διαθέσιμων sensors ενός εικονικού σώματος. Το sensor είναι μία

σχεδιαστική προσέγγιση για εικονικούς αισθητήρες. Το perception aspect, ως σύνολο sensors, αντιπροσωπεύει το σύνολο των δυνατοτήτων αντίληψης ενός εικονικού πράκτορα.



Perception Aspect

Σύμφωνα με τις απαιτήσεις, κάθε sensor πρέπει, κατ' αρχήν, να μπορεί να συσχετιστεί με συγκεκριμένο και καλά ορισμένο τμήμα της φυσικής αναπαράστασης του εικονικού σώματος. Έτσι, σε πλήρη αναλογία με ό,τι ισχύει για τη σχέση μεταξύ accesspoint και φυσικής αναπαράστασης του item, ένα sensor μπορεί να έχει τη μορφή σημείου, επιφάνειας, όγκου, κ.τ.λ., ενώ ο συσχετισμός επιτυγχάνεται με τη δυνατότητα αναφοράς του sensor, ως δομής δεδομένων, σε συγκεκριμένο κόμβο του physical aspect του ίδιου item. Έτσι, το sensor αποκτά, και αυτό, υπόσταση στο χώρο, ανάλογα με τα γεωμετρικά χαρακτηριστικά του κόμβου του physical aspect στον οποίο αναφέρεται καθώς και τη σχετική θέση και τον προσανατολισμό του στο σύστημα συντεταγμένων του εικονικού σώματος. Η ιδιότητα αυτή του sensor, σε συνδυασμό με τη δυνατότητα ορισμού σχετικής μετατόπισης και περιστροφής του ως προς τον κόμβο του physical aspect με τον οποίο σχετίζεται, επιτρέπουν, όπως θα φανεί στη συνέχεια, τον ορισμό και την κατά τη λειτουργία του εικονικού περιβάλλοντος επιβολή περιορισμών στις διαδικασίες αντίληψης, οι οποίοι εξαρτώνται από την απόσταση των εμπλεκόμενων πλευρών, τον σχετικό τους προσανατολισμό, καθώς και άλλες παραμέτρους σχετικές με την ανά πάσα στιγμή διάταξή τους στο χώρο.

Σύμφωνα με τα παραπάνω, το πληροφοριακό περιεχόμενο του access aspect είναι δομημένο όπως φαίνεται στο σχήμα παραπάνω. Όπως απεικονίζεται στο σχήμα, η δομή αυτή συνίσταται σε ένα αριθμό κόμβων οι οποίοι αναπαριστούν sensors, κάθε ένας εκ των οποίων σχετίζεται με κάποιο κόμβο του physical aspect.

Οι απαιτήσεις από το sensor ως υλοποίηση αισθητήρων εικονικών πρακτόρων διαφοροποιούνται ανάλογα με το είδος του αισθητήριου συστήματος που αναπαρίσταται, καθώς και τις ανάγκες που εξ' ορισμού δημιουργεί, σε τεχνικό επίπεδο, η αξιοποίησή του σε σύνθετα και απαιτητικά υπολογιστικά συστήματα όπως είναι τα εικονικά περιβάλλοντα. Η λειτουργία του περιλαμβάνει διαδικασίες λήψης δεδομένων αντίληψης από το εικονικό περιβάλλον, επεξεργασίας των δεδομένων αυτών με στόχο την παραγωγή γνώσης για το περιβάλλον του εικονικού πράκτορα, και, τέλος, αποθήκευσης της γνώσης αυτής και μετέπειτα ασύγχρονης παροχής της στα υπόλοιπα συστατικά του.

Λήψη δεδομένων αντίληψης

Μία από τις βασικές λειτουργίες του sensor είναι η λήψη πληροφορίας υπό τη μορφή δεδομένων αντίληψης. Η λειτουργία αυτή αποτελεί το πρώτο στάδιο κάθε ολοκληρωμένης διαδικασίας αντίληψης.

Σύμφωνα με την προτεινόμενη αναπαράσταση, ένας εικονικός κόσμος αποτελείται από items και μόνο. Επίσης, ο μοναδικός τρόπος να λάβει η όποια ενδιαφερόμενη πλευρά πληροφορία σχετικά με κάποιο item είναι μέσω του semantic aspect του τελευταίου. Έτσι, η διαδικασία λήψης δεδομένων αντίληψης, όπως πραγματοποιείται από ένα sensor, συνίσταται στην

πρόσβαση στο semantic aspect ενός αριθμού items και στην επεξεργασία της πληροφορίας που παρέχουν. Για το σκοπό αυτό, το sensor έχει την ικανότητα συνεργάζεται με το semantic aspect και να το χειρίζεται συντακτικά. Τα items των οποίων το semantic aspect το sensor θα προσπελάσει επιλέγονται με βάση συγκεκριμένους εριορισμούς αντίληψης, όπως επίσης και ανάλογα με τη θέση και τον προσανατολισμό του sensor όπως αυτή επηρεάζεται τόσο από δικά του χαρακτηριστικά όσο και από παραμέτρους που αφορούν το εικονικό σώμα.

Επεξεργασία δεδομένων

Αμέσως μετά την ολοκλήρωση της λειτουργίας λήψης δεδομένων αντίληψης, το sensor προχωράει στην πραγματοποίηση μίας εξ' ίσου σημαντικής λειτουργίας: την επεξεργασία των δεδομένων αυτών με στόχο την παραγωγή του συνόλου της γνώσης σχετικά με τον εικονικό κόσμο, όπως αυτή θα είναι τελικά διαθέσιμη στα υπόλοιπα συστατικά του εικονικού πράκτορα.

Για το σκοπό αυτό, είναι απαραίτητο το sensor να έχει την ικανότητα να χειρίζεται το semantic aspect, ως δομή δεδομένων, τόσο συντακτικά - όπως είναι απαραίτητο, άλλωστε, και για τη διαδικασία λήψης δεδομένων αντίληψης - όσο και σημασιολογικά. Συγκεκριμένα, κάθε sensor έχει τουλάχιστον τη δυνατότητα να αναγνωρίζει και να χειρίζεται δεδομένα που αφορούν χαρακτηριστικά του item (item class, θέση, προσανατολισμός, διαστάσεις περικλείοντος σχήματος). Από εκεί και πέρα, ανάλογα με την υλοποίησή του και την καταγεγραμμένη σε αυτή γνώση πάνω στη σημασιολογία ενός εύρους item classes, το sensor μπορεί να εκτελεί εξειδικευμένες διαδικασίες επεξεργασίας δεδομένων αντίληψης για κάποιο item αξιοποιώντας τους διαθέσιμους γι' αυτό σχολιασμούς.

Αποθήκευση γνώσης

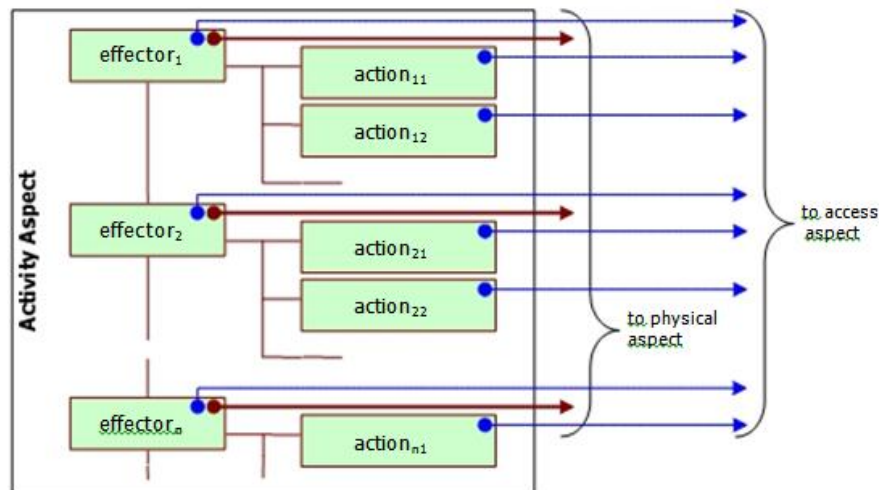
Είναι ιδιαίτερα σημαντικό να τονιστεί πως η γνώση που παράγεται από ένα sensor σαν αποτέλεσμα επεξεργασίας δεδομένων αντίληψης, αναπαρίσταται εσωτερικά και αποθηκεύεται σε συμφωνία με την ίδια γραμματική που ισχύει και για τα δεδομένα που αποθηκεύονται όπως λαμβάνονται από το εικονικό περιβάλλον χωρίς να υφίστανται κάποια επεξεργασία. Έτσι, σε συντακτικό επίπεδο, τα αποτελέσματα της λειτουργίας του sensor είναι διαθέσιμα στα υπόλοιπα συστατικά του εικονικού πράκτορα με ομοιομορφία και διαφάνεια, χωρίς να απαιτούν εξειδικευμένο χειρισμό, ενώ είναι διαχωρίσιμα σε σημασιολογικό και μόνο επίπεδο.

Με βάση το παραπάνω, η αναπαράσταση εισάγει ένα βασικό τύπο δεδομένων για αναπαράσταση γνώσης εσωτερικά στο sensor και πρόσβαση από άλλες πλευρές σε αποθηκευμένη γνώση. Ο τύπος αυτός ονομάζεται percept, ενώ, σαν δομή δεδομένων, αναπαριστά είτε (α) ένα και μοναδικό χαρακτηριστικό ή σχολιασμό του item, όπως ακριβώς το sensor το έλαβε από το εικονικό περιβάλλον, ή (β) το αποτέλεσμα ενός βήματος επεξεργασίας δεδομένων αντίληψης. Για τη φιλοξενία του πληροφοριακού του περιεχομένου, το percept φέρει ένα και μόνο πεδίο με την ονομασία value, του οποίου ο συμπαγής τύπος - άρα και ο μετέπειτα σημασιολογικός χειρισμός - εξαρτάται αποκλειστικά από την υλοποίηση. Τελικά, η συνολική γνώση που συνιστά το αποτέλεσμα μίας ολοκληρωμένης διαδικασίας αντίληψης, αποθηκεύεται εσωτερικά στο sensor όσο και διατίθεται στην όποια ενδιαφερόμενη πλευρά υπό τη μορφή ενός συνόλου percepts.

Activity aspect

Το activity aspect περιγράφει τις δυνατότητες ενός εικονικού πράκτορα για αλληλεπίδραση. Συγκεκριμένα, περιγράφει πως οι δυνατότητες αυτές μπορούν να χρησιμοποιηθούν για να αξιοποιήσουν διαθέσιμη λειτουργικότητα από τη μεριά items, πως η λειτουργικότητα αυτή δεσμεύεται προσωρινά με στόχο την αποκλειστική χρήση από συγκεκριμένο εικονικό πράκτορα, και πως η διαδικασία της αλληλεπίδρασης αυτή καθ' αυτή επικυρώνεται, εκκινείται, παρακολουθείται και ολοκληρώνεται.

Σε πλήρη αντιστοιχία με το perception aspect, το activity aspect, ως δομή δεδομένων, αποτελεί ένα σημείο πρόσβασης στο σύνολο των διαθέσιμων effectors ενός εικονικού πράκτορα. Το effector είναι μία σχεδιαστική προσέγγιση για εικονικούς επιδραστής. Το activity aspect, ως σύνολο effectors, αντιπροσωπεύει το σύνολο των δυνατοτήτων ενός εικονικού πράκτορα για αλληλεπίδραση με το περιβάλλον του.



Activity Aspect

Όπως ισχύει και για το sensor, οι απαιτήσεις από το effector διαφοροποιούνται ανάλογα με το είδος του επιδραστή που αναπαρίσταται σε κάθε περίπτωση, καθώς και τις ανάγκες σε τεχνικό επίπεδο που εξ' ορισμού δημιουργεί η αξιοποίηση του σε σύνθετα υπολογιστικά συστήματα όπως είναι τα εικονικά περιβάλλοντα. Η λειτουργία του περιλαμβάνει διαδικασίες εντοπισμού διαθέσιμης από τα items του εικονικού περιβάλλοντος λειτουργικότητας, δέσμευσης και αξιοποίησης αυτής της λειτουργικότητας με στόχο την επιρροή στον εικονικό κόσμο με συγκεκριμένο - γνωστό στον εικονικό πράκτορα εκ των προτέρων ή ακόμα και άγνωστο - τρόπο, και, ταυτόχρονα, παρακολούθησης των αλληλεπιδράσεων σε σημασιολογικό επίπεδο και συγκεκριμένα ως προς την κατάσταση, τη διάρκεια και την εφικτότητά τους.

Agent Action

Ένα effector, όπως ορίστηκε παραπάνω ως αναπαράσταση κάποιου εικονικού επιδραστή, έχει τη δυνατότητα να εκτελέσει συγκεκριμένα είδη ενεργειών ανάλογα με το τι ακριβώς αναπαριστά. Έτσι, ένα effector που αναπαριστά ένα ανθρώπινο χέρι έχει - αφαιρετικά - τη δυνατότητα να πιέσει πλήκτρα, να συλλάβει αντικείμενα, να τα μετακινήσει, να τα αφήσει ελεύθερα, κ.τ.λ. Για την αναπαράσταση των δυνατοτήτων αυτών, η αναπαράσταση εισάγει την έννοια του action.

Διαισθητικά, ένα action αντιστοιχεί σε κάποια συγκεκριμένη, μοναδική δυνατότητα του effector και έτσι, ως δομή δεδομένων, διαθέτει τα απαραίτητα χαρακτηριστικά που το καθιστούν σαφώς και μοναδικά ορισμένο στα πλαίσια ενός συγκεκριμένου effector. Η συμπαγής υλοποίηση των χαρακτηριστικών αυτών εξαρτάται από τις τεχνικές λεπτομέρειες της συνολικής υλοποίησης του εικονικού περιβάλλοντος. Μία συνιστάμενη - αλλά όχι απαραίτητη - προσέγγιση είναι να χαρακτηρίζεται το κάθε action από ένα μοναδικό ανά effector action name, οι τιμές του οποίου είναι αλφαριθμητικά αυθαίρετου μήκους.

Ένα action μπορεί να εκτελεστεί. Η εκτέλεσή του μπορεί να είναι στιγμιαία ή να έχει χρονική διάρκεια, καθώς επίσης και να πετύχει ή να αποτύχει. Επιπρόσθετα, μπορεί να εξαρτάται από διάφορες παραμέτρους. Για παράδειγμα, η εκτέλεση ενός action που αντιστοιχεί σε εφαρμογή πίεσης μπορεί να παραμετροποιείται ως προς τη ασκούμενη δύναμη και τη γωνία εφαρμογής. Για το λόγο αυτό, το action διαθέτει και τα αναγκαία - για την παραμετροποίηση που απαιτεί σε επίπεδο εκτέλεσης - χαρακτηριστικά. Επιπλέον, το κατά πόσο η εκτέλεση ενός action είναι

εφικτή ή όχι εξαρτάται από διάφορους περιορισμούς δυνατοτήτων αλληλεπίδρασης, όπως επίσης και ανάλογα με τη θέση και τον προσανατολισμό του effector όπως αυτή επηρεάζεται τόσο από δικά του χαρακτηριστικά όσο και από παραμέτρους που αφορούν το εικονικό σώμα

Ίσως η σημαντικότερη ιδιότητα του action είναι το ότι δεν περιγράφει συγκεκριμένα αποτελέσματα επί των items στα οποία εκτελείται. Αντίθετα, όπως θα φανεί αναλυτικά παρακάτω, περιγράφει συγκεκριμένους τρόπους αλληλεπίδρασης με items βάσει της διαθέσιμης λειτουργικότητάς των τελευταίων, με την όποια επίδραση σε αυτά να αναδύεται ως αποτέλεσμα των εκτελούμενων αλληλεπιδράσεων. Συγκεκριμένα, η εκτέλεση ενός action δεν συνίσταται στην απ' ευθείας μεταβολή του μοντέλου ενός item αλλά στην εκτέλεση - σε σειρά, παράλληλα ή με οποιαδήποτε άλλη λογική - κατάλληλα επιλεγμένων functions διαθέσιμων μέσω του access aspect του item, καθώς το τελευταίο είναι, σύμφωνα με την προτεινόμενη αναπαράσταση, ο μόνος τρόπος πρόσβασης σε διαθέσιμη από το item λειτουργικότητα. Για το σκοπό αυτό, το effector έχει την ικανότητα να χειρίζεται συντακτικά το access aspect ως δομή δεδομένων. Η προσέγγιση αυτή δημιουργεί πλεονεκτήματα σε πολλαπλά επίπεδα, ενώ παράλληλα εισάγει συγκεκριμένες σχεδιαστικές απαιτήσεις.

Ευφυής Πράκτορας και ομιλία

Εισαγωγή

Όπως είπαμε παραπάνω ένας εικονικός πράκτορας μπορεί να θεωρηθεί στα πλαίσια ενός εικονικού περιβάλλοντος, ως ένα υποσύστημα που αποτελείται από ένα συστατικό παραγωγής συμπεριφοράς και ένα εικονικό σώμα. Στο κεφάλαιο αυτό θα αναφερθούμε μόνο στο κομμάτι του εικονικού σώματος και της λειτουργίας από την σκοπιά της αλληλεπίδρασης του με τον εικονικό κόσμο.

Στα πλαίσια της αναπαράστασης REVE το σώμα ενός εικονικού πράκτορα είναι ένα item με διευρυμένες ιδιότητες. Επομένως όλες οι σχεδιαστικές απαιτήσεις, περιορισμοί αλλά και δυνατότητες του item καθορίζουν το σώμα του εικονικού πράκτορα, κάνοντας έτσι συγκεκριμένη την υλοποίησή του.

Στην παρούσα εργασία εξελίξαμε έναν εικονικό πράκτορα ο οποίος είχε ήδη ένα εικονικό σώμα, δυνατότητες αλληλεπίδρασης με τον εικονικό κόσμο και με τα συστατικά του σώματος του. Είχε επίσης δυνατότητα να αντιλαμβάνεται τον κόσμο, να αποθηκεύει και να επεξεργάζεται πληροφορίες. Η συμβολή μας στην παρούσα εργασία είναι η εξέλιξη των δυνατοτήτων αλληλεπίδρασης του πράκτορα δίνοντας του την ικανότητα κατανόησης και εκφοράς λόγου, τόσο σε ακουστικό όσο και οπτικό επίπεδο. Με τον τρόπο αυτό μπορεί πλέον ο πράκτορας μας, πέρα από τις άλλες δυνατότητες που έχει για αλληλεπίδραση και αντίληψη του περιβάλλοντος του, να αλληλεπιδρά με τρόπο φυσικό με άλλους πράκτορες ή με τον χρήστη.

Για να επιτευχθεί ο στόχος αυτός είναι αναγκαία η προσθήκη χαρακτηριστικών και ιδιοτήτων σε καθένα από τα τρία επίπεδα που ορίζουν το σώμα του εικονικού πράκτορα, δηλαδή στο physical aspect, access aspect, semantic aspect, όπως τα ορίσαμε παραπάνω, αλλά και στις ιδιότητες που έχει ως εικονικός πράκτορας, και συγκεκριμένα στο activity aspect του.

Εικονικό σώμα

Το εικονικό σώμα του πράκτορα, δεδομένου ότι είναι ένα item ορίζεται ως το σύνολο των item aspects που το αποτελούν. Στο κεφάλαιο αυτό θα αναφερθούμε μόνο στις απαραίτητες προσθήκες που πρέπει να γίνουν σε καθένα από αυτά χωρίς να γίνεται αναφορά στο στις ήδη υπάρχουσες δομές και ιδιότητες του πράκτορα.

Physical Aspect

Το physical aspect ορίζει την αναπαράσταση του εικονικού αντικειμένου, στην περίπτωση μας το σώμα του πράκτορα, στα πλαίσια του εικονικού κόσμου. Ορίζει δηλαδή πως θα εμφανίζεται ο πράκτορας, τι φυσικά χαρακτηριστικά θα έχει(μέγεθος, μέλη του σώματος κτλ) και οποιαδήποτε άλλη πληροφορία θέλουμε να εμφανίζεται ως μέρος του εικονικού σώματος.

Επομένως στο επίπεδο αυτό χρειαζόμαστε μια δομή που να αναπαριστά την ομιλία του πράκτορα έτσι ώστε να ενσωματωθεί στο ήδη υπάρχον σώμα του. Μια βασική απαίτηση που πρέπει να ικανοποιηθεί είναι η λειτουργικότητα και η ευκολία που πρέπει παρέχει στον χρήστη να παρατηρεί την ομιλία, αλλά και από άποψη επικοινωνίας πρακτόρων θέλαμε το speech του πράκτορα σε σχεδιαστικό επίπεδο να είναι άμεση προέκταση του σώματός του ώστε να μπορεί ένας άλλος πράκτορας χωρίς εμπόδια ή άλλες επιπλοκές στο οπτικό του πεδίο να “δει” το κείμενο. Η σχεδιαστική επιλογή που έγινε ήταν η δημιουργία ενός πλαισίου πάνω από το κεφάλι του εικονικού μας πράκτορα, μέσα στο οποίο εμφανίζεται το κείμενο της ομιλίας.

Η υλοποίηση του speechbox, του πλαισίου με την ομιλία, γίνεται προσθέτοντας έναν κόμβο στο γράφημα σκηνής του σώματος του agent, ορίζοντας τη θέση του και τις ιδιότητες που θα έχει. Στη δική μας περίπτωση μας αρκεί ένας ένας κόμβος text και η ιδιότητα που μας ενδιαφέρει είναι η String, η τιμή της οποίας θα είναι το speech του πράκτορα.

Semantic aspect

Το επόμενο βήμα στη διαδικασία της μοντελοποίησης είναι οι προσθήκες που πρέπει να γίνουν στο semantic aspect. Εδώ, σε αντίθεση με το physical aspect που κάναμε μια σχεδιαστική επιλογή για να ενσωματώσουμε την αφηρημένη ομιλία, τα πράγματα είναι πιο συγκεκριμένα, μιας και η πληροφορία η οποία είναι αναγκαία να είναι διαθέσιμη για το item “speechbox” είναι συγκεκριμένη. Ειδικότερα αυτό που μας ενδιαφέρει δεν είναι ούτε η θέση ούτε κάποια άλλη κατάσταση του item παρά μόνο το string της ομιλίας. Έτσι ο στόχος της διαδικασίας προσθήκης στο semantic aspect για το συγκεκριμένο item είναι να αναπαριστά και να καθιστά διαθέσιμο προς αντίληψη αυτό το string. Δημιουργούμε έτσι έναν τύπο κόμβο “name says”, όπου name το όνομα του πράκτορα μας. Ο τύπος αυτός αντιστοιχεί στην ομιλία του πράκτορα, και πιο συγκεκριμένα στην ιδιότητα String του κόμβου Text που ορίσαμε παραπάνω. Σε επίπεδο υλοποίησης η άντληση της απαιτούμενης πληροφορίας γίνεται διαμέσου ενός αντικειμένου CompoundSymbol για την διαχείριση της πληροφορίας και ενός X3DConnector αντικειμένου για την σύνδεση με τον κόμβο Text του γραφήματος σκηνής.

Access aspect

Η διαδικασία της μοντελοποίησης ολοκληρώνεται με τις απαραίτητες προσθήκες στο access aspect. Και εδώ τα πράγματα είναι τελείως συγκεκριμένα καθώς η διαθέσιμη λειτουργικότητα που θέλουμε να προσδώσουμε στον agent και επομένως και η δομή του access aspect, είναι συγκεκριμένη. Όπως αναφέρεται παραπάνω αυτό που χρειαζόμαστε είναι λειτουργικότητα αλλαγής του String της ομιλίας. Έτσι ο στόχος της διαδικασίας σχεδιασμού του access aspect για το συγκεκριμένο item είναι το να αναπαριστά και να καθιστά διαθέσιμη την παραπάνω λειτουργικότητα. Για το στόχο αυτό ορίζονται οι παρακάτω τύποι κόμβων, με την αντίστοιχη σημασιολογία.

AccessPoint: Ο τύπος αυτός αντιστοιχεί στο πρώτο και ίσως, αμεσότερης χρησιμότητας και γενικότερης σημασίας είδος πληροφορίας που παρέχει το access aspect. Αναπαριστά ένα accesspoint ως μέσο πρόσβασης σε λειτουργίες και σε συγκεκριμένη χωρική σχέση με την αναπαράσταση του item. Καθώς η διαθέσιμη λειτουργικότητα είναι μόνο η ομιλία χρειαζόμαστε ένα μόνο accesspoint. Λόγω της σχέσης που πρέπει να υπάρχει ανάμεσα στο accesspoint και στη φυσική αναπαράσταση του item, ο κόμβος αυτός συνδέεται με τον κόμβο τύπου Transform του physical aspect του item ώστε η πρόσβαση στη λειτουργία που καθιστά διαθέσιμη το accesspoint να είναι ανά πάσα στιγμή πιστευτά εντοπισμένη στο χώρο. Παρόλο που στην δική

μας περίπτωση η λειτουργικότητα είναι κατά κάποιον τρόπο “εσωτερική” του πράκτορα και δεν προβλέπεται η πρόσβαση σε αυτή από άλλον πράκτορα ή αντικείμενο, ακολουθούμε την παραπάνω αρχιτεκτονική για τη δημιουργία των accesspoints γιατί πέρα από τα σχεδιαστικά πλεονεκτήματα που έχει (χωρική αναπαράσταση) έχει και μεγάλη λειτουργική και προγραμματιστική αξία(τεράστια ευκολία στην επέκταση και τροποποίηση του κώδικα).

Talkfunction : Ο τύπος αυτός πληροφορεί σχετικά με την ιδιότητα του text της ομιλίας να αλλάζει. Έτσι επιτρέπεται εξειδικευμένη αλληλεπίδραση με το συγκεκριμένο αντικείμενο. Σε επίπεδο υλοποίησης χρησιμοποιείται μια δική μας παραλλαγή της κλάσης X3DFieldFunction, η εκτέλεση της οποίας έχει σαν αποτέλεσμα ορισμένη μεταβολή της τιμής συγκεκριμένου πεδίου συγκεκριμένου κόμβου X3D του physical aspect του item. Η X3DFieldFunction αποτελεί μια απλή, γενικής χρήσης προσέγγιση στην υλοποίηση της παραπάνω ιδιότητας, καθώς δεν πραγματοποιεί καμία επεξεργασία στην τιμή που θέτει το πεδίο στο οποίο αναφέρεται, επομένως έπρεπε να ορίσουμε εμείς τον τύπο και την απαραίτητη επεξεργασία του.

Πλέον έχουμε ολοκληρώσει την απαραίτητη υποδομή για την λειτουργικότητα του σώματος πράκτορας και το μόνο που μας είναι οι διαδικασίες και ιδιότητες εκείνες από την πλευρά του πράκτορα που θα μας επιτρέψουν την χρήση της παραπάνω υποδομής.

Activity Aspect

Όπως είπαμε προηγουμένως το activity aspect περιγράφει τις δυνατότητες ενός εικονικού πράκτορα για αλληλεπίδραση. Για το σκοπό αυτό χρησιμοποιεί effectors(σχεδιαστική προσέγγιση εικονικών επιδραστών) και actions(συγκεκριμένη, μοναδική δυνατότητα του effector). Στην περίπτωση μας χρειαζόμαστε έναν effector που διαισθητικά να προσιδιάζει το στόμα, την δυνατότητα μας να εκφραστούμε, και μια action που να παραπέμπει την ομιλία αυτή καθεαυτή, σαν δυνατότητα του στόματος.

Effector: Δημιουργούμε έναν effector, τον οποίο συνδέουμε με έναν κόμβο του physical aspect του πράκτορα σε τέτοιο σημείο ώστε να καλύπτονται οι περιορισμοί και οι σχεδιαστικές απαιτήσεις που είδαμε προηγουμένως. Ο effector αυτός θα αναλάβει το ρόλο να προσκολληθεί στο accesspoint και να διαμεσολαβήσει της action.

Action : Στην περίπτωση μας το action είναι η ομιλία, η εμφάνιση ενός string στην τιμή του αντίστοιχου πεδίου στον κόμβο text που έχουμε δημιουργήσει. Ο βασικός ρόλος της action αυτής είναι να παράγει τα απαραίτητα function execution arguments, δηλαδή το νέο string που θα αναπαριστά την ομιλία, και να υποβάλει ένα αίτημα για εκτέλεση του Talkfunction(γιατί με αυτό θα συνδεθεί ο effector).

Πλατφόρμα jReve

Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται η πλατφόρμα jREVE, μία ολοκληρωμένη, πλήρως υλοποιημένη και λειτουργική πλατφόρμα ανάπτυξης λογισμικού (software development kit, SDK) για την ανάπτυξη και τη λειτουργία εικονικών περιβαλλόντων, βασισμένη στην προτεινόμενη αναπαράσταση REVE. Κύρια συστατικά της είναι, αφ’ ενός μία παραδειγματική υλοποίηση (sample implementation) της αναπαράστασης items και εικονικών σωμάτων βάσει της έννοιας του item aspect υπό τη μορφή βιβλιοθήκης κλάσεων (class library) αξιοποιήσιμης μέσω κατάλληλης διεπαφής ανάπτυξης εφαρμογών (application programming interface, API), αφ’ ετέρου συνοδευτικά συστατικά λογισμικού που στοχεύουν στην υποστήριξη της υλοποίησης στα πλαίσια εφαρμογών, την διασύνδεσή της με άλλα τμήματα τους, καθώς και την παρακολούθηση της λειτουργίας της για σκοπούς διερεύνησης, βελτιστοποίησης και αποσφαλμάτωσης. Η πλατφόρμα είναι εξ’ ολοκλήρου υλοποιημένη στη γλώσσα Java.

Η πλατφόρμα έχει σχεδιαστεί και υλοποιηθεί έτσι ώστε να καλύπτει συγκεκριμένα κριτήρια ως προϊόν λογισμικού.

Να είναι ικανή να αντιμετωπίσει αποτελεσματικά τις σύγχρονες ανάγκες στο χώρο του σχεδιασμού, της υλοποίησης και της λειτουργίας εικονικών περιβαλλόντων.

Να είναι επεκτάσιμη, συντηρήσιμη και προσαρμόσιμη σε εξελισσόμενες τεχνολογίες, έτσι ώστε να μπορεί να σταθεί ως επένδυση στα πλαίσια μίας γενικότερης και μακροπρόθεσμης ενασχόλησης με εικονικά περιβάλλοντα. Αυτός είναι, ίσως, και ο σημαντικότερος στόχος της πλατφόρμας.

Να βασίζεται, κατά το μέγιστο, σε λογισμικό ανοικτού κώδικα ή/και ελεύθερο λογισμικό,

έτσι ώστε η εσωτερική της λειτουργία να είναι απόλυτα διαφανής, ενώ ταυτόχρονα η δυνατότητα αξιοποίησης και διακίνησής της να είναι αδέσμευτη από όρους εμπορικής εκμετάλλευσης.

Όπου αυτό είναι αναγκαίο, να αξιοποιεί λογισμικό τρίτων κατασκευαστών το οποίο αναπτύσσεται και υποστηρίζεται ενεργά, ενώ παρέχει και εγγυήσεις για υπολογίσιμη και

ικανοποιητική διάρκεια ζωής.

Να είναι πλήρως τεκμηριωμένη.

Να είναι εύχρηστη και προσβάσιμη, ώστε να είναι δυνατή η αξιοποίησή της σε περιπτώσεις όπου η συμβολή της ενδέχεται να είναι σημαντική.

Να αποτελεί ένα ολοκληρωμένο, κομψό, σύγχρονο και αποτελεσματικό προϊόν λογισμικού.

Extensible 3D Graphics (X3D) και Xj3D

Το physical aspect δεν χρησιμοποιεί κάποια ίδια εσωτερική αναπαράσταση για την πληροφορία που περιέχει και που αφορά το επίπεδο υλοποίησης, αλλά οποιαδήποτε διαθέσιμη αναπαράσταση έχει τη μορφή γραφήματος σκηνής. Σαν αποτέλεσμα, προκειμένου να είναι στην πράξη χρήσιμη και λειτουργική, μία συμπαγής υλοποίηση της αφηρημένης κλάσης PhysicalAspect οφείλει να καθορίζει με ακρίβεια την επιλογή εσωτερικής αναπαράστασης για το επίπεδο υλοποίησής της. Επιπλέον, οφείλει να τυποποιεί τους μηχανισμούς αλληλεπίδρασής μεταξύ των επιπέδων της υλοποίησης και item και, σαν αποτέλεσμα, να διαθέτει τη δυνατότητα αξιοποίησης ως συστατικό της αναπαράστασης του item στα πλαίσια κάποιου συγκεκριμένου υλοποιημένου εικονικού περιβάλλοντος. Η πλατφόρμα, τόσο για τη συμπαγή υλοποίηση του physical aspect αλλά και σαν υποστηριζόμενη μορφή αρχικών δεδομένων γραφημάτων σκηνής για χρήση σε ορισμούς VERL, υιοθετεί το πρότυπο Extensible 3D Graphics (X3D).

Συντακτικά, το X3D είναι απόλυτα συμβατό με το physical aspect, καθώς τόσο η μορφή αρχείου που υποστηρίζει όσο και η εσωτερική του αναπαράσταση τρισδιάστατων σκηνών βασίζονται στην έννοια του γραφήματος σκηνής. Έτσι, δεδομένα X3D μπορούν, ταυτόχρονα, να φιλοξενηθούν στη δομή του physical aspect - όπου, συγκεκριμένα, κάθε κόμβος του physical aspect αντιστοιχεί σε ένα κόμβο X3D - και άρα να αποτελέσουν μέρος του μοντέλου ενός item, καθώς και να αξιοποιηθούν, ταυτόχρονα, από μία βασισμένη σε X3D μηχανή μορφοποίησης.

Οι μορφές αρχείου που υποστηρίζει το X3D μπορούν να βρίσκονται είτε σε δυαδική, συμπιεσμένη μορφή με προσανατολισμό στην ταχύτητα επεξεργασίας και μετάδοσης και το μικρό μέγεθος, είτε σε μορφή κειμένου, ευανάγνωστη και διαθέσιμη προς επεξεργασία από χρήστες. Επιπλέον, καθώς το X3D ουσιαστικά αποτελεί το «διάδοχο» της γλώσσας VRML, υποστηρίζει επιπρόσθετα και τη μορφή αρχείου VRML97, γεγονός που αυξάνει δραματικά τον αριθμό διαθέσιμων προς χρήση τρισδιάστατων μοντέλων σκηνών και αντικειμένων, αυξάνοντας,

έτσι, τη συνολική χρηστικότητα της πλατφόρμας και τη διαθεσιμότητα αρχικού υλικού για το σχεδιασμό εικονικών κόσμων.

Η βιβλιοθήκη Xj3D είναι μία υλοποίηση του προτύπου X3D σε γλώσσα Java. Η έκδοσή της η οποία χρησιμοποιήθηκε στην ανάπτυξη της πλατφόρμας παρέχει πλήρη υποστήριξη για το σύνολο των διαθέσιμων κόμβων, ενώ συμπεριλαμβάνει και λογισμικό διασύνδεσης εφαρμογών με σκηνές X3D. Είναι ένα ενεργό project που υποστηρίζεται τόσο από το Web3D Consortium όσο και από μεγάλο αριθμό επαγγελματιών και χρηστών. Για τους λόγους αυτούς και εξ' αιτίας της έλλειψης ισχυρά ανταγωνιστικών - με βάση τους στόχους της πλατφόρμας - επιλογών, η βιβλιοθήκη Xj3D επικράτησε ως προτιμητέα επιλογή για την ανάπτυξη των σχετικών με X3D μερών του physical aspect και άλλων συστατικών της πλατφόρμας.

Συμβολική αναπαράσταση γνώσης

Για σκοπούς συμβολικής αναπαράστασης γνώσης, η πλατφόρμα υιοθετεί μία προσέγγιση δανεισμένη από το λογικό προγραμματισμό. Έτσι, παρέχει υποστήριξη για ατομικά (atomic) και σύνθετα (compound) σύμβολα, όπως επίσης και για μη θεμελιωμένα σύμβολα (non-ground) σύμβολα και μεταβλητές.

Σε σχεδιαστικό επίπεδο, ένα ατομικό σύμβολο αναπαρίσταται από ένα μοναδικό και χωρίς συσχετίσεις κόμβο της εσωτερικής δομής του semantic aspect, ο οποίος αντιστοιχεί σε χαρακτηριστικό ή σχολιασμό. Ένα σύνθετο σύμβολο αναπαρίσταται και αυτό από ένα μοναδικό κόμβο της εσωτερικής δομής του semantic aspect, ο οποίος και σε αυτή την περίπτωση αντιστοιχεί είτε σε χαρακτηριστικό είτε σε σχολιασμό, με τη διαφορά ότι σχετίζεται με συγκεκριμένο αριθμό άλλων κόμβων οι οποίοι αναπαριστούν τα ορίσματα του σύνθετου συμβόλου ως σύμβολα.

Σε επίπεδο υλοποίησης, κάθε σύμβολο έχει τη δυνατότητα να παρέχει μια συγκεκριμένη τιμή, ή μια αόριστη τιμή αν είναι μη θεμελιωμένο. Η τιμή ενός ατομικού συμβόλου εξαρτάται από τον τύπο και την ανά πάσα στιγμή κατάστασή του, ενώ η τιμή ενός σύνθετου συμβόλου είναι η n-άδα των τιμών των n ορισμάτων του.

Κύκλος ζωής

Στα πλαίσια ενός ολοκληρωμένου κύκλου ζωής ενός εικονικού περιβάλλοντος, η λογική αξιοποίηση της πλατφόρμας έχει όπως παρακάτω :

1. Δημιουργείται ένα αντικείμενο Browser το οποίο εισάγεται στη γραφική διεπαφή χρήστη του εικονικού περιβάλλοντος.
2. Δημιουργείται ένα νέο αντικείμενο Scene για το παραπάνω Browser.
3. Το scene αρχικοποιείται είτε με χρήση των μηχανισμών επεξεργασίας ορισμών εικονικών κόσμων σε VERL όπως παρέχονται από το πακέτο `rene.verl`, ή μέσω άλλων υποστηριζόμενων μεθόδων. Η διαδικασία είναι ανεξάρτητη του μηχανισμού:
 - a. Για κάθε ορισμό item:
 - i. Κατασκευάζεται ένα νέο PhysicalAspect.
 - ii. Καλείται η `create` του PhysicalAspect για το υποδέντρο με κόμβο-ρίζα `source` όπως αυτό ορίζεται από τα `resource` και `behaviours` του γονικού `itemgroup`. Η `create` αρχικοποιεί τον κόμβο-ρίζα με βάση τον ορισμό `TransformInfo`, και τους ειδικευμένους κόμβους με βάση τα `fit` και `fitCentre`.
 - iii. Κατασκευάζεται ένα νέο SemanticAspect.
 - iv. Για κάθε ορισμό συμβόλου, κατασκευάζεται ένα `BaseCompoundSymbol`.
 - v. Για κάθε ορισμό ορίσματος, κατασκευάζεται ένα νέο αντικείμενο της κλάσης-υλοποίησης του `interface Argument` που ορίζεται από το `class`.
 - vi. Καλείται η `create` του αντικειμένου `Argument` με ορίσματα όπως ορίζει το `args`.

- vii. Το αντικείμενο προστίθεται στα ορίσματα του BaseCompoundSymbol.
 - viii. Καλείται η create του BaseCompoundSymbol με όνομα όπως ορίζει το name.
 - ix. Τα BaseCompoundSymbols που δημιουργήθηκαν προστίθενται στα σύμβολα του SemantiAspect.
 - x. Καλείται η create του SemanticAspect με item class όπως ορίζεται από το class του γονικού item.
 - xi. Για κάθε ορισμό accesspoint, κατασκευάζεται ένα AccessPoint.
 - xii. Για κάθε ορισμό function, κατασκευάζεται ένα νέο αντικείμενο της κλάσης-υλοποίησης του interface BaseFunction που ορίζεται από το class.
 - xiii. Καλείται η create του αντικειμένου BaseFunction με ορίσματα όπως αυτά ορίζονται από το args.
 - xiv. Το αντικείμενο BaseFunction προστίθεται στα ορίσματα του AccessPoint.
 - xv. Καλείται η create του AccessPoint με όνομα όπως ορίζει το name και κόμβο του physical aspect με όνομα όπως ορίζει το node.
 - xvi. Τα AccessPoints που δημιουργήθηκαν προστίθενται στα accesspoints του AccessAspect.
 - xvii. Καλείται η create του AccessAspect.
 - xviii. Καλείται η create του item με τα PhysicalAspect, SemanticAspect και AccessAspect, και όνομα όπως ορίζει το name.
- b. Δημιουργούνται debug artifacts.
4. Δημιουργείται ο μηχανισμός επεξεργασίας αιτήσεων για σύνδεση σε body controllers μέσω δικτύου και άλλων υποστηριζόμενων μεθόδων.
5. Διαχείριση εικονικών σωμάτων (το στάδιο επαναλαμβάνεται καθ' όλη τη διάρκεια της λειτουργίας του εικονικού περιβάλλοντος):
- a. Δημιουργία εικονικών σωμάτων:
 - i. Κατασκευάζεται ένα νέο Body.
 - ii. Το Body αρχικοποιείται, π.χ. με κλήση μίας μεθόδου create, η συμπαγής υλοποίηση της οποίας δημιουργεί συγκεκριμένα sensors και effectors.
 - iii. Δημιουργείται ένα BodyControlFacade για το Body, το οποίο φυλάσσεται για μελλοντική χρήση από ενδιαφερόμενα BodyControllers ή, σε περίπτωση όπου η δημιουργία του Body ήταν αποτέλεσμα αίτησης σε κάποιο ενεργό BodyController, προωθείται σε αυτό για περαιτέρω διαχείριση.
 - b. Αλληλεπίδραση με εικονικά σώματα: ενεργός body controller - για παράδειγμα, κάποιος CLIBodyController ή TCIPBodyController - λαμβάνει εντολή από συνδεδεμένο client, την οποία και επεξεργάζεται και εντοπίζει τη σχετική λειτουργικότητα του BodyControlFacade του εικονικού σώματος το οποίο διαχειρίζεται. Για παράδειγμα: καλείται η μέθοδος sense του BodyControlFacade. Καλείται η μέθοδος getSensors του εικονικού σώματος. Για κάθε sensor, καλείται η μέθοδος sense. Τα αποτελέσματα συγκεντρώνονται και επιστρέφονται. Εναλλακτικά, αν έχει δοθεί ως όρισμα το όνομα κάποιου sensor, γίνεται απόπειρα εντοπισμού του και, αν υπάρχει, καλείται η δική του μέθοδος sense της οποίας τα αποτελέσματα επιστρέφονται.
 - c. Διαγραφή εικονικών σωμάτων, είτε ως αποτέλεσμα αιτήσεων ενεργών BodyControllers ή με πρωτοβουλία του συστήματος. Σε κάθε περίπτωση, ένα BodyController του οποίου το διαχειριζόμενο εικονικό σώμα διεγράφη, πραγματοποιεί τις αναγκαίες ενέργειες εκκαθάρισης (π.χ. ειδοποίηση χρηστών, αποδέσμευση πόρων, κατάργηση συνδέσεων δικτύου, κ.τ.λ.).
6. Τερματισμός εικονικού περιβάλλοντος.

Ομιλία και τεχνολογίες ομιλίας

Ομιλία ως μέσο επικοινωνίας

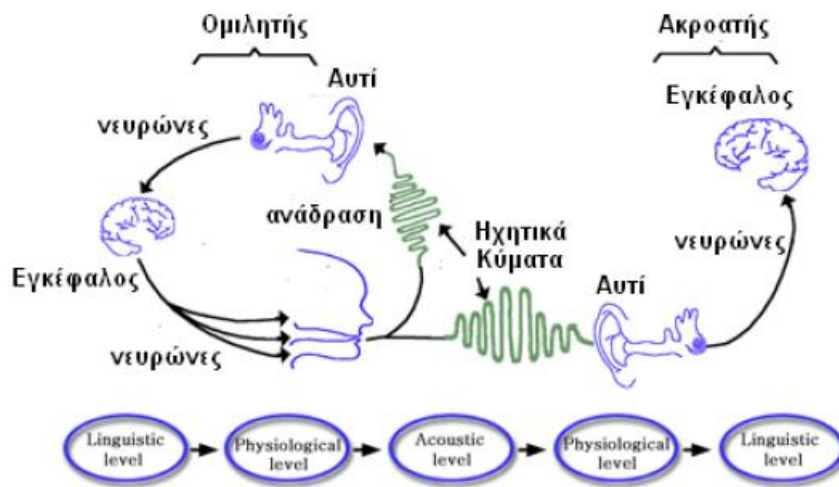
Ένας καταλυτικός παράγοντας στην ανάπτυξη του ανθρώπινου πολιτισμού είναι η ομιλία. Η ομιλία αποτελεί τον βασικό τρόπο επικοινωνίας μεταξύ των ανθρώπων και μαζί με τον γραπτό λόγο αποτελούν τα δύο κύρια μέσα ανταλλαγής απόψεων, ιδεών, γνώσης και πολιτισμού. Η ανταλλαγή εμπειριών, ιδεών, αλλά και η μεταφορά γνώσης από τη μια γενιά ανθρώπων στην επόμενη πραγματοποιήθηκε σε μεγάλο βαθμό προφορικά. Αν και ο όγκος των βιβλίων, των εφημερίδων και όλων των άλλων εντύπων που τυπώνονται σήμερα είναι τεράστιος, ωστόσο η ποσότητα της πληροφορίας που ανταλλάσσεται με την προφορική ομιλία είναι πολύ μεγαλύτερη. Επιπλέον, η προφορική επικοινωνία είναι ταχύτερη και πιο εύχρηστη από την γραπτή. Αυτά τα δεδομένα καθιστούν την ομιλία και ότι σχετίζεται με αυτή απαραίτητα, από την πιο απλή επικοινωνία μεταξύ δύο ανθρώπων, έως την επικοινωνία μέσα σε διοικητικές δομές ή ακόμη και κράτη.

Η ομιλία, μέσα από την ακατάπαυστη χρήση της στην καθημερινή ζωή ως εργαλείο επικοινωνίας αναπτύχθηκε σε ένα εξαιρετικό αποδοτικό σύστημα επικοινωνίας με στόχο την ανταλλαγή ακόμα και των πιο πολυσήμαντων ιδεών. Σ' αυτό βοήθησε το γεγονός ότι η προφορική ομιλία παραμένει λειτουργικά ανεπηρέαστη από την διαφορετικότητα της φωνής των ανθρώπων, των ιδιαιτεροτήτων ομιλίας, των διαφορών στη προφορά που μπορούν να εμφανιστούν, όταν εκατομμύρια άνθρωποι χρησιμοποιούν την ίδια γλώσσα.

Η εξέλιξη διάφορων τομέων της τεχνολογίας και της επιστήμης επέτρεψε την ανάπτυξη συστημάτων και εργαλείων με τα οποία μπορούν να αυτοματοποιηθούν διαδικασίες που σχετίζονται με την ομιλία. Είναι προφανές ότι τα συστήματα που έχουν να κάνουν με άμεση επικοινωνία ή συνδιαλλαγή με τον άνθρωπο, για να είναι φιλικά προς τον χρήστη, θα πρέπει να αντιγράψουν τον τρόπο επικοινωνίας μεταξύ των ανθρώπων. Τέτοια συστήματα μπορεί να είναι αυτόματα τηλεφωνικά κέντρα, υπηρεσίες πληροφόρησης (για παράδειγμα τουριστικά κιόσκια), υπηρεσίες εξυπηρέτησης (για παράδειγμα κράτηση εισιτηρίων), έλεγχος χώρων/εισοδών με ομιλία, έξυπνα σπίτια, έξυπνα αυτοκίνητα κ.λπ.. Για παράδειγμα σε ένα τηλεφωνικό κέντρο, η χρήση ανθρώπων τηλεφωνητών αντικαθίσταται από αυτόματα συστήματα εξυπηρέτησης μέσω τηλεφώνου. Ειδικά για την περίπτωση των εφαρμογών που λειτουργούν υπό σταθερές συνθήκες θορύβου ή τρόπου ομιλίας, η σύγχρονη τεχνολογία ομιλίας επιτρέπει την αντικατάσταση ανθρώπων-υπαλλήλων συνομιλητών από διαλογικά συστήματα.

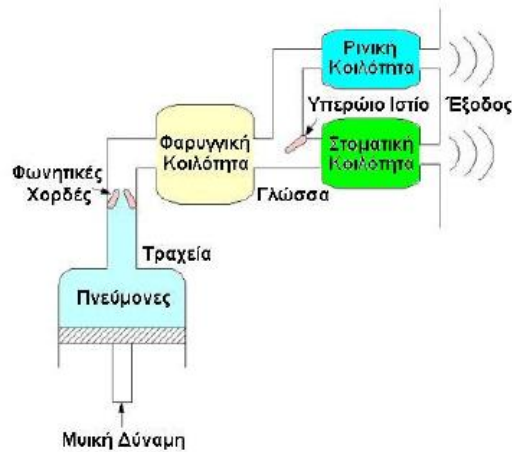
Η τεχνολογία ομιλίας είναι ο τομέας της επιστήμης που ασχολείται με την επικοινωνία μεταξύ ανθρώπου και μηχανής (human-computer interaction – HCI) με τον πλέον φυσικό τρόπο, δηλαδή την ομιλία. Επιπλέον, η τεχνολογία ομιλίας ενδιαφέρεται για τον αρμονικό συνδυασμό της ομιλίας με άλλα είδη επικοινωνίας, όπως οπτική επαφή, χειρονομίες κ.λπ.

Όπως αναφέρθηκε και παραπάνω, τις τελευταίες δεκαετίες η ανάπτυξη συστημάτων που αφορούν την φωνητική αλληλεπίδραση μεταξύ ανθρώπου και μηχανής έχει αυξηθεί σημαντικά. Η επικοινωνία ανθρώπου μηχανής περιλαμβάνει την μετάδοση μηνυμάτων από τον άνθρωπο στην μηχανή, δηλαδή την μετατροπή φυσικής ομιλίας σε κείμενο (ή αναγνώριση ομιλίας), και την μετάδοση μηνυμάτων από την μηχανή στον άνθρωπο, δηλαδή την μετατροπή ενός κειμένου που παράγεται αυτόματα σε συνθετική ομιλία (ή σύνθεση ομιλίας).



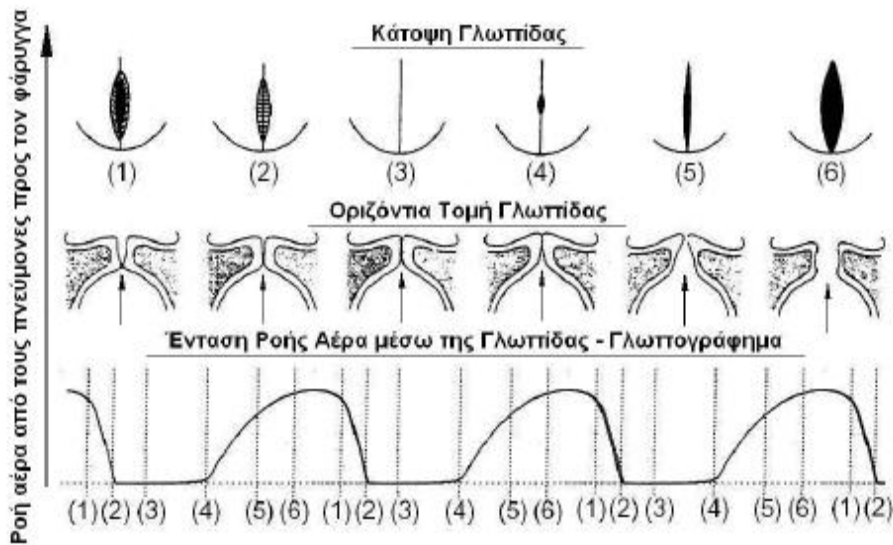
Η αλυσίδα επικοινωνίας από άνθρωπο σε άνθρωπο μέσω ομιλίας

Στην εικόνα φαίνεται η «αλυσίδα ομιλίας», δηλαδή η από τον ομιλητή έως τον ακροατή διαδρομή μετάδοσης ενός δεδομένου (νοήματος). Αρχικά κάποια νοήματα (semantics) παράγονται από τον εγκέφαλο. Τα νοήματα αυτά μεταφράζονται από τον εγκέφαλο σε προτάσεις φυσικής ομιλίας και μέσα από τους νευρώνες του νευρικού συστήματος, δίνεται εντολή σε όλους τους μύες που εμπλέκονται μέσα από το φωνητικό κανάλι στην παραγωγή ομιλίας να παράγουν το κύμα αέρα που αντιστοιχεί στην ομιλία της αντίστοιχης πρότασης. Συγκεκριμένα, οι μύες των πνευμόνων δημιουργούν μια πίεση αέρα που διέρχεται διαμέσου των φωνητικών χορδών, δηλαδή διαμέσου της γλωττίδας. Ανάλογα με το αν εκείνη την στιγμή παράγεται ηχηρή (voiced) ή άηχη (unvoiced) ομιλία οι φωνητικές χορδές βρίσκονται αντίστοιχα σε ταλάντωση ή χαλάρωση, δημιουργώντας μια ροή αέρα με παλμική ή τυρβώδη μορφή αντίστοιχα. Στη συνέχεια, η εξερχόμενη της γλωττίδας ροή αέρα διέρχεται μέσα από τον φάρυγγα και τον λάρυγγα, και στη συνέχεια διαμέσου της στοματικής κοιλότητας, σε σύζευξη ή όχι με την ρινική κοιλότητα, εξέρχεται διαμορφωμένη με την μορφή του σήματος ομιλίας. Ανάλογα με το τι προφέρεται ανά χρονική στιγμή η φωνητική οδός μαζί με την σιαγόνα και την γλώσσα παίρνουν την κατάλληλη θέση άρθρωσης ώστε να παραχθεί ο κατάλληλος ήχος. Από την μεριά του ακροατή, το αυτί λαμβάνει το σήμα ομιλίας και το μετατρέπει στα αντίστοιχα νευρικά σήματα. Στη συνέχεια τα νεύρα μεταφέρουν την πληροφορία αυτή στον εγκέφαλο, ο οποίος με την σειρά του την μετατρέπει σε νοήματα τα οποία και επεξεργάζεται αναλόγως. Ένα μπλοκ διάγραμμα του μηχανισμού παραγωγής ομιλίας φαίνεται στην εικόνα.



Μπλοκ διάγραμμα του μηχανισμού παραγωγής ομιλίας

Η ομιλία είναι στην πραγματικότητα ένα σύνολο από στοιχειώδεις ήχους. Οι ήχοι αυτοί ονομάζονται φωνήματα και είναι διαφορετικοί για κάθε γλώσσα. Τα φωνήματα, δηλαδή οι στοιχειώδεις ήχοι που απαρτίζουν την ανθρώπινη ομιλία, χωρίζονται σε δυο κατηγορίες, τα ηχηρά και τα άηχα φωνήματα. Ανάλογα με το εάν προφέρεται ηχηρό ή άηχο φώνημα κάθε στιγμή κατά την διάρκεια της ομιλίας, οι φωνητικές χορδές πάλλονται (ταλαντώνονται) στην γλωττίδα ή διατηρούνται ακίνητες, αντίστοιχα. Ενώ κατά την προφορά των άηχων φωνημάτων ο αέρας βγαίνει από την γλωττίδα με μια τυρβώδη ροή, κατά την διάρκεια των ηχηρών φωνημάτων δημιουργείται μια ροή αέρα που έχει την μορφή παλμών. Ένα παράδειγμα ροής αέρα κατά την διάρκεια ηχηρής ομιλίας δίνεται στην εικόνα.

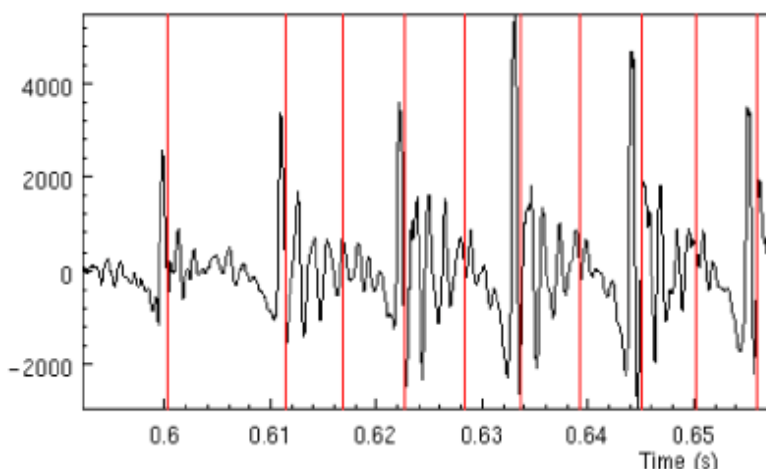


Παράδειγμα ροής αέρα εξόδου από την γλωττίδα κατά την διάρκεια ηχηρής ομιλίας

Η περίοδος ταλάντωσης των φωνητικών χορδών ονομάζεται θεμελιώδης περίοδος (T_0), και η αντίστοιχη συχνότητα ονομάζεται θεμελιώδης συχνότητα (F_0) ή pitch. Σε πολλές εφαρμογές της τεχνολογίας ομιλίας, όπως για παράδειγμα στην σύνθεση ομιλίας, η επεξεργασία των δεδομένων γίνεται σύγχρονα με το pitch, ή όπως έχει επικρατήσει, η ανάλυση είναι pitch-synchronous.

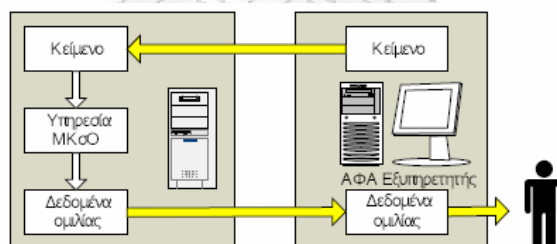
Ένα σημαντικό μέγεθος στην pitch-synchronous ανάλυση είναι η εξαγωγή των σημείων που κλείνει (ή ανοίγει) η γλωττίδα. Τα σημεία αυτά έχουν επικρατήσει στην βιβλιογραφία με τον όρο pitchmarks και χρησιμοποιούνται κυρίως στην σύνθεση ομιλίας (Dutoit, 1996; Black and Lenzo, 2007). Ένα παράδειγμα κυματομορφής ομιλίας όπου φαίνονται τα αντίστοιχα

pitchmarks φαίνεται στην εικόνα. Είναι προφανές ότι δύο διαδοχικά pitchmarks απέχουν μεταξύ τους απόσταση ίση με την θεμελιώδη περίοδο T_0 .



Παράδειγμα κυματομορφής ομιλίας όπου φαίνονται τα αντίστοιχα pitchmarks

Κάθε φώνημα από μόνο του δεν έχει καμία νοηματική αντιστοίχιση, ωστόσο συνδυασμοί αυτών των ήχων και μάλιστα σε συγκεκριμένη σειρά αντιστοιχούν σε νοήματα, δηλαδή στις λέξεις κάθε γλώσσας. Μάλιστα το νόημα που αντιστοιχίζεται σε κάθε λέξη μπορεί να διαφοροποιείται για την ίδια ακολουθία φωνημάτων, σύμφωνα με το που τονίζεται η λέξη, ή σύμφωνα με τις λέξεις που προηγούνται ή ακολουθούν. Παρόλο που κάθε γλώσσα αποτελείται από το δικό της σύνολο φωνημάτων, όλες οι ομιλούμενες γλώσσες θεωρούνται υποσύνολα ενός υπερσυνόλου φωνημάτων που ονομάζεται διεθνές φωνητικό αλφάβητο (international phonetic alphabet – IPA) (IPA,1999).

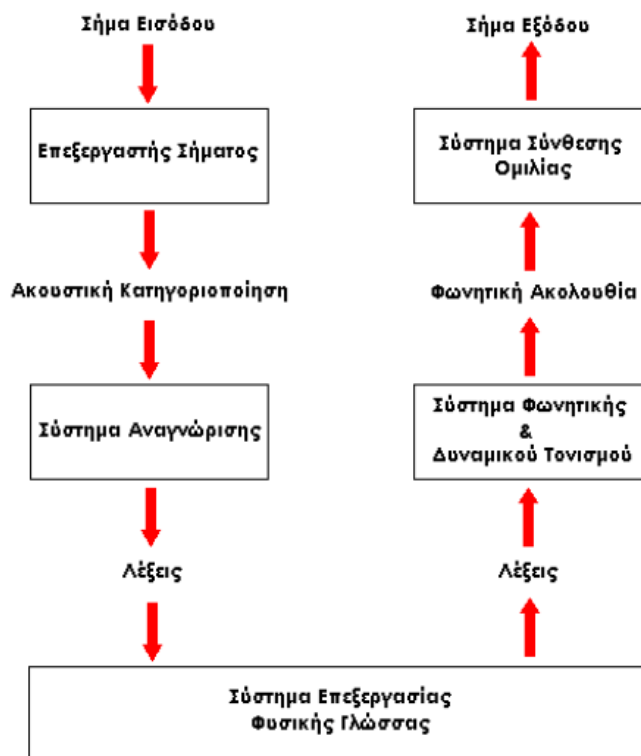


Σύστημα αμφίδρομης φωνητικής απόκρισης - ΑΦΑ, (Interactive Voice Response – IVR)

Λόγω λοιπόν του ότι η ομιλία ανέκαθεν υπήρξε ο βασικός τρόπος επικοινωνίας μεταξύ των ανθρώπων, εκτενής έρευνα έχει πραγματοποιηθεί για την κατανόηση και παραγωγή της από μηχανές. Η κατανόηση από της μηχανές αφορά το πρόβλημα της αναγνώρισης ομιλίας (speech recognition) ενώ το πρόβλημα της παραγωγής καλείται σύνθεση ομιλίας (speech synthesis). Ένα σύστημα το οποίο μπορεί να συνδυάζει και τις δύο τεχνολογίες καλείται σύστημα αμφίδρομης φωνητικής απόκρισης (ΑΦΑ) και το διάγραμμα λειτουργίας του παρουσιάζεται στην εικόνα.

Υποκατηγορίες τεχνολογίας ομιλίας

Η τεχνολογία ομιλίας προσπαθεί να αντιγράψει ή ακόμα και να βελτιώσει τις διαδικασίες που λαμβάνουν χώρα κατά την διάρκεια της προφορικής επικοινωνίας μεταξύ δύο ανθρώπων. Η διαδικασία επικοινωνίας ανθρώπου-μηχανής παρουσιάζεται στην εικόνα, και αποτελεί την βασική δομή ενός τυπικού διαλογικού συστήματος.



Μπλοκ διάγραμμα ενός τυπικού διαλογικού συστήματος.

Όπως φαίνεται στην εικόνα, το σήμα ομιλίας από τον άνθρωπο-χρήστη επεξεργάζεται και οδηγείται σε ένα σύστημα αυτόματης αναγνώρισης ομιλίας

(automatic speech recognizer – ASR), όπου το μετατρέπει στην ακολουθία των αναγνωρισμένων λέξεων. Στη συνέχεια ένα σύστημα επεξεργασίας φυσικής γλώσσας, το οποίο αποτελεί και την καρδιά του διαλογικού συστήματος θα επεξεργαστεί τα δεδομένα εισόδου. Η επεξεργασία αυτή αποτελείται από την μετατροπή από κείμενο σε νοήματα, με ένα σύστημα αντίληψης φυσικής ομιλίας (natural language understanding – NLU), την επεξεργασία των νοημάτων και την δημιουργία νέων νοημάτων, δηλαδή της κατάλληλης απόκρισης του συστήματος, την μετατροπή των νέων αυτών νοημάτων σε φυσική ομιλία με ένα σύστημα δημιουργίας φυσικής ομιλίας (natural language generation – NLG), και τέλος, την μετατροπή του κειμένου που αντιστοιχεί στην απόκριση του συστήματος προς τον άνθρωπο-χρήστη σε συνθετική ομιλία, με ένα σύστημα μετατροπής από κείμενο σε ομιλία (text-to-speech – TTS).

Είναι προφανές ότι η τεχνολογία ομιλίας περιλαμβάνει ένα πλήθος από συστήματα και υπο-συστήματα, για την πλήρη εκτέλεση της επικοινωνίας ανθρώπου-μηχανής. Η τεχνολογία ομιλίας μπορεί να χωριστεί σε διάφορες κατηγορίες και υπο-κατηγορίες όπως φαίνεται στην εικόνα.



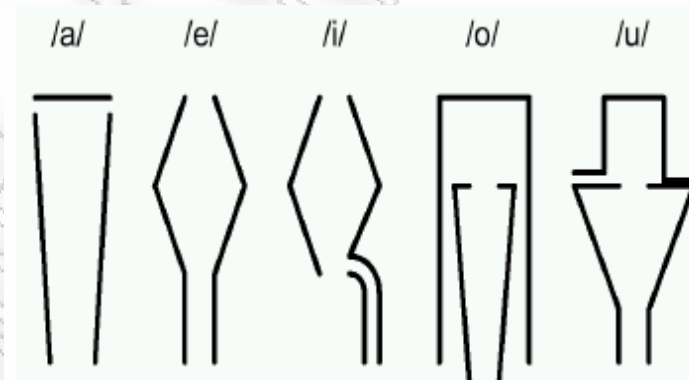
Κατηγορίες της τεχνολογίας ομιλίας.

Σύνθεση φωνής

Ιστορική αναδρομή

Η δημιουργία μιας μηχανής ικανής να παράγει τεχνητή ομιλία, ήταν το όνειρο του ανθρώπου για αιώνες. Οι πρώτες προσπάθειες για δημιουργία συνθετικής ομιλίας έγιναν πριν δύο αιώνες.

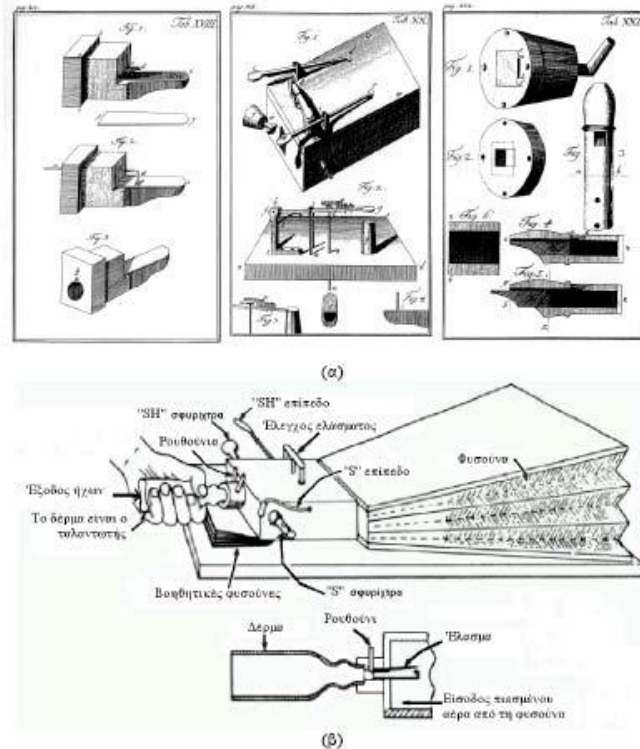
Στην Αγία Πετρούπολη το 1779 ο Ρώσος καθηγητής Christian Kratzenstein εξήγησε τις διαφορές μεταξύ 5 φωνηέντων «/a/, /e/, /i/, /o/, /u/» και έφτιαξε μια συσκευή για να τα παράγει τεχνητά. Κατασκεύασε ακουστικούς ταλαντωτές, εικόνα, παρόμοιους με την ανθρώπινη φωνητική οδό τους οποίους τους ενεργοποιούσε με αέρα όπως τα πνευστά μουσικά όργανα.



Οι ακουστικοί ταλαντωτές του Christian Kratzenstein, 1779

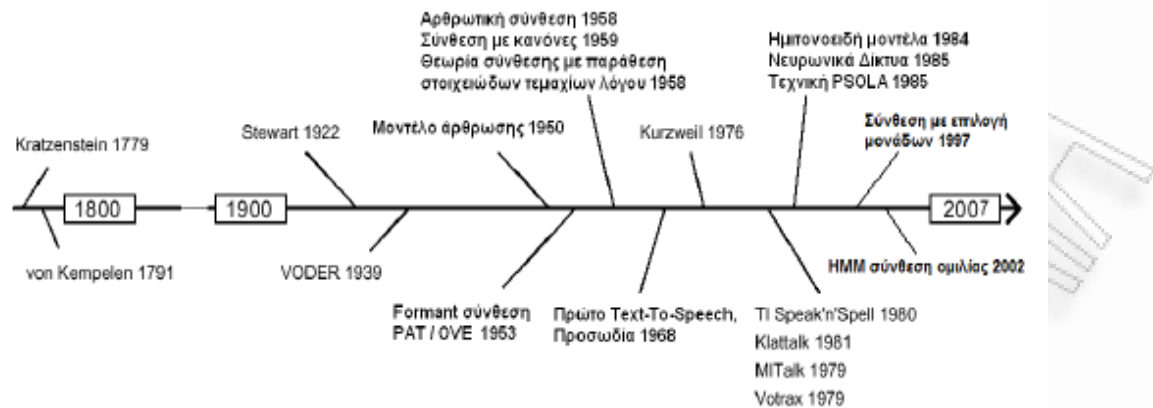
Λίγα χρόνια αργότερα στην Βιέννη το 1791 ο Wolfgang von Kempelen εισήγαγε την ακουστικο-μηχανική μηχανή φωνής η οποία μπορούσε να παράγει απλούς ήχους και μερικούς συνδυασμούς. Στην πραγματικότητα ο Kempelen ξεκίνησε την εργασία του πριν τον Kratzenstein, το 1769, και μετά από 20 χρόνια έρευνας εξέδωσε βιβλίο που περιγράφει τις μελέτες του στην παραγωγή ανθρώπινης φωνής και τα πειράματά του με το μηχάνημα παραγωγής φωνής. Το μηχάνημα του von Kempelen μοντελοποιούσε τους πνεύμονες με μια φυσούνα, τις φωνητικές χορδές με ένα παλλόμενο ελασμα και τη φωνητική οδό με ένα δερμάτινο σωλήνα, εικόνα. Αλλάζοντας το σχήμα του δερμάτινου σωλήνα μπορούσε να παράγει τα φωνηέντα. Η παραγωγή συμφώνων γινόταν ελέγχοντας με τα δάχτυλα τέσσερα διαφορετικά περάσματα αέρα. Η μελέτη του δημιούργησε την θεωρία ότι η φωνητική οδός είναι το κυριότερο

μέρος της συνάρθρωσης. Μέχρι τότε θεωρείτο μόνο ο λάρυγγας ως το κέντρο της παραγωγής φωνής. Στα μέσα του 1800 ο Charles Wheatstone επέκτεινε την μηχανή του Kempelen η οποία τώρα μπορούσε να παράγει πιο πολύπλοκα σύμφωνα, μπορούσε δηλαδή να παράγει συνδυασμούς ήχων αλλά ακόμη και ολόκληρες λέξεις.



Η μηχανές των (α) Wolfgang von Kempelen και (β) Charles Wheatstone

Η σχέση ενός συγκεκριμένου φωνήεντος με τη γεωμετρία της φωνητικής οδού μελετήθηκε από τον Willis το 1838. Έκανε συνθέσεις διαφορετικών φωνηέντων χρησιμοποιώντας ταλαντωτές που έμοιαζαν με σωληνοειδή πνευστά όργανα. Παρατήρησε ότι η ποιότητα των φωνηέντων εξαρτάται μόνο από το μήκος των σωλήνων και όχι από τη διάμετρο. Η έρευνα και τα πειράματα με μηχανικά και ημι-ηλεκτρικά συστήματα συνεχίστηκε μέχρι το 1960 χωρίς αξιοσημείωτα αποτελέσματα. Διάσημος επιστήμονας που ασχολήθηκε με ημι-ηλεκτρικά συστήματα παραγωγής φωνής ήταν και ο Herman von Helmholtz. Ο πρώτος ηλεκτρικός συνθέτης δημιουργήθηκε από τον Stewart το 1922. Σαν διέγερση είχε ένα βομβητή και δύο κυκλώματα συντονισμού τα οποία μοντελοποιούσαν τους ακουστικούς συντονισμούς της φωνητικής οδού. Το μηχάνημα μπορούσε να παράγει μεμονωμένα φωνήεντα με τα δυο πρώτα formants, αλλά όχι σύμφωνα ή ολοκληρωμένες εκφωνήσεις. Παρόμοια δουλειά έκανε και ο Wagner βάζοντας 4 ηλεκτρικούς συντονιστές παράλληλα. Το 1932 οι Ιάπωνες ερευνητές Obata και Teshima ανακάλυψαν και τρίτο formant στα φωνήεντα.



Ιστορικό χρονοδιάγραμμα της σύνθεσης φωνής

Η πρώτη συσκευή που θεωρήθηκε ως συνθέτης φωνής ήταν ο VODER (Voice Operating Demonstrator) από τον Homer Dudley που παρουσιάστηκε στην διεθνή έκθεση της Νέας Υόρκης το 1939. Ο VODER δημιουργήθηκε εμπνευσμένος από τον VOCODER (Voice Coder) που αναπτύχθηκε στα Bell Laboratories στα μέσα του '30. Ο VODER ήταν μια μηχανή η οποία ανέλυε την φωνή σε ακουστικές παραμέτρους και κατόπιν οδηγούσε τα αποτελέσματα σε ένα συνθέτη ο οποίος επαναδημιουργούσε μια προσέγγιση του αρχικού σήματος. Ο VODER δεχόταν χειροκίνητα τις παραμέτρους για τη σύνθεση και μπορούσε να παράγει προτάσεις. Ο έλεγχος της F0 γινόταν με ένα πεντάλ! Η ποιότητα της φωνής δεν ήταν καλή αλλά αυτή η μηχανή αποτέλεσε την αρχή για παραγωγή τεχνητής ομιλίας. Οι επιστήμονες μετά τον VODER άρχισαν να ενδιαφέρονται περισσότερο για τη σύνθεση φωνής.

Ο πρώτος συνθέτης με κανόνες (formant) ήταν ο PAT (Parametric Artificial Talker) που παρουσιάστηκε από τον Walter Lawrence το 1953. Ο συνθέτης PAT είχε τρεις συντονιστές συνδεδεμένους παράλληλα (για κάθε αρμονική). Η είσοδος του ήταν βόμβος ή θόρυβος. Με τη χρησιμοποίηση μιας κινούμενης διαφάνειας μετατρέπονταν οι ζωγραφισμένες πατέντες της σε έξι συναρτήσεις χρόνου, οι οποίες διαμόρφωναν τις τρεις αρμονικές συχνότητες, την ένταση, την F0 και τα επίπεδα θορύβου. Την ίδια περίοδο ο Gunnar παρουσίασε τον πρώτο τύπου "με κανόνες" συνθέτη (OVE - Orator Verbis Electricis) με συντονιστές σε σειρά. Δέκα χρόνια αργότερα, το 1962 παρουσιάστηκε ο OVE II από τους Fant και Martony ενώ στη συνέχεια οι OVE III και GLOVE στο Kungliga Tekniska Hogskolan της Σουηδίας. Ο πρώτος συνθέτης συνάρθρωσης (articulatory synthesizer) παρουσιάστηκε το 1958 από τον George Rosen στο Massachusetts Institute of Technology (MIT). Τα σήματα ελέγχου του συνθέτη DAVO (Dynamic Analog of the VOcal tract) περιείχονταν σε μια ηχογράφιση. Τα σήματα αυτά εισάγονταν χειρονακτικά.

Το 1979 οι Allen, Hunnicutt και Klatt παρουσίασαν το MITalk (Allen *et al.*, 1987), ένα εργαστηριακό σύστημα σύνθεσης φωνής από κείμενο το οποίο αναπτύχθηκε στο Massachusetts Institute of Technology. Αργότερα ο D. Klatt παρουσίασε το Klattalk (Klatt, 1982), μια επέκταση του MITalk. Η τεχνολογία που χρησιμοποιήθηκε στο MITalk και Klattalk αποτέλεσε την βάση για την ανάπτυξη συνθετών όπως ο DECtalk και ο Prose-2000. Την περίοδο 1970-1980 παρουσιάστηκαν αρκετά εμπορικά συστήματα μετατροπής κειμένου σε ομιλία. Το πρώτο ολοκληρωμένο κύκλωμα για σύνθεση φωνής ήταν το Votrax chip το οποίο αποτελούνταν από έναν συνθέτη με κανόνες και απλά χαμηλοδιαβατά φίλτρα.

Τεχνολογία και Συστήματα Μετατροπής Κειμένου σε Ομιλία

Μια εφαρμογή στην οποία έχει δοθεί μεγάλη έμφαση σε ερευνητικό επίπεδο σχετικά με την παραγωγή ομιλίας από μηχανή, είναι τα συστήματα μετατροπής κειμένου σε ομιλία (ΜΚΣΟ ή συστήματα Text-To-Speech, TTS).

Τα συστήματα TTS είναι αντικείμενο συστηματικής και συνεχόμενης έρευνας και αυτό έχει ως αποτέλεσμα τη βελτίωση της ποιότητας τους καθώς και τη σημαντική μείωση του κόστους τους. Τα συστήματα αυτά βρίσκουν ήδη εφαρμογή στις επικοινωνίες, όπου μηνύματα βασισμένα σε κείμενο, όπως τα email ή τα fax, ή πληροφορίες που συνδυάζουν κείμενο και εικόνα, όπως οι ιστοσελίδες, αποδίδονται φωνητικά. Υπάρχει επίσης, το πρόγραμμα VoiceXML που παρέχει διαδραστικές υπηρεσίες ομιλίας διαμέσου του διαδικτύου. Γενικότερα, τα συστήματα TTS καλύπτουν την ανάγκη για φωνητική απόδοση πληροφοριών, όλων των ειδών, που βρίσκονται αποθηκευμένες στις βάσεις δεδομένων, όπως για παράδειγμα τηλεφωνικοί αριθμοί, διευθύνσεις ή πληροφορίες πλοήγησης αυτοκινήτων. Ακόμη, βρίσκουν εφαρμογή στις αυτόματες υπηρεσίες πληροφόρησης, που παρέχουν για παράδειγμα πληροφορίες για τοποθεσίες και μενού εστιατορίων ή πραγματοποιούν αυτόματη εκφώνηση δελτίων καιρού και ειδήσεων μέσω τηλεφώνου. Δίνουν επίσης λύσεις και σε πιο κλασικές ανάγκες του ανθρώπου, όπως είναι η αυτόματη ανάγνωση εντύπων από μια μηχανή (ομιλούντα βιβλία), που εξυπηρετεί ιδιαίτερα άτομα με προβλήματα όρασης.

Επιπλέον, η έρευνα στρέφεται και στην δημιουργία συστημάτων που αποσκοπούν στην μετατροπή εννοιών σε ομιλία (Concept to speech systems) όπου στην περίπτωση αυτή ο συνθέτης δέχεται ως είσοδο την έξοδο ενός συστήματος δημιουργίας φυσικής γλώσσας (Theune *et al.*, 2001). Το πλεονέκτημα που έχουν τέτοιου είδους συστήματα είναι ότι παρέχουν στον συνθέτη περισσότερη πληροφορία για την δημιουργία του τεχνητού λόγου. Έτσι μαζί με τις λέξεις που θα εκφωνηθούν, μπορεί να δοθεί συντακτική, σημασιολογική και προσωδιακή πληροφορία που αλλιώς ο συνθέτης θα έπρεπε να εξάγει ή να προβλέψει ο ίδιος.

Οι πιθανές εφαρμογές των υψηλής ποιότητας TTS συστημάτων είναι πράγματι πολυάριθμες. Παρακάτω παρουσιάζονται μερικά παραδείγματα:

Τηλεπικοινωνιακές υπηρεσίες. Τα TTS συστήματα καθιστούν πιθανό να έχει κανείς πρόσβαση σε πληροφορίες με μορφή κειμένου από το τηλέφωνο. Οι πληροφορίες σε μορφή κειμένου μπορεί να είναι από απλά μηνύματα, όπως οι τοπικές πολιτιστικές εκδηλώσεις (κινηματογράφοι, θέατρα...), στις τεράστιες βάσεις δεδομένων που μπορούν μετά βίας να διαβαστούν και να αποθηκευτούν ως ψηφιοποιημένη ομιλία. Οι ερωτήσεις ανάκτησης σε τέτοια συστήματα πληροφοριών θα μπορούσαν να θεθούν μέσω της φωνής του χρήστη (με τη βοήθεια ενός λεκτικού συστήματος αναγνώρισης), ή μέσω του τηλεφωνικού πληκτρολογίου (με τα συστήματα *DTMF*). Μερικές τηλεφωνικές υπηρεσίες που μπορούν να υλοποιηθούν με τη χρήση TTS συστημάτων είναι: Ηχητική αναγνώριση κλήσεων (μπορεί κανείς να ακούσει το όνομα του καλούντος πριν συνδεθεί η κλήση και να την αποφύγει), ολοκληρωμένο μήνυμα (μπορεί κανείς να ακούσει το ηλεκτρονικό ταχυδρομείο ή το *facsimiles* αυτόματα από το τηλέφωνο). Αυτές οι εφαρμογές έχουν αποδειχθεί αποδεκτές, και ακόμα και δημοφιλείς, υπό τον όρο ότι η σαφήνεια των συνθετικών εκφράσεων είναι αρκετά υψηλή. Η φυσικότητα δεν ήταν ένα σημαντικό ζήτημα στις περισσότερες περιπτώσεις.

Γλωσσική εκπαίδευση. Η υψηλής ποιότητας TTS σύνθεση μπορεί να συνδυαστεί με ένα σύστημα εκμάθησης με βοήθεια υπολογιστή, και να παρέχει ένα χρήσιμο εργαλείο για να μάθει κανείς μια νέα γλώσσα. Κάτι που δεν έχει υλοποιηθεί πλήρως ακόμα, λαμβάνοντας υπόψη τις κρίσιμες απαιτήσεις τέτοιων σκοπών σε αντιδιαστολή με τη σχετικά κακή ποιότητα, που είναι διαθέσιμη στα εμπορικά συστήματα.

Βοήθεια στους ανθρώπους με ειδικές ανάγκες. Οι αναπηρίες φωνής δημιουργούνται από διανοητικές ή σωματικές διαταραχές. Οι μηχανές μπορούν να είναι μια ανεκτίμητη βοήθεια στην τελευταία περίπτωση: με τη βοήθεια ενός ειδικά σχεδιασμένου πληκτρολογίου και ενός γρήγορου προγράμματος σύνθεσης προτάσεων, η συνθετική ομιλία μπορεί να παραχθεί σε μερικά δευτερόλεπτα για να ξεπεράσει τέτοιου είδους εμπόδια. Όσοι άνθρωποι επίσης πάσχουν από προβλήματα όρασης μπορούν να ωφεληθούν από TTS συστήματα, τα οποία μπορούν να συνδεθούν με τα οπτικά συστήματα αναγνώρισης (OCR) και να προσφέρουν πρόσβαση σε γραπτές πληροφορίες.

Ομιλούντα βιβλία και παιχνίδια. Η αγορά παιχνιδιών έχει αγγιχτεί ήδη από τη λεκτική σύνθεση. Πολλά παιχνίδια ομιλίας έχουν εμφανιστεί, η κακή ποιότητα των οποίων αναπόφευκτα

σταματά την εκπαιδευτική φιλοδοξία τέτοιων προϊόντων. Η υψηλής ποιότητας σύνθεση όμως σε προσιτές τιμές μπορεί να ανατρέψει κάτι τέτοιο.

Φωνητικός έλεγχος. Σε μερικές περιπτώσεις, οι προφορικές πληροφορίες είναι αποδοτικότερες από τα γραπτά μηνύματα. Η φωνητική έκκληση είναι ισχυρότερη, ενώ η προσοχή μπορεί να εστιάσει και σε άλλες οπτικές πηγές πληροφοριών.

Πολυμέσα, επικοινωνία ανθρώπου-μηχανής. Μακροπρόθεσμα, η ανάπτυξη των υψηλής ποιότητας TTS συστημάτων είναι ένα απαραίτητο βήμα (όπως είναι η αύξηση των λεκτικών συστημάτων αναγνώρισης) προς τους πληρέστερους τρόπους επικοινωνίας μεταξύ των ατόμων και των υπολογιστών. Τα πολυμέσα είναι μια πρώτη αλλά ελπιδοφόρος κίνηση σε αυτήν την κατεύθυνση.

Βασική και εφαρμοσμένη έρευνα. Τα Text To Speech (TTS) συστήματα έχουν ένα πολύ ιδιαίτερο χαρακτηριστικό γνώρισμα που τα κάνει θαυμάσια εργαστηριακά εργαλεία για τους γλωσσολόγους: είναι πλήρως ελεγχόμενα, έτσι ώστε όταν επαναλαμβάνεται η ίδια διαδικασία να μπορεί να παρέχει τα ίδια αποτελέσματα (κάτι που δεν συμβαίνει με τους ανθρώπους). Ένας ιδιαίτερος τύπος TTS συστημάτων, που είναι βασισμένα σε μια περιγραφή του φωνητικού σήματος μέσω των ηχηρών συχνοτήτων του (formants) έχει επίσης χρησιμοποιηθεί εκτενώς από επιστήμονες στην μελέτη της ομιλίας σε σχέση με τους ακουστικούς κανόνες.

Ανάλυση Συστημάτων Μετατροπής Κειμένου σε Ομιλία

Τα συστήματα μετατροπής κειμένου-σε-ομιλία (ΜΚσΟ) δέχονται ως είσοδο κείμενο και παράγουν συνθετικό προφορικό λόγο, παρέχοντας με αυτόν τον τρόπο στους ανθρώπους πληροφορίες κειμένου μέσω φωνητικών μηνυμάτων. Το κείμενο μπορεί να εισάγεται απευθείας στον υπολογιστή από κάποιον χρήστη ή να σαρώνεται και στην συνέχεια να περνάει από ένα σύστημα οπτικής αναγνώρισης χαρακτήρων (Optical Character Recognition system – OCR). Στο παρακάτω σχήμα περιγράφεται το γενικό μοντέλο ενός συστήματος ΜΚσΟ.

Αρχικά το προς σύνθεση κείμενο εισέρχεται στο στάδιο της επεξεργασίας φυσικής γλώσσας όπου υφίσταται μορφολογική, λεξιλογική και γλωσσολογική ανάλυση. Η ανάλυση αυτή οδηγεί στην απόκτηση της απαραίτητης πληροφορίας για την δημιουργία των προσωδιακών προδιαγραφών του συνθετικού λόγου. Σαν επόμενο στάδιο έχοντας μια λεπτομερή φωνητική και γλωσσολογική αναπαράσταση του κειμένου εισόδου καθώς και την προσωδία των φωνημάτων προχωράμε στην ψηφιακή επεξεργασία του σήματος που θα δώσει σαν έξοδο την ομιλία.



Λειτουργικό διάγραμμα συστήματος ΜΚσΟ

Οι καρποί της συστηματικής και συνεχόμενης έρευνας στον τομέα της ΜΚσΟ έχει ως αποτέλεσμα τη βελτίωση της ποιότητας της συνθετικής ομιλίας πετυχαίνοντας ταυτόχρονα χαμηλό κόστος. Το γεγονός αυτό, σε συνδυασμό με τις ολοένα αυξανόμενες ανάγκες σε έναν κόσμο διακίνησης τεράστιου όγκου πληροφορίας, όπου πολλές από αυτές είναι πληροφορίες κειμένου, έδωσαν το κίνητρο για την ευρεία εξάπλωση των ΜΚσΟ εφαρμογών.

Υλοποίηση συστήματος ΜΚσΟ σε ευφυή πράκτορα

Για να μπορεί ο ευφυής πράκτορας να μετατρέπει το κείμενο που θέλει να «πεί» σε ήχο έχει γίνει μία υλοποίηση η οποία χρησιμοποιεί τεχνολογίες που αναλύονται σε αυτό το κεφάλαιο. Στα χαρακτηριστικά του πράκτορα υπάρχει ένα αντικείμενο τύπου VoiceGenerator το οποίο αναλαμβάνει αφού δημιουργηθεί το κείμενο σε φυσική γλώσσα από το σύστημα ELIZA να το

μετατρέψει σε ομιλία. Η κλάση VoiceGenerator στηρίζεται στο σύστημα FreeTTS και κάνει χρήση του Java Speech Api.

Java Speech Api

Το Java Speech API είναι ένα από τα Java Media APIs, ένα πακέτο από διεπαφές λογισμικού που παρέχουν cross-platform πρόσβαση σε ήχο, βίντεο και άλλου είδους αναπαραγωγή πολυμέσων, δισδιάστατη και τρισδιάστατη γραφική παράσταση, animation, τηλεφωνία, προηγμένη απεικόνιση, και άλλα. Το Java Speech API, σε συνδυασμό με τα άλλα Java Media APIs, επιτρέπουν στους προγραμματιστές να εμπλουτίσουν τις εφαρμογές και τα applets της Java με πλούσιες δυνατότητες πολυμέσων και επικοινωνίας που ικανοποιούν τις προσδοκίες των σημερινών χρηστών, και μπορούν να ενισχύσουν την διαπροσωπική επικοινωνία.

Το Java Speech API εκμεταλλεύεται τις δυνατότητες άλλων Java APIs. Τα χαρακτηριστικά γνωρίσματα διεθνοποίησης της γλώσσας προγραμματισμού της Java συν τη χρήση του συνόλου χαρακτήρων Unicode απλοποιούν την ανάπτυξη πολύγλωσσων εφαρμογών ομιλίας. Οι κλάσεις και οι διεπαφές του Java Speech API ακολουθούν τα πρότυπα σχεδιασμού του JavaBeans. Τέλος, τα γεγονότα του Java Speech API ενωποιούνται με τους μηχανισμούς γεγονότων των AWT, JavaBeans και Java Foundation Classes (JFC).

Για να χρησιμοποιήσει το Java Speech API, ένας χρήστης πρέπει να έχει διαθέσιμα ορισμένα ελάχιστα στοιχεία λογισμικού και υλικού. Στη συνέχεια περιγράφεται ένα ευρύ δείγμα των απαιτήσεων. Οι μεμονωμένες απαιτήσεις των συνθετών ομιλίας και των συστημάτων αναγνώρισης ομιλίας μπορούν να ποικίλουν αρκετά και οι χρήστες πρέπει να ελέγξουν προσεκτικά τις απαιτήσεις των προϊόντων.

Λογισμικό ομιλίας: Απαιτείται ένα σύστημα αναγνώρισης ομιλίας συμβατό με το Jsapi ή ένας συνθέτης.

Απαιτήσεις συστημάτων: τα περισσότερα συστήματα αναγνώρισης ομιλίας επιτραπέζιων υπολογιστών και μερικοί συνθέτες ομιλίας απαιτούν τους σχετικά ισχυρούς υπολογιστές για να τρέξουν αποτελεσματικά. Συνεπώς ο χρήστης θα πρέπει να ελέγξει τις ελάχιστες και συνιστώμενες απαιτήσεις για την ΚΜΕ, τη μνήμη και τη χωρητικότητα δίσκου κατά τον αγορά ενός προϊόντος ομιλίας.

Ακουστικό υλικό: Οι συνθέτες ομιλίας απαιτούν έξοδο ήχου. Τα συστήματα αναγνώρισης ομιλίας απαιτούν είσοδο ήχου. Οι περισσότεροι επιτραπέζιοι και φορητοί υπολογιστές που πωλούνται στις μέρες μας έχουν ικανοποιητική υποστήριξη ήχου. Τα περισσότερα συστήματα υπαγόρευσης αποδίδουν καλύτερα με κάρτες ήχου καλής ποιότητας.

Μικρόφωνο: Τα συστήματα αναγνώρισης ομιλίας επιτραπέζιων υπολογιστών λαμβάνουν την είσοδο του ήχου μέσω ενός μικροφώνου. Μερικά συστήματα αναγνώρισης, ειδικά τα συστήματα υπαγόρευσης, είναι ευαίσθητα ως προς το μικρόφωνο και τα περισσότερα προϊόντα αναγνώρισης συστήνουν συγκεκριμένα μικρόφωνα. Τα μικρόφωνα κεφαλής παρέχουν συνήθως την καλύτερη απόδοση, ειδικά σε θορυβώδη περιβάλλοντα. Τα επιτραπέζια μικρόφωνα μπορούν να χρησιμοποιηθούν σε ορισμένα μόνο περιβάλλοντα για ορισμένες εφαρμογές.

FreeTTS

Το FreeTTS είναι ανοικτού κώδικα σύστημα ερευνητικό σύστημα ΜΚσΟ γραμμένο εξ ολοκλήρου στη γλώσσα προγραμματισμού της Java. Αναπτύχθηκε από την ομάδα Ενσωμάτωσης Ομιλίας των εργαστηρίων της εταιρείας Sun Microsystems και είναι βασισμένο σε δύο συστήματα για φωνητική σύνθεση:

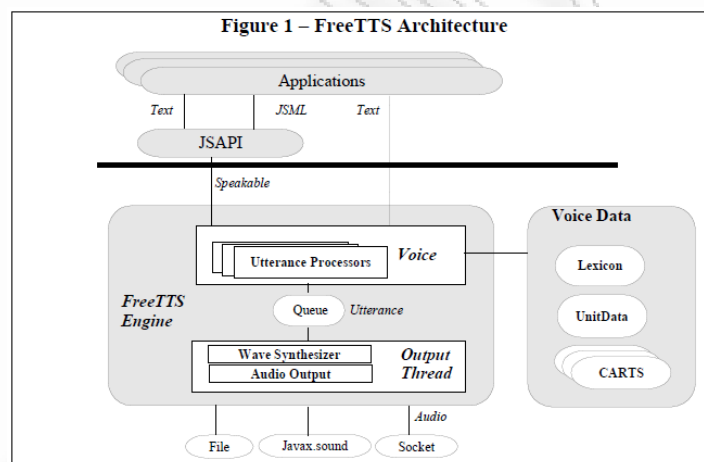
Στο σύστημα Festival που αναπτύχθηκε στο πανεπιστήμιο του Εδιμβούργου. Το σύστημα αυτό χρηματοδοτήθηκε από την εταιρεία Sun Microsystems και αναπτύχθηκε μέσω συνεργατικής μελέτης. Είναι ένα ευέλικτο ανοικτού κώδικα ερευνητικό σύστημα ΜΚσΟ γραμμένο στην γλώσσα προγραμματισμού C++.

Στο σύστημα Flite που αναπτύχθηκε στο πανεπιστήμιο Carnegie και βασίζεται στο σύστημα Festival. Είναι γραμμένο στην γλώσσα προγραμματισμού C και δίνει έμφαση στο μέγεθος και στην απόδοση στις ενσωματωμένες πλατφόρμες (embedded platforms).

Οι απαιτήσεις μεγέθους και απόδοσης του συστήματος Flite, εντούτοις, μείωσαν δραστικά την ευελιξία του. Συνεπώς προκειμένου να χρησιμοποιήσουν το βέλτιστο και των δύο κόσμων, οι δημιουργοί του συστήματος FreeTTS βάσισαν τους αλγορίθμους του συστήματος FreeTTS στο σύστημα Flite, αλλά την αρχιτεκτονική του στο σύστημα Festival. . Αξίζει να σημειωθεί ότι το σύστημα περιλαμβάνει μια μηχανή φωνητικής σύνθεσης που υποστηρίζει έναν ορισμένο αριθμό φωνών (γυναικείων και ανδρικών) σε διάφορες συχνότητες. (Walket et all, 2002)

Για να συνθέσει την ομιλία, το σύστημα FreeTTS διασπά το κείμενο εισαγωγής σε σύνολα φωνημάτων και έπειτα τα μετατρέπει σε ευδιάκριτη ομιλία, μέσω της εκτέλεσης διαδοχικών διαδικασιών στο κείμενο εισαγωγής. Το FreeTTS αποθηκεύει τα συσσωρευτικά αποτελέσματα κάθε λειτουργίας σε μια δομή έκφρασης που κρατά την πλήρη ανάλυση του κειμένου. Το σχήμα παρακάτω παρουσιάζει την γενική αρχιτεκτονική του συστήματος FreeTTS.

Ο πυρήνας του συστήματος είναι μια μηχανή που περιέχει μια φωνή και ένα νήμα παραγωγής. Η φωνή αποτελείται από ένα σύνολο επεξεργαστών έκφρασης που δημιουργούν, επεξεργάζονται, και σχολιάζουν μια δομή έκφρασης. Συνδεδεμένο με την φωνή είναι ένα σύνολο στοιχείων που χρησιμοποιείται από κάθε έναν από τους επεξεργαστές έκφρασης. Το νήμα παραγωγής είναι υπεύθυνο για δύο ενέργειες: την σύνθεση μιας έκφρασης στα ακουστικά στοιχεία και έπειτα την κατεύθυνση αυτού του στοιχείου στον κατάλληλο μηχανισμό ακουστικής αναπαραγωγής ήχου.



Οι δομές φωνής και έκφρασης

Η καρδιά του συστήματος FreeTTS βρίσκεται στις δομές της φωνής και της έκφρασης. Η φωνή διατηρεί σφαιρικές πληροφορίες για τη διαδικασία της σύνθεσης: το σύνολο τοπικής προσαρμογής, το λεξικό προφοράς, την βάση δεδομένων μονάδων, και τον συνθέτη κυμάτων ήχου. Η φωνή διατηρεί επίσης το σύνολο των επεξεργαστών έκφρασης που χρησιμοποιείται για να δημιουργηθεί και να σχολιαστεί η δομή έκφρασης.

Η δομή έκφρασης είναι ένα προσωρινό αντικείμενο που η φωνή το δημιουργεί για κάθε ακουστικό κύμα που παράγει. Η φωνή αρχικοποιεί τη δομή έκφρασης με το κείμενο εισαγωγής και έπειτα περνάει την δομή έκφρασης σε ένα σύνολο συσχετιστικών επεξεργαστών έκφρασης. Μόλις επεξεργαστεί το κείμενο εισαγωγής (π.χ., στάλθηκε σε μια ακουστική συσκευή παραγωγής), η φωνή απορρίπτει τη δομή έκφρασης.

Κάθε επεξεργαστής έκφρασης προσθέτει πρόσθετα στοιχεία στη δομή έκφρασης με ιεραρχική και σχετική μέθοδο. Παραδείγματος χάριν, ένας επεξεργαστής έκφρασης δημιουργεί μια σχέση στη δομή έκφρασης που αποτελείται από τα στοιχεία που ενώνουν τις λέξεις του κειμένου εισαγωγής. Ένας άλλος επεξεργαστής έκφρασης δημιουργεί μια σχέση που αποτελείται από αντικείμενα που περιγράφουν συλλαβές για τις λέξεις, με κάθε στοιχείο

συλλαβών να δείχνει πίσω στη μεμονωμένη λέξη η οποία δημιουργήθηκε από άλλον επεξεργαστή έκφρασης.

Με την οργάνωση της δομής έκφρασης μέσω της χρήσης σχέσεων, οι επεξεργαστές έκφρασης μπορούν να εκτελέσουν περίπλοκες ερωτήσεις συσχέτισης στη δομή έκφρασης.

Για παράδειγμα ένα κείμενο ερώτησης θα μπορούσε να είναι το εξής: "R:SylStructure.parent.parent.word_numsyls."

Διαβάζοντας από δεξιά προς αριστερά αυτό σημαίνει "βρες τον αριθμό των συλλαβών στη λέξη που ο γονέας του γονέα της σχέσης συλλαβών για το αντικείμενο που μας ενδιαφέρει. Αυτοί οι τύποι ερωτήσεων χρησιμοποιούνται σε όλο το σύστημα FreeTTS από τους διάφορους επεξεργαστές έκφρασης.

Υπάρχουν διάφορα βήματα στη διαδικασία της σύνθεσης. Τα βασικά βήματα της λειτουργίας της σύνθεσης του συστήματος FreeTTS απεικονίζονται στο σχήμα παρακάτω. Πολλά από αυτά τα βήματα πρέπει να υποστούν τοπική προσαρμογή και προσαρμογή στον τύπο της σύνθεσης που χρησιμοποιείται. Μια χαρακτηριστική φωνή του συστήματος FreeTTS θα εκτελέσει τα ακόλουθα βήματα για να μετατρέψει το γραπτό κείμενο σε ομιλία:

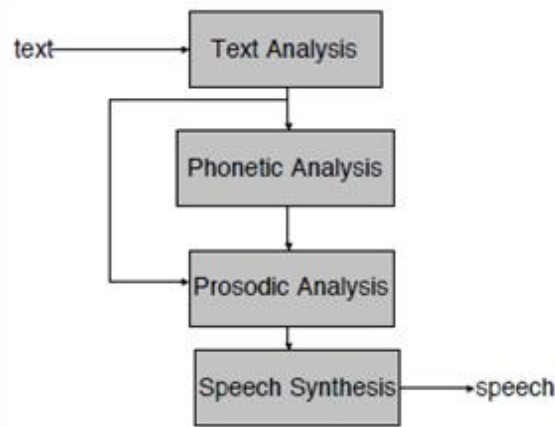
Κανονικοποίηση Κειμένων – Εκτελείται μέσω ενός επεξεργαστή έκφρασης που μετατρέπει το κείμενο εισόδου σε μια ροή λέξεων. Παραδείγματος χάριν, το κείμενο "ο κ. Παπαδόπουλος διαμένει στην οδό Υμηττού 33" θα μετατρέπεται στο κείμενο "ο κ. Παπαδόπουλος διαμένει στην οδό Υμηττού τριάντα τρία". Η διαδικασία κανονικοποίησης κειμένου εξετάζει μια ευρεία ποικιλία περιπτώσεων συμπεριλαμβανομένων αριθμών, ημερομηνιών, στοιχείων χρόνου, τίτλων και ονομάτων τοποθεσιών.

Γλωσσική ανάλυση – Εκτελείται μέσω ενός επεξεργαστή έκφρασης που καθορίζει σημασιολογικές πληροφορίες όπως οι πληροφορίες διατύπωσης και μέρους του λόγου.

Λεξικολογική ανάλυση – Εκτελείται μέσω ενός επεξεργαστή έκφρασης που καθορίζει την προφορά, τον προσδιορισμό των συλλαβών, και τον τονισμό της κάθε λέξης της έκφρασης. Το σύστημα FreeTTS θα χρησιμοποιήσει ένα λεξικό για να καθορίσει αυτές τις πληροφορίες. Εάν εντούτοις μια λέξη δεν είναι μέσα στο λεξικό, το σύστημα απευθύνεται σε ένα σύνολο περίπλοκων κανόνων γραμμάτων σε ήχο.

Παραγωγή Προσωδίας – Εκτελείται μέσω ενός επεξεργαστή έκφρασης που καθορίζει τις παραμέτρους των παύσεων, των θεμελιωδών συχνότητων (pitches), της διάρκειας, του τόνου, της έντασης, και του εύρους. Αυτοί οι επεξεργαστές θα χρησιμοποιήσουν ιδιαίτερα δέντρα ταξινόμησης και κατηγοριοποίησης (CARTS) για να παράγουν της πληροφορίες της προσωδίας.

Σύνθεση Ομιλίας – παράγει ακουστικά δεδομένα, τυπικά μέσω της σύνδεσης των λεκτικών μονάδων με βάση τα δίφωνα ή άλλες μονάδες της ομιλίας. Η διαδικασία της σύνθεσης μπορεί να κάνει ιδιαίτερα εντατική χρήση της μνήμης δεδομένου ότι περιλαμβάνει έναν μεγάλο αριθμό διαδικασιών κινητής υποδιαστολής. Όταν διασπάται σε αυτά τα ξεχωριστά βήματα, η διαδικασία σύνθεσης είναι σχετικά απλή. Η διαδικασία, εντούτοις, ωθεί τα όρια απόδοσης σε δύο διαστάσεις. Κατ' αρχάς, υπάρχουν μεγάλα σύνολα δεδομένων για να συσχετιστούν με το λεξικό και την βάση δεδομένων μονάδων, αποτελώντας το μέγιστο πλήθος δεδομένων. Δεύτερον, εμπλέκεται ένας σημαντικός αριθμός υπολογισμών κινητής υποδιαστολής. Συνεπώς όπως ήταν αναμενόμενο, η επεξεργασία σύνθεσης ήχου διαρκεί περισσότερο από τα άλλα βήματα.



Βήματα της Λειτουργίας της Σύνθεσης του συστήματος FreeTTS

VoiceGenerator

Βασισμένη στο FreeTTS και στο Java Speech Api είναι η κλάση VoiceGenerator που έχει δημιουργηθεί. Αντικείμενο της κλάσης αυτής υλοποιείται σε κάθε πράκτορα τη στιγμή της δημιουργίας του. Η κλάση αυτή ακολουθεί το design patten facade ως προς την κλάση VoiceManager που παρέχει το σύστημα FreeTTS. Το design pattern facade χρησιμοποιείται για να παρέχει ενός υψηλού επιπέδου interface που κάνει το υποσύστημα πιο εύκολο στη χρήση και βοηθά στη δημιουργία ενός εντοπιημένου interface για μία ομάδα interfaces του υποσυστήματος.

Κατά τη δημιουργία του αντικείμενου VoiceGenerator(κλήση στον constructor του) ορίζονται και οι παράμετροι της φωνής που θα έχει ο πράκτορας. Οι παράμετροι αυτοί φαίνονται στον παρακάτω πίνακα.

Constructor input	Type	Description
Pitch	Float	Ορίζουμε τη συχνότητα της φωνής του agent
Pitchshift	Float	Ορίζουμε τις μεταβολές στη συχνότητα της φωνής (προσωδία)
Name	String	Ορίζουμε ένα όνομα από τις default φωνές που παρέχονται με το FreeTTS (π.χ. "Kevin", "kevin16" κλπ)
Style	String	Ορίζουμε ένα στυλ φωνής (π.χ. "business", "casual", "robotic", "breathy")

Κατά τη στιγμή της δημιουργίας του πράκτορα έχουμε και το αντικείμενο για τη δημιουργία της ομιλίας του. Το αντικείμενο αυτό κάνει expose μία και μόνο μέθοδο με το όνομα speak. Δέχεται σαν είσοδο ένα string και σαν αποτέλεσμα κάνει ανάγνωση (στην Αγγλική γλώσσα) του string που δόθηκε.

Η κλάση VoiceGenerator δημιουργήθηκε έτσι ώστε να κάνει expose τα λιγότερα δυνατά στο περιβάλλον και να αποτελεί ένα εύκολο τρόπο με κλήση μίας και μόνο απλής μεθόδου ο πράκτορας να μπορεί να μετατρέψει σε ομιλία το κείμενο που έχει παραχθεί από τη γεννήτρια φυσικής γλώσσας.

Φυσική γλώσσα

Επεξεργασία Φυσικής Γλώσσας

Η επιστημονική περιοχή που φέρει αυτόν τον τίτλο σε αδρές γραμμές καλύπτει τις ακόλουθες κατευθύνσεις.

Σχεδιασμός και υλοποίηση υπολογιστικών μοντέλων της φυσικής γλώσσας, και πιο συγκεκριμένα: την αναγνώριση ή κατανόηση της φυσικής γλώσσας από υπολογιστή, και την παραγωγή φυσικής γλώσσας ή σύνθεση.

Οι εφαρμογές της περιοχής αυτής αγκαλιάζουν τον διάλογο με τον υπολογιστή την μηχανική μετάφραση (από γλώσσα σε γλώσσα, από γλώσσα σε βάση δεδομένων και αντιστρόφως κλπ), και την αναδίφηση (browsing) ή την διήθηση (filtering) κειμένων φυσικής γλώσσας από έναν πράκτορα ή παράγοντα (agent).

Ο όρος Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing) είναι παραδοσιακός και προέρχεται από τον ομώνυμο κλάδο της Τεχνητής Νόησης (Artificial Intelligence). Σήμερα αντιστοιχεί στην εφαρμοσμένη Υπολογιστική Γλωσσολογία.

Ιστορική αναδρομή

Με τους πρώτους υπολογιστές (1950 - 1960) ο χειρισμός αριθμών και στη συνέχεια λέξεων έδωσε τη δυνατότητα μιας πρώτης 'ηλεκτρονικής' παράστασης και επεξεργασίας γλωσσικής γνώσης. Αργότερα, η μετρική και η στατιστική του κειμένου επέτρεψαν την καλλιέργεια της υφολογίας (style analysis) και της εύρεσης της ταυτότητας του συγγραφέα ή συντάκτη ενός κειμένου. Ακολούθησε η αυτόματη δημιουργία δεικτών (indices) και η εύρεση των λέξεων μέσα στα συμφραζόμενά τους (keywords in context: concordance lists). Σήμερα η περιοχή αυτή εξακολουθεί να καλλιεργείται και ονομάζεται Literary and Linguistic Computing. (Την αποδίδουμε ελεύθερα ως Υπολογιστική Γραμματεία και Γλωσσολογία). Η μηχανική (και όχι αυτόματη) μετάφραση (machine translation) έδωσε τα πρώτα δείγματα στα μέσα της δεκαετίας του '60.

Ο T. Winograd το 1972, με το σύστημα SHRDLU, γραμμένο σε Lisp, προσέφερε το πρώτο σύστημα κατανόησης φυσικής γλώσσας (natural language understanding) θεμελιώνοντας ουσιαστικά την "περί υπάρξεως απόδειξη" της περιοχής. Ερμηνεία ερωτήσεων, δηλώσεων και οδηγιών, ικανότητα συνεπαγωγής, εκμάθηση νέων λέξεων αποτέλεσαν μερικές από τις αρετές του συστήματος.

Η διαδικασιακή (procedural) ήταν η πρώτη μορφή παράστασης γνώσης. Ακολούθησε η δηλωτική (declarative). Η διαδικασιακή επιχειρεί να περιγράψει την κίνηση ή τη δράση και φιλοδοξεί να παραστήσει τη γνώση (γλωσσική ή μη) με διαδικασίες, με άλλα λόγια το "πώς" συμβαίνει η γνώση. Η δηλωτική επιχειρεί την παράσταση των συμβολικών δομών καθ'αυτών, μ' άλλα λόγια προσπαθεί να περιγράψει "τι" συμβαίνει. Ο δηλωτικός προγραμματισμός εξελίχθηκε και συμπεριέλαβε κανόνες συνεπαγωγής (inference rules). Ο προγραμματισμός αυτός λέγεται Λογικός Προγραμματισμός και η σημαντικότερη αντίστοιχη γλώσσα είναι η Prolog.

Τα αυτόματα πεπερασμένων καταστάσεων (finite state automata: FSA) επιστρατεύτηκαν για να αναγνωριστούν γλωσσικές οντότητες (π.χ. λεξικά στοιχεία, μέρη του λόγου) και τα δίκτυα μεταφοράς πεπερασμένων καταστάσεων (Finite State Transition Networks: FSTN) για να δημιουργήσουν γλωσσικές οντότητες. Σημαντικό σημείο αναφοράς στην περιοχή αυτή αποτελεί το πρόγραμμα ELIZA του J. Weizenbaum που βασίστηκε σ'αυτά, το 1960.

Τα αναδρομικά δίκτυα μεταφοράς (Recursive Transition Networks: RTN) που ακολούθησαν ήσαν δίκτυα μεταφοράς πεπερασμένων καταστάσεων (FSTN) με αναδρομή. Μπορούσαν να παραστήσουν γραμματικές του τύπου $a^n b^n$ και τις παραστάσεις με παρενθέσεις.

Τα διευρυμένα δίκτυα μεταφοράς (Augmented Transition Networks: ATN) είχαν επιπλέον και μνήμη για να μην επαναλαμβάνουν διαδρομές. Παριστάνοντας δε μια μορφή δράσης ή ενέργειας (action) κατέστρεφαν τον δηλωτικό χαρακτήρα τους.

Οι πρώτοι “τεχνολογητές χάρτη” ή “συντακτικοί αναλυτές χάρτη” (chart parsers) που κατέγραφαν τις πετυχημένες απόπειρες, για να μην επαναλαμβάνουν άσκοπα τους ελέγχους νομιμότητας, επιβιώνουν μέχρι σήμερα ως ιδέα με διαφορετικές μεθόδους υλοποίησης.

Με τη γέννηση της Prolog και της λογικής συνεπαγωγής (inference) που αυτή επιτρέπει δημιουργήθηκαν οι πρώτοι Τεχνολογητές (parsers) γραμματικών ορισμένης πρότασης (Definite Clause Grammars: DCG). Αυτοί είναι Τεχνολογητές γραμματικών φραστικής δομής (Phrase Structure Grammars: PSG) σε γλώσσα Prolog, και αναμιγνύουν διαδικασιακή με δηλωτική μορφή παράστασης.

Η αμφισημία ως εγγενές στοιχείο της φυσικής γλώσσας που απαντάται σε όλα τα επίπεδα μελέτης της φυσικής γλώσσας, αποτελεί ‘πυροκρότητο’ της μηχανικής επεξεργασίας φυσικής γλώσσας. Έρχεται σε αντίθεση με τον ντετερμινισμό των υπολογιστικών μηχανών και τις ντετερμινιστικές φιλοσοφικές δοξασίες. Η άρση της αμφισημίας αποτέλεσε και αποτελεί πρόκληση στην έρευνα της Γλωσσολογίας και της Υπολογιστικής Γλωσσολογίας.

Chatter bots

Το **chatter bot** (ή chatbot) είναι ένας τύπος πράκτορα προσανατολισμένου για συζήτηση, ένα πρόγραμμα υπολογιστή σχεδιασμένο για να προσομοιώνει έναν ευφυή διάλογο με έναν ή περισσότερους ανθρώπους- χρήστες μέσω συστημάτων διεπαφής, πληκτρολόγιο, ηχεία, οθόνη κτλ. Αν και πολλά από αυτά τα προγράμματα περιέχουν έναν ευφυή τρόπο να επεξεργαστούν την φράση του χρήστη πριν να δώσουν μια απάντηση, τα περισσότερα chatterbots απλά ανιχνεύουν λέξεις κλειδιά μέσα στην εισαγωγή και τραβούν μια απάντηση επιλέγοντας αυτή με τις περισσότερες λέξεις κλειδιά ή το πιο παρόμοιο σχέδιο διατύπωσης από μία τοπική βάση δεδομένων.

Πρόγραμμα Eliza

Η Ελίζα είναι ένα πρόγραμμα που αρχικά δημιουργήθηκε το 1966 στο MIT από τον Joseph Weizenbaum με την πρόθεση να επιδείξει ότι μια συνομιλία σε φυσική γλώσσα μεταξύ ανθρώπου και υπολογιστή είναι εφικτή. Σκοπός του ήταν να προσομοιώσει έναν ψυχολόγο. Οι προτάσεις που πληκτρολογεί ο χρήστης (input sentences) αναλύονται με βάση κανόνες αποσύνθεσης που σχετίζονται με συγκεκριμένες λέξεις κλειδιά (key words) μέσα από την πρόταση. Οι απαντήσεις παράγονται από κανόνες σύνθεσης που εξαρτώνται από τους αντίστοιχους κανόνες αποσύνθεσης. Στην ουσία δηλαδή, το input του χρήστη αναλύεται και ερευνάτε για παρουσία λέξεων κλειδιών, οι οποίες όταν βρεθούν μετασχηματίζονται με βάση τον κανόνα που αφορά στη συγκεκριμένη λέξη κλειδί, ή σε άλλες περιπτώσεις ένα σχόλιο ή σε μια παλιότερη απάντηση. Υστέρα από αυτή τη διαδικασία η απάντηση που έχει δημιουργηθεί επιστρέφει στο χρήστη.

Υλοποίηση φυσικής γλώσσας σε ευφυή πράκτορα

Για να δοθεί σε έναν ευφυή πράκτορα η δυνατότητα να επεξεργάζεται φυσική γλώσσα και να απαντά βάση του ερεθίσματος που δέχθηκε έχει χρησιμοποιηθεί μία υλοποίηση ενός chatterbot τύπου eliza. Η υλοποίηση έχει πραγματοποιηθεί σε Java από τον Charles Hayden και είναι πιστή αναπαράσταση του προγράμματος που είχε περιγράψει το 1966 ο Joseph Weizenbaum. Από την υλοποίηση αυτή έχει χρησιμοποιηθεί ο βασικός κορμός (χωρίς τα applets / front end) και έχει τροποποιηθεί πλήρως το script της ομιλίας ώστε να υποστηρίξει το σενάριο που αναλύεται στο case study.

Όταν ένας ευφυής πράκτορας δεχθεί κάποιο ερέθισμα ομιλίας από το εξωτερικό περιβάλλον (άλλους πράκτορες) τότε στέλνει το κείμενο στο σύστημα επεξεργασίας για να πάρει

από την Eliza την κατάλληλη απάντηση. Μόλις λάβει την απάντηση την μετατρέπει σε φωνή μέσα από το σύστημα ΜΚσΟ και παράλληλα προχωρά και στο action του να πει αυτό που χρειάζεται.

Στην υλοποίηση υπάρχει ακόμα και ένα temporary buffer στον agent ώστε αν λάβει ερέθισμα ομιλίας πάνω από μία φορές χωρίς να έχει αλλάξει αυτό που λέει ο συνομιλητής να μην επαναλαμβάνει την απάντησή του.

Case Study

Η πρόκληση την οποία καλούμαστε να αντιμετωπίσουμε έπεται από τις παραπάνω τεχνολογίες και δυνατότητες που αναλύσαμε είναι ο συνδυασμός τους σε ένα πραγματικό σενάριο. Το σενάριο θα υλοποιηθεί σε έναν εικονικό κόσμο γραμμένο σε VRML ο οποίος θα τροποποιηθεί και με χρήση VERL θα εισαχθεί στο σύστημα REVE Worlds. Εκεί θα δράσουν ευφυείς πράκτορες οι οποίοι μέσω φυσικής γλώσσας θα μπορούν να ανταλλάσουν αιτήματα και να υλοποιούν τα αιτήματα άλλων πρακτόρων.

Τεχνολογίες που χρησιμοποιήθηκαν

Στο κεφάλαιο αυτό περιγράφονται οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση του Case Study της εργασίας. Αρχικά, παρουσιάζονται κάποια στοιχεία για τη γλώσσα Java. Έπειτα παρουσιάζεται το περιβάλλον ανάπτυξης IDE (Integrated Development Environment) Netbeans, με την βοήθεια του οποίου πραγματοποιήθηκε η υλοποίηση της όλης εφαρμογής. Οι τεχνολογίες Java Speech API, FreeTTS και το chatter bot eliza αναλύονται στα αντίστοιχα κεφάλαια.

Γλώσσα Java

Η Java είναι η γλώσσα προγραμματισμού που επιλέχθηκε για την εργασία. Η επιλογή έγινε επειδή το σύστημα ReveWorlds και οι ευφυείς πράκτορες στους οποίους βασίστηκε η υλοποίηση είναι υλοποιημένοι σε Java Η Java είναι μια αντικειμενοστραφής γλώσσα ανοιχτού κώδικα. Το συντακτικό της βασίζεται στο συντακτικό της C++.



Java

Κύρια χαρακτηριστικά της Java είναι τα εξής:

Η μεταγλώπτιση δεν γίνεται σε εκτελέσιμο native κώδικα αλλά σε ενδιάμεσο κώδικα που έχει την ονομασία bytecode. Ο bytecode εκτελείται μέσω interpreter ή με just-in-time compilation από το runtime μέρος της Java το Java Virtual Machine. Αυτό έχει ως μειονέκτημα την απώλεια ταχύτητας καθώς δε γίνεται απευθείας εκτέλεση του κώδικα αλλά πλεονέκτημα το ότι οποιαδήποτε υλοποίηση JVM σε οποιαδήποτε αρχιτεκτονική μπορεί να τρέξει κάποιο bytecode αρχείο. Δηλαδή επιτρέπει το ίδιο το πρόγραμμα να μπορεί να εκτελεστεί σε πολλά διαφορετικά λειτουργικά συστήματα.

Η Java παρέχει την δυνατότητα να εκτελεστεί κώδικας Java από απομακρυσμένες πηγές με ασφάλεια.

Η γλώσσα έχει σχεδιαστεί για να προσφέρει ευκολία και μεγαλύτερη παραγωγικότητα στον προγραμματιστή.

Η Java είναι γενικά εύκολη στη χρήση χρησιμοποιώντας τα καλά στοιχεία άλλων αντικεμενοστραφών γλωσσών προγραμματισμού και επίσης προσπαθεί να κάνει πιο εύκολη ιδιαίτερα την ανάπτυξη διαδικτυακών εφαρμογών.

Η Java εκμεταλλεύεται την ύπαρξη του virtual machine ώστε να προσφέρει πιο εξελιγμένες runtime υπηρεσίες πχ το reflection.

Η Java προσφέρει βιβλιοθήκες για μεγάλο πλήθος αναγκών, ως μέρος του Java platform.

Κατ' αρχήν χωρίς η Java να αποτελεί κάποιο πρότυπο για διαδικτυακές εφαρμογές αποτελεί ίσως την πιο δημοφιλή λύση. Μεγάλο μέρος από ανάλογα προγράμματα χρησιμοποιούν αυτήν τη γλώσσα. Σε πρώτη φάση η επιλογή της "πιο διαδεδομένης" γλώσσας προγραμματισμού γίνεται κυρίως για λόγους οικονομικούς. Είναι πιο εύκολο να βρεθούν προγραμματιστές που θα γνωρίζουν τη Java ώστε να προγραμματίσουν και εν συνεχεία να συντηρήσουν κάποια εφαρμογή.

Η διάδοση της Java στην ανάπτυξη ανάλογων εφαρμογών εξασφαλίζει ότι θα υπάρχει ήδη εμπειρία και τεχνικές και σε τελική φάση ακόμα και χρησιμοποίησιμος έτοιμος κώδικας που θα βοηθήσουν στην ανάπτυξη του προγράμματος. Σε μια διπλωματική άσκηση δεν τίθενται παράγοντες οικονομικοί ή εύρεσης προγραμματιστών όμως σε αντιστοιχία με το τι θα γινόταν κατά την ανάπτυξη μιας εμπορικής εφαρμογής πρέπει η διάδοση της Java να ληφθεί υπόψη.

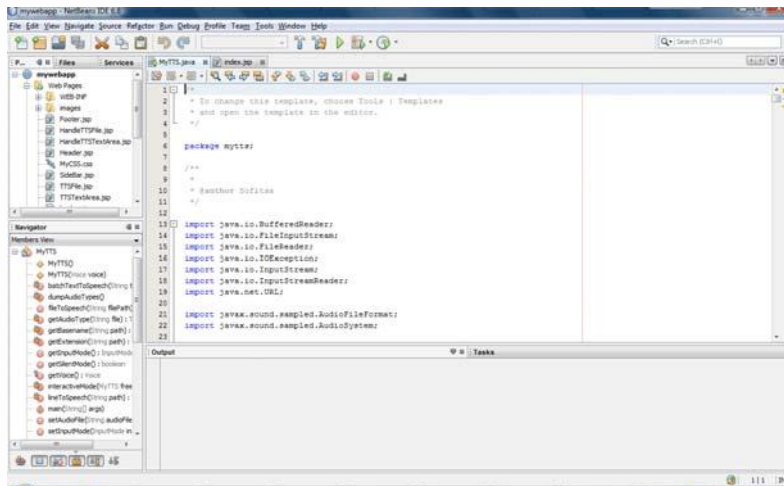
Πέραν της διάδοσης της σε ανάλογες εφαρμογές, άλλα χαρακτηριστικά της Java για τα οποία έγινε η επιλογή της είναι η ευκολία στη χρήση και η αυξημένη παραγωγικότητα που παρέχει η Java μέσω χαρακτηριστικών της γλώσσας (όπως το αυτόματο garbage collection) και των βιβλιοθηκών της.

Πρέπει να σημειωθεί ότι στην εργασία χρησιμοποιήθηκε η 1.6 έκδοση της Java.

Περιβάλλον Netbeans

Η υλοποίηση της εφαρμογής VoiceGenerator, της δυνατότητας του πράκτορα για ομιλία και της σύνδεσής του με τα υπόλοιπα συστήματα που χρησιμοποιήθηκαν πραγματοποιήθηκε στο περιβάλλον NetBeans. Το NetBeans αποτελεί μια ανοιχτή πλατφόρμα ανάπτυξης λογισμικού σε γλώσσες Java, Javascript, PHP, Python, Ruby, Groovy, C και C++. Επίσης αποτελεί και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment - IDE) και προσφέρει εργαλεία εύκολα στη χρήση. Τα εργαλεία αυτά καλύπτουν όλα τα στάδια: από την υλοποίηση και τη μεταγλώττιση έως την εφαρμογή του λογισμικού που παράχθηκε.

Για την παρούσα εφαρμογή έγινε χρήση της έκδοσης NetBeans IDE 6.8. Το NetBeans 6.8 αποτελεί το βασικό περιβάλλον ανάπτυξης προγραμματιστικού κώδικα και πολύπλοκων εφαρμογών. Οι εφαρμογές που μπορούν να αναπτυχθούν στο περιβάλλον ανάπτυξης NetBeans περιλαμβάνουν από απλές Java εφαρμογές επιπέδου κλάσεων και κωνσόλας μέχρι πολύπλοκες διαδικτυακές και επιχειρηματικές εφαρμογές με γραφικό περιβάλλον και διαδικτυακές υπηρεσίες. Επομένως γίνεται κατανοητό ότι αποτελεί το περιβάλλον στο οποίο στηρίχθηκε η ανάπτυξη, η υλοποίηση, η ενορχήστρωση, η εκτέλεση και η αποθήκευση/δημοσίευση των διαδικτυακών υπηρεσιών και των σύνθετων επιχειρηματικών διαδικασιών.

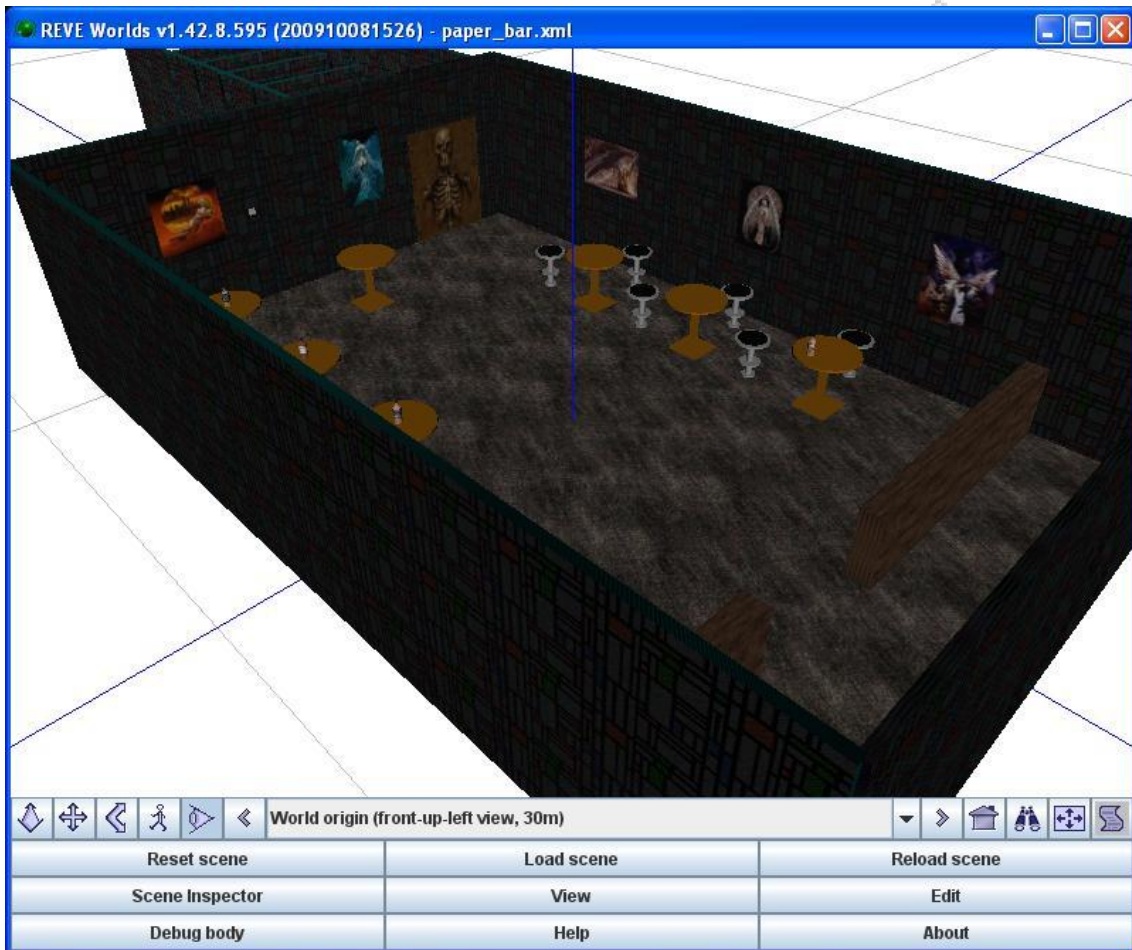


Το περιβάλλον NetBeans IDE 6.8

Σενάριο

Το σενάριο που θα εκτελεστεί στα πλαίσια του case study είναι ένα σενάριο εκτέλεσης παραγγελίας σε ένα bar. Ένας ευφυής πράκτορας θα δώσει την παραγγελία στο σερβιτόρο αφού ενημερωθεί για το τι μπορεί να παραγγείλει. Ο σερβιτόρος με τη σειρά του θα μεταβεί στο bar όπου και θα μεταφέρει στο barman την παραγγελία του πελάτη προς εκτέλεση. Μόλις ετοιμαστεί η παραγγελία θα επιστρέψει στο τραπέζι του πελάτη όπου θα αφήσει την παραγγελία και θα ενημερώσει για το λογαριασμό. Το σενάριο σταματά μόλις ολοκληρωθεί η πληρωμή.

Virtual environment (bar)



Γενική απεικόνιση του εικονικού περιβάλλοντος

Ο εικονικός κόσμος είναι ένα σύνολο από item aspects, όπως έχουμε ήδη πει σε προηγούμενο κεφάλαιο. Το πρώτο βήμα για την μοντελοποίηση του κόσμου μας σύμφωνα με την αναπαράσταση REVE είναι ο ορισμός του physical aspect των items που τον αποτελούν. Δεδομένης της αφηρημένης δομής του physical aspect σε σχεδιαστικό επίπεδο χρησιμοποιείται η VRML ως συγκεκριμένο πρότυπο αναπαράστασης αναπαράστασης τρισδιάστατων σκηνών σε μορφή διαγράμματος σκηνής. Στη συνέχεια ορίζεται το semantic aspect των items του κόσμου μας. Εδώ τα πράγματα είναι πιο συγκεκριμένα, καθώς η πληροφορία η οποία είναι αναγκαίο να είναι διαθέσιμη για το item και, κατ' επέκταση, η ακριβής δομή του semantic aspect του, είναι συγκεκριμένες. Η διαδικασία μοντελοποίησης ολοκληρώνεται με τον ορισμό του access aspect. Και εδώ τα πράγματα είναι τελείως συγκεκριμένα καθώς η διαθέσιμη λειτουργικότητα του item και, κατ' επέκταση, η ακριβής δομή του access aspect του, προκειμένου το τελευταίο να είναι ικανό να την αναπαριστά, είναι συγκεκριμένες.

Η VERL μας δίνει τη δυνατότητα να επιλέξουμε τον βαθμό λεπτομέρειας και του επιπέδου αφαίρεσης της μοντελοποίησης μας, δηλαδή την αντιστοίχιση των συστατικών του γραφήματος σκηνής σε items. Στην περίπτωση μας επιλέξαμε να αντιστοιχήσουμε κάθε συστατικό του γραφήματος σκηνής σε ένα item, για να έχουμε απλούστερη γεωμετρική δομή, μέγιστη σημασιολογική και αλληλεπιδραστική αυτονομία, καθώς και γενικότητα ως προς το "ρόλο" τους στον εικονικό κόσμο. Αυτή η επιλογή μας βοηθάει στη μετέπειτα χρήση του κόσμου από τον εικονικό μας πράκτορα, ώστε να μπορεί να αντιλαμβάνεται κάθε στοιχείο του κόσμου ξεχωριστά, να μπορεί να αλληλεπιδρά μαζί τους και να κινείται στον κενό χώρο που έτσι δημιουργείται.

Μοντελοποίηση εικονικού κόσμου

Αρχικά κάνουμε μια κατανομή των items του εικονικού μας κόσμου σε item groups με σκοπό τόσο την ταξινόμηση των items σε σύνολα με κοινά γεωμετρικά χαρακτηριστικά, σημασιολογία και δυνατότητα αλληλεπίδρασης, όσο και την ευκρίνεια του κώδικα μας και τη δυνατότητα συντήρησης και επαναχρησιμοποίησης.



Εσωτερικό του εικονικού περιβάλλοντος

Έτσι στο πρώτο item group με ονομασία *bar_building* έχουμε το σύνολο των δομικών συστατικών του μπαρ, όπως τοίχους, πατώματα, πόρτες, που αποτελούν σχεδιαστικά και εννοιολογικά το περιβάλλον μέσα στο οποίο θα δρά ένας ευφυής εικονικός πράκτορας. Το item group αυτό δεν περιλαμβάνει δυναμικά στοιχεία, όπως animations, scripts, συμπεριφορές κτλ επομένως το όρισμα behaviours θα είναι false. Τέλος το όρισμα location, που προσδιορίζει την πηγή από την οποία αντλούνται τα δεδομένα φυσικής αναπαράστασης σε δένδροειδή δόμηση είναι το αρχείο bar\bar.wrl. Επομένως το item group bar_building ορίζεται όπως παρακάτω.

```
<itemGroup name="bar_building" location="bar/bar.wrl"  
  behaviours="false">  
</itemGroup>
```

Κάθε item του item group αυτού θα περιλαμβάνει ένα μοναδικό όνομα name και την κλάση στην οποία ανήκει. Η κλάση όλων των items θα έχει τιμή real, ώστε να γίνονται αυτά αντιληπτά από έναν εικονικό πράκτορα. Στη συνέχεια ορίζεται το physical aspect του item στο οποίο

καθορίζουμε το source της φυσικής αναπαράστασης για το συγκεκριμένο item, το translation orientation, όπου αυτό χρειάζεται. Π.χ ο ορισμός του item για την αναπαράσταση του δαπέδου είναι ο εξής:

```
<item name="floor" class="real">
  <virtualModel source="Floor">
    <transform translation="0, 0, 0"/>
  </virtualModel>
</item>
```

Στη συνέχεια με τον ίδιο τρόπο ορίζονται και τα υπόλοιπα item groups, συγκεκριμένα το bathroom_utilities, δηλαδή τουαλέτες και νιπτήρας, Tables, που περιλαμβάνει τα τραπέζια, το chairs, για τις καρέκλες, το bar_chairs, για τα σκαμπό της μπάρας, το paintings για τους πίνακες και άλλα εικαστικά στοιχεία, το item group DJ, που περιλαμβάνει το σύνολο των items που αποτελούν το χώρο του DJ και τέλος το Bar_Table, που αντιστοιχεί στη μπάρα και τα ποτά πίσω από αυτή.

	Element	Location		Element
bar_building	floor	0,0,0	Tables	Table_1a
	wall_01	4,1,5,0		Table_1b
	wall_02	-4,1,5,0		Table_1c
	wall_03	0,1,5,6		Table_1d
	great_wall	-1,1,5,-6		Table_1e
	small_wall_01	3.75, 1.5, -6		Table_2a
	small_wall_02	2.75, 2.5, -6		Table_2b
	corridor_Wall	1.95, 1.5, -6.5		Table_2c
	Corridor_Wall_02	3.55, 1.5, -9		Table_2d
	Corridor_Wall_03	1.95, 1.5, -8.5		Table_2e
	Bathroom_Wall_01	0.9, 1.5, -6.95		Table_3a
	Bathroom_Wall_02	0.9, 1.5, -8.05		Table_3b
	Bathroom_Wall_03	0.9, 1.5, -8.95		Table_3c
	Bathroom_Wall_04	0.9, 1.5, -10.05		Table_3d
	Bathroom_Wall_05	2.75, 1.5, -12		Table_3e
	Bathroom_Wall_06	0, 1.5, -9		Table_4a
	Bathroom_Small_Wall	1.95, 1.5, -11		Table_5a
	Bathroom_Floor	2.75, 0, -9		Table_5b
	Bathroom_Door_01	3.48, 1, -6		Table_5c
	Bathroom_Door_02	2.02, 1, -6		Table_5d
	Bathroom_Door_03	1.9, 1, -7		Table_5e
	Bathroom_Door_04	1.9, 1, -9		Table_6a
	DJ	DJ_table		-3,0.65,4.2
DJ_floor		-3,0.15,5.1	Chair_02	
DJ_console		-3,0.45,5.1	Chair_03	
Bar_Table	Bar_Table	2,0.5,4.2	Chair_04	
	Bar_Drinks	0,1.1,5.85	Chair_05	
Paintings	Painting_01	3.9,1.7,0	Chair_06	
	Battleaxe	3.9,2.2,1.5	Chair_07	
	Guitar	3.9,2.2,-1.5	Chair_08	
	Painting_02	3.9,1.7,3	Chair_09	
	Painting_03	3.9,1.7,-3	Chair_10	
	Painting_04	-3.9,1.7,-2	Chair_11	
	Painting_05	-3.9,1.7,3	Chair_12	
	Painting_06	0,2,5.9	Chair_Bar_01	
Bathroom_utilities	Painting_07	-2,1.7,-5.9	Chair_Bar_02	
	Painting_08	1,1.7,-5.9	Chair_Bar_03	
	Basin	2,75,0.8,-11.5	Chair_Bar_04	
	Toilet_1	0.75,0.1,-9.5	Chair_Bar_05	
	Toilet_2	0.75,0.1,-7.5		

Παράδειγμα items και items groups

Για να γίνει κατανοητή η παραπάνω διαδικασία θα γίνει μια ολοκληρωμένη μοντελοποίηση του item Painting_01. Το VRML μοντέλο ορίζεται όπως παρακάτω.

```
Painting1
DEF Painting_01
  Transform[ translation 3.9 1.7 0
  children[
    Shape {appearance Appearance{ material Material()
      texture ImageTexture {url ["paintings/gothicangel.jpg"]
        repeatS TRUE
        repeatT TRUE
      }
      textureTransform TextureTransform{ scale 1 1 rotation 0.0
        | center 0.0 0.0 translation 0.0 0.0
      }
    }
    geometry Box {size 0.0001 1 0.8}
  ]
}
```

Αρχικά ορίζουμε το item name και το item class του αντικειμένου. Σε σχεδιαστικό επίπεδο το item name μπορεί να είναι οποιοδήποτε αλφαριθμητικό, αλλά σε επίπεδο υλοποίησης θέλουμε το όνομα να προσιδιάζει το αντικείμενο, είτε γεωμετρικά είτε σημασιολογικά είτε λειτουργικά. Έτσι ώστε και ο χρήστης να μπορεί στη συνέχεια να αναγνωρίσει το αντικείμενο αυτό με την ονόμασία του αλλά και γιατί το όνομα του item παρέχεται ως πληροφορία στον εικονικό μας πράκτορα, επομένως για να μπορεί να μοντελοποιηθεί η συμπεριφορά είναι αναγκή τα αντικείμενα να έχουν κάποια αντιστοίχιση με την φυσική τους αναπαράσταση. Για την ικανοποίηση των παραπάνω απαιτήσεων δίνουμε στο item μας το όνομα Painting_01. Η κλάση του αντικειμένου μας ορίζει την δυνατότητα αντίληψης του από έναν ευφυή εικονικό πράκτορα. Έτσι ορίζουμε class=real, ώστε για να είναι δυνατή η μετάδοση πληροφορίας για το item και άρα η δυνατότητα επεξεργασίας και αποθήκευσης της με σκοπό την αλληλεπίδραση του εικονικού μας πράκτορα με αυτό.

Στη συνέχεια πρέπει να ορίσουμε το physical aspect του αντικειμένου. Η γλώσσα ορισμού VERL για να ορίσει το item χρειάζεται, όπως είπαμε την πηγή της φυσικής αναπαράστασης, η οποία στο παραδειγμα μας είναι το DEF Painting_01. Με αυτό τον τρόπο ορίζουμε στην VRML ένα σύνολο κόμβων που αποτελούν γεωμετρικά και εννοιολογικά ένα αντικείμενο προς επεξεργασία. Έτσι έχοντας ορίσει μέσα στην VRML το αντικείμενο Painting_01 μπορούμε να το χρησιμοποιήσουμε για να ορίσουμε το source ενός item στη VERL.

Το επόμενο βήμα είναι να ορίσουμε τη θέση του αντικειμένου στο χώρο. Η VRML προβλέπει την ύπαρξη κόμβων τύπου Transform οι οποίοι εφαρμόζουν τρισδιάστατους χωρικούς μετασχηματισμούς στους κόμβους - παιδιά τους, μέσω των πεδίων τους translation και rotation για μετατόπιση και περιστροφή, αντίστοιχα. Επίσης περιλαμβάνει και χαρακτηριστικά για τις διάφορες ιδιότητες του αντικειμένου που ορίζει, όπως τον τύπο του υλικού το texture που θα απεικονίζει, τις διαστάσεις του. Το μόνο που χρειαζόμαστε είναι να ορίσουμε το translation του item. Όσο για τις υπόλοιπες ιδιότητες του αντικειμένου θα εξαχθούν από το μοντέλο φυσικής αναπαράστασης.

Τα semantic και physical aspects δεν είναι αναγκαία για τη μοντελοποίηση του item. Στην περίπτωση μας δεν θα χρειαστούμε άλλες ιδιότητες για το item πέρα από τη θέση και τη φυσική του αναπαράσταση. Επομένως από τη στιγμή που ο εικονικός μας πράκτορας δεν θα αλληλεπιδράσει με άλλο τρόπο με το αντικείμενο αυτό ο παρακάτω ορισμός του item είναι επαρκής.

Μεταπτυχιακή Διατριβή Γεωργίσης Δημήτριος

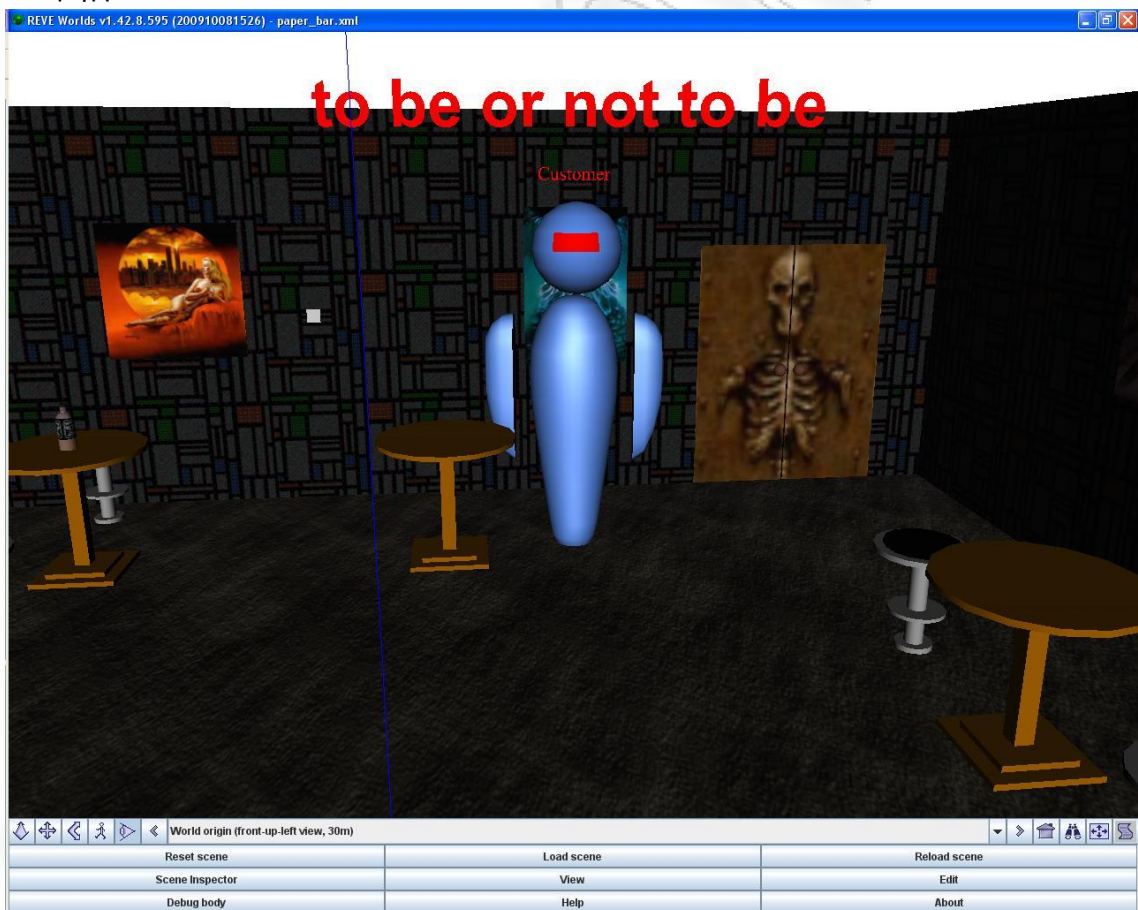
```
<item name="Painting_01" class="unreal">  
  <virtualModel source="Painting_01">  
    <transform translation="3.9,1.7,0"/>  
  </virtualModel>  
</item>
```

Agents in bar

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση των ευφυών πρακτόρων που συμμετέχουν στο σενάριο.

Πράκτορας Customer

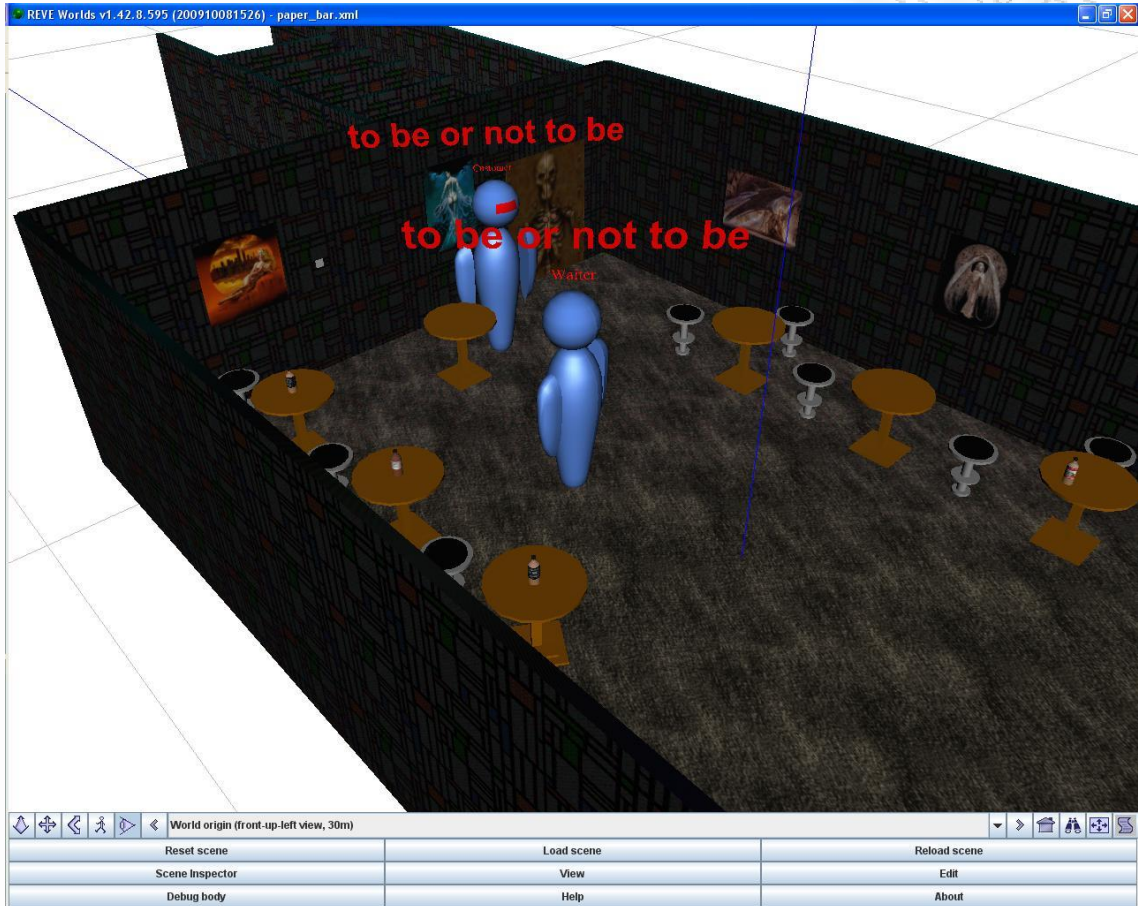
Είναι ο πελάτης που ζητά να παραγγείλει ένα ποτό. Ως οντότητα ενεργοποιείται τη στιγμή που ο σερβιτόρος τον ρωτά αν θα ήθελε να παραγγείλει κάτι. Αφότου εξαντληθεί ο διάλογος και ο σερβιτόρος φύγει με την παραγγελία, ο πράκτορας αυτός αδρανεί μέχρι να του σερβιριστεί αυτό που παρήγγειλε.



Πράκτορας Customer

Πράκτορας Waiter

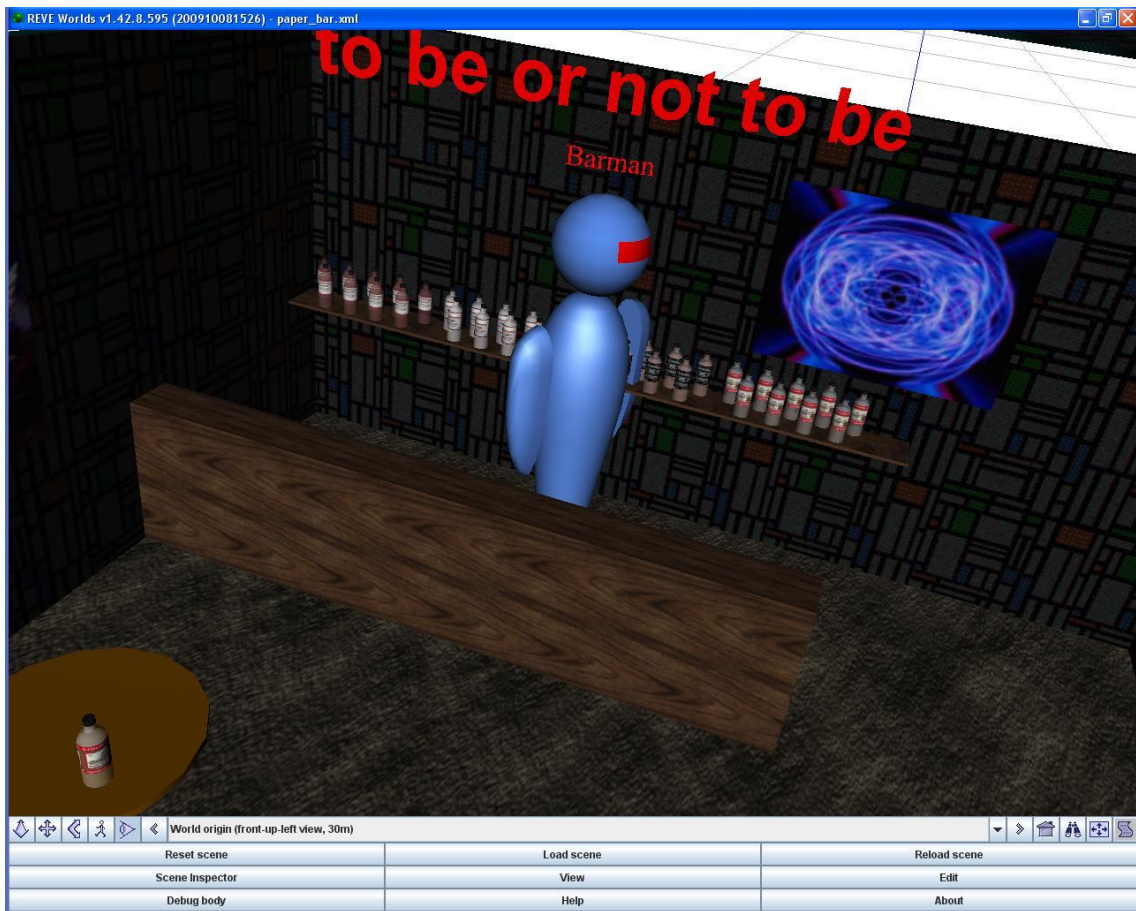
Αποτελεί το σερβιτόρο στο σενάριό μας. Είναι μονίμως ενεργός και αφού ενημερώνει τον πελάτη φεύγει για να μεταφέρει στο barman την παραγγελία του πελάτη. Μόλις αυτή είναι έτοιμη την πηγαίνει στον πελάτη μαζί με το λογαριασμό.



Ο πράκτορας Waiter με τον Customer

Πράκτορας Barman

Ο ρόλος του barman είναι να ετοιμάσει την παραγγελία που του μεταφέρει ο σερβιτόρος. Μόλις την ετοιμάσει ενημερώνει το σερβιτόρο για να την παραδώσει.



Ο πράκτορας barman

Συμπεράσματα

Το σενάριο εκτελέστηκε επιτυχώς όπως φαίνεται στο επισυναπτόμενο video και η προσομοίωση ήταν αισθητά πιο ρεαλιστική από την απλή ανταλλαγή συστημικών μηνυμάτων ανάμεσα στους πράκτορες. Κάθε άλλο όμως παρά εξαντλήθηκαν οι δυνατότητες επέκτασης αυτής της προσέγγισης που ακολουθήθηκε. Η υποδομή η οποία έχει δημιουργηθεί είναι εύκολα επεκτάσιμη με αρκετούς τρόπους. Παρακάτω θα προτείνουμε μερικούς από αυτούς.

Voice Recognition

Η πρώτη και πιο σημαντική αλλαγή που θα μπορούσε να γίνει είναι να αποκτήσει ο πράκτορας δυνατότητες ΜΟσΚ (μετατροπής ομιλίας σε κείμενο). Αυτή τη στιγμή το σύστημα που δημιουργεί φυσική γλώσσα κάνει απλά ανάγνωση των κειμένων που δημιουργήθηκαν. Ο ευφυής πράκτορας δεν «ακούει» την ομιλία του έτερου αλλά «διαβάζει» το κείμενο το οποίο έχει δημιουργηθεί και βάση αυτού απαντά. Θα μπορούσε λοιπόν να εμπλουτιστεί ο πράκτορας με ένα σύστημα ASR (automatic speech recognizer) όπως αναλύεται σε προηγούμενο κεφάλαιο και έτσι να μη χρειάζεται η παρέμβαση του κειμένου. Κάτι τέτοιο θα έδινε τη δυνατότητα πολύ πιο άμεσης επικοινωνίας και με πραγματικούς χρήστες οι οποίοι αυτή τη στιγμή θα πρέπει να γράφουν αυτό που θέλουν να «πουν» στον πράκτορα.

Επικοινωνία στα Ελληνικά

Στα πλαίσια της εργασίας έγινε μία πρώτη ανάλυση για την πιθανή επέκταση του συστήματος στην Ελληνική γλώσσα. Το σύστημα που χρησιμοποιούμε για παραγωγή και επεξεργασία φυσικής γλώσσας(ELIZA) εύκολα τροποποιήθηκε για να την υποστηρίξει. Για την ανάγνωση των κειμένων μπορεί να χρησιμοποιηθεί ο συνθέτης ομιλίας «Δημοσθένης» όπου παρέχεται δωρεάν με το αντίστοιχο API αλλά είναι γραμμένος σε γλώσσα C.

Interfaces

Έχοντας αναλύσει τα δομικά στοιχεία που χρειάζονται για να έχει ο ευφυής πράκτορας δυνατότητα ομιλίας σε φυσική γλώσσα, μία επέκταση θα μπορούσε να είναι η προτυποποίησή των δυνατοτήτων αυτών σε Interfaces ώστε να μπορούν να υλοποιηθούν έπειτα εύκολα από οποιονδήποτε χρειάζεται κάτι διαφορετικό. Πιο συγκεκριμένα το interface της ομιλίας που θα μπορούσε να υλοποιηθεί θα είχε τις εξής μεθόδους.

Method Name	Input	Description
Speak	String	Γίνεται ανάγνωση του κειμένου που θέλει να πει ο πράκτορας.
Listen	Byte	Γίνεται επεξεργασία των ηχητικών δεδομένων από άλλο πράκτορα και μετατροπή τους σε κείμενο.
Listen	ByteArrayInputStream	Γίνεται επεξεργασία του ηχητικού stream από άλλο πράκτορα και μετατροπή του σε κείμενο.
Understand	String	Εδώ θα μπορούσε να καλείται το module της επεξεργασίας φυσικής γλώσσας. Παίρνει σαν είσοδο ένα κείμενο και το επεξεργάζεται. Είναι αυτή η μέθοδος που θα πρέπει να ενημερώσει τον πράκτορα αν του ζητείται να κάνει κάποιο action, αν θα πρέπει να απαντήσει με βάση τη γνώση του, αν θα πρέπει να αγνοήσει την ομιλία κ.ο.κ.
Respond	String	Εδώ θα καλείται το module δημιουργίας φυσικής γλώσσας. Λογικά θα καλείται μέσα από την Understand όταν ο πράκτορας επιθυμεί να απαντήσει κάτι.

Η λογική που έχει σχεδιαστεί αυτό το interface είναι ώστε να δώσει τη δυνατότητα σε όποιον επιθυμεί να δημιουργήσει δικό του module ομιλίας. Για παράδειγμα θα μπορούσε να έχει το δικό του μηχανισμό για δημιουργία ομιλίας, ή το δικό του μηχανισμό για τις απαντήσεις στα πλαίσια μιας συζήτησης. Το interface έχει σχεδιαστεί ακολούθως.

```
/**
 *
 * @author dgeorgitsis
 */
public interface ISpeech {

    public void Speak(String text);

    public String Listen(Byte voice);
```

```
public String Listen(java.io.ByteArrayInputStream voice);  
  
public void Understand(String text);  
  
public string[] Respond(String text);  
}
```

Intelligent Agent Full Communication Interface

Επίσης ένα τμήμα αυτού του interface θα μπορούσε να υλοποιηθεί ξεχωριστά ως προς τη δυνατότητα της ακοής της ομιλίας. Αυτό θα μπορούσε να δώσει τη δυνατότητα σε άλλους πράκτορες να γνωρίζουν πότε κάποιος έτερος πράκτορας έχει τη δυνατότητα να ακούει αυτό που θέλουν να του μεταδώσουν με φυσική γλώσσα.

```
/**  
 *  
 * @author dgeorgitsis  
 */  
public interface IListening {  
  
public void Listen(Byte voice);  
  
public void Listen(java.io.ByteArrayInputStream voice);  
}
```

Intelligent Agent Basic Communication Interface

Βιβλιογραφία

Anastassakis, G. and Panayiotopoulos, T., "A System for Logic-Based Intelligent Virtual Agents", *International Journal of Artificial Intelligence Tools*, Vol. 13, N3, pp. 593-621, World Scientific, September, 2004.

Arnellos, A., Vosinakis, S., Anastassakis, G. and Darzentas, J., "Autonomy in Virtual Agents: Integrating Perception and Action on Functionally Grounded Representations", in *Artificial Intelligence: Theories, Models and Applications*, Darzentas, J., Vouros, G.A., Vosinakis, S. and Arnellos, A. (Eds.), *Lecture Notes in Computer Science*, Vol. 5138, pp. 51-63, Springer, 2008.

Vosinakis, S. and Anastassakis, G., "Modeling Spatiotemporal Uncertainty in Dynamic Virtual Environments", in *Intelligent Virtual Agents*, Pelachaud, C., Martin, J.C., Andri, E., Chollet, G., Karpouzis, K. and Peli, D. (Eds.), *Lecture Notes in Computer Science*, Vol. 4722, pp. 409-410, Springer, 2007 (poster).

Anastassakis, G. and Panayiotopoulos, T., "A Framework for Uniform Development of Intelligent Virtual Environments", in *Advances in Artificial Intelligence*, Antoniou, G., Potamias, G., Spyropoulos, C. and Plexousakis, D. (Eds.), *Lecture Notes in Artificial Intelligence*, Vol. 3955, N1, pp. 13-24, Springer, 2006.

Anastassakis, G. and Panayiotopoulos, T., "Towards a General-Purpose Intelligent-Agent-Oriented Programming Language", 2004 (short paper, presented at the 3rd Hellenic Conference on Artificial Intelligence).

Anastassakis, G., Ritchings, T. and Panayiotopoulos, T., "Multi-agent Systems as Intelligent Virtual Environments", in *KI2001: Advances in Artificial Intelligence*, Baader, F., Brewka, G. and Eiter, T. (Eds.), *Lecture Notes in Artificial Intelligence*, Vol. 2174, pp. 381-395, Springer, 2001.

Anastassakis, G., Panayiotopoulos, T. and Ritchings, T., "Virtual Agent Societies with the mVITAL Intelligent Agent System", in *Intelligent Virtual Agents*, de Antonio, A., Aylett, R. and Ballin, D. (Eds.), *Lecture Notes in Artificial Intelligence*, Vol. 2190, pp. 112-125, Springer, 2001.

Anastassakis, G., "Intelligent Agents in Virtual Worlds", Master's Thesis, University of Manchester Institute of Science and Technology (UMIST), 2000.

Panayiotopoulos, T. and Anastassakis, G., "Towards a Virtual Reality Intelligent Agent Language", in *Advances in Informatics*, Fotiadis, D. and Nikolopoulos, S. (Eds.), pp. 249-259, World Scientific, 1999.

Vosinakis, S. and Panayiotopoulos, T., "A Task Definition Language for Virtual Agents", *Journal of WSCG*, Vol. 11, N3, pp. 512-519, 2003.

Vosinakis, S. and Panayiotopoulos, T., "A Tool for Constructing 3D Environments with Virtual Agents", *Multimedia Tools and Applications*, Vol. 25, N2, pp. 253-279, February, 2005.

Adami A., Hermansky H., (2003), "Segmentation of speech for speaker and language recognition", In Proc. EUROSPEECH 2003, pp. 841-844, Geneva.

Adell J, Bonafonte A, Gomez JA, Castro MJ. (2005), Comparative study of automatic phone segmentation methods for TTS, In: Proc. 2005 IEEE

Black A.W., Lenzo, K., (2007), "Building Synthetic Voices", (available at: <http://www.festvox.org/bsv/bsv.pdf>).

Black A. W., Lenzo K. A., (2003), "Building Voices in the FESTIVAL Speech Synthesis System", <http://festvox.org/bsv>.

Black A. W., Taylor P., Caley R., (1998), "The FESTIVAL Speech Synthesis System", <http://www.festvox.org>.

Boersma P., Weenink D., (2005), "Praat: doing phonetics by computer", available at: <http://www.praat.org>.

Dutoit T., Bagein M., Malfrere F., Pagel V., Ruelle A., Tounsi N., Wynsberghe D., (2000), "EULER : an Open, Generic, Multi-lingual and Multi-Platform Text-To-Speech System", Proc. 2nd Int'l Conf. Language Resources and Evaluation (LREC 2000), Athens, Greece, pp. 563-566.

Dutoit, T., Pagel, V., Pierret, N., Bataille, F., Van Der Vreken, O., (1996), "The MBROLA Project: Towards a Set of High-Quality Speech Synthesizers Free of Use for Non-Commercial Purposes", Proc. 4th Int'l Conf. Spoken Language Processing (ICSLP '96), Philadelphia, PA, USA, vol. 3, pp. 1393-1396.

Dutoit., T., Leich, H., (1993), "Text-to-speech synthesis based on a MBR re-synthesis of segments database", Speech Communication, vol. 13, pp. 435-440.

Ekman, P., (1994), "All emotions are basic", eds. P. Ekman and R. Davidson, The Nature of Emotion: Fundamental Questions, Oxford University Press, New York, pp. 56-58.

Fordyce, C., S., and Osterdorf, M., :1998, Prosody Prediction for Speech Synthesis Using Transformational Rule-Based Learning. Proceedings of International Conference on Spoken Language Processing, pp. 682-685

FreeTTS; <http://freetts.sourceforge.net>.

Hirschberg, J., (1993) "Pitch accent in context: predicting intonational prominence from text", Artificial Intelligence 63, pp. 429-432.

Jarifi S, Pastor D, Rosec (2008) O. A fusion approach for automatic speech segmentation of large corpora with application to speech synthesis. Speech Communication 2008;50:67-80. (available at:https://wiki.inf.ed.ac.uk/twiki/pub/CSTR/SpeakPreviousSemesters/tts_segmentation.pdf)

JavaTMSpeech API Programmer's Guide, <http://java.sun.com/products/java-media/speech/forDevelopers/jsapi-guide>

Klatt, D. H. (1982). "The Klattalk text-to-speech conversion system", Proceedings on the International Conference on Acoustic, Speech and Signal Processing'82, Paris, pp.1589--1592.

Kominek J, Black A. (2004) A family-of-models approach to HMM-based segmentation for unit selection speech synthesis. In: Proc. 8th Internat. Conf. on Spoken Language Processing (ICSLP 2004); 2004. p. 1385-1388. (available at: http://www.cs.cmu.edu/~awb/papers/ICSLP2004/WeC1401o.3_p1146.pdf)

Page J. H., Breen A. P., (1996), "The Laureate text-to-speech system architecture and applications", BT Technology Journal, vol. 14, no. 1, pp. 57-67.

Palmer, D., Hearst, M., (1997), "Adaptive Multilingual Sentence Boundary Disambiguation", Journal of Computational Linguistics, vol. 23, issue 2, pp. 241–267.

Reiter E., Dale. R. "Building Natural Language Generation Systems", Cambridge University Press, 2000, (available at: <http://assets.cambridge.org/052162/0368/sample/0521620368WSN01.pdf>).

Sproat, R., (1996), "Multilingual text analysis for text-to-speech synthesis", Journal of Natural Language Engineering, 2(4):369—380.

Tatham, M., Morton, K., (2004), "Expression in Speech: Analysis and Synthesis", Oxford University Press, Oxford.

The FESTIVAL Speech Synthesis System homepage <http://www.cstr.ed.ac.uk/projects/festival/>

The Java EE 6 Tutorial, <http://docs.sun.com/app/docs/doc/820-7627/gfirp?a=view>

Theune, M., Klabbers, E., Odijk, J., Pijper, J. R. de, and Krahmer, E., (2001), "From Data to Speech: A General Approach", Natural Language Engineering, vol. 7, no. 1, pp.47-86.

Walker W., Lamere P., Kwok P. (2002), FreeTTS - A Performance Case Study, Sun Microsystems

Παράρτημα

Οδηγός για προσθήκη των δυνατοτήτων στον Ευφυή Πράκτορα

Προσθήκη effector Talk στο access point Mouth στον Agent

Στον Agent μας έχει ακολουθηθεί όλη η λογική effector/access point του reve. Για να προσθέσουμε τη δυνατότητα ομιλίας πρέπει να δημιουργήσουμε τον effector Talk ο οποίος δρα πάνω στο access point Mouth.

Για να γίνει αυτό προσθέτουμε τον παρακάτω κώδικα στο MySimpleBody.java. Ο κώδικας αυτός δημιουργεί το access point mouth στο οποίο μπορεί να γίνει η χρήση του effector Talk.

1. Προσθέτουμε στη μέθοδο create τον κώδικα που παρατίθεται. Έτσι μπορούμε να δημιουργήσουμε τα nodes με τα οποία θα αλληλεπιδρούμε στο 3d μοντέλο του πράκτορα για να εμφανίζεται η ομιλία.

```
x3dSpeechNode = scene.getX3DNodeByName(sSpeechNodeName);
if (x3dSpeechNode != null) {
    System.out.println("Found speech node");
    try{
        mfstringSpeechField = (MFString) x3dSpeechNode.getField(sSpeechFieldName);
        // mfstringSpeechField = (SFString) x3dSpeechNode.getField(sSpeechFieldName);
        // mfstringSpeechField = (SFColor) x3dSpeechNode.getField(sSpeechFieldName);
        System.out.println("Found speech field");
    }
    catch (Exception ex) {
        throw new IllegalArgumentException("Speech field not found");
    }
}
else {
    throw new IllegalArgumentException("Speech node not found");
}
x3dMouthNode = scene.getX3DNodeByName(sMouthNodeName);
if (x3dMouthNode != null) {
    System.out.println("Found mouth node");
}
else {
    throw new IllegalArgumentException("Mouth node not found");
}
```

```
}  
  
x3dSpeechNode = scene.getX3DNodeByName(sSpeechNodeName);  
if (x3dSpeechNode != null) {  
    System.out.println("Found speech node");  
    try{  
        mfstringSpeechField = (MFString) x3dSpeechNode.getField(sSpeechFieldName);  
        // mfstringSpeechField = (SFString) x3dSpeechNode.getField(sSpeechFieldName);  
        // mfstringSpeechField = (SFColor) x3dSpeechNode.getField(sSpeechFieldName);  
        System.out.println("Found speech field");  
    }  
    catch (Exception ex) {  
        throw new IllegalArgumentException("Speech field not found");  
    }  
}  
else {  
    throw new IllegalArgumentException("Speech node not found");  
}  
  
x3dMouthNode = scene.getX3DNodeByName(sMouthNodeName);  
if (x3dMouthNode != null) {  
    System.out.println("Found mouth node");  
}  
else {  
    throw new IllegalArgumentException("Mouth node not found");  
}
```

2. Προσθέτουμε τον παρακάτω κώδικα στη μέθοδο postAdd(). Οι μέθοδοι που καλούνται στο τέλος είναι ο τρόπος με τον οποίο εμφανίζουμε στο 3d μοντέλο μας αυτό που θέλει να πει ο πράκτορας.

```
if (x3dSpeechNode != null) {  
    SimpleAccessPoint apSpeech = new SimpleAccessPoint();  
    apSpeech.create("SpeechAccessPoint", x3dSpeechNode, parentsArray);  
    X3DFieldFunctionL1 fx3dMouth = new X3DFieldFunctionL1(x3dSpeechNode);  
    fx3dMouth.setScene(scene);  
    fx3dMouth.create(sSpeechFieldName);  
    apSpeech.add("mouth", fx3dMouth);  
    this.getAccessModel().add(apSpeech);  
    Effector efTalk = new Effector();  
    efTalk.setScene(scene);  
    efTalk.setItem(this);  
    efTalk.create("Talk",x3dMouthNode);  
    SimpleSingleFunctionActionL1 aTalk = new SimpleSingleFunctionActionL1();  
    aTalk.create();  
    efTalk.add("now", aTalk);  
    SelectActionL1 sTalk = new SelectActionL1();  
    sTalk.setScene(scene);  
    sTalk.setEffector(efTalk);  
    efTalk.add("select", sTalk);  
    efTalk.attach(apSpeech, false, false);  
    this.getEffectors().add(efTalk);  
}
```

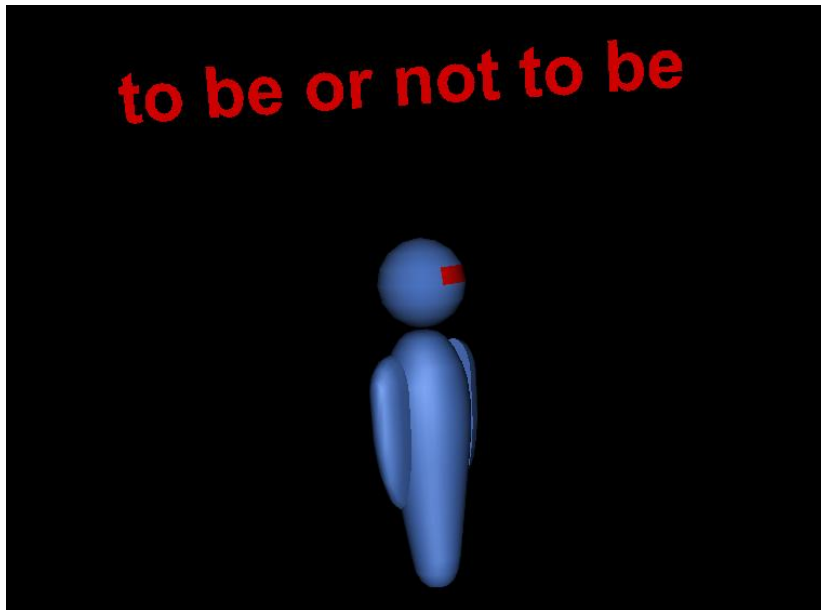
```
String temp = name + " says";  
phrase.create(temp, 1);  
  
X3DConnectorL1 talkconnector = new X3DConnectorL1();  
talkconnector.setScene(scene);  
talkconnector.create(sSpeechNodeName,sSpeechFieldName);  
phrase.setComponent(0, talkconnector);  
  
}
```

Προσθήκη nodes στο vrml μοντέλο μας

Εφόσον πραγματοποιήσουμε το προηγούμενο βήμα πρέπει να προσθέσουμε και στο 3d μοντέλο μας το node εκείνο στο οποίο θα εμφανίζεται το κείμενο της ομιλίας του agent (το mouth access point που περιγράψαμε παραπάνω). Για να γίνει αυτό πρέπει να προσθέσουμε το παρακάτω το vrml αρχείο του agent που φορτώνει το ReveWorlds. Έτσι δημιουργούμε ένα Billboard node στο οποίο μπορούμε να προσθέτουμε κείμενο.

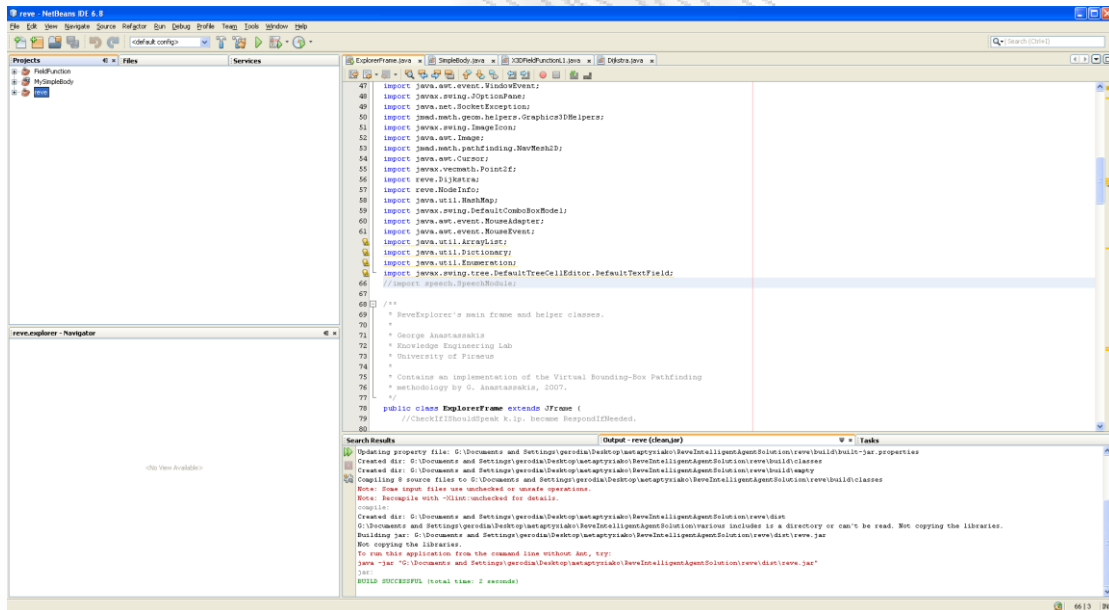
```
DEF r2d2 Transform {  
  
  children [  
  
    Billboard {  
axisOfRotation 0 1 0 #will turn around the Y axis  
children [  
  DEF Mouth Transform {  
translation 0.0 2.0 0.0  
children [  
  Shape {  
    appearance Appearance {  
      material DEF Mat Material { diffuseColor 1 0 0 }  
    }  
    geometry DEF Speech Text {  
      string ["to be or not to be"]  
      fontStyle FontStyle {  
        family "SANS"  
        style "BOLD"  
        size 0.5  
        justify "MIDDLE"  
      }  
    }  
  }  
] ]  
} ]  
}
```

Μετά τον κώδικα ο agent μας θα πρέπει να φαίνεται κάπως έτσι:

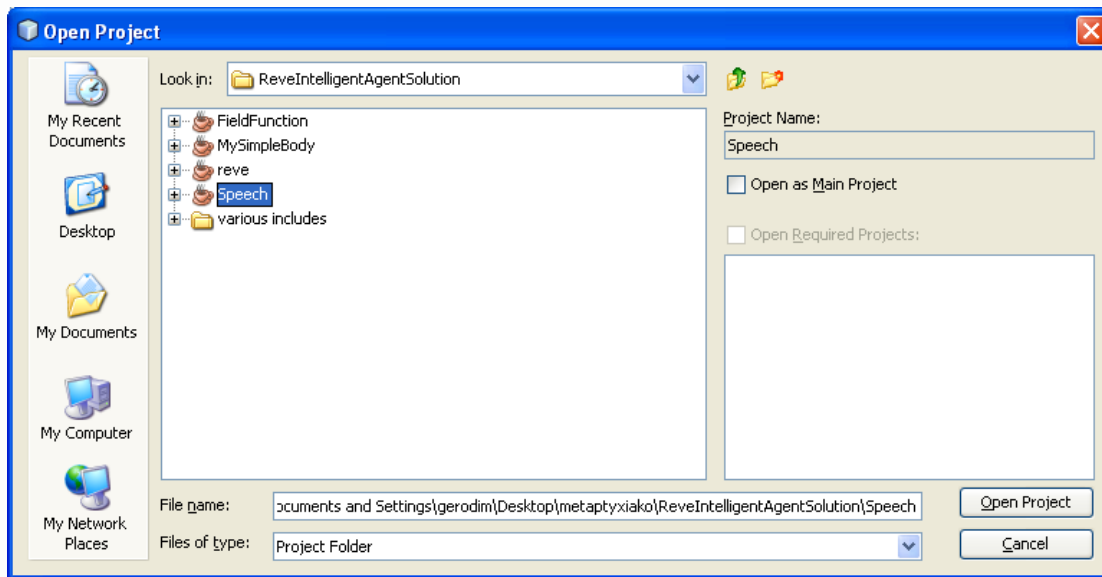


Προσθήκη του SpeechModule στον κώδικα του Ε.Π.

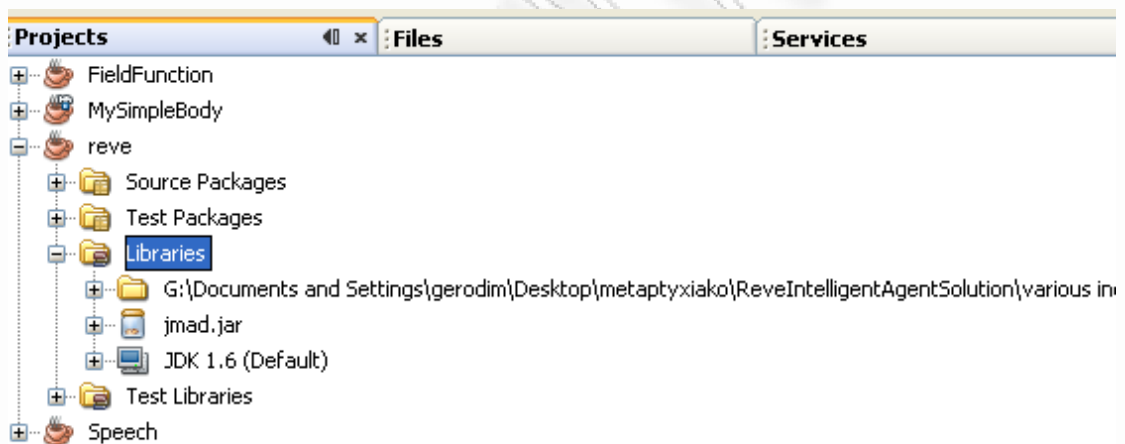
1. Ανοίγουμε το Netbeans και τα project που αποτελούν τον ευφυή πράκτορά μας.



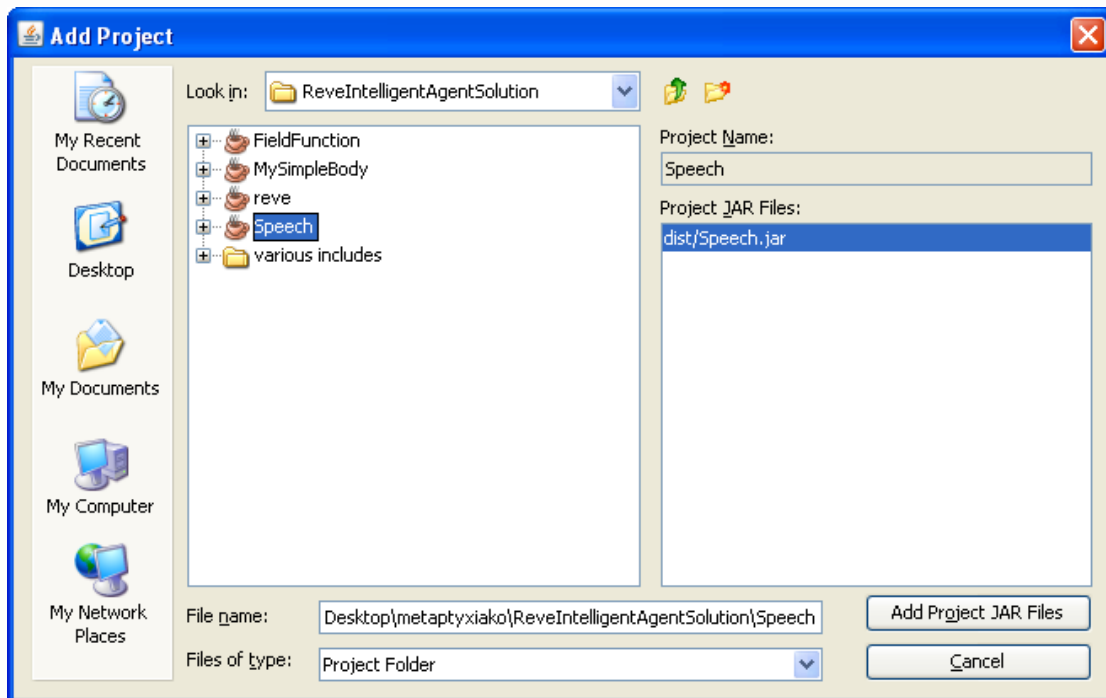
2. Επιλέγουμε File->Open Project και ανοίγουμε το project Speech.



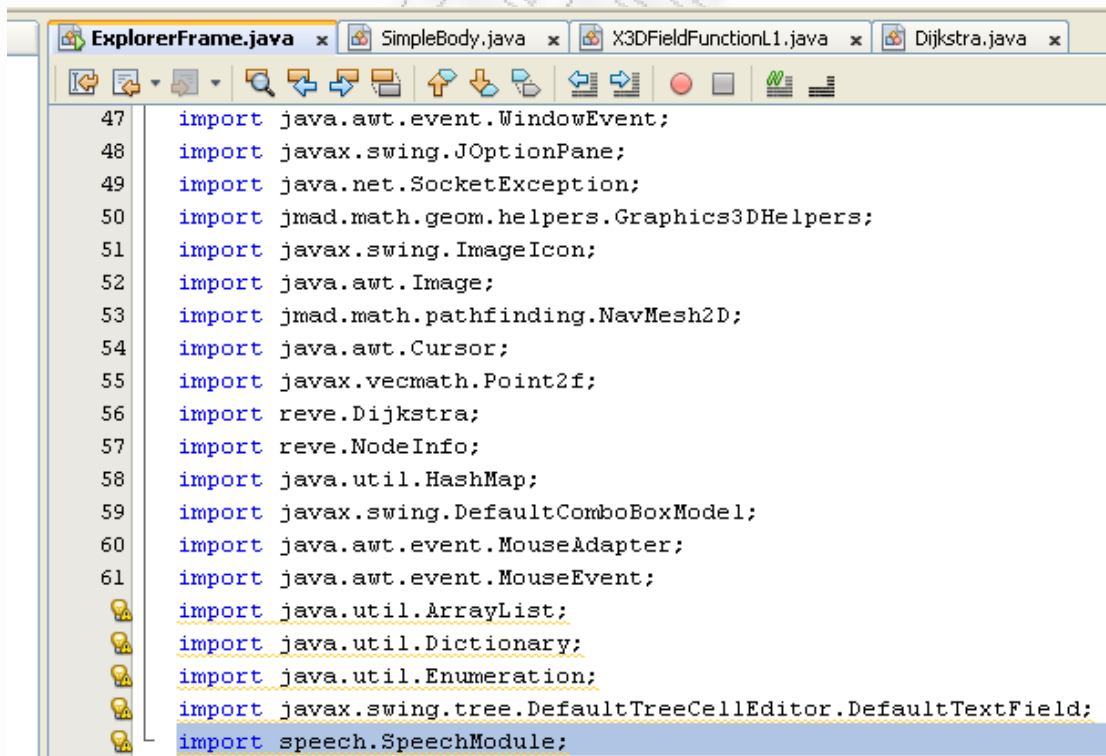
3. Στο project reve κάνουμε δεξί click στις Libraries και επιλέγουμε add project.



4. Επιλέγουμε το project Speech και πατάμε Add Project JAR files.



5. Με την ίδια διαδικασία προσθέτουμε και το freetts.jar. Προσέχουμε να έχει στον ίδιο φάκελο όλα τα distributables του freetts.
6. Στην class ExplorerFrame (αρχείο ExplorerFrame.java) προσθέτουμε στα imports το «speech.SpeechModule» όπως φαίνεται παρακάτω.



7. Στις μεταβλητές της κλάσης προσθέτουμε το παρακάτω.

```
/**
 * ReveExplorer's main frame and helper classes.
 *
 * George Anastassakis
 * Knowledge Engineering Lab
 * University of Piraeus
 *
 * Contains an implementation of the Virtual Bounding-Box
 * methodology by G. Anastassakis, 2007.
 */
```

```
public class ExplorerFrame extends JFrame {
    speech.SpeechModule speechmodule;
```

8. Στη μέθοδο sense της κλάσης ExplorerFrame προσθέτουμε τον παρακάτω κώδικα μετά την κλήση στη μέθοδο RefreshUI().

```
    if (speechmodule == null){
        speechmodule = new SpeechModule(this.ownName);
    }
    for (String response : speechmodule.Listen(symbols)){
        this.speechmodule.Speak(response);
        act("act Talk now "+response,false);
    }
```

```
        input = in.readLine();
        if (input.equals("?COMMAND")) {
        } else {
            Log.error(this, "sense", "Invalid prompt \"" + input + "\" waitin
            socket.close();
            result = false;
            status = Status.IDLE;
            refreshUI();
        }
```

```
        if (speechmodule == null){
            speechmodule = new SpeechModule(this.ownName);
        }
        for (String response : speechmodule.Listen(symbols)){
            this.speechmodule.Speak(response);
            act("act Talk now "+response,false);
        }
    } catch (IOException ex) {
```

Επεξήγηση του SpeechModule

Το SpeechModule είναι το κύριο τμήμα της υλοποίησης που έχει γίνει για τη δυνατότητα της ομιλίας. Σε αυτό περιλαμβάνεται η παραγωγή φωνής, η κατανόηση γλώσσας καθώς και η δημιουργία απάντησης (ή με κατάλληλη τροποποίηση τα actions τα οποία θα πρέπει να κάνει ο πράκτορας).

Ο πράκτοράς μας πρέπει να κάνει χρήση της κλάσης SpeechModule η οποία περιέχει τις εξής μεθόδους:

- Listen: δέχεται σαν είσοδο τη λίστα με ότιδήποτε κάνει sense ο agent. Βάσει αυτής καταλαβαίνει αν και σε ποιόν πρέπει να απαντήσει. Η μέθοδος αυτή επιστρέφει στον agent όλα τα actions που θα πρέπει να πραγματοποιήσει.
- Speak: η μέθοδος αυτή δέχεται σαν string και πραγματοποιεί ανάγνωση του κειμένου.
- Respond(private): Στη μέθοδο αυτή γίνονται όλες οι διαδικασίες με βάση τις οποίες επιστρέφονται τα actions που πρέπει να κάνει ο agent. Είναι η μέθοδος που πρέπει να τροποποιήσουμε αν θέλουμε συγκεκριμένα actions για το κάθε σενάριο.

Elisa

Η eliza σαν σύστημα που έχει χρησιμοποιηθεί είναι μία ακριβής υλοποίηση της eliza που είχε περιγράψει ο Weizenbaum το 1966. Οι απαντήσεις και οι φράσεις που καταλαβαίνει ορίζονται σε ένα script του οποίου το path υπάρχει ορισμένο στο επίπεδο του speechmodule. Το script μπορεί και αναλύει αυτό που ειπώθηκε, κανονίζει ποια απάντηση θα δώσει από μία λίστα με κάποια κριτήρια ταυτοποίησης, και μπορεί να αποθηκεύει στη μνήμη και να επαναχρησιμοποιεί τμήματα από φράσεις που ειπώθηκαν νωρίτερα στη συνομιλία.

Ένα παράδειγμα είναι:

```
decomp: * i was *
reasm: Were you really ?
reasm: Why do you tell me you were (2) now ?
reasm: Perhaps I already know you were (2).
```

Εδώ λοιπόν όταν κάποιος αναφέρει μία πρόταση που περιέχει το τμήμα «i was»(token), η ελίζα θα σπάσει την πρόταση σε 3 μέρη. Το τμήμα πριν το token, το ίδιο το token, το τμήμα μετά. Στη συνέχεια επιλέγει μία από τις τρεις απαντήσεις που έχουν οριστεί στο script. Κάποιες είναι στατικές και κάποιες χρησιμοποιούν τμήματα από την προηγούμενη πρόταση.

Αν για παράδειγμα κάποιος είχε πει:

Q:Last summer I was in Greece

Μία πιθανή απάντηση από την ελίζα θα ήταν η:

```
Why do you tell me you were (2) now ?
```

Αντικαθιστώντας το (2) με το στοιχείο μετά το token. Δηλαδή:

```
Why do you tell me you were in Greece now ?
```

Ο αριθμός στην παρένθεση αντιπροσωπεύει το "*" στην πρόταση που ταυτοποιήθηκε.

Περισσότερα για το script και για τη χρήση της μνήμης μπορούμε να βρούμε στη διεύθυνση <http://chayden.net/eliza/instructions.txt>

Αναλυτικά οι κλάσεις που περιέχει η ελίζα είναι αυτές:

- [Decomp.java](#) – Κλάση που ορίζει ένα κανόνα αποσύνθεσης.
- [DecompList.java](#) – Λίστα με κανόνες αποσύνθεσης.
- [Key.java](#) – Το κλειδί ταυτοποίησης.
- [KeyList.java](#) - Λίστα με τα κλειδιά ταυτοποίησης.
- [KeyStack.java](#) – Η δομή στοίβας που μπαίνουν τα κλειδιά ταυτοποίησης για πιο γρήγορη αναζήτηση.

- [Mem.java](#) – Εδώ αποθηκεύονται στη μνήμη προτάσεις για χρήση αργότερα.
- [PrePost.java](#) – Εδώ αποθηκεύονται τα τμήματα πριν και μετά το token.
- [PrePostList.java](#) – Λίστα από τμήματα πριν και μετά το token.
- [ReasembList.java](#) – η λίστα από τους κανόνες επανασύνθεσης.
- [EString.java](#) – Βοηθητικές λειτουργίες για ταυτοποίηση και σύνθεση των strings.
- [SynList.java](#) – λίστα με συνώνυμα (χρησιμοποιούνται στην ταυτοποίηση).
- [WordList.java](#) – λίστα από λέξεις για τις οποίες θα αναζητηθούν συνώνυμα.
- [Eliza.java](#) – η κεντρική κλάση της eliza που χρησιμοποιούμε στο Speech generator.
- [ElizaApp.java](#) (δεν χρησιμοποιείται) Ένα γραφικό Interface για την Eliza

Περισσότερες πληροφορίες για την υλοποίηση της Eliza σε Java μπορούμε να βρούμε στη διεύθυνση <http://chayden.net/eliza/Eliza.html>

FreeTTS

Για παραγωγή ομιλίας γίνεται χρήση του VoiceGenerator που με τη σειρά του χρησιμοποιεί το FreeTTS της SUN.

Πληροφορίες για το FreeTTS Μπορούμε να βρούμε στη διεύθυνση:

<http://freetts.sourceforge.net/docs/index.php>

Χρήση της ομιλίας

Αν θέλουμε να κάνουμε χρήση της ομιλίας (είτε μέσω του client είτε από κώδικα) θα πρέπει να επιλέξουμε πρώτα το action point με τον effector. Αυτό γίνεται με το action του agent

Action:”**act Talk select now mouth**”

Στη συνέχεια αν θέλουμε ο agent να πει κάτι πρέπει να ορίσουμε το action ως εξής.

Action:”**act Talk now hello world**”