



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής  
Πρόγραμμα Μεταπτυχιακών Σπουδών  
«Προηγμένα Συστήματα Πληροφορικής»

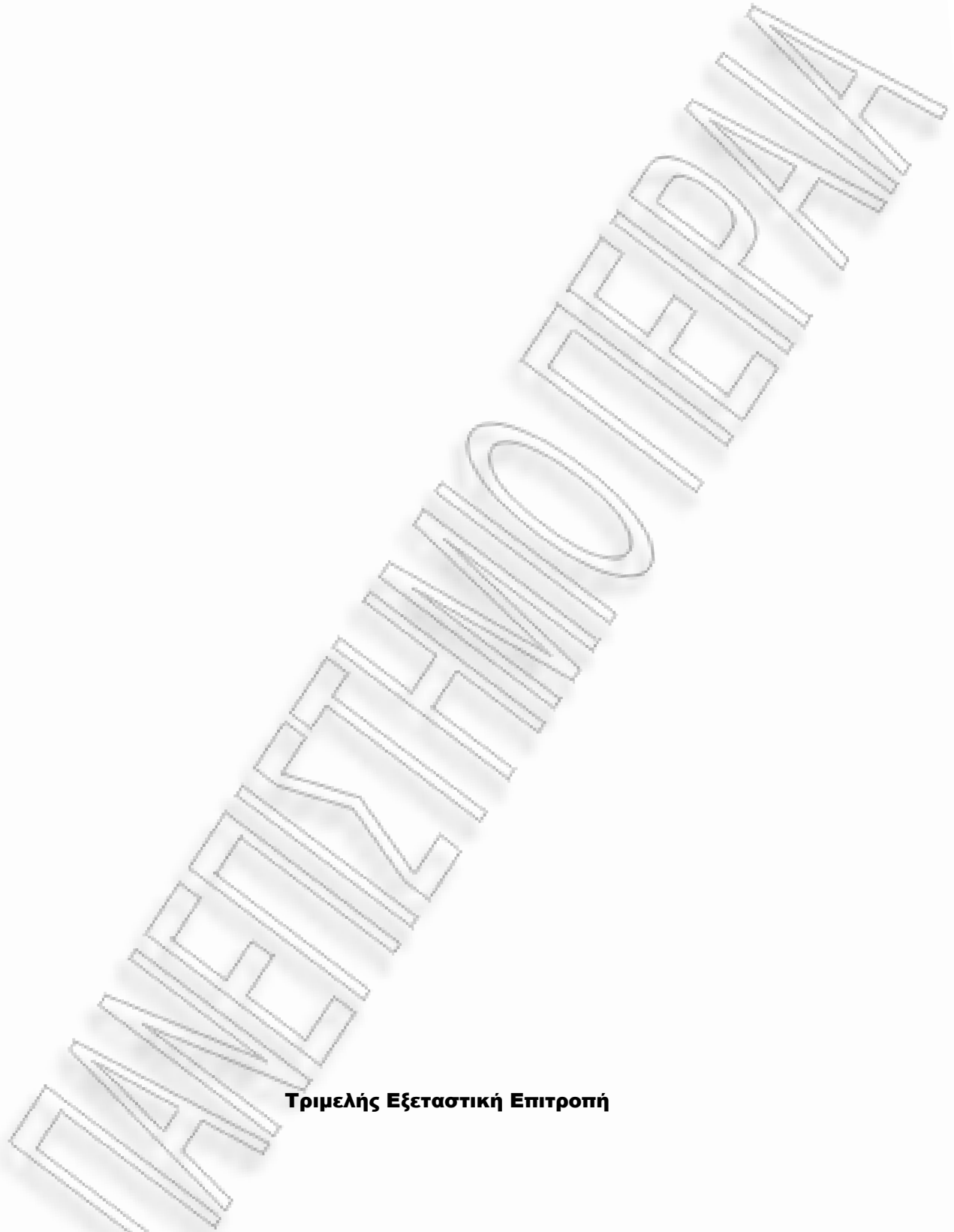
Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	<b>Ευφυής Εικονικός Πράκτορας με δυνατότητα ομιλίας με βάση την αναπαράσταση REVE</b>
Όνοματεπώνυμο Φοιτητή	<b>Γιακουμής Ευστάθιος του Ιωάννη</b>
Αριθμός Μητρώου	<b>ΜΠΣΠ/07051</b>
Κατεύθυνση	<b>Ευφυείς Τεχνολογίες Επικοινωνίας Ανθρώπου - Υπολογιστή</b>
Επιβλέπων	<b>Παναγιωτόπουλος Θεμιστοκλής, Καθηγητής</b>

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

Ημερομηνία Παράδοσης **Απρίλιος 2011**

---



**Τριμελής Εξεταστική Επιτροπή**

Παναγιωτόπουλος Θεμιστοκλής  
Καθηγητής

Δεσπότης Δημήτριος  
Καθηγητής

Αποστόλου Δημήτριος  
Επίκουρος Καθηγητής

Σκοπός της εργασίας αυτής είναι η παρουσίαση και η δράση εικονικών πρακτόρων με εφυή συμπεριφορά μέσα σε ένα ολοκληρωμένο και δυναμικό εικονικό κόσμο. Ένας εικονικός πράκτορας είναι η αναπαράσταση μιας ενσώματης οντότητας με την ικανότητα αυτόνομης συμπεριφοράς στα πλαίσια ενός εικονικού κόσμου. Στα πλαίσια αυτής της εργασίας χρησιμοποιώντας την πλατφόρμα υλοποίησης jReve επεκτείνουμε τις δυνατότητες ενός εικονικού πράκτορα ώστε να περιλαμβάνει την ικανότητα εκφοράς και κατανόησης λόγου. Πιο συγκεκριμένα μπορεί πλέον ο εικονικός μας πράκτορας να αντιλαμβάνεται την ομιλία ενός άλλου εικονικού πράκτορα, επεξεργάζεται αυτή την πληροφορία και να την αποθηκεύει για μελλοντική χρήση. Έχει επιπλέον την ικανότητα να αναπαριστά τόσο οπτικά όσο και ακουστικά την ομιλία του. Ένας εφυής εικονικός πράκτορας πρέπει να έχει και ένα μοντέλο παραγωγής συμπεριφοράς. Στην παρούσα εργασία παρουσιάζουμε ένα πράκτορα που χρησιμοποιεί ως μοντέλο εφυούς συμπεριφοράς το πρόγραμμα Eliza, ένα πρόγραμμα που σαν σκοπό είχε την επίδειξη συνομιλίας σε φυσική γλώσσα μεταξύ υπολογιστή και χρήστη. Για να αξιολογήσουμε τα αποτελέσματα της εργασίας μας εισάγαμε τον πράκτορα Eliza σε ένα εικονικό περιβάλλον που αναπαριστά ένα μπαρ, σχεδιασμένο σε VRML και φορτωμένο στην πλατφόρμα jReve με τη χρήση της γλώσσας ορισμού αναπαράστασης εικονικών κόσμων VERL. Στον εικονικό αυτό κόσμο ο εισάγεται και ένας άλλος πράκτορας και γίνεται μια συνομιλία επιδεικνύοντας την δυνατότητα αντίληψης, κατανόησης και εκφοράς λόγου τόσο από τον πράκτορα Eliza όσο και τον άλλο πράκτορα.

Λέξεις Κλειδιά: REVE, VERL, ELIZA, FreeTTS, TTS Systems, Εικονικοί πράκτορες , Σύνθεση Ομιλίας, chatter bots, εικονικά περιβάλλοντα

The purpose of this project is the presentation and interaction of virtual agents with intelligent behavior inside a complete and dynamic virtual world. A virtual agent is the representation of an embodied entity with the capacity of autonomous behavior in the context of a virtual world. In this project using the development platform jReve we extended the abilities of our agent with the ability of speech and understanding of speech. In detail our virtual agent can now understand the speech of another virtual agent can process this information and store it for future use. It also has the ability to manifest visually and in sense of hearing its speech. An intelligence virtual agent must have a behavioral model. Our agent uses as a behavioral model the program Eliza, a program with the purpose of demonstration a natural language conversation between man and machine. To evaluate the results of our project we enter agent Eliza in a virtual environment representing a bar, designed in VRML and loaded into jReve platform using VERL, a language for defining virtual worlds representations. In this virtual world we enter another agent and a conversation is made demonstrating the ability of perception, understanding and speaking of both agents.

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

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής μου εργασίας κ. Θεμιστοκλή Παναγιωτόπουλο, για την καθοδήγηση και την υπομονή του. Επίσης θέλω να ευχαριστήσω τον Γιώργο Αναστασάκη, γιατί χωρίς τη δουλειά που έκανε για το διδακτορικό του δεν θα μπορούσε να γίνει αυτή η εργασία. Τέλος να ευχαριστήσω τον συμφοιτητή μου Δημήτρη Γεωργίτση για την άφογη συνεργασίας μας και την πολύτιμη βοήθεια του.

## Πίνακας περιεχομένων

Ευχαριστίες.....	4
1.Εισαγωγή.....	8
2. Εικονική Πραγματικότητα και Εικονικά Περιβάλλοντα.....	11
2.1 Εφαρμογές εικονικών περιβαλλόντων.....	13
3. Αναπαράσταση Εικονικών αντικειμένων.....	17
3.1 Εισαγωγή.....	17
3.2 Item aspect.....	17
3.2.1 Το item ως σύνολο item aspects.....	17
3.2.2 Physical aspect.....	18
3.2.3 Semantic aspect.....	19
3.2.4 Access aspect.....	21
4. Αναπαράσταση Εικονικών πρακτόρων.....	24
4.1 Εισαγωγή.....	24
4.2 Ο εικονικός πράκτορας ως item.....	24
4.3 Perception aspect.....	25
4.3.1 Λήψη δεδομένων αντίληψης.....	26
4.3.2 Επεξεργασία δεδομένων.....	26
4.3.3 Αποθήκευση γνώσης.....	27
4.4 Activity aspect.....	27
4.4.1 Action.....	28
5. VERL.....	29
5.1 Εισαγωγή.....	29
5.2 Σύνταξη και σημασιολογία.....	29
5.3 Το στοιχείο «Item» ως σύνολο στοιχείων «item aspects».....	31
5.3.1 Το στοιχείο «psysicalAspect».....	31
5.3.2 Το στοιχείο «semanticAspect».....	32
5.3.3 Το στοιχείο «accessAspect».....	32
6. Πλατφόρμα jReve.....	33
6.1 Εισαγωγή.....	33
6.2 Extensible 3D Graphics (X3D) και Xj3D.....	33
6.3 Κύκλος ζωής.....	34
7. Μοντελοποίηση Εικονικού Πράκτορα.....	35
7.1 Εισαγωγή.....	35
7.2 Το εικονικό σώμα του πράκτορα.....	36

7.2.1 Physical Aspect .....	37
7.2.2 Semantic aspect .....	38
7.2.3 Access aspect .....	38
7.3 Activity Aspect .....	39
8. Σύστημα παραγωγής συμπεριφοράς .....	40
8.1 Εισαγωγή.....	40
8.2 Υπολογιστική Γλωσσολογία ή Επεξεργασία Φυσικής Γλώσσας .....	41
8.2.1 Υπολογιστική Γλωσσολογία .....	41
8.2.3 Η παράδοση της Επεξεργασίας Φυσικής Γλώσσας.....	42
8.2.4 Η επιβολή της Δομής.....	42
8.3 Chatterbots .....	43
8.3.1 Μέθοδος Λειτουργίας .....	43
8.3.2 Ιστορική Αναδρομή.....	44
8.4 Το πρόγραμμα Eliza.....	44
8.4.1 Αρχές Λειτουργίας .....	45
8.5 Τεχνολογία και Συστήματα Μετατροπής Κειμένου σε Ομιλία.....	48
8.5.1 Εισαγωγή.....	48
8.5.2 Java Speech Api .....	49
8.5.3 FreeTTS .....	49
8.5.4 VoiceGenerator .....	50
9. Case Study .....	51
9.1 Εισαγωγή.....	51
9.2 Εικονικό Bar.....	51
9.2.1 Εισαγωγή.....	51
9.2.2 Μοντελοποίηση εικονικού κόσμου.....	51
9.2.3 Παράδειγμα ολοκληρωμένης μοντελοποίησης item.....	54
9.3 Σενάριο και Παρουσίαση .....	56
10. Συμπεράσματα .....	59
10.1 Μελλοντικές Προσεγγίσεις.....	60
11. Βιβλιογραφία .....	60
12. Παράρτημα .....	62
12.1 Το script της Eliza.....	62
12.2 Ο πλήρης ορισμός του εικονικού bar στη γλώσσα VERL .....	75

## Πίνακας Εικόνων

3.1. Physical Aspect .....	16
3.2. Semantic Aspect .....	17
3.3 Access Aspect .....	19
4.1 Perception Aspect .....	22
4.2. Activity Aspect .....	25
7.1 Το σώμα του εικονικού πράκτορα .....	35
7.2 Ο κόμβος Text .....	37
7.3 Ο εικονικός πράκτορας μέσα στο ReveWorlds .....	39
8.1 Μπλοκ διάγραμμα ενός τυπικού διαλογικού συστήματος .....	40
8.2 Κατηγορίες της τεχνολογίας ομιλίας .....	41
9.1 Αναπαράσταση του εικονικού μπαρ .....	52
9.2 Αναπαράσταση του εικονικού μπαρ από άλλη γωνία .....	52
9.3 Πίνακας των items .....	54
9.4 Το item Painting_01 σε VRML .....	55
9.5 Το item Painiting_01 στο ReveWorlds .....	55
9.6 Πρώτο στιγμιότυπο από το σενάριο .....	57
9.6 Δεύτερο στιγμιότυπο από το σενάριο .....	58
9.6 Τρίτο στιγμιότυπο από το σενάριο .....	59

## 1.Εισαγωγή

Τα εικονικά περιβάλλοντα είναι ένα σχετικά νέο επιστημονικό πεδίο με αντικείμενο την αναπαράσταση εικονικών κόσμων με τη βοήθεια ηλεκτρονικών υπολογιστών. Η προσπάθεια να αναπαρασταθούν κόσμοι φαντάστικοι ή μη δεν είναι αντικείμενο αποκλειστικά της επιστήμης των υπολογιστών, αλλά ανήκει σε ένα μεγάλο εύρος επιστημονικών πεδίων και τεχνών. Τόσο η ζωγραφική όσο και ο κινηματογράφος για παράδειγμα είναι προσπάθειες αναπαράστασης ενός κόσμου. Από την σκοπία της πληροφορικής, με την ραγδαία εξέλιξη της τεχνολογίας στον τομέα των γραφικών κυρίως αλλά όχι μόνο, έχει δώσει τη δυνατότητα για μοντέλοποίηση και αναπαράσταση κόσμων με όλο και μεγαλύτερη πιστευτότητα και ρεαλιστικότητα. Είτε πρόκειται για μοντέλοποίηση ενός κόσμου για ερευνητικούς σκοπούς είτε για ψυχαγωγία είναι πλέον φανερό πως τα εικονικά περιβάλλοντα θα διαδραματίζουν συνεχώς και μεγαλύτερο ρόλο.

Ένα υπολογιστικό σύστημα που έχει ως στόχο την αναπαράσταση ενός κόσμου, την παρουσίασή του στον χρήστη και την αλληλεπίδραση του χρήστη με τον κόσμο αυτό, ονομάζεται εικονικό περιβάλλον (virtual environment). Ο όρος αυτός είναι κατάλληλος για ένα ανθρωποκεντρικό υπολογιστικό σύστημα προσανατολισμένο στον χρήστη, καθώς η λέξη «περιβάλλον» προϋποθέτει εννοιολογικά την ύπαρξη μίας κεντρικής οντότητας «περί» της οποίας βρίσκεται ο χώρος στον οποίο γίνεται αναφορά. Από την άλλη μεριά, η αναπαράσταση του κόσμου, η οποία προκύπτει ως αποτέλεσμα της λειτουργίας του εικονικού περιβάλλοντος, ονομάζεται εικονικός κόσμος (virtual world). Η αναπαράσταση αυτή μπορεί να κινείται σε πολλαπλά επίπεδα (φυσικής υπόστασης, σημασιολογίας, αλληλεπιδραστικότητας, συμπεριφοράς, κ.α.) και να υλοποιείται χάρη σε διάφορα μέσα (οπτικά, ακουστικά, κ.α.)

Ένα από τα βασικότερα χαρακτηριστικά των εικονικών περιβαλλόντων είναι πως οι κόσμοι που αναπαριστούν κυβερνώνται από νόμους. Έτσι, ένας εικονικός κόσμος μπορεί να έχει βαρύτητα και τριβή, να εμφανίζει συγκρούσεις μεταξύ αντικειμένων με την ανάλογη επιρροή επηρεάζοντας την κίνησή τους, κ.τ.λ. Οι νόμοι που ισχύουν σε έναν ορισμένο εικονικό κόσμο εξαρτώνται τόσο από το θεματικό αντικείμενό του όσο και από τους στόχους της εφαρμογής στην οποία αξιοποιείται η αναπαράσταση. Έτσι, μπορεί να εξομοιώνουν, έστω και σε προσαρμοσμένη βάση, αντίστοιχους νόμους από το φυσικό κόσμο, ή να είναι τελείως «εξωπραγματικοί». Σε κάθε περίπτωση, όμως, το χαρακτηριστικό και σημαντικότερο κριτήριο σχετικά με το σύνολο των νόμων που κυβερνούν έναν εικονικό κόσμο είναι το κατά πόσο το σύνολο αυτό είναι καλά ορισμένο, οπότε και, κατ' επέκταση, το κατά πόσο η ισχύς των νόμων που περιγράφει είναι προβλέψιμη, καθολική και συνεπής. Το χαρακτηριστικό αυτό προσδιορίζει την πιστευτότητα (believability) του εικονικού κόσμου και έχει σημαντικό αντίκτυπο τόσο στη συμπεριφορά ευφυών οντοτήτων στα πλαίσιά του, όσο και στη συνολική εμπειρία που αποκομίζει ένας άνθρωπος-χρήστης κατά την επαφή του με το εικονικό περιβάλλον.

Παράλληλα, ο βαθμός ρεαλισμού (realism) ενός εικονικού κόσμου έχει να κάνει, με απλά λόγια, με το κατά πόσο ο εικονικός αυτός κόσμος «θυμίζει» το φυσικό κόσμο, όπως αυτός γίνεται αντιληπτός από έναν μέσο άνθρωπο-χρήστη. Τα εικονικά περιβάλλοντα σήμερα είναι ικανά για υψηλά επίπεδα ρεαλισμού, επιτρέποντας την αναπαράσταση κόσμων που εξομοιώνουν τον φυσικό με ολόένα και μεγαλύτερη λεπτομέρεια. Αυτό έχει θετικές επιπτώσεις στη συνολική ποιότητα της εμπειρίας από τη χρήση μίας μερίδας εφαρμογών των εικονικών περιβαλλόντων για τις οποίες η ομοιότητα με το φυσικό περιβάλλον είναι σημαντική. Κυριότερο, ίσως, ρόλο στην προσπάθεια επίτευξης ρεαλισμού παίζουν τα υποσυστήματα γραφικών και ήχου των χρησιμοποιούμενων υπολογιστικών συστημάτων.

Ένα ευφύες εικονικό περιβάλλον (intelligent virtual environment, IVE) είναι ένα εικονικό περιβάλλον στο οποίο το στοιχείο της ευφυούς συμπεριφοράς παίζει βασικό ρόλο. Το επιστημονικό πεδίο των Ευφυών Εικονικών Περιβαλλόντων προέκυψε ως αποτέλεσμα της



χρήσης τεχνικών από τον ευρύτερο χώρο της Τεχνητής Νοημοσύνης στα Εικονικά Περιβάλλοντα, ενώ θεωρείται, ακόμα και σήμερα, ότι διανύει τα πρώιμά του στάδια. Δεν είναι τυχαίο, λοιπόν, το ότι τα Ευφυή Εικονικά Περιβάλλοντα θεωρούνται από πολλούς ως «η ένωση Τεχνητής Νοημοσύνης και Εικονικής Πραγματικότητας». Χαρακτηριστικό των συστημάτων αυτών είναι πως, εκτός από στατικά, «άψυχα» αντικείμενα, όπως επίσης και εικονικές αναπαραστάσεις άλλων χρηστών και - στην καλύτερη περίπτωση - έμβιων όντων με αυστηρά προκαθορισμένη, όμως, συμπεριφορά, περιλαμβάνουν και αναπαραστάσεις ευφυών όντων, αληθοφανών ή φανταστικών, με ουσιαστική αυτονομία και δυνατότητες αντίληψης, αλληλεπίδρασης και επικοινωνίας. Σε τέτοια συστήματα, ο ρόλος του εικονικού πράκτορα, όπως αυτός παρουσιάζεται στη συνέχεια, έχει τεράστια σημασία.

Η αναπαράσταση μίας ενσώματης οντότητας με την ικανότητα αυτόνομης συμπεριφοράς στα πλαίσια ενός εικονικού περιβάλλοντος ονομάζεται «εικονικός πράκτορας» (virtual agent, VA). Στη διεθνή βιβλιογραφία συναντά κανείς πλήθος άλλων ονομασιών, όπως «ευφυής εικονικός πράκτορας» (intelligent virtual agent, IVA), «συνθετικός δράστης» (synthetic actor), «εικονικός ηθοποιός» (virtual actor), «εικονικός άνθρωπος» (virtual human) και άλλες, λιγότερο ή περισσότερο προσανατολισμένες σε συγκεκριμένες ιδιότητες και προσεγγίσεις. Το επιστημονικό πεδίο των Εικονικών Πρακτόρων διακρίνεται από μια ισχυρή συνάφεια με το αντίστοιχο των Ευφυών Πρακτόρων (Intelligent Agents). Σύμφωνα με την ασθενή θεώρηση της έννοιας του πράκτορα (weak notion of agency), ένας ευφυής πράκτορας είναι ένα προϊόν λογισμικού το οποίο διακρίνεται από συγκεκριμένα χαρακτηριστικά:

- Αυτονομία (autonomy). Οι ευφυείς πράκτορες λειτουργούν χωρίς την ανάγκη άμεσης παρέμβασης από ανθρώπους-χρήστες, ενώ έχουν (σε κάποιο βαθμό) έλεγχο πάνω στις ενέργειες και την εσωτερική τους κατάσταση.
- Κοινωνικότητα (social ability). Οι ευφυείς πράκτορες επικοινωνούν μεταξύ τους και, σε αρκετές περιπτώσεις, με ανθρώπους-χρήστες.
- Αντίληψη (perception) και δράση (action). Οι ευφυείς πράκτορες αντιλαμβάνονται το περιβάλλον τους και έχουν τη δυνατότητα επιρροής σε αυτό μέσω ενεργειών.
- Θεληματικότητα (deliberation). Η συμπεριφορά των ευφυών πρακτόρων δεν συνίσταται σε απλή αντίδραση σε ερεθίσματα του περιβάλλοντος, αλλά επιδιώκει την επίτευξη συγκεκριμένων, συχνά μακροπρόθεσμων, στόχων.

Τα παραπάνω χαρακτηριστικά είναι απαραίτητα στους εικονικούς πράκτορες. Έτσι, αρκετά συχνά, ο σχεδιασμός και η λειτουργία τους δανείζεται στοιχεία, μεθοδολογίες και τεχνικές από το χώρο των Ευφυών Πρακτόρων. Τέτοια στοιχεία είναι, κατά κανόνα, η λογική συμπερασματολογία (logic-based reasoning), η συμπερασματολογία άνω στο χώρο και το χρόνο (spatial και temporal reasoning), τα μοντέλα αντίληψης και δράσης (perception και action), η συμπεριφορά με προσανατολισμό στο στόχο (goal-oriented behaviour), η μοντελοποίηση συναισθημάτων (emotion modelling) και ο σχεδιασμός ενεργειών (planning).

Η ύπαρξη περιορισμών όπως οι παραπάνω δεν είναι δεσμευτική, αλλά, αντίθετα, εξαρτάται από τις ιδιαίτερες απαιτήσεις της εκάστοτε εφαρμογής καθώς και την σχεδιαστική προσέγγιση που επιλέγεται. Εντούτοις, είναι αμφίβολο το κατά πόσο οι πρακτικές της παντοδυναμίας (omnipotence) και της παντογνωσίας (omniscience) ωφελούν γενικά στο σχεδιασμό και την υλοποίηση των εικονικών πρακτόρων και στη δημιουργία αίσθησης πιστευτότητας και αληθοφάνειας. Αυτό ισχύει έστω και αν χρησιμοποιούνται ως αντίμετρα είτε στις σε μεγάλο βαθμό ελλιπείς ανθρώπινες γνώσεις πάνω στη λειτουργία των έμβιων όντων και

της νοημοσύνης τους, είτε στις περιορισμένες δυνατότητες των διαθέσιμων υπολογιστικών συστημάτων.

Στην παρούσα εργασία χρησιμοποιείται η αναπαράσταση «REVE» (Representation and Execution of Virtual Environments). Η REVE αποσκοπεί στο να εισάγει μία τυπική, πρότυπη, ρεαλιστική, ευέλικτη και σε μεγάλο βαθμό ολοκληρωμένη θεωρητική και σχεδιαστική βάση για μία αποτελεσματική προσέγγιση στο πρόβλημα, λαμβάνοντας υπ' όψη τις ανάγκες τις οποίες αντιμετωπίζει σήμερα ο αντίστοιχος χώρος, αλλά και αυτές που πιθανότατα θα αντιμετωπίσει στο άμεσο μέλλον, δεδομένης της ραγδαίας προόδου στον τομέα των ηλεκτρονικών υπολογιστών, των τηλεπικοινωνιών και της Πληροφορικής γενικότερα. Ένα από τα βασικότερα χαρακτηριστικά της αναπαράστασης αυτής είναι η θεώρηση του εικονικού κόσμου σε ένα εννοιολογικό πλαίσιο το οποίο δεν εξαρτάται από την υλοποίηση αλλά, αντίθετα, από γενικές, υψηλού επιπέδου έννοιες όπως ο κόσμος, το αντικείμενο, η σημασιολογία και η αλληλεπίδραση. Εξ' ίσου βασικό της χαρακτηριστικό είναι ο διαχωρισμός, αλλά και η διατήρηση της συνέπειας, μεταξύ «απόψεων» του εικονικού αντικειμένου, κάθε μία από τις οποίες αφορά συγκεκριμένη πλευρά του εικονικού περιβάλλοντος ως συστήματος.

Η πλατφόρμα ανάπτυξης εφαρμογών πηλοφορείται «jREVE» και αποτελεί μία παραδειγματική υλοποίηση της αναπαράστασης REVE βασισμένη στη γλώσσα Java. Αποτελείται από μία διεπαφή προγραμματισμού εφαρμογών (application programming interface, API) και από πλήθος συστατικών λογισμικού και βοηθημάτων. Σαν αποτέλεσμα, παρέχει μία ολοκληρωμένη βάση για την ανάπτυξη εικονικών περιβαλλόντων σύμφωνα με την αναπαράσταση REVE.

Κινούμενη μεταξύ των επιπέδων της εννοιολογικής προσέγγισης και της υλοποίησης, η γλώσσα VERL (Virtual Environment Representation Language) είναι το σημείο επαφής μεταξύ της αναπαράστασης και της πλατφόρμας. Βασισμένη στη μεταγλώσσα XML, αποτελεί ένα μέσο ορισμού εικονικών κόσμων με ταυτόχρονο προσανατολισμό, τόσο στην ευκολία επεξεργασίας από ανθρώπους, όσο και στην τροφοδότηση υλοποιημένων συστημάτων με τα απαραίτητα για τη λειτουργία τους δεδομένα.

Στα πλαίσια της αναπαράστασης REVE και της πλατφόρμας jREVE ένας εφυής εικονικός πράκτορας έχει ήδη μοντελοποιηθεί με εξελιγμένες δυνατότητες αντίληψης του εικονικού κόσμου και δυνατότητες αλληλεπίδρασης με αυτόν. Παρόλα αυτά υπάρχουν ακόμα ιδιότητες και ικανότητες που δεν έχουν υλοποιηθεί και αξιολογηθεί. Μια από αυτές είναι η δυνατότητα αντίληψης και εκφοράς λόγου τόσο οπτικά όσο και ακουστικά του εικονικού πράκτορα. Η υλοποίηση αυτή συνδυάζει κομμάτια από διάφορες ερευνητικές περιοχές της τεχνητής νοημοσύνης και της επιστήμης των υπολογιστών γενικότερα. Συγκεκριμένα χρειάζεται να εφαρμοστούν τεχνικές από την επεξεργασία φυσικής γλώσσας αλλά και την μετατροπή κειμένου σε φωνή (text to speech).

Η εργασία αυτή έγινε σε συνέργασία με τον Δημήτρη Γεωργίση στο κομμάτι που αφορούσε στη μοντελοποίηση του εικονικού πράκτορα. Στη συνέχεια ο ίδιος ανέλυσε και υλοποίησε την μετατροπή κειμένου σε φωνή ενώ η δική μου συνεισφορά ήταν στην επεξεργασία φυσικής γλώσσας και τον πράκτορα Eliza.

## 2. Εικονική Πραγματικότητα και Εικονικά Περιβάλλοντα

Ο όρος Εικονική Πραγματικότητα χρησιμοποιήθηκε για πρώτη φορά από τον Jaron Lanier (Τζάρον Λέινιερ) το 1989. Ο Lanier είναι ένας από τους πρωτοπόρους της Εικονικής Πραγματικότητας και ιδρυτής της εταιρείας VPL Research (από τη φράση Virtual Programming Languages) η οποία ανέπτυξε μερικά από τα πρώτα συστήματα τη δεκαετία του 1980.

Η Εικονική Πραγματικότητα χρησιμοποιεί ηλεκτρονικούς υπολογιστές, για να δημιουργήσει και να προσομοιώσει υπαρκτά ή μη περιβάλλοντα, από τα οποία ο χρήστης έχει την ψευδαίσθηση ότι περιβάλλεται και στα οποία μπορεί να κινηθεί ελεύθερα, αλληλεπιδρώντας παράλληλα με τα αντικείμενα που περιλαμβάνουν, όπως θα έκανε και στον πραγματικό κόσμο.

Για να είναι όσο πιο πετυχημένη γίνεται η εμπύθιση ενός χρήστη σε ένα περιβάλλον Εικονικής Πραγματικότητας, είναι σημαντικό να απομονωθεί ο χρήστης και οι αισθήσεις του από το πραγματικό κόσμο, επικαλύπτοντας τα ερεθίσματα του πραγματικού κόσμου με αντίστοιχα εικονικά, φτιαγμένα από το σύστημα της Εικονικής Πραγματικότητας. Από τις πέντε αισθήσεις, οι πιο σημαντικές κατά φθίνουσα σειρά είναι η όραση, η ακοή και η αφή. Έτσι είναι πρωταρχικής σημασίας ένα σύστημα Εικονικής Πραγματικότητας να παρέχει στερεοσκοπική εικόνα, δηλαδή δύο εικόνες από διαφορετική οπτική γωνία, μία για κάθε μάτι του χρήστη, έτσι ώστε να δημιουργηθεί η αίσθηση του βάθους στο χώρο. Παράλληλα η ύπαρξη στέρεοσκοπικού ήχου βοηθάει το χρήστη να κατανοεί τι γίνεται γύρω του στον εικονικό χώρο που τον περιβάλλει με πολύ φυσικό τρόπο, ενώ ταυτόχρονα αποκλείει τον χρήστη από τους ήχους του πραγματικού κόσμου, οι οποίοι θα μπορούσαν να καταστρέψουν την εικονική του εμπειρία. Τέλος η αφή, μπορεί να χρησιμοποιηθεί με κατάλληλες συσκευές είτε για να μπορεί ο χρήστης να νιώθει τον κόσμο, π.χ. να ακουμπά ένα αντικείμενο και να νιώθει αντίσταση, είτε για να καθοδηγήσουμε το χρήστη διευκολύνοντάς τον στην εκτέλεση κάποιων συγκεκριμένων ενεργειών, π.χ. μοντελοποίηση τρισδιάστατων αντικειμένων. Αν όλα τα παραπάνω συνδιαστούν και με την ανίχνευση των κινήσεων του χρήστη με κατάλληλες συσκευές ανίχνευσης, έτσι ώστε το εικονικό περιβάλλον να συμπεριφέρεται όπως και το πραγματικό, τότε η όλη εμπειρία που θα αποκτήσει ο χρήστης μπορεί να είναι άκρως ρεαλιστική.

Η ιστορία της Εικονικής Πραγματικότητας, ξεκινά από τις πρώτες στιγμές που ο άνθρωπος θέλησε να εκφραστεί, περίπου 15000 χρόνια π.Χ., με τις προϊστορικές ζωγραφιές σε σπηλιές, όπως το σπήλαιο Λασκώ στη νότια Γαλλία αλλά και με τα διάφορα θρησκευτικά τελετουργικά, που προσπαθούσαν να αγκαλιάσουν όλες τις ανθρώπινες αισθήσεις και να προκαλέσουν δέος και θαυμασμό. Τέτοια παραδείγματα εμπύθισης στην ιστορία της τέχνης υπάρχουν πάρα πολλά. Σε αυτά συμπεριλαμβάνονται το αρχαίο ελληνικό δράμα και τα Διονύσια.

Επίσης κατά τον 5ο αιώνα π.Χ., όπου γίνονται οι πρώτες ιστορικές αναφορές στην τέχνη από τον Πλάτωνα και τους σύγχρονους του, δίνεται ιδιαίτερη προσοχή στη δραματική χρήση της προοπτικής στα σκηνικά των έργων του Αισχύλου και του Σοφοκλή. Μάλιστα ένας από τους πιο καινοτόμους σκηνογράφους, ο Αγάθαρχος, έγραψε σημειώσεις για το πώς χρησιμοποιούσε ο ίδιος την προοπτική σύγκλιση, οι οποίες ενέπνευσαν πολλούς Έλληνες γεωμέτρους εκείνης της εποχής να αναλύσουν μαθηματικά το μετασχηματισμό προβολής. Δυστυχώς δεν έχουν διασωθεί αρχαία ελληνικά σκίτσα ή ζωγραφιές που χρησιμοποιούν την προοπτική, αλλά μπορούμε ίσως να πάρουμε μια γεύση από τα Ρωμαϊκά αντίγραφα, φτιαγμένα μάλλον από Έλληνες ζωγράφους στην Πομπηία του πρώτου αιώνα μ.Χ. Φαίνεται ότι οι Έλληνες και Ρωμαίοι ζωγράφοι έφταναν σε ένα πολύ υψηλό επίπεδο τρισδιάστατου ρεαλισμού στα έργα

τους χρησιμοποιώντας τη διαίσθησή τους, παρά σχεδιάζοντας τα πάντα απο την αρχή με ακρίβεια.

Θα πρέπει να φτάσουμε στο 14ο αιώνα, στη Φλωρεντία, όπου ο Giotto di Bondone ανακάλυψε εντελώς ξαφνικά ένα διαισθητικό τρόπο για την προβολή 3D προοπτικής σε μια 2D επιφάνεια, όπως είναι ο καμβάς. Η μέθοδος αυτή βασίζεται στην οργάνωση των αντικειμένων και των σχέσεων τους σαν να υπάρχει ένα και μοναδικό σημείο θέασης, πράγμα που δημιουργεί μια αίσθηση βάθους. Δεν είναι τυχαίο άλλωστε το γεγονός ότι θεωρείται ο ιδρυτικός της Δυτικής ζωγραφικής.

Η επόμενη εξέλιξη στον τομέα της Εικονικής Πραγματικότητας, έρχεται το 1778, όταν ο Σκωτσέζος ζωγράφος Robert Barker ζωγράφισε μια άποψη της πόλης του Εδιμβούργου 360 μοιρών. Ο καμβάς ύψους περίπου 3 μέτρων τοποθετήθηκε σε ένα κυκλικό δωμάτιο με διάμετρο περίπου 18 μέτρα. Οι θεατές εισέρχονταν στο κέντρο του δωματίου και βρίσκονταν περικυκλωμένοι απο τη σκηνή. Ο Barker αρχικά ονόμασε την εφεύρεσή του 'la nature á coup d'oeil', αλλά σε διαφημίσεις του 1791 για μια αντίστοιχη ζωγραφιά για το Λονδίνο, χρησιμοποίησε τον όρο 'Πανόραμα', απο τις ελληνικές λέξεις παν και όραμα.

Στα μέσα του 18ου αιώνα, η νέα τεχνολογία της φωτογραφίας γίνεται δημοφιλής, δίνοντας τη δυνατότητα στον άνθρωπο για πρώτη φορά στην ιστορία του να παίρνει και να ξαναδημιουργεί πιστά αντίγραφα εικόνων, γεωγραφικών τόπων, ανθρώπων ή γεγονότων. Το 1833 ο Wheatstone, επινόησε τη στερεοσκοπική οθόνη, η οποία επέτρεπε τη θέαση στερεοσκοπικών εικόνων, δίνοντας έτσι στο θεατή μια αίσθηση του βάθους. Ο David Brewster επεξεργάστηκε ακόμα περισσότερο την εφεύρεση αυτή το 1844, πράγμα που έκανε δυνατή την δημιουργία ενός προϊόντος ευρείας κατανάλωσης με το όνομα Viewmaster στα μέσα του 19ου αιώνα.

Το 1929 ο Edward Link κατασκευάζει τον πρώτο απλό μηχανικό εξομοιωτή πτήσης, για την εκπαίδευση πιλότων σε εσωτερικούς χώρους και μακριά από πραγματικά αεροπλάνα. Το 1946 κατασκευάζεται ο πρώτος ηλεκτρονικός υπολογιστής, με την ονομασία ENIAC, από το πανεπιστήμιο της Πενσυλβάνια, για τον αμερικάνικο στρατό. Στη δεκαετία του 1950 ο Αμερικανός κινηματογραφιστής Morton Heilig προτείνει "το σινεμά του μέλλοντος", το οποίο θα περικυκλώνει το θεατή με αισθήσεις φτιαγμένες από μηχανήματα και θα μεταφέρει τους θεατές σε μια άλλη διάσταση. Το Sensorama που κατασκευάζεται από τον ίδιο το 1956, προσφέρει μια βόλτα με μοτοσυκλέτα στους δρόμους του Μανχάταν. Χρησιμοποιούνται 3D γραφικά, στερεοσκοπικός ήχος και δονητές. Ο χρήστης του μπορεί επίσης να νοιώσει τον αέρα να τον χτυπάει στο πρόσωπο και να μυρίσει αρώματα της πόλης, όπως γιασεμί και ιβίσκο. Τελικά όμως το Sensorama αποδεικνύεται πολύ επαναστατικό για την εποχή του και αποτυγχάνει.

Το 1961 οι μηχανικοί της εταιρίας Philco Comeau και Bryan δημιουργούν ένα HMD (Head Mounted Display) με την ονομασία Headsight TV Surveillance System απομακρυσμένης παρακολούθησης, με ανίχνευση της κίνησης του κεφαλιού. Για να το επιτύχουν αυτό χρησιμοποιούν ένα ειδικά κατασκευασμένο ηλεκτρομαγνητικό σύστημα. Το HMD αυτό χρησιμοποιήθηκε για την απομακρυσμένη παρακολούθηση επικίνδυνων καταστάσεων. Το 1963 ο διδακτορικός φοιτητής του MIT Ivan Sutherland εισάγει τα αλληλεπιδραστικά γραφικά μέσω υπολογιστή με την εφαρμογή του Sketchpad. Η συγκεκριμένη εφαρμογή χρησιμοποιεί ένα ελαφρύ στυλό για την επιλογή αντικειμένων, παράλληλα με τη χρήση του πληκτρολογίου. Ο ίδιος το 1965 κάνει τα πρώτα βήματα στο να συνδυάσει τους υπολογιστές και τη δημιουργία Εικονικών Κόσμων με την εργασία του "The ultimate display". Στην εργασία αυτή ουσιαστικά περιγράφει ένα δωμάτιο, όπου τα πάντα ελέγχονται από τον υπολογιστή και όλες οι ενέργειες του χρήστη μέσα σε αυτό έχουν τον ίδιο αντίκτυπο που θα είχαν και στον πραγματικό κόσμο. Όπως αναφέρει και ο ίδιος "It is a looking glass into a mathematical wonderland".

Το 1967, ο Fred Brooks επηρεασμένος από την εργασία του Sutherland, ξεκινάει το project GROPE, που έχει σαν στόχο να εξερευνήσει τη χρήση απτικής αλληλεπίδρασης για να βοηθήσει τους βιοχημικούς να "αισθανθούν" τις αλληλεπιδράσεις μεταξύ μορίων πρωτεΐνης. Το 1968, ο Sutherland κατασκευάζει το Sword of Damocles (Σπαθί του Δαμοκλή), ένα HDM το οποίο πήρε το όνομα του από το γεγονός ότι κρεμόταν από το ταβάνι. Χρησιμοποιούσε καθοδικές λυχνίες, είχε μηχανική ανίχνευση της κίνησης του κεφαλιού και πρόβαλλε εικόνες πάνω στον πραγματικό κόσμο. Το εύρος πεδίου του ήταν 40 μοίρες και ο χρήστης μπορούσε να δει σε πραγματικό χρόνο, αντικείμενα σε wireframe μορφή να προβάλλονται πάνω στον πραγματικό κόσμο. Την ίδια χρονιά ο ίδιος και ο David Evans ιδρύουν την εταιρία Evans and Sutherland Computer Corp. (E&S), η οποία ασχολείται με συστήματα οπτικοποίησης τα οποία χρησιμοποιούνται στο στρατό, σε εμπορικούς εξομοιωτές καθώς και σε πλανητάρια και αλληλεπιδραστικά θέατρα. Η εταιρία αυτή εξακολουθεί να υπάρχει!

Το 1972, η εταιρία Atari προσφέρει στο ευρύ κοινό αλληλεπιδραστικά γραφικά πραγματικού χρόνου, με το παιχνίδι Pong. Η ίδια εταιρία στη συνέχεια θα συγκεντρώσει στους κόλπους της πολλούς μελλοντικούς πρωτοπόρους της Εικονικής Πραγματικότητας, όπως είναι οι Alan Kay, Fisher, Bricken, Foster, Laurel, Walsler, Robinett και Zimmerman. Το 1974 ο Myron Krueger δημιουργεί τα πρωτοποριακά του έργα, Metaplay και Videoplase, όπου εξερευνά τις δυνατότητες της αλληλεπίδρασης με τη βοήθεια υπολογιστή. Δημιουργούνται έτσι αλληλεπιδραστικά καλλιτεχνικά περιβάλλοντα, σχεδιασμένα με τέτοιο τρόπο ώστε να δίνουν στους χρήστες τους τη δυνατότητα ελευθερίας επιλογής και προσωπικής έκφρασης. Το 1976 κατασκευάζεται το GROPE II, από τους P. J. Kilpatrick και Fred Brooks, το οποίο παρείχε force feedback (ανάδραση δύναμης) και χρησιμοποιούσε μηχανικούς βραχίονες, για να μεταφερθούν οι κινήσεις των χεριών των χρηστών που χρησιμοποιούσαν το σύστημα, στα άτομα φαρμάκων και να μεταβάλλουν τη συμπεριφορά τους.

## 2.1 Εφαρμογές εικονικών περιβαλλόντων

Τα εικονικά περιβάλλοντα έχουν εφαρμογές σε τομείς όπου απαιτείται η εξομοίωση ενός κόσμου ή μέρος του, η εκδήλωση δυναμικής και συμπεριφοράς ευφών 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 ευδοκίμουν 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.

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 μέτρησης, 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 ευφείς 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 διαλόγου 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 εμπύθισης, 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 εμπειρα 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). Στη 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 Warcraft<sup>1</sup> και EVE Online<sup>2</sup> 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 εμβύθισης 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 Sims<sup>1</sup> 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 βάση ενός απλού μοντέλου συναισθημάτων. Επιπλέον, αξιοσημείωτες είναι και οι δυνατότητές τους για εκδήλωση κοινωνικότητας και πολυεπίπεδων σχέσεων μεταξύ τους.

Ένα άλλο παράδειγμα παιχνιδιού με χαρακτηριστικά εικονικού περιβάλλοντος είναι το Creatures<sup>2</sup>. Το συγκεκριμένο παιχνίδι εξελίσσεται σε ένα εικονικό κόσμο ο οποίος δεν είναι τρισδιάστατος αλλά οι χαρακτήρες που κατοικούν σ' αυτόν διαθέτουν εντυπωσιακά χαρακτηριστικά εξελισμότητας, 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.

### 3. Αναπαράσταση Εικονικών αντικειμένων<sup>1</sup>

#### 3.1 Εισαγωγή

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 διαθεσιμότητα προς αλληλεπίδραση λειτουργικότητα.

#### 3.2 Item aspect

Είναι ανάγκη το item aspect να οριστεί σε δύο επίπεδα, ένα εννοιολογικό και ένα ως δομή δεδομένων. Σε εννοιολογικό 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.

##### 3.2.1 Το item ως σύνολο item aspects

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

<sup>1</sup> Μέρος του παρόντος κεφαλαίου προέρχεται από την Διδακτορική Διατριβή του Γιώργου Αναστασάκη και περιλαμβάνεται σε αυτή την εργασία με την άδεια του.

που απορρέουν από την έννοια της ενσωμάτωσης(θέση, προσανατολισμό, κάληψη 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.

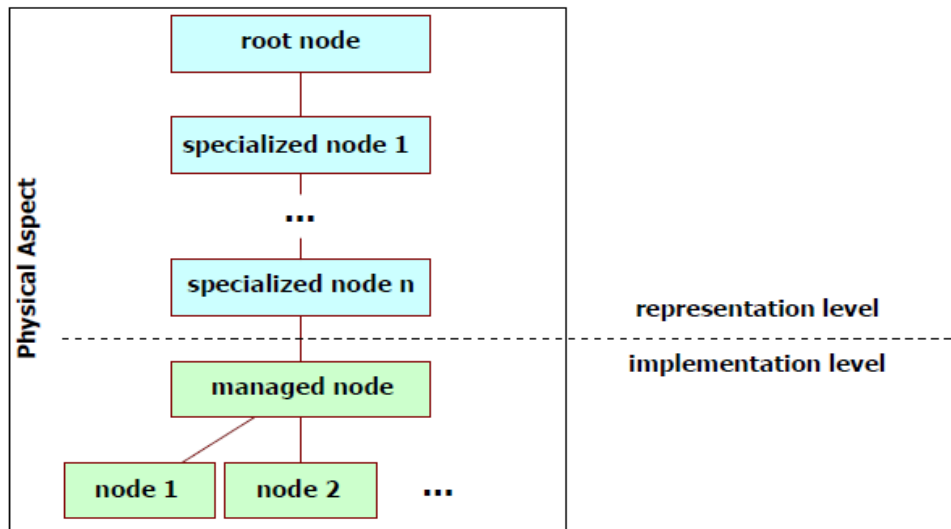
**Access aspect:** Αναπαριστά τη διαθέσιμη από τη μεριά του item λειτουργικότητα, επιτρέποντας την αλληλεπίδραση μεταξύ items και της όποιας πλευράς είναι ικανή για αλληλεπίδραση (π.χ. από εικονικούς πράκτορες). Είναι το μέρος της πληροφορίας για το item που δηλώνει τις ιδιότητες που μπορεί να έχει τις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

### 3.2.2 Physical aspect

Το **physical aspect** αναπαριστά τα οπτικά, ακουστικά και απτικά χαρακτηριστικά ενός item, την πιθανή δομική ή σκελετική ιεραρχία του, καθώς και την δυναμική μεταβολής των χαρακτηριστικών αυτών στο χρόνο. Ουσιαστικά, το **physical aspect** αφορά την ενσώματη φυσική υπόσταση του item ως συστατικού του εικονικού κόσμου, με στόχο τη μορφοποίησή της και την αλληλεπίδραση με αυτή στα πλαίσια του εικονικού περιβάλλοντος με χρήση τεχνικών από το χώρο της Εικονικής Πραγματικότητας.

Καθώς τα υποσυστήματα ενός τυπικού εικονικού περιβάλλοντος, τα οποία κατά κανόνα αναλαμβάνουν τις διαδικασίες διαχείρισης και αξιοποίησης του μοντέλου του εικονικού κόσμου που αναπαρίσταται, βασίζονται σε τεχνικές από το χώρο της Εικονικής Πραγματικότητας, ο σχεδιασμός της εσωτερικής δομής του **physical aspect** - αν και αφηρημένος, όπως θα φανεί παρακάτω - είναι σε μεγάλο βαθμό επηρεασμένος από τη λογική, τις δημοφιλέστερες τεχνικές μοντελοποίησης και μορφοποίησης, καθώς και τις υπολογιστικές ιδιαιτερότητες του χώρου.

Συγκεκριμένα, η πληροφορία που αναπαριστά το **physical aspect** είναι δομημένη όπως στο σχήμα 3.1 παρακάτω, όπου και φαίνεται ότι ο κόμβος-ρίζα του **physical aspect** ακολουθείται από μία ένα-προς-ένα ακολουθία ειδικευμένων κόμβων (*specialized nodes*), η οποία με τη σειρά της ακολουθείται από ένα υποδέντρο με ρίζα ένα συγκεκριμένο κόμβο που ονομάζεται διαχειριζόμενος (*managed node*). Το υποδέντρο με ρίζα τον διαχειριζόμενο κόμβο αντιστοιχεί στο μοντέλο της φυσικής αναπαράστασης του αναπαριστώμενου αντικειμένου όπως είναι διαθέσιμο σε πρωτογενή, ανεξάρτητη του εικονικού περιβάλλοντος, μορφή. Όπως είναι ευνόητο, η δομή του μοντέλου αυτού δεν είναι προκαθορισμένη από την προτεινόμενη αναπαράσταση αλλά εξαρτάται από το αναπαριστώμενο αντικείμενο και τη σχεδιαστική προσέγγιση που έχει επιλεγεί για τη δημιουργία του τρισδιάστατου μοντέλου του.



Εικόνα 3.1 Physical Aspect

Το σύνολο κόμβων του physical aspect ενός item που αποτελείται από τον κόμβο-ρίζα και όλους τους ειδικευμένους κόμβους ονομάζεται επίπεδο αναπαράστασης, καθώς οι κόμβοι αυτοί επιτρέπουν την ισοδύναμη αναπαράσταση της φυσικής υπόστασης του οποιουδήποτε item από το physical aspect κατά τα πρότυπα της προτεινόμενης αναπαράστασης. Αντίθετα, οι υπόλοιποι κόμβοι - αυτοί που ανήκουν στο υποδέντρο με ρίζα τον διαχειριζόμενο κόμβο – αναφέρονται συγκεντρωτικά ως επίπεδο υλοποίησης, καθώς αφορούν το πως το συγκεκριμένο item υλοποιείται γεωμετρικά, δομικά, οπτικά, ακουστικά, κ.τ.λ.

### 3.2.3 Semantic aspect

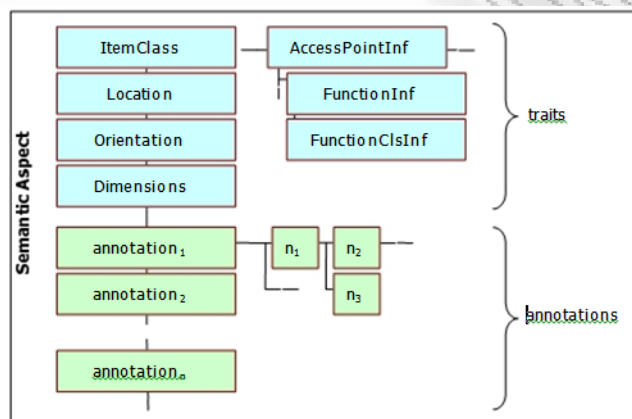
Το semantic aspect αφορά την πληροφορία σχετικά με το item η οποία είναι διαθέσιμη προς αντίληψη. Βασίζεται σε κατάλληλα επιλεγμένο ανά εφαρμογή σύνολο εννοιών προκειμένου να «περιγράψει» το αναπαριστώμενο αντικείμενο, τον ρόλο του, της δυνατότητας αλληλεπίδρασης του καθώς και κατά περίπτωση άλλες πλευρές της «ύπαρξης» του μέσα στον εικονικό κόσμο, αφαιρετικά, σε εννοιολογικό επίπεδο και σε κατάλληλα επιλεγμένη αναπαράσταση ώστε να υποστηρίξει περαιτέρω διαδικασίες συμπερασματολογίας και επεξεργασίας γνώσης. Έτσι, μπορεί να θεωρηθεί ως ένα μέσο πρόσβασης στην σημασιολογία του item ως συστατικού του εικονικού κόσμου.

Η πληροφορία που παρέχει το semantic aspect είναι διαθέσιμη υπό τη μορφή δεδομένων αντίληψης (sensory data). Η σύνταξη και η σημασιολογία των δεδομένων αυτών - η γραμματική της αναπαράστασης στην οποία βασίζεται το semantic aspect για μία συγκεκριμένη εφαρμογή - αποτελεί επιλογή του σχεδιαστή του εικονικού περιβάλλοντος και εξαρτάται από τις απαιτήσεις μοντελοποίησης της κάθε εφαρμογής, το επιλεγμένο εννοιολογικό υπόβαθρο, τις τεχνικές αποθήκευσης γνώσης, συμπερασματολογίας και παραγωγής συμπεριφοράς, τα διαθέσιμα μέσα υλοποίησης, την αξιοποιήσιμη τεχνολογία, κ.τ.λ. Ανεξάρτητα, όμως, από την μορφή των δεδομένων αντίληψης, η πληροφορία που καλούνται να παρέχουν είναι σε κάποιο βαθμό προδιαγεγραμμένη, επιτρέποντας, κατά τα άλλα, στο σχεδιαστή ενός εικονικού κόσμου να συμπεριλάβει και όποιες έννοιες κρίνει απαραίτητο προκειμένου να επιτύχει την επιθυμητή σημασιολογική αναπαράσταση. Συγκεκριμένα, το semantic aspect καλείται να παρέχει, ως τμήμα του συνόλου της πληροφορίας που καθιστά διαθέσιμη, πληροφορία βάσει

συγκεκριμένων, προδιαγεγραμμένων από την αναπαράσταση εννοιών, επικεντρωμένων στο αναπαριστώμενο αντικείμενο ως κεντρική οντότητα αναφοράς.

Σαν αποτέλεσμα, το semantic aspect περιγράφει τόσο κοινά, βασικά χαρακτηριστικά (traits) του item, όσο και αυθαίρετο αριθμό σχολιασμών (annotations) που αντιστοιχούν σε πληροφορία που ορίζεται μεμονωμένα ανά item ανάλογα με τη σημασιολογία του καθενός. Έτσι, η πληροφορία που αναπαριστά το semantic aspect είναι δομημένη όπως στο σχήμα 3.2 παρακάτω. Όπως απεικονίζεται, η δομή συνίσταται σε ένα σύνολο κόμβων που αναπαριστούν τόσο χαρακτηριστικά όσο και σχολιασμούς.

Από τη μία πλευρά, οι κόμβοι της πρώτης κατηγορίας αναπαριστούν συγκεκριμένη πληροφορία σχετικά με το item, η οποία ισχύει και είναι εφαρμόσιμη γενικά, ανεξάρτητη από οποιοδήποτε άλλο χαρακτηριστικό του. Οι κόμβοι αυτοί είναι συγκεκριμένοι. Το ίδιο, όπως θα φανεί αναλυτικά στη συνέχεια, ισχύει τόσο για τη δομή τους σε συντακτικό επίπεδο (και, κατ' επέκταση, τις διαδικασίες πρόσβασης σε πληροφορία τις οποίες επιτρέπουν) όσο και για την αλληλοσυσχέτισή τους με στόχο την αναπαράσταση σύνθετου πληροφοριακού περιεχομένου. Από την άλλη πλευρά, οι κόμβοι της δεύτερης κατηγορίας μπορούν να είναι διαφόρων, μη-προκαθορισμένων τύπων και οποιουδήποτε πλήθους. Επιπλέον, μπορούν να συνιστούν συλλογικά διάφορες δενδροειδείς δομές. Έτσι, απευθύνονται σε ανάγκες αναπαράστασης εξειδικευμένου πληροφοριακού περιεχομένου με βάση τη σημασιολογία κάποιου μεμονωμένου item, κάποιου συγκεκριμένου τύπου item ή για κάποια συγκεκριμένη εφαρμογή. Το σύνολο των κόμβων της κατηγορίας αυτής ορίζεται κατά κανόνα από το σχεδιαστή του εικονικού κόσμου ή, εξαιρετικώς, από το σχεδιαστή κάποιου δεδομένου item ή τύπου item.



Εικόνα 3.2 Semantic Aspect

Όπως θα φανεί παρακάτω, η διάκριση μεταξύ traits και annotations επιτρέπει τόσο μία γενική αντιμετώπιση του item σαν αναπαράσταση αντικειμένων από διαδικασίες συμπερασματολογίας και παραγωγής συμπεριφοράς, όσο και την εξειδικευμένη αντιμετώπισή του ανά αναπαριστώμενο αντικείμενο, μέσω της - προκαθορισμένης ή δυναμικής - διεύρυνσης του εννοιολογικού υποβάθρου της αναπαράστασης, ανάλογα με τις συγκεκριμένες απαιτήσεις της κάθε εφαρμογής. Πρέπει να σημειωθεί ότι η διάκριση αυτή είναι κυρίως σημασιολογική, ενώ σε σχεδιαστικό, συντακτικό επίπεδο επηρεάζει τη μέθοδο πρόσβασης στην πληροφορία που αναπαριστά ο κάθε τύπος κόμβου και μόνο. Συγκεκριμένα, προδιαγράφεται ότι η πρόσβαση στην πληροφορία που αναπαριστούν τα traits, βασίζεται κατ' αποκλειστικότητα σε ένα προκαθορισμένο και εξειδικευμένο πεδίο value, με δεδομένο το είδος της πληροφορία σχετικά με το item την οποία τα traits καλούνται να αναπαραστήσουν, όπως αυτή ορίζεται στη

συνέχεια. Έτσι, τα traits καταλήγουν να δομούνται ως ζεύγη της μορφής (όνομα, τιμή) και, σαν αποτέλεσμα, η πληροφορία που αναπαριστούν να είναι μονοσήμαντη για το καθένα, καθώς και συντακτικά προσβάσιμη με συγκεκριμένο, προκαθορισμένο τρόπο. Το όνομα είναι ένα μοναδικό ανά semantic aspect αλφαριθμητικό (ο προσδιοριστής του κόμβου), ενώ η τιμή μπορεί να είναι οποιοδήποτε τύπου και δεν ορίζεται σαφώς από το semantic aspect αλλά, αντίθετα, εξαρτάται συνολικά από την εφαρμογή και εξατομικευμένα από τη συμπαγή υλοποίηση του κάθε τύπου. Από την άλλη πλευρά, καθώς η πληροφορία που αναπαριστούν τα annotations δεν είναι προδιαγεγραμμένη, η δομή τους δεν καθορίζεται από την προτεινόμενη αναπαράσταση αλλά παραμένει στη διάθεση του σχεδιαστή του εικονικού κόσμου, ο οποίος έχει τη δυνατότητα να την διαμορφώσει με βάση τις ανάγκες αναπαράστασης σημασιολογίας για το item της κάθε εφαρμογής.

### 3.2.4 Access aspect

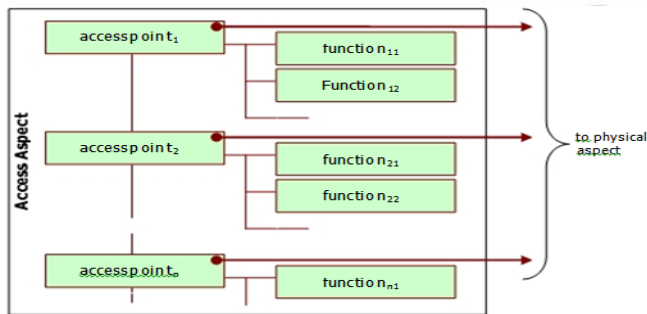
Το access aspect αφορά πληροφορία σχετικά με τις δυνατότητες αλληλεπίδρασης με το item. Σε αντίθεση με το semantic aspect (το οποίο βασίζεται σε κατά το μέγιστο αυθαίρετη, εξαρτώμενη από την εφαρμογή, επιλογή του εννοιολογικού υποβάθρου της αναπαράστασης που χρησιμοποιεί), το access aspect αξιοποιεί ένα συγκεκριμένο, ανεξάρτητο από την εφαρμογή εννοιολογικό μοντέλο για να περιγράφει διαδικασίες αλληλεπίδρασης και, σαν αποτέλεσμα, να θεωρείται ως ένα προτυποποιημένο μέσο πρόσβασης στη διαθέσιμη από το item λειτουργικότητα. Όπως θα φανεί στη συνέχεια, το μοντέλο αυτό περιγραφής αλληλεπιδράσεων χαρακτηρίζεται από σχετική απλότητα. Εντούτοις, είναι ικανό να περιγράψει ένα ιδιαίτερα ευρύ φάσμα αλληλεπιδράσεων. Επιπλέον, παρότι στηρίζεται σε μία εξ' ορισμού αφαιρετική αναπαράσταση, είναι άμεσα υλοποιήσιμο και, κατ' επέκταση, αξιοποιήσιμο από ένα εικονικό περιβάλλον ως υπολογιστικό σύστημα.

Πιο συγκεκριμένα, το access aspect αντιμετωπίζει την έννοια της αλληλεπίδρασης με ένα αντικείμενο ως μία διμερή διαδικασία μεταξύ πλευρών με συγκεκριμένα χαρακτηριστικά, η οποία λαμβάνει χώρα σε προκαθορισμένο αριθμό σταδίων. Η πλευρά που αντιστοιχεί στο item ορίζεται ως accesspoint, ενώ η άλλη πλευρά ορίζεται ασαφώς ως οποιαδήποτε οντότητα είναι ικανή να συμμετέχει στη διαδικασία με βάση τον συνολικό ορισμό του μοντέλου αλληλεπίδρασης όπως αυτός θα ολοκληρωθεί στη συνέχεια. Η έννοια του accesspoint προέρχεται από την ανάγκη κάλυψης των απαιτήσεων συσχετισμού της λειτουργικότητας ενός item με συγκεκριμένα σημεία πρόσβασης.

Με βάση τις απαιτήσεις αυτές, το accesspoint πρέπει κατ' αρχάς να μπορεί να συσχετιστεί με συγκεκριμένο και καλά ορισμένο τμήμα της φυσικής αναπαράστασης του item. Έτσι, ένα accesspoint μπορεί να έχει τη μορφή σημείου, επιφάνειας ή και όγκου, ανάλογα με το επιλεγμένο τμήμα της φυσικής αναπαράστασης του item με το οποίο σχετίζεται και τον αριθμό των χωρικών διαστάσεων στις οποίες εκτείνεται, όπως επίσης και σε συνάρτηση με τις ανάγκες πρόσβασης σε λειτουργικότητα που καλείται να καλύψει. Στα πλαίσια της αντιμετώπισης του item ως σύνολο item aspects και του ορισμού του physical aspect, όπως υιοθετούνται από την προτεινόμενη αναπαράσταση, ο συσχετισμός αυτός επιτυγχάνεται με τη δυνατότητα αναφοράς του accesspoint, ως δομής δεδομένων, σε συγκεκριμένο κόμβο του physical aspect του ίδιου item. Έτσι, το accesspoint αποκτά έμμεσα 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 item. Η ιδιότητα αυτή του accesspoint επιτρέπει, όπως θα φανεί στη συνέχεια, τον ορισμό και τον κατά τη λειτουργία του εικονικού περιβάλλοντος έλεγχο της ισχύος προϋποθέσεων εφικτότητας μίας αλληλεπίδρασης, οι οποίες αφορούν την ανά πάσα στιγμή διάταξη στο χώρο των εμπλεκόμενων στην αλληλεπίδραση πλευρών, όπως η απόστασή τους και ο σχετικός τους προσανατολισμός.

Ο βασικός, όμως, ρόλος του *accesspoint* παραμένει η πρόσβαση στην διαθέσιμη λειτουργικότητα του *item*. Προκειμένου αυτό να είναι εφικτό, η προτεινόμενη αναπαράσταση εισάγει την έννοια του *function*, η οποία αποτελεί μία γενική και αφαιρετική προσέγγιση σε διάφορες δυνατές μεταβολές στο μοντέλο ενός *item*. Έτσι, ένα *function* ορίζεται ως μία διαδικασία μεταβολής συγκεκριμένης πληροφορίας σχετικά με το *item*, με καλά ορισμένο, προβλέψιμο και χρονικά πεπερασμένο τρόπο. Σε σχεδιαστικό επίπεδο, ο ορισμός αυτός αντιστοιχεί σε διαδικασία μεταβολής της τιμής συγκεκριμένου χαρακτηριστικού ενός συγκεκριμένου κόμβου σε οποιοδήποτε από τα *aspects* που απαρτίζουν το μοντέλο του *item*.

Έτσι, με βάση τις έννοιες του *accesspoint*, του *function* και της συσχέτισης μεταξύ *accesspoints* και κόμβων του *physical aspect*, η πληροφορία που αναπαριστά το *access aspect* είναι δομημένη όπως στο σχήμα 3.3 που ακολουθεί. Όπως απεικονίζεται, η δομή αυτή συνίσταται σε ένα αριθμό κόμβων οι οποίοι αναπαριστούν *accesspoints*, κάθε ένας εκ των οποίων σχετίζεται, από τη μία πλευρά, με κάποιο κόμβο του *physical aspect* και, από την άλλη, με ένα αριθμό κόμβων που αναπαριστούν *functions*.



Εικόνα 3.3 Access Aspect

Η συσχέτιση μεταξύ ενός κόμβου ο οποίος αναπαριστά *accesspoint* και ενός κόμβου του *physical aspect* ορίζει, επιπρόσθετα, και ένα μετασχηματισμό μετατόπισης και περιστροφής του *accesspoint* στο σύστημα συντεταγμένων του κόμβου του *physical aspect*. Το χαρακτηριστικό αυτό επιτρέπει τον ορισμό και την επιβολή περιορισμών εφικτότητας αλληλεπιδράσεων με αυξημένη πιστευτότητα και ρεαλιστικότερη αναπαράσταση στον εικονικό κόσμο, καθώς επιτρέπει την τοποθέτηση και τον προσανατολισμό του *accesspoint* στο σύστημα συντεταγμένων του *item* συνολικά, ανεξάρτητα από τα χωρικά χαρακτηριστικά του κόμβου του *physical aspect*, τα οποία είναι πολύ πιθανό να έχουν επιλεγεί έτσι ώστε να εξυπηρετούν αποκλειστικά στόχους οπτικής και γεωμετρικής αναπαράστασης και να είναι ασύμβατα σημασιολογικά με τις ζητούμενες αλληλεπιδράσεις που καλείται το *accesspoint* να καλύψει.

Όπως ορίζεται παραπάνω, ένα *function* είναι μία αναπαράσταση μίας συγκεκριμένης μεταβολής στο μοντέλο του *item*, όπως αυτό υφίσταται ως συλλογή *item aspects* με συγκεκριμένο πληροφοριακό περιεχόμενο η κάθε μία. Η μεταβολή αυτή ορίζεται με βάση το *function class*, το οποίο, στη συνέχεια του κειμένου, θα αναφέρεται είτε ως *function class* είτε ως κλάση *function* και θα δηλώνεται, για κάποιο συγκεκριμένο *function f*, ως  $fc(f)$ . Σε αντιστοιχία με το *item class*, το *function class* επιλέγεται μέσα από μία ιεραρχική ταξινόμηση μοναδικών ανά επίπεδο της ιεραρχίας συμβόλων τα οποία ανήκουν σε κάποιο συγκεκριμένο σύνολο.

Για το *function class* ορίζεται, σε πρώτο στάδιο, ότι *functions* του ίδιου *function class* προκαλούν σημασιολογικά ανάλογες μεταβολές στο μοντέλο του *item*, παραμετροποιήσιμες βάσει συγκεκριμένων ανά *function class* παραμέτρων. Παρ' όλα αυτά, όσο αφορά τη μεταβολή

ως αλγοριθμική (όπως εξυπακούεται) διαδικασία, την εξάρτησή της από παραμέτρους και τα αναμενόμενα αποτελέσματά της, η προτεινόμενη αναπαράσταση δεν υιοθετεί κάποια τυπική γλώσσα ορισμού ή περιγραφής. Οι κυριότεροι λόγοι για αυτό είναι οι παρακάτω:

1. Η δομή των *item aspects* που συνθέτουν το *item* είναι σε μεγάλο βαθμό αφηρημένη. Έτσι, δεν είναι δυνατό να οριστεί στο ίδιο, αφηρημένο και καθαρά σχεδιαστικό επίπεδο, μία γλώσσα ικανή να περιγράφει τυπικά, ολοκληρωμένα και με συνέπεια μεταβολές σε συγκεκριμένο πληροφοριακό περιεχόμενο μοντέλων βασισμένων στα αφηρημένα αυτά *item aspects*. Κάτι τέτοιο θα απαιτούσε τη σύνδεση συντακτικών δομών της γλώσσας (συμβόλων, τελεστών, κ.τ.λ.) με συγκεκριμένες δομές αναπαράστασης πληροφορίας, οι οποίες δεν είναι διαθέσιμες στο επίπεδο που κινούνται σχεδιαστικά οι ορισμοί των *item aspects*.

2. Η υιοθέτηση μίας συγκεκριμένης γλώσσας θα εισήγαγε περιορισμούς στο τι μπορεί να οριστεί και να περιγραφεί ως μεταβολή στο μοντέλο του *item*, όταν ο ίδιος ο σχεδιασμός των εννοιών στις οποίες αυτό βασίζεται προκειμένου να αποκτήσει συγκεκριμένη και συμπαγή υπόσταση έχει γίνει έτσι ώστε να μην επιβάλλονται παρά οι ελάχιστοι περιορισμοί στο τι πληροφορία μπορεί να αναπαρασταθεί. Η δομή του *physical aspect* είναι τέτοια ώστε να μπορεί να φιλοξενήσει οποιοδήποτε είδος γραφήματος σκηνής, ενώ, αντίστοιχα, το *semantic aspect* είναι ικανό να αναπαραστήσει τη σημασιολογία του *item* εκμεταλλευόμενο ένα εύρος προσεγγίσεων αναπαράστασης πληροφορίας και γνώσης. Η γενική αυτή ιδιότητα των σχεδιαστικών προσεγγίσεων πίσω από συγκεκριμένα *item aspects* όπως αυτά ορίζονται από την προτεινόμενη αναπαράσταση, δεν είναι επιτρέπει την δέσμευση από συγκεκριμένες μεθοδολογίες ορισμού μεταβολών στο μοντέλο του *item* και, κατά συνέπεια, αλληλεπιδράσεων με αυτό.

Έτσι, πάντα σε σχεδιαστικό επίπεδο, η προτεινόμενη αναπαράσταση ορίζει ότι η ακριβής σημασιολογία ενός *function* ως διαδικασία μεταβολής του μοντέλου του *item* προσδιορίζεται σαφώς και αποκλειστικά με βάση το αντίστοιχο *function class*. Η ιεραρχία των *function classes* δεν είναι προκαθορισμένη αλλά, ακριβώς όπως και η αντίστοιχη ιεραρχία των *item classes*, ορίζεται για κάθε συγκεκριμένο εικονικό κόσμο από το σχεδιαστή του. Το γεγονός αυτό δίνει την ικανότητα στην προτεινόμενη αναπαράσταση να υποστηρίζει διαφορετικές ανά εικονικό κόσμο μορφές αλληλεπίδρασης με *items* και, σαν αποτέλεσμα, να αντιμετωπίζει εξατομικευμένα τις διαφορετικές ανάγκες μοντελοποίησης της κάθε εφαρμογής.

Με βάση τα παραπάνω και δεδομένης της αντικειμενοστρεφούς προσέγγισης πίσω από το σχεδιασμό των βασικών δομών της προτεινόμενης αναπαράστασης, ορίζεται ότι το *function class* ενός *function* ισοδυναμεί με το πλήρες όνομα της κλάσης - αφηρημένης ή συμπαγούς - η οποία αναπαριστά το *function*. Η ζητούμενη λειτουργικότητα μεταβολής του μοντέλου του *item* με συγκεκριμένο τρόπο εξασφαλίζεται (ή όχι, όπως συζητείται παρακάτω) χάρη σε μία κατάλληλη υλοποίηση της κλάσης, ικανή να καλύπτει τις προϋποθέσεις οι οποίες ορίζονται για το αντίστοιχο *function class*. Έτσι, για μία δεδομένη εφαρμογή, το σύνολο *function classes* ορίζεται ως το σύνολο των ονομάτων πακέτων και κλάσεων που διαθέτει η εφαρμογή με στόχο την αναπαράσταση *functions* όπως αυτά αποτελούν μέρος του συνολικού αντικειμενοστρεφούς σχεδιασμού της.

Αξίζει να σημειωθεί ότι το μοντέλο αυτό δίνει μεγάλη ελευθερία τόσο στο σχεδιαστή της ιεραρχίας των *function classes*, όσο και στον υπεύθυνο για την υλοποίησή τους. Εκτός από τα προφανή πλεονεκτήματα που έχει αυτό, εισάγει και την πιθανότητα ασυμβατότητας μεταξύ της αναπαράστασης, από τη μία πλευρά, και της σημασιολογίας *functions*, από την άλλη, χαρακτηριστικό το οποίο μπορεί να αποδεχθεί ιδιαίτερα ενδιαφέρον σε σενάρια στα οποία εμπλέκονται ευφυείς εικονικοί πράκτορες με δυνατότητες σχεδιασμού ενεργειών και μάθησης.

## 4. Αναπαράσταση Εικονικών πρακτόρων<sup>2</sup>

### 4.1 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.

### 4.2 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

<sup>2</sup> Μέρος του παρόντος κεφαλαίου προέρχεται από την Διδακτορική Διατριβή του Γιώργου Αναστασάκη και περιλαμβάνεται σε αυτή την εργασία με την άδεια του.



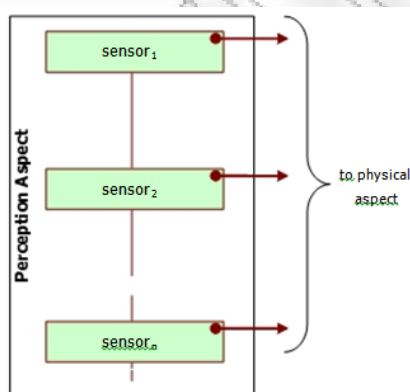
διαθέσιμα ή προαποθηκευμένα δεδομένα αντίληψης. Καλύπτει τις απαιτήσεις δυνατότητων εικονικών πρακτόρων για αντίληψη.

- **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 δυνατότητων 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-προσαρμόσιμα 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 κάποιο από τα δύο (ή και για τα δύο) να μην ορίζονται αισθητήρες και επιδραστές, αντίστοιχα.

### 4.3 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.



### 4.1 Perception Aspect

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 4.1 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.

#### 4.3.1 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 κριτηριαμούς 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.

#### 4.3.2 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.

#### 4.3.3 Αποθήκευση γνώσης

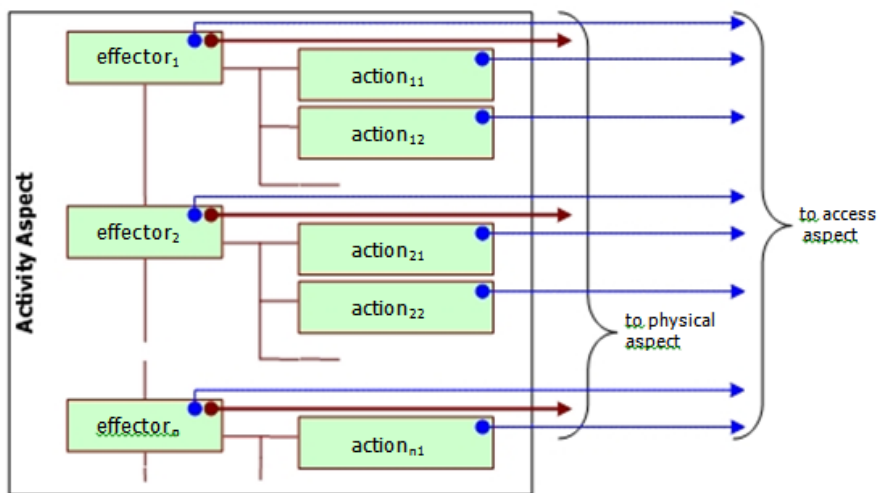
Είναι ιδιαίτερα σημαντικό να τονιστεί πως η γνώση που παράγεται από ένα sensor σαν αποτέλεσμα επεξεργασίας δεδομένων αντίληψης, αναπαρίσταται εσωτερικά και αποθηκεύεται σε συμφωνία με την ίδια γραμματική που ισχύει και για τα δεδομένα που αποθηκεύονται όπως λαμβάνονται από το εικονικό περιβάλλον χωρίς 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.

#### 4.4 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.



## 4.2 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.

### 4.4.1 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 εριορισμούς 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 οποία 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.

## 5. VERL

### 5.1 Εισαγωγή

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.

### 5.2 Σύνταξη και σημασιολογία

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 εννοιολογική, υψηλού επιπέδου περιγραφή ενός εικονικού κόσμου, με την τεχνική, χαμηλού επιπέδου υλοποίησή του, στα πλαίσια ενός συγκεκριμένου εικονικού περιβάλλοντος, διατηρώντας ξεκάθαρα τη διάκριση μεταξύ των δύο πλευρών και μη εμπλεκόντας 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 "" >

```

Στη γλώσσα VERL, αρχικά ορίζουμε ένα στοιχείο «itemGroup» το οποίο ομαδοποιεί items στη βάση σχεδιαστικών επιλογών (χωρικά, σημασιολογικά κτλ). Κάθε τέτοιο «ItemGroup» ορίζεται μονοσήμαντα από το όρισμα «name». Το στοιχείο «ItemGroup» έχει ακόμα δύο ορίσματα που το προσδιορίζουν, το «location» και το «behaviours». Το όρισμα «location» καθορίζει τη φυσική θέση των δεδομένων αναπαράστασης του συνόλου των items που

ομαδοποιεί. Στη πιο συνηθισμένη περίπτωση δείχνει τη διαδρομή του φυσικού αρχείου στο οποίο ορίζεται το item. Το όρισμα «behaviours» μπορεί να πάρει δύο τιμές, true ή false, ανάλογα με το κατά πόσο τα items που ορίζονται μέσα στο «ItemGroup» θα περιέχουν δυναμική συμπεριφορά(scripts, animation,κτλ).

Όπως είπαμε προηγουμένως το στοιχείο «ItemGroup» ομαδοποιεί στοιχεία «item». Κάθε τέτοιο στοιχείο έχει τέσσερα βασικά ορίσματα. Το όρισμα «name» χαρακτηρίζει μοναδικά το item και είναι αυτο που χρησιμοποιείται σε σχεδιαστικό αλλά και λογικό επίπεδο για να αναφερθούμε σε αυτό. Το όρισμα «class» είναι μια τιμή τυπου string που αντιστοιχεί στο στην κλάση του αντικείμενου item, και χρησιμοποιείται κυρίως σε προγραμματιστικό επίπεδο, όταν θέλουμε για παράδειγμα να χρησιμοποιηθεί το συγκεκριμένο item στο pathfinding του πράκτορα μας ή σε άλλες λειτουργίες του. Τέλος τα ορίσματα «fit» και «fitCentre» βοηθούν στην τοποθέτηση του item με συγκεκριμένο τρόπο μέσα στο φυσικό μας κόσμο. Το όρισμα «fit» καθορίζει αν το item θα προσαρμοστεί ενώ το «fitCentre» αν η προσαρμογή θα γίνει με βάση την αρχή του συστήματος συντεταγμένων του ή τη βάση του περικλείοντος σχήματος του.

### 5.3 Το στοιχείο «Item» ως σύνολο στοιχείων «item aspects»

Όπως αναλύθηκε στο κεφάλαιο 3, ένα item είναι το σύνολο των item aspects του, έτσι και στη γλώσσα VERL το κάθε στοιχείο item αποτελείται από τα αντίστοιχα στοιχεία «physical Aspect», «semanticAspect», «accessAspect».

#### 5.3.1 Το στοιχείο «psysicalAspect»

Για κάθε στοιχείο «physicalAspect » ορίζεται το όρισμα «source» η τιμή του οποίου αντιστοιχεί στον κόμβο ο οποίος διατίθεται μέσω του resource του γονικού itemGroup και θα χρησιμοποιηθεί ως κόμβος-ρίζα του physical aspect. Επίσης ορίζεται προαιρετικά ένα στοιχείο «transform» για τον καθόρισμό της φυσικής αναπαράστασης του item.Πιο συγκεκριμένα το στοιχείο «transform» περιέχει τα ορίσματα «translation», «rotation» και «scale» οι τιμές των οποίων καθορίζουν τις αρχικές τιμές θέσης, προσανατολισμού και κλιμάκωσης αντίστοιχα.

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

### 5.3.2 Το στοιχείο «semanticAspect»

Κάθε στοιχείο «semanticAspect» περιέχει ένα ή περισσότερα στοιχεία «symbol» τα οποία ορίζουν τα σύμβολα του semantic aspect του Item. Για κάθε στοιχείο «symbol» ορίζεται ένα μοναδικό ανά semantic aspect όνομα μέσω του ορίσματος «name». Κάθε στοιχείο «symbol» περιλαμβάνει έναν αριθμό στοιχείων «argument» για να οριστούν 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.

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

### 5.3.3 Το στοιχείο «accessAspect»

Για κάθε στοιχείο «accessAspect» ορίζεται ένας αριθμός στοιχείων «accessPoint» τα οποία δηλώνουν τα accessPoints του access aspect του Item. Κάθε «accessPoint» έχει τα ορίσματα «name» που το προσδιορίζουν μοναδικά για κάθε access aspect, «node» ο οποίος ορίζει τον κόμβο του physical aspect του item με το οποίο σχετίζεται το accessPoint και τέλος το όρισμα «visible» για να καθορίσει εάν είναι επιθυμητό να απεικονίζονται οπτικοί σχολιασμοί για το συγκεκριμένο accesspoint.

Επίσης κάθε στοιχείο «accessPoint» περιέχει έναν αριθμό 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



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

## 6. Πλατφόρμα jReve

### 6.1 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.

### 6.2 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 αλληλεπίδρασης 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.

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.

### 6.3 Κύκλος ζωής

Στα πλαίσια ενός ολοκληρωμένου κύκλου ζωής ενός εικονικού περιβάλλοντος, η λογική αξιοποίησης 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 `gene.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.

## 7. Μοντελοποίηση Εικονικού Πράκτορα

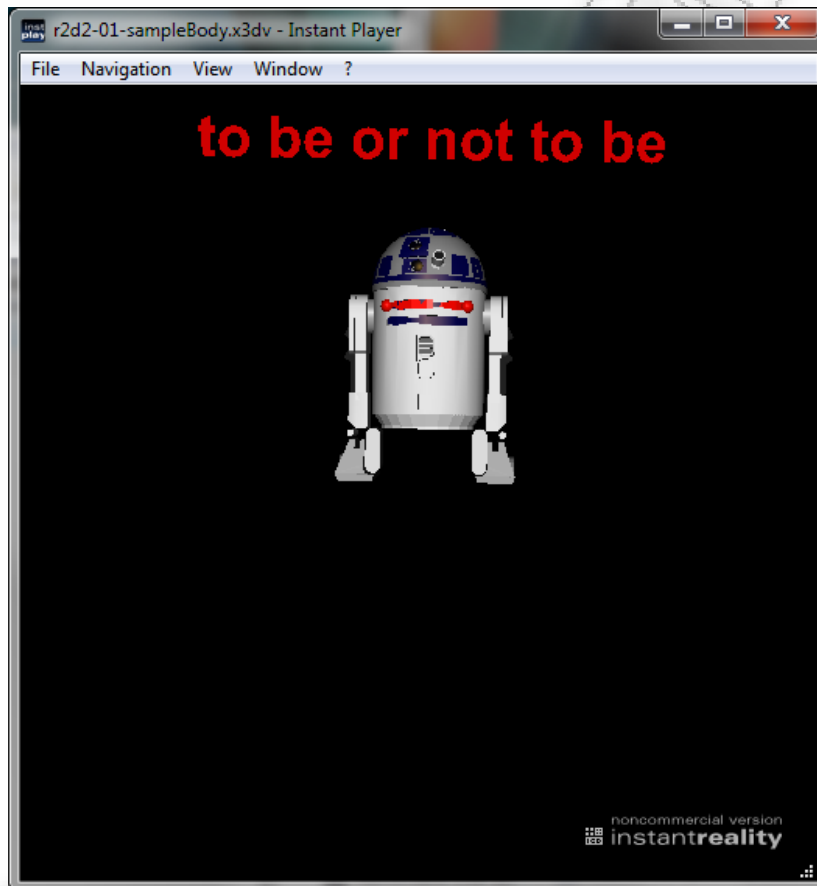
### 7.1 Εισαγωγή

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 υλοποίηση 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.



### Εικόνα 7.1 Το σώμα του εικονικού πράκτορα

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.

### 7.2 Το εικονικό σώμα του πράκτορα

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.

### 7.2.1 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.

```

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"
          }
        }
      }
    ]
  }
]
}

```

### 7.2 Ο κόμβος Text

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.

### 7.2.2 Semantic aspect

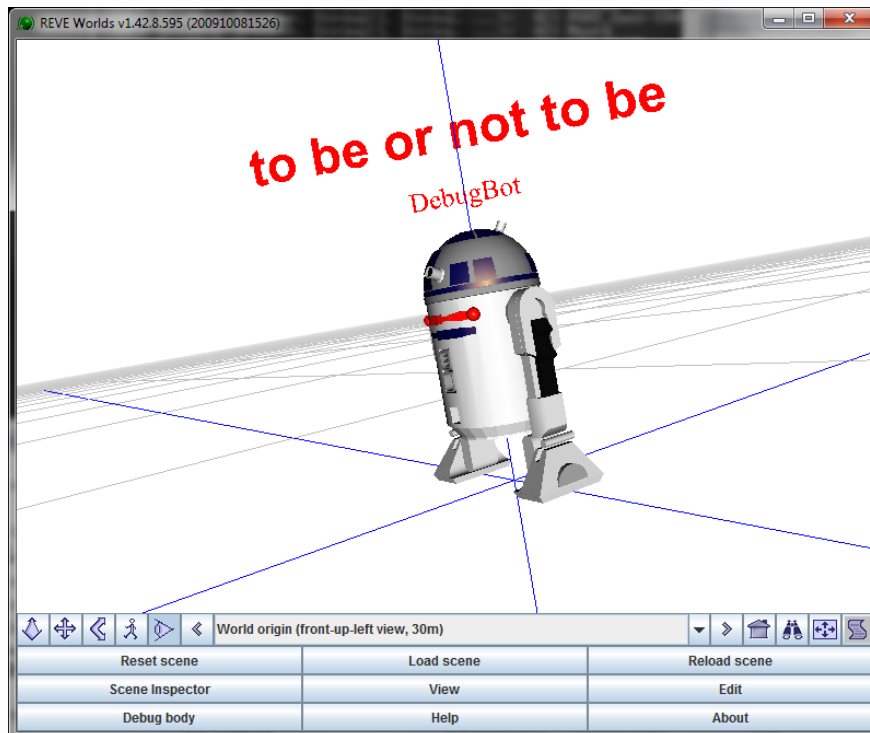
Το επόμενο βήμα στη διαδικασία της μοντελοποίησης είναι οι προσθήκες που πρέπει να γίνουν στο semantic aspect. Εδώ, σε αντίθεση με το physical aspect που κάναμε μια σχεδιαστική επιλογή για να ενσωματώσουμε την αφηρημένη ομιλία, τα πράγματα είναι πιο συγκεκριμένα, μιας και η πληροφορία η οποία είναι αναγκαίο 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 αντικείμενου CompoundSymbol για την διαχείριση της πληροφορίας και ενός X3DConnector αντικείμενου για την σύνδεση με τον κόμβο Text του γραφήματος σκηνής.

### 7.2.3 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.



Εικόνα 7.3 Ο εικονικός πράκτορας μέσα στο ReveWorlds

Πλέον έχουμε ολοκληρώσει την απαραίτητη υποδομή για την λειτουργικότητα του σώματος πράκτορα 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.

### 7.3 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.

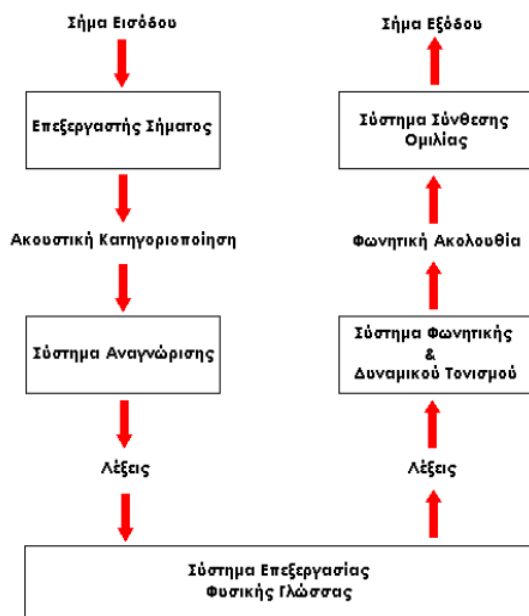
**Action :** Στην περίπτωση μας το action είναι η ομιλία, η εμφάνιση ενός string στην τιμή του αντίστοιχου πεδίου στον κόμβο text που έχουμε δημιουργήσει. Ο βασικός ρόλος της action αυτής είναι να παράγει τα απαραίτητα function execution arguments, δηλαδή το νέο string που θα αναπαριστά την ομιλία, και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

## 8. Σύστημα παραγωγής συμπεριφοράς

Όπως είπαμε παραπάνω για να θεωρηθεί ένας εικονικός πράκτορας ευφυής απαιτείται να περιλαμβάνει ένα μοντέλο παραγωγής συμπεριφοράς. Στην παρούσα εργασία παρουσιάζεται το πρόγραμμα Eliza, ένα πρόγραμμα προορισμένο στην προσομοίωση συνομιλίας σε φυσική γλώσσα με το χρήστη ή έναν άλλο εικονικό πράκτορα με αντίστοιχη δυνατότητα. Επεκτείνοντας τις δυνατότητες τις Eliza εισάγαμε ένα μοντέλο μετατροπής κειμένου σε ομιλία. Η Eliza πλέον δεν εμφανίζει απλά ένα κείμενο στην οθόνη του υπολογιστή αλλά εκφράζει φωνητικά την ομιλία της. Για να γίνει αυτό έχουν υλοποιηθεί διάφορες τεχνολογίες από τον τομέα της τεχνολογίας ομιλίας, οι οποίες αναλύονται παρακάτω.

### 8.1 Εισαγωγή

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 8.1, και αποτελεί την βασική δομή ενός τυπικού διαλογικού συστήματος.



Εικόνα 8.1 Μπλοκ διάγραμμα ενός τυπικού διαλογικού συστήματος.

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 8.2.



Εικόνα 8.2 Κατηγορίες της τεχνολογίας ομιλίας.

## 8.2 Υπολογιστική Γλωσσολογία ή 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.

### 8.2.1 Υπολογιστική Γλωσσολογία

Η μελέτη της γλώσσας αποτελεί αντικείμενο της κλασικής Γλωσσολογίας. Η Υπολογιστική Γλωσσολογία ως διεπιστημονική γνωστική περιοχή βρίσκεται ανάμεσα στην Γλωσσολογία και την Πληροφορική. Ανήκει στις Γνωστικές (Cognitive) Επιστήμες και επικαλύπτεται με την Τεχνητή Νόηση (Artificial Intelligence), κλάδο της Πληροφορικής (Computer Science), που ασχολείται με τα υπολογιστικά μοντέλα της ανθρώπινης γνώσης.

Η Υπολογιστική Γλωσσολογία έχει θεωρητικό και πρακτικό κλάδο. Η θεωρητική Υπολογιστική Γλωσσολογία πραγματεύεται τις τυπικές θεωρίες παράστασης της γλωσσικής γνώσης και τους φορμαλισμούς που έχουν ανάγκη οι άνθρωποι για να παράγουν και να κατανοήσουν τη γλώσσα. Οι θεωρίες αυτές σήμερα έχουν φτάσει σε τέτοια πολυπλοκότητα που

μόνο με υπολογιστές μπορεί κανείς να τις διαχειριστεί. Οι επιστήμονες του κλάδου αυτού αναπτύσσουν τυπικά μοντέλα που εξομοιώνουν απόψεις της ανθρώπινης γλωσσολογίας και τα υλοποιούν σε υπολογιστικά συστήματα. Αυτά αποτελούν τη βάση αποτίμησης και περαιτέρω ανάπτυξης των αντίστοιχων θεωριών. Στην περιοχή αυτή της επιστήμης σημαντικό ρόλο παίζουν ευρήματα από την Γνωστική Ψυχολογία. Επίσης, η Ψυχογλωσσολογία, κλάδος της Ψυχολογίας, εξετάζει τις γνωστικές διαδικασίες που συνιστούν την χρήση της ανθρώπινης γλώσσας.

Η εφαρμοσμένη Υπολογιστική Γλωσσολογία εστιάζεται στα πρακτικά αποτελέσματα της μοντελοποίησης της χρήσης της ανθρώπινης γλώσσας. Οι μέθοδοι, οι τεχνικές, τα εργαλεία και οι εφαρμογές αυτής της περιοχής συχνά συμπεριλαμβάνονται στον όρο (ανθρώπινη) γλωσσική μηχανική (language engineering) ή γλωσσική τεχνολογία (language technology). Παρότι τα σημερινά υπολογιστικά συστήματα απέχουν πολύ από τις ανθρώπινες ικανότητες έχουν πολλές πιθανές εφαρμογές. Στόχος είναι η ανάπτυξη συστημάτων με κάποια γνώση της ανθρώπινης γλώσσας. Τα συστήματα αυτά χρησιμοποιούνται στην καλύτερευση της επικοινωνίας ανθρώπου - μηχανής. Οι σημερινοί υπολογιστές δεν 'καταλαβαίνουν' τη γλώσσα μας αλλά, από την άλλη πλευρά, οι (τεχνητές) γλώσσες προγραμματισμού είναι δύσκολο να μαθευτούν και δεν αντιστοιχούν στη δομή της ανθρώπινης σκέψης.

### 8.2.3 Η παράδοση της Επεξεργασίας Φυσικής Γλώσσας

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.

### 8.2.4 Η επιβολή της Δομής

Τα αυτόματα πεπερασμένων καταστάσεων (finite state automata: FSA) επιστρατεύθηκαν 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 ‘πονοκέφαλο’ 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.

### 8.3 Chatterbots

Το **chatterbot** (ή chatbot) είναι ένας τύπος πράκτορα προσανατολισμένου για συζήτηση, ένα πρόγραμμαυπολογιστή σχεδιασμένο για να προσομοιώνει έναν ευφυή διάλογο με έναν ή περισσότερους ανθρώπους- χρήστες μέσω συστημάτων διεπαφής,πληκτολόγιο, 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.

#### 8.3.1 Μέθοδος Λειτουργίας

Μια καλή κατανόηση μίας συνομιλίας απαιτείται για να συνεχίσει έναν συνεκτικό διάλογο αλλά οι περισσότεροι chatterbots δεν προσπαθούν να κάνουν αυτό. Αντ' αυτού «συζητούν» με την αναγνώριση λέξεων ή των φράσεων κλειδιά από τον ανθρώπινο-χρήστη, οι οποίες τους επιτρέπουν να χρησιμοποιήσουν τις προετοιμασμένες ή προυπολογισμένες απαντήσεις που τους επιτρέπουν να συνεχίσουν να κινούν τη συνομιλία με έναν σχετικά συνεκτικό τρόπο χωρίς να απαιτείται η κατανόηση της ίδιας της συζήτησης ως τέτοιας.

Παραδείγματος χάριν, εάν ένας άνθρωπος δακτυλογραφεί, «είμαι πολύ ανήσυχος τελευταία » το chatterbot μπορεί να προγραμματιστεί για να αναγνωρίσει τη φράση « είμαι» και να απκριθεί με την αντικατάστασή της με την «γιατί είστε» κα ένα ερωτηματικό στο τέλος, δίνοντας την απάντηση, «γιατί είστε πολύ ανήσυχος τελευταία;» Μια παρόμοια προσέγγιση που χρησιμοποιεί τις λέξεις κλειδιά είναι το πρόγραμμα να απαντάει σε κάθε σχόλιο που περιλαμβάνει ονόματα διασήμων με το «Πιστεύω ότι είναι πολύ καλός, εσύ;». Οι άνθρωποι καιειδικότερα αυτοί που δεν έχουν εμπειρία με chatterbots, πολλές φορές βρίσκουν ενδιαφέρουσα μια τέτοια συνομιλία. Κριτικοί των chatterbots ονομάζουν αυτή την τάση Eliza effect.

Μερικά προγράμματα που ταξινομούνται ως chatterbots χρησιμοποιούν άλλες αρχές. Το Jabberwacky για παράδειγμα προσπαθεί να μοντελοποιήσει τον τρόπο που οι άνθρωποι μαθαίνουν καινούργια γεγονότα και γλώσσες. Η Ella χρησιμοποιεί επεξεργασία φυσικής γλώσσας για να δώσει απαντήσεις με νόημα στον χρήστη.

### 8.3.2 Ιστορική Αναδρομή

Ο όρος Chatterbot πρωτοχρησιμοποιήθηκε από τον Michael Mauldin, δημιουργό του πρώτου Verbot, Julia, το 1994 για να περιγράψει τα προγράμματα προσαρμοσμένα για διάλογο. Τα αρχικά chatterbots ήταν η Eliza και ο Parry, ενώ αργότερα δημιουργήθηκαν προγράμματα όπως είναι τα Racter, Verbots, A.L.I.C.E και ELLA.

Η ανάπτυξη του επιστημονικού πεδίου των chatterbots επέφερε την μετάπτωση από την απλή διενέργεια διαλόγου σε άλλες εφαρμογές. Ενώ η Eliza και ο Parry το μόνο που έκαναν ήταν αποκλειστικά μια προσομοίωση διαλόγου, ο Racter χρησιμοποιήθηκε για τη συγγραφή μιας ιστορίας με το όνομα "The Policeman's Beard is Half Constructed", ενώ η ELLA περιλάμβανε μια συλλογή παιχνιδιών και εφαρμογών.

Ο όρος Chatterbot πρωτοχρησιμοποιήθηκε από τον Michael Mauldin, δημιουργό του πρώτου Verbot, Julia, το 1994 για να περιγράψει τα προγράμματα προσαρμοσμένα για διάλογο.

Η σύγχρονη έρευνα γύρω από την τεχνητή νοημοσύνη εστιάζεται σε πρακτικούς στόχους εφαρμοσμένης μηχανικής. Αυτό είναι γνωστό ως αδύνατο AI και διακρίνεται από ισχυρό AI, το οποίο θα απαιτούσε αντίληψη και δυνατότητες συλλογισμού.

Ένας σχετικός τομέας της έρευνας της τεχνητής νοημοσύνης είναι η φυσική γλώσσα. Συνήθως οι αδύνατοι τομείς AI υιοθετούν τις εξειδικευμένες γλώσσες λογισμικού ή προγραμματισμού που δημιουργούνται για αυτό το σκοπό. Παραδείγματος χάριν, μια από τις καλύτερα μοντελοποιημένες αναπαραστάσεις φυσικής γλώσσας chatterbots, A.L.I.C.E., χρησιμοποιεί μια γλώσσα προγραμματισμού αποκαλούμενη AIML που είναι συγκεκριμένη για το πρόγραμμά του, και τους διάφορους κλώνους του, που ονομάζονται Alicebots. Εντούτοις, A.L.I.C.E. είναι ακόμα βασισμένος στο pattern matching χωρίς οποιοδήποτε συλλογισμό ή δυνατότητα αντίληψης. Αυτή είναι η ίδια τεχνική την οποία η ELIZA, το πρώτο chatterbot, χρησιμοποιούσε το 1966.

Η αυστραλιανή επιχείρηση MyCyberTwin ασχολείται επίσης με το ισχυρό AI, επιτρέποντας στους χρήστες της να δημιουργούν και να διατηρούν τις εικονικές προσωπικότητές τους on-line. Το MyCyberTwin.com λειτουργεί επίσης σε μια εταιρική κατεύθυνση, επιτρέποντας σε επιχειρήσεις να δημιουργούν εικονικούς βοηθούς AI. Επίσης τέτοια προγράμματα χρησιμοποιούνται σαν υποδοχές σε τηλεφωνικά κέντρα μεγάλων εταιριών και τραπεζών. Αυτά έχουν την δυνατότητα να κατευθύνουν τους χρήστες στα διάφορα τμήματα ανάλογα με τα ενδιαφέροντα ή τις απαιτήσεις τους, όπως επίσης να συλλέξουν, να επεξεργαστούν και να αποθηκεύσουν πληροφορίες και προσωπικά δεδομένα.

### 8.4 Το πρόγραμμα Eliza

Η Ελιζα 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 σε ένα σχόλιο ή σε μια παλιότερη απάντηση. Ύστερα από αυτή τη διαδικασία η απάντηση που έχει δημιουργηθεί επιστρέφει στο χρήστη.

Η ιστορία του Joseph Weizenbaum είναι ίσως το ίδιο ενδιαφέρουσα με αυτή του Turing. Ένας από τους πρωτοπόρους της επιστήμης των υπολογιστών, ο Weizenbaum ήταν ένας από τους τυχερούς που εργάστηκαν στο εργαστήριο τεχνητής νοημοσύνης του MIT στις αρχές της δεκαετίας του 60. Το μεγαλύτερο επίτευγμά του ήταν η ανάπτυξη της Eliza.

Ο ίδιος έλεγε πως είχε σοκαριστεί όταν παρουσίασε την Eliza (ή αλλιώς "Γιατρός") στο προσωπικό του εργαστηρίου που δεν είχαν σχέση με την επιστήμη των υπολογιστών. Γραμματείς και διοικητικό προσωπικό πίστεψαν ότι το πρόγραμμα ήταν ένας αληθινός

ψυχίατρος και πέρασαν ώρες αποκαλύπτοντας προσωπικά τους προβλήματα με αυτό. Όταν ο Weizenbaum τους ενημέρωσε ότι είχε πρόσβαση στις συνομιλίες αυτές, η γραμματέας του αντέδρασε σε αυτή την παραβίαση της ιδιωτικής της ζωής. Το γεγονός αυτό αλλά και άλλα περιστατικά σόκαραν τον Weizenbaum με την ευκολία που ένα τόσο απλό πρόγραμμα μπορούσε να ξεγελάσει άπειρους και αφελείς χρήστες στο να αποκαλύψουν προσωπικά τους δεδομένα.

Αυτό που παρατηρούσε ο Weizenbaum ήταν ότι οι “ασθενείς” της Eliza πίστευαν πως καταλάβαινε τα προβλήματά τους. Πίστευαν ότι το ρομπότ-θεραπευτής μπορούσε πράγματι να τους βοηθήσει. Το γεγονός αυτό προβλημάτισε πολύ τον Weizenbaum, έφτασε μάλιστα στο σημείο να σταματήσει την Eliza τελείως. Στη συνέχεια έγραψε ένα βιβλίο το 1976 με τίτλο *Computer Power and Human Reason*, στο οποίο επιτίθεται στην τεχνητή νοημοσύνη, στην ίδια την Eliza, και στην έρευνα στον τομέα της επιστήμης των υπολογιστών γενικότερα.

#### 8.4.1 Αρχές Λειτουργίας

Σε γενικές γραμμές η διαδικασία που ακολουθεί το πρόγραμμα είναι απλή, η εισαγωγή (input) του χρήστη διαβάζεται και ανιχνεύεται η παρουσία λέξεων-κλειδιών (keywords). Όταν βρεθεί μια τέτοια λέξη η πρόταση μετασχηματίζεται σύμφωνα με ένα κανόνα (rule) που αντιστοιχεί στη λέξη, ένα ή σε μερικές περιπτώσεις ένας προηγούμενος μετασχηματισμός. Έτσι υπολογίζεται το κείμενο απάντησης.

Στην πραγματικότητα η διαδικασία είναι πιο σύνθετη. Για παράδειγμα οι λέξεις κλειδιά έχουν ένα βαθμό (rank). Η διαδικασία είναι ευαίσθητη σε αυτούς τους βαθμούς, με την έννοια ότι σε μία ανάγνωση θα επιλεγθεί η λέξη-κλειδί με το μεγαλύτερο βαθμό. Επίσης η διαδικασία αναγνωρίζει σημεία στίξης, όπως τελείες και κόμματα. Όταν το πρόγραμμα αναγνώσει ένα από αυτά και μια λέξη-κλειδί έχει ήδη βρεθεί, τότε όλο το υπόλοιπο κομμάτι της πρότασης αγνοείται. Σε περίπτωση που δεν βρεθεί καμία λέξη-κλειδί η φράση στα αριστερά του σημείου στίξης αγνοείται. Αυτό έχει σαν αποτέλεσμα το μετασχηματισμό μόνο απλών φράσεων ή προτάσεων.

Οι λέξεις-κλειδιά και οι αντίστοιχοι κανόνες μετασχηματισμού αποτελούν το script της Eliza. Σημαντική ιδιότητα της Eliza είναι ότι το script είναι δεδομένα, δηλαδή δεν αποτελεί μέρος του προγράμματος καθ'εαυτού. Με αυτό τον τρόπο η Eliza δεν περιορίζεται σε συγκεκριμένα πρότυπα αναγνώρισης ή απαντήσεων, είτε ακόμα σε συγκεκριμένη γλώσσα.

Τα βασικά προβλήματα που πρέπει να αντιμετωπίσει η Eliza είναι τα ακόλουθα :

1. Την αναγνώριση της πιο “σημαντικής” λέξης-κλειδί μέσα στην πρόταση.
2. Την αναγνώριση ενός ελάχιστου πλαισίου μέσα στο οποίο υπάρχει η λέξη-κλειδί. Για παράδειγμα υπάρχει η λέξη “are” μετά τη λέξη-κλειδί “you” ;
3. Την επιλογή του κατάλληλου κανόνα μετασχηματισμού και βέβαια τον ίδιο το μετασχηματισμό της πρότασης.
4. Ένα μηχανισμό που θα επιτρέπει στην Eliza να απαντά με κάποια ευφυία στην περίπτωση που η πρόταση δεν περιέχει κάποια λέξη-κλειδί.

Το κεντρικό ζήτημα είναι σαφώς ο χειρισμός του κειμένου εισαγωγής και στον πυρήνα του ζητήματος αυτού είναι ο κανόνας μετασχηματισμού που όπως είπαμε συσχετίζεται με κάποια συγκεκριμένη λέξη-κλειδί. Ένας κανόνας μετασχηματισμού συνίσταται πρώτον στην αποσύνθεση ενός string με βάση την ικανοποίηση κάποιων απαιτήσεων και δεύτερον στην ανασύνθεση του αποσυντεθημένου string με βάση κάποιους κανόνες.

Σκοπός μας δεν είναι να αναλύσουμε την Eliza πλήρως αλλά για να γίνει κατανοητή η λειτουργία της πρέπει να γίνει μια αναφορά στον τρόπο που γίνονται αυτοί οι μετασχηματισμοί.

Ας υποθέσουμε ότι έχουμε την πρόταση “I am very unhappy these days”. Ας υποθέσουμε επίσης ότι ένας άνθρωπος με πολύ λίγες γνώσεις αγγλικών ακούγοντας αυτή την πρόταση καταλάβαινε μόνο τις λέξεις “I am”. Προσπαθώντας να συνεχίσει τη συζήτηση θα μπορούσε να απαντήσει λέγοντας “How long have you been unhappy these days?”. Αυτό που έκανε ήταν να

εφαρμογή ένα κανόνα στην αρχική πρόταση, ένα μέρος του οποίου ταίριαζε με τις δύο λέξεις “I am” και ένα άλλο με το οποίο απομονωνε τις λέξεις “very unhappy these days”. Επίσης χρησιμοποίησε και ένα μετασχηματισμό κατά τον οποίο κάθε πρόταση της μορφής “I am foo” μπορεί να μετασχηματιστεί σε “How long have you been foo” ανεξάρτητα από το νόημα του foo. Ας δούμε ένα πιο περίπλοκο παραδειγμα χρησιμοποιώντας την πρόταση “It seems that you hate me”. Στην συγκεκριμένη περίπτωση ο άνθρωπος μας καταλαβαίνει μόνο τις λέξεις “you” και “me” και χρησιμοποιώντας ένα κανόνα αποσυνθέτει την πρόταση σε τέσσερα μέρη, όπως φαίνεται παρακάτω.

1	2	3	4
It seems that	you	hate	me

Ένας κανόνος ανασύνθεσης θα μπορούσε να είναι “What makes you think I hate you”, αγνοώντας το πρώτο μέρος, μετατρέποντας τις γνωστές λέξεις (το “you” σε “I” και το “me” σε “you”) και βάζοντας μπροστά την προκαθορισμένη φράση “What makes you think”. Μια τυποποίηση του παραπάνω προτύπου αποσύνθεσης είναι

*0 YOU 0 ME*

και ο αντίστοιχος κανόνας ανασύνθεσης ο ακόλουθος:

*WHAT MAKES YOU THINK I 3 YOU*

Το 0 στον κανόνα μετασχηματισμού σημαίνει ένας άπειρος αριθμός λέξεων, ενώ το 3 στον κανόνα ανασύνθεσης αναφέρεται στο τρίτο μέρος της αρχικής μας πρότασης. Στην παραπάνω πρόταση θα ταίριαζε και ο κανόνας

*0 YOU 1 ME*

Ένας αριθμός  $n$  διάφορος του μηδενός σε ένα κανόνα αποσύνθεσης σημαίνει ότι το μέρος αυτό αποτελείται από ακριβώς  $n$  λέξεις. Από τους δύο παραπάνω κανόνες αποσύνθεσης μόνο ο πρώτος θα μπορούσε να εφαρμοστεί στην πρόταση “It seems you love and hate me” γιατί υπάρχουν παραπάνω από μία λέξεις ανάμεσα στο “you” και το “me”.

Στην Eliza η επιλογή του καταλληλότερου κανόνα αποσύνθεσης είναι ζήτημα κομβικής σημασίας. Αυτή η επιλογή παρουσιάζει κάποια ιδιαίτερα προβλήματα, όπως φαίνεται από το παράδειγμα που ακολουθεί. Έστω η φράση “It seems that you hate”. Στην περίπτωση αυτή κανένας από τους παραπάνω κανόνες αποσύνθεσης δεν μπορεί να εφαρμοστεί γιατί προφανώς λείπει η λέξη “me”. Επομένως πρέπει να βρεθεί ένας άλλος κανόνας και αν ούτε αυτός εφαρμόζεται να συνεχιστεί αυτή η διαδικασία μέχρι να βρεθεί ένας κατάλληλος. Έτσι διαφαίνεται η ανάγκη για ένα μηχανισμό που να περιορίζει τους υποψήφιους κανόνες αποσύνθεσης και αυτοί είναι οι λέξεις-κλειδιά.

Η πρόταση “διαβάζεται” από τα δεξιά προς τα αριστερά και κάθε λέξη αναζητείται σε ένα λεξικό λέξεων-κλειδιά. Για κάθε λέξη-κλειδί που ανιχνεύεται όλοι οι κανόνες αποσύνθεσης που περιέχουν τη λέξη αυτή πρέπει να δοκιμάζονται και μάλιστα με συγκεκριμένη σειρά.

Πλέον ανακύπτουν δύο προβλήματα. Το πρώτο από το γεγονός ότι ελάχιστες είναι οι λέξεις σε μία πρόταση που είναι λέξεις-κλειδιά και το δεύτερο από το γεγονός ότι πρέπει με τη δεδομένη λέξη κλειδί να συσχετιστούν τόσο κάποιοι κανόνες αποσύνθεσης όσο και ανασύνθεσης. Τα δύο αυτά προβλήματα λύνονται βάζοντας τις λέξεις-κλειδιά και τους αντίστοιχους κανόνες σε μία λίστα. Μια τυπική τέτοια λίστα είναι η παρακάτω.

$$\begin{pmatrix} (K ((D_1) (R_{1,1}) (R_{1,2}) \cdots (R_{1,m_1})) \\ ((D_2) (R_{2,1}) (R_{2,2}) \cdots (R_{2,m_2})) \\ \cdot \\ \cdot \\ \cdot \\ ((D_n) (R_{n,1}) (R_{n,2}) \cdots (R_{n,m_n})) \end{pmatrix}$$

Όπου  $K$  είναι η λέξη-κλειδί,  $D_i$  ο  $i$  κανόνος αποσύνθεσης για την  $K$  και  $R_{ij}$  ο  $j$  κανόνος ανασύνθεσης που αντιστοιχεί στον  $i$  κανόνα αποσύνθεσης.

Αναλυτικότερα η διαδικασία που ακολουθείται είναι η εξής: Η Eliza διαβάζει το script file και περιμένει το input του χρήστη. Όταν διαβάζει το input ακολουθεί τα εξής βήματα :

1. Η πρόταση διασπάται στις λέξεις που την αποτελούν και η περαιτέρω ανάλυση γίνεται σε αυτές τις λέξεις και όχι σε μεμονομένους χαρακτήρες.
2. Γίνεται μια αρχική αντικατάσταση ( αποκόβονται καταλήξεις , άρθρα, συγκεκριμένες λέξεις)
3. Στη συνέχεια η Eliza παίρνει όλες τις εναπομείναντες λέξεις και δημιουργεί μια λίστα από τις λέξεις-κλειδιά που βρίσκει, τις οποίες στη συνέχεια τοποθετεί κατά φθίνουσα σειρά ανάλογα με το βάρος που έχει η κάθε λέξη.
4. Ξεκινά μια διαδικασία την οποία για την πρώτη λέξη στη λίστα προσπαθεί να βρεθεί ένας κανόνος αποσύνθεσης. Ο πρώτος κανόνος που μπορεί να εφαρμοστεί επιλέγεται αλλιώς αν δεν υπάρχει κανένας προχωράμε στην επόμενη λέξη-κλειδί.
5. Για τον κανόνα αποσύνθεσης που βρέθηκε επιλέγεται ένας κανόνος ανασύνθεσης. Μπορεί να υπάρχουν πολλοί κανόνες ανασύνθεσης που να αντιστοιχούν στον επιλεγμένο κανόνα αποσύνθεσης οπότε αυτοί θα χρησιμοποιούνται κυκλικά κάθε φορά που θα βρίσκεται ο ίδιος κανόνος.
6. Τέλος γίνεται μια ακόμα διαδικασία αντικατάστασης όπως στο βήμα 2 αλλά αυτή τη φορά σε συνθετικό ύφος.
7. Το αποτέλεσμα της παραπάνω διαδικασίας είναι η απάντηση στο χρήστη.

Το script πέρα από τις λέξεις κλειδιά και του κανόνες περιέχει και τους μετασχηματισμούς των βημάτων 2 και 6, όπως επίσης και μια διαδικασία εύρεσης συνωνύμων.

Παρακάτω ακολουθεί ένας διάλογος της Eliza και μιας νέας κοπέλας.

**young woman:** Men are all alike.

**eliza:** In what way?

**young woman:** They're always bugging us about something specific or other.

**eliza:** Can you think of a specific example?

**young woman:** Well, my boyfriend made me come here.

**eliza:** Your boyfriend made you come here?

**young woman:** He says I'm depressed much of the time.

**eliza:** I am sorry to hear you are depressed.

**young woman:** It's true. I'm unhappy.

**eliza:** Do you think coming here will help you not to be unhappy?

**young woman:** I need some help, that much seems certain.

**eliza:** What would it mean to you if you got some help?

**young woman:** Perhaps I could learn to get along with my mother.

**eliza:** Tell me more about your family.

**young woman:** My mother takes care of me.

**eliza:** Who else in your family takes care of you?

**young woman:** My father.

**eliza:** Your father.

**young woman:** You are like my father in some ways.

**eliza:** What resemblance do you see?

**young woman:** You are not very aggressive but I think you don't want me to notice that.

**eliza:** What makes you think I am not very aggressive?

**young woman:** You don't argue with me.

**eliza:** Why do you think I don't argue with you?

**young woman:** You are afraid of me.

**eliza:** Does it please you to believe I am afraid of you?

**young woman:** My father is afraid of everybody.

**eliza:** What else comes to mind when you think of your father?

**young woman:** Bullies.

**eliza:** Does that have anything to do with the fact that your boyfriend made you come here?

## 8.5 Τεχνολογία και Συστήματα Μετατροπής Κειμένου σε Ομιλία

Για να μπορεί ο ευφυής πράκτορας να μετατρέπει το κείμενο που θέλει να «πει» 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.

### 8.5.1 Εισαγωγή

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. 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.

### 8.5.2 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.

### 8.5.3 FreeTTS

Το FreeTTS είναι ανοικτού κώδικα σύστημα ερευνητικό σύστημα MKCΟ 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 MKCΟ 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 αλγόριθμους 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.

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.

#### 8.5.4 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 patter 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 αντικειμένου 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
<b>Pitch</b>	Float	Ορίζουμε τη συχνότητα της φωνής του agent
<b>Pitchshift</b>	Float	Ορίζουμε τις μεταβολές στη συχνότητα της φωνής (προσωδία)
<b>Name</b>	String	Ορίζουμε ένα όνομα από τις default φωνές που παρέχονται με το FreeTTS (π.χ. "Kevin", "kevin16" κλπ)
<b>Style</b>	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.

## 9. Case Study

### 9.1 Εισαγωγή

Στο Case study αυτό θα αξιολογηθεί η συμπεριφορά του ευφυούς πράκτορα που μοντελοποιήσαμε μέσα σε ένα ολοκληρωμένο εικονικό περιβάλλον. Το εικονικό αυτό περιβάλλον αναπαριστά ένα bar. Ακολουθεί μια συντομη περιγραφή του εικονικού κόσμου και στη συνέχεια το σενάριο και η δράση των πρακτόρων μας.

### 9.2 Εικονικό Bar

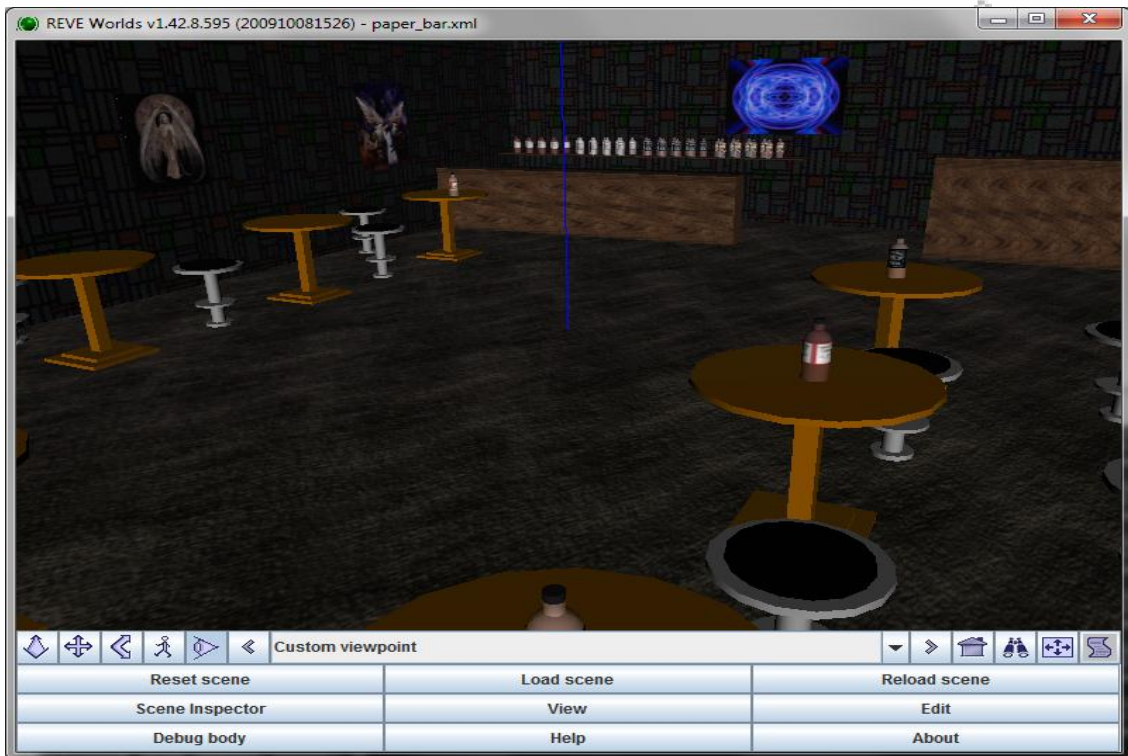
#### 9.2.1 Εισαγωγή

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.

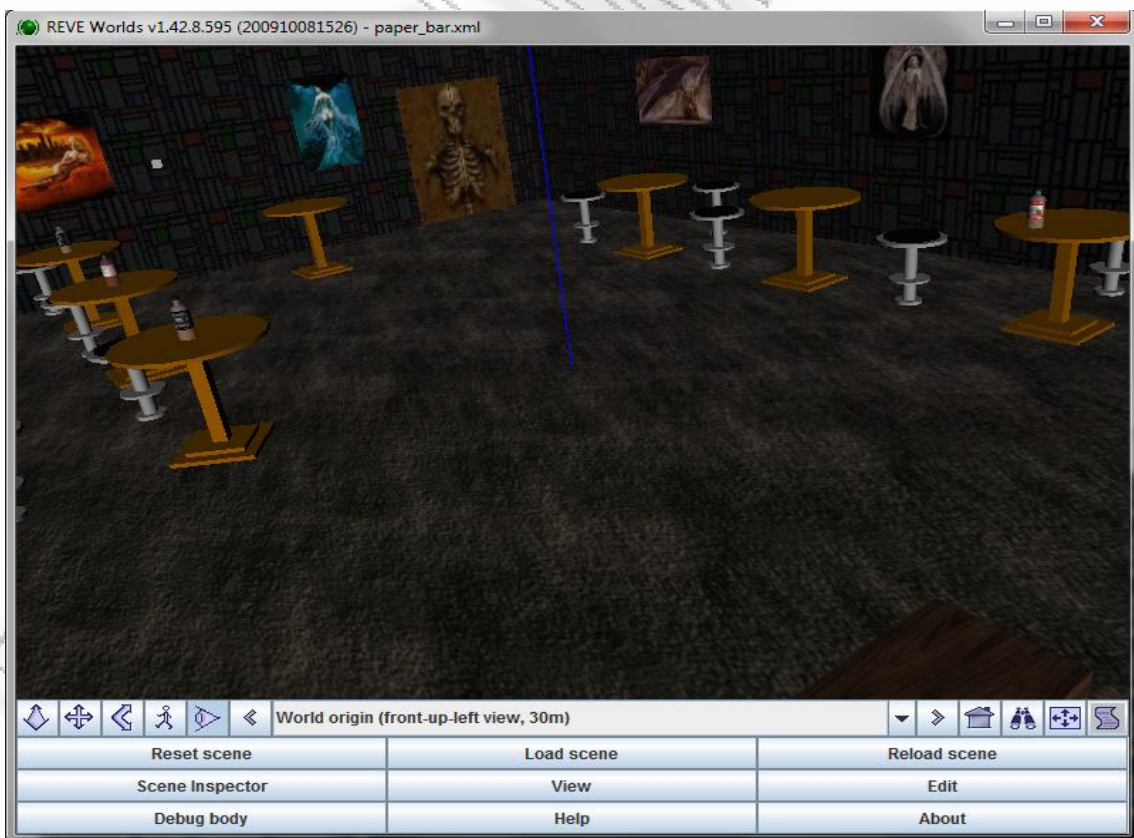
Η VERL μας δίνει τη δυνατότητα να επιλεξουμε τον βαθμό λεπτομέρειας και του επιπέδου αφαίρεσης της μοντελοποίησης μας, δηλαδή την αντιστοίχιση των συστατικών του γραφήματος σκηνής σε items. Στην περίπτωση μας επιλέξαμε να αντιστοιχήσουμε κάθε συστατικό του γραφήματος σκηνής σε ένα item, για να έχουμε απλούστερη γεωμετρική δομή, μέγιστη σημασιολογική και αλληλεπιδραστική αυτόνομία, 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.

#### 9.2.2 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 αλληλεπιδρασης, όσο και την ευκρίνεια του κώδικα μας και τη δυνατότητα συντήρησης και επαναχρησιμοποίησης.



Εικόνα 9.1 Αναπαράσταση του εικονικού μπαρ



Εικόνα 9.2 Αναπαράσταση του εικονικού μπαρ από άλλη γωνία

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 εφυής 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 δενδροειδή 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	Location
bar_building	floor	0,0,0	Tables	Table_1a	0,0,1,-4,5
	wall_01	4,1,5,0		Table_1b	-2,3,0,1,-0,5
	wall_02	-4,1,5,0		Table_1c	-2,3,0,9,-0,5
	wall_03	0,1,5,6		Table_1d	-2,15,0,9,-0,5
	great_wall	-1,1,5,-6		Table_1e	-2,55,0,9,-0,37
	small_wall_01	3,75,1,5,-6		Table_2a	-2,3,0,1,-2,5
	small_wall_02	2,75,2,5,-6		Table_2b	-2,3,0,9,-2,5
	corridor_Wall	1,95,1,5,-6,5		Table_2c	-2,15,0,9,-2,5
	Corridor_Wall_02	3,55,1,5,-9		Table_2d	-2,35,0,9,-2,33
	Corridor_Wall_03	1,95,1,5,-8,5		Table_2e	-2,55,0,9,-2,37
	Bathroom_Wall_01	0,9,1,5,-6,95		Table_3a	-2,3,0,1,-4,5
	Bathroom_Wall_02	0,9,1,5,-8,05		Table_3b	-2,3,0,9,-4,5
	Bathroom_Wall_03	0,9,1,5,-8,95		Table_3c	-2,15,0,9,-4,5
	Bathroom_Wall_04	0,9,1,5,-10,05		Table_3d	-2,35,0,9,-4,33
	Bathroom_Wall_05	2,75,1,5,-12		Table_3e	-2,55,0,9,-4,37
	Bathroom_Wall_06	0,1,5,-9		Table_4a	2,5,0,1,-2
	Bathroom_Small_Wall	1,95,1,5,-11		Table_5a	2,5,0,1,2,13
	Bathroom_Floor	2,75,0,-9		Table_5b	2,3,0,9,2
	Bathroom_Door_01	3,48,1,-6		Table_5c	2,35,0,9,1,8
	Bathroom_Door_02	2,02,1,-6		Table_5d	2,65,0,9,2,1
	Bathroom_Door_03	1,9,1,-7		Table_5e	2,55,0,9,2,2
	Bathroom_Door_04	1,9,1,-9		Table_6a	2,5,0,1,0
DJ	Dj_table	-3,0,65,4,2	Chairs	Chair_01	3,5,0,1,0
	DJ_floor	-3,0,15,5,1		Chair_02	3,5,0,1,2
	Dj_console	-3,0,45,5,1		Chair_03	3,5,0,1,-2
Bar_Table	Bar_Table	2,0,5,4,2		Chair_04	2,5,0,1,-1
	Bar_Drinks	0,1,1,5,85		Chair_05	2,5,0,1,1
Paintings	Painting_01	3,9,1,7,0		Chair_06	2,5,0,1,-3
	Battleaxe	3,9,2,2,1,5		Chair_07	-3,5,0,1,-2,5
	Guitar	3,9,2,2,-1,5		Chair_08	-3,5,0,1,-0,5
	Painting_02	3,9,1,7,3		Chair_09	-3,5,0,1,-4,5
	Painting_03	3,9,1,7,-3		Chair_10	-2,5,0,1,-3,5
	Painting_04	-3,9,1,7,-2		Chair_11	-2,5,0,1,-1,5
	Painting_05	-3,9,1,7,3		Chair_12	-2,5,0,1,-5,5
	Painting_06	0,2,5,9	Chair_Bar_01	3,3,0,1,3,5	
Painting_07	-2,1,7,-5,9	Chair_Bar_02	2,6,0,1,3,5		
Painting_08	1,1,7,-5,9	Chair_Bar_03	1,9,0,1,3,5		
Bathroom_utilities	Basin	2,75,0,8,-11,5	Chair_Bar_04	1,2,0,1,3,5	
	Toilet_1	0,75,0,1,-9,5	Chair_Bar_05	0,5,0,1,3,5	
	Toilet_2	0,75,0,1,-7,5			

Εικόνα 9.3 Πίνακας των items

### 9.2.3 Παράδειγμα ολοκληρωμένης μοντελοποίησης item

Για να γίνει κατανοητή η παραπάνω διαδικασία θα γίνει μια ολοκληρωμένη μοντελοποίηση του item Painting\_01. Το VRML μοντέλο ορίζεται όπως παρακάτω.

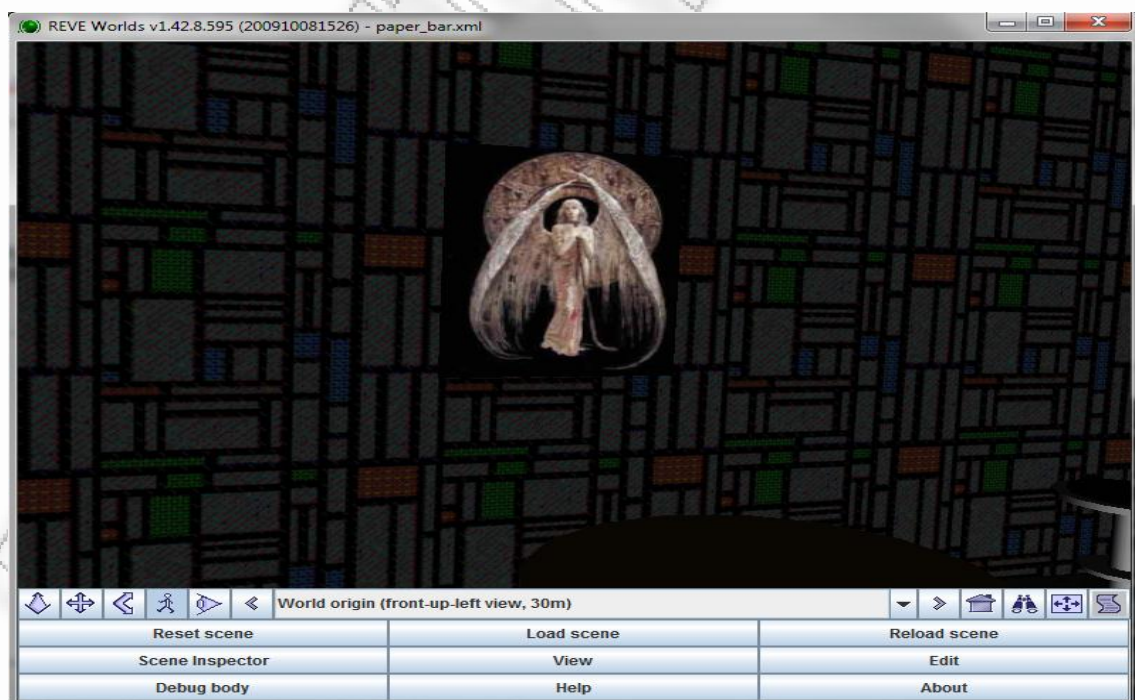
```

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}
                      ]
                    }

```

Εικόνα 9.4 Το item Painting\_01 σε VRML

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 αποθήκευση της με σκοπό την αλληλεπίδραση του εικονικού μας πράκτορα με αυτό.



Εικόνα 9.5 Το item Painting\_01 στο ReveWorlds

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 VRLM 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.

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

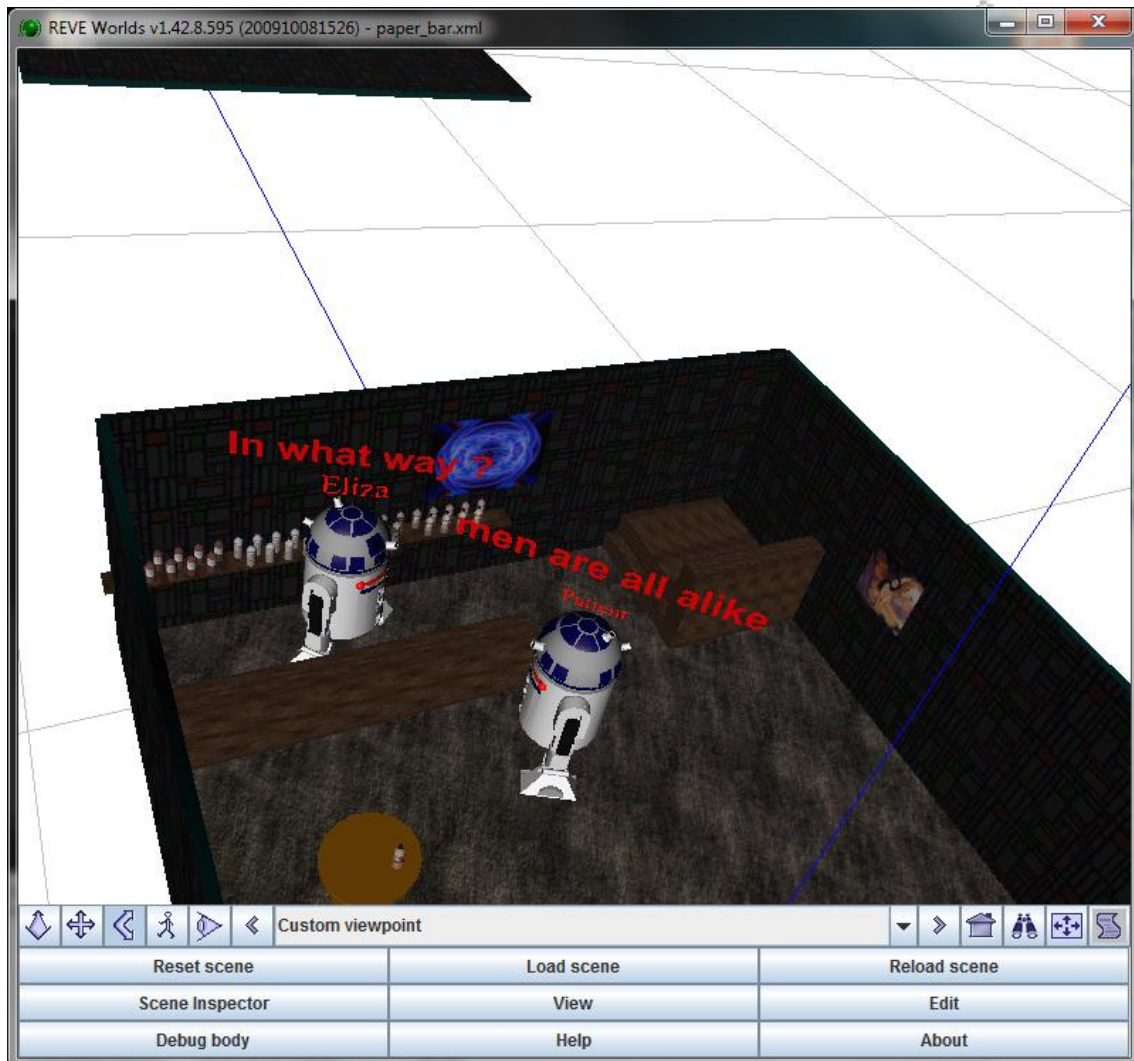
### 9.3 Σενάριο και Παρουσίαση

Είδαμε παραπάνω τον εικονικό κόσμο μέσα στον οποίο θα κινηθούν οι πράκτορες μας. Στο συγκεκριμένο *case study* θα παρουσιάσουμε τον ευφυή εικονικό πράκτορα *Eliza* σε μια συνομιλία με έναν άλλο πράκτορα, ο οποίος προσομοιώνοντας τον χρήστη έχει μια αυστηρή *scripted* συμπεριφορά. Πιο συγκεκριμένα ο δεύτερος *agent* μια νεαρή κοπέλα η οποία νιώθοντας την ανάγκη να μιλήσει με την ψυχολόγο της *Eliza* βρίσκονται σε ένα *bar* και συζητούν.

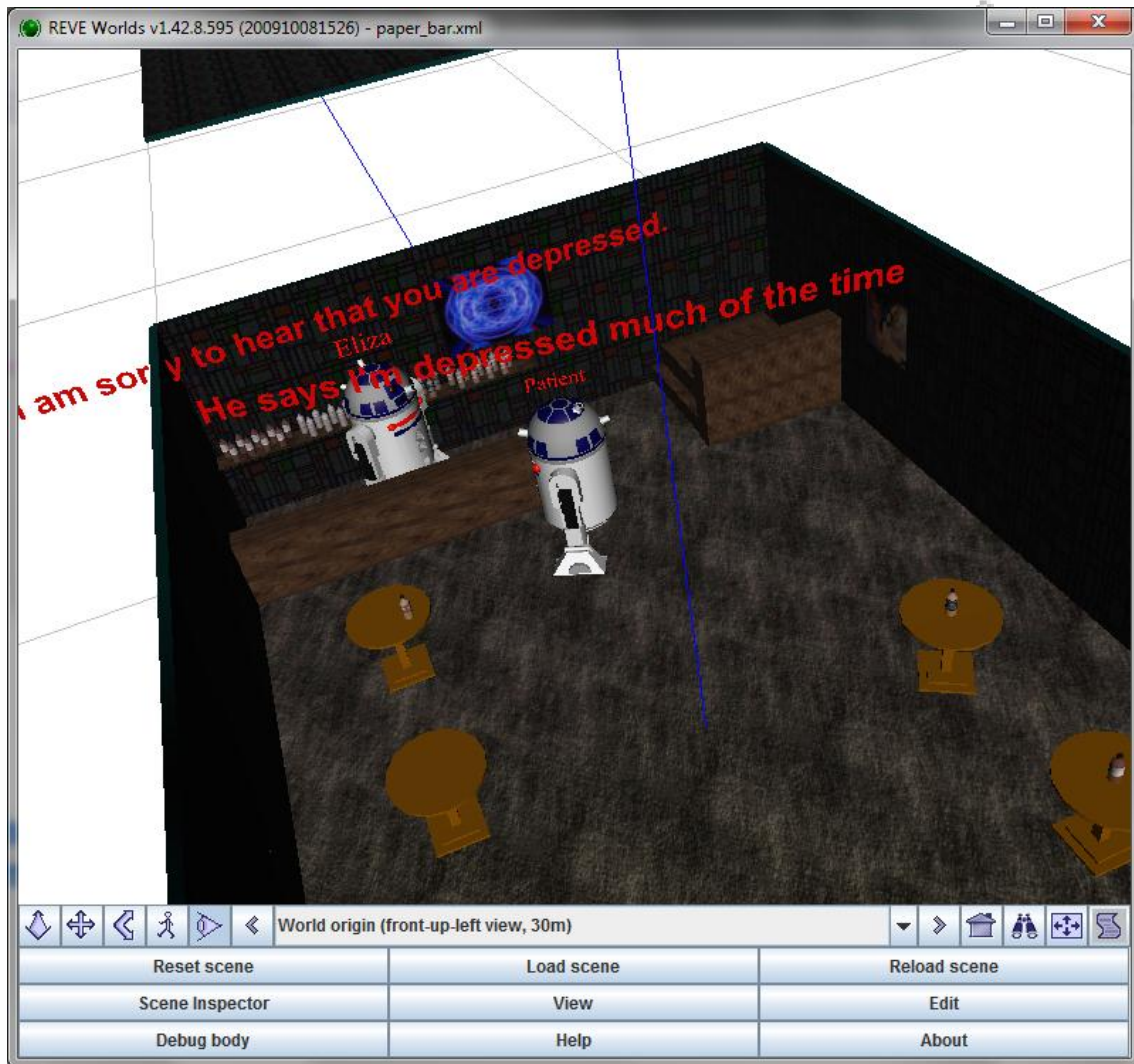
Ο πράκτορας *Eliza* ακολουθεί κανονικά το μοντέλο συμπεριφοράς που του ορίσαμε παραπάνω απαντώντας στην *Patient* σύμφωνα με το *script* του κώδικα της. Ο πράκτορας *Patient* από την άλλη έχει ένα *script* στην ίδια αρχιτεκτονική με αυτό της *Eliza* αλλά με μόνη δυνατότητα να ακολουθεί τη συγκεκριμένη συζήτηση που παρουσιάζουμε σε αυτό το *case study*. Στην ουσία είναι άλλος ένας πράκτορας *Eliza* αλλά με πολύ περιορισμένη δυνατότητα διαλόγου.

Παρακάτω ακολουθούν κάποια χαρακτηριστικά αποσπάσματα διαλόγου της *Eliza* και του *Patient*.

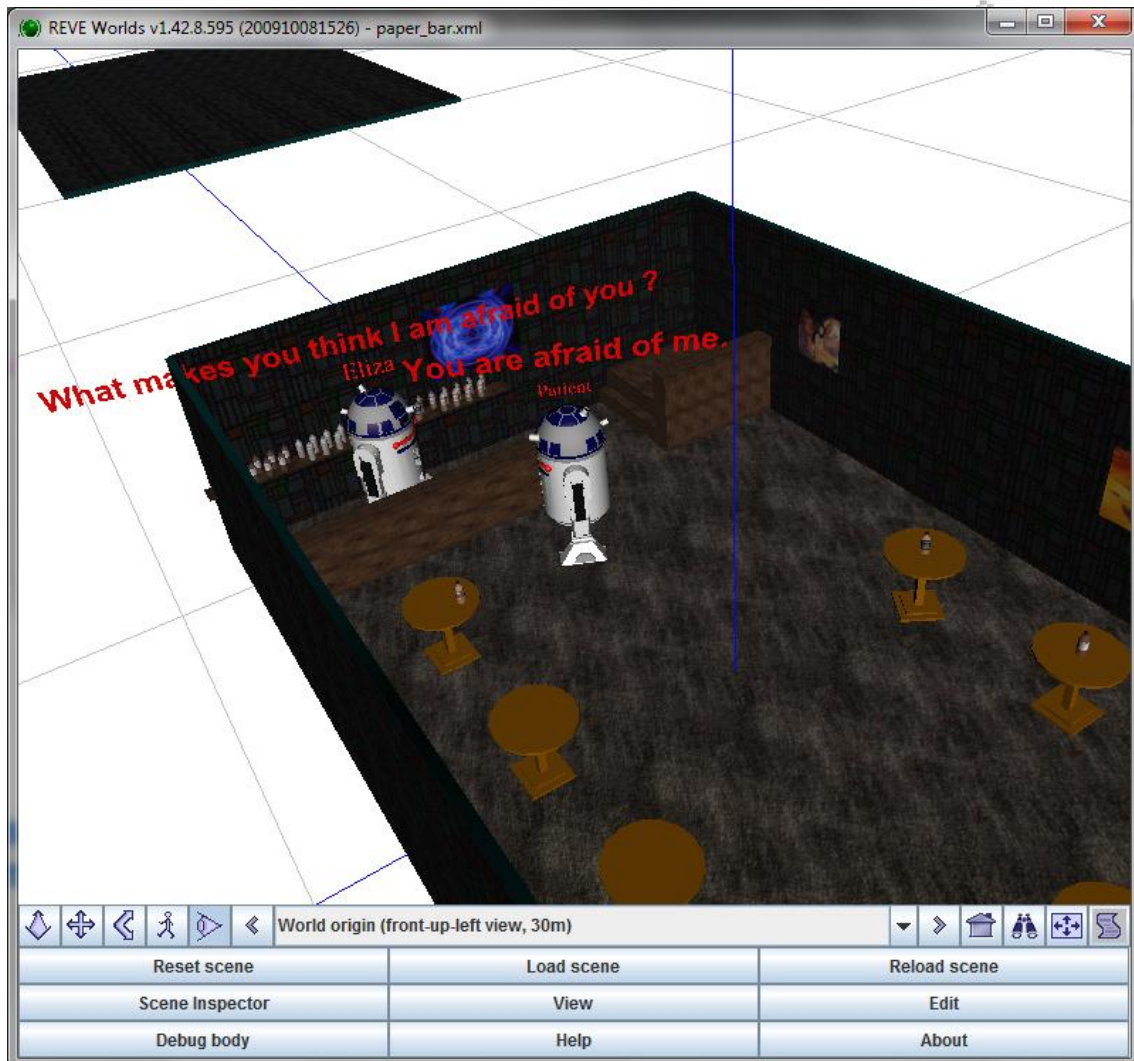




Εικόνα 9.6 Πρώτο στιγμιότυπο από το σενάριο



Εικόνα 9.7 Δεύτερο στιγμιότυπο από το σενάριο



Εικόνα 9.8 Τρίτο Στιγμιότυπο από το σενάριο

## 10. Συμπεράσματα

Στην εργασία αυτή μοντελοποιήσαμε έναν εικονικό πράκτορα δίνοντας του τη δυνατότητα αντίληψης και εκφοράς λόγου τόσο οπτικά όσο και ακουστικά. Για να αξιολογήσουμε την υλοποίησή μας εισάγαμε τον πράκτορα Eliza και έναν άλλο πράκτορα σε έναν εικονικό κόσμο που αναπαριστούσε ένα bar και έγινε μια ευφυής συζήτηση μεταξύ. Η Eliza ήταν σε θέση να “δει” την ομιλία του άλλου πράκτορα, να αντιληφθεί ότι αυτή είναι ομιλία και να την επεξεργαστεί και αποθηκεύσει ως τέτοια. Στη συνέχεια συνέχισε την συνομιλία με έναν αρκετά ικανοποιητικό τρόπο.

Ένα σημαντικό συμπέρασμα από την παραπάνω υλοποίηση είναι η ευκολία μοντελοποίησης μιας τόσο σύνθετης λειτουργίας όπως είναι η ομιλία με βάση την αναπαράσταση REVE και την πλατφόρμα υλοποίησης jREVE. Οι δυνατότητες για επέκταση των εικονικών πρακτόρων και των εικονικών κόσμων γενικότερα είναι τεράστιες.

## 10.1 Μελλοντικές Προσεγγίσεις

Το ερευνητικό ενδιαφέρον για την αναπαράσταση REVE είναι μεγάλο και το εύρος των ερωτημάτων που μπορεί να απαντήσει είναι πολλά, αλλά στην περίπτωση μας δύο είναι οι κύριοι στόχοι που θα μπορούσε να θέσει κανείς.

Ο πρώτος είναι η επέκταση της μοντελοποίησης του πράκτορα ώστε να περιλαμβάνει και την ακοή. Με τον τρόπο αυτό θα έχουμε έναν πράκτορα ο οποίος έχει το σύνολο σχεδόν των ανθρώπινων αισθήσεων (εκτός από αυτήν της γεύσης), φτάνοντας έτσι ένα βήμα πιο κοντά στην φυσική αναπαράσταση ενός ανθρώπου.

Μια δεύτερη προσέγγιση που χρίζει προσοχής είναι αυτή της επέκτασης των δυνατοτήτων του μοντέλου παραγωγής συμπεριφοράς συνδυάζοντας την ομιλία με την επέκταση της βάσης γνώσης του πράκτορα. Πιο συγκεκριμένα η αντίληψη που έχει ο πράκτορας για τον κόσμο να μην περιορίζεται στις αισθητήριακές αντιλήψεις αλλά να περιέχει και αντιλήψεις για τον κόσμο (beliefs) και στόχους, όχι α priori αλλά μέσα από συζήτηση με άλλους εικονικούς πράκτορες ή με τους χρήστες.

## 11. Βιβλιογραφία

Anastassakis, G. Αναπαράσταση και Λειτουργία Εικονικών Περιβαλλόντων, PhD Thesis 2009

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.
- Sun Microsystems, "Java Platform, Standard Edition 6, API Specification", <http://java.sun.com/javase/6/docs/api/>, 2009.
- Web3D Consortium, "X3D and Related Specifications", <http://www.web3d.org/x3d/specifications/>, 2009.
- Black A.W., Lenzo, K., (2007), "Building Synthetic Voices"
- Stuart Russel - Peter Norvig*, *Artificial Intelligence - A Modern Approach*, Prentice Hall, 1995
- Michael A. Covington*, *Natural Language Processing for Prolog Programmers*, Prentice Hall, 1994
- Daniel Jurafsky & James H. Martin*, *Speech and Language Processing*, Prentice Hall, 2000
- Weizenbaum J. ELIZA — A Computer Program for the Study of Natural Language Communication between Man and Machine, *Communications of the Association for Computing Machinery* 9 (1966)
- Weizenbaum J. Symmetric list processor. *Comm. ACM* 6, (Sept. 1963), 524-544.
- Yngve, J. *COMIT Programming Manual*. MIT Press, Cambridge, Mass., 1961.
- Black A.W., Lenzo, K., (2007), "Building Synthetic Voices"
- Boersma P., Weenink D., (2005), "Praat: doing phonetics by computer"
- 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.

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

JavaTMSpeech API Programmer's Guide, [http://java.sun.com/products/java-](http://java.sun.com/products/java-media/speech/forDevelopers/jsapi-guide)

## 12. Παράρτημα

### 12.1 To script της Eliza

```
initial: How do you do. Please tell me your problem.
final: Goodbye. It was nice talking to you.
quit: bye
quit: goodbye
quit: done
quit: exit
quit: quit
pre: dont don't
pre: cant can't
pre: wont won't
pre: recollect remember
pre: recall remember
pre: dreamt dreamed
pre: dreams dream
pre: maybe perhaps
pre: certainly yes
pre: machine computer
pre: machines computer
pre: computers computer
post: am are
post: your my
pre: were was
post: me you
pre: you're you are
pre: i'm i am
```

```
post: myself yourself
post: yourself myself
post: i you
post: you I
post: my your
post: i'm you are
pre: same alike
pre: identical alike
pre: equivalent alike
synon: belief feel think believe wish
synon: family mother mom father dad sister brother wife children child
synon: desire want need
synon: sad unhappy depressed sick
synon: happy elated glad better
synon: cannot can't
synon: everyone everybody nobody noone
synon: be am is are was
key: xnone 0

  decomp: *

    reasmb: I'm not sure I understand you fully.
    reasmb: Please go on.
    reasmb: What does that suggest to you ?
    reasmb: Do you feel strongly about discussing such things ?

      reasmb: That is interesting. Please continue.
      reasmb: Tell me more about that.
      reasmb: Does talking about this bother you ?

key: sorry 0

  decomp: *

    reasmb: Please don't apologise.
    reasmb: Apologies are not necessary.
    reasmb: I've told you that apologies are not required.

      reasmb: It did not bother me. Please continue.

key: apologise 0
```

```
decomp: *  
    reasmb: goto sorry  
key: remember 5  
decomp: * i remember *  
    reasmb: Do you often think of (2) ?  
    reasmb: Does thinking of (2) bring anything else to mind ?  
    reasmb: What else do you recollect ?  
    reasmb: Why do you remember (2) just now ?  
    reasmb: What in the present situation reminds you of (2) ?  
    reasmb: What is the connection between me and (2) ?  
        reasmb: What else does (2) remind you of ?  
decomp: * do you remember *  
    reasmb: Did you think I would forget (2) ?  
    reasmb: Why do you think I should recall (2) now ?  
    reasmb: What about (2) ?  
    reasmb: goto what  
    reasmb: You mentioned (2) ?  
decomp: * you remember *  
    reasmb: How could I forget (2) ?  
    reasmb: What about (2) should I remember ?  
    reasmb: goto you  
key: forget 5  
decomp: * i forget *  
    reasmb: Can you think of why you might forget (2) ?  
    reasmb: Why can't you remember (2) ?  
    reasmb: How often do you think of (2) ?  
    reasmb: Does it bother you to forget that ?  
    reasmb: Could it be a mental block ?  
    reasmb: Are you generally forgetful ?  
    reasmb: Do you think you are suppressing (2) ?  
decomp: * did you forget *  
    reasmb: Why do you ask ?  
    reasmb: Are you sure you told me ?
```



```
reasm: Would it bother you if I forgot (2) ?

reasm: Why should I recall (2) just now ?

reasm: goto what

reasm: Tell me more about (2).

key: if 3

decomp: * if *

reasm: Do you think its likely that (2) ?

reasm: Do you wish that (2) ?

reasm: What do you know about (2) ?

reasm: Really, if (2) ?

reasm: What would you do if (2) ?

reasm: But what are the chances that (2) ?

reasm: What does this speculation lead to ?

key: dreamed 4

decomp: * i dreamed *

reasm: Really, (2) ?

reasm: Have you ever fantasized (2) while you were awake ?

reasm: Have you ever dreamed (2) before ?

reasm: goto dream

key: dream 3

decomp: *

reasm: What does that dream suggest to you ?

reasm: Do you dream often ?

reasm: What persons appear in your dreams ?

reasm: Do you believe that dreams have something to do with your problem ?

key: perhaps 0

decomp: *

reasm: You don't seem quite certain.

reasm: Why the uncertain tone ?

reasm: Can't you be more positive ?

reasm: You aren't sure ?

reasm: Don't you know ?

reasm: How likely, would you estimate ?
```

```
key: name 15

decomp: *

reasm: I am not interested in names.

reasm: I've told you before, I don't care about names -- please continue.

key: deutsch 0

decomp: *

reasm: goto xforeign

reasm: I told you before, I don't understand German.

key: francais 0

decomp: *

reasm: goto xforeign

reasm: I told you before, I don't understand French.

key: italiano 0

decomp: *

reasm: goto xforeign

reasm: I told you before, I don't understand Italian.

key: espanol 0

decomp: *

reasm: goto xforeign

reasm: I told you before, I don't understand Spanish.

key: xforeign 0

decomp: *

reasm: I speak only English.

key: hello 0

decomp: *

reasm: How do you do. Please state your problem.

reasm: Hi. What seems to be your problem ?

key: computer 50

decomp: *

reasm: Do computers worry you ?

reasm: Why do you mention computers ?

reasm: What do you think machines have to do with your problem ?

reasm: Don't you think computers can help people ?
```

```
reasm: What about machines worries you ?
reasm: What do you think about machines ?
    reasm: You don't think I am a computer program, do you ?
key: am 0
decomp: * am i *
    reasm: Do you believe you are (2) ?
reasm: Would you want to be (2) ?
reasm: Do you wish I would tell you you are (2) ?
reasm: What would it mean if you were (2) ?
reasm: goto what
decomp: * i am *
    reasm: goto i
decomp: *
    reasm: Why do you say 'am' ?
reasm: I don't understand that.
key: are 0
decomp: * are you *
    reasm: Why are you interested in whether I am (2) or not ?
reasm: Would you prefer if I weren't (2) ?
reasm: Perhaps I am (2) in your fantasies.
reasm: Do you sometimes think I am (2) ?
reasm: goto what
    reasm: Would it matter to you ?
    reasm: What if I were (2) ?
decomp: * you are *
    reasm: goto you
decomp: * are *
    reasm: Did you think they might not be (2) ?
reasm: Would you like it if they were not (2) ?
reasm: What if they were not (2) ?
    reasm: Are they always (2) ?
reasm: Possibly they are (2).
    reasm: Are you positive they are (2) ?
```

```
key: your 0

decomp: * your *

  reasmb: Why are you concerned over my (2) ?

  reasmb: What about your own (2) ?

  reasmb: Are you worried about someone else's (2) ?

  reasmb: Really, my (2) ?

    reasmb: What makes you think of my (2) ?

    reasmb: Do you want my (2) ?

key: was 2

decomp: * was i *

  reasmb: What if you were (2) ?

  reasmb: Do you think you were (2) ?

  reasmb: Were you (2) ?

  reasmb: What would it mean if you were (2) ?

  reasmb: What does ' (2) ' suggest to you ?

  reasmb: goto what

decomp: * i was *

  reasmb: Were you really ?

  reasmb: Why do you tell me you were (2) now ?

  reasmb: Perhaps I already know you were (2).

decomp: * was you *

  reasmb: Would you like to believe I was (2) ?

  reasmb: What suggests that I was (2) ?

  reasmb: What do you think ?

  reasmb: Perhaps I was (2).

  reasmb: What if I had been (2) ?

key: i 0

decomp: * i @desire *

  reasmb: What would it mean to you if you got (3) ?

  reasmb: Why do you want (3) ?

  reasmb: Suppose you got (3) soon.

  reasmb: What if you never got (3) ?

  reasmb: What would getting (3) mean to you ?
```

```
reasm: What does wanting (3) have to do with this discussion ?
decomp: * i am* @sad *
reasm: I am sorry to hear that you are (3).
reasm: Do you think coming here will help you not to be (3) ?
reasm: I'm sure it's not pleasant to be (3).
reasm: Can you explain what made you (3) ?
decomp: * i am* @happy *
reasm: How have I helped you to be (3) ?
reasm: Has your treatment made you (3) ?
reasm: What makes you (3) just now ?
reasm: Can you explain why you are suddenly (3) ?
decomp: * i was *
reasm: goto was
decomp: * i @belief i *
reasm: Do you really think so ?
reasm: But you are not sure you (3).
reasm: Do you really doubt you (3) ?
decomp: * i* @belief *you *
reasm: goto you
decomp: * i am *
reasm: Is it because you are (2) that you came to me ?
reasm: How long have you been (2) ?
reasm: Do you believe it is normal to be (2) ?
reasm: Do you enjoy being (2) ?
reasm: Do you know anyone else who is (2) ?
decomp: * i @cannot *
reasm: How do you know that you can't (3) ?
reasm: Have you tried ?
reasm: Perhaps you could (3) now.
reasm: Do you really want to be able to (3) ?
reasm: What if you could (3) ?
decomp: * i don't *
reasm: Don't you really (2) ?
```

```
reasm: Why don't you (2) ?  
reasm: Do you wish to be able to (2) ?  
reasm: Does that trouble you ?  
decomp: * i feel *  
reasm: Tell me more about such feelings.  
reasm: Do you often feel (2) ?  
reasm: Do you enjoy feeling (2) ?  
reasm: Of what does feeling (2) remind you ?  
decomp: * i * you *  
reasm: Perhaps in your fantasies we (2) each other.  
reasm: Do you wish to (2) me ?  
reasm: You seem to need to (2) me.  
reasm: Do you (2) anyone else ?  
decomp: *  
reasm: You say (1) ?  
reasm: Can you elaborate on that ?  
reasm: Do you say (1) for some special reason ?  
reasm: That's quite interesting.  
key: you 0  
decomp: * you remind me of *  
reasm: goto alike  
decomp: * you are *  
reasm: What makes you think I am (2) ?  
reasm: Does it please you to believe I am (2) ?  
reasm: Do you sometimes wish you were (2) ?  
reasm: Perhaps you would like to be (2).  
decomp: * you* me *  
reasm: Why do you think I (2) you ?  
reasm: You like to think I (2) you -- don't you ?  
reasm: What makes you think I (2) you ?  
reasm: Really, I (2) you ?  
reasm: Do you wish to believe I (2) you ?  
reasm: Suppose I did (2) you -- what would that mean ?
```

```
reasm: Does someone else believe I (2) you ?
decomp: * you *
reasm: We were discussing you -- not me.
reasm: Oh, I (2) ?
reasm: You're not really talking about me -- are you ?
reasm: What are your feelings now ?
key: yes 0
decomp: *
reasm: You seem to be quite positive.
reasm: You are sure.
reasm: I see.
reasm: I understand.
key: no 0
decomp: * no one *
reasm: Are you sure, no one (2) ?
reasm: Surely someone (2) .
reasm: Can you think of anyone at all ?
reasm: Are you thinking of a very special person ?
reasm: Who, may I ask ?
reasm: You have a particular person in mind, don't you ?
reasm: Who do you think you are talking about ?
decomp: *
reasm: Are you saying no just to be negative?
reasm: You are being a bit negative.
reasm: Why not ?
reasm: Why 'no' ?
key: my 2
decomp: $ * my *
reasm: Lets discuss further why your (2).
reasm: Earlier you said your (2).
reasm: But your (2).
reasm: Does that have anything to do with the fact that your (2) ?
decomp: * my* @family *
```

```
reasm: Tell me more about your family.
reasm: Who else in your family (4) ?
reasm: Your (3) ?
reasm: What else comes to mind when you think of your (3) ?
decomp: * my *
reasm: Your (2) ?
reasm: Why do you say your (2) ?
reasm: Does that suggest anything else which belongs to you ?
reasm: Is it important to you that your (2) ?
key: can 0
decomp: * can you *
reasm: You believe I can (2) don't you ?
reasm: goto what
reasm: You want me to be able to (2).
reasm: Perhaps you would like to be able to (2) yourself.
decomp: * can i *
reasm: Whether or not you can (2) depends on you more than on me.
reasm: Do you want to be able to (2) ?
reasm: Perhaps you don't want to (2).
reasm: goto what
key: what 0
decomp: *
reasm: Why do you ask ?
reasm: Does that question interest you ?
reasm: What is it you really want to know ?
reasm: Are such questions much on your mind ?
reasm: What answer would please you most ?
reasm: What do you think ?
reasm: What comes to mind when you ask that ?
reasm: Have you asked such questions before ?
reasm: Have you asked anyone else ?
key: who 0
decomp: who *
```



```
    reasmb: goto what
key: when 0
    decomp: when *
        reasmb: goto what
key: where 0
    decomp: where *
        reasmb: goto what
key: how 0
    decomp: how *
        reasmb: goto what
key: because 0
    decomp: *
        reasmb: Is that the real reason ?
        reasmb: Don't any other reasons come to mind ?
        reasmb: Does that reason seem to explain anything else ?
        reasmb: What other reasons might there be ?
key: why 0
    decomp: * why don't you *
        reasmb: Do you believe I don't (2) ?
        reasmb: Perhaps I will (2) in good time.
        reasmb: Should you (2) yourself ?
        reasmb: You want me to (2) ?
        reasmb: goto what
    decomp: * why can't i *
        reasmb: Do you think you should be able to (2) ?
        reasmb: Do you want to be able to (2) ?
        reasmb: Do you believe this will help you to (2) ?
        reasmb: Have you any idea why you can't (2) ?
        reasmb: goto what
    decomp: *
        reasmb: goto what
key: everyone 2
    decomp: * @everyone *
```

```
reasm: Really, (2) ?
reasm: Surely not (2).
reasm: Can you think of anyone in particular ?
reasm: Who, for example?
reasm: Are you thinking of a very special person ?
reasm: Who, may I ask ?
reasm: Someone special perhaps ?
reasm: You have a particular person in mind, don't you ?
reasm: Who do you think you're talking about ?
key: everybody 2
decomp: *
reasm: goto everyone
key: nobody 2
decomp: *
reasm: goto everyone
key: noone 2
decomp: *
reasm: goto everyone
key: always 1
decomp: *
reasm: Can you think of a specific example ?
reasm: When ?
reasm: What incident are you thinking of ?
reasm: Really, always ?
key: alike 10
decomp: *
reasm: In what way ?
reasm: What resemblance do you see ?
reasm: What does that similarity suggest to you ?
reasm: What other connections do you see ?
reasm: What do you suppose that resemblance means ?
reasm: What is the connection, do you suppose ?
reasm: Could there really be some connection ?
```

```

reasm: How ?

key: like 10

  decomp: * @be *like *

    reasm: goto alike

key: different 0

  decomp: *

    reasm: How is it different ?

    reasm: What differences do you see ?

    reasm: What does that difference suggest to you ?

    reasm: What other distinctions do you see ?

    reasm: What do you suppose that disparity means ?

    reasm: Could there be some connection, do you suppose ?

    reasm: How ?

```

## 12.2 Ο πλήρης ορισμός του εικονικού bar στη γλώσσα VERL

```

<!DOCTYPE abr SYSTEM "file:///F:/semester_b/Intelligence Virtual
Enviroment/reve_material/REVEWorlds/dist/worlds/ABRML-1.0-20080514.dtd">

<abr version="1.0">

  <world version="1.0" name="bar-01" dimensions="">

    <info

      author="Giakoumis Stathis, Georgitsis Dimitris"

      date="21/5/2008"

      revision="1"

      email="p3000011@hotmail.com"

      organization="Knowledge Engineering Lab, Department of Informatics, University of
Piraeus"

      description="bar v01 scene"

      url="http://kelnet.cs.unipi.gr/reve"

    />

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

    <item name="floor_1" class="unreal">

      <virtualModel source="Floor">

```

```
        <transform translation="0, 0, 0"/>
    </virtualModel>
</item>
<item name="floor" class="unreal">
    <virtualModel source="Tavani">
        <transform translation="0, 3, 20"/>
    </virtualModel>
</item>
<item name="wall_01" class="unreal">
    <virtualModel source="Wall_01">
        <transform translation="4, 1.5, 0"/>
    </virtualModel>
</item>
<item name="wall_02" class="unreal">
    <virtualModel source="Wall_02">
        <transform translation="-4, 1.5, 0"/>
    </virtualModel>
</item>
<item name="wall_03" class="unreal">
    <virtualModel source="Wall_03">
        <transform translation="0, 1.5, 6"/>
    </virtualModel>
</item>
<item name="great_wall" class="unreal">
    <virtualModel source="Great_Wall_Door">
        <transform translation="-1, 1.5, -6"/>
    </virtualModel>
</item>
<item name="small_wall_01" class="unreal">
    <virtualModel source="Small_Wall_Door_01">
        <transform translation="3.75, 1.5, -6"/>
    </virtualModel>
</item>
```

```
<item name="small_wall_02" class="unreal">
    <virtualModel source="Small_Wall_Door_02">
        <transform translation="2.75, 2.5, -6"/>
    </virtualModel>
</item>
<item name="corridor_Wall" class="unreal">
    <virtualModel source="Corridor_Wall">
        <transform translation="1.95, 1.5, -6.5"/>
    </virtualModel>
</item>
<item name="Corridor_Wall_02" class="unreal">
    <virtualModel source="Corridor_Wall_02">
        <transform translation="3.55, 1.5, -9"/>
    </virtualModel>
</item>
<item name="Corridor_Wall_03" class="unreal">
    <virtualModel source="Corridor_Wall_03">
        <transform translation="1.95, 1.5, -8.5"/>
    </virtualModel>
</item>

<item name="Bathroom_Wall_01" class="unreal">
    <virtualModel source="Bathroom_Wall_01">
        <transform translation="0.9, 1.5, -6.95"/>
    </virtualModel>
</item>
<item name="Bathroom_Wall_02" class="unreal">
    <virtualModel source="Bathroom_Wall_02">
        <transform translation="0.9, 1.5, -8.05"/>
    </virtualModel>
</item>
```

```
<item name="Bathroom_Wall_03" class="unreal">
    <virtualModel source="Bathroom_Wall_03">
        <transform translation="0.9, 1.5, -8.95"/>
    </virtualModel>
</item>
<item name="Bathroom_Wall_04" class="unreal">
    <virtualModel source="Bathroom_Wall_04">
        <transform translation="0.9, 1.5, -10.05"/>
    </virtualModel>
</item>
<item name="Bathroom_Wall_05" class="unreal">
    <virtualModel source="Bathroom_Wall_05">
        <transform translation="2.75, 1.5, -12"/>
    </virtualModel>
</item>
<item name="Bathroom_Wall_06" class="unreal">
    <virtualModel source="Bathroom_Wall_06">
        <transform translation="0, 1.5, -9"/>
    </virtualModel>
</item>
<item name="Bathroom_Small_Wall" class="unreal">
    <virtualModel source="Bathroom_Small_Wall">
        <transform translation="1.95, 1.5, -11"/>
    </virtualModel>
</item>
<item name="Bathroom_Floor" class="unreal">
    <virtualModel source="Bathroom_Floor">
        <transform translation="2.75, 0, -9"/>
    </virtualModel>
</item>
<item name="Bathroom_Door_01" class="unreal">
    <virtualModel source="Bathroom_Door_01">
        <transform translation="3.48, 1, -6"/>
    </virtualModel>
</item>
```

```
</virtualModel>
</item>
<item name="Bathroom_Door_02" class="unreal">
  <virtualModel source="Bathroom_Door_02">
    <transform translation="2.02, 1, -6"/>
  </virtualModel>
</item>
<item name="Bathroom_Door_03" class="unreal">
  <virtualModel source="Bathroom_Door_03">
    <transform translation="1.9, 1, -7"/>
  </virtualModel>
</item>
<item name="Bathroom_Door_04" class="unreal">
  <virtualModel source="Bathroom_Door_04">
    <transform translation="1.9, 1, -9"/>
  </virtualModel>
</item>
</itemGroup>
<itemGroup name="Bathroom_utilities" location="bar/bar.wrl" behaviours="false">
  <item name="Toilet_1" class="unreal">
    <virtualModel source="Toilet_1">
      <transform translation="0.75,0.1,-9.5"/>
    </virtualModel>
  </item>
<item name="Toilet_2" class="unreal">
  <virtualModel source="Toilet_2">
    <transform translation="0.75,0.1,-7.5"/>
  </virtualModel>
</item>
</itemGroup>
<itemGroup name="DJ" location="bar/bar.wrl" behaviours="false">
```

```
<item name="Dj_01" class="unreal">
    <virtualModel source="Dj_01">
        <transform translation="-3,0.65,4.2"/>
    </virtualModel>
</item>

<item name="Dj_02" class="unreal">
    <virtualModel source="Dj_02">
        <transform translation="-3,0.15,5.1"/>
    </virtualModel>
</item>

<item name="Dj_03" class="unreal">
    <virtualModel source="Dj_03">
        <transform translation="-3,0.45,5.1"/>
    </virtualModel>
</item>
</itemGroup>

<itemGroup name="Bar_Table" location="bar/bar.wrl" behaviours="false">
    <item name="Bar_Table" class="unreal">
        <virtualModel source="Bar_Table">
            <transform translation="2,0.5,4.2"/>
        </virtualModel>
    </item>

<item name="Bar_Drinks" class="unreal">
    <virtualModel source="Bar_Drinks">
        <transform translation="0,1.1,5.85"/>
    </virtualModel>
</item>
</itemGroup>

<itemGroup name="Paintings" location="bar/bar.wrl" behaviours="false">
    <item name="Painting_01" class="unreal">
```



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

<item name="Painting_02" class="unreal">
  <virtualModel source="Painting_02">
    <transform translation="3.9,1.7,3"/>
  </virtualModel>
</item>

<item name="Painting_03" class="unreal">
  <virtualModel source="Painting_03">
    <transform translation="3.9,1.7,-3"/>
  </virtualModel>
</item>

<item name="Painting_04" class="unreal">
  <virtualModel source="Painting_04">
    <transform translation="-3.9,1.7,-2"/>
  </virtualModel>
</item>

<item name="Painting_05" class="unreal">
  <virtualModel source="Painting_05">
    <transform translation="-3.9,1.7,3"/>
  </virtualModel>
</item>

<item name="Painting_06" class="unreal">
  <virtualModel source="Painting_06">
    <transform translation="0,2,5.9"/>
  </virtualModel>
</item>

<item name="Painting_07" class="unreal">
  <virtualModel source="Painting_07">
```

```
<transform translation="-2,1.7,-5.9"/>
</virtualModel>
</item>
<item name="Painting_08" class="unreal">
  <virtualModel source="Painting_08">
    <transform translation="1,1.7,-5.9"/>
  </virtualModel>
</item>
</itemGroup>
<itemGroup name="Chairs" location="bar/bar.wrl" behaviours="false">
<item name="Chair_01" class="unreal">
  <virtualModel source="Chair_01">
    <transform translation="3.5,0.1,0"/>
  </virtualModel>
</item>
<item name="Chair_02" class="unreal">
  <virtualModel source="Chair_01">
    <transform translation="3.5,0.1,2"/>
  </virtualModel>
</item>
<item name="Chair_03" class="unreal">
  <virtualModel source="Chair_01">
    <transform translation="3.5,0.1,-2"/>
  </virtualModel>
</item>
<item name="Chair_04" class="unreal">
  <virtualModel source="Chair_01">
    <transform translation="2.5,0.1,-1"/>
  </virtualModel>
</item>
<item name="Chair_05" class="unreal">
```

```
<virtualModel source="Chair_01">
  <transform translation="2.5,0.1,1"/>
</virtualModel>
</item>
<item name="Chair_06" class="unreal">
  <virtualModel source="Chair_01">
    <transform translation="2.5,0.1,-3"/>
  </virtualModel>
</item>
<item name="Chair_07" class="unreal">
  <virtualModel source="Chair_01">
    <transform translation="-3.5,0.1,-2.5"/>
  </virtualModel>
</item>
<item name="Chair_08" class="unreal">
  <virtualModel source="Chair_01">
    <transform translation="-3.5,0.1,-0.5"/>
  </virtualModel>
</item>
<item name="Chair_09" class="unreal">
  <virtualModel source="Chair_01">
    <transform translation="-3.5,0.1,-4.5"/>
  </virtualModel>
</item>
<item name="Chair_10" class="unreal">
  <virtualModel source="Chair_01">
    <transform translation="-2.5,0.1,-3.5"/>
  </virtualModel>
</item>
<item name="Chair_11" class="unreal">
  <virtualModel source="Chair_01">
    <transform translation="-2.5,0.1,-1.5"/>
  </virtualModel>
```

```
</item>
<item name="Chair_12" class="unreal">
  <virtualModel source="Chair_01">
    <transform translation="-2.5,0.1,-5.5"/>
  </virtualModel>
</item>
</itemGroup>

<itemGroup name="Tables" location="bar/bar.wrl" behaviours="false">
<item name="Table_1a" class="real.general">
  <virtualModel source="Table_1a">
    <transform translation="0,0.1,-4.5"/>
  </virtualModel>
</item>
<item name="Table_1b" class="real.general">
  <virtualModel source="Table_1b">
    <transform translation="-2.3,0.1,-0.5"/>
  </virtualModel>
</item>
<item name="Table_1c" class="real.general">
  <virtualModel source="Table_1c">
    <transform translation="-2.3,0.9,-0.5"/>
  </virtualModel>
</item>
<item name="Table_1d" class="real.general">
  <virtualModel source="Table_1d">
    <transform translation="-2.15,0.9,-0.5"/>
  </virtualModel>
</item>
<item name="Table_1e" class="real.general">
  <virtualModel source="Table_1e">
    <transform translation="-2.35,0.9,-0.33"/>
  </virtualModel>
</item>
```

```
</item>
<item name="Table_1f" class="real.general">
  <virtualModel source="Table_1f">
    <transform translation="-2.55,0.9,-0.37"/>
  </virtualModel>
</item>
<item name="Table_2a" class="real.general">
  <virtualModel source="Table_2a">
    <transform translation="-2.3,0.1,-2.5"/>
  </virtualModel>
</item>
<item name="Table_2b" class="real.general">
  <virtualModel source="Table_2b">
    <transform translation="-2.3,0.9,-2.5"/>
  </virtualModel>
</item>
<item name="Table_2c" class="real.general">
  <virtualModel source="Table_2c">
    <transform translation="-2.15,0.9,-2.5"/>
  </virtualModel>
</item>
<item name="Table_2d" class="real.general">
  <virtualModel source="Table_2d">
    <transform translation="-2.35,0.9,-2.33"/>
  </virtualModel>
</item>
<item name="Table_2e" class="real.general">
  <virtualModel source="Table_2e">
    <transform translation="-2.55,0.9,-2.37"/>
  </virtualModel>
</item>
<item name="Table_3a" class="real.general">
  <virtualModel source="Table_3a">
```

```
<transform translation="-2.3,0.1,-4.5"/>
</virtualModel>
</item>
<item name="Table_3b" class="real.general">
  <virtualModel source="Table_3b">
    <transform translation="-2.3,0.9,-4.5"/>
  </virtualModel>
</item>
<item name="Table_3c" class="real.general">
  <virtualModel source="Table_3c">
    <transform translation="-2.15,0.9,-4.5"/>
  </virtualModel>
</item>
<item name="Table_3d" class="real.general">
  <virtualModel source="Table_3d">
    <transform translation="-2.35,0.9,-4.33"/>
  </virtualModel>
</item>
<item name="Table_3e" class="real.general">
  <virtualModel source="Table_3e">
    <transform translation="-2.55,0.9,-4.37"/>
  </virtualModel>
</item>
<item name="Table_4a" class="real.general">
  <virtualModel source="Table_4a">
    <transform translation="2.5,0.1,-2"/>
  </virtualModel>
</item>
<item name="Table_5a" class="real.general">
  <virtualModel source="Table_5a">
    <transform translation="2.5,0.1,2.13"/>
  </virtualModel>
</item>
```

```
<item name="Table_5b" class="real.general">
  <virtualModel source="Table_5b">
    <transform translation="2.3,0.9,2"/>
  </virtualModel>
</item>
<item name="Table_5c" class="real.general">
  <virtualModel source="Table_5c">
    <transform translation="2.35,0.9,1.8"/>
  </virtualModel>
</item>
<item name="Table_5d" class="real.general">
  <virtualModel source="Table_5d">
    <transform translation="2.65,0.9,2.1"/>
  </virtualModel>
</item>
<item name="Table_5e" class="real.general">
  <virtualModel source="Table_5e">
    <transform translation="2.55,0.9,2.2"/>
  </virtualModel>
</item>
<item name="Table_6a" class="real.general">
  <virtualModel source="Table_6a">
    <transform translation="2.5,0.1,0"/>
  </virtualModel>
</item>
</itemGroup>

</world>
</abr>
```