



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΦΙΛΟΣΟΦΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΦΙΛΟΛΟΓΙΑΣ
ΤΟΜΕΑΣ ΓΛΩΣΣΟΛΟΓΙΑΣ



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ ΕΛΕΓΧΟΥ ΚΑΙ
ΡΟΜΠΟΤΙΚΗΣ

ΔΙΕΠΙΣΤΗΜΟΝΙΚΟ ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΕΧΝΟΓΛΩΣΣΙΑ VIII

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Υλοποίηση της ΟΜΑΣ-III ως
Γραμματικού Φορμαλισμού για Ρομποτικές εφαρμογές

Ιωάννης Γιάχος

Επιβλέπων Καθηγητής:

Παπακίτσος Ευάγγελος

ΑΘΗΝΑ
ΣΕΠΤΕΜΒΡΙΟΣ 2015

ΕΙΣΑΓΩΓΗ

Προκειμένου να γίνει πλήρως κατανοητό το αντικείμενο με το οποίο ασχολείται η παρούσα εργασία, πρέπει αρχικά και προτού εκφράσουμε το στόχο (βλ. 1.2), να παραθέσουμε ορισμούς, έννοιες και πληροφορίες γενικότερα, για τα δεδομένα που θα αναφέρονται και θα χρησιμοποιούνται κατά τη διάρκεια της ανάπτυξής της (βλ. 1.1).

1.1 Βασικές Έννοιες & Ορισμοί.

Αρχικά επικεντρωνόμαστε στις λέξεις-εκφράσεις: Toki Pona, τεχνητή ή κατασκευασμένη γλώσσα, Φιλοσοφική γλώσσα, Φυσική Σημασιολογική Μεταγλώσσα, ROILA project, SostiMatiko, ΟΜΑΣ-III.

1.1.1 Τεχνητή / κατασκευασμένη γλώσσα.

Στο ξεκίνημα να αναφέρουμε ότι η *Toki Pona* είναι μία τεχνητή γλώσσα με την οποία θα ασχοληθούμε λίγο πιο κάτω (βλ. 1.1.5). **Τεχνητή ή κατασκευασμένη γλώσσα**¹ (γνωστή άτυπα στη διεθνή ορολογία και ως **conlang**), είναι μια γλώσσα της οποίας η φωνολογία, η γραμματική και το λεξιλόγιο, έχουν επινοηθεί συνειδητά από ένα άτομο ή μια ομάδα, αντί να έχουν εξελιχθεί φυσικά. Υπάρχουν πολλοί πιθανοί λόγοι για να δημιουργηθεί μια τεχνητή γλώσσα: από το να διευκολύνει την ανθρώπινη επικοινωνία, μέχρι να φέρει έναν φανταστικό ή κατασκευασμένο κόσμο στην ζωή. Επίσης για τον γλωσσικό πειραματισμό ή για λόγους καλλιτεχνικής δημιουργίας και έκφρασης. Τέλος μπορεί να δημιουργηθεί για *γλωσσικά παιχνίδια*. Μερικοί προτιμούν τον όρο (έκφραση) «*σχεδιασμένη γλώσσα*» αντί του "τεχνητή", καθώς ο τελευταίος μπορεί να έχει μια αρνητική χροιά σε κάποιες γλώσσες. Ο ίδιος όρος μερικές φορές χρησιμοποιείται για να δηλώσει μια διεθνή βοηθητική γλώσσα και άλλες γλώσσες που έχουν σχεδιαστεί για πραγματική χρήση στην ανθρώπινη επικοινωνία. Έξω από την Εσπεραντική² κοινότητα, ο όρος *γλωσσικός σχεδιασμός* σημαίνει τις οδηγίες που δίνονται σε μια φυσική γλώσσα με σκοπό την "κανονικοποίησή" της. Από αυτήν την άποψη, ακόμα και "φυσικές γλώσσες" μπορεί να είναι τεχνητές σε κάποιες εκφράσεις τους. Στην περίπτωση μιας *επίσημης γραμματικής*, όπου υπάρχουν καθολικοί (τεχνητοί ως επί το πλείστον) κανόνες, η διαχωριστική γραμμή είναι δύσκολο να αποσαφηνιστεί. Ο όρος *γλωσσοποιία* (αγγλ.: glossopoeia), που επινοήθηκε από τον Τζον Τόλκιν (ο οποίος κατασκεύασε την Κουένια³), χρησιμοποιήθηκε επίσης για να δηλώσει την κατασκευή της γλώσσας, ιδίως της καλλιτεχνικής.

1.1.2 Φιλοσοφική γλώσσα.

Φιλοσοφική γλώσσα⁴ (αγγλ.: Philosophical language) είναι κάθε τεχνητή γλώσσα που κατασκευάζεται από τους πρώτους βασικούς κανόνες, όπως μια λογική γλώσσα, αλλά μπορεί να αξιώνει την απόλυτη τελειότητα, την υπερβατικότητα, ακόμη και την μυστικιστική αλήθεια και όχι την ικανοποίηση των ρεαλιστικών στόχων. Οι

¹ https://en.wikipedia.org/?title=Constructed_language

² Εσπεράντο: η πιο διαδεδομένη τεχνητή γλώσσα.

³ Quenya: Η γλώσσα των ξωτικών, στα βιβλία του συγγραφέα. Έκανε την εμφάνισή της το 1917.

⁴ https://en.wikipedia.org/wiki/Philosophical_language

Φιλοσοφικές γλώσσες ήταν δημοφιλής στη νεότερη εποχή, εν μέρει ως κίνητρο από τον στόχο της ανάκτησης της χαμένης Αδαμικής⁵ ή Θεϊκής γλώσσας (Adamic or Divine). Ο όρος ιδανική γλώσσα μερικές φορές χρησιμοποιείται σχεδόν συνώνυμα, αν και οι πιο σύγχρονες φιλοσοφικές γλώσσες όπως η Toki Pona είναι λιγότερο πιθανό να αγγίζουν μια τέτοια υψηλή απαίτηση της τελειότητας. Τα αξιώματα και οι γραμματικές των γλωσσών διαφέρουν από τις κοινώς ομιλούμενες γλώσσες σήμερα. Στις περισσότερες γηραιότερες φιλοσοφικές γλώσσες, και κάποιες νεότερες, οι λέξεις κατασκευάζονται από ένα περιορισμένο σύνολο μορφημάτων που αντιμετωπίζονται ως "στοιχειώδης" ή θεμελιώδης.

Τα λεξιλόγια των ολιγοσυνθετικών⁶ (oligosynthetic) γλωσσών κατασκευάζονται από σύνθετες λέξεις, οι οποίες επινοήθηκαν από ένα μικρό (θεωρητικά ελάχιστο) αριθμό μορφημάτων. Ομοίως οι oligoisolating γλώσσες, όπως η Toki Pona, χρησιμοποιούν ένα περιορισμένο σύνολο λέξεων ριζών, αλλά παράγουν φράσεις που αποτελούν σειρές από διακριτές λέξεις.

Η Láadan⁷ έχει σχεδιαστεί για να λεξικολογοποιεί και να γραμματικοποιεί τις έννοιες και διακρίσεις που είναι σημαντικές για τις γυναίκες, με βάση την "muted group theory"⁸.

Η Toki Pona βασίζεται σε μινιμαλιστική απλότητα, ενσωματώνοντας στοιχεία του Ταοϊσμού.

Οι "a priori" (εξ αρχής) είναι κατασκευασμένες γλώσσες, όπου το λεξιλόγιο δημιουργήθηκε ευθύς εξαρχής (π.χ. Dama Diwan⁹), αντί να προέρχεται από άλλες υπάρχουσες γλώσσες (όπως η Εσπεράντο ή η Ιντερλίνγκουα). Οι Φιλοσοφικές γλώσσες είναι σχεδόν όλες "a priori", αλλά οι περισσότερες "a priori" γλώσσες δεν είναι φιλοσοφικές. Για παράδειγμα, Κουέννα, Σίνταριν και Κλίνγκον (Quenya, Sindarin, and Klingon) είναι όλες "a priori", αλλά όχι φιλοσοφικές γλώσσες. Στόχος τους είναι να φαίνονται σαν φυσικές γλώσσες, ακόμη και αν δεν έχουν καμία γενετική σχέση με οποιαδήποτε φυσική γλώσσα.

Ιστορικά οι Φιλοσοφικές γλώσσες εμφανίζονται το 1647 με πρωτοπόρο τον Francis Lodwick. Αξιοσημείωτο είναι ότι το 1678 ο Gottfried Leibniz, με στόχο να δημιουργήσει ένα λεξικό των χαρακτήρων κατά την οποία ο χρήστης μπορεί να εκτελέσει υπολογισμούς που θα δώσει πραγματικές προτάσεις αυτόματα, ως παρενέργεια ανέπτυξε τον δυαδικό λογισμό.

Σε αυτά τα χρόνια δημιουργήθηκαν προγράμματα (projects) που σκοπό είχαν όχι μόνο να μειωθεί ή να μοντελοποιηθεί η γραμματική, αλλά και να οργανωθεί όλη η ανθρώπινη γνώση σε «χαρακτήρες» ή ιεραρχίες. Αυτή η ιδέα οδήγησε τελικά στην Εγκυκλοπαίδεια, στην εποχή του Διαφωτισμού. Ο Leibniz και οι εγκυκλοπαιδιστές συνειδητοποίησαν ότι είναι αδύνατο να οργανωθεί η ανθρώπινη γνώση κατηγορηματικά, όπως ένα δέντρο, και έτσι είναι αδύνατο να κατασκευαστεί εκ των προτέρων γλώσσα που βασίζεται σε μια τέτοια ταξινόμηση των εννοιών.

Μετά την Εγκυκλοπαίδεια, τα σχέδια για μια "a priori" γλώσσα μετακινούνται όλο και περισσότερο στο περιθώριο.

⁵ Η γλώσσα που σύμφωνα με τους Εβραίους (όπως καταγράφεται στο Midrashim) αλλά και ορισμένους χριστιανούς, ομιλείτο από τον Αδάμ στον Παράδεισο.

⁶ https://en.wikipedia.org/wiki/Oligosynthetic_language

⁷ <https://en.wikipedia.org/wiki/Láadan> και <http://www.laadanlanguage.org/>

⁸ Θεωρία που διερευνά την ασύμμετρη σχέση μεταξύ κυρίαρχων και ήσυχων ομάδων

⁹ http://linguifex.com/wiki/Dama_Diwan

1.1.3 Φυσική Σημασιολογική Μεταγλώσσα.

Η Φυσική Σημασιολογική Μεταγλώσσα¹⁰ (αγγλ.: Natural Semantic Metalanguage) NSM, είναι μια γλωσσολογική θεωρία και πρακτική, που σκοπεύει στο να εξαλειφθούν όλες οι συγχύσεις της διαπολιτισμικής (cross-cultural) επικοινωνίας. Αυτό πετυχαίνεται με τη χρήση ενός συνόλου βασικών και καθολικών εννοιών, γνωστών ως semantic primes, οι οποίες μπορούν να εκφραστούν με λόγια ή άλλες γλωσσικές εκφράσεις σε όλες τις γλώσσες. Η θεωρία της NSM έκανε την εμφάνισή της το 1972 στο βιβλίο "Semantic Primitives" της Πολωνό-Αυστραλιανής καταγωγής γλωσσολόγου Anna Wierzbicka και βασίζεται σε μια μακραίωνη ιδέα μιας γλώσσας του νου. Σήμερα αναγνωρίζεται διεθνώς ως μία από τις κορυφαίες θεωρίες στον κόσμο της γλώσσας και του νοήματος.

Η χρήση της NSM μας επιτρέπει την ανάπτυξη αναλύσεων που είναι σαφείς, ακριβείς, cross-translatable (δια μεταφραστέες), μη Αγγλοκεντρικές, και κατανοητές από ανθρώπους χωρίς εξειδικευμένη γλωσσική κατάρτιση. Η μέθοδος έχει εφαρμογές στη διαπολιτισμική επικοινωνία, τη λεξικογραφία, τη διδασκαλία γλωσσών, τη μελέτη πάνω στην *κατάκτηση της μητρικής γλώσσας* (child language acquisition), και σε άλλους τομείς. Παρακάτω παρουσιάζονται κάποιες NSM έννοιες (primes) κωδικοποιημένες ως αγγλικές λέξεις. Αυτές οι έννοιες υποτίθεται ότι είναι γλωσσικά καθολικές, με τις περισσότερες από αυτές να έχουν δοκιμαστεί σε μια ευρεία ποικιλία γλωσσών χωρίς να αντιμετωπίζουν αβεβαιότητες. Είναι πολύ σημαντικό να συνειδητοποιήσουμε ότι ορισμένοι από τους εκθέτες στην ακόλουθη λίστα (Πίν. 1.0.1) έχουν σημασία μόνο στην Αγγλική γλώσσα, αλλά όταν χρησιμοποιούνται ως εκθέτες στην NSM μας αφορά μόνο η καθολική τους έννοια:

Πίνακας 1.0.1.

PROPOSED AND EXPERIMENTALLY SUPPORTED SEMANTIC PRIMES						
CATEGORY	PRIMES					
Substantives	I	YOU	SOMEONE/ PERSON	PEOPLE		
Relational Substantives	SOMETHING/ THING	BODY	KIND	PART		
Determiners	THIS	THE SAME	OTHER			
Quantifiers	ONE	TWO	SOME	ALL	MANY/MUCH	
Evaluators	GOOD	BAD				
Descriptors	BIG	SMALL				
Mental/Experiential Predicates	THINK	KNOW	WANT	FEEL	SEE	HEAR
Speech	SAY	WORDS	TRUE			
Actions And Events	DO	HAPPEN	MOVE			
Existence And Possession	THERE IS/ EXIST	HAVE				
Life And Death	LIVE	DIE				
Time	WHEN/TIME	NOW	BEFORE	AFTER	A LONG TIME	A SHORT TIME
	FOR SOME TIME	MOMENT				

¹⁰ https://en.wikipedia.org/wiki/Natural_semantic_metalanguage

	WHERE/PLACE,	HERE	ABOVE	BELOW	FAR	NEAR
Space	SIDE	INSIDE	TOUCH (CONTACT)			
Logical Concepts	NOT	MAYBE	CAN	BECAUSE	IF	
Intensifier, Augmentor	VERY	MORE				
Similarity	LIKE/WAY					

Σαν παράδειγμα μπορούμε να δώσουμε μία πρόταση στην Αγγλική χωρίς χρήση καθολικών εννοιών (non-prime concept), ακολουθούμενη από αντίστοιχες προτάσεις που συνθέτουν το ίδιο νόημα γραμμένες με semantic primes:

Someone X killed someone Y:
 someone X did something to someone else Y
 because of this, something happened to Y at the same time
 because of this, something happened to Y's body
 because of this, after this Y was not living anymore

Ο όρος μεταγλώσσα χρησιμοποιείται και στη γλωσσολογία, αλλά και στην επιστήμη γενικότερα, ιδιαίτερα στους υπολογιστές.

Στη γλωσσολογία, μεταγλώσσα ονομάζεται ένα σύνολο από λέξεις, φράσεις, όρους, σημεία και σύμβολα, που χρησιμοποιούνται για να μιλήσει κάποιος, να περιγράψει ή να αναλύσει την ίδια τη γλώσσα (την «πραγματική» γλώσσα που χρησιμοποιούμε για να μιλήσουμε για τον κόσμο μας). Έτσι, μια μεταγλώσσα μπορεί να θεωρηθεί ως μια γλώσσα για μια άλλη γλώσσα. Μια τέτοια γλώσσα μπορεί να χρησιμοποιηθεί για να διατυπωθούν κανόνες, νόμοι, θεωρίες ή σχέσεις που αφορούν στην πραγματική γλώσσα.

Ας υποθέσουμε ότι μια μητέρα λέει σε μια φίλη της για το γιο της: «Ο Γιαννάκης μου, το χρυσό μου, κάθε απόγευμα μελετάει τα μαθήματά του προσεχτικά».

Όταν πούμε: το «Ο» είναι οριστικό άρθρο, αρσενικού γένους, ενικού αριθμού, ονομαστικής πτώσης, το «Γιαννάκης» είναι ουσιαστικό, κύριο όνομα υποκοριστικό, αρσενικού γένους, ονομαστικής πτώσης, το «μου» είναι κτητική αντωνυμία, πρώτου προσώπου, που δηλώνει ένα κτήτορα, κ.ο.κ., όλα αυτά είναι μια μεταγλώσσα που χρησιμοποιείται για να αναλύσει στο γραμματικό επίπεδο τα λόγια της πραγματικής γλώσσας που ειπώθηκε από κάποια μητέρα. Οι όροι αυτής της μεταγλώσσας είναι: *οριστικό άρθρο, αρσενικού γένους, ενικού αριθμού, ονομαστικής πτώσης, κτητική αντωνυμία, πρώτου προσώπου, κτλ.*

Όταν πούμε: «Ο Γιαννάκης μου, το χρυσό μου» είναι το ονοματικό μέρος της πρότασης που αναλύεται στο υποκείμενο «Ο Γιαννάκης μου» και στον ομοιόπρωτο προσδιορισμό-παράθεση «το χρυσό μου» και το «κάθε απόγευμα μελετάει τα μαθήματά του προσεχτικά» είναι το ρηματικό μέρος που αναλύεται στο ρήμα «μελετάει», στο αντικείμενο «τα μαθήματά του», στον επιρρηματικό προσδιορισμό του χρόνου «κάθε απόγευμα» και στον επιρρηματικό προσδιορισμό του τρόπου «προσεχτικά», όλα αυτά είναι μια μεταγλώσσα που χρησιμοποιείται για να αναλύσει στο συντακτικό επίπεδο τα πιο πάνω λόγια της πραγματικής γλώσσας. Όροι αυτής της μεταγλώσσας είναι: *ονοματικό μέρος, πρότασης, υποκείμενο, ομοιόπρωτο προσδιορισμό, παράθεση, ρηματικό μέρος, ρήμα, αντικείμενο, επιρρηματικό προσδιορισμό του χρόνου, επιρρηματικό προσδιορισμό του τρόπου.*

Δηλαδή, θα μπορούσε κατά μία έννοια να θεωρηθεί η πραγματική γλώσσα ως ένα κείμενο, και η μεταγλώσσα ως ένα σχόλιο πάνω σ' αυτό το κείμενο.

1.1.4 Robot Interaction Language (ROILA).

Το **ROILA (Robot Interaction Language) project**¹¹ είναι μία διεθνής ανοιχτή εργασία σε εξέλιξη που σκοπό έχει να αναπτύξει μία γλώσσα αποκλειστικά για ρομπότ. Για ποιο λόγο θα θέλαμε να φτιάξουμε μία τέτοια γλώσσα; Αν αντιμετωπίσουμε την ταχύτατη αύξηση των μηχανών που προορίζονται στο να αλληλεπιδρούν με τους ανθρώπους, τόσο στην προσωπική όσο και στην επαγγελματική τους ζωή, ως μία αναδυόμενη κοινωνία, τότε θα καταλάβουμε την επιτακτική ανάγκη της απόλυτης κατανόησης μεταξύ των εμπλεκομένων! Ο ευκολότερος τρόπος για να επικοινωνεί ο άνθρωπος με τέτοιου είδους ρομπότ θα ήταν η φυσική γλώσσα. Η τρέχουσα τεχνολογία αναγνώρισης ομιλίας όμως, δεν έχει φτάσει ακόμη στο επίπεδο στο οποίο θα είναι αυτό εφικτό σε αρκετά μεγάλο βαθμό. Συχνά τα ρομπότ παρερμηνεύουν τα λόγια ή δεν είναι σε θέση να κατανοήσουν τι τους λένε. Ορισμένοι ερευνητές υποστηρίζουν ότι η αναγνώριση ομιλίας από τις μηχανές δεν θα φτάσει ποτέ στο επίπεδο του ανθρώπου. Έτσι διαπιστώνεται πως ο πιο εύκολος τρόπος για να κατανοούν πλήρως τα ρομπότ τον άνθρωπο είναι η κατασκευή μιας νέας γλώσσας στα δικά τους μέτρα. Η γλώσσα των ρομπότ θα πρέπει να αντιμετωπίζει τα προβλήματα που σχετίζονται με την αλληλεπίδραση της ομιλίας, που εισάγει η χρήση των φυσικών γλωσσών, καθώς και να είναι εύκολο για τον άνθρωπο να την μάθει χωρίς ιδιαίτερες γνώσεις και επίπεδο μόρφωσης. Υπό αυτό το σκεπτικό γεννήθηκε η ROILA και πληροί τα παραπάνω. Έχει απλή γραμματική χωρίς να παρουσιάζονται ανωμαλίες και οι λέξεις της αποτελούνται από φωνήματα που μοιράζονται μεταξύ της πλειοψηφίας των φυσικών γλωσσών. Το σύνολο των μεγάλων φωνημάτων δημιουργήθηκε από την επισκόπηση των φυσικών γλωσσών. Επιπλέον, έχει συντεθεί ένας γενετικός αλγόριθμος που δημιουργεί λέξεις ROILA με τρόπο που είναι εύκολο να προφερθεί. Ο ίδιος αλγόριθμος εξασφαλίζει ότι κάθε μία από αυτές τις λέξεις ακούγεται διαφορετικά από τις άλλες στο μέτρο του δυνατού. Αυτό βοηθά το σύστημα φωνητικής αναγνώρισης της μηχανής να κατανοήσει με ακρίβεια τον ομιλητή άνθρωπο.

Όπως αρχικά αναφέρθηκε, το project είναι σε εξέλιξη και βελτιώνει συνεχώς τις διαδικασίες αναγνώρισης και παραγωγής φωνής. Καλεί ανθρώπους που ασχολούνται ή θέλουν να ασχοληθούν με το αντικείμενο τόσο στο να εκπαιδευτούν πάνω στη ROILA όσο και να γίνουν μέρος της εξέλιξής της. Από τον ιστότοπο της ομάδας ROILA [] διατίθενται όλες οι πληροφορίες και όλα τα υπάρχοντα εργαλεία για όποιον ενδιαφερόμενο.

Ακολουθεί το αλφάβητο της ROILA (Πίν. 1.0.2) μαζί με την προφορά στο IPA (Διεθνές Φωνητικό Αλφάβητο) και στο ARPABET¹² (φωνητική γραφή, κώδικας που αναπτύχθηκε από την Advanced Research Projects Agency, ως μέρος των εργασιών αναγνώρισης ομιλίας που διεξήγαγαν μεταξύ 1971 και 1976). Στην τελευταία στήλη παρουσιάζεται μία Αγγλική λέξη που χρησιμοποιεί το αντίστοιχο γράμμα.

¹¹ <http://roila.org>

¹² <https://en.wikipedia.org/wiki/Arpabet>

Πίνακας 1.0.2

letter	IPA pronouciation	ARPABET pronouciation	example
a	æ	<u>A</u> E	fast
e	ε	<u>E</u> H	red
i	ι	<u>I</u> H	big
o	ο	<u>A</u> O	frost
u	υ	<u>A</u> H	but
b	b	<u>B</u>	buy
f	f	<u>F</u>	for
j	dʒ	<u>J</u> H	just
k	k	<u>K</u>	key
l	l	<u>L</u>	late
m	m	<u>M</u>	man
n	n	<u>N</u>	no
p	p	<u>P</u>	pay
s	s	<u>S</u>	say
t	t	<u>T</u>	take
w	w	<u>W</u>	way

Το λεξιλόγιο ROILA έχει τρεις τύπους λέξεων:

- i) CVCV
- ii) CVCVC
- iii) CVCVCV

όπου C μπορεί να είναι οποιοδήποτε από τα 11 σύμφωνα και V οποιοδήποτε από τα 5 φωνήεντα, όπως υποδεικνύεται στον παραπάνω πίνακα. Αυτοί οι τύποι λέξεων είχε εμπειρικά αποδειχθεί ότι αποδίδουν καλύτερα στην αναγνώριση ομιλίας σε σύγκριση με τις μη-CV τύπου λέξεις.

Επειδή δεν έχουμε διφθόγγους, τα φωνήεντα αρθρώνονται με τον ίδιο τρόπο, ανεξάρτητα από τη θέση τους μέσα στη λέξη, ενώ για τα σύμφωνα η προφορά είναι ίδια με αυτή των συμφώνων εντός των Αγγλικών λέξεων.

Η γραμματική της ROILA είναι απλή. Οι κανόνες ισχύουν εξίσου για όλες τις λέξεις και δεν υπάρχει καμία εξαίρεση. Μετά από το ρήμα υπάρχει περίπτωση να προστεθεί μία λέξη ώστε να δηλώσει χρόνο, παρελθόντα ή μέλλοντα, όταν αυτό απαιτείται. Μετά από ουσιαστικό μπορεί να προστεθεί μία λέξη ώστε να δηλώσει την πλειονότητα όταν χρειαστεί. Καμία κλίση και τροποποίηση λέξης δεν υφίσταται.

Η ROILA έχει τα ακόλουθα μέρη του λόγου: ουσιαστικά, ρήματα, επιρρήματα, επίθετα και 4 αντωνυμίες (I, you, he, she - εγώ, εσύ, αυτός και αυτή).

Τα ονόματα των ατόμων είναι τα ίδια με αυτά των φυσικών γλωσσών.

Το πρώτο γράμμα της λέξης στην αρχή πρότασης κεφαλαιοποιείται.

Το φύλο δεν σημειώνονται στη ROILA, τόσο για τα έμψυχα όντα, όσο και για τα άψυχα αντικείμενα. Το φύλο δηλώνεται μόνο διαμέσου των αντωνυμιών (αυτός / αυτή).

Μόνο ο πληθυντικός αριθμός δηλώνεται, όπως προαναφέραμε και αυτό βάζοντας τη λέξη "Tuji" μετά το ουσιαστικό. Η έννοια του "Tuji" είναι "πολύ".

I love this fruit
Pito loki wikute
I love fruit

I love all fruits
Pito loki wikute tuji
I love fruit <word marker for plural>

Αναφορές στο πρόσωπο γίνεται με τη χρήση των «εγώ» και «εσύ» (Pito και Bama αντίστοιχα). Αυτό είναι αρκετό, αφού η ROILA προορίζεται κυρίως για τη διαδραστική επικοινωνία μεταξύ δύο ομιλητών. Οι αντωνυμίες αυτός και αυτή μπορούν να αντικατασταθούν από τη χρήση των ονομάτων αυτών. Ωστόσο, η εκπροσώπηση των δύο φύλων γίνεται με τις λέξεις "liba" (αυτός) και "mona" (αυτή).

I can go left or right
Pito leto fosit webufo buno besati

Οι Χρόνοι κατανέμονται στα βασικά τρία επίπεδα: παρελθόν, παρόν και μέλλον. Για τον Ενεστώτα συνεπάγεται κανονική φράση χωρίς αλλαγές. Για το Παρελθόν προστίθεται η λέξη "jifi" και για το Μέλλον η λέξη "Jifo", μετά το ρήμα. Για παράδειγμα:

I am walking to the house
Pito fosit bubas
I walk house

I walked to the house
Pito fosit jifi bubas
I walk <word marker for past tense> house

I will walk to the house
Pito fosit jifo bubas
I walk <word marker for future tense> house

Η πολικότητα απλά θα εκπροσωπείται για τα «ναι/ok/καλό» (yes/ok/good) από τη "wopa" και για το «δεν/όχι» (no) από το "Buse".

Do not listen to her
Buse lulaw mona
No listen her

Η ερώτηση δηλώνεται με τη λέξη "biwu", η οποία μεταφράζεται κυριολεκτικά σε Τι.

What color is the museum?
Biwu wekepo buse kulil bubas?
What color not new house

Οι σύνδεσμοι «και» (and) με το “sowu”, ή (or) με το “buno”, επειδή (because) με το “kijo”.

I think this museum is very dirty and bad
Pito wetok pimo buse kulil bubas topik sowu bujeti
I think that not new house dirty and bad

Οι προτάσεις είναι της μορφής SVO (υποκείμενο, ρήμα, αντικείμενο).

He (S) saw (V) (the) bird (O)
Liba (S) make jifi (V) mipuki (O)
He see <word marker for past tense> bird

Στίξη. Κάθε πρόταση ολοκληρώνεται με μια τελεία (.). Ερωτηματικά μπορούν να χρησιμοποιηθούν σε φράσεις όπου διατυπώνεται μια ερώτηση. Δεν υποστηρίζονται κόμματα, απόστροφοι και εισαγωγικά.

Ελλείψεις στη γραμματική σε σχέση με τις φυσικές γλώσσες εμφανίζονται. Κατά τον σχεδιασμό της γραμματικής ROILA λάβαμε υπόψη μόνο ορισμένα κριτήρια και αυτό σημαίνει ότι κάποιες γραμματικές κατηγορίες αποσύρθηκαν εντελώς. Οι κατηγορίες αυτές μπορεί να βρεθούν σε πολλές φυσικές γλώσσες. Αυτό είναι ένα θέμα που εξετάζεται.

- Ορισμένοι Χρόνοι, όπως οι αποδιδόμενοι Perfect της Αγγλικής (υπερσυντέλικος, συντελεσμένος, παρακείμενος), δεν υποστηρίζονται πλήρως. Σε ορισμένες περιπτώσεις, θα ήταν δυνατό να εκφραστούν αυτοί οι χρόνοι με την αναδιατύπωση της πρωτότυπης πρότασης.

I am going (I am about to go) / Pimo kapim kanek / I about go

- Η δήλωση κτήσης στην Αγγλική με το «'s» δεν υποστηρίζεται. Η κτήση δηλώνεται μέσω αντωνυμιών, άρα η πρόταση πρέπει να αναδιατυπωθεί κατάλληλα.

*This is Omar's book (This is the book of Omar) / Bamas fojato fomu Omar /
This book of Omar*

- Το ποιόν ενέργειας¹³ (Aspect) μπορεί απλά (εύκολα) να εναλλάσσεται με τους χρόνους στις περισσότερες γλώσσες. Στην ROILA αυτή η δυνατότητα δεν υποστηρίζεται ρητά.

- Η τροπικότητα¹⁴ (Modality) μπορεί να εκφραστεί εν μέρει από τη χρήση λέξεων όπως «μπορεί» αντί «ίσως» για την υποτακτική:

He might be ill (He may be ill) / Liba bemotu mufe pubemu

¹³ **Ποιόν ενέργειας** είναι μια μορφολογική κατηγορία που αναφέρεται στον τρόπο με τον οποίο παρουσιάζει ο ομιλητής το αν η ενέργεια που δηλώνει το ρήμα εμφανίζεται ως ολοκληρωμένη, ως εξελισσόμενη, ως μοναδικό γεγονός κτλ.

¹⁴ τροπικότητα είναι «μια μορφοσυντακτική κατηγορία που δηλώνει τη σχέση του ομιλητή προς το εκφώνημα και του εκφωνήματος προς την πραγματικότητα»

- Η Ενεργητική φωνή είναι προεπιλεγμένη στη ROILA. Έτσι το θέμα ή το υποκείμενο είναι πάντα πριν από το ρήμα. Ως εκ τούτου, δεν υπάρχει άμεση στήριξη για την έκφραση παθητική φωνής.
- Τα άρθρα δεν αποτελούν μέρος του λεξιλογίου της ROILA, οπότε δεν υπάρχει «a», «an» ή «the».

Στο λεξιλόγιο της ROILA υπάρχουν 850 λέξεις συνολικά και αποτελεί μια συνοπτική εκδοχή αυτού της Αγγλικής, εφόσον οι σημασίες των λέξεων λαμβάνονται από τα βασικά Αγγλικά.

1.1.5 Toki Pona.

Η **Toki Pona**¹⁵, όπως προαναφέρθηκε, είναι μια κατασκευασμένη γλώσσα. Σχεδιάστηκε από την μεταφράστρια και γλωσσολόγο **Sonja Lang**, από το Τορόντο. Δημοσιεύτηκε για πρώτη φορά στο διαδίκτυο στα μέσα του 2001. Είναι μία μινιμαλιστική γλώσσα και όπως όλα τα γλωσσικά κατασκευάσματα (Pidgin), επικεντρώνεται στις απλές έννοιες και τα σχετικά κοινά στοιχεία μεταξύ των πολιτισμών. Η Toki Pona σχεδιάστηκε για να εκφράσει τη μέγιστη σημασία με ελάχιστη πολυπλοκότητα και αξιώνει να είναι η πιο απλή γλώσσα στον κόσμο και συνάμα ιδανική για απόδοση βασικών εννοιών. Κατά τη δημιουργία των λέξεων της γλώσσας αυτής αξιοποιήθηκαν μετρήσεις που παρέχονται από το Τμήμα Ψυχολογίας της Γάνδης, ώστε οι πιο συχνά εμφανιζόμενες λέξεις (ως Αγγλική έννοια πάντα) να έχουν το μικρότερο μήκος στο λεξιλόγιο. Αυτό συνεπάγεται την χρήση του ελάχιστου αριθμού γραμμάτων. Έχει 14 φωνήματα και 120 λέξεις ρίζας. Όλες οι λέξεις της ενσωματώνονται στο λεξιλόγιο της ROILA και αυτός είναι ο λόγος που προτείνεται να ξεκινήσει κανείς την εκπαίδευσή του με αυτή και μετέπειτα να επεκταθεί στη ROILA.

1.1.6 SostiMatiko.

Η **SostiMatiko**¹⁶ είναι επίσης μία κατασκευασμένη γλώσσα. Έχει σημασία για εμάς γιατί είναι Ελληνικής καταγωγής, συνεπώς οι λέξεις της προέρχονται από Ελληνικές ρίζες. Είναι μια πολύ εύκολη γλώσσα στο να τη μάθει κανείς ενώ μέσα από αυτή μπορεί να εκφράσει σχεδόν τα πάντα. Έτσι όπως και η Toki Pona μπορεί να εκφράσει τη μέγιστη σημασία με ελάχιστη πολυπλοκότητα. Το λεξιλόγιό της αποτελείται από μόνο 222 λέξεις, ενώ ταυτόχρονα έχει απεριόριστες δυνατότητες στην παραγωγή νέων λέξεων με πολύ απλό τρόπο. Δεν θα επεκταθούμε περισσότερο σε αυτό το σημείο, καθώς στο κεφάλαιο της Ανάλυσης που ακολουθεί γίνεται μία εκτενής αναφορά σε αυτή.

1.1.7 ΟΜΑΣ-III.

Η **ΟΜΑΣ-III**¹⁷ είναι μια σχεδιασμένη διαδικασία για την επίτευξη του άριστου δυνατού προσδιορισμού της οργάνωσης (δομής και λειτουργίας/συμπεριφοράς) ενός αντικειμένου ή φαινομένου (συστήματος), σύμφωνα με την εφαρμογή βασικών οργανωτικών κανόνων, προσαρμοσμένων σε συγκεκριμένες συνθήκες. Η ΟΜΑΣ

¹⁵ http://en.wikipedia.org/wiki/Toki_Pona

¹⁶ www.unilang.org/blog/SostiMatiko/

¹⁷ Παπακίτσος 2013

ανήκει στην οικογένεια τεχνικών SADT¹⁸ και IDEFx αποτελώντας σχεδιαστική εξέλιξή τους. Η ΟΜΑΣ-III είναι η τρίτη βελτιωμένη έκδοση της αρχικής μεθόδου. Σε αυτό το σημείο να πούμε εν συντομία (εφόσον θα επακολουθήσει εκτενής αναφορά), ότι η πλήρης κατανόηση ενός συστήματος δια μέσου της συγκεκριμένης μεθόδου, απαιτεί να δοθούν απαντήσεις στα μοναδικά επτά θεμελιώδη ερωτήματα που το αφορούν:

- Γιατί υπάρχει και λειτουργεί;
- Τι αποτελέσματα και συμπεράσματα δίνει;
- Πόσα μέσα χρειάζεται (πόρους);
- Πώς λειτουργεί;
- Ποιος παρακολουθεί ή καθοδηγεί τη λειτουργία του;
- Πού λειτουργεί;
- Πότε λειτουργεί;

Η κατανόηση του συστήματος οδηγεί στην πλήρη περιγραφή του ή και αντίστροφα, η πλήρης περιγραφή του οδηγεί στην κατανόησή της **δομής** (η διάρθρωση ενός συνόλου) και της **οργάνωσης** και **λειτουργίας** του (η διευθέτηση που αφορά τις σχέσεις μεταξύ των οντοτήτων της).

1.2 Σκοπός της εργασίας.

Έχοντας λοιπόν υπόψη μας όλα τα παραπάνω μπορούμε πλέον να θέσουμε με πολύ απλά λόγια ποιος είναι ο σκοπός της παρούσας εργασίας:

Βασιζόμενοι στο ROILA Project έχουμε ως δεδομένο ότι μπορούμε να μιλήσουμε σε ένα ρομπότ και να το καθοδηγήσουμε στο να εκτελέσει κάποιες εντολές – κινήσεις. Αυτό γίνεται μιλώντας του σε μία δική του γλώσσα (ROILA ή Toki Pona). Εδώ δεν μας αφορά πώς το ρομπότ αναλύει ηλεκτρονικά την όποια δοθείσα ηχητική εντολή, συνεπώς θεωρούμε πως η μηχανή (συνομιλητής μας) λαμβάνει προτάσεις και έχει πλήρη επίγνωση των λέξεων που συμπεριλαμβάνονται σε αυτές (και τη σειρά τους φυσικά). Από το σημείο αυτό αρχίζουν να γεννιούνται διάφορα ερωτήματα. Λέμε για παράδειγμα στο ρομπότ να προχωρήσει εμπρός: Πόσο εμπρός; Του λέμε να στρίψει προς μία κατεύθυνση: Πότε; Του λέμε να κάνει κάτι αλλά μήπως πρέπει να του πούμε και τον τρόπο, που ίσως κάθε φορά είναι διαφορετικός; Μπορούμε να του περιγράψουμε μία πορεία η οποία μπορεί να δίνεται σε άνω της μίας πρότασης; Στο τελευταίο η απάντηση είναι: όχι χωρίς τη χρήση κάποιου συστημικού μοντέλου. Για το σύνολο όμως των ερωτημάτων ένα συστημικό μοντέλο θα φροντίζει να προλαμβάνει και να δίνει απαντήσεις σε τέτοια απλά ερωτήματα, αλλά και πολύ πιο σύνθετα, όπως η ανταλλαγή πληροφοριών. Επειδή τα ερωτήματα αυτά ταυτίζονται με αυτά της ΟΜΑΣ-III πειραματιζόμαστε με αυτό το μοντέλο ως προς την ικανότητά του να τα καλύψει.

Ένα δεύτερο σημείο της εργασίας είναι ότι θα χρησιμοποιήσουμε την SostiMatiko ως γλώσσα επικοινωνίας με το ρομπότ, και όχι την Toki Pona. Θα πάρουμε ως δεδομένο πως το ρομπότ μας μιλάει τη **SostiMatiko**, είτε με ενσωμάτωση είτε με ενδιάμεσο μεταφραστικό σύστημα, δεν μας αφορά το πως.

Συγκεντρώνοντας τα παραπάνω, είναι η ώρα να προσδιορίσουμε το στόχο μας: «Θέλουμε να επικοινωνούμε με ένα ρομπότ, στην Ελληνική **SostiMatiko**, με σημασιολογική επαύξηση μέσω του συστημικού μοντέλου ΟΜΑΣ-III. Αυτό θα μας δώσει τη δυνατότητα η επικοινωνία να γίνεται με τη μορφή σύνθετων εντολών ή με ανταλλαγή πληροφοριών, σε ελάχιστη φυσική γλώσσα». Να επιστημόνουμε εδώ ότι

¹⁸ https://en.wikipedia.org/wiki/Structured_Analysis_and_Design_Technique

ως ελάχιστη φυσική γλώσσα θα θεωρήσουμε την Ελληνική σε πολύ απλοϊκή μορφή (παραιτώντας αρκετούς γραμματικούς κανόνες) και ότι ως αντικείμενο συνδιαλλαγής δεν θα είναι ένα ρομπότ αλλά ένα Demo πρόγραμμα που θα του μεταφέρουμε εντολές μέσω αρχείου κειμένου.

Αξίζει εδώ να επισημάνουμε πως με τις δυνατότητες που δίνει η **SostiMatiko** και με τη χρήση του ΟΜΑΣ-III, μία μηχανή ίσως να μπορεί να εξελίξει το επίπεδο επικοινωνίας, όντας σε θέση να συνθέτει νέες λέξεις και έννοιες, καθώς και να μπορεί να κατανοήσει και να αναλύσει εισερχόμενες άγνωστες λέξεις και έννοιες, στη βάση των κανόνων δημιουργίας της γλώσσας. Τόσο για το κομμάτι αυτό όσο και για την χρήση πραγματικά φυσικής γλώσσας, τα πλαίσια της Διπλωματικής εργασίας είναι αρκετά στενά.

1.3 Παραπομπές και URLs αναφορών κεφαλαίου

- **Ολιγοσυνθετική Γλώσσα:**
https://en.wikipedia.org/wiki/Oligosynthetic_language [6]
- **Τεχνητή ή Κατασκευασμένη Γλώσσα:**
https://en.wikipedia.org/?title=Constructed_language [1]
- **Φιλοσοφική Γλώσσα:**
https://en.wikipedia.org/wiki/Philosophical_language [4]
- **Φυσική Σημασιολογική Μεταγλώσσα:**
https://en.wikipedia.org/wiki/Natural_semantic_metalanguage [10]
- **Arpabet:**
<https://en.wikipedia.org/wiki/Arpabet> [12]
- **Dama Diwan:**
http://linguifex.com/wiki/Dama_Diwan [9]
- **Láadan Language:**
<https://en.wikipedia.org/wiki/Láadan> και <http://www.laadanlanguage.org/> [7]
- **ROILA Project:**
<http://roila.org> [11]
- **SADT και IDEFx :**
https://en.wikipedia.org/wiki/Structured_Analysis_and_Design_Technique[18]
- **Toki Pona:**
http://en.wikipedia.org/wiki/Toki_Pona [15]
- **SostiMatiko:**
www.unilang.org/blog/SostiMatiko/ [16]
- Παπακίτσος Ε.Χ. (2013). **Γλωσσική Τεχνολογία Λογισμικού: Ι. Προετοιμασία**. Εκπαιδευτικό Εγχειρίδιο, ISBN: 978-960-93-5636-7. Εθνική Βιβλιοθήκη της Ελλάδος, Αθήνα. [17]

ΑΝΑΛΥΣΗ

Σε αυτό το κεφάλαιο θα παρουσιάσουμε αναλυτικά τα δύο σημαντικά εργαλεία που όπως προαναφέραμε θα χρησιμοποιήσουμε για την επίτευξη του στόχου μας, την Τεχνητή Γλώσσα SostiMatiko (παράγραφοι 2.1.x) και το συστημικό μοντέλο ΟΜΑΣ-III (παράγραφοι 2.2.x).

2.1 SostiMatiko

Όπως ήδη έχουμε αναφέρει, είναι μία κατασκευασμένη γλώσσα με μικρό λεξιλόγιο και γραμματική. Η σύλληψη της ιδέας από τον συγγραφέα¹⁹ ήρθε το 1993. Μια ιδέα που οδήγησε το 2007 στην "Free Greek Language" η οποία χρησιμοποιούσε μία μικρή (ελάχιστη) γραμματική αλλά όλο το Ελληνικό λεξιλόγιο. Από τότε και μέχρι τον Ιούνιο του 2013 διαμορφωνόταν το ελάχιστο δυνατό λεξιλόγιο το οποίο και κατάληξε να έχει μόνο 222 λέξεις. Μέχρι και το τέλος του 2013 γινόντουσαν διάφορες τροποποιήσεις. Κατά τη δημιουργία αυτού του λεξιλογίου, των 222 βασικών λέξεων, προέκυψε η ανάγκη να τυποποιηθεί μια εξαιρετικά απλή γραμματική που να μπορεί να εφαρμοστεί στο μεγαλύτερο μέρος του. Αυτοί οι δύο παράγοντες οδήγησαν στην ολοκλήρωση και δημιουργία μίας νέας γλώσσας στο τέλος του 2013. Η γλώσσα αυτή είναι η "Συστηματική" (Systematic Language) που φέρει το όνομα **SostiMatiko**.

2.1.1 Λεξιλόγιο (Vocabulary)

Σύμφωνα με τα λεγόμενα του συγγραφέα δεν ήταν η Ελληνική καταγωγή του που τον έκανε να εφαρμόσει ένα λεξιλόγιο ελληνικής καταγωγής. Αρχικά έκανε προσπάθειες να χρησιμοποιήσει τα Αγγλικά, Λατινικά, και ιαπωνικά λεξιλόγια αλλά δεν απέδωσαν. Αυτές οι γλώσσες έχουν κάποια πλεονεκτήματα, αλλά δεν θα μπορούσαν με απλότητα, ευελιξία, και λιτότητα να ενσωματώσουν τα διάφορα πρόσθετα που μπόρεσε η Ελληνική. Και όμως παρότι το λεξιλόγιο προκύπτει από Ελληνικές ρίζες ένας απλός ομιλητής της Ελληνικής, δεν είναι εύκολα να το κατανοήσει. Έτσι και ένας Έλληνας θα πρέπει να εκπαιδευτεί πάνω στην γλώσσα αυτή σαν να μην ήταν Έλληνας. Το πλήρες λεξιλόγιο της γλώσσας δίνεται στο Παράρτημα Α'.

2.1.2 Φωνολογία (Phonology and Sandhi²⁰)

Ιδανικά, όλα τα σύμφωνα πρέπει να προφέρονται βάση του Διεθνούς Φωνητικού Αλφαβήτου.

Για τα φωνήεντα, ισχύει ότι και σε οποιαδήποτε από τις πολλές γλώσσες που χρησιμοποιούν πέντε φωνήεντα (a e i o u).

Το "r" μπορεί να προφέρεται ως φωνήεν, αν βρίσκετε μεταξύ δύο συμφώνων.

¹⁹ I.K. Κεσανίδης

²⁰ Sandhi: Η διαδικασία αρθρωτών κατά την οποία η προφορά μιας λέξης ή μορφήματος, αλλάζει όταν ακολουθείται αμέσως από ένα άλλο (ειδικά σε άπταιστη ομιλία)

Σύμφωνα με την μητρική γλώσσα του ομιλητή, όποιες αποκλίσεις στην προφορά οφείλονται σε αυτή, δεν προκαλούν κάποιο πρόβλημα.

Το μόνο γράμμα που χρήζει ιδιαίτερης προσοχής είναι το "x", αν και ο ήχος είναι γνωστός σε πολλές γλώσσες²¹. Σε περίπτωση που κάποιος δεν μπορεί να προφέρει το "x", τότε μπορεί να το αντικαταστήσει με το "h" ή με άλλο παρόμοιο ήχο.

Δεν υπάρχει άηχο γράμμα όπως π.χ. στα Γαλλικά. Όλα τα γράμματα πρέπει να είναι αρκετά ηχηρά.

Δεν υπάρχει ήχος που αντιπροσωπεύεται από περισσότερα από ένα γράμματα. Για το λόγο αυτό το Ελληνικό "th" (θ) έχει ελαττωθεί σε "t".

Γράμματα ασαφή, όπως η "c" και "q", δεν χρησιμοποιούνται.

Μερικές λέξεις (anang-, feng-, eng-, suring-) έχουν ένα "n" πριν το "g". Σε αυτή την περίπτωση, το "n" πρέπει να προφέρεται ως ουρανικό ή ρινικό.

Από τα πιο σημαντικά που πρέπει να διαχειριστούμε σε μία τέτοια γλώσσα είναι η χασμωδία. Η χασμωδία δεν απαγορεύεται παρότι είναι επιθυμητό να αποφευχθεί. Έτσι μπορεί κανείς να προφέρει (αλλά όχι να γράψει) ένα "λαρυγγικό" μετά από "a", "j" μετά από "i", ή ακόμα "e" και "w" μετά από "u" ή και "o".

Είναι επίσης δυνατό κάποιος να προφέρει (αλλά όχι να γράψει) "ejj" στη θέση του "ii" και "owu" στη θέση του "uu", εφόσον τα "ii" και "uu" θεωρούνται δύσκολα.

Το "j" είναι "ένα μισό i" και το "w" ένα "μισό u", είναι δηλαδή ημίφωνα (semivowels) και επιστημονικά ονομάζονται approximants. Το "j" αρθρώνεται όπως το "i" και το "w" όπως το "u", αλλά δεν μπορούν να παραταθούν ή να τονιστούν.

Το "i" δίπλα σε "a", "o" ή "e" μπορεί προφερθεί σαν "j" (αλλά γράφεται πάντα "i") και το "u" δίπλα σε "a", "o" ή "e" μπορεί προφερθεί σαν "w" (αλλά γράφεται πάντα "u").

Η χασμωδία μπορεί επίσης να απαλειφθεί παρεμβάλλοντας μεταξύ των φωνηέντων έναν λαρυγγικό φθόγγο. Ο λαρυγγικός φθόγγος δεν θα πρέπει να εμφανίζεται στη γραφή.

Μερικοί άνθρωποι, εξαιτίας της μητρικής τους γλώσσας, δυσκολεύονται να προφέρουν σύμφωνα που δεν ακολουθούνται από φωνήεντα. Για το λόγο αυτό, οι Έλληνες έχουν την τάση να χρησιμοποιούν "u", μετά από ένα σύμφωνο, οι Ιάπωνες συνήθως προσθέτουν ένα αντίστοιχο ιαπωνικό μικρό "u", οι Τούρκοι χρησιμοποιούν συχνά το δικό τους αντίστοιχο "i", οι Κορεάτες χρησιμοποιούν έναν ήχο όμοιο με το Τουρκικό "i", και άλλοι χρησιμοποιούν το γράμμα του Αζεριανού αλφάβητου "ə" ("schwa")²². Ένα τέτοιο βοηθητικό φωνήεν (για την προφορά συμφώνου που δεν ακολουθείτε από φωνήεν) δεν πρέπει να εμφανίζεται στην αλφαβητική γραφή.

²¹ Ποικιλότητα του x: http://en.wikipedia.org/wiki/Voiceless_velar_fricative

²² <https://en.wikipedia.org/wiki/Schwa>

Τα "ow" και "wo" μπορεί να προφέρονται σαν "aw" και "wa" αντίστοιχα, εάν αυτό προτιμάται για κάποιο ή κάποιους λόγους.

2.1.3 Έμφαση συλλαβής/φωνήεντος (*Syllable / vowel emphasis*)

Έμφαση σε μία συλλαβή δίνεται με επιμήκυνση των φωνηέντων της, με μεγαλύτερο βήμα φωνήεντος, ή τονίζοντάς το. Η έμφαση είναι ένα χαρακτηριστικό του λόγου αλλά εντελώς προαιρετικό εφόσον δεν αλλάζει το νόημα μίας λέξης. Άρα ο χρήστης δεν χρειάζεται να τονίζει συλλαβή ή φωνήεν.

Η έμφαση στην πρώτη συλλαβή της λέξης επισημαίνει την έννοια της ρίζας της λέξης. Το άγχος για μια κατάληξη είναι να επισημανθεί η κατάληξη. Η έμφαση δεν είναι απαραίτητο να αποδίδεται διά της γραφής. Εάν ο χρήστης επιθυμεί, μπορεί να την επισημαίνει μέσου σημάτων προφοράς. Για παράδειγμα á αντί a, é αντί e, ó αντί o κτλ., ή βάζοντας ένα ! (θαυμαστικό) πριν το προς έμφαση φωνήεν. Είναι δυνατό να γράψουμε !a για ένα "a" με υψηλότερο βήμα, !aa για ένα πιο μακρύ "a" και !A για ένα πιο ισχυρό "a".

Επειδή όλα αυτά είναι μόνο για να δώσουμε έμφαση σε ένα «a» και σε άλλα φωνήεντα και επειδή η έννοια παραμένει πραγματικά η ίδια, είναι καλύτερο να αποφεύγεται στη γραφή ή αυτό να γίνεται με φειδώ, μόνο για λόγους στυλ. Ένα! (θαυμαστικό) αμέσως μετά από μια λέξη δίνει έμφαση ολόκληρης τη λέξης αλλά το ίδιο μπορεί να επιτευχθεί με απλή υπογράμμιση ή με κεφαλαιοποίηση όλων των γραμμάτων της

2.1.4 Γραφή (*Writing system*)

Η Γλώσσα θα πρέπει κατά προτίμηση να είναι γραμμένη με το λατινικό αλφάβητο ως πρώτη επιλογή, με το Κυριλλικό αλφάβητο ως δεύτερη και ως τρίτη επιλογή είναι η Ιαπωνική Κατακάνα²³. Ο συγγραφέας έχει δώσει το λεξιλόγιο και στις τρεις αυτές γραφές. Εκτός όμως από αυτές μπορεί να μεταγράφεται και σε όποια άλλη γραφή εκτός την Ελληνική. Ο λόγος είναι ότι το ελληνικό αλφάβητο είναι δεσμευμένο για την FreeGreek γλώσσα των 222 λέξεων, το οποίο και αποτελεί τις ρίζες του λεξιλογίου της SostiMatiko.

Τα Κύρια ονόματα πρέπει να γράφονται όπως στη γλώσσα στην οποία ανήκουν άρα πιθανόν στη μητρική γλώσσα των ανθρώπων που απευθύνονται. Τα Ελληνικά κύρια ονόματα πρέπει να μεταγράφονται με Λατινικούς χαρακτήρες όταν χρησιμοποιούνται για την FreeGreek Γλώσσα ή στη μητρική γλώσσα των ανθρώπων που απευθύνονται.

Μόνο πεζά, ή μόνο κεφαλαία γράμματα πρέπει να χρησιμοποιούνται. Δεν υπάρχει υποχρέωση να ξεκινάει μία πρόταση με κεφαλαίο γράμμα. Τα κύρια ονόματα όμως πρέπει να γράφονται με το πρώτο γράμμα κεφαλαίο και όλα τα υπόλοιπα πεζά (ακόμη και σε κείμενα εξ'ολοκλήρου γραμμένα με κεφαλαία).

2.1.5 Γραμματική (*Grammar*)

Αν και η σειρά των λέξεων μπορεί να ορίζεται από το χρήστη και δεν πρέπει να είναι αυστηρή, ο συγγραφέας υποστηρίζει ότι σε μία μινιμαλιστική γλώσσα η σειρά των λέξεων πρέπει να είναι SVO (υποκείμενο - ρήμα - αντικείμενο) και AN. "AN" σημαίνει ότι οι λέξεις που χαρακτηρίζουν ένα ουσιαστικό (επίθετα ή επιρρήματα)

²³ <https://el.wikipedia.org/wiki/%CE%9A%CE%B1%CF%84%CE%B1%CE%BA%CE%AC%CE%BD%CE%B1>

καθώς και τα συμπληρώματα του, θα πρέπει να προηγούνται (του ουσιαστικού). Σε γενικές γραμμές, όλες οι λέξεις που χαρακτηρίζουν ή συμπληρώνουν μια οποιαδήποτε λέξη, συμπεριλαμβανομένου και των αναφορικών προτάσεων ουσιαστικού, πρέπει να προηγούνται της κύριας λέξης. Ωστόσο, το οριστικό άρθρο/αντωνυμία "ΤΟ", καθώς και το δεικτικό "ΑΥΤΟ", μπορούν να χρησιμοποιηθούν ως αναφορικές αντωνυμίες ώστε να εισάγουν μία αναφορική πρόταση μετά το ρήμα.

Ο λόγος για τον οποίο πρέπει να ακολουθούμε τη δομή SVO και AN είναι ότι σε άλλη περίπτωση η γλώσσα δεν θα είναι μινιμαλιστική.

Η SVO δομή προτείνεται μεν για τη χρήση σε αυτή τη γλώσσα, ωστόσο απαιτεί κάποια ένδειξη για τη διάκριση του υποκειμένου από το αντικείμενο.

Κάθε συντακτικό γλώσσας βασίζεται στην έννοια "το πρώτο είναι το δεύτερο", ή "το πρώτο έχει το δεύτερο", δηλαδή η δεύτερη λέξη είναι κτήμα της πρώτης. Επομένως, όταν λέμε "έργο εύκολο", θα πρέπει να σημαίνει "το έργο είναι εύκολο", έτσι χρησιμοποιούμε την ελάχιστη σημασία, χωρίς την ανάγκη των συζεύξεων ή άρθρων. Αν λέμε "εύκολο έργο", βάση της ίδιας αρχής θα έπρεπε να σημαίνει "αυτό το εύκολο πράγμα είναι ένα έργο", αλλά τώρα αυτή η πληροφορία δεν είναι απαραίτητη, έτσι καταλαβαίνουμε "αυτό το εύκολο πράγμα το οποίο είναι ένα έργο" και πιο απλά "ένα εύκολο έργο". Ομοίως, "λουλούδι λευκό" σημαίνει "το λουλούδι είναι/ήταν λευκό", αλλά "λευκό λουλούδι" σημαίνει "το λουλούδι που είναι λευκό". Με άλλα λόγια, το συμπλήρωμα πριν από το ουσιαστικό λειτουργεί ως αναφορική πρόταση, ενώ το ίδιο μετά το ουσιαστικό λειτουργεί ως κατηγορήμα, χωρίς καμία ανάγκη για οποιαδήποτε άλλη ένδειξη. Αλλά αν η φυσική αρχή "το πρώτο είναι το δεύτερο" δεν έχει καθοριστεί, χρειαζόμαστε επιπλέον δείκτες, όπως συνδετικό ρήμα, άρθρο και δείκτη αναφορικής πρότασης π.χ. το σύγχρονο ελληνικό "ΠΟΥ". Οι επιπλέον δείκτες όμως κάνουν τη γλώσσα μη μινιμαλιστική, ενώ, υπενθυμίζεται και πάλι, ότι η αρχή "το πρώτο είναι το δεύτερο" είναι φυσικό, όταν αυτό ακολουθεί τη φυσιολογική διαδικασία της γνωστικής λειτουργίας.

Συμπληρώματα του ουσιαστικού, όπως γενική πτώση, επίθετο και αναφορικές προτάσεις, περιγράφουν μία εικόνα, στην οποία το ουσιαστικό είναι αυτό που φαίνεται ή εστιάζεται. Αυτός είναι πραγματικά ο πιο συνηθισμένος τρόπος της γνωστικής λειτουργίας: από το γενικό στο ειδικό. Αυτό είναι επίσης και το πιο πρακτικό.

Με μερικά παραδείγματα θα γίνει πιο κατανοητό.

Στα κινεζικά, μια φυσική γλώσσα που χρησιμοποιεί ένα ελαχιστοποιημένο συντακτικό και γραμματική, λέει: "**mind correct not fear mistake**" (Σκέψη σωστή όχι φόβος λάθους), εννοώντας "if the mind is correct, there is no fear of making a mistake / one is not afraid of making a mistake" (Αν η σκέψη είναι σωστή, δεν υπάρχει κανένας φόβος να γίνει λάθος / κάποιος δεν φοβάται να κάνει λάθος). Αν χρησιμοποιήσουμε οποιαδήποτε άλλη δομή εκτός από SVO και AN, αυτό δεν μπορεί να εκφραστεί σε τόσο λίγες λέξεις και τόσο απλή γραμματική, θα χρειαζούσαμε επιπλέον δείκτες.

Τα Αγγλικά, παρόλο που δεν χρησιμοποιούν τόσο ελαχιστοποιημένη γραμματική, θεωρούνται διεθνώς μια γλώσσα με πολύ απλή γραμματική και αρκετά πρακτική, ακριβώς επειδή βασίζεται στη δομή SVO και AN.

Για παράδειγμα, σκεφτείτε τη φράση: "**a rolling stone catches no moss**". Με ελαχιστοποιημένη γραμματική αυτή θα ήταν "roll stone catch no moss". Όμως παρότι οι δύο φράσεις είναι πολύ κοντά, η δεύτερη είναι ανεπαρκής, διότι δεν έχει καθοριστεί ότι ακόμη και ένα ρήμα πριν από ένα ουσιαστικό είναι επίθετο -επίσης στην αγγλική γλώσσα, δεν είναι συχνά σαφές ποιο είναι το ουσιαστικό και ποιο είναι

το ρήμα, γι 'αυτό χρειαζόμαστε επιθήματα όπως "ing" - και το κυριότερο, είναι επίσης αναγκαίο να γίνει διάκριση μεταξύ του απαιτούμενου "catch" (πιάνω - πιάσει) και του πραγματικού "catches" (αλιεύω, συλλέγω, μαζεύω ... - αλιεύει).

Από την αρχή της κατασκευής της FreeGreek Γλώσσας (2007), ως πιο απαραίτητο χαρακτηριστικό της γραμματικής, έχει κριθεί ότι είναι η διάκριση μεταξύ της ενέργειας που έγινε και της ενέργειας που πρέπει να γίνει (DONE, TO BE DONE). Οτιδήποτε άλλο, όπως πληθυντικός αριθμός, ενεργητική και παθητική φωνή, πτώση, φύλλο κλπ, μπορούν να εκφράζονται χωρίς οποιαδήποτε χρήση κλίσης. Στη σύγχρονη Ελληνική ένα απλό άτονο "NA" πριν από το ρήμα είναι αυτό που δίνει το διαχωρισμό μεταξύ μίας ενέργειας που έγινε (done) και μίας ενέργειας που θα γίνει ή που πρέπει να γίνει (to be done). Ως εκ τούτου, όλα τα συζευκτικά, οι κλίσεις, οι πτώσεις, τα άρθρα, τα γένη κτλ μπορούν να απορρίπτονται στην FreeGreek. Το "NA" μπορεί να θεωρηθεί επαρκές για την προαναφερθείσα διάκριση όπως ακριβώς το "ut" στη Latino Sine Flexione²⁴ (Λατινικά χωρίς κλίσεις).

Ακόμα, η επισήμανση της προστακτικής είναι τόσο σημαντική στην πραγματική ζωή, που κρίθηκε απαραίτητο να βρεθεί λύση. Έτσι επινοήθηκε η παράληψη του τελικού φωνήεντος του ρήματος ώστε να επισημαίνεται η προστακτική φωνή ή η ενέργεια που απαιτείται. Το παραλειπόμενο τελικό φωνήεν "i" του ρήματος έχει επίσης καταστεί σημαντικό χαρακτηριστικό και της SostiMatiko, για την προβολή της απαιτούμενης ενέργειας.

2.1.6 Περισσότερα για τη σειρά των λέξεων

Η σχέση μεταξύ δύο λέξεων/όρων που συνθέτονται σε μία πρόταση, είναι η εξής:

1	2
γενικότερη	ειδικότερη
συμπληρωματική	κύρια
πιο αντικειμενική	πιο υποκειμενική
πιο περιγραφική	πιο επικριτική

Όταν ακολουθούμε αυτή τη σειρά (πρώτα 1 και μετά 2), προχωρούμε από το γενικό στο ειδικό, από το πιο αντικειμενικό στο πιο υποκειμενικό, από περιγραφικό σε πιο επικριτικό, και από τον συμπληρωματικό στον κύριο όρο. Όλα αυτά είναι η φυσική διαδικασία ώστε να μαθαίνουμε πράγματα. Για παράδειγμα έχουμε το "λευκό λουλούδι" (white flower). Το "Λευκό" είναι μία αίσθηση, ενώ το "Λουλούδι" είναι ο χαρακτηρισμός του αντικειμένου των αισθήσεων. Ως εκ τούτου, "λευκό" είναι το αντικείμενο καθώς το "λουλούδι" δηλώνεται ως υποκείμενο. Το "Λευκό" είναι ένας πολύ γενικός όρος, ενώ το "λουλούδι" είναι πολύ πιο συγκεκριμένος. Όταν ο κύριος όρος ακολουθεί, έχουμε την αίσθηση ότι η δήλωση είναι πλήρης και τελική (ότι έπρεπε να προστεθεί, έχει ήδη προστεθεί στο κύριο όρο). Οι Τούρκοι το εκφράζουν αυτό με το γνωμικό "Allah bir, söz bir" (Ο Θεός είναι ένας, η λέξη είναι μία - η λέξη που ειπώθηκε δεν πρόκειται να αλλάξει). Έχουμε, επίσης, την αίσθηση της αντικειμενικότητας, επειδή η πληροφορίες του υποκειμένου έρχονται μετά τα δεδομένα του αντικειμένου. Η ροή της σκέψης είναι σαφής και φυσιολογική, πρώτα έρχεται το πιο γενικό και μετά το πιο ειδικό το οποίο και εξαρτάτε από τον προηγούμενο όρο. Υπάρχουν βέβαια γλώσσες, όπως τα Αραβικά ή η Tagalog²⁵, που ακλουθούν τη αντίστροφη σειρά (πρώτα 2 και μετά 1). Θεωρητικά, η πληροφορία

²⁴ https://en.wikipedia.org/wiki/Latino_sine_flexione

²⁵ <https://en.wikipedia.org/wiki/Tagalog>

μπορεί να είναι η ίδια όπως και πριν, αλλά στην πράξη, το στυλ αυτών των γλωσσών κινείται μεταξύ της σύγχυσης και της αλαζονείας, επειδή πρώτα ακούμε αυτό που ο ομιλητής θεωρεί σημαντικό, και στη συνέχεια την εξήγηση γιατί το θεωρεί σημαντικό. Αυτό ταιριάζει με το στυλ του Κορανίου, όπου ο Θεός μιλάει και δεν είναι υποχρεωμένος να εξηγήσει πολλά, ενώ εμείς πρέπει να υποθέσουμε ότι όλα αυτά είναι ακριβώς ο λόγος του Θεού, χωρίς την παραμικρή ανθρώπινη παρέμβαση. Σε αντίστοιχο ύφος είναι και η Παλαιά Διαθήκη, όπου δεν είναι μια γλώσσα που έγινε για να εξηγηθούν τα πράγματα, αλλά μόνο για να γίνει αποδεκτή ως ο λόγος του Θεού. Έτσι δεν μπορούμε να ρωτάμε γιατί. Αν ρωτήσουμε "γιατί είναι το κοτόπουλο καθαρό και το κουνέλι βρώμικο" ή "γιατί πρέπει να μετράμε την ημέρα αυτή ως έβδομη και όχι ως έκτη", εφόσον η γραφή στηρίζεται από την παράδοση, το ερώτημα είναι βλάσφημο. Σε γλώσσες που η σειρά είναι πρώτα το 1 και μετά το 2, το ύφος είναι αρκετά διαφορετικό. Για να το καταλάβουμε αυτό, θα πρέπει να μελετήσουμε κείμενα Ιαπωνικά, παλιά Τουρκικά, ή Ταμίλ, για παράδειγμα. Είναι το ύφος της τέχνης της πειθούς, ένα αντικειμενικό ύφος. Εκτός από το στυλ, μπορείς να ξέρεις αν ένας Τούρκος ή ένας Ιάπωνας λέει ψέματα; Θεωρητικά, φυσικά και είναι δυνατόν να ειπωθεί ένα ψέμα σε Τουρκικά ή στα Ιαπωνικά ή Κορεατικά κλπ, αλλά δεν υπάρχει χαρακτηριστικό παράδειγμα να αναφερθεί για να δείξουν ότι είναι δυνατόν σε μία τέτοια γλώσσα με τέλεια σειρά λέξεων (1 - 2). Αυτό που η ίδια γλώσσα αναγκάζει να γίνει και μπορεί να βεβαιωθεί είναι ότι οι ιθαγενείς (μιας τέτοιας γλώσσας) θα τη χρησιμοποιήσουν για τη διπλωματία, αλλά όχι για το ψέμα. Αυτό (κατά τη γνώμη του συγγραφέα), φαίνεται να είναι και το βαθύτερο νόημα της Οθωμανικής παροιμίας: "Türkçe bilmez, Allahtan Korkmaz" ("a person who knows not the Turkish language, fears not God" ή "ένα άτομο που δεν γνωρίζει την τουρκική γλώσσα, δεν φοβάται το Θεό").

Έτσι και για τη SostiMatiko προτείνεται έντονα η συγκεκριμένη σειρά λέξεων (1 - 2), με την εξαίρεση του αντικείμενου το οποίο μπορεί να ακολουθήσει το ρήμα (χάριν της ελάχιστης σύνταξης), αλλά ακόμη και αυτό (με -n (ο) επίθημα) μπορεί να προηγείται του ρήματος. Όμως, στην Κινεζική σύνταξη, αν τοποθετηθεί το αντικείμενο μετά το ρήμα, υπάρχει απόλυτη ταύτιση με την εν λόγο σειρά λέξεων (1-2), επειδή (στην Κινεζική) το ρήμα είναι στην πραγματικότητα ένα ουσιαστικό. Για παράδειγμα "the emperor sacrifices pork and rice" στα Κινέζικα είναι κυριολεκτικά "the emperor's sacrifice (is) pork and rice", επίσης στη φράση "i prefer that", στα Κινέζικα μεταφράζεται "my preference (is) that", και ούτω καθεξής. Είναι φυσικά δυνατόν να μιμηθούμε την κινεζική σύνταξη χρησιμοποιώντας το λεξιλόγιο της SostiMatiko, έτσι ώστε να χρησιμοποιηθούν όλες οι λέξεις ως ουσιαστικά, και ως εκ τούτου χωρίς τη χρήση καταλήξεων, ή με τη χρήση μόνο της κατάληξης -ο, για να διευκολυνθεί η προφορά. Ωστόσο, είναι κρίμα να την περιορίσουμε τόσο πολύ, αφού οι δυνατότητες της γλώσσας χρησιμοποιώντας το σύνολο των καταλήξεων που της δίνονται, υπερβαίνουν κατά πολύ το σημείο αυτό. Μπορεί να είναι μια γλώσσα με μικρό λεξιλόγιο και με πέντε (5) βασικές καταλήξεις, άρα ουσιαστικά και μια μικρή γραμματική, συνδυάζοντάς τα όλα όμως, **η γλώσσα γίνεται μεγίστων δυνατοτήτων.**

2.1.7 Δείκτης υποκειμένου (Subject indicator)

Στη συστηματική γλώσσα, συνήθως δεν χρειάζεται κάποια ένδειξη του υποκειμένου. Η σειρά των λέξεων, τα σημεία στίξης και ο δείκτης αντικείμενου "no", δείχνουν ποιά από τα ουσιαστικά είναι το αντικείμενο της πρότασης. Παρόλα αυτά, σε περίπτωση που αισθανόμαστε ότι πρέπει να σηματοδοτηθεί το υποκείμενο, π.χ. όταν ένα άλλο ουσιαστικό το ακολουθεί, ή όταν αυτό (υποκείμενο) έρχεται μετά από το ρήμα, ο πιο

πρακτικός δείκτης είναι ένα "ti" να το ακολουθεί. Το "ti" αυτό σημαίνει "κάνει κάτι" - "does that". Ας δώσουμε ένα παράδειγμα για να το αποσαφηνίσουμε: "meḡo Erpo ti arnitoṇ uka fagi" = "ο κροκόδειλος δεν τρώει το πουλί", ή "ugiro arnito no fagi meḡo erpo ti" = "ένας κροκόδειλος έφαγε το πουλί του νερού".

2.1.8 Λέξεις που δείχνουν γραμματικές σχέσεις (Grammatical words)

Πολλές λέξεις της SostiMatiko λειτουργούν ως προθέσεις, σύνδεσμοι, άρθρα και γραμματικά μόρια γενικά. Τα μόρια συνήθως αντιμετωπίζονται ως επιρρήματα και θα πρέπει να έχουν την κατάληξη -a. Αλλά, ένα μόριο χωρίς την κατάληξη -a μπορεί να θεωρηθεί ως προστακτικό για τον αποδέκτη ή για τον εαυτό του, και γι 'αυτό μπορεί να μην φέρει επίθημα, δηλαδή να είναι η κύρια λέξη χωρίς καμία κατάληξη. Με μερικά παραδείγματα αυτό θα γίνει πιο ξεκάθαρο. "kas" είναι η κύρια λέξη που σημαίνει ένωση. Για το "και" χρησιμοποιούμε τη λέξη "kasa" (επίρρημα, κατάληξη -a). όμως μπορούμε να το δούμε και ως προστακτικό, ώστε απλά να πούμε "kas" (προστακτικό ρήμα, χωρίς κατάληξη) που σημαίνει "πρόσθεσε / ένωσε" (στην Αγγλική: " add/join", ή "let me add/join") και αυτό είναι σχεδόν το ίδιο με το "kasa". Ομοίως, το Αγγλικό "of" είναι "apa" ("starting from"/"ξεκινώντας από", επιρρηματικό -a), αλλά μπορεί επίσης να είναι "ap" ("start from"/"εκκίνηση από", προστακτικό ρήμα χωρίς κατάληξη). Ο δείκτης αντικειμένου είναι το "n", έτσι θα μπορούσε να είναι "na" (επιρρηματικά), αλλά απλώς "n" σημαίνει "θεώρησέ το αντικείμενο του ρήματος", οπότε το απλό "n" δουλεύει μια χαρά ως δείκτης αντικειμένου. Το ίδιο μπορεί να ειπωθεί και για τις περισσότερες από τις γραμματικές λέξεις που αν και ιδανικά πρέπει να φέρουν το επίθημα -a, μπορούν να δουλέψουν το ίδιο καλά και ελλείψη του. Να προσθέσουμε ακόμα κάτι σχετικά με τον δείκτη αντικειμένου. Όταν έχουμε ένα αντικείμενο σε αναφορική πρόταση που προσδιορίζει ένα ουσιαστικό, π.χ. "the hen that layed the golden eggs" (η κότα που γέννησε τα χρυσά αυγά), είναι καλύτερα να βάλουμε αυτό το αντικείμενο πριν από το ρήμα, όπως "the golden eggs laying hen" (τα χρυσά αυγά - φωτόκου - κότας/όρνιθας), έτσι ώστε να μην παρεμβάλλεται κάτι μεταξύ του ουσιαστικού (θέμα) και του ρήματος. Αυτό φαίνεται στο παράδειγμα: "kitrino airo awgosoio arnito ten exeti tewoa" που στα Αγγλικά σημαίνει "the golden eggs laying hen will belong to you" και στα Ελληνικά "Τα χρυσά αυγά κότας θα ανήκουν σε εσένα".

2.1.9 Αριθμοί (Numbers)

Για να εκφραστούν όλοι οι αριθμοί χρησιμοποιούνται μόνο οι παρακάτω λέξεις:

edno =1 (ένα),

duo =2 (δύο),

xero =hand or 5 (χέρι / παλάμη ή πέντε),

dexo =10 (δέκα).

Αυτές οι λέξεις, αριθμοί, αρκούν για να εκφραστούν όλοι οι αριθμοί στην καθημερινότητα μας. Για να πεις τρία (3) ή θα χρησιμοποιήσεις έναν πρακτικό τρόπο που είναι το "to to to" (αυτό, αυτό και αυτό) ή "duo edno" (2 1 = 2+1). Το "du" εκφράζει το διπλασιασμό (du = 2x), το "xer" τον πολλαπλασιασμό με το πέντε (xer = 5x) και το dex τον πολλαπλασιασμό με το δέκα (dex = 10x). Για παράδειγμα αν θέλουμε να πούμε 10x10 λέμε "dex dexo", ενώ για 10x100 λέμε 10x10x10 δηλαδή "dex dex dexo". Σε ένα άλλο παράδειγμα για να εκφράσουμε το 1821 μπορούμε να ακολουθήσουμε δύο δρόμους. Μπορούμε να πούμε: 10x10x10 και 2x2x2x10x10 και 2x10 και 1 (τα "και" παραλείπονται) δηλαδή "dex dex dexo (kas) du du du dex dexo

du dexo edno", ή να πούμε: 1 8 2 1 (ένα οκτώ δύο ένα) δηλαδή edno (1), du du duo (2x2x2 = 8), duo (2), edno (1). για ένα κλάσμα x/y, θα λέγαμε από τα "y" μέρη (τμήματα), τα "x". Έτσι για το 22/7 θα πούμε από τα επτά μέρη τα είκοσι δύο δηλαδή "xero duo mera (μέρη / parts) du dexo duo".

2.1.10 Συνένωση καταλήξεων (*Concatenation of suffixes*)

Η συνένωση των καταλήξεων περιορίζεται μόνο από τη δημιουργικότητα του ομιλητή και την κατανόηση του ακροατή. Κάθε πρόσθεση επιθήματος αναφέρεται στη λέξη που σχηματίζεται πριν από αυτό. Π.χ. xresi = (αυτή/αυτός) χρησιμοποιεί, xresia = χρησιμοποιώντας/(δια) μέσω, xreso = χρήση/χρησιμοποίηση/μεταχείριση xresoa = ανάλογα με τον τρόπο χρησιμοποίησης, και ούτω καθεξής. Μπορούμε, επίσης, να διπλασιάσουμε την κατάληξη ρήματος -i όπως xresii και τότε θα σημαίνει «να κάνει χρήση κάποιου», δηλαδή ένα αιτιολογικό ρήμα, όπως επίσης xresieti = (αυτή/αυτός) είναι φτιαγμένοι/αναγκάζονται να χρησιμοποιούν, xresetii = (αυτή/αυτός) αναγκάζει (αυτόν / αυτή κτλ) να χρησιμοποιηθεί ως. Το "engo" = Εγώ, οπότε το "engoo" = "δικό μου" (engo-o δηλαδή ένα πράγμα ή πρόσωπο που ανήκει σε εμένα).

2.1.11 Δημιουργώντας ρήματα και επιρρήματα από ουσιαστικά (*Making verbs and adverbs from nouns*)

Στη συστηματική Γλώσσα, κάθε βασική λέξη θεωρείται ένα ουσιαστικό. Από κάθε ουσιαστικό, δημιουργούμε επίρρημα, ρήμα, και άλλες λέξεις που παράγονται ποικιλοτρόπως με τη σύνδεση των διαθέσιμων επιθημάτων. Για να δημιουργήσουμε ένα επίρρημα από ένα ουσιαστικό, είναι εύκολο. Πρέπει να σκεφτούμε αυτό το πράγμα στη λειτουργία του, ή σαν ένα, ή απλά να προσθέσουμε μία γνωστή επιρρηματική κατάληξη. Για παράδειγμα το ουσιαστικό "kako" που έχει κυριολεκτική έννοια στην Ελληνική γίνεται σαν επίρρημα "kaka" επίσης με κυριολεκτική έννοια. Επίσης η λέξη "ewo" που σημαίνει καλή, γίνεται σαν επίρρημα "ewa" που σημαίνει καλά. Ακόμη και ένα άγνωστο επίρρημα μπορεί εύκολα να γίνει κατανοητό, π.χ. "agoro" = αγορά/μαγαζί/κατάστημα, άρα, "agora" = σύμφωνα με την αγορά (κατάστημα) και σύμφωνα με το τι συνηθίζεται εκεί, εννοεί τον επιχειρηματικό τρόπο, το εμπόριο και ούτω καθεξής. Ωστόσο, για να δημιουργηθεί ένα ρήμα από ένα ουσιαστικό δεν φαίνεται τόσο απλό, επειδή ένα ρήμα μπορεί να προέλθει με διάφορους τρόπους, όπως τονίζει ο συγγραφέας και επιχειρεί να περιγράψει διαδικτυακά²⁶.

Στην πραγματικότητα, αν και φαίνεται απελπιστικά περίπλοκο, μπορεί να είναι μια εντελώς απλοποιημένη διαδικασία, αν λάβουμε υπόψη μας κάποιο πιο πρακτικό τρόπο. Αν το ουσιαστικό είναι "x", το προερχόμενο ρήμα είναι "να προκαλέσει x". Το "να προκαλέσει x" βασικά σημαίνει "να προκαλέσει (κάτι) να είναι x" το οποίο μπορεί να σημαίνει "να χρησιμοποιεί (ως) x", επειδή χρησιμοποιώντας το x, έχει κατά κάποιο τρόπο ως αποτέλεσμα αυτό να είναι x. Για παράδειγμα, χρησιμοποιώντας έναν δρόμο καθιστάτε (υφίσταται) ένας δρόμος, ενώ αν κανείς δε χρησιμοποιεί ένα δρόμο δεν υφίσταται ένας δρόμος. Όταν το ενεργό ρήμα είναι κατανοητό, τότε το παθητικό ρήμα ακολουθεί απλά και φυσικά. Για παράδειγμα, το ουσιαστικό "omo" σημαίνει όμοιο πράγμα, το ίδιο με κάτι άλλο, κατά κάποιο τρόπο. Στη συνέχεια το "omi", το ρήμα, σημαίνει "προκαλεί κάτι να είναι όμοιο", να το "χρησιμοποιήσει ως όμοιο", ή "να χρησιμοποιήσει μία παρομοίωση" δηλαδή να παρομοιάζεται κάτι ως

²⁶ http://www.academia.edu/4226602/making_verbs_from_nouns.

κάτι άλλο. Έτσι το παθητικό ρήμα "ometi" προφανώς σημαίνει "να παρομοιαστεί", ή "να χρησιμοποιηθεί ως παράδειγμα", ή "να γίνει παρόμοιο".

2.1.12 Συντομεύσεις (Shortcuts):

Η γλώσσα, όπως περιγράφεται παραπάνω μπορεί να λειτουργήσει θαυμάσια καλά. Αν η Toki Pona με 124 λέξεις (συμπεριλαμβανομένων των γραμματικών προθεμάτων) υποτίθεται ότι είναι σε θέση να εκφράσει σχεδόν τα πάντα, τότε είναι ακόμα πιο εύκολο γι' αυτή τη Συστηματική Γλώσσα που έχει ευρύτερο λεξιλόγιο και γραμματικές έννοιες. Ωστόσο, για να γίνει ακόμα πιο πρακτική, μια σειρά από επισυνάψεις έχουν κριθεί χρήσιμες. Οι επισυνάψεις αυτές δεν είναι απαραίτητες εφόσον μπορούν να καλυφτούν από άλλες έννοιες που προβλέπονται για τη συστηματική Γλώσσα. Όμως μπορούν και εξακολουθούν να είναι χρήσιμα ως «συντομεύσεις». Για μια γλώσσα που αξιώνει να χρησιμοποιηθεί ως διεθνές εργαλείο η βραχύτητα και συνοπτικότητα θεωρούνται πολύ σημαντικά. Για να μπορούμε λοιπόν να έχουμε μία πολύ πρακτική και ευέλικτη γλώσσα και προαιρετικά, μπορούμε να χρησιμοποιήσουμε τις παρακάτω επισυνάψεις:

2.1.12.1 Επιθήματα συντόμευσης (Shortcut suffixes)

-in(o) είναι το επίθημα. Αν πάρουμε τη ρίζα λέξης αντικειμένου που έχει ένα χαρακτηριστικό χρώμα και προσθέσουμε το επίθημα αυτό τότε έχουμε το χρώμα. Έτσι τα βασικά χρώματα είναι:

awgo (αυγό) → awgino (το χρώμα του αυγού)

foto (φως) → fotino (το χρώμα του φωτός, άσπρο)

saxo (αίμα) → saxino (κοκκίνο)

kitro (κίτρο / λεμόνι) → kitrino (κίτρινο)

atrako (άνθρακας) → atrakino (ανθρακί / μαύρο)

skoto (σκοτάδι) → skotino (σκούρο)

kuano (κυανό) → kuanino (μπλε)

urano (ουρανός) → uranino (ανοιχτό μπλε/γαλάζιο)

praso(πράσο / κρεμμύδι / σκόρδο και συναφή) → prasino (πράσινο) ή

futo (φυτό) → futino (πράσινο)

airo (σίδηρο) → airino (μεταλλικό / λαμπερό / γυαλιστερό)

geo (γη) → geino (καφέ)

pnowo (πνοή / αέρας / άνεμος) → pnowino (διάφανο) ή

uko (ουδέν / κενό) → ukino (διάφανο).

-ak(o) είναι υποκοριστικό, πχ oiko = σπίτι άρα oikako = oik-ak-o = μικρό σπίτι (σπιτάκι) και είναι συντόμευση για το "mikro oiko".

-ar(o) είναι διεύρυνση, πχ woxo = όχημα άρα woxaro = wox-ar-o = μεγάλο όχημα και είναι συντόμευση για το "mego woxo".

-id(o) σημαίνει κάτι παρόμοιο, πχ antropido = antrop-id-o = παρόμοιο με ανθρώπινο.

-ik(o) σημαίνει κάτι που σχετίζεται / αφορά, πχ futiko = fut-ik-o = φυτικό

-itik(o) ανοίκει σε, πχ agriitiko = agri-itik-o = κάτι που ανήκει στην άγρια φύση και aderfitiko = κάτι που ανήκει σε κάποιον αδερφό.

-ux(o) έχει ή περιέχει, πχ sakaruxo = sakar-ux-o = ζαχαρούχο.

-es(i) σημαίνει θέλει ή επιθυμεί, πχ ugiresi = ugir-es-i = θέλει νερό / είναι διψασμένο. Πιο ειδικά: geo ugiresi = η γη χρειάζεται νερό / διψασμένη γη.

ewi = να κάνει κάτι καλό, άρα ewiesi = ewi-es-i επιθυμεί να κάνει κάτι καλό
erxeti = έρχεται / φτάνει, άρα erxeteso = erx-et-es-o = επιθυμεί να επισκεφθεί.

-ter(o) σημαίνει περισσότερο και είναι ιδιαίτερα χρήσιμο για επίθετα ή επιρρηματικές λέξεις, πχ makra = μακριά, makratero = makr-a-ter-o = κάτι που βρίσκεται ακόμα μακρύτερα.

-an(i) σημαίνει, ικανότητα να, πχ grafi = φτιάχνει μια ζωγραφιά / σχέδιο, grafani = graf-an-i = μπορεί να φτιάξει ένα αντιπροσωπευτικό σχέδιο / γράφημα και graf-an-o = η ικανότητα να ζωγραφίζει / σχεδιάζει.

orti = στέκεται όρθιο, ortano = ort-an-o = η ικανότητα να σταθεί όρθιο, ή να αποκαταστήσει κάτι.

-tr(o) είναι η κατάληξη για όλα τα είδη των εργαλείων ή μηχανημάτων. Αποτελεί την συντόμευση για "makeno". Πχ "ugirtro" ή "ugiritro" είναι προφανώς μια βρύση (στρόφιγγα) ή λάστιχο νερού ή χειροσυσκευή ντους, ανάλογα με τα συμφραζόμενα. "anitro" είναι ένα ασανσέρ. "purtro" ή "puritro" είναι αναπτήρας ή σπύρο. "fottro" ή "fotitro" είναι μια λάμπα ή δέσμη φωτός. "pnowtro" μπορεί να είναι ένας ανεμιστήρας, ή μπορεί να είναι "pnowet(i)tro (pnow-et-i-tro) ", δηλαδή "κάτι που χρησιμοποιείται για να φυσήσει αέρας επάνω του" ή ένας φορητός ανεμιστήρας και ούτω καθεξής.

-nes(o) είναι η θηλυκή κατάληξη και λειτουργεί ως συντόμευση για 'guno', πχ, bowo = βόδι, bowneso = αγελάδα. Το "to" είναι το οριστικό άρθρο, αλλά με το επίθημα ως "tneso" ή "toneso", είναι το θηλυκό οριστικό άρθρο ή "αυτή" ή "εκείνη που...".

-wis(o) είναι η αρσενική κατάληξη που αποτελεί συντόμευση για "andro", πχ "ippro" είναι άλογο και "iprwiso" επιβήτορας (αρσενικό άλογο). Αντίστοιχα το αρσενικό οριστικό άρθρο ή η αρσενική προσωπική (ή σχετική) αντωνυμία είναι "twiso" ή "towiso".

2.1.12.2 Προθήματα συντόμευσης (Shortcut prefixes)

kse- είναι το πρόθημα της επιστροφής ή ακύρωσης, πχ "perii" είναι να κυκλώσει ή περιβάλει ή τυλίξει, έτσι "kseperii" = kse-peri-i = αυτή / αυτός ξετυλίγει. Άλλο παράδειγμα είναι η λέξη "Pairi" = αυτή / αυτός παίρνει, άρα "ksepairi" = αυτή / αυτός δίνει πίσω ότι είχε λάβει, επιστρέφει κάτι.

u(o)- αρνητικό πρόθημα ή πρόθημα ανατροπής. Έτσι "pairi" = αυτή / αυτός παίρνει και "urairi" = u-pair-i = αρνείται να πάρει. Άλλο παράδειγμα, "neki" = σκοτώνει, έτσι "unekrio" = u-nekr-fo = κάποιος που δεν σκοτώνει και "unekria" = όχι μέσω της θανάτωσης. "Tambo" = θαμπό / θολό, έτσι "utambo" = u-Tamb-o = ρητό / κατηγορηματικό και "utambi" = να καταστήσει σαφές κ.λπ.

um(e)- σημαίνει ότι δεν έχει, δεν περιέχει, δηλαδή το αντίθετο από ότι με την κατάληξη -ux, πχ "umpraso" = um-pras-o =(τρόφιμα) που δεν περιέχουν κρεμμύδι ή κάτι παρόμοιο / κοντινό σε αυτό. "umarmio" = ανάλατο.

ksana- σημαίνει "ξανά", λειτουργεί ως υποκατάστατο πχ των "nea" / "dua" / "kasa" , για παράδειγμα αντικαθιστώντας π.χ. Νέα / Dua / Κάσα? για παράδειγμα, "ksanaameri" = Ksana-Ameri = ο ήλιος λάμπει και πάλι, ο ήλιος φέρνει μια καινούργια μέρα, ξαναγίνεται μέρα.

2.1.12.3 Ενθήματα συντόμευσης (Shortcut infixes)

Τα ενθήματα είναι για να μπαίνουν εντός της κύριας λέξης σύμφωνα με τους παρακάτω κανόνες:

Αμέσως μετά το πρώτο φωνήεν της λέξης, αν η λέξη αυτή δεν έχει σύμπλεγμα συμφώνων ή διαδοχικά φωνήεντα.

Αν η λέξη έχει ένα σύμπλεγμα συμφώνων (διαφορετικό από τον τύπο ρινικό + άλλο) τότε μπαίνει μεταξύ των δύο συμφώνων, πχ "atranko" = άνθρακας, at[wa]raki= "φτιάχνεις άνθρακα" (ή κάρβουνα για ψήσιμο).

Αν η λέξη έχει δύο συνεχόμενα φωνήεντα τότε μπαίνει μεταξύ αυτών, πχ "rairi" = παίρνει, ra[jo]iri =εγώ παίρνω.

Όλα αυτά τα ενθήματα είναι για να φανεί το πρόσωπο της λέξης, δηλαδή το υποκείμενο του ρήματος, ή το κατηγορημα:

-(j)o- είναι "εγώ" (ή "εμείς" εάν η λέξη έχει το στοιχείο του πληθυντικού αριθμού), πχ "ajoisteti" (a[jo]ist-eti) = αισθάνομαι.

-(w)a- είναι "εσύ" (ή "εσείς" εάν η λέξη έχει το στοιχείο του πληθυντικού αριθμού), πχ "eraxetso" (er[wa]x-et-so) = (υπάρχει πληθυντικός -so) πρέπει να έρθετε .

-(j)e- είναι αυτός, αυτή, αυτό, πχ "makjera" (mak[je]r-a) = μακριά (με) αυτό το πρόσωπο.

Αυτά τα ενθήματα χάνουν τα ημίφωνα της αρχής τους όταν ακολουθούν ένα άλλο ένθημα.

Είναι δυνατόν να συνδυαστούν δύο ενθήματα, ως εξής:

-(j)o(w)a- = εγώ και εσύ

-(j)o(j)e- = εγώ και αυτός / αυτή

-(w)a(j)o- = εσύ και εγώ

-(w)a(j)e- = εσύ και αυτός / αυτή

-(j)e(j)o- = αυτός / αυτή και εγώ

-(j)e(w)a- = αυτός / αυτή και εσύ

Παραδείγματα.

rajoaxo = εγώ και εσύ παχαίνουμε (raxo).

saokotino = εσύ και εγώ, αποκτούμε σκούρα χρια (skoto).

rajeaizi = αυτός και εσύ παίζεται (raizi). raejoi (από το "rai") = αυτός και εγώ ταξιδεύουμε.

Ανάλογα την περίπτωση, τα ενθήματα μπορεί να είναι κτητικά, δηλαδή αν λέμε "mawatino" (ma[wa]t-in-o) = "εσύ - χρώμα ματιών", αν και δεν βγάζει νόημα ως "εσύ είσαι χρώμα ματιών", ωστόσο σημαίνει "το χρώμα των ματιών σου".

Τα ενθήματα προορίζονται να χρησιμοποιηθούν πραγματικά με φειδώ. Το καλύτερο είναι να αφήσετε το πρόσωπο να γίνεται κατανοητό από τα συμφραζόμενα. Επόμενος καλλίτερος τρόπος είναι να χρησιμοποιούμε τις αντωνυμίες (engo, tewo, to) και μόνο αν δεν βγαίνει νόημα, ικανοποιητικά, μπορεί κανείς να χρησιμοποιήσει τα προσωπικά ενθήματα.

Έτσι, οι συντομεύσεις είναι 13 επιθήματα, 4 προθήματα, και 3 ενθήματα, επιπρόσθετα με τα 5 κύρια επιθήματα (-o, -a, -i, eti, και η παράλειψη του -i). Για μερικούς ανθρώπους, αυτό μπορεί να φαίνεται περίπλοκο, αλλά αν συγκριθεί με μία φυσική γλώσσα, ή ακόμη και με μια τεχνητή γλώσσα όπως η 'Esperanto', το όλο σύστημα είναι εξαιρετικά στοιχειώδες. Πρέπει να θυμόμαστε ότι οι συντομεύσεις δεν

είναι υποχρεωτικές και υπάρχουν μόνο για να μας βοηθήσουν να βρούμε έναν τρόπο για να πούμε κάτι. Η γλώσσα μπορεί να χρησιμοποιηθεί αρκετά καλά χρησιμοποιώντας μόνο τα πέντε (5) κύρια επιθήματα, (συγκεκριμένα: -o, -a, -i, etí, και την παράλειψη του -i) τα οποία δεν είναι καθόλου πολλά. Ακόμα και η Toki Pona, η οποία είναι η πιο ακραία στην προσπάθεια απλούστευσης της γραμματικής, έχει τρία (3) γραμματικά προθήματα μαζί με την απόλυτα αυστηρή σειρά των λέξεων (το οποίο είναι επίσης ένας γραμματικός δείκτης).

Και τελευταίο αλλά όχι λιγότερο σημαντικό, να θυμόμαστε ότι η συστηματική γλώσσα μπορεί να χρησιμοποιηθεί χωρίς καμία κατάληξη, με την '-o' να προστίθεται προαιρετικά και μόνο για να διευκολυνθεί η προφορά. Όλες οι λέξεις που θεωρούνται ουσιαστικά με την αυστηρή σειρά των λέξεων, Επίθετο - Ουσιαστικό (A-N, Adjective-Noun, συμπληρωματικό - κύριο) του κινεζικού τύπου, όπου δεν υπάρχει καμία κατάληξη και ακόμη η γλώσσα μπορεί να λειτουργήσει ικανοποιητικά. Για παράδειγμα, ο συγγραφέας μας παροτρύνει να ξεκινήσουμε μια ιστορία σαν αυτή:

"ten gnos antrop en, e bor gnos panto kruf ko uko wora.....", (Υπήρξε ένας προφήτης, που μπορούσε να γνωρίζει όλα τα κρυφά και αόρατα πράγματα ...). Επίσης μας παραθέτει το πρώτο του ποίημα στη χωρίς επισυνάψεις Συστηματική γλώσσα και μας ζητά να προσπαθήσουμε να το καταλάβουμε: "geo pnowo ktupo wexo - foto xromo airo xero - suringo agaro sakaro fraso - akuo psuxo ewo ejno." (είναι ένα ποίημα για μία μελωδία).

Για την εκμάθηση θα πρέπει κανείς να ξεκινήσει από το λεξιλόγιο. Ο γνώστης του λεξιλογίου προσπαθώντας να κάνει προτάσεις και να μιλήσει για τα πάντα με Συστηματική γλώσσα, θα καταλάβει την ανάγκη ύπαρξης γραμματικών στοιχείων και στοιχείων παραγωγής προτάσεων.

2.1.13 Αμφισημία (Ambiguity)

Υπάρχει μικρή πιθανότητα για αμφισημία, διότι η αποφυγή της υπήρξε το πιο σημαντικό κριτήριο, κατά τη διάρκεια σχηματισμού του λεξιλογίου και των κανόνων της SostiMatiko. Ωστόσο, σε περίπτωση που μία μορφή (σχήμα), φαίνεται να μπορεί να ερμηνευτεί και με διαφορετικό τρόπο, τότε η πιο βασική μορφή επικρατεί (βλ. 2.1.14). Ακόμη και κάθε λέξη με κατάληξη '-o' θα μπορούσε να ερμηνευθεί ως "χωρίς κατάληξη" (απαιτείται ενέργεια). Για παράδειγμα το "anango" θα μπορούσε να εκληφθεί ως anang = πρέπει να επιβάλει μια αναγκαιότητα + o = αυτός που πρέπει να επιβάλει αναγκαιότητα. Αλλά δεν είναι έτσι, γιατί πρέπει να τη λάβουμε ως την πιο βασική μορφή και σε αυτή την περίπτωση "anango" είναι απλά η "αναγκαιότητα". Για ευκολότερη ελαχιστοποίηση της πιθανότητας για ασάφεια, το πρόθημα 'u-' έχει την εναλλακτική μορφή 'uo-' και το πρόθημα 'um-' έχει επίσης την εναλλακτική μορφή 'ume-'.

2.1.14 Βασικά στοιχεία και μορφές της γραμματικής

"o" είναι η κατάληξη που δηλώνει πράγμα. Σηματοδοτεί ένα ουσιαστικό (πράγμα ή πρόσωπο)

"a" είναι η κατάληξη που δηλώνει τρόπο. Σηματοδοτεί ένα επίρρημα (ή, μια λέξη που περιγράφει τον τρόπο) και προέρχεται από το ουσιαστικό

"i" είναι η κατάληξη που δηλώνει πράξη. Σηματοδοτεί ένα ενεργό ρήμα που προέρχεται από το ουσιαστικό

"etí" είναι η κατάληξη που δηλώνει διενέργεια ή τη λήψη κάποιας ενέργειας. Σηματοδοτεί ένα παθητικό ρήμα που προέρχεται από το ουσιαστικό

Ελλείπει "-i" σηματοδοτείται η προστακτική (ή υποτακτική) του ρήματος (μερικά ρήματα μπορούν επίσης να αντιμετωπίζονται ως μόρια)

Σύνθεση (ως μία λέξη) μπορεί να σχηματιστεί μόνο με το δεύτερο μέρος να είναι n(o)=αντικείμενο, m(o)=απαρέμφατο, s(o)=πληθυντικός.

Οι προθέσεις αναφέρονται και αντιμετωπίζονται ως λέξεις και είναι ap(a), is(a), di(a), ek'(a). Οι προθέσεις είναι άτονες ή τονίζονται στην τελευταία συλλαβή τους. Συνήθως χαρακτηρίζονται με "-" να τις ακολουθεί.

Σύνδεσμοι, επιφωνήματα και όλα τα άκλητα παρατίθενται επίσης και αντιμετωπίζονται ως λέξεις. Το συνδετικό ρήμα είναι συνήθως μηδενικό "zero" (copula is usually zero).

Λέξεις που φέρουν επιθήματα μπορούν να πάρουν και περαιτέρω επιθήματα τόσο όσο το αποτέλεσμα έχει λογική, π.χ. afi/et/i/o^s = πράγματα ή πρόσωπα που απελευθερώνονται και afi/et/ZERO/o^s = afietos = πρόσωπα που πρέπει να ελευθερωθούν.

Η χασμωδία (χάσμα) μπορεί προαιρετικά να αποφευχθεί με την προσθήκη ενός "λαρυγγικού" ήχου δίπλα στα "a", ενός "j" δίπλα στο "i" ή στο "e", ενός "w" δίπλα στο "u" ή στο "o".

Εάν ένα επίρρημα χρησιμοποιηθεί postpositionally²⁷, συνήθως ακολουθείται από στίξη ή ειδικότερα θα πρέπει να τονιστεί στην πρώτη συλλαβή του.

Η έμφαση των φωνηέντων είναι ελεύθερη (πχ τράβηγμα ή τονισμός). Ειδική έμφαση δίνεται με τις προθέσεις, τα postpositional επιρρήματα, και τις αναστροφα σχηματιζόμενα λέξεις.

Σύμφωνο χωρίς φωνήεν δύσκολα προφέρεται και γι' αυτό προσθέτουμε ένα φωνήεν μετά (από το σύμφωνο αυτό).

2.2 ΟΜΑΣ-III

Η Οργανωτική Μέθοδος Ανάλυσης Συστημάτων (ΟΜΑΣ)²⁸ είναι μία διαγραμματική τεχνική Ανάλυσης Συστημάτων και ανοίκει στην κατηγορία των τεχνικών γενικής περιγραφής "Διαγραμμάτων Ροής Δεδομένων". Οι διαγραμματικές τεχνικές ανάλυσης συστημάτων (systems analysis) αναπτύχθηκαν ως εργαλεία συστημικής σκέψης και απεικόνισης, παρέχοντας έναν όσο γίνεται πιο ολοκληρωμένο αλλά και ευέλικτο τρόπο περιγραφής των σχετικών εννοιών για κάθε προβλέψιμο πεδίο εφαρμογής τους, που δίνει έμφαση στην εποπτική αναπαράσταση με τη χρήση σχημάτων. Η ΟΜΑΣ-III²⁹ (βλ. Παράρτημα Β') είναι η τρίτη βελτιωμένη έκδοση της αρχικής ΟΜΑΣ.

Όπως ήδη προαναφέρθηκε (βλ. 1.1.7) είναι μια σχεδιασμένη διαδικασία για την επίτευξη του άριστου δυνατού προσδιορισμού της οργάνωσης (δομής και λειτουργίας/συμπεριφοράς) ενός αντικειμένου ή φαινομένου (συστήματος), σύμφωνα με την εφαρμογή βασικών οργανωτικών κανόνων, προσαρμοσμένων σε συγκεκριμένες συνθήκες.

²⁷ **Postposition** είναι η τοποθέτηση ενός τροποποιητή ή άλλου στοιχείου ομιλίας μετά τη λέξη που τροποποιεί ή μετά από αυτό με το οποίο είναι συντακτικά συναφές. Για παράδειγμα μία πρόθεση τοποθετείται **postpositionally** στην περίπτωση που θα μπει μετά το αντικείμενό της. Έτσι στη φράση these facts notwithstanding, το notwithstanding είναι μία postposition.

²⁸ Παπακίτσος (2010)

²⁹ Παπακίτσος (2013)

2.2.1 Βασική Υπόθεση

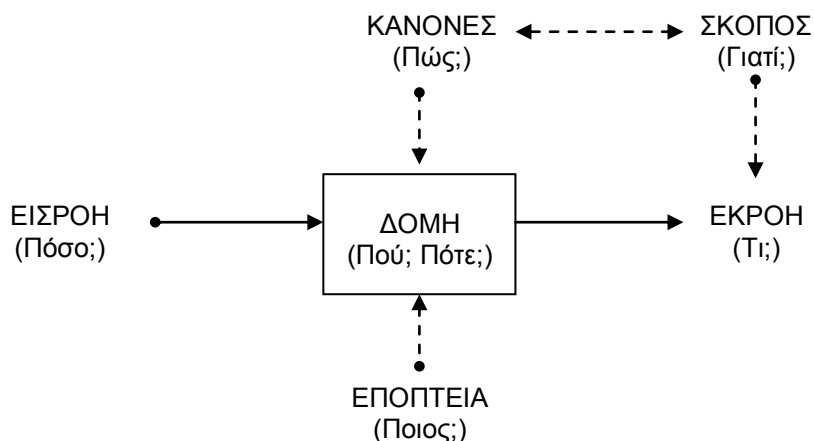
Η πλήρης κατανόηση ενός συστήματος απαιτεί να έχουν δοθεί απαντήσεις στα μοναδικά επτά θεμελιώδη ερωτήματα που το αφορούν:

- i) **Γιατί** υπάρχει και λειτουργεί;
- ii) **Τι** αποτελέσματα και συμπεράσματα δίνει;
- iii) **Πόσα** μέσα χρειάζεται (πόρους);
- iv) **Πώς** λειτουργεί;
- v) **Ποιος** παρακολουθεί ή καθοδηγεί τη λειτουργία του;
- vi) **Πού** λειτουργεί;
- vii) **Πότε** λειτουργεί;

Η κατανόηση του συστήματος οδηγεί στην πλήρη περιγραφή του ή και αντίστροφα, η πλήρης περιγραφή του οδηγεί στην κατανόησή της **δομής** (η διάρθρωση ενός συνόλου) και της **οργάνωσης** και **λειτουργίας** του (η διευθέτηση που αφορά τις σχέσεις μεταξύ των οντοτήτων της).

2.2.2 Βασική Περιγραφή

Η βασική περιγραφή ενός συστήματος γίνεται με τη βοήθεια της σημειογραφίας, όπως φαίνεται στο Σχεδιάγραμμα 2.0.1, υλοποιώντας τη Βασική Υπόθεση:



Σχεδιάγραμμα 2.0.1: Βασική σημειογραφία τής ΟΜΑΣ-III.

Κάθε σύστημα έχει **στοιχεία** δύο θεμελιωδών κατηγοριών:

- **Μέρη** που αφορούν τη Δομή του (βλ. 2.2.1.vi-vii) και
- **Παράγοντες** που αφορούν τις σχέσεις μεταξύ των Μερών του, δηλαδή την Οργάνωσή του.

Οι παράγοντες ενός συστήματος εξυπηρετούν τη Βασική Υπόθεση ως εξής:

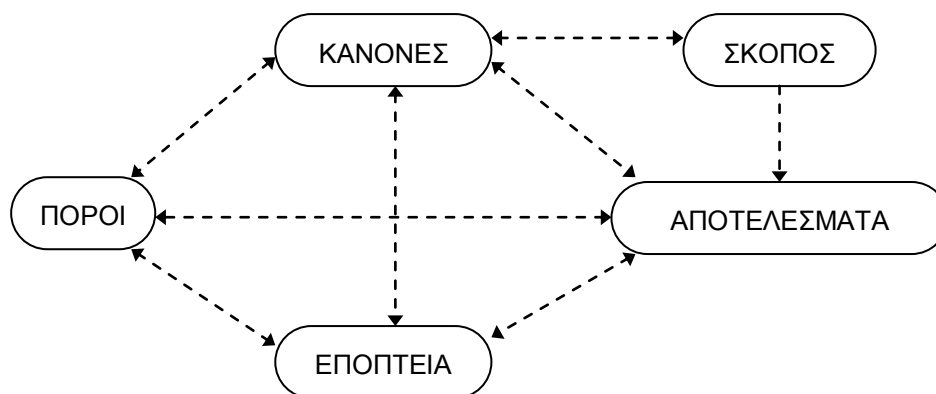
- **Σκοπός:** τα αίτια (βλ. 2.2.1.i) της λειτουργίας του (ποιοτικό στοιχείο).
- **Εισροή:** τα Μέσα (βλ. 2.2.1.iii), υλικά ή μη (Πόροι), που απαιτούνται γι' αυτήν (ποσοτικό στοιχείο).
- **Εκροή:** τα Αποτελέσματα και τα Συμπεράσματα (βλ. 2.1.1.ii) της λειτουργίας του.

- **Κανόνες:** Αρχές, Νόμοι, Κανονισμοί, Συμβάσεις και Συνθήκες που διέπουν τη λειτουργία του (βλ. 2.2.1.iv).
- **Εποπτεία:** οι φυσικές ή νομικές οντότητες που παρατηρούν, εκτελούν ή καθοδηγούν τη λειτουργία του (βλ. 2.2.1.v). Η απλή παρατήρηση είναι κατά περίπτωση δυνατόν να επηρεάσει τη συμπεριφορά του συστήματος (Αρχή της Απροσδιοριστίας) όπως ακριβώς και η καθοδήγησή του.

Οι παράγοντες του Σκοπού, της Εισροής, των Κανόνων και της Εποπτείας αποτελούν τα **Δεδομένα** λειτουργίας του Συστήματος. Τα Μέρη του συστήματος πραγματοποιούν τη λειτουργία του στο χώρο (βλ. 2.2.1.vi) και στο χρόνο (βλ. 2.2.1.vii), με τον επιμερισμό των διεργασιών ή της κατάστασής του σε **Επίπεδα** (χώρος) και **Φάσεις** (χρόνος). Ο συνδυασμός των Επιπέδων και Φάσεων προσδιορίζει τη **δομή** του. Κάθε Μέρος του συστήματος, είτε είναι στατικό είτε είναι δυναμικό, αποτελεί ένα υποσύστημα του αρχικού, για το οποίο ισχύει η ίδια βασική περιγραφή.

2.2.3 Ανάλυση

Η ανάλυση ενός συστήματος διενεργείται με τον προσδιορισμό όλων των στοιχείων του. Οι παράγοντες του συστήματος είναι διακριτοί αλλά αλληλένδετοι, γι' αυτό κάθε μεταβολή των χαρακτηριστικών τού ενός μπορεί να επηρεάζει και τους υπόλοιπους. Επομένως η μελέτη τους οφείλει να συμπεριλάβει όλους τους δυνατούς συνδυασμούς των μεταξύ τους σχέσεων, όπως καθορίζονται στο Σχεδιάγραμμα 2.0.2:



Σχεδιάγραμμα 2.0.2: Συσχέτιση Παραγόντων του Συστήματος.

Οι σχέσεις αυτές μπορεί να είναι ποιοτικές ή ποσοτικές κάθε είδους, όπως είναι οι σχέσεις επιρροής προς κάθε κατεύθυνση (μονόδρομη, αμφίδρομη). Ο προσδιορισμός τους γίνεται με την εξής διαδικασία:

Προσδιορίζεται ο Σκοπός ύπαρξης και λειτουργίας.

2.0.3. Προσδιορίζεται ο Σκοπός ύπαρξης και λειτουργίας.

2.0.4. Καθορίζονται τα ζητούμενα Αποτελέσματα (Εκροές), σύμφωνα με το Σκοπό.

2.0.5. Συγκεντρώνονται οι Κανόνες που αφορούν και διέπουν τη λειτουργία του. Εξετάζονται οι πιθανές σχέσεις τους με το Σκοπό και τ' Αποτελέσματα.

2.0.6. Επιλέγονται οι Πόροι (Εισροές) που πιθανολογείται ότι θα αποδώσουν τα ζητούμενα Αποτελέσματα σύμφωνα με τους Κανόνες.

2.0.7. Ορίζεται ο τρόπος Εποπτείας του συστήματος, έτσι ώστε να εκτελείται η λειτουργία του σύμφωνα με τους Κανόνες και τα ζητούμενα Αποτελέσματα. Εξετάζεται η καταλληλότητα της Εποπτείας σχετικά με τους Πόρους.

2.0.8. Διασπάται σταδιακά το Αποτέλεσμα σε μικρότερα επιμέρους αποτελέσματα (Στόχοι), ως προς τις συνιστώσες του χρόνου (Φάσεις) και του χώρου (Επίπεδα), με κριτήριο τη μεγαλύτερη δυνατή αυτονομία τους, μέχρι να καταλήξουμε στην αρχική λήψη των Πόρων. Κάθε επιμέρους αποτέλεσμα ταυτίζεται με ένα Μέρος του συστήματος που εκτελεί την αντίστοιχη διεργασία παραγωγής του.

2.0.9. Εξετάζεται σε κάθε στάδιο η ανάλυση ως προς τους παράγοντες και συνδέεται με αυτούς. Αναθεωρείται ο επιμερισμός των διεργασιών, αν κριθεί αναγκαίο.

2.0.10. Οριστικοποιείται η Δομή του Συστήματος με τον ορισμό των Μερών του. Με την έναρξη λειτουργίας του συστήματος, παρατηρούνται διαρκώς τα παραγόμενα Αποτελέσματα και αξιολογούνται ως προς το Σκοπό ύπαρξής του και ως προς τη σχέση τους με τους Κανόνες. Η αξιολόγηση αυτή αποτελεί τα Συμπεράσματα, που είναι διακριτό μέρος της Εκροής του συστήματος. Μελετάται ο τρόπος που τα Συμπεράσματα επηρεάζουν τα γνωρίσματα των υπολοίπων παραγόντων και πιθανώς επιβάλλουν την αναθεώρησή τους (Ανάδραση).

2.3 Παραπομπές και URLS αναφορών κεφαλαίου

- **Katakana:**
<https://el.wikipedia.org/wiki/%CE%9A%CE%B1%CF%84%CE%B1%CE%BA%CE%AC%CE%BD%CE%B1> [23]
- **Latino_Sine_Flexione:**
https://en.wikipedia.org/wiki/Latino_sine_flexione [24]
- **Schwa:**
<https://en.wikipedia.org/wiki/Schwa> [22]
- **Tagalog:**
<https://en.wikipedia.org/wiki/Tagalog> [25]
- **Verbs_From_Nouns:**
http://www.academia.edu/4226602/making_verbs_from_nouns [26]
- **X και η ποικιλότητά του:**
http://en.wikipedia.org/wiki/Voiceless_velar_fricative [21]
- Παπακίτσος Ε.Χ. (2010). **Οργανωτική Μέθοδος Ανάλυσης Συστημάτων.** Διαγραμματική τεχνική Συστημικής Θεωρίας με παραδείγματα εφαρμογών. ΠΚΙΕ 626: Ε. Κ. Θεσσαλού, Αθήνα. [28]
- Παπακίτσος Ε.Χ. (2013). **Γλωσσική Τεχνολογία Λογισμικού: Ι. Προετοιμασία.** Εκπαιδευτικό Εγχειρίδιο, ISBN: 978-960-93-5636-7. Εθνική Βιβλιοθήκη της Ελλάδος, Αθήνα. [29]

ΣΧΕΔΙΑΣΗ

Στο παρόν κεφάλαιο θα επιδιώξουμε να προσεγγίσουμε σχεδιαστικά τις υπολογιστικές λειτουργίες του συστήματός μας ώστε να καταλήξουμε στο διάγραμμα ροής δεδομένων που θα το περιγράφει.

3.1 Προσεγγίζοντας τις βασικές λειτουργίες

Σε γενικές γραμμές το σύστημά μας θα πρέπει:

- να χειρίζεται δεδομένα που θα φτάνουν σε μορφή μίας πρότασης ή ενός συνόλου προτάσεων,
- να προβαίνει σε διευκρινιστικά ερωτήματα εκεί που απαιτείται και είναι εφικτό,
- να παρουσιάζει μία λίστα ενεργειών, τοποθετημένες με χρονική σειρά και ακολουθούμενες από τα πλήρη στοιχεία του ποιος, που, πως πότε και γιατί (εφόσον δίδεται).

Να επισημάνουμε, το πώς τα δεδομένα φτάνουν (π.χ. μετατροπή ομιλίας σε κείμενο) και το πως η λίστα ενεργειών μετατρέπεται σε μηχανικές κινήσεις είναι άλλο αντικείμενο και δεν καλύπτεται στην παρούσα εργασία.

3.1.1 Χειρισμός δεδομένων

Μία πρόταση ορίζεται από το πρώτο γράμμα έως μία τελεία ή μεταξύ από δύο τελείες. Αυτή είναι η πρώτη διαδικασία που γίνεται ώστε να πάρουμε όλες τις λέξεις που συμπεριλαμβάνει (η πρόταση) και να τις ελέγξουμε σύμφωνα με τη γραμματική, τη θέση και το νόημά τους. Στην πραγματικότητα οι λέξεις θα εισέρχονται μία προς μία έως ότου βρούμε τελεία. Κάθε λέξη θα αναλύεται³⁰ και τα αποτελέσματά της θα καταχωρούνται σε έναν προσωρινό πίνακα. Προχωράμε δηλαδή σε μία διαδικασία αναδρομής ώστε ταυτόχρονα να ελέγξουμε και τη γραμματικότητα της εκάστοτε πρότασης. Σύμφωνα με την γραμματική της **SostiMatiko** (βλέπε 2.1.5) και για να διατηρείτε η έννοια της μινιμαλιστικής γλώσσας θα πρέπει η σειρά των λέξεων να είναι SVO³¹ και AN³². Άρα νόμιμες θα ορίζονται όσες προτάσεις πληρούν τα παραπάνω. Φυσικά προτάσεις που θα παρουσιάζουν ελλείψεις σε υποκείμενο και αντικείμενο, δεν θα απορρίπτονται αλλά αντίθετα θα διεγείρουν έναν μηχανισμό διευκρινίσεων, όπου αυτό κριθεί απαραίτητο και δεν εννοείτε κάτι συγκεκριμένο. Για παράδειγμα στην περίπτωση προστακτικής εννοείτε πάντα το υποκείμενο που είναι το β' πρόσωπο για τον εντολέα (άρα το α' πρόσωπο για το σύστημά μας), όμως το αντικείμενο όχι (δεν εννοείτε πάντα). Έτσι στην πρόταση "έλα" εννοούμε "εσύ έλα εδώ", αλλά στην πρόταση "πήγαινε" εννοούμε "εσύ πήγαινε κάπου" αυτό το κάπου δεν εννοείται και το σύστημα θα πρέπει να το αναζητήσει. Οι διευκρινίσεις που θα χρειαστούν αρχικά θα αναζητούνται σε υπάρχοντα δεδομένα του εισερχόμενου κειμένου (εφόσον πρόκειται για κείμενο) και κατόπιν αν δεν έχει

³⁰ Μην ξεχνάμε πως η **SostiMatiko** σε μία λέξη μπορεί και ενσωματώνει άνω του ενός γραμματικά δεδομένα (άρθρο, ρήμα κτλ §2.1).

³¹ Υποκείμενο - ρήμα - αντικείμενο

³² Οι λέξεις που χαρακτηρίζουν ένα ουσιαστικό (επίθετα ή επιρρήματα) καθώς και τα συμπληρώματα του, θα πρέπει να το ακολουθούν.

βρεθεί απάντηση τότε θα υποβάλλονται ερωτήματα (βλέπε 3.1.2). Είναι απαραίτητο να χρησιμοποιήσουμε αρχικές (μηδενικές - default) τιμές, οι οποίες και θα κατέχουν τη θέση των εννοούμενων (γραμματικών) στοιχείων. Στο παράδειγμα της προστακτικής που είδαμε πιο πάνω μία μηδενική τιμή για το σύστημα είναι το εγώ (του). Έτσι με αυτό τον τρόπο αναδρομής και συμπλήρωσης της κάθε πρότασης ο προσωρινός πίνακας θα οριστικοποιείται και τα πλήρη δεδομένα θα μεταφέρονται στην τελική λίστα ενεργειών όπου και θα τοποθετούνται στην σωστή χρονική θέση.

3.1.2 διευκρινιστικά ερωτήματα

Η διαδικασία αυτή θα πυροδοτείται εφόσον προκύπτουν ελλείψεις υποκειμένων – αντικειμένων. Σκοπός είναι να προσδώσουμε ευφυΐα στο σύστημα ώστε να εξαλείφονται οι αοριστίες στην υλοποίηση της επικοινωνίας με τη μηχανή και να μην απορρίπτονται ως μη νόμιμες, κάποιες (τουλάχιστο) προτάσεις, που κρύβουν εντός τους γραμματικά – συντακτικά στοιχεία. Τα ερωτήματα προκύπτουν χάρη στην εφαρμογή του μοντέλου της OMAS III, που τα υποβάλει, όπως θα δούμε σε επόμενη παράγραφο (βλέπε 3.2).

3.1.3 Τελική λίστα ενεργειών

Επιδιώξή μας είναι το σύστημα να αποδίδει την ενέργεια που του ζητήθηκε (σε μορφή λίστας εδώ) και ταυτόχρονα να μπορεί να προσδιορίζει χρονικά την στιγμή απόδοσής της. Και αυτή η διαδικασία προσδίδει ευφυΐα στο σύστημα εφόσον δεν το περιορίζει στα στενά πλαίσια των FIFO και LIFO³³ συστημάτων. Αρχικά η διαλογή θα γίνεται για τις τρεις βασικές ομάδες χρονικού προσδιορισμού, που είναι αυτές του παρελθόντος, παρόντος και μέλλοντος. Αυτή η πληροφορία είναι ούτως ή άλλως μέσα σε κάθε πρόταση. Στη συνέχεια και εφόσον η πρόταση περιέχει πιο συγκεκριμένη χρονική πληροφορία η ενέργεια θα χαρακτηρίζεται περαιτέρω. Αυτό θα μπορούσε να συνεχιστεί και για ακόμη πιο συγκεκριμένη πληροφορία εντός της γενικότερης (αν υπάρχει), αλλά σε αυτή την εργασία δεν θα προχωρήσουμε σε βαθύτερο επίπεδο διαλογής από αυτό. Δημιουργούμε έτσι τρεις ομάδες που κάθε ομάδα αποτελείται από γεγονότα ειδικά και από γεγονότα αόριστα. Επειδή τα ειδικά τα θεωρούμε πιο σημαντικά (στο θέμα του χρόνου), θα ορίσουμε ότι σε κάθε μία ομάδα από τις τρεις θα υπάρχουν δύο υποομάδες, όπου η μία θα προηγείται της άλλης. Για κάθε κατηγορία η πρώτη (1^η) υποομάδα είναι αυτή με τα αόριστα γεγονότα και η δεύτερη (2^η) αυτή με τα πιο συγκεκριμένα, όπου και θα προηγείται. Στην υποκατηγορία των αορίστων (1^{ης}) θα ισχύει το FIFO. Το ίδιο σύστημα θα ισχύει και στα χρονικά όμοια γεγονότα της δεύτερης (2^{ης}) υποκατηγορίας.

Για παράδειγμα λέμε τις εξής προτάσεις: 1) “θα πας στο σημείο Ε”, 2) “θα πας στο σημείο Γ αύριο”, 3) “θα πας στο σημείο Β αύριο”, 4) “θα πας στο σημείο Δ αύριο το μεσημέρι”, 5) “θα πας στο σημείο Α” και 6) “θα πας στο σημείο ΣΤ στις 2 η ώρα”. Και οι έξι προτάσεις ανήκουν στην κύρια ομάδα μέλλον. Από αυτές οι «1» και «5» είναι αόριστες και εφόσον η «1» έρχεται πρώτη, προηγείτο της «5». Όσον αφορά τις άλλες παρουσιάζονται τρία στοιχεία χρονικά, η ώρα, το αύριο και το μεσημέρι. Από αυτά η SostiMatiko διαθέτει μόνο το “ώρα” σαν λέξη. Το μεσημέρι μπορούμε να το πούμε ως μέσο της ημέρας αφού υπάρχουν και οι δύο λέξεις. Αντίστοιχα θα μπορούσαμε να φτιάξουμε το αύριο ως «πέραν μίας ημέρας» κτλ . Οι προτάσεις που

³³ FIFO: First in First out και LIFO: Last in First out. Χαρακτηρισμός συστημάτων αποθήκευσης (δεδομένων, προϊόντων κτλ) σύμφωνα με τον τρόπο που εισέρχονται και εξέρχονται τα προς αποθήκευση (δεδομένα, προϊόντα κτλ)

περιέχουν τη λέξη “αύριο” είναι τρεις. Θα παρατηρήσουμε πως μία πρόταση αναφέρει και 2^ο προσδιορισμό αλλά θα μπορούσε και 3^ο κοκ. Εμείς δεν ξεπερνάμε το 2^ο προσδιορισμό και έτσι όλες οι προτάσεις που περιέχουν το αύριο θα τοποθετηθούν σύμφωνα με το FIFO πρότυπο. Μετά από αυτά η σειρά τους θα είναι: «2», «3», «4». Τέλος, περισσεύει η πρόταση «6» η οποία δίνει μόνο ώρα. Η ώρα προηγείται της ημέρας άρα η «6» θα μπει μπροστά από όλες τις άλλες προτάσεις και έτσι θα πάρουμε την παρακάτω λίστα:

- 6) “θα πας στο σημείο ΣΤ στις 2 η ώρα
- 2) “θα πας στο σημείο Γ αύριο”
- 3) “θα πας στο σημείο Β αύριο”
- 4) “θα πας στο σημείο Δ αύριο το μεσημέρι”,
- 1) “θα πας στο σημείο Ε
- 5) “θα πας στο σημείο Α”

Να επισημάνουμε πως οι προτάσεις του παραδείγματος δίνονται σε απλή Ελληνική και όχι στη SostiMatiko και ούτε ακολουθούν τη σύνταξη αυτής. Έτσι είναι πιο εύκολο να το καταλάβει ο αναγνώστης πριν περάσουμε αποκλειστικά στην τεχνητή γλώσσα.

Να επισημάνουμε επίσης πως στη SostiMatiko δεν υπάρχουν οι ημέρες και οι μήνες αλλά ούτε και η λέξη μήνας η οποία μπορεί να προσδιοριστεί μέσα από τη λέξη «fengo» που προσδιορίζει το φως του φεγγαριού, το φεγγάρι κτλ.

3.2 Η ΟΜΑΣ III και ο Φορμαλισμός

Εδώ και πριν προχωρήσουμε με τις διαδικασίες του προγράμματος θα πρέπει να παρουσιάσουμε τη χρήση του μοντέλου της ΟΜΑΣ III ως εργαλείο φορμαλισμού για τον έλεγχο και την ανάλυση της κάθε πρότασης. Αυτό που έχουμε να κάνουμε είναι να αντιστοιχίσουμε τα 7 ερωτήματα του μοντέλου με τα στοιχεία γραμματικής – συντακτικού από τα οποία και βάση τα οποία δομείται η κάθε πρόταση. Έχοντας απαντηθεί και τα 7 ερωτήματα, το σύστημά μας λαμβάνει όλη τη «γνώση» που μπορεί να του προσφερθεί. Έτσι μέριμνά του είναι να αντλήσει όλες τις απαιτούμενες απαντήσεις και αν δεν μπορεί να το κάνει στο εσωτερικό του πρέπει να εξωτερικεύει τις απορίες του. Αυτό και μόνο ανεβάζει την ευφυΐα του. Ας αντιστοιχίσουμε ένα προς ένα τα 7 μοναδικά θεμελιώδη ερωτήματα.

3.2.1 Το ερώτημα Γιατί

Δεν χρήζει απάντησης στην παρούσα εργασία γι’ αυτό και δεν θα επιστρέφεται ερώτηση στην περίπτωση της μη ικανοποίησης του. Στο “Γιατί” τοποθετούνται οι αιτιολογικοί και επεξηγηματικοί σύνδεσμοι «γιατί» και «επειδή». Επίσης και το “να” ως σύνδεσμος μπορεί να δηλώνει επεξήγηση ή αιτιολογία. Σε κάθε περίπτωση μιλάμε για δευτερεύουσες προτάσεις όπου θα περνάνε και αυτές την διαδικασία αναδρομής. Επισημαίνουμε επίσης πως η SostiMatiko έχει μόνο το “να” (ina) ως σύνδεσμο που δηλώνει και σκοπό.

3.2.2 Το ερώτημα Τι

Είναι η έξοδος του μοντέλου και η απάντηση είναι: «αυτό που λέει το ρήμα». Το ρήμα λοιπόν είναι η ζητούμενη έξοδος. Το σύστημα εντοπίζει το ρήμα στην εισερχόμενη πρόταση. Αυτό που στην πραγματικότητα κάνει το σύστημα, είναι να απογυμνώνει τις λέξεις από τα διάφορα πρόσθετα που μπορεί να έχουν φορτωθεί (βλέπε 2.1.12.x) και εκεί βρίσκει ή όχι το ρήμα. Στη περίπτωση που το σύστημα δεν βρει ρήμα στην τρέχουσα πρόταση και βρίσκει ενδείξεις υποκειμένου ή αντικειμένου (κυρίως), τότε πρέπει να υποβάλει ερώτημα τι πρέπει να γίνει, τι ενέργεια υποτίθεται ότι πρέπει να πράξει.

3.2.3 Το ερώτημα Πόσο

Η είσοδος του μοντέλου μας. Εδώ θα έπρεπε να τοποθετούμε τα επίθετα, τα άρθρα και γενικότερα όλα τα προσδιοριστικά των ουσιαστικών της πρότασης. Αυτό θα ήταν σωστό από γραμματικής πλευράς. Εμάς όμως μας ενδιαφέρει η σημασιολογική εξέταση του θέματος και έτσι στο ερώτημα αυτό θα τοποθετείται κάθε αντικείμενο των προτάσεων, όλοι οι ποσοτικοί προσδιορισμοί και γενικότερα οτιδήποτε κατέχει θέση αντικειμένου και δεν είναι τροπικά, χρονικά, τοπικά επιρρήματα και προσδιορισμοί.

3.2.4 Το ερώτημα Πως

Δηλώνει τον τρόπο που θα δοθεί η ενέργεια του ρήματος. Ο τρόπος δίνεται από τα τροπικά επιρρήματα, τις ενεργητικές μετοχές απαρέμφατα ως επιρρηματικοί προσδιορισμοί και κάθε άλλος προσδιορισμός που δηλώνει τρόπο. Στη SostiMatiko δεν έχουμε μετοχές και επίσης όλα τα επιρρήματα δηλώνουν τρόπο. Έτσι, εκτός όποιου άλλου τροπικού προσδιορισμού, μέσω συμπληρωματικής πρότασης ίσως, το ερώτημα απαντάται κυρίως με όποιο επίρρημα εντοπίζεται στην πρόταση.

3.2.5 Το ερώτημα Ποιος

Η απάντηση εδώ είναι το υποκείμενο. Έχοντας το ρήμα και το αντικείμενο είναι ξεκάθαρο πιο είναι αυτό μέσα στην πρόταση. Παρά ταύτα αν δούμε μετά από ουσιαστικό το “τι” γίνεται αμέσως ξεκάθαρο πως το ουσιαστικό αυτό είναι υποκείμενο (βλέπε 2.1.7). Επίσης το υποκείμενο μπορεί να ενσωματώνεται μέσα στο ρήμα (βλέπε 2.1.12.3) ή ακόμα και να εννοείτε, άρα παραλείπεται. Αν καμία από τις παραπάνω περιπτώσεις δεν πληρείται τότε το σύστημα εξωτερικεύει την ερώτηση “ποιος(;)”!

3.2.6 Το ερώτημα Που

Αυτό είναι ένα ερώτημα που δεν χρήζει απάντησης. Όταν δεν προσδιορίζεται ο τόπος τότε εννοείται (ως τόπος) εκείνος που υπάρχει το σύστημά μας, τη δεδομένη στιγμή, είτε ως φυσική παρουσία, είτε ως νοητή! Ο τόπος μπορεί να ορίζεται από κάποια προηγούμενη πρόταση είτε κάποια επόμενη, του τρέχοντα κειμένου. Ο τόπος μπορεί να ορισθεί στην τρέχουσα πρόταση είτε ως απόλυτη θέση, είτε ως μία περιοχή (ή έκταση) από έως, με η χωρίς εντολή κίνησης. Εάν η τοποθεσία απαιτείται και δεν δίνεται ούτε μπορεί να προσδιοριστεί τότε μία ερώτηση γίνεται από το σύστημα.

3.2.7 Το ερώτημα Πότε

Εδώ θέλουμε το χρόνο ή τη στιγμή, που γίνεται ή θα γίνει ή έγινε, η ενέργεια! Θα μπορούσε να είναι η γενικότερη μορφή του χρόνου που δηλώνεται πάντα από το ρήμα (όπως δηλώνεται με τους χρόνους στις φυσικές γλώσσες) ή από χρονικούς προσδιορισμούς. Αυτή η προσέγγιση αφορά το ευρύτερο τώρα (αυτή την περίοδο, σήμερα, τον τρέχοντα μήνα κτλ) το μέλλον με την ευρύτερη έννοιά του και το παρελθόν. Το πότε μπορεί να γίνει πιο συγκεκριμένο προσδιορίζοντας την ώρα την ημέρα ή γενικότερα με αυστηρό χρονικό προσδιορισμό.

Στην παράγραφο 2.2.2 αναφέρεται πως κάθε σύστημα έχει **στοιχεία** δύο θεμελιωδών κατηγοριών:

- **Μέρη** που αφορούν τη Δομή του
- **Παράγοντες** που αφορούν την Οργάνωσή του.

Παρατηρούμε λοιπόν από τα παραπάνω και όπως τα έχουμε αντιστοιχίσει πως από τα 3.2.1 έως και 3.2.5, αντλούμε τα στοιχεία που αφορούν τη οργάνωση της πρότασης και προσδιορίζουν την διεξαγωγή κάποιας ενέργειας, ενώ από τα 3.2.6 και 3.2.7 αντλούμε τη δομή του συστήματος και όσο αφορά το χώρο και το χρόνο διεξαγωγή της ενέργειας αυτής.

3.3 Ειδικότερα με τις διαδικασίες του προγράμματος

Σε αυτό πλέον το σημείο και έχοντας προσεγγίσει τις βασικές λειτουργίες (βλέπε 3.1.x) του συστήματος, καθώς και έχοντας δομήσει τον φορμαλισμό με την χρήση της ΟΜΑΣ III (βλέπε 3.2.x), θα προσδιορίσουμε στη γενική τους μορφή τις διαδικασίες που θα πλαισιώνουν το πρόγραμμά του.

3.3.1 Επιλογή αρχείου κειμένου

Σε κάποιο σημείο στο δίσκο θα υπάρχουν αρχεία με κειμενάκια που θα περιέχουν προτάσεις σε SostiMatiko. Κάποιες προτάσεις από το σύνολο θα συνδέονται μεταξύ τους, πχ όσον αφορά το υποκείμενο κτλ., οπότε θα μπορούν να είναι ελλειπτικές ως προς αυτό.

Η διαδικασία αυτή είναι υπεύθυνη για το άνοιγμα επιλεγμένου αρχείου και το κλείσιμό του μετά το πέρας άντλησης δεδομένων κοκ. Λαμβάνει αναφορά από τη διαδικασία της *υπόθεσης 1* και αλληλεπιδρά με αυτή της *τοποθέτησης γραμμής*³⁴. Όταν η *υπόθεση 1* ενημερώσει ότι έχει τέλος αρχείου τότε προωθείται ένα αίτημα στην *τοποθέτηση γραμμής*³⁴. Όταν και αυτή η διαδικασία ολοκληρώσει όλες τις εργασίες της επιστρέφει μία επιβεβαίωση για να επιτραπεί το κλείσιμο του τρέχοντος αρχείου και ένα νέο να ανοίξει αν αυτό είναι επιθυμητό.

3.3.2 Ανάγνωση λέξης

Η κάθε λέξη πλαισιώνεται από κενό ή από κενό και τελεία. Η διαδικασία αυτή διαβάσει την επόμενη λέξη στο αρχείο ή την τελεία ή το τέλος του κειμένου. Το αποτέλεσμα το στέλνει σε μία υπόθεση τύπου “case” (*Υπόθεση 1*).

³⁴ Διαδικασία “τοποθέτηση γραμμής στον τελικό πίνακα, στην σωστή χρονική σειρά”

3.3.3 Υπόθεση 1

Το εισερχόμενο δεδομένο περνάει από φίλτρο τριών επιλογών:

- Αν είναι λέξη τότε στέλνεται για ανάλυση.
- Αν είναι τελεία τότε ενημερώνεται η διαδικασία *Σύνταξη προσωρινής Δομής* ότι η πρόταση ολοκληρώθηκε.
- Αν είναι το τέλος του κειμένου τότε επιστρέφουμε στη διαδικασία *Επιλογή Αρχείου Κειμένου* για το κλείσιμο του αρχείου και άνοιγμα νέου, εφόσον αυτό είναι επιθυμητό. Για να τερματίσει το τρέχον αρχείο θα πρέπει να έρθει αναφορά και από την διαδικασία *Τοποθέτησης Γραμμής στον Τελικό Πίνακα*, στη *Σωστή Χρονική Σειρά*.

3.3.4 Ανάλυση Λέξης

Εδώ γίνεται η αποδόμηση της λέξης και απαλλαγή της από τα διάφορα πρόσθετα συστατικά της, τα οποία όμως ένα προς ένα μπαίνουν ως δείκτες στις διάφορες κατηγορίες που ανήκει η λέξη αυτή. Δηλαδή, είναι ρήμα, είναι πληθυντικός, υπάρχουν αντωνυμίες ώστε ταυτόχρονα να δηλώνεται και το υποκείμενο, είναι παθητική φωνή κτλ. Όλο αυτό επιτυγχάνεται με την συνεχή επικοινωνία με τις βάσεις δεδομένων γραμματικής και λεξικού που έχουν αναπτυχθεί στην μνήμη του συστήματος. Το αποτέλεσμα της διαδικασίας είναι ένας μονοδιάστατος πίνακας με τη βασική λέξη και κάποιες θέσεις που τοποθετείτε “0” ή “1” (false ή true) οι οποίες αντιπροσωπεύουν τους δείκτες σταθερών κατηγοριών που μπορεί να ανοίξει η όποια λέξη. Το αποτέλεσμα αυτό στέλνεται σε μία υπόθεση τύπου “case” (Υπόθεση 2). Να επισημάνουμε εδώ πως η διαδικασία θα διαθέτει μεταβλητή άμεσης επανάκλησης, από την επόμενη της, για να δοθεί και όποια άλλη λέξη υπάρχει κρυμμένη εντός της αρχικής (και μπορεί να κατέχει θέση υποκειμένου για παράδειγμα).

3.3.5 Υπόθεση 2

Το εισερχόμενο δεδομένο περνάει από φίλτρο έξι επιλογών:

- Αν είναι ρήμα (απαντά στο Τι).
- Αν είναι προσδιορισμός επεξήγησης (απαντά στο Γιατί).
- Αν είναι επίρρημα ή άλλος προσδιορισμός τρόπου (απαντά στο Πώς).
- Αν είναι αντικείμενο ή άλλοι ποσοτικοί προσδιορισμοί (απαντά στο Πόσο).
- Αν είναι υποκείμενο (απαντά στο Ποιός).
- Αν δηλώνεται ρητά τόπος ή χρόνος (απαντά στο Που ή Πότε).

Να επισημάνουμε ότι αν μιλάμε για προσδιορισμό επεξήγησης ή ένδειξη συμπληρωματικής πρότασης, τότε, η πληροφορία αυτή ενεργοποιεί αντίστοιχη διαδικασία εντός της *Σύνταξης προσωρινής Δομής* (διαδικασίας), ώστε να υποδεχθεί τη νέα αυτή πρόταση λέξη προς λέξη, διαμέσου της ίδιας υπόθεσης (Υπόθεση 2).

Η παρούσα διαδικασία εκτός από το να δρομολογεί τα δεδομένα προς την *Σύνταξη προσωρινής Δομής* (διαδικασίας), με την οποία και παρουσιάζει κάποια αμφίδρομη επικοινωνία, είναι αυτή που ενημερώνει την διαδικασία *Ανάγνωσης Λέξης* να προχωρήσει σε νέα ανάγνωση. Επίσης υπάρχει μία αμφίδρομη επικοινωνία με την διαδικασία *Προσδιορισμού Χώρου - Χρόνου* σε θέματα που αφορούν το αντίστοιχο φίλτρο της υπόθεσης. Ο χώρος και ο χρόνος δεν αφορούν τη δομή της πρότασης αλλά το επίπεδο διεξαγωγής του γεγονότος που αναφέρει αυτή, σε σχέση και με το που και με το πότε. Είναι στοιχεία απαραίτητα ακόμα και στην αοριστία και έτσι τα ανακτούμε από ρητές δηλώσεις, από τα συμφραζόμενα ή από αρχικές τιμές και τα εντάσσουμε μετέπειτα στην τελική γραμμή γεγονότος.

3.3.6 Προσδιορισμός Χώρου – Χρόνου

Ανταλλάσει πληροφορίες με την *Υπόθεση 2* καθώς και επικοινωνεί με τη βάση δεδομένων γραμματικής ώστε να διαμορφώσει το χρονικό και τοπικό πλαίσιο διεξαγωγής γεγονότος. Εάν το αποτέλεσμα δεν είναι το επιθυμητό τότε η τελική διαμόρφωση θα επιτευχθεί με ανταλλαγή - προώθηση πληροφοριών από την *Σύνταξη προσωρινής Δομής* (διαδικασίας), η οποία χειρίζεται τα θέματα αρχικών τιμών περιβάλλοντος και εξωτερίκευσης ερωτημάτων. Η διαδικασία επίσης φροντίζει να σηματοδοτήσει με μία ακολουθία χαρακτήρων το επίπεδο του χρόνου. Η ακολουθία αυτή θα αντιστοιχεί σε μία κλίμακα από (α) ελάχιστο έως (β) μέγιστο τη χρονική στιγμή που αφορά το συμβάν. Έτσι κάθε συμβάν θα φέρει ένα τέτοιο χαρακτηριστικό που με μία απλή καταχώρηση-τακτοποίηση από μικρότερο προς μεγαλύτερο θα παίρνει τη σωστή χρονική θέση μέσα στον πίνακα εξόδου.

3.3.7 Σύνταξη Προσωρινής Δομής

Όχι η μόνη αλλά η πιο σημαντική διαδικασία του κώδικα καθώς αποτελεί το πιο κομβικό σημείο για πολλές άλλες διαδικασίες. Μέχρι στιγμής είδαμε ότι:

- Από την *Υπόθεση 1* ενημερώνεται πως η πρόταση ολοκληρώθηκε.
- Από την *Υπόθεση 2* λαμβάνει τους πίνακες λέξεων με τα διακριτικά που τις χαρακτηρίζει.
- Με την διαδικασία *Προσδιορισμού Χώρου – Χρόνου* ανταλλάσει πληροφορίες για την ανάκτηση της γνώσης του “που και πότε”.

Τα επιπλέον στοιχεία της διαδικασίας είναι:

- Αν η πρόταση δεν πληρεί τα χαρακτηριστικά της γραμματικότητας, σύμφωνα με το συντακτικό της γλώσσας, τότε προωθείτε προς τον πραγματικό κόσμο ένα αντίστοιχο μήνυμα.
- Για την συμπλήρωση ελλείψεων σε θέματα δομής της πρότασης (όπως υποκείμενο) γίνεται μία αμφίδρομη επικοινωνία σε διαδικασίες λήψεως *αρχικών τιμών, Περιβάλλοντος* ή και *υποβολής ερώτησης* προς τον πραγματικό κόσμο. Οι τρεις αυτές διαδικασίες μπορούν να κληθούν άμεσα ή η μία δια μέσου τις άλλης από την παρούσα.
- Εξωτερίκευση υπόθεσης τύπου “ If ” *Ελλείψεων Δομής* ώστε να τερματίσει ή να συνεχίσει τη διαδικασία σύνταξης.

Με απλά λόγια η παρούσα διαδικασία δημιουργεί έναν προσωρινό μονοδιάστατο πίνακα στον οποίο αναπτύσσει τα δεδομένα μίας πρότασης ώστε να προκύψει μία γραμμή δεδομένων που θα λέει: ποιος θα πράξει, ποια ενέργεια, γιατί να γίνει (ίσως), με ποιο τρόπο, με ποια ένταση ή διάρκεια, σε ποιο χρόνο και σε ποιο τόπο. Λέμε γραμμή δεδομένων γιατί το αποτέλεσμα δεν θα είναι μία νέα πρόταση αλλά δεδομένα εξόδου ώστε να οργανωθεί το σύστημα και να πράξει αυτό που του ζητείται μέσω της πρότασης.

3.3.8 Ελλείψεις Δομής

Εδώ γίνεται έλεγχος εάν η προσωρινή δομή της γραμμής δεδομένων προς έξοδο είναι πλήρης ή έχει ελλείψεις. Σε περίπτωση που υπάρχουν ελλείψεις (απαντάει καταφατικά) τότε ενημερώνεται η *διαδικασία προσωρινής δομής* για αναζήτηση δεδομένων. Αυτό επαναλαμβάνεται έως ότου η απάντηση είναι αρνητική (στην έλλειψη δεδομένων) ώστε να ανοίξει ο δρόμος της τοποθέτησης της γραμμής στον διδιάστατο πίνακα εξόδου (δεδομένων). Πρέπει να δοθεί προσοχή στο να αποφευχθούν ατέρμονες βρόχοι.

3.3.9 Τοποθέτηση Γραμμής στον Τελικό Πίνακα, στην Σωστή Χρονική Σειρά

Όποτε μια νέα γραμμή δεδομένων φτάνει εδώ τοποθετείται στο κάτω μέρος του δισδιάστατου πίνακα εξόδου και αμέσως ακολουθεί η τακτοποίηση γραμμών σύμφωνα με το χρόνο διεξαγωγής των συμβάντων. Επίσης τα δεδομένα που δημιουργεί η διαδικασία λειτουργούν ως βάση πληροφοριών για το περιβάλλον του συστήματος στο τρέχον κείμενο. Όταν έχουν ολοκληρωθεί όλες οι προτάσεις του κειμένου και σε αμφίδρομη επικοινωνία με την επιλογή αρχείου κειμένου (διαδικασία), δίνει τα δεδομένα στον πραγματικό κόσμο και ταυτόχρονα δίνει την άδεια επιλογής και ανοίγματος νέου αρχείου.

3.3.10 Εμφάνιση Πίνακα Ενεργειών

Αυτό είναι η Έξοδος του συστήματος. Συγκεκριμένα είναι ένας δισδιάστατος πίνακας με τα συμβάντα που αναφέρονται στο κείμενο εισόδου. Αυτός εμφανίζεται στον πραγματικό κόσμο σηματοδοτώντας τις ενέργειες που καλείται να πράξει ή να γνωρίζει το ρομποτικό μας σύστημα.

3.3.11 Αρχικές Τιμές

Αρχικές ή μηδενικές τιμές θα είναι εδώ η προεπιλογή για κενά σημεία στις προτάσεις που όμως εννοούνται από την γραμματική. Διαδικασία που αλληλεπιδρά με τη σύνταξη προσωρινής δομής (διαδικασία), που στην ουσία της θέτει τα κενά προς πλήρωση. Αλληλεπιδρά επίσης με την καλούμενη ως περιβάλλον διαδικασία για να προωθήσει ένα ερώτημα που δεν λύθηκε ή να δεχθεί απάντηση για επιστροφή σε ερώτηση που της έγινε.

3.3.12 Περιβάλλον

Διαδικασία που λαμβάνει πληροφορίες (από την τοποθέτηση γραμμής στον τελικό πίνακα, στην σωστή χρονική σειρά) για το ευρύτερο περιβάλλον διεξαγωγής συμβάντων στο τρέχον κείμενο. Αλληλεπιδρά άμεσα με τη σύνταξη προσωρινής δομής (διαδικασία), καθώς και με τις διαδικασίες αρχικών τιμών και υποβολής ερωτήματος. Σκοπός είναι η διαχείριση ερωτημάτων και υπάρχουσας γνώσης για να πληρωθούν κενά στις γραμμές δεδομένων που υπάρχουν.

3.3.13 Υποβολή Ερώτησης

Διαδικασία που αλληλεπιδρά άμεσα με τη σύνταξη προσωρινής δομής (διαδικασία), καθώς και με τις διαδικασίες που καλείται περιβάλλον. Σκοπός είναι να εξωτερικεύονται ερωτήματα προς τον πραγματικό κόσμο. Τα ερωτήματα αυτά προκύπτουν από κενά που όμως δεν μπορούν να καλυφθούν από τη γραμματική και τα συμφραζόμενα. Η απάντηση επιστρέφει με τον τρόπο που το ερώτημα έφτασε και ακριβώς εκεί που τέθηκε.

3.3.14 Γραμματική

Μία τύπου βάσης δεδομένων φορτωμένη στη μνήμη με όλες τις απαραίτητες γνώσεις γραμματικής της SostiMatiko. Χρησιμοποιείται από τις διαδικασίες της ανάλυσης της λέξης και του προσδιορισμού χώρου – χρόνου.

3.3.15 Λεξιλόγιο

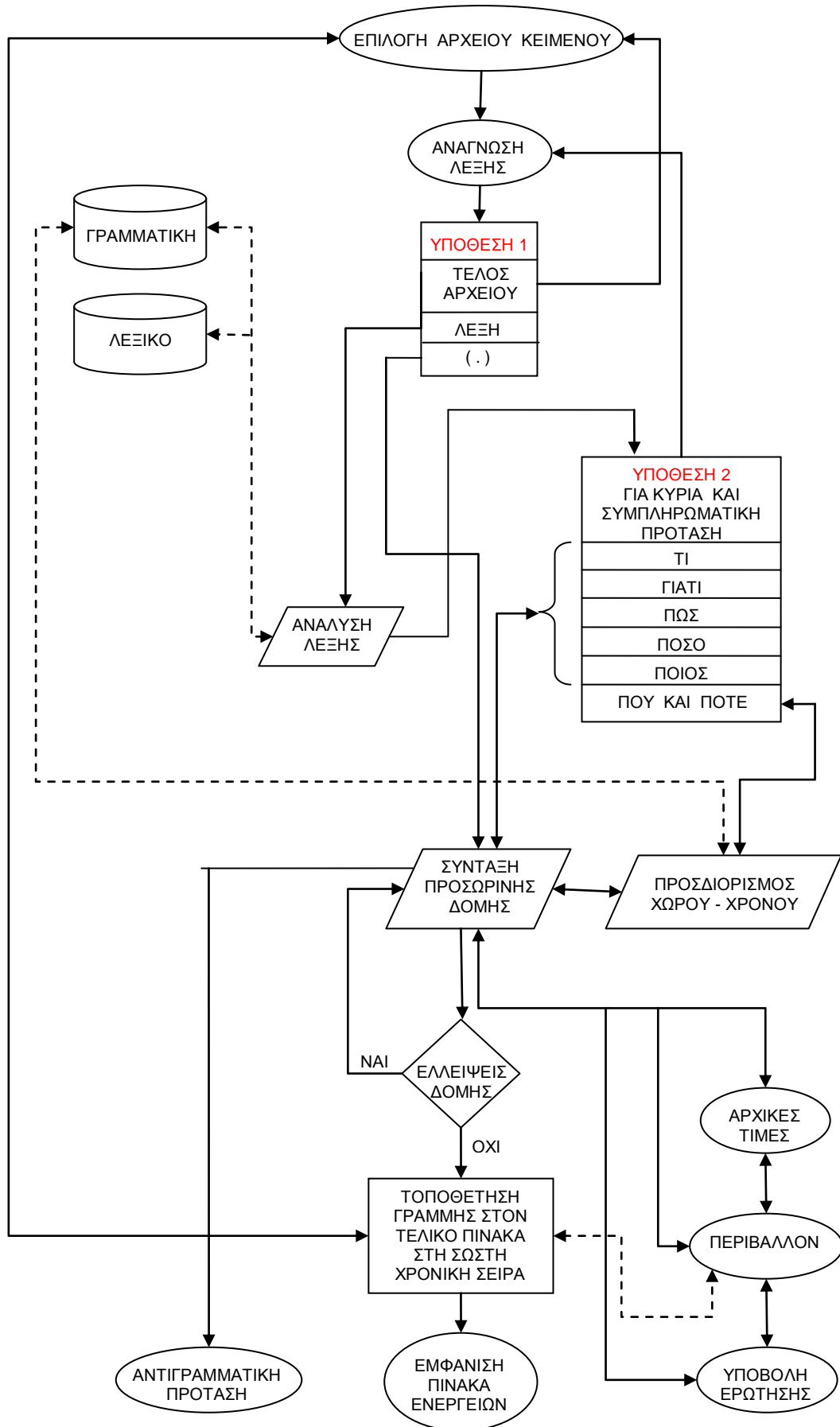
Μία τύπου βάσης δεδομένων φορτωμένη στη μνήμη με όλο το λεξιλόγιο της SostiMatiko. Χρησιμοποιείται από τη διαδικασία της ανάλυσης της λέξης.

3.4 Το Interface του προγράμματος

Το λογισμικό περιλαμβάνει διεπαφή (Interface) η οποία φαίνεται στο παράρτημα Γ.

3.5 Βασικός Αλγόριθμος

Ο βασικός αλγόριθμος λειτουργίας του συστήματος φαίνεται στο σχεδιάγραμμα που ακολουθεί.



ΥΛΟΠΟΙΗΣΗ

Στο σημείο αυτό θα υλοποιήσουμε το λογισμικό του συστήματος. Για το χτίσιμο του κώδικα θα χρησιμοποιήσουμε το ECLIPSE³⁵ το οποίο είναι το πιο δημοφιλές πακέτο ανάπτυξης για Java³⁶ και είναι ελεύθερης λήψης και χρήσης. Η Java λοιπόν θα είναι η γλώσσα στην οποία θα επιχειρήσουμε να γράψουμε το πρόγραμμα του συστήματος. Είναι γλώσσα υψηλού επιπέδου αντικειμενοστραφούς προγραμματισμού και εκτελείτε κάτω από όλα τα υπολογιστικά συστήματα ανεξαρτήτως λειτουργικού.

Ο κώδικας του προγράμματός μας αποδίδεται μέσα από μία σειρά Classes³⁷ οργανωμένα και ομαδοποιημένα σύμφωνα με το είδος της λειτουργίας που καλούνται να υπηρετήσουν. Κάθε Κλάση χωρίζεται σε επιμέρους διαδικασίες που αποτελούν ανεξάρτητες μικρές μονάδες όπου κάθε μία αναλαμβάνει να υλοποιήσει ένα τμήμα λειτουργίας του κώδικα. Κάθε διαδικασία καλεί άλλες και απαντάει σε κλήσεις άλλων αντίστοιχων διαδικασιών, κάτω από τον συνεχή έλεγχο της βασικής διαδικασίας “main”. Η “main” βρίσκεται σε ομώνυμη κλάση και είναι αυτή που διαβάζει ο Compiler³⁸ κατά την εκτέλεση του κώδικα. Εκεί δημιουργούμε το βρόχο εντολών και δια μέσω αυτού, με τη μορφή κλήσεων, ενεργοποιούνται οι διαδικασίες των διαφόρων κλάσεων ώστε να κάνουν αυτό για το οποίο σχεδιάστηκαν. Να επισημάνουμε εδώ πως στο κεντρικό σημείο της “main” βρίσκεται ο βρόχος μέσα στον οποίο υλοποιείται η αναδρομή της κάθε πρότασης.

Αρχικά θα αναπτύξουμε τη διαδικασία “main” για την οποία και έγινε ο λόγος και στη συνέχεια θα παρουσιαστούν με απλά λόγια οι κλάσεις και οι διαδικασίες αυτών. Τέλος θα δείξουμε με ένα απλό παράδειγμα τον τρόπο που ο κώδικας χειρίζεται μια πρόταση, πως αναλύει τις εισερχόμενες λέξεις και πως διαμορφώνει την τελική γραμμή – εντολή προς το ρομπότ.

³⁵ https://en.wikipedia.org/wiki/Eclipse_%28software%29 και <https://eclipse.org/>

³⁶ https://en.wikipedia.org/wiki/Java_%28programming_language%29

³⁷ https://en.wikipedia.org/wiki/Class_%28computer_programming%29

³⁸ <https://en.wikipedia.org/wiki/Compiler>

4.1 Η διαδικασία “main”

<pre> 1 public static void main(String[] args) 2 { 3 int IdNumOfCommandLine = 0; 4 int SentenceBeginIndex = 0; 5 int verbPosition = 0; 6 int Pcount; 7 int PresentSentenceLength; 8 int CommonCounter = 0; 9 int LastSentenceBeginIndex = 0; 10 int i = 0; 11 int wordCounter = 0; 12 int FinalObjectCounter = 0; 13 boolean VerbFound; 14 boolean VerbFlag; 15 boolean AdjConfirm; 16 boolean Error_Statment; 17 boolean Check_Next_Statment; 18 boolean Adv_Statment; 19 boolean Verb_Statment; 20 boolean Particle_Statment; 21 boolean Noun_Statment; 22 boolean Adjective_Statment; 23 boolean EndlessBypassMark; 24 boolean andCouplingDetection = false; 25 boolean IllustrativeFlag; 26 boolean TimeFlag; 27 28 String TextFromFile = SupportFunctions.InFile(fileName); 29 30 String[] inputText = SupportFunctions.tokenize(TextFromFile); 31 32 33 while(i < inputText.length) 34 { 35 if (inputText[i] != ".") 36 { 37 aline = SupportFunctions.ObjectArrayIncrement(aline, 1); 38 aline[wordCounter] = 39 WordAnalysis.analysis(inputText[i],IdNumOfCommandLine); 40 wordCounter++; 50 } 51 else if (inputText[i] == ".") 52 { 53 Pcount = SentenceBeginIndex; 54 VerbFound = false; 55 VerbFlag = false; 56 Error_Statment = false; 57 Check_Next_Statment = false; 58 Adv_Statment = false; 59 Verb_Statment = false; 60 Particle_Statment = false; 61 Noun_Statment = false; 62 Adjective_Statment = true; 63 AdjConfirm = false; 64 EndlessBypassMark = false; 65 IllustrativeFlag = false; 66 TimeFlag = false; 67 do 68 { 69 if (aline[Pcount].Name() == "WRONG") 70 { </pre>	<p>#1. Ο κεντρικός βρόχος του προγράμματος</p> <p>#3-26. Μεταβλητές: Μετρητές βρόχων, δείκτες θέσης περιεχομένου πίνακα, Flags τύπου Boolean.</p> <p>28. Μεταβλητή στην οποία τοποθετείτε όλο το αρχείο κειμένου με μορφή ενός String</p> <p>#30. Πίνακας στον οποίο τοποθετούνται όλες οι λέξεις και τελείες του κειμένου μετά την αποδόμησή του σε αυτά τα επί μέρους τμήματα (tokenizing).</p> <p>#33 - 339. Ο Βασικός βρόχος του κώδικα. Εκτελείται μέχρι να τελειώσει ο πίνακας λέξεων και εσωκλείει όλες τις βασικές διαδικασίες εντός των επόμενων βρόχων όπως θα δούμε αμέσως.</p> <p>#35 - 337. Είναι ένα block IF-THEN-ELSE εντός του Βασικού, και στα όρια αυτού (του βασικού). Εδώ όσο δεν βρίσκεται τελεία που σημαίνει τέλος πρότασης (από τελεία σε τελεία), προχωράει στην ανάλυση και καταχώρηση των λέξεων. Εντός γίνονται οι διαδικασίες όπως ανάλυση των λέξεων, έλεγχος δομών και διαμόρφωση της τελικής λίστας εντολών.</p> <p>#37. Ο πίνακας λέξεων-αντικείμενων αυξάνεται να δεχθεί νέο δεδομένο (λέξη - αντικείμενο).</p> <p>#38. Το δεδομένο λέξη αναλύεται γίνεται αντικείμενο γεμάτο πληροφορίες και τοποθετείται στην επόμενη νέα θέση που έχει δημιουργηθεί.</p> <p>#51. Αν έχει βρεθεί τελεία τότε...</p> <p>#53 - 66 ..διαμορφώνονται αρχικές τιμές σε μεταβλητές και...</p> <p>#67 - 327. ..περνάμε στον επόμενο και κεντρικό DO-WHILE βρόχο, ο οποίος αρχίζει την αναδρομή, λαμβάνοντας λέξη-λέξη όλα τα δεδομένα-αντικείμενα του πίνακα.</p> <p>#69 - 78. Αν η λέξη που ελέγχεται είναι λάθος (βρέθηκε κατά την</p>
--	--

<pre> 71 Error_Statment = true; 72 Check_Next_Statment = false; 73 Adv_Statment = false; 74 Verb_Statment = false; 75 Particle_Statment = false; 76 Adjective_Statment = false; 77 Noun_Statment = false; 78 } 79 if (Error_Statment) 80 { 81 Error_Statment = false; 82 SupportFunctions.ErrorGenerator(3, 4, dNumOfCommandLine); 83 aline = SupportFunctions.ObjectArrayIncrement(aline, 84 -(wordCounter - SentenceBeginIndex)); 85 wordCounter = SentenceBeginIndex; 86 Pcount = wordCounter; 87 } 88 89 if (Check_Next_Statment) 90 { 91 if(aline[Pcount].UsefulInfos() == "Number") 92 { 93 if(Pcount+1 < wordCounter) 94 { 95 if(aline[Pcount+1].UsefulInfos() == "Time") 96 { 97 aline[Pcount].SetQuestionsMark("When"); 98 aline[Pcount+1].SetQuestionsMark("When"); 99 aline[Pcount].SetAbsoluteTime(SupportFunctions.TimeFix (100 Pcount,verbPosition)); 101 102 aline[Pcount+1].SetAbsoluteTime(aline[Pcount].AbsoluteTime()); 103 Pcount = Pcount + 2; 104 } 105 else if(aline[Pcount+1].UsefulInfos() == "CountUnit") 106 { 107 aline[Pcount].SetQuestionsMark("WhereTo"); 108 aline[Pcount+1].SetQuestionsMark("WhereTo"); 109 aline[Pcount].SetUsefulInfos("EndPoint"); 110 aline[Pcount+1].SetQuestionsMark("WhereTo"); 111 Pcount = Pcount + 2; 112 } 113 else 114 { 115 Adjective_Statment = true; 116 Check_Next_Statment = false; 117 } 118 } 119 else {Adjective_Statment = true;} 120 EndlessBypassMark = false; 121 } 122 else if(aline[Pcount].UsefulInfos() == "Distance") 123 { 124 CommonCounter = 125 SentenceConfirmation.DistanceCheck(Pcount,wordCounter); 126 127 if(CommonCounter > Pcount) 128 { 129 Pcount = CommonCounter; 130 } 131 else if(CommonCounter <= Pcount) 132 { 133 Check_Next_Statment = false; 134 Error_Statment = true; 135 } 136 EndlessBypassMark = false; 137 } </pre>	<p>ανάλυση) τότε ενεργοποιείται το επίπεδο λάθους και η πρόταση απορρίπτεται στο σύνολο της.</p> <p>#79 - 87. Είναι το επίπεδο λάθους. Όταν ένα λάθος βρεθεί το πρόγραμμα έρχεται εδώ και αφού διαμορφώσει συνθήκες και μηνύματα βγαίνει από την διαδικασία αναδρομής της τρέχουσας πρότασης.</p> <p>#89 - 211. Είναι το επίπεδο όπου τόπο χρόνο και συνδέσμους. Περιλαμβάνει μία σειρά από διαδοχικά συνδεδεμένα IF-THEN-ELSE και ανοίκει στη διαδικασία αναδρομής. Ποιο συγκεκριμένα:</p> <p>#91 - 121. Αν η λέξη είναι αριθμός τότε αν υπάρχει πληροφορία χρόνου η λέξη παίρνει ταμπέλα πότε. Αν αφορά μονάδα μέτρησης τότε γίνεται που και πιο συγκεκριμένα προορισμός.</p> <p>#121 - 137. Αν η λέξη είναι διάστημα τότε ελέγχεται και διαμορφώνεται, συναρτήσσει και των επόμενων λέξεων, το από έως. Γίνεται δηλαδή που.</p> <p style="text-align: center;">***</p> <p>Να επισημάνουμε πως στα παραπάνω επίπεδα, όπως και σε αυτά που ακολουθούν εντός της αναδρομής υπάρχουν τα Flags (μεταβλητές τύπου boolean) όπου ενεργοποιούν ή απενεργοποιούν την πρόσβαση στα διάφορα αυτά επίπεδα. Ανάλογα το αποτέλεσμα σε κάθε επίπεδο ανοίγει ο δρόμος για το κατάλληλο επόμενο.</p> <p style="text-align: center;">***</p>
--	--

<pre> 138 else if(aline[Pcount].UsefulInfos() == "Coupling") 139 { 140 for(CommonCounter = Pcount; CommonCounter < aline.length-1; 150 CommonCounter++) 151 aline[CommonCounter] = aline[CommonCounter+1]; 152 aline = SupportFunctions.ObjectArrayIncrement(aline, -1); 153 wordCounter--; 154 155 Omas_III_Agreements_Modulation.newFinaleLine(wordCounter, 156 SentenceBeginIndex,FinalObjectCounter); 157 FinalObjectCounter++; 158 VerbFound = false; 159 AdjConfirm = false; 160 Check_Next_Statment = false; 161 if (Pcount < wordCounter) 162 { 163 Adjective_Statment = true; andCouplingDetection = true; 164 } 165 else{andCouplingDetection = false; 166 SupportFunctions.ErrorGenerator(11); 167 } 168 169 IdNumOfCommandLine++; 170 LastSentenceBeginIndex = SentenceBeginIndex; 171 SentenceBeginIndex = Pcount; 172 for(CommonCounter = Pcount; CommonCounter < wordCounter; 173 CommonCounter++) 174 aline[CommonCounter].SetIDnumOfLine 175 (IdNumOfCommandLine); 176 EndlessBypassMark = false; 177 } 178 else if(aline[Pcount].UsefulInfos() == "Illustrative") 179 { 180 for(CommonCounter = Pcount; CommonCounter < aline.length-1; 181 CommonCounter++) 182 aline[CommonCounter] = aline[CommonCounter+1]; 183 aline = SupportFunctions.ObjectArrayIncrement(aline, -1); 184 wordCounter--; 185 VerbFound = false; 186 AdjConfirm = false; 187 Check_Next_Statment = false; 188 if (Pcount < wordCounter) {Adjective_Statment = true;} 189 else{SupportFunctions.ErrorGenerator(13);} 190 191 for(CommonCounter = Pcount; CommonCounter < wordCounter; 192 CommonCounter++) 193 aline[CommonCounter].SetNumOfSentence(aline[Pcount-1]. 194 NumOfSentence ()+1); 195 EndlessBypassMark = false; 196 } 197 else 198 { 199 if(EndlessBypassMark) 200 { 201 SupportFunctions.ErrorGenerator(6, aline[Pcount].Name()); 202 Error_Statment = true; 203 Check_Next_Statment = false; 204 } 205 } 206 else 207 { 208 Adjective_Statment = true; 209 Check_Next_Statment = false; 210 } 211 } 212 213 if (Adv_Statment) 214 { </pre>	<p>#138 – 177. Αν η λέξη είναι ο σύνδεσμος «και» τότε κλείνει την τρέχουσα πρόταση (#155) και αφού ελεγχθεί ότι ακολουθεί νέα (#161-164), διαμορφώνει τις συνθήκες και ανοίγει τον δρόμο για την αναδρομή αυτής (#169-174).</p> <p>#178 – 196. Αν η λέξη είναι ο σύνδεσμος «να» τότε επαναδιαμορφώνει τις συνθήκες και ανοίγει τον δρόμο για να αρχίσει νέα αναδρομή για την πρόταση που ακολουθεί</p> <p>#199 – 205. Αν τίποτα από τα παραπάνω δεν ισχύει και φτάνουμε ως εδώ έχοντας ενεργοποιήσει (κατά την αναδρομή) το Flag ατέρμονης παραμονής, τότε προχωράμε σε ενεργοποίηση λάθους για τερματισμό της αναδρομής</p> <p>#206 – 210. Αν τίποτα από τα παραπάνω δεν ισχύει τότε ενεργοποιούμε την αρχή της αναδρομής, ξεκινώντας από το επίπεδο εύρεσης επιθέτου.</p> <p>#213 – 218. Επίπεδο εύρεσης επιρρήματος.</p>
--	--

<pre> 215 if(SentenceConfirmation.IsAdverb(Pcount)){Pcount++;} 216 Check_Next_Statment = true; 217 Adv_Statment = false; 218 } 219 if (Verb_Statment) 220 { 221 if(SentenceConfirmation.IsVerb(Pcount)) 222 { 223 VerbFound = true; 224 verbPosition = Pcount++; 225 Adv_Statment = true; 226 } 227 else 228 { 229 SupportFunctions.ErrorGenerator(5, aline[Pcount].Name()); 230 Error_Statment = true; 231 } 232 Verb_Statment = false; 233 } 234 if (Particle_Statment) 235 { 236 if(SentenceConfirmation.IsParticle(Pcount)){Pcount++;} 237 Verb_Statment = true; 238 Particle_Statment = false; 239 } 240 250 if (Noun_Statment) 251 { 252 if(aline[Pcount].UsefulInfos() == "Number") 253 if(Pcount+1 < wordCounter) 254 if(aline[Pcount+1].UsefulInfos() == "Time") 255 TimeFlag = true; 256 if (!TimeFlag) 257 { 258 if(SentenceConfirmation.IsNoun(Pcount, VerbFound) 259 { 260 if(AdjConfirm) 261 { 262 aline[Pcount-1].SetKind("Adjective"); 263 } 264 Pcount++; 265 } 266 else if(AdjConfirm) 267 { 268 if(aline[Pcount-1].Kind() == "Adjective" 269 { 270 aline[Pcount-1].SetKind("Adj2Noun"); 271 } 272 } 273 if(VerbFound){EndlessBypassMark = true;} 274 } 275 AdjConfirm = false; 276 Noun_Statment = false; 277 if(!VerbFound){Particle_Statment = true;} 278 else {Check_Next_Statment = true;} 279 } 280 else 281 { 282 if(AdjConfirm) 283 { 284 if(aline[Pcount-1].Kind() == "Adjective") 285 { 286 aline[Pcount-1].SetKind("Adj2Noun"); 287 } 288 } 289 AdjConfirm = false; 290 TimeFlag = false; </pre>	<p>#219 – 250. Επίπεδο εύρεσης ρήματος. Το ρήμα είναι απαραίτητο για ύπαρξη πρότασης για το λόγο αυτό και ενεργοποιούμε ένα Flag κατά την εύρεσή του αλλά και κρατάμε τη θέση του με δείκτη για να ανατρέξουμε σε αυτό όταν χρειαστεί</p> <p>#234 – 239. Επίπεδο εύρεσης μορίου ένδειξης χρόνου. Αν αυτό δεν υφίσταται τότε περνάμε αμέσως στο ρήμα</p> <p>#250 – 294. Επίπεδο εύρεσης ουσιαστικού. Αν(#252-255) εδώ έχω ένδειξη αριθμού και αν ακολουθεί επόμενη λέξη που θα έχει ένδειξη χρόνου τότε το Flag χρόνου που έχουμε ορίσει ενεργοποιεί. Τώρα αν δεν έχω ενεργό το Flag σημαίνει πως έχω αντικείμενο ή υποκείμενο και προχωρώ σε ελέγχους. Αν έχω ρήμα ψάχνω για το ουσιαστικό αντικείμενο, αν δε έχω για το υποκείμενο. Αν υπάρχει επίθετο και εδώ βρω επίθετο τότε χαρακτηρίζεται ως ουσιοστικοποιημένο επίθετο. Αν δεν υπάρχει επίθετο τότε αυτό εδώ είναι. Αν έχω Flag χρόνου και το ρήμα φυσικά υπάρχει τότε έχω έλλειψη αντικειμένου και προχωράμε.</p>
--	--

<pre> 291 Noun_Statment = false; 292 Check_Next_Statment = true; 293 } 294 } 295 if (Adjective_Statment) 296 { 297 if(SentenceConfirmation.IsAdjective(Pcount, VerbFound)) 298 { 299 AdjConfirm = true; 300 Pcount++; 301 } 302 Noun_Statment = true; 303 Adjective_Statment = false; 304 } 305 if((Pcount == wordCounter) && andCouplingDetection) 306 { 307 wordCounter = wordCounter + 308 SentenceConfirmation.CompleteAndCoupling(SentenceBeginInit 309 ,LastSentenceBeginInit,IdNumOfCommandLine); 310 Pcount = wordCounter; 311 andCouplingDetection = false; 312 } 313 if(Pcount == wordCounter && !VerbFound && !VerbFlag) 314 { 315 Error_Statment = true; 316 Check_Next_Statment = false; 317 Adv_Statment = false; 318 Verb_Statment = false; 319 Particle_Statment = false; 320 Noun_Statment = false; 321 Adjective_Statment = false; 322 SupportFunctions.ErrorGenerator(12); 323 Pcount--; 324 VerbFlag = true; 325 } 326 }while(Pcount < wordCounter); 327 328 if(SentenceBeginInit < aline.length) 329 { 330 Omas_III_Agreements_Modulation.newFinaleLine(wordCounter, 331 SentenceBeginInit,FinalObjectCounter); 332 FinalObjectCounter++; 333 } 334 IdNumOfCommandLine++; 335 SentenceBeginInit = wordCounter; 336 } 337 i++; 338 } 339 } 340 </pre>	<p>#296 – 305. Επίπεδο εύρεσης επιθέτου. Αφού βρεθεί προχωράμε στο επίπεδο εύρεσης ουσιαστικού</p> <p>#306 – 313. μέσω τις ισότητας των δύο μετρητών καθορίζεται το τέλος βρόχου της ανάλυσης της πρότασης. Αν αυτό έχει γίνει και υπάρχει σύνδεσμος και ελέγχεται και συμπληρώνεται η πληρότητα της 2^{ης}</p> <p>#306 – 313. Όπως και παραπάνω αν ο βρόχος πρέπει να τερματίσει και δεν έχω ρήμα πρέπει πρώτα να το περάσω από τη διαδικασία επιπέδου λάθους. Το VerbFlag χρησιμοποιείτε για να μπλοκάρει την είσοδο εδώ (εφόσον υπάρξει μία πρώτη) για την αποφυγή ατέρμονου βρόχου</p> <p>#329 – 334. Εφόσον έχουμε επιτυχή αναδρομή τότε διαμορφώνεται η πρόταση αυτή ως γραμμή εντολής.</p>
---	---

4.2 Οι Classes του κώδικα

Εδώ παρουσιάζονται οι κλάσεις του κώδικα και μία επεξήγηση κάθε μίας διαδικασίας που περιλαμβάνουν.

4.2.1 SentenceConfirmation Class

Εδώ υπάρχουν εκείνες οι διαδικασίες που καλούνται να διαπιστώσουν το μέρος του λόγου της λέξης που εξετάζεται. Η κλήση της κατάλληλης διαδικασίας-συνάρτησης γίνεται στοχευμένα σύμφωνα με το τι περιμένουμε να είναι η τρέχουσα λέξη:

Αναλυτικά:

- *public static boolean **IsNoun(int i, boolean ThereIsVerb)*** Συνάρτηση που επιστρέφει “true” ή “false” ανάλογα με το αν η τρέχουσα λέξη είναι Ουσιαστικό ή όχι. Επίσης επεμβαίνει στο αντικείμενο της λέξης για να την χαρακτηρίσει ως υποκείμενο ή αντικείμενο. Δέχεται ως ορίσματα την θέση του αντικειμένου λέξης εντός του πίνακα και το Flag ύπαρξης ή όχι ρήματος στην παρούσα στιγμή.
- *public static boolean **IsAdjective(int i,boolean ThereisVerb)*** Συνάρτηση που επιστρέφει “true” ή “false” ανάλογα με το αν η τρέχουσα λέξη είναι Επίθετο ή όχι. Επίσης επεμβαίνει στο αντικείμενο της λέξης για να την χαρακτηρίσει ως υποκείμενο ή αντικείμενο. Δέχεται ως ορίσματα την θέση του αντικειμένου λέξης εντός του πίνακα και το Flag ύπαρξης ή όχι ρήματος στην παρούσα στιγμή.
- *public static boolean **IsParticle(int i)*** Συνάρτηση που επιστρέφει “true” ή “false” ανάλογα με το αν η τρέχουσα λέξη είναι μόριο για τον προσδιορισμό του χρόνου. Δέχεται ως όρισμα την θέση του αντικειμένου λέξης εντός του πίνακα.
- *public static boolean **IsVerb(int i)*** Συνάρτηση που επιστρέφει “true” αν η τρέχουσα λέξη είναι ρήμα, παθητική φωνή ή προστακτική ή “false” αν δεν είναι τίποτα από όλα αυτά. Δέχεται ως όρισμα την θέση του αντικειμένου λέξης εντός του πίνακα.
- *public static boolean **IsAdverb(int i)*** Συνάρτηση που επιστρέφει “true” ή “false” ανάλογα με το αν η τρέχουσα λέξη είναι επίρρημα ή όχι. Δέχεται ως όρισμα την θέση του αντικειμένου λέξης εντός του πίνακα.
- *public static int **DistanceCheck(int i,int max_i)*** Συνάρτηση που ελέγχει την ορθότητα του διαστήματος «από-έως» εντός της πρότασης. Δέχεται ως ορίσματα την τρέχουσα θέση του αντικειμένου λέξης μέσα στον πίνακα και το σύνολο των λέξεων που έχουν βρεθεί σε αυτή την πρόταση. Επιστρέφει τον αριθμό λέξεων που αποτελείται το διάστημα που ελέγχθηκε.
- *public static int **CompleteAndCoupling(int BeginIndex,int LastBeginIndex,int IDL)*** Συνάρτηση που μεταφέρει τα εννοούμενα στοιχεία της κύριας πρότασης στην 2^η όταν βρεθεί ο σύνδεσμος “και”. Επίσης μεγαλώνει τον πίνακα αντικειμένων λέξης κατά τις θέσεις που θα χρειαστεί για την προσθήκη και επιστρέφει τον αριθμό λέξεων που προστέθηκαν. Δέχεται ως ορίσματα τον δείκτη της αρχής της 2^{ης} πρότασης, τον δείκτη αρχής της κύριας πρότασης και τον τρέχοντα χαρακτηριστικό αριθμό γραμμής (μοναδικός για κάθε πρόταση) για να χαρακτηριστούν τα αντικείμενα της 2^{ης} πρότασης.

4.2.2 WordAnalysis Class

Εδώ βρίσκονται όλες οι σκληρές διαδικασίες του κώδικα. Είναι το τμήμα εκείνο όπου αναλύει τη λέξη αλλά και της προσδίδει τα διάφορα άλλα χαρακτηριστικά εντός ενός αντικειμένου λέξης. Αναλυτικά:

- *private static String[] **Take_A_Prefix (String aWord)*** Διαδικασία όπου λαμβάνοντας ως όρισμα ένα τμήμα λέξης ή ολόκληρη λέξη, μας επιστρέφει ένα πίνακα τιμών που περιλαμβάνει, ένα πρόθημα και τη θέση του στη βάση δεδομένων αν εντόπισε κάποιο. Αν δεν έχει εντοπίσει κάτι ο πίνακας επιστρέφει ως “none”.
- *private static String[] **Take_A_Suffix(String aWord, int _2ndCompoFlag)*** Διαδικασία όπου λαμβάνοντας ως όρισμα ένα τμήμα λέξης ή ολόκληρη λέξη και έναν αριθμητικό δείκτη αν η λέξη αυτή είναι τελευταίο συνθετικό κατάληξης, μας επιστρέφει ένα πίνακα τιμών που περιλαμβάνει, ένα επίθημα και τη θέση του στη βάση δεδομένων αν εντόπισε κάποιο. Αν δεν έχει εντοπίσει κάτι ο πίνακας επιστρέφει ως “none”.

- *public static String[][] FoundRoot(String aWord)* Διαδικασία όπου λαμβάνοντας ως όρισμα μία λέξη μας επιστρέφει έναν δυσδιάστατο πίνακα τιμών που περιλαμβάνει, τόσες σειρές όσες οι ρίζες που εντόπισε στη λέξη καθώς και τμήματα λέξης που προηγούνται ή έπονται της ρίζας σε κάθε γραμμή που θα ελεγχθούν ως προθέματα και επιθήματα αντίστοιχα.
- *public static String[][] FoundPrefix(String[] isPrefix)* Διαδικασία όπου λαμβάνοντας ως όρισμα ένα τμήμα λέξης ή ολόκληρη λέξη μας επιστρέφει ένα πίνακα τιμών δύο διαστάσεων, τόσων γραμμών όσες και οι ρίζες που έχουν εντοπιστεί στη λέξη που ελέγχθηκε στην παραπάνω διαδικασία. Για κάθε γραμμή περιλαμβάνει, για τα προθήματα που ίσως εντόπισε στη βάση δεδομένων, ποσότητα θέση και λέξη ή δεν περιλαμβάνει τίποτα αν δεν έχει εντοπίσει κάτι. Στην περίπτωση που το όρισμα είναι το κενό τότε ο πίνακας επιστρέφεται με 0 στην πρώτη του θέση σε αυτή τη γραμμή για να δείξει πως δεν υπάρχει πρόθημα.
- *public static String[][] FoundSuffix(String[] isSuffix)* Διαδικασία όπου λαμβάνοντας ως όρισμα ένα τμήμα λέξης ή ολόκληρη λέξη μας επιστρέφει ένα πίνακα τιμών δύο διαστάσεων, τόσων γραμμών όσο και οι ρίζες που έχουν εντοπιστεί στη λέξη που ελέγχθηκε στην παραπάνω διαδικασία. Για κάθε γραμμή περιλαμβάνει, για τα επιθήματα που ίσως εντόπισε στη βάση δεδομένων, ποσότητα θέση και λέξη ή δεν περιλαμβάνει τίποτα αν δεν έχει εντοπίσει κάτι.
- *public static DynamicWord analysis(String aWord, int inocl)* Η Διαδικασία που χτίζει το αντικείμενο λέξης. Δέχεται ως ορίσματα την προς έλεγχο λέξη και τον αριθμό γραμμής πρότασης που είναι μοναδικός για κάθε μια (πρόταση). Αφού λάβει τα δεδομένα των παραπάνω αναλύσεων της κλάσης τα επεξεργάζεται και εφόσον η λέξη είναι σωστή και με μοναδικό νόημα προχωράει στην διαδικασία του αντικειμένου λέξης. Βασικά του προσδίδει τα πρώτα βασικά χαρακτηριστικά όπως ρήμα, επίρρημα, αριθμό, έννοια κτλ. Επιστρέφει το αντικείμενο λέξης τύπου DynamicWord που το κατασκευάσαμε για τις δικές μας ανάγκες.

4.2.3 DynamicWord Class

Η συγκεκριμένη κλάση είναι ένας constructor αντικειμένου. Είναι αυτό που αναφέρουμε ως τώρα αντικείμενο λέξης. Είναι μία μοναδική οντότητα για κάθε λέξη που απαρτίζεται από πολλές πληροφορίες για αυτή. Επιγραμματικά:

- *public DynamicWord()* Constructor. Υπάρχει σε τρεις εκδόσεις, με ορίσματα, χωρίς ορίσματα (όπως το δείχνουμε εδώ) και με λίγα ορίσματα. Με την κλήση δημιουργείτε το αντικείμενο λέξη.
- *public void SetName(String aWord)* τοποθετεί το όνομα της λέξης στο αντικείμενο
- *public void SetMeaning(String mean)* τοποθετεί το Ελληνικό νόημα της λέξης
- *public void SetKind(String PoS)* τοποθετεί το μέρος του λόγου της λέξης
- *public void SetTense(String TnS)* τοποθετεί τον χρόνο της λέξης (ρήμα)
- *public void SetPosition(String PtN)* τοποθετεί τη θέση της λέξης στην πρόταση
- *public void SetGrammarNum(String GrmN)* ενικός ή πληθυντικός αριθμός
- *public void SetTimeCategory(String cat)* Σημειώνει Παρελθόν, Παρόν ή Μέλλον
- *public void SetTimeSubCategory(String subCat)* Ορισμένου ή αόριστου χρόνου
- *public void SetAbsoluteTime(long absTime)* Τοποθετεί τον ακριβή χρόνο σε ώρες
- *public void SetQuestionsMark(String question)* Ποιο ερώτημα απαντάει η λέξη;
- *public void SetUsefulInfos(String usfInf)* Τοποθέτηση βοηθητικής πληροφορίας
- *public void SetNumOfSentence(int NoS)* Τοποθέτηση αριθμού υποπρότασης
- *public void SetIDnumOfLine(int IdL)* Τοποθέτηση αριθμού πρότασης
- *public String Name ()* Επιστρέφει το όνομα της λέξης από το αντικείμενο

- *public String Meaning* () Επιστρέφει Ελληνικό νόημα της λέξης
- *public String Kind* () Επιστρέφει το μέρος του λόγου της λέξης
- *public String Tense* () Επιστρέφει το τον χρόνο της λέξης (ρήμα)
- *public String Position* () Επιστρέφει τη θέση της λέξης στην πρόταση
- *public String GrammarNum*() Επιστρέφει τον αριθμό της λέξης
- *public String TimeCategory*() Επιστρέφει Παρελθόν, Παρόν ή Μέλλον
- *public String TimeSubCategory*() Επιστροφή δείκτη ορισμένου ή αόριστου χρόνου
- *public long AbsoluteTime*() Επιστρέφει τον χρόνο σε ώρες
- *public String QuestionsMark*() Επιστρέφει το ερώτημα που απαντά η λέξη
- *public String UsefulInfos*() Επιστρέφει την σημαντική πληροφορία
- *public int NumOfSentence* () Επιστρέφει τον αριθμό της υποπρότασης
- *public int IDnumOfLine* () Επιστρέφει τον αριθμό της πρότασης
- *public void EquateTheID (DynamicWord existObject)* εξισώνει τον αριθμό πρότασης δύο αντικειμένων.

4.2.4 *DynamicLineClass*

Και η συγκεκριμένη κλάση είναι ένας constructor αντικειμένου. Είναι το αντικείμενο της γραμμής εντολών. Είναι μία μοναδική οντότητα για κάθε γραμμή που απαρτίζεται από πολλές πληροφορίες για αυτή. Η κάθε γραμμή είναι και μία πρόταση που έχει βρεθεί σωστή και τοποθετείται εδώ αναλυτικά. Πριν το τελευταίο στάδιο όλα αυτά τα αντικείμενα θα τοποθετηθούν με χρονική σειρά όπως ακριβώς έχουμε ορίσει σε προηγούμενο κεφάλαιο (βλ. 3.1.3). Επιγραμματικά:

- *public DynamicLine*() Δημιουργία του constructor
- *public void Set_Why_Name(String aWord)*
- *public void Set_Why_Meaning(String mean)*
- *public void Set_What_Name(String aWord)*
- *public void Set_What_Meaning(String mean)*
- *public void Set_HowMuch_Name(String aWord)*
- *public void Set_HowMuch_Meaning(String mean)*
- *public void Set_How_Name(String aWord)*
- *public void Set_How_Meaning(String mean)*
- *public void Set_Who_Name(String aWord)*
- *public void Set_Who_Meaning(String mean)*
- *public void Set_Where_from_Name(String aWord)*
- *public void Set_Where_from_Meaning(String mean)*
- *public void Set_Where_to_Name(String aWord)*
- *public void Set_Where_to_Meaning(String mean)*
- *public void Set_When_Name(String aWord)*
- *public void Set_When_Meaning(String mean)*

Χάριν συντομίας αναφέρουμε απλά πως, οι παραπάνω διαδικασίες τοποθετούν στο αντικείμενο τις λέξεις που απαντούν στα ερωτήματα του μοντέλου.

- *public void SetTense(String TnS)* τοποθετείται ο χρόνος του ρήματος
- *public void SetTimeCategory(String cat)* Τοποθετεί τη χρονική κατηγορία (βλ.3.1.3)
- *public void SetTimeSubCategory(String subCat)* Χρονική υποκατηγορία (βλ.3.1.3)
- *public void SetAbsoluteTime(long absTime)* Τοποθέτηση του χρόνου σε ώρες
- *public void SetNumOfSentence(int NoS)* Δηλώνει σε ποια πρόταση ανήκει
- *public void SetIDnumOfLine(int IdL)* Δηλώνεται ο αριθμός γραμμής εντολής

- `public void SetSubSentencePointer(int SSSP)` Δηλώνεται ποια υποπρόταση την ακολουθεί.

Ακολουθούν οι συναρτήσεις του constructor που επιστρέφουν τις αντίστοιχες τιμές όπως δηλώνονται στις παραπάνω διαδικασίες.

- `public String Why_Name()`
- `public String Why_Meaning()`
- `public String What_Name()`
- `public String What_Meaning()`
- `public String HowMuch_Name()`
- `public String HowMuch_Meaning()`
- `public String How_Name()`
- `public String How_Meaning()`
- `public String Who_Name()`
- `public String Who_Meaning()`
- `public String Where_from_Name()`
- `public String Where_from_Meaning()`
- `public String Where_to_Name()`
- `public String Where_to_Meaning()`
- `public String When_Name()`
- `public String When_Meaning()`
- `public String Tense ()`
- `public String TimeCategory()`
- `public String TimeSubCategory()`
- `public long AbsoluteTime()`
- `public int NumOfSentence ()`
- `public int IDnumOfLine ()`
- `public int SubSentencePointer()`

4.2.5 SupportFunctions Class

Η κλάση που έχει όλες τις βοηθητικές αλλά πολύ σημαντικές συναρτήσεις για την υποστήριξη των λειτουργιών της γλώσσας και του μοντέλου.

- `private static Object resizeArray (Object oldArray, int newSize)` Μεταβάλλει έναν πίνακα αντικείμενο στο μέγεθος “newSize”.
- `public static String[][] _2dArrayIncrement (String[][] modArray, int XSizeIncr, int YSizeIncr)` Μεταβάλλει έναν δυσδιάστατο πίνακα “modArray” συμβολοσειράς τόσο όσο ορίζεται από τα ορίσματα “XsizeIncr” και “YsizeIncr”.
- `public static String[] _1dArrayIncrement (String[] modArray, int XSizeIncr)` Μεταβάλλει έναν πίνακα “modArray” συμβολοσειράς τόσο όσο ορίζεται από το όρισμα “XSizeIncr”.
- `public static DynamicWord[] ObjectArrayIncrement (DynamicWord[] modArray, int XSizeIncr)` Μεταβάλλει έναν πίνακα “modArray” αντικειμένων λέξης όσο ορίζεται από το όρισμα “XSizeIncr”.
- `public static DynamicLine[] LineObjectArrayIncrement (DynamicLine[] modArray, int XSizeIncr)` Μεταβάλλει έναν πίνακα “modArray” αντικειμένων γραμμών εντολών όσο ορίζεται από το όρισμα “XSizeIncr”.
- `public static void ErrorGenerator(int ErrorNum)` Εμφανίζει αριθμημένο μήνυμα λάθους
- `public static void ErrorGenerator(int ErrorNum, String aWord)` Εμφανίζει αριθμημένο μήνυμα λάθους για συγκεκριμένη λέξη

- `public static void ErrorGenerator(int ErrorNum, int ErrorNum2, int NumOfLine)` Εμφανίζει συνδυασμό αριθμημένων μηνυμάτων λάθους
- `public static String[] NumberReconstruction(String[] anArray)` Επανατοποθετεί έναν πίνακα που ορίζει έναν αριθμό και τον επιστρέφει ως ενιαία συμβολοσειρά
- `public static boolean isCapitalChar(char aChar)` Επιστρέφει “true” η “false” αν βρεθεί ο χαρακτήρας που λαμβάνει να είναι κεφαλαίο.
- `public static String[] GeneralNamesDetect(String[] anArray)` Δέχεται έναν πίνακα λέξεων και επιστρέφει τον ίδιο πίνακα με τα κύρια ονόματα επανατοποθετημένα και αφού τα έχει κρατήσει στην βάση γνώσεων.
- `public static String[] tokenize(String textString)` Δέχεται μία μεγάλη συμβολοσειρά που απαρτίζεται από πολλές λέξεις και επιστρέφει έναν πίνακα με τις λέξεις αυτές μία-μία σε κάθε του θέση αφού έχει αναγνωρίσει και επανατοποθετήσει τα κύρια ονόματα και τους αριθμούς.
- `public static String FixColor(String aWord)` Αποκαθιστά την σημασιολογική ονομασία των χρωμάτων όπως προκύπτουν από τη SostiMatiko. Δέχεται τη λέξη του χρώματος της SostiMatiko και την επιστρέφει στην Ελληνική.
- `public static long TimeFix(int nameCount, int VerbPos)` Δέχεται τον δείκτη αντικειμένου λέξης που αφορά την πληροφορία χρόνου καθώς και τον δείκτη θέσης αντικειμένου λέξης-ρήμα και επιστρέφει έναν αριθμό που χαρακτηρίζει μοναδικά το ρήμα. Ο αριθμός αυτός συμπεριλαμβάνει όλες τις πληροφορίες που θα κατατάξουν την πρόταση, που περιέχει το ρήμα αυτό, με απλή ταξινόμηση στη σωστή θέση, έναντι και των υπολοίπων.
- `public static String InFile(String FileName)` Δέχεται μία συμβολοσειρά που είναι το όνομα ενός αρχείου κειμένου στη μονάδα δίσκου και επιστρέφει όλα τα δεδομένα σε μία ενιαία συμβολοσειρά.
- `public ShowCommandsData()` Τακτοποιεί χρονολογικά τα αντικείμενα γραμμών με απλή ταξινόμηση, συμπληρώνει ό,τι κενά δομής υπάρχουν ακόμα σε εκκρεμότητα, βάση της ΟΜΑΣ-III και αφού τα καταχωρήσει σε αρχείο τα εμφανίζει στον χρήστη.

4.2.6 Omas_III_Agreements_Modulation Class

Είναι η πιο λιτή κλάση από άποψη διαδικασιών αλλά μία από τις πιο σημαντικές του κώδικα. Εδώ πλέον συγκεντρώνουμε και δημιουργούμε την τελική λίστα εντολών προς το σύστημα. Περιλαμβάνει:

- `public static DynamicLine TempStructConfig(DynamicWord aSentence[])`
Λαμβάνει έναν πίνακα που συμπεριλαμβάνει μία πρόταση και επιστρέφει ένα αντικείμενο γραμμής εντολών γεμάτο με τα δεδομένα που αντλεί από τα επιμέρους αντικείμενα λέξεις. Κάποια κενά που θα υπάρξουν αν υπάρξουν θα καλυφτούν από την τελική χρονολογική ταξινόμηση.
- `public static void newFinaleLine(int wordCount,int SentenceBegin,int FinalCounter)` Απλά είναι η διαδικασία που δημιουργεί μία νέα θέση στον τελικό πίνακα αντικειμένων γραμμής και την γεμίζει σε συνεργασία με την προηγούμενη συνάρτηση.

4.3 Παράδειγμα λειτουργίας του κώδικα

Τις εισροές και εκροές του προγράμματος τις γνωρίζουμε αφού είναι αυτό το interface του, μεταξύ ημών και της μηχανής. Για την επίδειξη του συστήματος υπάρχει ήδη ένα demo πρόγραμμα που δίνεται μαζί με την εργασία, όπως εξάλλου

και ολόκληρο το βασικό πρόγραμμα. Το τι ακριβώς όμως γίνεται στο εσωτερικό του προγράμματος θα το δούμε εδώ και μέσα από απλά βήματα. Στην πραγματικότητα παγιδεύσαμε και αποσπάσαμε δεδομένα κατά την λειτουργία ανάλυσης μίας πρότασης και τα εμφανίζουμε εδώ, σχολιάζοντάς τα κατάλληλα.

Είσοδος του συστήματος

Γράψαμε την εντολή : “ ksana deksa dex duo podos edno joro.” Που σημαίνει: «ξαναπήγαινε 20 πόδια δεξιά σε μία ώρα»

Ολόκληρη η πρόταση θα περάσει από τον **tokenizer** και έτσι θα προκύψει ένας πίνακας λέξεων. Εδώ το σύστημα ανιχνεύει δύο αριθμούς τον dex duo και τον edno. Αυτούς τους επεξεργάζεται και προκύπτει ο dex_duo και ο edno που παραμένει. Τους αριθμούς τους βάζει στην βάση γνώσης του με την έννοιά τους να είναι το 20 και το 1 αντίστοιχα.

Κάθε λέξη περνά διαδοχικά από τον αναλυτή. Ο αναλυτής αρχικά εφαρμόζει την μέθοδο **Longest Matching First**³⁹ για να αποδομήσει τη λέξη στα επιμέρους στοιχεία της. Ας δούμε λοιπόν, τι συμβαίνει σε κάθε μία από αυτές.

A) ψάχνει τη ρίζα στο λεξικό

```
202 ks an apa
203 ksana ap a
214 ksana pa
216 ks a napa
219 ksa n apa
221 k s anapa
```

```
θέση 202 βρέθηκε η ρίζα an
θέση 203 βρέθηκε η ρίζα ap
θέση 214 βρέθηκε η ρίζα pa
θέση 216 βρέθηκε η ρίζα a
θέση 219 βρέθηκε η ρίζα n
θέση 221 βρέθηκε η ρίζα s
```

```
null null null
null null null
1 0 ksana
null null null
null null null
null null null
```

```
το ks δεν είναι πρόθεμα
το ksana δεν είναι πρόθεμα
βρέθηκε ένα πρόθεμα στη θέση 0 το ksana
το ks δεν είναι πρόθεμα
το ksa δεν είναι πρόθεμα
το k δεν είναι πρόθεμα
```

```
null null null null null
1 15 a null null
0 null null null null
null null null null null
null null null null null
null null null null null
```

```
το apa δεν είναι επίθημα
βρέθηκε ένα επίθημα στη θέση 15, το a
δεν χρειάζεται επίθημα αυτή η περίπτωση
το napa δεν είναι επίθημα
το apa δεν είναι επίθημα
το anapa δεν είναι επίθημα
```

σωστή είναι η λέξη που η ρίζα έχει ταυτόχρονα σωστό και το πρόθεμα και το επίθημα ή αν κάποιο δεν υπάρχει ή και τα δύο να μην υπάρχουν. Εδώ λοιπόν μιλάμε για την Τρίτη σειρά ανάλυσης όπου, έχω το πρόθεμα ksana τη ρίζα pa και έλλειψη επιθήματος. Άρα βάση της γραμματικής έχουμε προστακτική του ρήματος ξαναπάω άρα έχουμε το «ξαναπήγαινε».

Ομοίως και για την επόμενη λέξη έχω τη λέξη με ρίζα deks και επίθημα a

39 deks a	0 null null	1 15 a
141 d eks a	null null null	1 15 a
216 deks a	null null null	0 null null
217 d e ksa	null null null	null null null
221 dek s a	null null null	1 15 a

Είναι το επίρρημα «δεξιά».

³⁹ Παπακίτσος 2014, σελ. 36.

Άλλες λέξεις δεν αναλύει αφού είναι συγκεκριμένες και τις λαμβάνει ως έχει. Το podos είναι πόδια (εδώ μονάδα μέτρησης μήκους) και το joro σημαίνει ώρα.

Β) προχωράει στην αναδρομή

Δεν βρίσκει υποκείμενο αλλά αφού υπάρχει προστακτική το υποκείμενο είναι **εσύ!**

Ρήμα βρίσκει το προστακτικό **ξαναπήγαινε!**

Αντικείμενο δεν έχει αλλά δεν το χρειάζεται αφού δεν είναι μεταγωγικό ρήμα

Αφετηρία δεν βρίσκει άρα παίρνει την προηγούμενη θέση που μπορεί να είναι και η αρχική όπως εδώ.

Προορισμός είναι τα 20 πόδια (δεξιά)

Η πράξη θα γίνει σε μία ώρα.

Δεν απαντάμε στο ερώτημα γιατί και το σύστημα δεν έχει εξουσιοδότηση να ρωτήσει!

Γ) Δημιουργεί το αντικείμενο γραμμής

```
Num Of Sentence : 0
Who --->      --->  εσύ
What --->     ksanapa --->  ξανά πάει
How --->     deksa --->   δεξί
HowMuch --->      --->  null
Why --->      --->   null
Where from --->      ---> Αρχική
Where to --->     dex_duo podos --->  20 πόδια
When --->     edno joro --->  1 ώρα
Tense: Present
TimeCategory : 2
TimeSubCategory : 2
AbsoluteTime : 2200001
NumOfSentence : 0
IDnumOfLine : 0
SubSentencePointer : 0
```

Έτσι ολοκληρώνεται η διαδικασία και εξωτερικεύεται το αποτέλεσμα αντλώντας μόνο τα απαραίτητα στοιχεία από τον παραπάνω χάρτη.

4.4 Παραπομπές και URLs αναφορών κεφαλαίου

- **ECLIPSE:**
https://en.wikipedia.org/wiki/Eclipse_%28software%29 και <https://eclipse.org/>
- **Java:**
https://en.wikipedia.org/wiki/Java_%28programming_language%29
- **Classes:**
https://en.wikipedia.org/wiki/Class_%28computer_programming%29
- **Compiler:**
<https://en.wikipedia.org/wiki/Compiler>
- **Long Matching First:** Παπακίτσος Ε.Χ. (2014). **Γλωσσική Τεχνολογία Λογισμικού: II. Πραγμάτωση**. Εκπαιδευτικό Εγχειρίδιο, ISBN: 978-960-93-6235-1. Εθνική Βιβλιοθήκη της Ελλάδος, Αθήνα.

ΣΥΜΠΕΡΑΣΜΑΤΑ - ΠΑΡΑΤΗΡΗΣΕΙΣ

Αν προσπαθήσουμε να αναλύσουμε απλά το θέμα αλληλεπίδρασης μεταξύ ενός ανθρώπου και ενός Robot, θα λέγαμε πως το πιο δύσκολο κομμάτι είναι αυτό της επικοινωνίας, λόγο έλλειψης ευφυΐας από τη μηχανή. Αν λύσουμε όμως αυτό το θέμα έχοντας ως δεδομένο συγκεκριμένες προγραμματισμένες διαδικασίες και μόνο, που θα κάνει η μηχανή, τότε μπορούμε να δημιουργήσουμε Robots με χιλιάδες λειτουργίες και να μεταθέσουμε το βάρος της δυσκολίας στο ηλεκτρομηχανικό τους κομμάτι. Έτσι θαυμάσια επιτεύγματα έχουν προκύψει και μας υπηρετούν καθημερινά στη βιομηχανία στην ιατρική στην επιστήμη κ.ο.κ. Θαυμάσια επιτεύγματα που ενεργούν και σε διαστάσεις νανομέτρων σαν κάτι απλό και φυσικό. Θαυμάσια επιτεύγματα που κινούνται και ενεργούν για παράδειγμα, ως ξεναγοί, ως τηλεφωνητές και αλλού. Η παρούσα εργασία επιχείρησε να εξερευνήσει μία άλλη πλευρά ενός τέτοιου συστήματος: Την πλευρά της πραγματικής επικοινωνίας και απόκτησης γνώσης μίας μηχανής. Επιχειρήθηκε να στηθεί μία ιδέα πάνω στην οποία μπορεί να αναπτυχθεί κάτι τέτοιο. Τα εργαλεία μας ήταν ένας μέσος υπολογιστής, ένας java compiler, η τεχνητή γλώσσα SostiMatiko και το σημασιολογικό μοντέλο ΟΜΑΣ-III. Η SostiMatiko μας εξασφαλίζει την απλότητα της επικοινωνίας ενώ πρωτεύοντα ρόλο εδώ έπαιξε το μοντέλο της ΟΜΑΣ-III, γιατί είναι αυτό που μας ανοίγει το δρόμο τόσο στην πληρότητα της πληροφορίας, στη απόκτηση γνώσης και στη διαχείρισή της, όσο και στην αμφίδρομη επικοινωνία, που όμως δεν υλοποιήθηκε στην παρούσα εργασία. Ακριβώς έτσι υλοποιήθηκε και παρουσιάζεται το κύριο μέρος του κεφαλαίου 5.

5.1 Για τη SostiMatiko

Μία σειρά από θετικά στοιχεία μας οδήγησαν στην επιλογή της παρούσης γλώσσας και αφού σε κάποιο βαθμό τα επιβεβαιώσαμε τα παραθέτουμε στην παράγραφο 5.1.1 που ακολουθεί. Μια σειρά όμως και από προβληματισμούς - δυσκολίες που αντιμετωπίσαμε αναφέρονται στην παράγραφο 5.1.2.

5.1.1 *Επισημάνσεις στα θετικά της στοιχεία επί του συστήματος*

Χάριν στη μινημαλιστική της δομής, αποτελεί ένα δυνατό μέσω ομιλίας για αλλόγλωσσους, αφού όπως είδαμε είναι στημένη πάνω στις ιδιαιτερότητες διαφόρων γλωσσών και πάνω σε Ελληνικής καταγωγής ριζών λέξεις.

Στην προσπάθεια αφομοίωσης των ελάχιστων γνώσεων για να μπορέσουμε να χτίσουμε τους κώδικες του προγράμματος, διαπιστώσαμε πως, η διαδικασία προγραμματισμού αποτελούσε ταυτόχρονα το βασικό εργαλείο της δικής μας εκπαίδευσης. Άρα το σύστημα μαθαίνει τη γλώσσα με τον ίδιο τρόπο που τη μαθαίνουμε και εμείς. Αν αυτό το αντιστρέψουμε και το επανατοποθετήσουμε θα λέγαμε πως, ο τρόπος που εκπαιδευόμαστε πάνω στην SostiMatiko είναι ο κώδικας που απαιτείται για να λειτουργήσει το σύστημα πάνω στην ίδια γλώσσα.

Υπάρχει η δυνατότητα εισαγωγής κύριων Ονομάτων, τα οποία τα διδάσκουμε στο σύστημα και ταυτόχρονα τα πλαισιώνουμε με πληροφορίες που αποτελούν, στοιχεία γενικής γνώσης επί των προσώπων και τόπων που γνωρίζει.

Μας δίνεται η δυνατότητα να επιλέξουμε τη συντακτική δομή των προτάσεων. Αυτό ωφελεί τον ομιλητή στο να είναι στη δική του ευχέρεια με πιο τρόπο και πόσο σύνθετα θα μιλήσει. Η επιλογή συντακτικής δομής όμως ωφελεί και το δικό μας σύστημα ώστε να επιλέξουμε τα μέγιστα δυνατά στοιχεία με την ελάχιστη πολυπλοκότητα. Έτσι καταφέραμε να απαλλαγούμε από άρθρα, δείκτες γραμματικών στοιχείων και μία πληθώρα άλλων μικρών λέξεων (που μπορούμε να τις ενσωματώσουμε σε μεγαλύτερες).

Μιλώντας τη SostiMatiko και απευθυνόμενος προς το σύστημα δεν περιορίζεται σε μία σειρά εντολών, τυποποιημένων ή μη. Το ίδιο το σύστημα αποδομεί και αναλύει όλα τα δεδομένα και στη συνέχεια προβαίνει σε ενέργειες (όπως χρονική τακτοποίηση) και υποβολή ερωτήσεων. Τέλος τα δεδομένα καθοδηγούν την ενεργοποίηση των αντίστοιχων δεικτών ώστε ο συνδυασμός όλων να βγάζει νόημα και να πραγματώνεται το ζητούμενο δίνοντας την αίσθηση συνομιλίας. Αρκεί λοιπόν να μεταφέρουμε την πληροφορία στο Ρομπότ με όποια σειρά προκύψει και αυτό θα αξιοποιήσει την πληροφορία είτε ως γνώση είτε ως εντολή εκτέλεσης και γνώση.

Όλες οι λέξεις που είναι καταχωρημένες στο λεξικό της μπορούν να υπάρξουν ως ουσιαστικά, επίθετα, επιρρήματα, ρήματα, σε παθητική φωνή, σε προστακτική. Κάποια, μπορούν να υπάρξουν ως μόρια, ως σύνδεσμοι κτλ. Όλες οι μεταβολές των λέξεων στα συγκεκριμένα μέρη του λόγου γίνονται με συγκεκριμένη κατάληξη και ίδια για όλες τις περιπτώσεις. Επίσης, λέξεις, προθέματα και καταλήξεις συνδυάζονται χωρίς περιορισμό και επαυξάνουν το νόημα της λέξης ως άθροισμα των επιμέρους νοημάτων κάτι που στο σύστημα δεν δημιουργεί πρόβλημα.

Από σχεδιασμού η γλώσσα δεν δημιουργεί ασάφειες αλλά ακόμα και αν κάποιος δίνει δεύτερη ερμηνεία στη λέξη αρκεί να ακολουθήσει τη βασική της μορφή και η ερμηνεία θα αποκατασταθεί!

5.1.2 Προβληματισμοί και Δυσκολίες εφαρμογής της γλώσσας επί του συστήματος

Μία γλώσσα εν γένει, ακόμα και τεχνητή, δεν μπορεί να έχει αποτέλεσμα σε ένα ρομπότ, αντίστοιχο με αυτό που εμφανίζει εφαρμοζόμενη από έναν άνθρωπο. Ως γνωστό ένα ρομπότ δεν έχει συναισθήματα, δεν μπορεί να τα εκφράσει και σαφώς δεν μπορεί να τα αντιληφθεί τόσο με το μάτι όσο και με τις εκφράσεις του σώματος. Αν και η τεχνολογία έχει φτάσει σε πολύ ακραία σημεία της εξέλιξης, δεν μπορούμε να πούμε με βεβαιότητα πως αυτόνομα ένα Ρομπότ θα κάνει αυτό που εμείς περιμένουμε στην πιο μικρή μονάδα του χρόνου συμπεριφοράς του. Μία γλώσσα εκτός από τους ρητούς κανόνες, τους ιδιωτισμούς κ.α. συμπληρώνεται και από οπτικά ερεθίσματα, αφή και αντίληψη. Κάθε προσπάθεια λοιπόν στο να φτιάξουμε έναν ομιλούντα ηλεκτρονικό εγκέφαλο δεν είναι παρά διαδικασίες μίμησης, γνωστών από τη δική μας φύση, συμπεριφορών. Η SostiMatiko επιλέχθηκε και για την πληρότητα της στο να εκφράσουμε τα πάντα, ενώ την κρατάμε στην πιο μινιμαλιστική μορφή της.

Το σύστημα προγραμματίζεται στο να λαμβάνει και να αναλύει μία μόνο δομή τη φορά, την οποία και θα πρέπει να γνωρίζουν όσοι αλληλεπιδρούν με αυτό. Συγκεκριμένα έχουμε επιλέξει την απλή μορφή SVO και AN. Τα επίθετα και οι προσδιορισμοί προηγούνται του ουσιαστικού όμως όλες οι λέξεις μπορεί να είναι τόσο επίθετα όσο και ουσιαστικά. Μπορούμε να κάνουμε μια διαλογή εμείς και να χαρακτηρίσουμε κάποιες λέξεις να είναι επίθετα ως προεπιλογή, αλλά και πάλι πιο πολύ θα βασιστούμε στη σειρά που εισέρχονται μέσα στο σύστημα οι λέξεις. Πάντα

θα πρέπει να γνωρίσουμε τη θέση του ρήματος ώστε και σε συνδυασμό με αυτό να αντιλαμβανόμαστε αν υπάρχει επίθετο ή όχι πριν. Το Επίρρημα είναι χαρακτηριστικό και το αντιλαμβανόμαστε τόσο εύκολα όσο και το ρήμα ενώ μόρια που δηλώνουν χρόνο και διάστημα μετακίνησης (θα ,από, έως κτλ) πρέπει να διαβάζονται ως έχουν και να μην υποστούν καμία ανάλυση. Συνδυάζοντας τα όλα έχουμε τη μέγιστη δυνατή εικόνα για την δοθείσα περίπτωση. Στην περίπτωση που θα θέλαμε να συμπεριλάβουμε και άλλες συντακτικές δομές τότε μία ιδέα είναι πρώτον να δοκιμάζουμε αναδρομικά όλες τις πιθανές περιπτώσεις. Δεύτερο το ίδιο το σύστημα θα κάνει και μία ανάλυση χωρίς να ακολουθεί κάποια συγκεκριμένη δομή. Όταν όλα αυτά τοποθετηθούν το ένα πάνω στο άλλο υποθέτουμε πως θα ταυτίζονται κάποια αποτελέσματα. Κάποια δομή θα εμφανίζει πιο πολλά κοινά με τις υπόλοιπες και αυτή θα είναι επικρατέστερη ως βασική στην παρούσα συζήτηση. Πάντα υπάρχει η περίπτωση να βρεθούν άνω της μίας επικρατέστερη δομή πράγμα που θα μας οδηγήσει σε αμφισημία. Παρ' όλα αυτά μετά από μερικούς ελέγχους το σύστημα εικάζουμε πως θα μπορεί να αυτοπροσδιορίζεται. Ένα τέτοιο στοιχείο γνώσης μπορεί να καταχωρηθεί ως χαρακτηριστικό του συγκεκριμένου ομιλητή. Αυτό είναι μία ιδέα που χρήζει έρευνας πέραν των πλαισίων της παρούσης εργασίας.

Μία άλλη δυσκολία είναι ο χαρακτηρισμός των ρημάτων ως μεταβατικά ή μη και ως μονόπρωτα και δίπρωτα. Αυτοί οι χαρακτηρισμοί απαιτούνται στην καθομιλουμένη, φυσική μας γλώσσα και αυτόματα ως φυσικοί εγκέφαλοι αντιλαμβανόμαστε την πληρότητα και την ορθότητα των προτάσεων που λαμβάνουμε. Και στην περίπτωση αυτή η SostiMatiko χάριν απλότητας δεν χρειάζεται να κάνει έναν τέτοιο διαχωρισμό. Έτσι οδηγηθήκαμε στο να βάλουμε εμείς μία υποσημείωση χαρακτηρίζοντας τα, ώστε να ελέγχουμε και να μην ζητάμε καθόλου αντικείμενο, ή να ζητάμε ένα ή δύο από αυτά μέσα στην πρόταση που εισέρχεται. Μία αστοχία εδώ μπορεί να μας οδηγήσει σε περιττές και άτοπες ερωτήσεις και ίσως λανθασμένη εισαγωγή γνώσεων. Ένα σύστημα το οποίο θα εκπαιδεύεται μπορεί από μόνο του να περνάει σε τέτοιου είδους συμπεράσματα και να αναπροσδιορίζει με πολλαπλές προσπάθειες τέτοιου είδους γνώσεις!

Με ένα ακόμα χαρακτηριστικό σημειώνουμε τις λέξεις εκείνες που ως ρήματα μας δίνουν κίνηση εφόσον δεν ομαδοποιούνται από την ίδια τη γλώσσα. Ένα τέτοιο ρήμα απαιτεί πάντα τόπο εκκίνησης και τόπο προορισμού. Ένα ρήμα που δεν έχει τέτοιο χαρακτήρα μιλάει για ενέργεια που γίνεται εδώ ή τέλος πάντων σε αμετάβλητο τόπο, για παράδειγμα το μιλάω. Και με την παθητική φωνή ο τόπος είναι αμετάβλητος. Να σημειώσουμε εδώ πως όταν η εκκίνηση απαιτείται και δεν δίνεται, το σύστημα θα αντλεί σαν αρχή το τέλος της χρονικά προηγούμενης μετακίνησης και πιο συγκεκριμένα τον προορισμό αυτής. Ο προορισμός μπορεί να απουσιάζει σε ρήματα που δηλώνουν μετακίνηση μόνο όταν δίνεται τρόπος και ποσότητα βηματισμού, δηλαδή δεξιά 10 πόδια.

Ως μονάδα μέτρησης μήκους έχουμε μόνο το πόδι που σαν λέξη μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει τρία πράγματα όπως και στη φυσική γλώσσα. Τα τρία αυτά πράγματα είναι το πόδι ως κυριολεκτική έννοια, το πόδι ως μέτρο μουσικής (τέμπο) και το πόδι ως μονάδα μέτρησης μήκους.

Σύνδεσμοι και συμπληρωματικές – αναφορικές προτάσεις είναι απαραίτητα στοιχεία των φυσικών γλωσσών. Και η SostiMatiko δίνει μία σειρά από συνδέσμους ώστε να αναπτυχθούν αναφορικές προτάσεις. Στην εργασία μας συμπεριλάβαμε μόνο δύο παραδείγματα όμως, γιατί εδώ ανοίγεται ένα άλλο τεράστιο κεφάλαιο που από μόνο του απαιτεί ένα ολοκληρωμένο λογισμικό και σίγουρα ξεφεύγουμε από τα πλαίσια της παρούσας εργασίας. Το σημασιολογικό μοντέλο απαιτεί συμπληρωματικές προτάσεις επεξήγησης αλλά δεν του επιτρέπουμε εδώ την

απαίτηση αυτή. Συγκεκριμένα το μοντέλο ζητάει το γιατί, όμως μία μηχανή δεν πρέπει να γνωρίζει και σαφώς δεν πρέπει να απαιτεί και να έχει πρόσβαση σε όλες της πληροφορίες επεξήγησης. Έτσι το δικό μας ρομπότ παίρνει επεξηγήσεις μόνο όταν αυτές του δίνονται χωρίς απαίτηση. Πρέπει λοιπόν να είναι σε θέση να αναγνωρίζει πότε ξεκινάει μία τέτοια πρόταση και αυτό γίνεται εδώ με το “na” (να). Σε μία τέτοια συνάντηση κινούνται εκ νέου οι διαδικασίες αναδρομής και ελέγχου για να εντοπιστούν τα δικά της γραμματικά στοιχεία. Το “kas” (και) είναι ο άλλος σύνδεσμος που ενσωματώνουμε στο σύστημα και πιο συγκεκριμένα στην περίπτωση που ζητάμε να γίνει «αυτό» και «εκείνο». Εδώ έχουμε κοινό υποκείμενο και ενδέχεται και κοινό αντικείμενο. Για παράδειγμα μπορούμε να πούμε «εσύ πάρε αυτό και πέταξέ το (σε απλά Ελληνικά). Το σύστημα θα λάβει τα κοινά στοιχεία που είναι το υποκείμενο (εσύ) και αντικείμενο (αυτό) και θα σπάσει την πρόταση σε δύο. Από το σημείο αυτό η κάθε πρόταση θα ακολουθήσει τη δική της πορεία εφόσον δεν είναι απαραίτητο να αναφερόμαστε σε διαδοχικά χρονικά συμβάντα.

5.1.3 Συμπέρασμα

Το εγχείρημα χρήσης της SostiMatiko για την επικοινωνία με μία μηχανή πέτυχε και με το παραπάνω. Δώσαμε περισσότερη νοημοσύνη στο σύστημα από ότι αν χρησιμοποιούσαμε απλές άκαμπτες λέξεις για γραμμές εντολών υπό αυστηρά πλαίσια και μόνο. Το ρομπότ μας έστω και στην πιο μινιμαλιστική μορφή της γλώσσας είναι σε θέση να μαθαίνει και να συγκεντρώνει πληροφορίες για πρόσωπα που γνωρίζει.

5.2 Για την OMAS-III

Το σημασιολογικό μοντέλο της OMAS-III επιχειρήθηκε για πρώτη φορά να υλοποιηθεί γραμματικό φορμαλισμό. Στην αρχή της εργασίας είχαμε σαν πεποίθηση την επιτυχή έκβαση ενός τέτοιου εγχειρήματος αλλά τελικά φαίνεται πως μπορούμε να πάρουμε πολλά περισσότερα των προσδοκιών. Συγκεκριμένα θέλαμε την σημασιολογική επαύξηση της μηχανής και βλέπουμε πως αυτό μπορεί να αποκτήσει και γνώσεις. Όπως προαναφέρθηκε (στην αρχή του κεφαλαίου 5) προγραμματίζοντας το σύστημα πάνω στη δομή του μοντέλου ταυτόχρονα με το σύστημα δομούμε και τον δικό μας τρόπο σκέψης επάνω στη SostiMatiko. Ο λόγος που συμβαίνει αυτό είναι γιατί το μοντέλο κτίστηκε πάνω σε επτά μοναδικά θεμελιώδη ερωτήματα (βλ.2.2.1) ίδια με εκείνα που έχουμε και οι άνθρωποι στην καθημερινότητά μας. Όταν κάποιος γνωρίζει τι πρέπει να γίνει, από ποιον, με ποιο τρόπο, με τί διάρκεια και ένταση, σε ποιο τόπο ή σε ποια διαδρομή και πότε, τότε απλά γνωρίζει καλά. Αν αυτές τις γνώσεις τις αποθηκεύσει και τις συνδυάσει με πρόσωπα και τόπους τότε κάνει και ένα μεγάλο βήμα προς τη γνώση. Όταν τις ίδιες γνώσεις τις αντλεί και τις διαχειρίζεται κατά το δοκούν τότε λέμε πως δείχνει ευφυΐα. Αυτό ακριβώς δημιουργεί το μοντέλο της OMAS-III στο ρομπότ και εδώ ακριβώς είναι που γίνεται η ταύτιση.

Το πλαίσιο μίας διπλωματικής εργασίας δεν βοηθάει τόσο στο να γίνει βαθιά έρευνα και υλοποίηση ενός τέτοιου συστήματος, είναι όμως ιδανικό για να μπουν οι βάσεις για κάτι τέτοιο. Ας το δούμε αυτό πιο συγκεκριμένα για κάθε ένα από τα επτά ερωτήματα σε εντελώς πρακτική βάση πια και όχι στη θεωρητική που αναφερθήκαμε στα προηγούμενα κεφάλαια.

5.2.1 Απάντηση για το Ερώτημα Τί

Το ρήμα της κάθε πρότασης. Η πράξη που θέλουμε να γίνει ή έγινε ή γίνεται. Μιλάμε για κάθε πράξη είτε σε ενεργητική είτε σε παθητική είτε στην προστακτική. Σε κάθε περίπτωση είναι το συμβάν. Δεν μπορεί να σταθεί πρόταση χωρίς ρήμα, απορρίπτεται άμεσα με τη διαπίστωση. Οι μονολεκτικές κραυγές, οι φθόγγοι και ο θαυμασμός δεν μπορούν να αποτελούν στοιχείο προσωπικότητας ενός ρομπότ αφού στερείται αισθήσεων και συναισθημάτων.

5.2.2 Απάντηση για το ερώτημα Ποιός

Το υποκείμενο το οποίο μπορεί να υπάρχει μπορεί και όχι. Σε κάθε περίπτωση όταν έχω ρήμα, έχω πρόταση (βλ. 5.2.1). Με αυτό το σκεπτικό όταν απουσιάζει το υποκείμενο, πρέπει να το βρει. Αν αυτό λοιπόν δεν εννοείται τότε το ρομπότ το αναζητάει ερωτώντας τους συνομιλητές του, κάτι που θα κάναμε και εμείς. Αυτό που κάνουμε επιπλέον στη διαδικασία αναγνώρισης του υποκειμένου είναι να καταχωρούμε σε μία βάση με το όνομα Knowledge πληροφορίες σχετικά με αυτό. Στην προκειμένη περίπτωση κρατάμε το σημείο του χώρου που βρίσκεται ή που προορίζεται να βρεθεί σε συγκεκριμένο χρόνο. Αυτό δίνει γνώση στο σύστημα!

5.2.3 Απάντηση για το ερώτημα Πώς

Τροπικό επίρρημα είναι η απάντηση εδώ. Στη SostiMatiko όλα τα επίρρηματα είναι τροπικά, συνεπώς όποιο επίρρημα βρίσκεται μετά το ρήμα είναι αυτό που μας δίνει την απάντηση. Με ποιο τρόπο λοιπόν να γίνει αυτό που ζητείται ή περιγράφεται; Ένα εύλογο ερώτημα που θα κάναμε και εμείς σε κάθε συζήτηση. Αν ο τρόπος δεν δίνεται τότε το ρομπότ τον αναζητάει ερωτώντας τους συνομιλητές του.

5.2.4 Απάντηση στο ερώτημα Πόσο

Είναι το σημείο που αποθηκεύουμε το ή τα αντικείμενα ανάλογα με το πόσα είναι αυτά που πρέπει να έχει η κάθε πρόταση. Τοποθετούνται σημασιολογικά στη θέση εκείνη παρ' ότι μπορεί να μην έχουν καμία σχέση με κάποια ποσότητα. Τα αντικείμενα ζητούνται ανάλογα με το πόσα θέλει το ρήμα να έχει. Και εδώ θα μπορούσε το ρομπότ να συλλέξει γνώση (όπως κάνουμε και στην δική μας πραγματικότητα) για αντικείμενα που ανήκουν οι μεταβιβάζονται για παράδειγμα, αλλά στην παρούσα εργασία δεν θα προχωρήσουμε σε αυτό!

5.2.5 Απάντηση στο ερώτημα Πότε

Η ενσωμάτωση ενός RTC (real time clock) σε ένα ρομποτικό σύστημα και ένα πλήρες λογισμικό χειρισμού χρόνου σε όλο το φάσμα αυτού, θα το έκανε να βιώνει εικονικά την κάθε στιγμή. Το ίδιο σύστημα θα απαιτούσε απόλυτες τιμές χρόνου πιο συχνά από ότι το φτιάξαμε να κάνει. Για τις παρούσες ανάγκες ο χρόνος ελέγχεται και δίνεται σε βάση μίας ώρας. Αν δεν δηλώνεται ρητά από τα εισερχόμενα δεδομένα τον προσεγγίζει από το ρήμα όσο καλύτερα μπορεί. Αυτό είναι στοιχείο ευφυΐας.

5.2.6 Απάντηση στο ερώτημα Πού

Το τελευταίο από τα επτά απαιτητά ερωτήματα. Πρέπει να δηλώνεται η αρχή και το τέλος κάποιας απόστασης. Αν από το ρήμα το σύστημα συμπεραίνει πως πρόκειται για στατικό σημείο τότε η αρχή και το τέλος είναι το ίδιο. Αν τώρα δεν δοθεί η αφετηρία, το σύστημα θα πάρει τον προορισμό της προηγούμενης χρονικά ενέργειας. Αν δεν του δώσουμε σημεία ούτε στοιχεία κίνησης αλλά του ζητήσουμε να επανατοποθετηθεί εκεί που βρίσκεται κάποιος άλλος, τότε αν αυτόν τον κάποιον έχει ήδη γνωρίσει τότε έχει και την αντίστοιχη πληροφορία και θα την χρησιμοποιήσει. Και τα δύο αυτά στοιχεία δείχνουν ευφυΐα!

5.2.7 Απάντηση στο ερώτημα Γιατί

Η απάντηση είναι συμπληρωματική πρόταση με επεξηγηματικό – αιτιολογικό σύνδεσμο. Δεν δίνεται πάντα γιατί δεν πρέπει να απαιτείται η γνώση αυτή από το σύστημα. Το σύστημα πρέπει να είναι σε θέση να την αναγνωρίσει και να τη δεχτεί εφόσον του δοθεί και μόνο.

5.2.8 Συμπέρασμα

Το βασικό συμπέρασμα εδώ είναι πως το συστημικό μοντέλο της ΟΜΑΣ-III είναι η βασική δομή για την ανάπτυξη μίας εφαρμογής που μπορεί να μαθαίνει, να υποβάλει ερωτήματα και να δίνει πληροφορίες σύμφωνα με τις γνώσεις που έχει αποκτήσει. Κρίνεται επαρκές για το στήσιμο ενός ρομποτικού συστήματος ώστε αυτό να φεύγει από τα πλαίσια των ρητών εντολών και μόνο!

5.3 Προβλήματα επεξεργασίας

Κατά την επεξεργασία παρουσιάστηκαν κάποια προβλήματα γενικής φύσεως όπως εμπλοκή σε ατέρμονες διαδικασίες κατά την αναδρομή σε λανθασμένες συντακτικά προτάσεις, αλλά και προβλήματα βαρύνουσας σημασίας για το όλο σύστημα. Τα μεν πρώτα τα διορθώσαμε με διάφορες συνθήκες και Flags τα δεύτερα όμως δεν μπόρεσαν να ξεπεραστούν εξ ολοκλήρου. Τα προβλήματα αυτά τα συναντήσαμε σε τρεις διαφορετικές βασικές κατηγορίες. Πιο συγκεκριμένα είναι:

5.3.1 Προβλήματα μορφολογίας

Ένα βασικό πρόβλημα βρέθηκε στην προσπάθειά μας να χρησιμοποιήσουμε ενθήματα. Τα ενθήματα παρεμβάλλονται εντός της λέξης και εκτός από την ιδιόμορφη σύνθεσή τους και την πολυπλοκότητά τους (λόγω πολλών συνδυασμών) κόβεται η λέξη στα δύο. Θα πρέπει να ελέγχουμε κανόνες ακολουθίας φωνηέντων και συμφώνων και εκεί να ψάχνουμε μία σειρά πολλών άλλων συνδυασμών. Δεν δώσαμε λύση στο πρόβλημα απλά το προσπεράσαμε αφού η SostiMatiko δεν το απαιτεί για να υπάρχει, αν και παρέχει τέτοια δυνατότητα. Επίσης, το πλήθος των πιθανών προσφυμάτων που μπορεί να αναζητηθούν σε μία λέξη θεωρήθηκε εξαιρετικά περιορισμένο σε σχέση με τις δυνατότητες της γλώσσας.

5.3.2 Προβλήματα σύνταξης

Υπάρχουν πολλοί τρόποι να εκφραστεί μία πρόταση συντακτικά αλλά μόνο ένας για να μπορέσουμε να την αντιληφτούμε σωστά. Μία σειρά κανόνων ανάλογα την

περίσταση πρέπει να τρέχουν για να εκφράζεται καλύτερα και χωρίς να μπερδεύονται τα μέρη του λόγου. Η SostiMatiko έχει ένα λεξικό όπου κάθε μία από τις λέξεις μπορεί να εκφράσει όλα τα μέρη του λόγου. Αυτό δημιουργεί μία σύγχυση στο σύστημα όπου θα πρέπει να χρησιμοποιούνται πολλοί δείκτες για να μην τα μπερδεύει. Το πρόβλημα το ξεπεράσαμε ακολουθώντας αυστηρή σειρά δεδομένων στην σύνταξη της πρότασης SVO και NA σε συνδυασμό με δείκτες για τον ορισμό του τόπου και τον ορισμό του χρόνου.

5.3.3 Σημασιολογικά προβλήματα

Επειδή το λεξικό είναι μικρό και περιορισμένο δεν εκφράζονται όλες οι λέξεις που θα θέλαμε. Πρέπει να συνδυάσουμε κάποιες μεταξύ τους για να πάρουμε αυτό που θέλουμε. Εδώ σημαντικό ρόλο παίζει η φαντασία του ομιλητή που μπορεί να διαφέρει από αυτή του ακροατή. Εφόσον δεν μπορούν να δημιουργηθούν νέες κοινές λέξεις, το νόημα για τον κάθε έναν μπορεί να διαφέρει. Για παράδειγμα πως θα μπορούσαμε να πούμε ρόδα; Κάποιος ίσως την έλεγε στρογγυλό πόδι, κάποιος κύκλο που πηγαίνει και όπως αλλιώς θα μπορούσε ο κάθε ένας μας να το προσεγγίσει. Όμως τελικά το πρόβλημα λύνεται μόνο καταχωρώντας σε μία βάση γνώσης τις νέες λέξεις στις οποίες θα έχει δοθεί ερμηνεία φυσικής γλώσσας.

Βέβαια ο τρόπος που συνθέτουμε λέξεις με τη SostiMatiko θα βοηθούσε στην περίπτωση που το πρόβλημα είναι αν δύο λέξεις περιγράφουν ένα ζεύγος επιθέτου-ουσιαστικού ή ένα μεγάλο ουσιαστικό. Με απλά Ελληνικά λοιπόν, αντί να πούμε «στρογγυλό πόδι» μπορούμε να πούμε «στρογγυλόποδο» και έτσι το σημασιολογικό πρόβλημα να λυθεί.

5.4 Τελικό Συμπέρασμα για το Σύνολο της Εργασίας

Αν και υπάρχουν όπως είδαμε προβλήματα κατά την επεξεργασία, κάποια από τα οποία μπορούμε να τα λύσουμε και κάποια όχι, το σύνολο υπό μερικές προϋποθέσεις που θέσαμε και που είναι εντός των κανόνων της τεχνητής γλώσσας κινήθηκε ικανοποιητικά. Πιο συγκεκριμένα, ο συνδυασμός της SostiMatiko και της OMAS-III αποτέλεσαν ένα πλαίσιο μίας καλής πλατφόρμας για την ανάπτυξη της παρούσας εργασίας. Το αποτέλεσμα πέρασε τις προσδοκίες καθότι προκύπτει πως η OMAS-III σαν σημασιολογικός φορμαλισμός μπορεί να αποτελέσει τη ραχοκοκαλιά ενός ευφυούς ρομποτικού συστήματος. Κάθε απόληξη του σκελετού αυτού αποτελεί μία ολόκληρη ιδέα για έρευνα και ανάπτυξη ενός αυτόνομου λογισμικού. Κάθε ένα τέτοιο λογισμικό, αν συνενωθεί με τα υπόλοιπα υπό τον έλεγχο του μοντέλου, τότε θα προκύψει ένα τέτοιο εντυπωσιακό σύστημα. Αυτή η εργασία πέτυχε το στόχο της, αποτελώντας και το μονοπάτι για πολλές επιμέρους εργασίες στο δρόμο της ανάπτυξης μιας πιο έξυπνης ρομποτικής εφαρμογής.

ΠΕΡΙΕΧΟΜΕΝΑ

1	ΕΙΣΑΓΩΓΗ.....	- 2 -
1.1	Βασικές Έννοιες & Ορισμοί.....	- 2 -
1.1.1	Τεχνητή / κατασκευασμένη γλώσσα.....	- 2 -
1.1.2	Φιλοσοφική γλώσσα.....	- 2 -
1.1.3	Φυσική Σημασιολογική Μεταγλώσσα.....	- 4 -
1.1.4	Robot Interaction Language (ROILA).....	- 6 -
1.1.5	Toki Pona.....	- 10 -
1.1.6	SostiMatiko.....	- 10 -
1.1.7	ΟΜΑΣ-III.....	- 10 -
1.2	Σκοπός της εργασίας.....	- 11 -
1.3	Παραπομπές και URLS αναφορών κεφαλαίου.....	- 12 -
2	ΑΝΑΛΥΣΗ.....	- 13 -
2.1	SostiMatiko.....	- 13 -
2.1.1	Λεξιλόγιο (Vocabulary).....	- 13 -
2.1.2	Φωνολογία (Phonology and Sandhi).....	- 13 -
2.1.3	Έμφαση συλλαβής/φωνήεντος (Syllable / vowel emphasis).....	- 15 -
2.1.4	Γραφή (Writing system).....	- 15 -
2.1.5	Γραμματική (Grammar).....	- 15 -
2.1.6	Περισσότερα για τη σειρά των λέξεων.....	- 17 -
2.1.7	Δείκτης υποκειμένου (Subject indicator).....	- 18 -
2.1.8	Λέξεις που δείχνουν γραμματικές σχέσεις (Grammatical words).....	- 19 -
2.1.9	Αριθμοί (Numbers).....	- 19 -
2.1.10	Συνένωση καταλήξεων (Concatenation of suffixes).....	- 20 -
2.1.11	Δημιουργώντας ρήματα και επιρρήματα από ουσιαστικά (Making verbs and adverbs from nouns).....	- 20 -
2.1.12	Συντομεύσεις (Shortcuts):.....	- 21 -
2.1.13	Αμφισημία (Ambiguity).....	- 24 -
2.1.14	Βασικά στοιχεία και μορφές της γραμματικής.....	- 24 -
2.2	ΟΜΑΣ-III.....	- 25 -
2.2.1	Βασική Υπόθεση.....	- 26 -
2.2.2	Βασική Περιγραφή.....	- 26 -
2.2.3	Ανάλυση.....	- 27 -
2.3	Παραπομπές και URLS αναφορών κεφαλαίου.....	- 28 -
3	ΣΧΕΔΙΑΣΗ.....	- 29 -
3.1	Προσεγγίζοντας τις βασικές λειτουργίες.....	- 29 -
3.1.1	Χειρισμός δεδομένων.....	- 29 -
3.1.2	διευκρινιστικά ερωτήματα.....	- 30 -
3.1.3	Τελική λίστα ενεργειών.....	- 30 -
3.2	Η ΟΜΑΣ III και ο Φορμαλισμός.....	- 31 -
3.2.1	Το ερώτημα Γιατί.....	- 31 -
3.2.2	Το ερώτημα Τι.....	- 32 -
3.2.3	Το ερώτημα Πόσο.....	- 32 -
3.2.4	Το ερώτημα Πως.....	- 32 -
3.2.5	Το ερώτημα Ποιος.....	- 32 -
3.2.6	Το ερώτημα Που.....	- 32 -
3.2.7	Το ερώτημα Πότε.....	- 33 -
3.3	Ειδικότερα με τις διαδικασίες του προγράμματος.....	- 33 -

3.3.1	Επιλογή αρχείου κειμένου.....	- 33 -
3.3.2	Ανάγνωση λέξης.....	- 33 -
3.3.3	Υπόθεση 1.....	- 34 -
3.3.4	Ανάλυση Λέξης.....	- 34 -
3.3.5	Υπόθεση 2.....	- 34 -
3.3.6	Προσδιορισμός Χώρου – Χρόνου.....	- 35 -
3.3.7	Σύνταξη Προσωρινής Δομής.....	- 35 -
3.3.8	Ελλείψεις Δομής.....	- 35 -
3.3.9	Τοποθέτηση Γραμμής στον Τελικό Πίνακα, στην Σωστή Χρονική Σειρά - 36 -	- 36 -
3.3.10	Εμφάνιση Πίνακα Ενεργειών.....	- 36 -
3.3.11	Αρχικές Τιμές.....	- 36 -
3.3.12	Περιβάλλον.....	- 36 -
3.3.13	Υποβολή Ερώτησης.....	- 36 -
3.3.14	Γραμματική.....	- 36 -
3.3.15	Λεξικό.....	- 37 -
3.4	Το Interface του προγράμματος.....	- 37 -
3.5	Βασικός Αλγόριθμος.....	- 37 -
4	ΥΛΟΠΟΙΗΣΗ.....	- 39 -
4.1	Η διαδικασία “main”.....	- 40 -
4.2	Οι Classes του κώδικα.....	- 44 -
4.2.1	SentenceConfirmation Class.....	- 44 -
4.2.2	WordAnalysis Class.....	- 45 -
4.2.3	DynamicWord Class.....	- 46 -
4.2.4	DynamicLineClass.....	- 47 -
4.2.5	SupportFunctions Class.....	- 48 -
4.2.6	Omas_III_Agreements_Modulation Class.....	- 49 -
4.3	Παράδειγμα λειτουργίας του κώδικα.....	- 49 -
4.4	Παραπομπές και URLS αναφορών κεφαλαίου.....	- 51 -
5	ΣΥΜΠΕΡΑΣΜΑΤΑ - ΠΑΡΑΤΗΡΗΣΕΙΣ.....	- 52 -
5.1	Για τη SostiMatiko.....	- 52 -
5.1.1	Επισημάνσεις στα θετικά της στοιχεία επί του συστήματος.....	- 52 -
5.1.2	Προβληματισμοί και Δυσκολίες εφαρμογής της γλώσσας επί του συστήματος.....	- 53 -
5.1.3	Συμπέρασμα.....	- 55 -
5.2	Για την ΟΜΑΣ-III.....	- 55 -
5.2.1	Απάντηση για το Ερώτημα Τί.....	- 56 -
5.2.2	Απάντηση για το ερώτημα Ποιός.....	- 56 -
5.2.3	Απάντηση για το ερώτημα Πώς.....	- 56 -
5.2.4	Απάντηση στο ερώτημα Πόσο.....	- 56 -
5.2.5	Απάντηση στο ερώτημα Πότε.....	- 56 -
5.2.6	Απάντηση στο ερώτημα Πού.....	- 57 -
5.2.7	Απάντηση στο ερώτημα Γιατί.....	- 57 -
5.2.8	Συμπέρασμα.....	- 57 -
5.3	Προβλήματα επεξεργασίας.....	- 57 -
5.3.1	Προβλήματα μορφολογίας.....	- 57 -
5.3.2	Προβλήματα σύνταξης.....	- 57 -
5.3.3	Σημασιολογικά προβλήματα.....	- 58 -
5.4	Τελικό Συμπέρασμα για το Σύνολο της Εργασίας.....	- 58 -

ΠΑΡΑΡΤΗΜΑ Α	Πλήρες Λεξικό της Τεχνητής Γλώσσας SostiMatiko.....	A1-A10
ΠΑΡΑΡΤΗΜΑ Β	Οργανωτική Μέθοδος Ανάλυσης Συστημάτων - ΙΙΙ.....	B1-B2
ΠΑΡΑΡΤΗΜΑ Γ	Οπτικός Δίσκος με το Λογισμικό του Συστήματος.....	Γ1