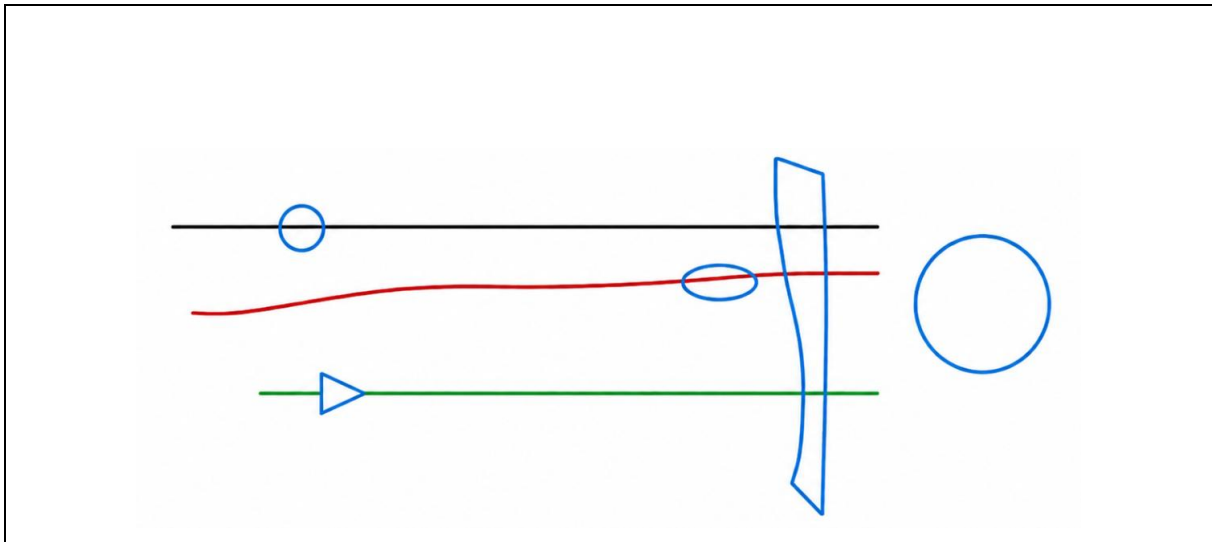


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Αυτόματος Προγραμματισμός Συναντήσεων με Χρήση
Πολυπρακτορικών Συστημάτων που Υποστηρίζονται
από Μεγάλα Γλωσσικά Μοντέλα»



Του φοιτητή
Αλέξιου Ηλιάδη.
Αρ. Μητρώου: 52020

Επιβλέπων
Ονοματεπώνυμο Ευστάθιος
Αντωνίου
Βαθμίδα Καθηγητής

Ημερομηνία 06-13-2026

Τίτλος Δ.Ε. Αυτόματος Προγραμματισμός Συναντήσεων με Χρήση Πολυπρακτορικών Συστημάτων
που Υποστηρίζονται από Μεγάλα Γλωσσικά Μοντέλα

Κωδικός Δ.Ε. [21310](#)

Ονοματεπώνυμο φοιτητή Αλέξιος Ηλιάδης

Ονοματεπώνυμο εισηγητή Ευστάθιος Αντωνίου

Ημερομηνία ανάληψης Δ.Ε. 12-10-2025

Ημερομηνία περάτωσης Δ.Ε. 06-13-2026

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Ηλιάδη Αλέξιου που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Η επιλογή της παρούσας διπλωματικής εργασίας προέκυψε από το ενδιαφέρον μου για τα πολυπρακτορικά συστήματα, την τεχνητή νοημοσύνη και τις δυνατότητες που προσφέρουν οι σύγχρονες υπολογιστικές τεχνολογίες στην αυτοματοποίηση σύνθετων διαδικασιών. Ο αυτόματος προγραμματισμός συναντήσεων αποτελεί ένα πρόβλημα με ιδιαίτερο πρακτικό ενδιαφέρον, καθώς συνδυάζει καταναμημένο συντονισμό, διαχείριση περιορισμών και αλληλεπίδραση με τον χρήστη μέσω φυσικής γλώσσας.

Μέσα από την εκπόνηση της εργασίας αυτής είχα την ευκαιρία να εμβαθύνω τόσο σε θεωρητικά ζητήματα, όπως οι ευφυείς πράκτορες και ο καταναμημένος συντονισμός, όσο και σε πρακτικά θέματα υλοποίησης, όπως η ανάπτυξη πολυπρακτορικών εφαρμογών, η χρήση διαδικτυακών υπηρεσιών και η υποστηρικτική αξιοποίηση Μεγάλου Γλωσσικού Μοντέλου σε ένα εκτελέσιμο πρωτότυπο σύστημα.

Το σημαντικότερο όφελος από την εκπόνηση της εργασίας ήταν η δυνατότητα να συνδυάσω διαφορετικές τεχνολογικές περιοχές σε μία ενιαία εφαρμογή και να αποκτήσω εμπειρία στη σχεδίαση, υλοποίηση και αξιολόγηση ενός ολοκληρωμένου συστήματος με ερευνητικό και πρακτικό ενδιαφέρον.

Περίληψη

Η παρούσα διπλωματική εργασία εξετάζει τον αυτόματο προγραμματισμό συναντήσεων με χρήση πολυπρακτορικού συστήματος που υποστηρίζεται από Μεγάλο Γλωσσικό Μοντέλο. Στόχος της εργασίας είναι η σχεδίαση, η υλοποίηση και η πειραματική αξιολόγηση ενός πολυπρακτορικού συστήματος για τον προγραμματισμό συναντήσεων μεταξύ μελών ενός οργανωμένου περιβάλλοντος, στο οποίο η είσοδος δίνεται σε φυσική γλώσσα, μετατρέπεται σε δομημένη προδιαγραφή και στη συνέχεια επεξεργάζεται από συνεργαζόμενους πράκτορες.

Η προτεινόμενη αρχιτεκτονική βασίζεται σε διακριτούς ρόλους πρακτόρων, όπως πράκτορα συντονισμού, πράκτορες-μελών και πράκτορες-χώρων. Ο μηχανισμός προγραμματισμού συνδυάζει φάση συλλογής τοπικών αξιολογήσεων και φάση οριστικοποίησης, ώστε να διαχειρίζεται περιορισμούς που σχετίζονται με διαθεσιμότητα, χρονικά παράθυρα, πόρους και οργανωσιακές απαιτήσεις. Το Μεγάλο Γλωσσικό Μοντέλο χρησιμοποιείται υποστηρικτικά για τη σημασιολογική κατανόηση των αιτημάτων φυσικής γλώσσας και τη μετατροπή τους σε δομημένη είσοδο, χωρίς να συμμετέχει άμεσα στη λήψη της τελικής απόφασης.

Η υλοποίηση πραγματοποιήθηκε με χρήση SPADE/XMPP για το πολυπρακτορικό επίπεδο, REST API για τη διαδικτυακή αλληλεπίδραση και παραγωγή αρχείων .ics για διαλειτουργική έξοδο. Η πειραματική αξιολόγηση έδειξε ότι το σύστημα λειτουργεί ορθά όταν υπάρχει εφικτή λύση, αποτυγχάνει με ελεγχόμενο τρόπο όταν δεν υπάρχει διαθέσιμο χρονικό διάστημα και παραμένει λειτουργικό ακόμη και με αυξημένο αριθμό συμμετεχόντων. Τα αποτελέσματα δείχνουν ότι η συνδυαστική χρήση πολυπρακτορικού συντονισμού και υποστήριξης από Μεγάλο Γλωσσικό Μοντέλο αποτελεί μια λειτουργική και εφαρμόσιμη προσέγγιση για τον αυτόματο προγραμματισμό συναντήσεων.

.

« Automatic Meeting Scheduling Using LLM-Supported Multi-Agent Systems »

«Iliadis Alexios»

Abstract

This thesis investigates automatic meeting scheduling through a multi-agent system supported by a Large Language Model. Its objective is the design, implementation, and experimental evaluation of a multi-agent system for scheduling meetings among members of an organized environment, in which input is provided in natural language, transformed into a structured specification, and subsequently processed by cooperating agents. The proposed architecture is based on distinct agent roles, including a coordinator agent, member agents, and room agents. The scheduling mechanism combines a phase of collecting local evaluations with a separate finalization phase in order to manage constraints related to availability, time windows, resources, and organizational requirements. The Large Language Model is used as a supporting component for the semantic interpretation of natural-language requests and their transformation into structured input, without directly participating in the final scheduling decision. The implementation was developed using SPADE/XMPP for the multi-agent layer, a REST API for web-based interaction, and .ics file generation for interoperable output. The experimental evaluation demonstrated that the system operates correctly when a feasible solution exists, fails in a controlled manner when no valid time slot is available, and remains functional even with an increased number of participants. The results further indicate that the primary source of latency lies in the natural-language processing stage rather than in the scheduling mechanism itself. Overall, the results show that the combination of multi-agent coordination and Large Language Model support constitutes a functional and practical approach to automatic meeting scheduling.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της παρούσας διπλωματικής εργασίας, κ. Ευστάθιο Αντωνίου, καθώς και την οικογένειά μου για τη στήριξή τους κατά τη διάρκεια των σπουδών μου και της ολοκλήρωσης της παρούσας εργασίας.

Περιεχόμενα

Περιεχόμενα

Πρόλογος.....	iv
Περίληψη.....	v
Abstract	vi
Ευχαριστίες	vii
Περιεχόμενα	viii
Κατάλογος Σχημάτων	xii
Κατάλογος Πινάκων.....	xii
Συντομογραφίες.....	xii
Κεφάλαιο 1ο: Εισαγωγή	1
1.1 Εισαγωγή	1
1.2 Το πρόβλημα του αυτόματου προγραμματισμού συναντήσεων	1
1.3 Πολυπρακτορικά συστήματα ως προσέγγιση επίλυσης	1
1.4 Ρόλος του Μεγάλου Γλωσσικού Μοντέλου στη σημασιολογική κατανόηση αιτημάτων.....	2
1.5 Στόχος και συνεισφορά της εργασίας.....	2
1.6 Δομή της διπλωματικής εργασίας	2
Κεφάλαιο 2ο: Θεωρητικό Υπόβαθρο.....	4
2.1 Εισαγωγή.....	4
2.2 Πολυπρακτορικά Συστήματα	4
2.3 Ευφυείς πράκτορες και καταναεμημένος συντονισμός.....	4
2.4 Πρωτόκολλα διαπραγμάτευσης και μηχανισμοί συμφωνίας	4
2.5 Μεγάλα Γλωσσικά Μοντέλα και σημασιολογική κατανόηση	5
2.6 Πρότυπα διαλειτουργικότητας ημερολογίων	5
Κεφάλαιο 3ο: Σχετική Βιβλιογραφία.....	6
3.1 Εισαγωγή.....	6
3.2 Πολυπρακτορικές προσεγγίσεις στον προγραμματισμό συναντήσεων.....	6
3.3 Προσεγγίσεις βασισμένες σε διαπραγμάτευση	6
3.4 Κεντριοποιημένες και καταναεμημένες προσεγγίσεις προγραμματισμού συναντήσεων.....	7
3.5 Πρωτόκολλα συντονισμού και συνέπειας σε καταναεμημένα συστήματα	8
3.6 Σύγχρονες προσεγγίσεις με υποστήριξη μεγάλων γλωσσικών μοντέλων	9
3.7 Περιορισμοί και προκλήσεις της χρήσης μεγάλα γλωσσικά μοντέλα στον προγραμματισμό συναντήσεων	9
3.8 Συγκριτική τοποθέτηση της παρούσας εργασίας	10

Κεφάλαιο 4ο:	Αρχιτεκτονική του Συστήματος.....	12
4.1	Εισαγωγή.....	12
4.2	Γενική επισκόπηση της αρχιτεκτονικής.....	12
4.3	Πολυπρακτορικό μοντέλο και ρόλοι πρακτόρων.....	13
4.4	Μηχανισμός συντονισμού και ροή απόφασης.....	13
4.5	Πράκτορες συμμετεχόντων και τοπική αξιολόγηση.....	14
4.6	Γνωστικό υποσύστημα σημασιολογικής κατανόησης.....	14
4.7	Διεπαφή υπηρεσίας και επίπεδο ενορχήστρωσης.....	14
4.8	Υποδομή επικοινωνίας και διαχείριση ημερολογιακών δεδομένων.....	15
4.9	Βασικές αρχιτεκτονικές ιδιότητες του συστήματος.....	15
Κεφάλαιο 5ο:	Υλοποίηση.....	16
5.1	Εισαγωγή.....	16
5.2	Περιβάλλον υλοποίησης και τεχνολογίες.....	16
5.3	Δομή του έργου.....	17
5.4	Υλοποίηση πρακτόρων με SPADE.....	17
5.5	Υλοποίηση του πυρήνα προγραμματισμού.....	18
5.6	Υλοποίηση πρακτόρων ημερολογίου και διαθεσιμότητας.....	18
5.7	Ενσωμάτωση του Μεγάλου Γλωσσικού Μοντέλου στο API.....	19
5.8	Εκτέλεση βάσει εργασιών και ενορχήστρωση.....	19
5.9	Παραγόμενα αρχεία και αποτελέσματα εκτέλεσης.....	20
5.10	Μετρικές και παρακολούθηση συστήματος.....	20
Κεφάλαιο 6ο:	Πειραματική Αξιολόγηση.....	21
6.1	Εισαγωγή.....	21
6.2	Στόχοι της αξιολόγησης.....	21
6.3	Πειραματικό περιβάλλον και μεθοδολογία.....	21
6.4	Μετρικές αξιολόγησης.....	22
6.4.1	Ποσοστό επιτυχίας (Success Rate).....	22
6.4.2	Χρόνος απόκρισης (Latency).....	22
6.4.3	Συμμετοχικότητα και λογική απόφασης.....	22
6.5	Αξιολόγηση κλιμάκωσης.....	22
6.5.1	Περιγραφή.....	22
6.5.2	Πειραματική ρύθμιση.....	22
6.5.3	Αποτελέσματα.....	23
6.5.4	Ανάλυση.....	23
6.5.5	Συμπέρασμα.....	24

6.6	Ανάλυση απόδοσης και χρόνου απόκρισης.....	24
6.6.1	Περιγραφή	24
6.6.2	Πειραματική ρύθμιση	24
6.6.3	Αποτελέσματα	24
6.6.4	Ανάλυση	25
6.6.5	Συμπέρασμα.....	25
6.7	Επίδραση του βήματος αναζήτησης.....	25
6.7.1	Περιγραφή	25
6.7.2	Πειραματική ρύθμιση	25
6.7.3	Αποτελέσματα	26
6.7.4	Ανάλυση	26
6.7.5	Συμπέρασμα.....	26
6.8	Περίπτωση αποτυχίας: απουσία διαθέσιμου χρονικού διαστήματος	26
6.8.1	Περιγραφή	26
6.8.2	Πειραματική ρύθμιση	26
6.8.3	Αποτελέσματα	27
6.8.4	Ανάλυση	27
6.8.5	Συμπέρασμα.....	27
6.9	Συνολικά συμπεράσματα πειραματικής αξιολόγησης.....	27
6.10	Συμπέρασμα κεφαλαίου	27
Κεφάλαιο 7ο:	Συμπεράσματα και μελλοντική εργασία.....	29
7.1	Εισαγωγή.....	29
7.2	Συμπεράσματα.....	29
7.3	Περιορισμοί της προσέγγισης	30
7.4	Μελλοντική εργασία	30
7.5	Τελικό συμπέρασμα	31
BIBΛΙΟΓΡΑΦΙΑ.....		32
ΠΑΡΑΡΤΗΜΑ Α— Prompts προς το μεγάλο γλωσσικό μοντέλο και παραδείγματα κλήσεων API		33
7.6	A.1 System prompt προς το μεγάλο γλωσσικό μοντέλο	33
7.7	A.2 Πρότυπο εισόδου προς το μοντέλο.....	34
7.8	A.3 Βασικό αίτημα προγραμματισμού.....	34
7.9	A.4 Αίτημα με πολλούς συμμετέχοντες	35
7.10	A.5 Αίτημα κλιμάκωσης με 50 πράκτορες.....	35
7.11	A.6 Αίτημα περίπτωσης αποτυχίας	35
7.12	A.7 Μη έγκυρο χρονικό αίτημα	36

ΠΑΡΑΡΤΗΜΑ Β — Structured JSON και αρχεία εκτέλεσης	37
7.13 B.1 Δομημένο scheduling job (scheduler.json).....	37
7.14 B.2 Αρχείο παραμέτρων εκτέλεσης (scheduler.env).....	37
7.15 B.3 Αρχείο αποτελέσματος (result.json)	38
7.16 B.4 Αρχείο κατάστασης εκτέλεσης (status.json).....	38
7.17 B.5 Αρχείο καταγραφής εκτέλεσης (log.txt).....	38
7.18 B.6 Δομή καταλόγου εκτέλεσης εργασίας	39
ΠΑΡΑΡΤΗΜΑ Γ — Οδηγίες Εγκατάστασης και Εκτέλεσης.....	40
7.19 Γ.1 Προαπαιτούμενα	40
7.20 Γ.2 λήψη αποθετηρίου.....	40
7.21 Γ.3 Προετοιμασία γλωσσικού μοντέλου	40
7.22 Γ.4 Εκκίνηση συστήματος.....	40
7.23 Γ.5 Έλεγχος λειτουργίας API.....	40
7.24 Γ.6 Εκτέλεση ενδεικτικού αιτήματος	40
7.25 Γ.7 Τερματισμός συστήματος	41
7.26 Γ.7 Αποθετήριο πηγαίου κώδικα.....	41

Κατάλογος Σχημάτων

Σχήμα 4.2: Αρχιτεκτονική του πολυπρακτορικού συστήματος προγραμματισμού συναντήσεων 13

Κατάλογος Πινάκων

Πίνακας 6.5: Συνολικός χρόνος εκτέλεσης ως προς τον αριθμό των πρακτόρων	24
Πίνακας 6.6: Χρόνοι εκτέλεσης ανά επανάληψη του πειράματος	25
Πίνακας 6.6.1: Μέσες τιμές χρόνων εκτέλεσης	26
Πίνακας 6.7: Επίδραση του βήματος αναζήτησης στον αριθμό των υποψήφιων διαστημάτων και στον χρόνο εκτέλεσης	27

Συντομογραφίες

LLM	Μεγάλο Γλωσσικό Μοντέλο
MAS	Πολυπρακτορικό Σύστημα
API	Διεπαφή Προγραμματισμού Εφαρμογών
REST	Αναπαραστατική Μεταφορά Κατάστασης
JSON	JavaScript Object Notation
XMPP	Extensible Messaging and Presence Protocol
SPADE	Smart Python Multi-Agent Development Environment
RFC	Request for Comments
ICS	Αρχείο ημερολογίου iCalendar
CalDAV	Calendaring Extensions to WebDAV

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Ο αυτόματος προγραμματισμός συναντήσεων αποτελεί ένα απαιτητικό πρόβλημα, καθώς προϋποθέτει τον συνδυασμό διαθεσιμότητας, χρονικών περιορισμών, προτιμήσεων και διαθέσιμων πόρων. Παρότι υπάρχουν πολλά ψηφιακά εργαλεία ημερολογίου, η αυτοματοποίηση της διαδικασίας παραμένει δύσκολη, ιδιαίτερα όταν η πληροφορία είναι κατανεμημένη και οι εμπλεκόμενοι φορείς διατηρούν τοπική αυτονομία [1], [2].

Τα πολυπρακτορικά συστήματα προσφέρουν ένα κατάλληλο πλαίσιο για την αντιμετώπιση αυτού του προβλήματος, καθώς επιτρέπουν την αναπαράσταση χρηστών και πόρων ως αυτόνομων πρακτόρων που επικοινωνούν και συντονίζονται μέσω ανταλλαγής μηνυμάτων [1]. Με τον τρόπο αυτό, ο προγραμματισμός συναντήσεων προσεγγίζεται ως διαδικασία συνεργασίας μεταξύ πρακτόρων, όπου οι τοπικές αξιολογήσεις συλλέγονται και συντονίζονται από τον πράκτορα συντονισμού.

Παράλληλα, οι πρόσφατες εξελίξεις στα Μεγάλα Γλωσσικά Μοντέλα επιτρέπουν την υποβολή αιτημάτων σε φυσική γλώσσα και τη μετατροπή τους σε δομημένες αναπαραστάσεις κατάλληλες για επεξεργασία από υπολογιστικά συστήματα [3]. Ωστόσο, στο πλαίσιο της παρούσας εργασίας, η κατανόηση αιτημάτων σε φυσική γλώσσα δεν επαρκεί από μόνη της για τον αξιόπιστο προγραμματισμό συναντήσεων. Για τον λόγο αυτό, απαιτείται ένας μηχανισμός συντονισμού που να μπορεί να διαχειρίζεται περιορισμούς, να εξασφαλίζει συνέπεια και να υποστηρίζει ελέγξιμη λήψη αποφάσεων [2].

Η παρούσα διπλωματική εργασία εξετάζει πειραματικά μια υβριδική προσέγγιση για τον αυτόματο προγραμματισμό συναντήσεων, η οποία συνδυάζει πολυπρακτορική αρχιτεκτονική με υποστήριξη LLM στη σημασιολογική κατανόηση αιτημάτων. Πιο συγκεκριμένα, υλοποιείται ένα πολυπρακτορικό σύστημα προγραμματισμού, στο οποίο η είσοδος δίνεται σε φυσική γλώσσα, μετατρέπεται σε δομημένο αίτημα και εκτελείται από πράκτορες που συνεργάζονται μέσω πρωτοκόλλων ανταλλαγής μηνυμάτων.

1.2 Το πρόβλημα του αυτόματου προγραμματισμού συναντήσεων

Ο προγραμματισμός συναντήσεων δεν περιορίζεται στην εύρεση ενός κοινού ελεύθερου χρονικού διαστήματος. Στην πράξη, απαιτεί τη συνεκτίμηση πολλών παραγόντων, όπως η διαθεσιμότητα των συμμετεχόντων, η διάρκεια της συνάντησης, οι χρονικοί περιορισμοί, η διαθεσιμότητα χώρων και άλλων πόρων, καθώς και οργανωτικοί κανόνες ή προτιμήσεις [2].

Σε κεντρικές προσεγγίσεις, η επίλυση του προβλήματος προϋποθέτει συνήθως συγκέντρωση της πληροφορίας σε ένα ενιαίο σύστημα. Στα κατανεμημένα περιβάλλοντα, όμως, η υπόθεση αυτή δεν είναι πάντα ρεαλιστική. Για τον λόγο αυτό, ο προγραμματισμός συναντήσεων μπορεί να ιδωθεί ως πρόβλημα συνεργασίας μεταξύ πολλών αυτόνομων οντοτήτων, οι οποίες συμμετέχουν στη διαδικασία λήψης αποφάσεων χωρίς να χάνουν τον τοπικό έλεγχο της πληροφορίας τους [1], [2] [8].

1.3 Πολυπρακτορικά συστήματα ως προσέγγιση επίλυσης

Τα πολυπρακτορικά συστήματα είναι κατάλληλα για προβλήματα όπως ο προγραμματισμός συναντήσεων, επειδή επιτρέπουν την αποκέντρωση της γνώσης και της διαδικασίας συντονισμού [1]. Κάθε μέλος ή πόρος μπορεί να εκπροσωπείται από έναν πράκτορα, ο οποίος αξιολογεί τοπικά τις διαθέσιμες επιλογές και συμμετέχει σε διαδικασία συμφωνίας.

Η προσέγγιση αυτή προσφέρει πλεονεκτήματα όπως επεκτασιμότητα, σαφέστερο διαχωρισμό ρόλων και μεγαλύτερη αντιστοίχιση με πραγματικά οργανωσιακά περιβάλλοντα, στα οποία οι αποφάσεις λαμβάνονται μέσα από συνεργασία και όχι αποκλειστικά μέσω ενός κεντρικού αλγορίθμου [1].

1.4 Ρόλος του Μεγάλου Γλωσσικού Μοντέλου στη σημασιολογική κατανόηση αιτημάτων

Στην παρούσα εργασία, το Μεγάλο Γλωσσικό Μοντέλο αξιοποιείται ως υποστηρικτικό γνωστικό υποσύστημα που διευκολύνει τη σημασιολογική αλληλεπίδραση με τον χρήστη. Πιο συγκεκριμένα, χρησιμοποιείται για τη μετατροπή αιτημάτων φυσικής γλώσσας σε δομημένες προδιαγραφές προγραμματισμού συναντήσεων, ώστε το πολυπρακτορικό σύστημα να μπορεί να τις επεξεργαστεί με σαφή και ελέγξιμο τρόπο. Η επιλογή αυτή έγινε με γνώμονα την υλοποίηση ενός ελαφρού και πρακτικά διαχειρίσιμου πρωτοτύπου, στο οποίο το γλωσσικό μοντέλο λειτουργεί υποστηρικτικά και όχι ως κεντρικός μηχανισμός λήψης αποφάσεων.

Κατά συνέπεια, το Μεγάλο Γλωσσικό Μοντέλο δεν αποτελεί τον μηχανισμό λήψης της τελικής απόφασης, αλλά λειτουργεί ως ενδιάμεσο επίπεδο κατανόησης και μετασχηματισμού της πρόθεσης του χρήστη. Ο πραγματικός συντονισμός της διαδικασίας υλοποιείται από τους πράκτορες του συστήματος.

1.5 Στόχος και συνεισφορά της εργασίας

Στόχος της εργασίας είναι η σχεδίαση, η υλοποίηση και η πειραματική αξιολόγηση ενός πολυπρακτορικού συστήματος για τον αυτόματο προγραμματισμό συναντήσεων, με υποστήριξη φυσικής γλώσσας στην είσοδο. Η προσέγγιση συνδυάζει κατανεμημένους πράκτορες, πρωτόκολλα συντονισμού βασισμένα σε ανταλλαγή μηνυμάτων και ένα σημασιολογικό επίπεδο διεπαφής με υποστήριξη Μεγάλου Γλωσσικού Μοντέλου.

Η συνεισφορά της εργασίας εντοπίζεται κυρίως στα εξής σημεία:

- στον σχεδιασμό ενός κατανεμημένου συστήματος προγραμματισμού βασισμένου σε πράκτορες,
- στην ενσωμάτωση φυσικής γλώσσας ως μέσου υποβολής αιτημάτων,
- στην υλοποίηση μηχανισμού συντονισμού με συλλογή αξιολογήσεων, λογική απαρτίας και φάσεις `prepare/commit`,
- στην παραγωγή δομημένων αποτελεσμάτων και διαλειτουργικών τεχνουργημάτων ημερολογίου,
- και στην πειραματική αξιολόγηση της προσέγγισης ως προς τη λειτουργικότητα, την απόδοση και την κλιμάκωση.

Η εργασία δεν επιδιώκει να εισαγάγει ένα νέο θεωρητικό πρόβλημα, αλλά να διερευνήσει πειραματικά μια σύγχρονη υβριδική προσέγγιση που συνδυάζει πολυπρακτορικό συντονισμό και υποστήριξη Μεγάλου Γλωσσικού Μοντέλου για τη σημασιολογική κατανόηση αιτημάτων.

1.6 Δομή της διπλωματικής εργασίας

Η εργασία οργανώνεται ως εξής. Στο Κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο σχετικά με τα πολυπρακτορικά συστήματα, τους ευφυείς πράκτορες, τον κατανεμημένο συντονισμό, τα Μεγάλα Γλωσσικά Μοντέλα και τα πρότυπα διαλειτουργικότητας ημερολογίων. Στο Κεφάλαιο 3 παρουσιάζεται η σχετική βιβλιογραφία και η τοποθέτηση της παρούσας προσέγγισης. Στο Κεφάλαιο 4 περιγράφεται η αρχιτεκτονική του συστήματος, ενώ στο Κεφάλαιο 5 αναλύονται οι λεπτομέρειες της υλοποίησής του.

Στο Κεφάλαιο 6 παρουσιάζεται η πειραματική αξιολόγηση και η ανάλυση των αποτελεσμάτων. Τέλος, στο Κεφάλαιο 7 διατυπώνονται τα συμπεράσματα και οι κατευθύνσεις για μελλοντική εργασία.

Κεφάλαιο 2ο: Θεωρητικό Υπόβαθρο

2.1 Εισαγωγή

Το κεφάλαιο αυτό παρουσιάζει το θεωρητικό υπόβαθρο της παρούσας εργασίας. Η προτεινόμενη προσέγγιση βασίζεται στον συνδυασμό πολυπρακτορικών συστημάτων, μηχανισμών κατανεμημένου συντονισμού, Μεγάλων Γλωσσικών Μοντέλων και προτύπων διαλειτουργικότητας ημερολογίων. Για τον λόγο αυτό, είναι απαραίτητη η συνοπτική παρουσίαση των βασικών εννοιών που υποστηρίζουν τη σχεδίαση και την υλοποίηση του προτεινόμενου συστήματος.

2.2 Πολυπρακτορικά Συστήματα

Τα πολυπρακτορικά συστήματα αποτελούνται από πολλαπλές αυτόνομες υπολογιστικές οντότητες, τους πράκτορες, οι οποίες αλληλεπιδρούν μεταξύ τους με σκοπό την επίλυση προβλημάτων ή την επίτευξη επιμέρους και συλλογικών στόχων [1]. Κάθε πράκτορας διαθέτει έναν βαθμό αυτονομίας, τοπική γνώση και δυνατότητα λήψης αποφάσεων, ενώ η συνολική συμπεριφορά του συστήματος προκύπτει από τη συνεργασία και τον συντονισμό των επιμέρους οντοτήτων [1].

Η προσέγγιση αυτή είναι ιδιαίτερα κατάλληλη για προβλήματα στα οποία η πληροφορία είναι κατανεμημένη ή στα οποία οι εμπλεκόμενες οντότητες πρέπει να διατηρούν έναν βαθμό τοπικής αυτονομίας [1]. Στο πλαίσιο του προγραμματισμού συναντήσεων, κάθε χρήστης ή διαθέσιμος πόρος μπορεί να μοντελοποιηθεί από έναν ξεχωριστό πράκτορα, ο οποίος συμμετέχει στη διαδικασία λήψης απόφασης χωρίς να απαιτείται πλήρης κεντροκοποίηση της πληροφορίας.

2.3 Ευφυείς πράκτορες και κατανεμημένος συντονισμός

Οι ευφυείς πράκτορες είναι υπολογιστικές οντότητες που αντιλαμβάνονται το περιβάλλον τους, λαμβάνουν αποφάσεις και ενεργούν με σκοπό την επίτευξη συγκεκριμένων στόχων [1]. Συχνά διακρίνονται σε αντιδραστικούς πράκτορες, οι οποίοι ανταποκρίνονται άμεσα στις μεταβολές του περιβάλλοντος, και σε συλλογιστικούς πράκτορες, οι οποίοι λαμβάνουν αποφάσεις με βάση στόχους, σχέδια ή εσωτερικές αναπαραστάσεις της κατάστασης [1]. Σε πολλές σύγχρονες αρχιτεκτονικές, οι δύο αυτές προσεγγίσεις συνδυάζονται, ώστε το σύστημα να διαθέτει τόσο ευελιξία αντίδρασης όσο και δυνατότητα πιο σύνθετης λήψης αποφάσεων [10].

Σε ένα πολυπρακτορικό περιβάλλον, κεντρικό ρόλο παίζει ο κατανεμημένος συντονισμός. Ο συντονισμός αφορά τους μηχανισμούς μέσω των οποίων πολλοί πράκτορες ανταλλάσσουν πληροφορίες, ευθυγραμμίζουν τις τοπικές τους αποφάσεις και καταλήγουν σε συμβατές συλλογικές ενέργειες [1], [2]. Στην πράξη, ο κατανεμημένος συντονισμός μπορεί να υλοποιείται μέσω ανταλλαγής μηνυμάτων, διαδικασιών διαπραγμάτευσης, λογικών απαρτίων και πολυφασικών μηχανισμών συμφωνίας [2].

2.4 Πρωτόκολλα διαπραγμάτευσης και μηχανισμοί συμφωνίας

Σε πολλά πολυπρακτορικά συστήματα, η τελική απόφαση δεν προκύπτει άμεσα, αλλά μέσα από διαδικασίες διαπραγμάτευσης και σταδιακής συμφωνίας. Ένας πράκτορας μπορεί να ζητά προτάσεις, αξιολογήσεις ή δεσμεύσεις από άλλους πράκτορες, να συλλέγει τις απαντήσεις τους και στη συνέχεια να επιδιώκει την οριστικοποίηση μιας επιλογής [2].

Χαρακτηριστικό παράδειγμα αυτής της λογικής αποτελεί το Contract Net Protocol, το οποίο περιγράφει έναν μηχανισμό κατανεμημένης ανάθεσης και συντονισμού, όπου ένας κόμβος ανακοινώνει μια εργασία και άλλοι κόμβοι απαντούν με προτάσεις ή προσφορές [2]. Η βασική ιδέα τέτοιων

πρωτοκόλλων είναι ότι η λήψη απόφασης μπορεί να προκύπτει μέσα από κατανεμημένη επικοινωνία και όχι αποκλειστικά από έναν κεντρικό ελεγκτή.

Σε πιο σύνθετα περιβάλλοντα, η διαδικασία αυτή μπορεί να συνδυάζεται με μηχανισμούς πολλαπλών φάσεων, στους οποίους η συλλογή των τοπικών αποκρίσεων διαχωρίζεται από τη φάση της τελικής δέσμευσης σε μία απόφαση. Η λογική αυτή είναι ιδιαίτερα σημαντική σε προβλήματα όπου απαιτείται ελεγχόμενη οριστικοποίηση, όπως στον προγραμματισμό συναντήσεων.

2.5 Μεγάλα Γλωσσικά Μοντέλα και σημασιολογική κατανόηση

Τα Μεγάλα Γλωσσικά Μοντέλα έχουν ενισχύσει σημαντικά τη δυνατότητα επεξεργασίας φυσικής γλώσσας σε σύγχρονα ευφυή συστήματα. Μπορούν να αξιοποιηθούν για την κατανόηση αιτημάτων χρηστών, την εξαγωγή δομημένης πληροφορίας από φυσικό κείμενο και τη μετατροπή λεκτικών περιγραφών σε αναπαραστάσεις κατάλληλες για επεξεργασία από άλλα υποσυστήματα [3].

Στο πλαίσιο της παρούσας εργασίας, το Μεγάλο Γλωσσικό Μοντέλο λειτουργεί υποστηρικτικά ως προς τη σημασιολογική κατανόηση των αιτημάτων και τη μετατροπή τους σε δομημένες προδιαγραφές. Ο μηχανισμός προγραμματισμού και η τελική λήψη απόφασης υλοποιούνται από το πολυπρακτορικό σύστημα.

2.6 Πρότυπα διαλειτουργικότητας ημερολογίων

Για την πρακτική αξιοποίηση ενός συστήματος προγραμματισμού συναντήσεων είναι σημαντική η δυνατότητα διαλειτουργίας με πραγματικά ημερολογιακά συστήματα. Το πρότυπο iCalendar χρησιμοποιείται ευρέως για την αναπαράσταση γεγονότων, χρονικών ιδιοτήτων και μεταδεδομένων ημερολογίου [4]. Παράλληλα, οι επεκτάσεις του CalDAV Scheduling υποστηρίζουν πιο ολοκληρωμένες μορφές ανταλλαγής δεδομένων προγραμματισμού και συνεργατικής διαχείρισης συναντήσεων [5]. Στην παρούσα εργασία, η σημασία των προτύπων αυτών εντοπίζεται κυρίως στην παραγωγή διαλειτουργικών αποτελεσμάτων, όπως αρχεία .ics, τα οποία μπορούν να αξιοποιηθούν και εκτός του ερευνητικού πρωτοτύπου [4]. Έτσι, το προτεινόμενο σύστημα διατηρεί τη δυνατότητα διασύνδεσης με υφιστάμενα οικοσυστήματα ημερολογίων και σχετικές διαδικτυακές υπηρεσίες.

Κεφάλαιο 3ο: Σχετική Βιβλιογραφία

3.1 Εισαγωγή

Η έρευνα γύρω από τον αυτόματο προγραμματισμό συναντήσεων έχει αναπτυχθεί σε διαφορετικές αλλά αλληλοσυμπληρούμενες κατευθύνσεις. Οι σημαντικότερες από αυτές αφορούν τις πολυπρακτορικές προσεγγίσεις, τις προσεγγίσεις που βασίζονται σε διαπραγμάτευση, καθώς και τις πιο πρόσφατες εργασίες που αξιοποιούν Μεγάλα Γλωσσικά Μοντέλα για την αλληλεπίδραση με τον χρήστη και την υποστήριξη πολύπλοκων ροών εργασίας. Η παρούσα εργασία τοποθετείται στο σημείο τομής των παραπάνω κατευθύνσεων, με βασική έμφαση όμως στον καταναμημένο συντονισμό μεταξύ πρακτόρων.

3.2 Πολυπρακτορικές προσεγγίσεις στον προγραμματισμό συναντήσεων

Μία από τις πιο σχετικές εργασίες είναι η *Meeting Scheduling with Multi Agent Systems: Design and Implementation* [6], στην οποία προτείνεται ένα καταναμημένο σύστημα προγραμματισμού συναντήσεων με δύο βασικούς τύπους πρακτόρων: προσωπικούς πράκτορες, οι οποίοι εκπροσωπούν τους χρήστες, και πράκτορα τοποθεσίας, ο οποίος διαχειρίζεται τους χώρους συναντήσεων. Η εργασία αυτή είναι σημαντική, διότι δείχνει ότι ο διαχωρισμός ρόλων ανάμεσα σε πράκτορες που εκπροσωπούν συμμετέχοντες και σε πράκτορες που εκπροσωπούν πόρους αποτελεί μια βιώσιμη αρχιτεκτονική λύση για το πρόβλημα.

Συναφής είναι και η εργασία *Chronos: A Multi-Agent System for Distributed Automatic Meeting Scheduling* [7], η οποία περιγράφει ένα καταναμημένο σύστημα στο οποίο κάθε χρήστης εκπροσωπείται από έναν πράκτορα. Στο σύστημα αυτό δεν λαμβάνεται υπόψη μόνο η διαθεσιμότητα, αλλά και οι προτιμήσεις και οι συνήθειες των χρηστών [7]. Σε σύγκριση με την παρούσα εργασία, το *Chronos* είναι κοντινό ως προς την καταναμημένη λογική, αλλά διαφοροποιείται επειδή δίνει μεγαλύτερη έμφαση στην αξιοποίηση προτιμήσεων, ενώ η παρούσα εργασία δίνει μεγαλύτερο βάρος στον μηχανισμό συντονισμού και στη διαδικασία οριστικοποίησης.

Γενικότερα, οι πολυπρακτορικές προσεγγίσεις δείχνουν ότι ο προγραμματισμός συναντήσεων αποτελεί πεδίο στο οποίο οι πράκτορες είναι ιδιαίτερα κατάλληλοι, επειδή επιτρέπουν τοπική λήψη αποφάσεων, διατήρηση αποκεντρωμένης γνώσης και καλύτερη προσαρμογή σε περιβάλλοντα όπου τα δεδομένα δεν είναι πλήρως κεντριοποιημένα [6], [7].

3.3 Προσεγγίσεις βασισμένες σε διαπραγμάτευση

Μία σημαντική γραμμή έρευνας αντιμετωπίζει τον προγραμματισμό συναντήσεων ως πρόβλημα διαπραγμάτευσης μεταξύ πρακτόρων. Στο πλαίσιο αυτό έχουν προταθεί διαφορετικά πρωτόκολλα διαπραγμάτευσης και συντονισμού, στα οποία οι πράκτορες ανταλλάσσουν πληροφορίες σχετικά με τη διαθεσιμότητα, τις προτιμήσεις ή τους περιορισμούς τους [2]. Εξετάζονται περιπτώσεις στις οποίες οι πράκτορες ανταλλάσσουν πλήρη πληροφορία, μόνο προτιμήσεις, μόνο διαθέσιμα χρονικά διαστήματα ή πιο περιορισμένες μορφές πληροφόρησης. Οι προσεγγίσεις αυτές είναι σημαντικές, διότι αναδεικνύουν ότι ο προγραμματισμός συναντήσεων δεν αποτελεί μόνο πρόβλημα εύρεσης κοινού χρόνου, αλλά και πρόβλημα διαχείρισης ιδιωτικότητας, προτιμήσεων και μηχανισμών συντονισμού.

Σε πιο πρόσφατο επίπεδο, η εργασία *Multi-Agent Meeting Scheduling: A Negotiation Perspective* [8] επανατοποθετεί το πρόβλημα ως ένα σύγχρονο πεδίο διαπραγμάτευσης στα πολυπρακτορικά συστήματα. Παρότι δεν προτείνει ολοκληρωμένη υλοποίηση, είναι ιδιαίτερα χρήσιμη επειδή προσφέρει ένα σύγχρονο θεωρητικό πλαίσιο και αναδεικνύει ότι το πρόβλημα παραμένει ερευνητικά ανοιχτό [8].

Σε σχέση με τις παραπάνω εργασίες, η παρούσα εργασία βρίσκεται κοντά στο πνεύμα της κατανεμημένης διαπραγμάτευσης, αλλά διαφοροποιείται ως προς το ότι δεν περιορίζεται στη συλλογή προτιμήσεων ή αξιολογήσεων. Αντίθετα, συνδυάζει φάση συλλογής τοπικών αποκρίσεων με ξεχωριστή διαδικασία τελικής οριστικοποίησης.

3.4 Κεντριοποιημένες και κατανεμημένες προσεγγίσεις προγραμματισμού συναντήσεων

Ο προγραμματισμός συναντήσεων μπορεί να υλοποιηθεί είτε μέσω κεντριοποιημένων είτε μέσω κατανεμημένων αρχιτεκτονικών. Στις κεντριοποιημένες προσεγγίσεις, ένας κεντρικός scheduler διαθέτει συνολική εικόνα των ημερολογίων και αναλαμβάνει τη διαδικασία λήψης αποφάσεων σχετικά με την επιλογή κατάλληλων χρονικών διαστημάτων. Η προσέγγιση αυτή απλοποιεί τον συντονισμό και επιτρέπει άμεση πρόσβαση σε συγκεντρωμένη πληροφορία, ωστόσο παρουσιάζει σημαντικούς περιορισμούς ως προς την επεκτασιμότητα, την ανθεκτικότητα και την ιδιωτικότητα των δεδομένων.

Συγκεκριμένα, η ύπαρξη ενός μοναδικού κεντρικού κόμβου δημιουργεί μοναδικό σημείο αστοχίας (single point of failure), καθώς πιθανή αστοχία του επηρεάζει ολόκληρο το σύστημα. Παράλληλα, η αποθήκευση όλων των δεδομένων διαθεσιμότητας σε κεντρική βάση μειώνει την αυτονομία των χρηστών και μπορεί να δημιουργήσει ζητήματα προστασίας προσωπικών δεδομένων.

Αντίθετα, στις κατανεμημένες προσεγγίσεις, κάθε συμμετέχων εκπροσωπείται από έναν αυτόνομο πράκτορα (agent), ο οποίος διατηρεί τοπικά τη γνώση σχετικά με τη διαθεσιμότητα και τις προτιμήσεις του χρήστη [1]. Η διαδικασία λήψης αποφάσεων πραγματοποιείται μέσω ανταλλαγής μηνυμάτων και μηχανισμών διαπραγμάτευσης μεταξύ των πρακτόρων, χωρίς να απαιτείται πλήρης κεντριοποίηση της πληροφορίας [2], [8].

Οι πολυπρακτορικές αρχιτεκτονικές θεωρούνται ιδιαίτερα κατάλληλες για το πρόβλημα του προγραμματισμού συναντήσεων, καθώς επιτρέπουν αποκεντρωμένη λήψη αποφάσεων, τοπικό έλεγχο δεδομένων και καλύτερη προσαρμογή σε δυναμικά περιβάλλοντα [6], [7]. Επιπλέον, προσφέρουν αυξημένη ανθεκτικότητα, αφού η αποτυχία ενός πράκτορα δεν οδηγεί απαραίτητα σε συνολική κατάρρευση του συστήματος.

Ωστόσο, οι κατανεμημένες προσεγγίσεις εισάγουν αυξημένο κόστος συντονισμού, καθώς απαιτείται ανταλλαγή μεγάλου αριθμού μηνυμάτων μεταξύ των πρακτόρων. Επιπλέον, η ανάγκη διατήρησης συνέπειας (consistency) μεταξύ των συμμετεχόντων καθιστά απαραίτητη τη χρήση μηχανισμών συγχρονισμού και πρωτοκόλλων οριστικοποίησης αποφάσεων.

Η παρούσα εργασία υιοθετεί κατανεμημένη πολυπρακτορική αρχιτεκτονική βασισμένη στο SPADE και στο πρωτόκολλο XMPP, όπου κάθε χρήστης και κάθε χώρος συνάντησης εκπροσωπείται από ανεξάρτητο πράκτορα. Ο συντονισμός πραγματοποιείται μέσω ανταλλαγής μηνυμάτων και τοπικής αξιολόγησης υποψήφιων χρονικών διαστημάτων από τους πράκτορες. Παράλληλα, η τελική οριστικοποίηση της απόφασης βασίζεται σε λογική πρωτοκόλλου δύο φάσεων δέσμευσης (Two-Phase Commit – 2PC), με στόχο τη διατήρηση συνέπειας κατά τη διαδικασία προγραμματισμού.

Σε αντίθεση με πολλές σύγχρονες εφαρμογές προγραμματισμού συναντήσεων που βασίζονται σε πλήρως κεντριοποιημένη διαχείριση διαθεσιμότητας, η παρούσα προσέγγιση υιοθετεί κατανεμημένη πολυπρακτορική αρχιτεκτονική, στην οποία οι συμμετέχοντες εκπροσωπούνται από αυτόνομους πράκτορες που αξιολογούν τοπικά τις διαθέσιμες επιλογές και συμμετέχουν ενεργά στη διαδικασία λήψης απόφασης μέσω ανταλλαγής μηνυμάτων. Επιπλέον, συνδυάζει τον κατανεμημένο συντονισμό με διεπαφή φυσικής γλώσσας μέσω Μεγάλου Γλωσσικού Μοντέλου, δημιουργώντας ένα υβριδικό σύστημα το οποίο ενσωματώνει τόσο συμβολικές τεχνικές συντονισμού όσο και νευρωνικές τεχνικές επεξεργασίας φυσικής γλώσσας.

3.5 Πρωτόκολλα συντονισμού και συνέπειας σε καταναμεμημένα συστήματα

Στα καταναμεμημένα πολυπρακτορικά συστήματα, ο συντονισμός μεταξύ ανεξάρτητων οντοτήτων αποτελεί κρίσιμο ζήτημα, ιδιαίτερα όταν απαιτείται λήψη κοινής απόφασης σχετικά με μία ενέργεια που επηρεάζει πολλούς συμμετέχοντες. Στην περίπτωση του προγραμματισμού συναντήσεων, η ανάγκη αυτή εμφανίζεται κατά την επιλογή και οριστικοποίηση ενός κοινού χρονικού διαστήματος, το οποίο πρέπει να γίνει αποδεκτό από όλους τους εμπλεκόμενους πράκτορες.

Η διαδικασία αυτή σχετίζεται άμεσα με την έννοια της συνέπειας (consistency) στα καταναμεμημένα συστήματα. Συγκεκριμένα, είναι απαραίτητο να αποφεύγονται καταστάσεις στις οποίες διαφορετικοί πράκτορες διαθέτουν διαφορετική αντίληψη για την τελική κατάσταση της διαδικασίας προγραμματισμού. Για παράδειγμα, ένα χρονικό διάστημα δεν θα πρέπει να θεωρείται δεσμευμένο από ορισμένους πράκτορες και ταυτόχρονα διαθέσιμο από άλλους.

Για την αντιμετώπιση τέτοιων προβλημάτων χρησιμοποιούνται πρωτόκολλα συντονισμού και συμφωνίας (coordination and agreement protocols). Μία από τις πιο γνωστές κατηγορίες είναι τα πρωτόκολλα συναλλαγών (transaction-oriented protocols), τα οποία χρησιμοποιούνται ευρέως σε καταναμεμημένα πληροφοριακά συστήματα και βάσεις δεδομένων [11].

Ένα από τα σημαντικότερα πρωτόκολλα αυτής της κατηγορίας είναι το πρωτόκολλο δύο φάσεων δέσμευσης (Two-Phase Commit – 2PC). Το πρωτόκολλο αυτό οργανώνει τη διαδικασία λήψης απόφασης σε δύο διακριτές φάσεις [11]:

- Φάση προετοιμασίας (prepare phase)
- Φάση οριστικοποίησης (commit/abort phase)

Κατά τη φάση προετοιμασίας (prepare phase), ένας συντονιστής αποστέλλει αίτημα στους συμμετέχοντες κόμβους προκειμένου να αξιολογήσουν αν μπορούν να αποδεχθούν μία προτεινόμενη ενέργεια. Κάθε συμμετέχων επιστρέφει θετική ή αρνητική ψήφο (YES/NO vote), ανάλογα με την τοπική του κατάσταση.

Στη συνέχεια, κατά τη φάση οριστικοποίησης (commit phase), εφόσον πληρούνται οι απαιτούμενες συνθήκες συμφωνίας, ο συντονιστής αποστέλλει εντολή οριστικοποίησης της ενέργειας. Αντίθετα, σε περίπτωση αποτυχίας ή αρνητικής ψήφου, η διαδικασία ακυρώνεται, αποτρέποντας την εμφάνιση μη συνεπών καταστάσεων.

Η λογική αυτή είναι ιδιαίτερα κατάλληλη για προβλήματα προγραμματισμού συναντήσεων, καθώς επιτρέπει σε κάθε πράκτορα να διατηρεί τοπικό έλεγχο της διαθεσιμότητάς του, ενώ ταυτόχρονα εξασφαλίζεται ότι η τελική απόφαση λαμβάνεται συλλογικά και με συνέπεια.

Παράλληλα με το 2PC, σημαντικό ρόλο στα καταναμεμημένα συστήματα διαδραματίζουν και οι μηχανισμοί απαρτίας. Τα συστήματα απαρτίας βασίζονται στην ιδέα ότι μία απόφαση μπορεί να θεωρηθεί αποδεκτή όταν συγκεντρώνεται ένας ελάχιστος αριθμός θετικών απαντήσεων από τους συμμετέχοντες. Η προσέγγιση αυτή χρησιμοποιείται συχνά σε καταναμεμημένα περιβάλλοντα ώστε να επιτυγχάνεται ισορροπία μεταξύ αξιοπιστίας και απόδοσης.

Στην παρούσα εργασία, ο πράκτορας συντονισμού αναλαμβάνει τον συντονισμό της διαδικασίας προγραμματισμού. Αρχικά συλλέγονται οι τοπικές αξιολογήσεις των υποψήφιων χρονικών διαστημάτων από τους πράκτορες ημερολογίου, ενώ στη συνέχεια πραγματοποιείται η τελική οριστικοποίηση της απόφασης μόνο όταν ικανοποιούνται οι απαιτήσεις απαρτίας. Με τον τρόπο αυτό αποφεύγονται μη συνεπείς καταστάσεις και διασφαλίζεται ότι όλοι οι συμμετέχοντες διαθέτουν κοινή εικόνα για το αποτέλεσμα της διαδικασίας προγραμματισμού.

Η χρήση τέτοιων μηχανισμών συντονισμού επιτρέπει στο σύστημα να λειτουργεί με μεγαλύτερη αξιοπιστία σε κατανεμημένα περιβάλλοντα, όπου δεν υπάρχει καθολική κοινόχρηστη μνήμη ή κεντρική βάση δεδομένων που να ελέγχει άμεσα τη συνοχή των αποφάσεων.

3.6 Σύγχρονες προσεγγίσεις με υποστήριξη μεγάλων γλωσσικών μοντέλων

Τα τελευταία χρόνια εμφανίζονται συστήματα τα οποία ενσωματώνουν Μεγάλα Γλωσσικά Μοντέλα σε πολυπρακτορικά ή, γενικότερα, περιβάλλοντα βασισμένα σε πράκτορες. Μία χαρακτηριστική περίπτωση είναι το *ScheduleMe: Multi-Agent Calendar Assistant* [9], το οποίο υλοποιεί έναν βοηθό ημερολογίου που αξιοποιεί πολλαπλούς πράκτορες για τη διαχείριση γεγονότων και τη συνεργασία με τον χρήστη μέσω φυσικής γλώσσας [9]. Τέτοιες προσεγγίσεις δείχνουν τη στροφή της έρευνας προς συστήματα στα οποία η κατανόηση φυσικής γλώσσας και η διάσπαση λειτουργιών σε επιμέρους ρόλους συνδυάζονται σε μία ενιαία ροή λειτουργίας.

Ωστόσο, πολλές από αυτές τις προσεγγίσεις βρίσκονται πιο κοντά σε βοηθούς με κεντρικό συντονισμό παρά σε πλήρως κατανεμημένα συστήματα προγραμματισμού. Το κύριο ενδιαφέρον τους εντοπίζεται στην αλληλεπίδραση με τον χρήστη και στην οργάνωση επιμέρους λειτουργιών, ενώ η παρούσα εργασία δίνει μεγαλύτερο βάρος στον συντονισμό μέσω ανταλλαγής μηνυμάτων και στη διαδικασία λήψης της τελικής απόφασης.

Παράλληλα, πρόσφατες επισκοπήσεις για πολυπρακτορικά συστήματα με υποστήριξη Μεγάλων Γλωσσικών Μοντέλων επιβεβαιώνουν ότι τα μοντέλα αυτά μπορούν να λειτουργούν ως γνωστικά υποσυστήματα, τα οποία ενισχύουν την αλληλεπίδραση, την ερμηνεία αιτημάτων και, σε ορισμένες περιπτώσεις, τον συντονισμό μεταξύ επιμέρους υπομονάδων [3]. Η παρούσα εργασία κινείται σε αυτή τη λογική, αλλά διατηρεί διακριτό τον ρόλο του Μεγάλου Γλωσσικού Μοντέλου από τον ρόλο του βασικού μηχανισμού προγραμματισμού.

3.7 Περιορισμοί και προκλήσεις της χρήσης μεγάλα γλωσσικά μοντέλα στον προγραμματισμό συναντήσεων

Η ενσωμάτωση Μεγάλων Γλωσσικών Μοντέλων (Large Language Models – LLMs) σε συστήματα προγραμματισμού συναντήσεων προσφέρει σημαντικά πλεονεκτήματα ως προς την κατανόηση φυσικής γλώσσας και τη βελτίωση της αλληλεπίδρασης με τον χρήστη. Ωστόσο, η χρήση των μοντέλων αυτών συνοδεύεται και από συγκεκριμένους περιορισμούς και τεχνικές προκλήσεις.

Ένα βασικό χαρακτηριστικό των LLMs είναι ότι δεν λειτουργούν με απόλυτα ντετερμινιστικό τρόπο. Η έξοδος που παράγεται μπορεί να επηρεάζεται από τη διατύπωση του prompt, το μήκος του αιτήματος, καθώς και από εσωτερικούς μηχανισμούς παραγωγής κειμένου του μοντέλου [12]. Ως αποτέλεσμα, παρόμοια αιτήματα ενδέχεται να οδηγήσουν σε ελαφρώς διαφορετικές αποκρίσεις.

Μία από τις σημαντικότερες προκλήσεις είναι το φαινόμενο των ψευδαισθήσεων (hallucinations), κατά το οποίο το μοντέλο παράγει πληροφορία που εμφανίζεται συντακτικά σωστή αλλά δεν ανταποκρίνεται πλήρως στο πραγματικό αίτημα του χρήστη [12]. Στην περίπτωση συστημάτων προγραμματισμού συναντήσεων, το φαινόμενο αυτό μπορεί να οδηγήσει σε:

- λανθασμένη αναγνώριση συμμετεχόντων,
- μη έγκυρες χρονικές αναπαραστάσεις,
- ελλιπή δεδομένα JSON,
- ή δημιουργία πεδίων που δεν προβλέπονται από το schema του συστήματος.

Κατά την υλοποίηση της παρούσας εργασίας παρατηρήθηκε ότι η πιθανότητα μη έγκυρης εξόδου αυξανόταν όταν το prompt περιείχε μεγάλο αριθμό συμμετεχόντων ή συνδυασμό διαφορετικών γλωσσικών μορφών. Σε ορισμένες περιπτώσεις το μοντέλο επέστρεφε ελλιπές JSON ή αλλοιωμένες τιμές σε αναγνωριστικά XMPP, γεγονός που απαιτούσε επιπλέον μηχανισμούς επικύρωσης και κανονικοποίησης.

Για τον λόγο αυτό, το προτεινόμενο σύστημα δεν βασίζεται αποκλειστικά στο Μεγάλο Γλωσσικό Μοντέλο για τη λήψη αποφάσεων προγραμματισμού. Αντίθετα, ακολουθεί υβριδική αρχιτεκτονική, όπου το μοντέλο χρησιμοποιείται μόνο για τη σημασιολογική κατανόηση του αιτήματος, ενώ η τελική αξιολόγηση των περιορισμών και η επιλογή του χρονικού διαστήματος πραγματοποιούνται από ντετερμινιστικούς πράκτορες και μηχανισμούς επικύρωσης [3].

Η προσέγγιση αυτή συνδυάζει τα πλεονεκτήματα της φυσικής γλωσσικής κατανόησης με την αξιοπιστία μηχανισμών που βασίζονται σε κανόνες και κατανεμημένων μηχανισμών προγραμματισμού. Παρόμοιες κατευθύνσεις εμφανίζονται και σε σύγχρονες πολυπρακτορικές αρχιτεκτονικές βοηθών προγραμματισμού συναντήσεων, οι οποίες αξιοποιούν Μεγάλα Γλωσσικά Μοντέλα ως σημασιολογικό επίπεδο επεξεργασίας [3], [9].

Επιπλέον, η χρήση Μεγάλων Γλωσσικών Μοντέλων εισάγει αυξημένο υπολογιστικό κόστος, καθώς η διαδικασία παραγωγής αποκρίσεων (inference) αποτελεί σημαντικό μέρος του συνολικού χρόνου εκτέλεσης του συστήματος. Όπως παρουσιάστηκε και στο Κεφάλαιο 5, το στάδιο μετατροπής φυσικής γλώσσας σε JSON αποτέλεσε το βασικό σημείο συμφόρησης (bottleneck) του συστήματος, καταλαμβάνοντας μεγαλύτερο ποσοστό του συνολικού χρόνου απόκρισης (latency) σε σχέση με τον πράκτορα συντονισμού.

Συνολικά, τα Μεγάλα Γλωσσικά Μοντέλα προσφέρουν σημαντική ευελιξία και βελτιωμένη εμπειρία χρήστη, αλλά απαιτούνται μηχανισμοί ελέγχου, επικύρωσης και συνδυασμού με ντετερμινιστικά υποσυστήματα ώστε να εξασφαλίζεται η αξιοπιστία και η συνέπεια της διαδικασίας προγραμματισμού.

3.8 Συγκριτική τοποθέτηση της παρούσας εργασίας

Με βάση τη σχετική βιβλιογραφία, η παρούσα εργασία δεν παρουσιάζεται ως πλήρως νέα κατηγορία προβλήματος. Ο προγραμματισμός συναντήσεων μέσω πρακτόρων έχει ήδη μελετηθεί τόσο σε κλασικές πολυπρακτορικές προσεγγίσεις όσο και σε προσεγγίσεις βασισμένες σε διαπραγμάτευση [2], [6], [7]. Η διαφοροποίηση της παρούσας εργασίας εντοπίζεται κυρίως στον συγκεκριμένο τρόπο σύνθεσης των επιμέρους στοιχείων.

Πρώτον, η εργασία προτείνει μία κατανεμημένη πολυπρακτορική αρχιτεκτονική βασισμένη σε SPADE/XMPP, στην οποία υπάρχουν διακριτοί ρόλοι για πράκτορα συντονισμού, πράκτορες-μέλη και πράκτορες-χώρων. Δεύτερον, ο μηχανισμός προγραμματισμού δεν περιορίζεται σε έλεγχο απλής διαθεσιμότητας, αλλά αξιοποιεί τοπικές αξιολογήσεις υπονήφγιων χρονικών διαστημάτων από τους επιμέρους πράκτορες. Τρίτον, η διαδικασία απόφασης συνδυάζει μία φάση συλλογής αξιολογήσεων, η οποία προσεγγίζει λογική τύπου Contract Net Protocol [2], με μία ξεχωριστή φάση τελικής οριστικοποίησης. Τέταρτον, η είσοδος του συστήματος υποστηρίζεται από σημασιολογική κατανόηση αιτημάτων σε φυσική γλώσσα με τη βοήθεια Μεγάλου Γλωσσικού Μοντέλου, χωρίς όμως το μοντέλο αυτό να αναλαμβάνει τον ίδιο τον ρόλο του κατανεμημένου συντονισμού [3], [9].

Επομένως, η παρούσα εργασία τοποθετείται ανάμεσα στις κλασικές πολυπρακτορικές προσεγγίσεις για τον προγραμματισμό συναντήσεων και στις πιο πρόσφατες προσεγγίσεις με υποστήριξη Μεγάλων Γλωσσικών Μοντέλων. Πιο συγκεκριμένα, υιοθετεί τον πρακτοροκεντρικό χαρακτήρα και την

κατανεμημένη λογική των πολυπρακτορικών συστημάτων [6], [7], αξιοποιεί το θεωρητικό πλαίσιο της διαπραγμάτευσης που αναδεικνύεται τόσο στις κλασικές όσο και στις σύγχρονες σχετικές εργασίες [2], [8], αλλά ταυτόχρονα ενσωματώνει ένα σύγχρονο σημασιολογικό επίπεδο διεπαφής με υποστήριξη Μεγάλου Γλωσσικού Μοντέλου [3], [9].

Η βιβλιογραφική ανασκόπηση δείχνει ότι ο αυτόματος προγραμματισμός συναντήσεων αποτελεί ενεργό πεδίο έρευνας, στο οποίο συνδυάζονται τεχνικές πολυπρακτορικών συστημάτων, μηχανισμοί διαπραγμάτευσης και, πιο πρόσφατα, τεχνολογίες Μεγάλων Γλωσσικών Μοντέλων. Η παρούσα εργασία αξιοποιεί στοιχεία από τις παραπάνω κατευθύνσεις και προτείνει μία υβριδική πολυπρακτορική αρχιτεκτονική, η οποία παρουσιάζεται αναλυτικά στο επόμενο κεφάλαιο.

Κεφάλαιο 4ο: Αρχιτεκτονική του Συστήματος

4.1 Εισαγωγή

Το προτεινόμενο σύστημα αποτελεί ένα πολυπρακτορικό σύστημα για τον αυτόματο προγραμματισμό συναντήσεων. Η αρχιτεκτονική του συνδυάζει πολυπρακτορικό συντονισμό μέσω ανταλλαγής μηνυμάτων, σημασιολογική επεξεργασία αιτημάτων φυσικής γλώσσας και ενορχήστρωση της συνολικής ροής εκτέλεσης. Βασικός στόχος της σχεδίασης είναι ο σαφής διαχωρισμός ανάμεσα στην κατανόηση του αιτήματος, στον μηχανισμό προγραμματισμού και στην παραγωγή των τελικών αποτελεσμάτων [3], [10].

4.2 Γενική επισκόπηση της αρχιτεκτονικής

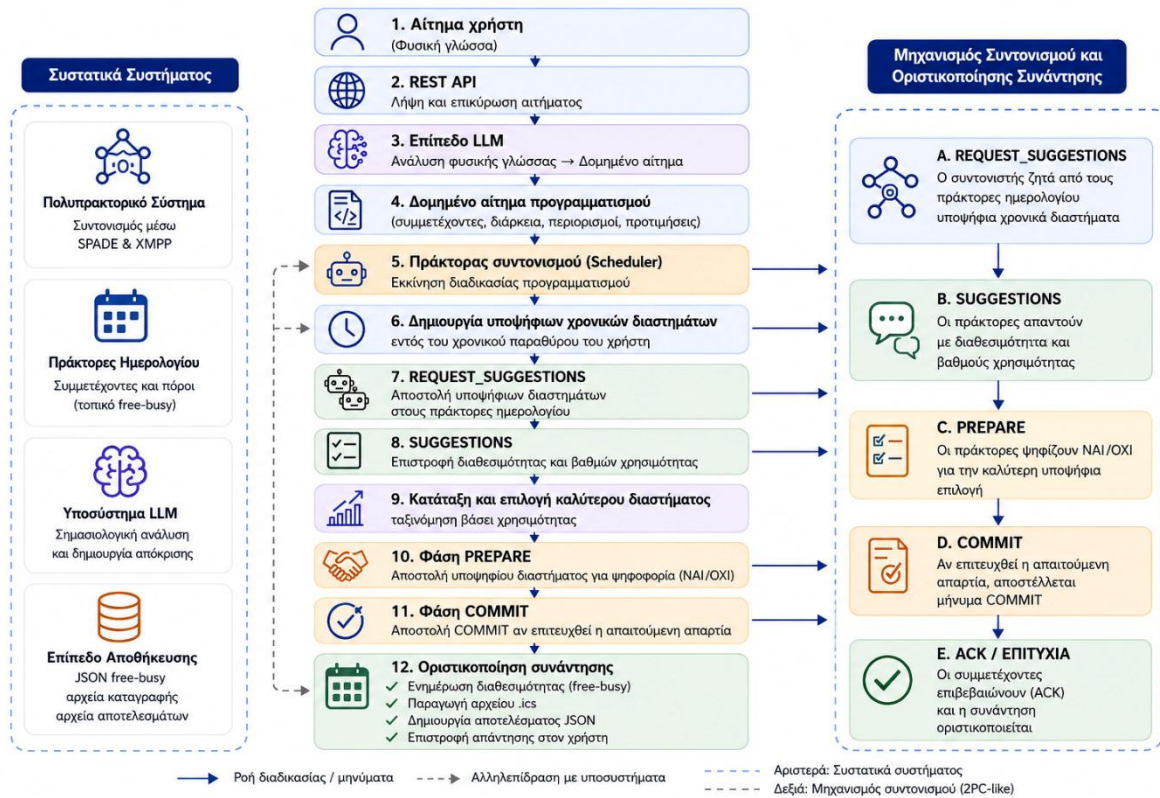
Η συνολική λειτουργία του συστήματος ξεκινά από την υποβολή ενός αιτήματος προγραμματισμού σε φυσική γλώσσα. Το αίτημα αυτό μετατρέπεται σε δομημένη προδιαγραφή, η οποία χρησιμοποιείται για την εκκίνηση της διαδικασίας προγραμματισμού. Στη συνέχεια, ένας πράκτορας συντονισμού ενεργοποιεί τη συνεργασία με τους επιμέρους πράκτορες, συλλέγει τις τοπικές αξιολογήσεις των υποψήφιων χρονικών διαστημάτων και επιχειρεί την οριστικοποίηση της συνάντησης. Μετά την ολοκλήρωση της διαδικασίας, το σύστημα παράγει δομημένο αποτέλεσμα και αρχείο ημερολογίου σε μορφή .ics [4].

Η αρχιτεκτονική οργανώνεται σε τέσσερα βασικά επίπεδα:

- επίπεδο εισόδου και αλληλεπίδρασης με τον χρήστη,
- επίπεδο σημασιολογικής κατανόησης,
- πολυπρακτορικό επίπεδο συντονισμού,
- επίπεδο διαχείρισης δεδομένων και εξαγωγής αποτελεσμάτων.

Ο διαχωρισμός αυτός διευκολύνει την επεκτασιμότητα, την επαναχρησιμοποίηση επιμέρους υποσυστημάτων και την πειραματική αξιολόγηση της συνολικής ροής εκτέλεσης [3], [10].

Η συνολική αρχιτεκτονική του συστήματος παρουσιάζεται στο Σχήμα 4.2.



Σχήμα 4.2: Αρχιτεκτονική του πολυπρακτορικού συστήματος προγραμματισμού συναντήσεων

Πηγή: Ίδια επεξεργασία.

4.3 Πολυπρακτορικό μοντέλο και ρόλοι πρακτόρων

Το σύστημα βασίζεται σε πολυπρακτορικό μοντέλο με σαφώς διακριτούς ρόλους. Οι βασικές οντότητες είναι ο πράκτορας συντονισμού, οι πράκτορες-μέλη και οι πράκτορες-χώρων. Ο πράκτορας συντονισμού έχει την ευθύνη της οργάνωσης της συνολικής διαδικασίας, ενώ οι υπόλοιποι πράκτορες εκπροσωπούν μέλη ή πόρους και συμμετέχουν με βάση την τοπική τους γνώση και τους τοπικούς περιορισμούς τους [6], [7], [10].

Η αρχιτεκτονική αυτή ακολουθεί λογική σχεδίασης βασισμένη σε ρόλους, όπου η συνολική λειτουργία του συστήματος προκύπτει από τη συνεργασία εξειδικευμένων πρακτόρων. Η χρήση του SPADE υποστηρίζει την υλοποίηση πολλαπλών συμπεριφορών και επιτρέπει σαφή αντιστοίχιση ανάμεσα σε ρόλο και λειτουργική ευθύνη [10].

4.4 Μηχανισμός συντονισμού και ροή απόφασης

Ο πυρήνας της αρχιτεκτονικής είναι ο μηχανισμός πολυπρακτορικού συντονισμού, μέσω του οποίου μία δομημένη προδιαγραφή συνάντησης μετατρέπεται σε διαδικασία αξιολόγησης, διαπραγμάτευσης και οριστικοποίησης.

Η βασική ροή περιλαμβάνει:

- παραγωγή υποψήφιων χρονικών διαστημάτων,
- αποστολή αιτημάτων αξιολόγησης προς τους επιμέρους πράκτορες,
- συλλογή τοπικών αποκρίσεων και αξιολογήσεων,

- κατάταξη των εναλλακτικών επιλογών,
- ενεργοποίηση φάσης προετοιμασίας,
- ενεργοποίηση φάσης τελικής δέσμευσης,
- οριστικοποίηση της συνάντησης ή αναφορά αποτυχίας.

Η λογική αυτή βασίζεται στη συλλογή και αξιολόγηση αποκρίσεων από τους συμμετέχοντες πράκτορες και στη συνέχεια στην πολυφασική οριστικοποίηση της τελικής επιλογής [2].

4.5 Πράκτορες συμμετεχόντων και τοπική αξιολόγηση

Οι πράκτορες συμμετεχόντων υλοποιούν την τοπική λογική αξιολόγησης των υποψήφιων χρονικών διαστημάτων. Κάθε τέτοιος πράκτορας διαχειρίζεται πληροφορία σχετική με τη διαθεσιμότητα και τους περιορισμούς της οντότητας που εκπροσωπεί, είτε πρόκειται για μέλος είτε για πόρο.

Η αξιολόγηση μπορεί να λαμβάνει υπόψη περιορισμούς όπως χρονικά διαστήματα μη διαθεσιμότητας, ωράρια εργασίας, αργίες, προτινόμενες ή ήσυχες ώρες, περιορισμούς διάρκειας και κανόνες αποφυγής υπερβολικά πυκνών συναντήσεων. Με βάση τα στοιχεία αυτά, κάθε πράκτορας επιστρέφει πληροφορία αποδοχής ή απόρριψης, καθώς και σχετική ποσοτική αξιολόγηση της καταλληλότητας του χρονικού διαστήματος. Η προσέγγιση αυτή διατηρεί την τοπική γνώση στους επιμέρους πράκτορες και επιτρέπει την ενσωμάτωση ετερογενών περιορισμών στη διαδικασία συντονισμού.

4.6 Γνωστικό υποσύστημα σημασιολογικής κατανόησης

Το γνωστικό επίπεδο του συστήματος βασίζεται σε Μεγάλο Γλωσσικό Μοντέλο και λειτουργεί ως σημασιολογική διεπαφή μεταξύ χρήστη και πολυπρακτορικού περιβάλλοντος. Ο ρόλος του είναι να μετατρέπει αιτήματα φυσικής γλώσσας σε δομημένες προδιαγραφές προγραμματισμού, οι οποίες μπορούν να αποτελέσουν είσοδο για τον μηχανισμό συντονισμού. Η αρχιτεκτονική αυτή διατηρεί διακριτό τον ρόλο του Μεγάλου Γλωσσικού Μοντέλου από τον ρόλο του πολυπρακτορικού μηχανισμού προγραμματισμού. Το μοντέλο αναλαμβάνει την κατανόηση της πρόθεσης του χρήστη και την εξαγωγή των σχετικών παραμέτρων, ενώ η πραγματική απόφαση για την επιλογή και οριστικοποίηση της συνάντησης λαμβάνεται από τους πράκτορες μέσω ανταλλαγής μηνυμάτων.

Κατά τη διαδικασία σημασιολογικής επεξεργασίας εφαρμόζονται επιπλέον έλεγχοι εγκυρότητας στα παραγόμενα δεδομένα. Ενδεικτικά, αιτήματα που αντιστοιχούν σε χρονικά διαστήματα του παρελθόντος απορρίπτονται πριν από την εκκίνηση της διαδικασίας προγραμματισμού. Με τον τρόπο αυτό αποφεύγεται η εκτέλεση μη έγκυρων αιτημάτων προγραμματισμού και ενισχύεται η αξιοπιστία της συνολικής ροής επεξεργασίας.

4.7 Διεπαφή υπηρεσίας και επίπεδο ενορχήστρωσης

Η αλληλεπίδραση με το σύστημα υλοποιείται μέσω διεπαφής REST API, η οποία λειτουργεί ως επίπεδο ενορχήστρωσης ανάμεσα στον χρήστη, στο σημασιολογικό υποσύστημα και στο πολυπρακτορικό επίπεδο εκτέλεσης. Η διεπαφή αυτή δέχεται τα αιτήματα, ενεργοποιεί τη σημασιολογική επεξεργασία, εκκινεί τη διαδικασία προγραμματισμού και δρομολογεί την εκτέλεση του συστήματος.

Η χρήση διακριτού επιπέδου ενορχήστρωσης επιτρέπει τον σαφή διαχωρισμό ανάμεσα στη διεπαφή υπηρεσίας, στη σημασιολογική κατανόηση και στον πολυπρακτορικό συντονισμό. Με τον τρόπο αυτό, η συνολική αρχιτεκτονική παραμένει αρθρωτή και ευκολότερα επεκτάσιμη.

Η εκτέλεση των επιμέρους υποσυστημάτων πραγματοποιείται σε απομονωμένα containers Docker, επιτρέποντας την ανεξάρτητη ανάπτυξη, εκτέλεση και πειραματική αξιολόγηση των πρακτόρων, του API και των υποστηρικτικών υπηρεσιών.

4.8 Υποδομή επικοινωνίας και διαχείριση ημερολογιακών δεδομένων

Η επικοινωνία μεταξύ των πρακτόρων βασίζεται στο SPADE και στο πρωτόκολλο XMPP. Η επιλογή αυτή επιτρέπει ασύγχρονη ανταλλαγή δομημένων μηνυμάτων και σαφή ταυτοποίηση των πρακτόρων στο επικοινωνιακό επίπεδο. Παράλληλα, επιτρέπει την ανάπτυξη και εκτέλεση των επιμέρους πρακτόρων ως διακριτών οντοτήτων, οι οποίες επικοινωνούν μέσω ανταλλαγής μηνυμάτων.

Η αρχιτεκτονική περιλαμβάνει επίσης επίπεδο διαχείρισης ημερολογιακών δεδομένων και παραγωγής αποτελεσμάτων. Το επίπεδο αυτό είναι υπεύθυνο για την αξιοποίηση πληροφοριών διαθεσιμότητας και για την παραγωγή διαλειτουργικών αποτελεσμάτων, όπως αρχεία .ics. Με τον τρόπο αυτό, το σύστημα δεν περιορίζεται σε ένα εσωτερικό πρωτότυπο, αλλά διατηρεί τη δυνατότητα διασύνδεσης με πραγματικά ημερολογιακά οικοσυστήματα που βασίζονται στο πρότυπο iCalendar [4].

4.9 Βασικές αρχιτεκτονικές ιδιότητες του συστήματος

Η αρχιτεκτονική του συστήματος σχεδιάζεται έτσι ώστε να υποστηρίζει τρεις βασικές ιδιότητες: λήψη αποφάσεων με λογική απαρτίας, συνέπεια κατά την οριστικοποίηση και ερμηνευσιμότητα των αποτελεσμάτων.

Η λογική απαρτίας επιτρέπει την ολοκλήρωση της διαδικασίας χωρίς να απαιτείται η σύμφωνη γνώμη όλων των συμμετεχόντων, ενώ ταυτόχρονα υποστηρίζει αυστηρότερους περιορισμούς όταν συγκεκριμένα μέλη ή πόροι θεωρούνται απαραίτητοι. Η συνέπεια εξασφαλίζεται μέσω του διαχωρισμού ανάμεσα στη φάση προετοιμασίας και στη φάση τελικής δέσμευσης, έτσι ώστε η τελική επιλογή να οριστικοποιείται μόνο όταν οι απαιτούμενες προϋποθέσεις εξακολουθούν να ισχύουν. Τέλος, η αρχιτεκτονική υποστηρίζει την ερμηνευσιμότητα των αποτελεσμάτων, καθώς μπορεί να διατηρεί πληροφωρία σχετικά με απορρίψεις, εναλλακτικές επιλογές και αιτίες αποτυχίας της διαδικασίας.

Συνολικά, η αρχιτεκτονική του προτεινόμενου συστήματος συνδυάζει πολυπρακτορικό συντονισμό, σημασιολογική κατανόηση, διαδικτυακή ενορχήστρωση και παραγωγή διαλειτουργικών αποτελεσμάτων, αποτελώντας τη βάση πάνω στην οποία αναπτύσσεται η υλοποίηση και η πειραματική αξιολόγηση που ακολουθούν.

Κεφάλαιο 5ο: Υλοποίηση

5.1 Εισαγωγή

Το κεφάλαιο αυτό παρουσιάζει την υλοποίηση του προτεινόμενου συστήματος και περιγράφει τα βασικά υποσυστήματα καθώς και τον τρόπο με τον οποίο συνεργάζονται για την επίτευξη καταναμεμημένου προγραμματισμού συναντήσεων. Η υλοποίηση ακολουθεί την αρχιτεκτονική που παρουσιάστηκε στο προηγούμενο κεφάλαιο και βασίζεται σε συνδυασμό τεχνολογιών πρακτόρων, διαδικτυακών υπηρεσιών, εκτέλεσης σε κοντέινερ και σημασιολογικής επεξεργασίας με χρήση Μεγάλου Γλωσσικού Μοντέλου.

5.2 Περιβάλλον υλοποίησης και τεχνολογίες

Η υλοποίηση του συστήματος πραγματοποιήθηκε σε περιβάλλον Docker, ώστε τα επιμέρους υποσυστήματα να εκτελούνται απομονωμένα και να είναι δυνατή η προσομοίωση καταναμεμημένου περιβάλλοντος. Η επιλογή αυτή επιτρέπει την ανεξάρτητη εκτέλεση των πρακτόρων, του διακομιστή XMPP, του API και των υποστηρικτικών υπηρεσιών.

Οι βασικές τεχνολογίες που χρησιμοποιήθηκαν είναι οι εξής: Python 3.11, SPADE για την υλοποίηση των πρακτόρων, XMPP ως πρωτόκολλο επικοινωνίας, Prosody ως εξυπηρετητής XMPP, FastAPI για την υλοποίηση του REST API, Ollama για την τοπική εκτέλεση του Μεγάλου Γλωσσικού Μοντέλου, καθώς και Docker και Docker Compose για την ανάπτυξη και εκτέλεση των επιμέρους υπηρεσιών.

Το SPADE χρησιμοποιήθηκε ως πλαίσιο ανάπτυξης των πρακτόρων του συστήματος. Υποστηρίζει την υλοποίηση αυτόνομων πρακτόρων με διακριτές συμπεριφορές, ασύγχρονη επικοινωνία και ενσωμάτωση με το πρωτόκολλο XMPP, στοιχεία που το καθιστούν κατάλληλο για την ανάπτυξη του πολυπρακτορικού μηχανισμού συντονισμού της παρούσας εργασίας [10].

Το XMPP χρησιμοποιήθηκε ως πρωτόκολλο επικοινωνίας μεταξύ των πρακτόρων, ενώ ο Prosody λειτουργεί ως εξυπηρετητής XMPP. Η επιλογή αυτή υποστηρίζει την ανταλλαγή δομημένων μηνυμάτων και τη σαφή ταυτοποίηση των πρακτόρων στο επικοινωνιακό επίπεδο. Η παραγωγή ημερολογιακών δεδομένων βασίζεται στο πρότυπο iCalendar, μέσω του οποίου δημιουργούνται αρχεία .ics για την αναπαράσταση των προγραμματισμένων συναντήσεων [4].

Το FastAPI χρησιμοποιήθηκε για την ανάπτυξη της διαδικτυακής διεπαφής του συστήματος. Μέσω αυτού υλοποιήθηκαν τα σημεία πρόσβασης (endpoints) που δέχονται αιτήματα φυσικής γλώσσας, ενεργοποιούν τη σημασιολογική επεξεργασία και δρομολογούν την εκτέλεση του μηχανισμού προγραμματισμού. Με τον τρόπο αυτό, το API λειτουργεί ως ενδιάμεσο επίπεδο μεταξύ του χρήστη, του γλωσσικού μοντέλου και του πολυπρακτορικού μηχανισμού προγραμματισμού.

Το Ollama χρησιμοποιήθηκε για την τοπική εκτέλεση του Μεγάλου Γλωσσικού Μοντέλου που υποστηρίζει τη σημασιολογική κατανόηση των αιτημάτων φυσικής γλώσσας. Η επιλογή του επέτρεψε την ενσωμάτωση γλωσσικού μοντέλου σε ελεγχόμενο τοπικό περιβάλλον, χωρίς εξάρτηση από εξωτερικές υπηρεσίες. Στην παρούσα εργασία χρησιμοποιήθηκε το μοντέλο qwen2.5:3b, το οποίο αξιοποιήθηκε στα πειράματα που αφορούσαν τη μετατροπή αιτημάτων φυσικής γλώσσας σε δομημένη μορφή. Η επιλογή του συγκεκριμένου μοντέλου βασίστηκε στη δυνατότητα τοπικής εκτέλεσης σε υποδομή με περιορισμένους υπολογιστικούς πόρους, διατηρώντας παράλληλα ικανοποιητική απόδοση για τις ανάγκες της σημασιολογικής επεξεργασίας. Η χρήση Μεγάλων Γλωσσικών Μοντέλων σε πολυπρακτορικά συστήματα εντάσσεται σε μια ευρύτερη σύγχρονη ερευνητική κατεύθυνση, όπου τα

γλωσσικά μοντέλα λειτουργούν ως σημασιολογικά υποσυστήματα για την κατανόηση αιτημάτων και τη διασύνδεση ανθρώπου-συστήματος [3], [9], [12].

Η επιλογή τοπικής εκτέλεσης του γλωσσικού μοντέλου κρίθηκε επίσης κατάλληλη για το εξεταζόμενο σενάριο χρήσης, καθώς τα αιτήματα προγραμματισμού ενδέχεται να περιλαμβάνουν πληροφορίες σχετικές με τη διαθεσιμότητα και τις δραστηριότητες των συμμετεχόντων. Η προσέγγιση αυτή περιορίζει την ανάγκη διαβίβασης δεδομένων σε εξωτερικές υπηρεσίες και διατηρεί μεγαλύτερο έλεγχο επί του περιβάλλοντος επεξεργασίας.

Η χρήση του Docker και του Docker Compose επιτρέπει την απομονωμένη εκτέλεση των επιμέρους υποσυστημάτων, ενισχύοντας την αναπαραγωγιμότητα, την ευκολία ανάπτυξης και την προσομοίωση κατανεμημένου πολυπρακτορικού περιβάλλοντος. Κάθε πράκτορας εκτελείται σε ξεχωριστό container, γεγονός που επιτρέπει την ανεξάρτητη παραμετροποίηση, εκκίνηση και παρακολούθησή του.

Για σκοπούς παρακολούθησης και debugging είχαν προβλεφθεί μηχανισμοί συλλογής μετρικών μέσω Prometheus-compatible endpoints, χωρίς όμως να αποτελέσουν βασικό μέρος της πειραματικής αξιολόγησης της παρούσας εργασίας.

5.3 Δομή του έργου

Η υλοποίηση οργανώνεται σε διακριτά υποσυστήματα, ώστε να υποστηρίζεται σαφής διαχωρισμός ευθυνών και ευκολότερη συντήρηση του κώδικα. Ο βασικός κατάλογος `api/` περιλαμβάνει το επίπεδο REST και τη λογική σημασιολογικής επεξεργασίας. Ο κατάλογος `src/core/` περιλαμβάνει τον πολυπρακτορικό πυρήνα του συστήματος, δηλαδή τη λογική εκκίνησης των πρακτόρων, τη φόρτωση παραμέτρων και τις βασικές συμπεριφορές του συντονιστή και των πρακτόρων ημερολογίου. Ο κατάλογος `src/adapters/` περιλαμβάνει τα υποσυστήματα διασύνδεσης με την αποθήκευση δεδομένων διαθεσιμότητας και την παραγωγή τελικών αρχείων.

Παράλληλα, οι κατάλογοι `config/`, `data/`, `out/` και `jobs/` χρησιμοποιούνται για την παραμετροποίηση του συστήματος, τη διαχείριση ημερολογιακών δεδομένων, την εξαγωγή αποτελεσμάτων και την αποθήκευση των δεδομένων κάθε εκτέλεσης. Η οργανωτική αυτή δομή αντανακλά άμεσα την αρχιτεκτονική του συστήματος, καθώς διαχωρίζει το σημασιολογικό επίπεδο, το πολυπρακτορικό επίπεδο, το επίπεδο προσαρμογών και το επίπεδο εκτέλεσης.

5.4 Υλοποίηση πρακτόρων με SPADE

Η πολυπρακτορική υλοποίηση βασίζεται στο SPADE, όπου κάθε πράκτορας εκτελείται ως οντότητα με δυνατότητα επικοινωνίας μέσω XMPP και μπορεί να φορτώνει διαφορετικές συμπεριφορές ανάλογα με τον ρόλο του. Η βασική κλάση πράκτορα υποστηρίζει ενεργοποίηση συμπεριφορών ανά ρόλο, επιτρέποντας την υλοποίηση τόσο της λογικής συντονισμού όσο και της λογικής συμμετοχής μέσα στο ίδιο γενικό υπόστρωμα.

Οι βασικοί τύποι πρακτόρων που υλοποιούνται είναι:

- ο πράκτορας συντονισμού,
- οι πράκτορες-μέλη,
- οι πράκτορες-χώρων.

Η υλοποίηση αυτή επιτρέπει ασύγχρονη ανταλλαγή μηνυμάτων, τοπική λήψη αποφάσεων και καθαρή αντιστοίχιση μεταξύ λογικού ρόλου και συμπεριφοράς πράκτορα.

5.5 Υλοποίηση του πυρήνα προγραμματισμού

Ο μηχανισμός προγραμματισμού υλοποιείται μέσω της συμπεριφοράς SchedulerBehaviour. Η συμπεριφορά αυτή αναλαμβάνει ολόκληρη τη ροή συντονισμού μίας συνάντησης, από την παραγωγή υποψήφιων χρονικών διαστημάτων μέχρι την οριστικοποίηση ή την αποτυχία της διαδικασίας. Η διαδικασία περιλαμβάνει τα εξής στάδια:

1. παραγωγή υποψήφιων χρονικών διαστημάτων μέσα στο χρονικό παράθυρο,
2. αποστολή αιτημάτων αξιολόγησης προς τους λοιπούς πράκτορες,
3. λήψη αποκρίσεων με δεδομένα διαθεσιμότητας και τιμές αξιολόγησης,
4. ταξινόμηση των υποψήφιων επιλογών,
5. φάση προετοιμασίας με λήψη ψήφων,
6. φάση τελικής δέσμευσης με λήψη επιβεβαιώσεων,
7. τελική παραγωγή αποτελέσματος.

Η λογική αυτή συνδυάζει μία φάση συλλογής και αξιολόγησης υποψήφιων επιλογών με μία φάση οριστικοποίησης δύο σταδίων. Παράλληλα, υποστηρίζονται όρια απαρτίας, υποχρεωτικοί συμμετέχοντες, επαναλήψεις προσπαθειών και χρονικά όρια, ώστε το σύστημα να είναι περισσότερο ανθεκτικό σε καθυστερήσεις ή προσωρινές αποτυχίες.

5.6 Υλοποίηση πρακτόρων ημερολογίου και διαθεσιμότητας

Οι πράκτορες συμμετεχόντων υλοποιούνται μέσω της συμπεριφοράς CalendarBehaviour. Κύριος ρόλος τους είναι η τοπική αξιολόγηση των υποψήφιων χρονικών διαστημάτων και η συμμετοχή στις φάσεις προετοιμασίας και οριστικοποίησης.

Κατά την εκκίνηση, κάθε πράκτορας ημερολογίου ανακτά τα ήδη δεσμευμένα χρονικά διαστήματα από το τοπικό αποθετήριο διαθεσιμότητας και τα διατηρεί στη μνήμη για γρήγορους ελέγχους επικάλυψης. Για κάθε χρονικό διάστημα που λαμβάνει, εξετάζει τοπικούς περιορισμούς όπως:

- επικαλύψεις με ήδη δεσμευμένο χρόνο,
- ωράρια εργασίας,
- αργίες,
- προτιμώμενες ή ήσυχες ώρες,
- όρια διάρκειας,
- περιορισμούς διαδοχικών συναντήσεων.

Με βάση τα στοιχεία αυτά, ο πράκτορας επιστρέφει ένδειξη αποδοχής ή απόρριψης και αντίστοιχη ποσοτική αξιολόγηση. Κατά τη φάση προετοιμασίας αποστέλλει ψήφο αποδοχής ή απόρριψης, ενώ κατά τη φάση τελικής δέσμευσης επιβεβαιώνει την οριστικοποίηση μόνο εφόσον το χρονικό διάστημα παραμένει αποδεκτό. Μετά από επιτυχή ολοκλήρωση, ενημερώνει το τοπικό αποθετήριο διαθεσιμότητας. Η υλοποίηση περιλαμβάνει επίσης μηχανισμό αποφυγής διπλών ενεργειών, ώστε να αποφεύγονται διπλές ψήφοι ή πολλαπλές επιβεβαιώσεις για το ίδιο αίτημα.

5.7 Ενσωμάτωση του Μεγάλου Γλωσσικού Μοντέλου στο API

Η είσοδος του χρήστη δίνεται σε φυσική γλώσσα και μετατρέπεται σε δομημένη μορφή μέσω διαδικασίας που βασίζεται σε Μεγάλο Γλωσσικό Μοντέλο. Το επίπεδο αυτό υλοποιείται στο API και χρησιμοποιεί το Ollama για την τοπική εκτέλεση του μοντέλου. Στην παρούσα εργασία χρησιμοποιήθηκε το μοντέλο qwen2.5:3b, το οποίο αξιοποιήθηκε σε όλα τα πειράματα που περιλάμβαναν σημασιολογική επεξεργασία αιτημάτων.

Η ροή επεξεργασίας περιλαμβάνει:

1. προεπεξεργασία του αιτήματος,
2. κλήση του γλωσσικού μοντέλου,
3. εξαγωγή δομημένης απάντησης σε μορφή JSON,
4. έλεγχο εγκυρότητας,
5. κανονικοποίηση των πεδίων.

Η προεπεξεργασία εφαρμόζει κανόνες κανονικοποίησης σε συνηθισμένες χρονικές εκφράσεις, όπως «αύριο 16» ή «το απόγευμα», ώστε να μειώνονται οι αμφισημίες πριν από την κλήση του μοντέλου και να παράγονται πιο συνεπή δομημένα αποτελέσματα. Στη συνέχεια, το μοντέλο καλείται με αυστηρά ορισμένο σύστημα οδηγιών, ώστε να επιστρέφει δομημένη έξοδο σε μορφή JSON. Ακολουθεί εξαγωγή της απάντησης και εφαρμογή σημασιολογικών ελέγχων, όπως έλεγχος εγκυρότητας του χρονικού παραθύρου, επιβεβαίωση ύπαρξης συμμετεχόντων, διαχείριση αιτημάτων με σταθερή ώρα έναρξης και απόρριψη αιτημάτων που αναφέρονται σε παρελθοντικό χρόνο.

Το τελικό αποτέλεσμα του σταδίου αυτού είναι μια δομημένη αναπαράσταση του αιτήματος προγραμματισμού, η οποία λειτουργεί ως ενδιάμεση αναπαράσταση της αρχικής φυσικής γλώσσας. Η χρήση του Μεγάλου Γλωσσικού Μοντέλου περιορίζεται στη σημασιολογική κατανόηση και στην παραγωγή δομημένης εισόδου και δεν επεκτείνεται στη λήψη της τελικής απόφασης.

5.8 Εκτέλεση βάσει εργασιών και ενορχήστρωση

Το API υλοποιείται με FastAPI και εκθέτει δύο βασικά endpoints:

- /nl2job, για μετατροπή φυσικής γλώσσας σε δομημένο αίτημα,
- /nl2schedule, για πλήρη εκτέλεση του μηχανισμού.

Το endpoint /nl2schedule λειτουργεί ως επίπεδο ενορχήστρωσης της συνολικής ροής. Συγκεκριμένα, αναλαμβάνει τη λήψη και επικύρωση του αιτήματος, τη μετατροπή του σε δομημένο αντικείμενο προγραμματισμού, τη δημιουργία του αντίστοιχου περιβάλλοντος εκτέλεσης, την εκκίνηση του συντονιστή, την ανάγνωση των παραγόμενων αποτελεσμάτων και την επιστροφή της τελικής απόκρισης προς τον χρήστη.

Η συνολική ροή μπορεί να συνοψιστεί ως εξής: φυσική γλώσσα, επικύρωση αιτήματος, μετατροπή σε δομημένο αντικείμενο, εκτέλεση του μηχανισμού προγραμματισμού, ανάγνωση αποτελεσμάτων, παραγωγή τελικής απόκρισης και καταγραφή μετρικών.

Η εκτέλεση ακολουθεί λογική βασισμένη σε εργασίες. Για κάθε αίτημα δημιουργούνται μοναδικά αναγνωριστικά και ξεχωριστός κατάλογος εργασίας. Στο περιβάλλον εκτέλεσης αποθηκεύονται τα

απαραίτητα αρχεία εισόδου, κατάστασης, αποτελέσματος και καταγραφής, γεγονός που ενισχύει την αναπαραγωγιμότητα, την ιχνηλασιμότητα και τη δυνατότητα πειραματικής αξιολόγησης του συστήματος.

Για κάθε εκτέλεση καταγράφονται επίσης μετρικές που αφορούν τον χρόνο σημασιολογικής επεξεργασίας, τον χρόνο εκτέλεσης του scheduler και τον συνολικό χρόνο απόκρισης. Οι πληροφορίες αυτές αξιοποιήθηκαν στην πειραματική αξιολόγηση του συστήματος και στην ανάλυση της απόδοσης των επιμέρους υποσυστημάτων.

5.9 Παραγόμενα αρχεία και αποτελέσματα εκτέλεσης

Το σύστημα παράγει πολλαπλά αρχεία, τα οποία καταγράφουν τόσο το τελικό αποτέλεσμα όσο και την ίδια τη διαδικασία εκτέλεσης.

Το `result.json` περιλαμβάνει, σε περίπτωση επιτυχούς εκτέλεσης, το επιλεγμένο χρονικό διάστημα, τη διάρκεια, τους συμμετέχοντες, τις επιβεβαιώσεις, το όριο απαρτίας, τους υποχρεωτικούς συμμετέχοντες και τη διαδρομή του παραγόμενου αρχείου ημερολογίου. Σε περίπτωση αποτυχίας, περιλαμβάνει λόγο αποτυχίας, υποψήφιες επιλογές που εξετάστηκαν και πληροφορίες σχετικά με τις απορρίψεις.

Το `status.json` χρησιμοποιείται ως συνοπτική αναπαράσταση της κατάστασης, ενώ το `log.txt` καταγράφει τη ροή εκτέλεσης του συστήματος. Παράλληλα, παράγεται αρχείο ημερολογίου σε μορφή `.ics`, συμβατό με το πρότυπο `iCalendar` [4].

Η παραγωγή αυτών των αρχείων επιτρέπει όχι μόνο την επιστροφή αποτελέσματος προς τον χρήστη, αλλά και την αναλυτική μελέτη της συμπεριφοράς του πρωτοτύπου κατά την πειραματική αξιολόγηση.

5.10 Μετρικές και παρακολούθηση συστήματος

Η υλοποίηση περιλαμβάνει μηχανισμούς παρακολούθησης τόσο σε επίπεδο συντονιστή όσο και σε επίπεδο συνολικής ροής εκτέλεσης. Στους πράκτορες χρησιμοποιούνται μετρητές για γεγονότα όπως οι φάσεις προετοιμασίας, οι φάσεις τελικής δέσμευσης, οι επαναλήψεις προσπαθειών και οι ματαιώσεις. Στο API καταγράφονται μετρικές καθυστέρησης για τα στάδια σημασιολογικής επεξεργασίας, εκτέλεσης του μηχανισμού προγραμματισμού και συνολικού χρόνου απόκρισης.

Συνολικά, η υλοποίηση του συστήματος συνδυάζει πολυπρακτορικό προγραμματισμό, σημασιολογική επεξεργασία, διαδικτυακή ενορχήστρωση και εκτέλεση σε κοντέινερ μέσα σε ένα ενιαίο περιβάλλον. Η χρήση SPADE/XMPP υποστηρίζει τον συντονισμό μεταξύ των πρακτόρων, το Μεγάλο Γλωσσικό Μοντέλο επιτρέπει φυσική αλληλεπίδραση και η λογική εκτέλεσης βάσει εργασιών ενισχύει την επαναληψιμότητα και την πειραματική αξιολόγηση του πρωτοτύπου. Η υλοποίηση αυτή αποτέλεσε τη βάση για την πειραματική αξιολόγηση που παρουσιάζεται στο επόμενο κεφάλαιο, όπου εξετάζονται η λειτουργικότητα, η απόδοση και η κλιμάκωση του προτεινόμενου συστήματος.

Ενδεικτικά prompts, παραδείγματα API requests, structured JSON outputs και οδηγίες εκτέλεσης παρατίθενται στα Παραρτήματα Α–Δ.

Κεφάλαιο 6ο: Πειραματική Αξιολόγηση

6.1 Εισαγωγή

Το κεφάλαιο αυτό παρουσιάζει την πειραματική αξιολόγηση του προτεινόμενου συστήματος. Στόχος της αξιολόγησης είναι η διερεύνηση της λειτουργικότητας, της ορθότητας, της απόδοσης και της κλιμάκωσης της πολυπρακτορικής προσέγγισης σε διαφορετικά σενάρια προγραμματισμού συναντήσεων. Η ανάλυση επικεντρώνεται τόσο στη συμπεριφορά του κατανεμημένου μηχανισμού συντονισμού όσο και στο συνολικό σύστημα, το οποίο περιλαμβάνει τη μετατροπή αιτημάτων φυσικής γλώσσας σε δομημένες προδιαγραφές και την εκτέλεση του μηχανισμού προγραμματισμού.

6.2 Στόχοι της αξιολόγησης

Η πειραματική αξιολόγηση σχεδιάστηκε ώστε να εξετάσει τα ακόλουθα βασικά ερωτήματα:

- αν το σύστημα μπορεί να εντοπίζει έγκυρες λύσεις όταν αυτές υπάρχουν,
- αν διαχειρίζεται σωστά χρονικούς και ημερολογιακούς περιορισμούς,
- αν αποτυγχάνει με ελεγχόμενο τρόπο όταν δεν υπάρχει διαθέσιμη λύση,
- πώς επηρεάζεται ο χρόνος εκτέλεσης από τον αριθμό των συμμετεχόντων,
- ποιο υποσύστημα συμβάλλει περισσότερο στο συνολικό χρόνο απόκρισης,
- και κατά πόσο η παράμετρος του βήματος αναζήτησης επηρεάζει την απόδοση του μηχανισμού προγραμματισμού.

Η αξιολόγηση εστιάζει κυρίως στον πειραματικό έλεγχο της αρχιτεκτονικής και της υλοποίησης και όχι στην άμεση σύγκριση με εμπορικά συστήματα προγραμματισμού συναντήσεων.

6.3 Πειραματικό περιβάλλον και μεθοδολογία

Τα πειράματα πραγματοποιήθηκαν στο περιβάλλον υλοποίησης που παρουσιάστηκε στο προηγούμενο κεφάλαιο. Το σύστημα εκτελέστηκε σε περιβάλλον κοντέινερ με Docker Compose, με διακριτές υπηρεσίες για τον πράκτορα συντονισμού, τους πράκτορες συμμετεχόντων, τον εξυπηρετητή XMPP, το REST API και τα εργαλεία παρακολούθησης.

Σε όλα τα πειράματα η εκτέλεση γινόταν μέσω του endpoint `/nl2schedule`, ώστε να αξιολογείται το πλήρες σύστημα από άκρο σε άκρο. Η διαδικασία περιλάμβανε:

- υποβολή αιτήματος σε φυσική γλώσσα,
- μετατροπή του αιτήματος σε δομημένη προδιαγραφή,
- ενεργοποίηση του μηχανισμού προγραμματισμού,
- συντονισμό μεταξύ των πρακτόρων,
- και παραγωγή τελικού αποτελέσματος.

Σε αρκετές εκτελέσεις εφαρμόστηκε πριν από κάθε δοκιμή καθαρισμός των δεδομένων διαθεσιμότητας, επανεκκίνηση των πρακτόρων συμμετεχόντων και απομόνωση του περιβάλλοντος εκτέλεσης, ώστε να περιοριστεί η επίδραση υπολειμματικής κατάστασης από προηγούμενα πειράματα.

6.4 Μετρικές αξιολόγησης

Για την αποτίμηση της συμπεριφοράς του συστήματος χρησιμοποιήθηκαν τρεις βασικές κατηγορίες μετρικών: ποσοστό επιτυχίας, χρόνος απόκρισης και συμπεριφορά λήψης απόφασης.

6.4.1 Ποσοστό επιτυχίας (Success Rate)

Το ποσοστό επιτυχίας ορίζεται ως το ποσοστό των αιτημάτων προγραμματισμού που ολοκληρώνονται επιτυχώς, δηλαδή οδηγούν σε εύρεση αποδεκτού χρονικού διαστήματος και σε επιτυχή οριστικοποίηση της συνάντησης. Η μετρική αυτή χρησιμοποιείται για να ελεγχθεί αν το σύστημα βρίσκει λύσεις όταν αυτές υπάρχουν και αν αποτυγχάνει μόνο όταν δεν υπάρχει πράγματι εφικτή επιλογή.

6.4.2 Χρόνος απόκρισης (Latency)

Ο συνολικός χρόνος εκτέλεσης διαχωρίζεται σε δύο κύρια στάδια:

- `nl2json_sec`: χρόνος μετατροπής φυσικής γλώσσας σε δομημένη προδιαγραφή μέσω του γλωσσικού μοντέλου,
- `scheduler_sec`: χρόνος εκτέλεσης του κατανεμημένου μηχανισμού προγραμματισμού.

Ο συνολικός χρόνος δίνεται από:

$total_sec = nl2json_sec + scheduler_sec$

Η μετρική αυτή επιτρέπει τον εντοπισμό του βασικού σημείου καθυστέρησης του συνολικού συστήματος.

6.4.3 Συμμετοχικότητα και λογική απόφασης

Παρότι το σύστημα δεν υλοποιεί τυπικό μαθηματικό κριτήριο δικαιοσύνης, αξιολογείται ως προς τη συμμετοχική λογική λήψης αποφάσεων. Ειδικότερα, εξετάζεται αν όλοι οι σχετικοί πράκτορες συμμετέχουν στη διαδικασία αξιολόγησης, αν η απόφαση εξαρτάται από τις αποκρίσεις των επιμέρους πρακτόρων και αν η τελική οριστικοποίηση ακολουθεί συνεπή λογική απαρτία. Η μετρική αυτή δεν αποτιμά τυπικά τη δικαιοσύνη, αλλά ελέγχει αν το σύστημα παραμένει συνεπές με τη συμμετοχική και πολυπρακτορική λογική λήψης αποφάσεων.

6.5 Αξιολόγηση κλιμάκωσης

6.5.1 Περιγραφή

Στόχος του πειράματος είναι η αξιολόγηση της κλιμάκωσης του συστήματος ως προς τον αριθμό των συμμετεχόντων. Εξετάζεται η επίδραση του πλήθους των πρακτόρων στον συνολικό χρόνο εκτέλεσης του συστήματος.

6.5.2 Πειραματική ρύθμιση

- Οι σταθερές συνθήκες του πειράματος ήταν:
- κοινή χρονική ζώνη,
- ίδιος μηχανισμός οριστικοποίησης,

- ίδια πολιτική απαρτίας,
- ίδια λογική προγραμματισμού,
- κενά δεδομένα διαθεσιμότητας,
- χρονικό παράθυρο 10:00–14:00,
- διάρκεια συνάντησης 60 λεπτών.
- Εξετάστηκαν οι εξής περιπτώσεις:
- 2 πράκτορες,
- 5 πράκτορες,
- 10 πράκτορες,
- 20 πράκτορες,
- 30 πράκτορες,
- 40 πράκτορες,
- 50 πράκτορες

6.5.3 Αποτελέσματα

Πίνακας 6.5: Συνολικός χρόνος εκτέλεσης ως προς τον αριθμό των πρακτόρων

Πράκτορες	Συνολικός Χρόνος (sec)
2	54.8
5	53.0
10	56.0
20	64.9
30	55.3
40	76.6
50	87.2

6.5.4 Ανάλυση

Τα αποτελέσματα δείχνουν ότι το σύστημα διατηρεί σχετικά σταθερό χρόνο εκτέλεσης για μικρό και μεσαίο αριθμό πρακτόρων, ενώ για μεγαλύτερα μεγέθη εμφανίζεται σαφέστερη αύξηση του συνολικού χρόνου απόκρισης. Η συμπεριφορά αυτή είναι αναμενόμενη, καθώς όσο αυξάνεται ο αριθμός των συμμετεχόντων αυξάνεται και ο αριθμός των ανταλλασσόμενων μηνυμάτων, το κόστος συλλογής και επεξεργασίας αποκρίσεων και το συνολικό κόστος συντονισμού.

Οι μη μονοτονικές αποκλίσεις που παρατηρούνται σε ορισμένες ενδιάμεσες περιπτώσεις μπορούν να αποδοθούν στη μεταβλητότητα του γλωσσικού υποσυστήματος, στην ασύγχρονη επικοινωνία των πρακτόρων και στη συμπεριφορά του περιβάλλοντος εκτέλεσης σε κοντέινερ.

6.5.5 Συμπέρασμα

Το σύστημα παραμένει λειτουργικό ακόμη και με 50 πράκτορες, γεγονός που δείχνει ότι η αρχιτεκτονική του παρουσιάζει ικανοποιητική κλιμάκωση ως προς τη λειτουργικότητα. Ωστόσο, η αύξηση του αριθμού των συμμετεχόντων επιφέρει αυξημένο κόστος συντονισμού και μεγαλύτερο συνολικό χρόνο εκτέλεσης.

6.6 Ανάλυση απόδοσης και χρόνου απόκρισης

6.6.1 Περιγραφή

Στόχος του πειράματος είναι η ανάλυση του συνολικού χρόνου εκτέλεσης, ώστε να προσδιοριστεί ποιο υποσύστημα συμβάλλει περισσότερο στην καθυστέρηση.

6.6.2 Πειραματική ρύθμιση

Το πείραμα πραγματοποιήθηκε με 5 πράκτορες και κοινό χρονικό παράθυρο 10:00–14:00. Σε κάθε εκτέλεση καταγράφηκαν οι μετρικές `nl2json_sec`, `scheduler_sec` και `total_sec`. Η εκτέλεση επαναλήφθηκε δέκα φορές, με καθαρισμό των δεδομένων διαθεσιμότητας και επανεκκίνηση των πρακτόρων πριν από κάθε δοκιμή.

6.6.3 Αποτελέσματα

Πίνακας 6.6: Χρόνοι εκτέλεσης ανά επανάληψη του πειράματος

Εκτέλεση	Χρόνος μετατροπής σε δομημένο αίτημα (s)	Χρόνος μηχανισμού προγραμματισμού (s)	Συνολικός χρόνος (s)
1	33.144	20.755	53.900
2	32.495	20.633	53.129
3	30.156	20.693	50.849
4	30.452	20.776	51.229
5	32.080	20.393	52.474
6	30.252	20.512	50.765
7	31.275	20.732	52.008
8	32.723	20.689	53.412

Εκτέλεση	Χρόνος μετατροπής σε δομημένο αίτημα (s)	Χρόνος μηχανισμού προγραμματισμού (s)	Συνολικός χρόνος (s)
9	30.444	20.600	51.044
10	31.484	20.671	52.155

Μέσοι όροι:

Πίνακας 6.7.1: Μέσες τιμές χρόνων εκτέλεσης

Μετρική	Μέση Τιμή (s)
Χρόνος μετατροπής αιτήματος	31.45
Χρόνος μηχανισμού προγραμματισμού (s)	20.65
Συνολικός χρόνος (s)	52.10

6.6.4 Ανάλυση

Τα αποτελέσματα δείχνουν ότι το στάδιο μετατροπής φυσικής γλώσσας σε δομημένη προδιαγραφή καταλαμβάνει το μεγαλύτερο μέρος του συνολικού χρόνου εκτέλεσης, περίπου το 60% του συνολικού χρόνου απόκρισης. Αντίθετα, ο χρόνος του μηχανισμού προγραμματισμού παραμένει πιο σταθερός και αντιστοιχεί περίπου στο 40% του συνολικού χρόνου.

Η μεγαλύτερη διακύμανση εντοπίζεται στο στάδιο της γλωσσικής επεξεργασίας, γεγονός που είναι συμβατό με τη μεταβλητότητα του επιπέδου inference του Μεγάλου Γλωσσικού Μοντέλου. Ο κατανεμημένος μηχανισμός προγραμματισμού εμφανίζει πιο σταθερή και προβλέψιμη συμπεριφορά.

6.6.5 Συμπέρασμα

Το βασικό σημείο καθυστέρησης του πλήρους συστήματος δεν είναι ο κατανεμημένος μηχανισμός προγραμματισμού, αλλά το υποσύστημα μετατροπής φυσικής γλώσσας σε δομημένο αίτημα. Κατά συνέπεια, πιθανές βελτιστοποιήσεις του συνολικού χρόνου απόκρισης θα πρέπει να στοχεύσουν κυρίως στη μείωση του κόστους του γλωσσικού επιπέδου.

6.7 Επίδραση του βήματος αναζήτησης

6.7.1 Περιγραφή

Το πείραμα αυτό αξιολογεί την επίδραση της παραμέτρου candidate step στον αριθμό των υποψήφιων χρονικών slots και στον συνολικό χρόνο εκτέλεσης του συστήματος.

6.7.2 Πειραματική ρύθμιση

Το πείραμα πραγματοποιήθηκε με:

- 5 πράκτορες,

- χρονικό παράθυρο 10:00–14:00,
- διάρκεια 60 λεπτών,
- κενά δεδομένα διαθεσιμότητας,
- βήμα αναζήτησης ίσο με 15, 30 και 60 λεπτά.

6.7.3 Αποτελέσματα

Πίνακας 6.7: Επίδραση του βήματος αναζήτησης στον αριθμό των υποψήφιων διαστημάτων και στον χρόνο εκτέλεσης

Βήμα (min)	Υποψήφια διαστήματα	Συνολικός χρόνος (sec)	Χρόνος μηχανισμού προγραμματισμού (sec)	Αποτέλεσμα
15	13	48.346	20.609	success
30	7	55.103	20.486	success
60	4	54.057	20.474	success

6.7.4 Ανάλυση

Η αύξηση του βήματος αναζήτησης μειώνει σημαντικά τον αριθμό των υποψήφιων χρονικών διαστημάτων. Συγκεκριμένα, με βήμα 15 λεπτών προκύπτουν 13 υποψήφιες επιλογές, με βήμα 30 λεπτών 7 επιλογές και με βήμα 60 λεπτών 4 επιλογές.

Ωστόσο, η μείωση αυτή δεν οδήγησε σε αντίστοιχη μείωση του συνολικού χρόνου εκτέλεσης, καθώς και στις τρεις περιπτώσεις το σύστημα εντόπισε λύση στο πρώτο υποψήφιο χρονικό διάστημα. Επομένως, σε απλά σενάρια χαμηλής σύγκρουσης το μέγεθος του χώρου αναζήτησης δεν φαίνεται να επηρεάζει σημαντικά τον χρόνο απόκρισης.

6.7.5 Συμπέρασμα

Το βήμα αναζήτησης επηρεάζει άμεσα το μέγεθος του χώρου αναζήτησης, αλλά σε περιπτώσεις χαμηλής σύγκρουσης η επίδρασή του στον χρόνο εκτέλεσης είναι περιορισμένη. Σε πιο απαιτητικά σενάρια αναμένεται ισχυρότερη επίδραση.

6.8 Περίπτωση αποτυχίας: απουσία διαθέσιμου χρονικού διαστήματος

6.8.1 Περιγραφή

Το πείραμα αυτό αξιολογεί τη συμπεριφορά του συστήματος σε περιπτώσεις όπου δεν υπάρχει κανένα διαθέσιμο χρονικό διάστημα για όλους τους συμμετέχοντες. Στόχος είναι να επιβεβαιωθεί ότι το σύστημα αποτυγχάνει με ελεγχόμενο και ορθό τρόπο.

6.8.2 Πειραματική ρύθμιση

Οι πέντε συμμετέχοντες ορίστηκαν ως πλήρως απασχολημένοι εντός του χρονικού παραθύρου 10:00–14:00, με διάρκεια συνάντησης 60 λεπτών.

6.8.3 Αποτελέσματα

Κατά την εκτέλεση του μηχανισμού προγραμματισμού, όλα τα υποψήφια χρονικά διαστήματα απορρίφθηκαν λόγω συγκρούσεων διαθεσιμότητας και το σύστημα κατέληξε σε ένδειξη NO_SLOT_FOUND

6.8.4 Ανάλυση

Το αποτέλεσμα δείχνει ότι το σύστημα εξετάζει συστηματικά τα διαθέσιμα υποψήφια χρονικά διαστήματα, λαμβάνει υπόψη τις αρνητικές αποκρίσεις των επιμέρους πρακτόρων, δεν παράγει μη έγκυρη λύση όταν δεν υπάρχει διαθέσιμη χρονική τομή και τερματίζει με σαφή και προβλέψιμο τρόπο.

Η συμπεριφορά αυτή είναι σημαντική, διότι επιβεβαιώνει ότι ο μηχανισμός προγραμματισμού διατηρεί συνέπεια και δεν παραβιάζει τους περιορισμούς διαθεσιμότητας προκειμένου να επιστρέψει αποτέλεσμα.

6.8.5 Συμπέρασμα

Το πείραμα επιβεβαιώνει ότι το σύστημα διαχειρίζεται ορθά περιπτώσεις πλήρους σύγκρουσης και αποτυγχάνει με ελεγχόμενο τρόπο όταν δεν υπάρχει εφικτή λύση.

6.9 Συνολικά συμπεράσματα πειραματικής αξιολόγησης

Η συνολική αξιολόγηση δείχνει ότι η προτεινόμενη αρχιτεκτονική επιτυγχάνει υψηλό επίπεδο λειτουργικότητας και ορθότητας στη διαδικασία προγραμματισμού συναντήσεων. Το σύστημα βρίσκει λύσεις όταν αυτές υπάρχουν, σέβεται χρονικούς και ημερολογιακούς περιορισμούς, αποτυγχάνει με ελεγχόμενο τρόπο όταν δεν υπάρχει διαθέσιμη λύση και παραμένει λειτουργικό ακόμη και με μεγάλο αριθμό συμμετεχόντων.

Από πλευράς απόδοσης, η ανάλυση δείχνει ότι ο ίδιος ο καταναεμημένος μηχανισμός προγραμματισμού παρουσιάζει σχετικά σταθερή συμπεριφορά, ενώ το σημαντικότερο μέρος του συνολικού κόστους εντοπίζεται στο στάδιο σημασιολογικής επεξεργασίας του αιτήματος. Το εύρημα αυτό είναι σημαντικό, διότι δείχνει ότι το βασικό σημείο βελτίωσης δεν βρίσκεται απαραίτητα στον μηχανισμό συντονισμού των πρακτόρων, αλλά στο φυσικογλωσσικό επίπεδο εισόδου.

Ως προς τη λογική λήψης αποφάσεων, τα πειράματα επιβεβαιώνουν ότι το σύστημα λειτουργεί με συμμετοχικό τρόπο, αφού η τελική απόφαση εξαρτάται από τις τοπικές αξιολογήσεις και τις αποκρίσεις των επιμέρους πρακτόρων, αφού η τελική απόφαση εξαρτάται από τις τοπικές αξιολογήσεις και τις αποκρίσεις των επιμέρους πρακτόρων. Παρότι δεν υλοποιείται τυπικό μαθηματικό κριτήριο δικαιοσύνης, η διαδικασία απόφασης βασίζεται σε συλλογική συμμετοχή, η διαδικασία απόφασης βασίζεται σε συλλογική συμμετοχή, λογική απαρτίας και τοπική αξιολόγηση από τους επιμέρους πράκτορες.

6.10 Συμπέρασμα κεφαλαίου

Η πειραματική αξιολόγηση δείχνει ότι το προτεινόμενο σύστημα αποτελεί λειτουργικό και μετρήσιμο πολυπρακτορικό πρωτότυπο για αυτόματο προγραμματισμό συναντήσεων. Η αρχιτεκτονική του υποστηρίζει σωστή διαχείριση περιορισμών, πολυπρακτορικό συντονισμό, δυνατότητα λειτουργίας με δεκάδες πράκτορες και ελεγχόμενη αποτυχία όταν δεν υπάρχει διαθέσιμη λύση.

Παράλληλα, τα αποτελέσματα δείχνουν ότι το κυριότερο σημείο βελτίωσης αφορά το στάδιο μετατροπής φυσικής γλώσσας σε δομημένο αίτημα προγραμματισμού. Επομένως, η περαιτέρω

Κεφάλαιο 6

βελτίωση του συστήματος μπορεί να στοχεύσει τόσο στη μείωση του κόστους του γλωσσικού επιπέδου όσο και σε πρόσθετες τεχνικές βελτιστοποίησης του επιπέδου συντονισμού.

Κεφάλαιο 7ο: Συμπεράσματα και μελλοντική εργασία

7.1 Εισαγωγή

Το κεφάλαιο αυτό συνοψίζει τα βασικά συμπεράσματα της εργασίας και παρουσιάζει ενδεικτικές κατευθύνσεις για μελλοντική επέκταση του συστήματος. Η παρούσα διπλωματική είχε ως αντικείμενο τη σχεδίαση, την υλοποίηση και την πειραματική αξιολόγηση ενός συστήματος αυτόματου προγραμματισμού συναντήσεων, το οποίο συνδυάζει πολυπρακτορική αρχιτεκτονική με υποστήριξη Μεγάλων Γλωσσικών Μοντέλων. Η κεντρική ιδέα της προσέγγισης είναι ότι ο προγραμματισμός συναντήσεων μπορεί να υλοποιηθεί μέσω πολυπρακτορικού συντονισμού μεταξύ αυτόνομων πρακτόρων, ενώ τα Μεγάλα Γλωσσικά Μοντέλα λειτουργούν υποστηρικτικά στη σημασιολογική κατανόηση των αιτημάτων του χρήστη.

7.2 Συμπεράσματα

Η υλοποίηση που αναπτύχθηκε δείχνει ότι ο αυτόματος προγραμματισμός συναντήσεων μπορεί να προσεγγιστεί αποτελεσματικά ως πρόβλημα πολυπρακτορικού συντονισμού. Το σύστημα που υλοποιήθηκε βασίζεται σε πράκτορες SPADE/XMPP, σε επικοινωνία μέσω ανταλλαγής μηνυμάτων και σε διακριτούς ρόλους, όπως πράκτορας συντονισμού, πράκτορες-μέλη και πράκτορες-χώρων. Με τον τρόπο αυτό, η διαδικασία προγραμματισμού δεν αντιμετωπίζεται ως απλή αναζήτηση κοινής διαθεσιμότητας σε ένα κεντρικό σύστημα, αλλά ως διαδικασία συλλογικής αξιολόγησης και οριστικοποίησης μεταξύ πολλών αυτόνομων οντοτήτων.

Ένα βασικό συμπέρασμα της εργασίας είναι ότι η συνδυαστική χρήση πολυπρακτορικού προγραμματισμού συναντήσεων και φυσικής γλώσσας είναι εφικτή και λειτουργική, αρκεί να υπάρχει σαφής διαχωρισμός ρόλων ανάμεσα στο σημασιολογικού επιπέδου και στον μηχανισμό λήψης απόφασης [3], [10], [12]. Στην προτεινόμενη αρχιτεκτονική, το LLM χρησιμοποιείται για τη μετατροπή αιτημάτων φυσικής γλώσσας σε δομημένα αιτήματα προγραμματισμού, ενώ ο πραγματικός συντονισμός και η τελική επιλογή του χρονικού διαστήματος πραγματοποιούνται από τους πράκτορες. Η διάκριση αυτή αποδείχθηκε χρήσιμη, επειδή επιτρέπει αφενός φυσικότερη αλληλεπίδραση με τον χρήστη και αφετέρου διατήρηση ελέγχιμου, επαναλήψιμου και πολυπρακτορικού μηχανισμού προγραμματισμού μηχανισμού.

Η πειραματική αξιολόγηση έδειξε ότι το σύστημα λειτουργεί σωστά τόσο σε απλά όσο και σε πιο απαιτητικά σενάρια. Ο προγραμματισμός ολοκληρώνεται επιτυχώς όταν υπάρχει διαθέσιμο χρονικό διάστημα, ενώ σε περιπτώσεις πλήρους σύγκρουσης το σύστημα αποτυγχάνει με ελεγχόμενο τρόπο, χωρίς να παράγει μη έγκυρα αποτελέσματα. Επιπλέον, η αρχιτεκτονική παρέμεινε λειτουργική ακόμη και όταν αυξήθηκε σημαντικά ο αριθμός των συμμετεχόντων, γεγονός που δείχνει ότι η προσέγγιση μπορεί να υποστηρίξει πειραματικά σενάρια με δεκάδες πράκτορες.

Η ανάλυση της απόδοσης έδειξε επίσης ότι το κύριο σημείο συμφόρησης του συνολικού συστήματος δεν βρίσκεται στον ίδιο τον κατανεμημένο μηχανισμό προγραμματισμού, αλλά στο στάδιο μετατροπής φυσικής γλώσσας σε δομημένο αίτημα. Το εύρημα αυτό είναι σημαντικό, επειδή δείχνει ότι ο πολυπρακτορικός πυρήνας του συστήματος έχει σχετικά σταθερή και προβλέψιμη συμπεριφορά, ενώ η μεγαλύτερη χρονική επιβάρυνση σχετίζεται με το επίπεδο γλωσσικής επεξεργασίας.

Συνολικά, η εργασία δεν επιδιώκει να εισαγάγει ένα νέο θεωρητικό πρόβλημα, αλλά να δείξει πειραματικά πώς μια σύγχρονη υβριδική προσέγγιση μπορεί να συνδυάσει:

- πολυπρακτορικό συντονισμό,
- διαπραγμάτευση μέσω ανταλλαγής μηνυμάτων,
- είσοδο σε φυσική γλώσσα,
- και παραγωγή διαλειτουργικών αποτελεσμάτων προγραμματισμού.

Με αυτή την έννοια, η βασική συμβολή της εργασίας εντοπίζεται στον σχεδιασμό και στην υλοποίηση ενός εκτελέσιμου, πολυπρακτορικού και αξιολογήσιμου πρωτοτύπου, το οποίο συνδυάζει σύγχρονα εργαλεία ευφώνων πρακτόρων με υποστήριξη Μεγάλου Γλωσσικού Μοντέλου.

7.3 Περιορισμοί της προσέγγισης

Παρά τα θετικά αποτελέσματα, η παρούσα προσέγγιση παρουσιάζει και ορισμένους περιορισμούς. Πρώτον, η ενσωμάτωση του Μεγάλου Γλωσσικού Μοντέλου περιορίζεται κυρίως στη σημασιολογική κατανόηση της εισόδου και στην παραγωγή δομημένων αιτημάτων προγραμματισμού. Το Μεγάλο Γλωσσικό Μοντέλο δεν συμμετέχει στη δυναμική λήψη απόφασης κατά τη διάρκεια της διαπραγμάτευσης μεταξύ των πρακτόρων. Αυτό είναι αρχιτεκτονικά σκόπιμο, αλλά σημαίνει ότι η γνωστική προσαρμοστικότητα του συστήματος παραμένει περιορισμένη σε σχέση με πιο εξελιγμένα πολυπρακτορικά συστήματα στα οποία τα Μεγάλα Γλωσσικά Μοντέλα συμμετέχουν ενεργά στη διαδικασία λήψης αποφάσεων.

Δεύτερον, η πληροφορία διαθεσιμότητας βασίζεται σε τοπική αποθήκευση free/busy δεδομένων σε αρχεία JSON και όχι σε άμεση διασύνδεση με πραγματικά συστήματα ημερολογίου, όπως Google Calendar, Outlook ή CalDAV servers. Επομένως, το σύστημα λειτουργεί σήμερα ως πειραματικό πρωτότυπο με διαλειτουργικά αποτελέσματα, αλλά όχι ως πλήρως ενσωματωμένη παραγωγική πλατφόρμα scheduling.

Τρίτον, η λογική λήψης απόφασης βασίζεται σε συνδυασμό ποσοτικής αξιολόγησης, ορίων απαρτίας και υποχρεωτικών συμμετεχόντων, αλλά δεν υλοποιεί αυστηρά μαθηματικά κριτήρια δικαιοσύνης ή πιο προχωρημένους μηχανισμούς βελτιστοποίησης. Παρότι η συμμετοχή των πρακτόρων είναι συλλογική και βασίζεται σε τοπικές αξιολογήσεις και μηχανισμούς απαρτίας, η έννοια της δικαιοσύνης παραμένει περισσότερο λειτουργική παρά τυπικά διατυπωμένη.

Τέταρτον, η αξιολόγηση πραγματοποιήθηκε σε ελεγχόμενο πειραματικό περιβάλλον, με συγκεκριμένα σενάρια, συμμετέχοντες και συνθήκες. Παρότι τα αποτελέσματα είναι ενθαρρυντικά, απαιτείται περαιτέρω μελέτη σε πιο σύνθετα ή πιο ρεαλιστικά περιβάλλοντα, όπου θα συνυπάρχουν πιο πλούσιες προτιμήσεις, διαφορετικοί τύποι πόρων και πιο έντονες συγκρούσεις διαθεσιμότητας.

7.4 Μελλοντική εργασία

Η παρούσα εργασία μπορεί να επεκταθεί σε αρκετές κατευθύνσεις. Μία πρώτη και ιδιαίτερα σημαντική κατεύθυνση είναι η σύνδεση του συστήματος με πραγματικά calendaring οικοσυστήματα. Η υποστήριξη διαλειτουργικών artifacts σε μορφή .ics αποτελεί ένα πρώτο βήμα, αλλά μία πληρέστερη ενσωμάτωση με Google Calendar, Outlook ή υπηρεσίες βασισμένες στο πρωτόκολλο CalDAV θα επέτρεπε στο σύστημα να χρησιμοποιεί πραγματικά δεδομένα και να προσεγγίσει περισσότερο σενάρια παραγωγικής χρήσης.

Μία δεύτερη κατεύθυνση αφορά τον εμπλουτισμό του μηχανισμού λήψης απόφασης. Το σημερινό σύστημα χρησιμοποιεί αξιολόγηση βασισμένη σε χρησιμότητα και οριστικοποίηση με λογική απαρτίας, αλλά μελλοντικά θα μπορούσαν να ενσωματωθούν πιο σύνθετες στρατηγικές, όπως βελτιστοποίηση περιορισμών, μοντελοποίηση μαθημένων προτιμήσεων ή υβριδική χρήση εξωτερικών επιλυτών για δυσκολότερα προβλήματα προγραμματισμού.

Μία τρίτη κατεύθυνση αφορά την ενίσχυση του γνωστικού επιπέδου. Παρότι το Μεγάλο Γλωσσικό Μοντέλο χρησιμοποιείται σήμερα για σημασιολογική ανάλυση, θα μπορούσε μελλοντικά να υποστηρίξει πλουσιότερη ερμηνευσιμότητα, καλύτερη διαχείριση ασαφών ή σύνθετων αιτημάτων, καθώς και δυναμική αναδιατύπωση ή διαλογική αλληλεπίδραση με τον χρήστη όταν οι διαθέσιμες πληροφορίες δεν επαρκούν για άμεση εκτέλεση.

Επιπλέον, ενδιαφέρουσα μελλοντική κατεύθυνση είναι η διερεύνηση πιο εξελιγμένων στρατηγικών διαπραγμάτευσης μεταξύ πρακτόρων. Η παρούσα υλοποίηση ακολουθεί σαφή και ελεγχόμενη λογική προετοιμασίας και οριστικοποίησης (prepare/commit), αλλά θα μπορούσαν να μελετηθούν και προσαρμοστικές πολιτικές διαπραγμάτευσης, πιο σύνθετα μοντέλα χρησιμότητας ή μηχανισμοί που λαμβάνουν υπόψη ιστορικό αλληλεπιδράσεων.

Τέλος, ιδιαίτερη σημασία έχει και η περαιτέρω πειραματική αξιολόγηση. Μελλοντικά θα μπορούσαν να εξεταστούν σενάρια με μεγαλύτερο αριθμό πρακτόρων, διαφορετικές οργανωσιακές δομές, πιο πυκνά μοτίβα συγκρούσεων, καθώς και συγκριτική αξιολόγηση με εναλλακτικές προσεγγίσεις προγραμματισμού. Μία τέτοια επέκταση θα επέτρεπε ακριβέστερη αποτίμηση των ορίων και των δυνατοτήτων της προτεινόμενης αρχιτεκτονικής. Η χρήση τεχνολογιών κοντέινερ και η εκτέλεση μέσω Docker Compose επιτρέπουν την αναπαραγωγή των πειραμάτων και τη μελλοντική επέκταση του πρωτοτύπου σε μεγαλύτερα ή περισσότερα καταναεμημένα περιβάλλοντα.

Η δομή του αποθετηρίου και βασικές οδηγίες αναπαραγωγής της εκτέλεσης παρουσιάζονται αναλυτικά στα Παραρτήματα Γ και Δ.

7.5 Τελικό συμπέρασμα

Συνοψίζοντας, η εργασία δείχνει ότι η σύζευξη πολυπρακτορικών συστημάτων και υποστήριξης Μεγάλου Γλωσσικού Μοντέλου για σημασιολογική κατανόηση αποτελεί μια ενδιαφέρουσα και λειτουργική προσέγγιση για τον αυτόματο προγραμματισμό συναντήσεων. Παρότι το σύστημα παραμένει σε επίπεδο πειραματικού πρωτοτύπου, τα αποτελέσματα της υλοποίησης και της αξιολόγησης δείχνουν ότι μπορεί να αποτελέσει βάση για περαιτέρω ερευνητική και τεχνολογική ανάπτυξη.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] M. Wooldridge and N. R. Jennings, “Intelligent Agents: Theory and Practice,” *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.
- [2] R. G. Smith, “The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver,” *IEEE Transactions on Computers*, vol. C-29, no. 12, pp. 1104–1113, Dec. 1980.
- [3] S. Chen, Y. Liu, W. Han, W. Zhang, and T. Liu, “A Survey on LLM-based Multi-Agent System: Recent Advances and New Frontiers in Application,” *arXiv preprint arXiv:2412.17481*, 2024.
- [4] B. Desruisseaux, “Internet Calendaring and Scheduling Core Object Specification (iCalendar),” RFC 5545, Sep. 2009.
- [5] C. Daboo et al., “Scheduling Extensions to CalDAV,” RFC 6638, Jun. 2012.
- [6] I. O. Demirel and N. Erdogan, “Meeting Scheduling with Multi Agent Systems: Design and Implementation,” in *Proc. 6th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems*, 2007, pp. 335–340.
- [7] A. Zunino and M. Campo, “Chronos: A Multi-Agent System for Distributed Automatic Meeting Scheduling,” *Expert Systems with Applications*, vol. 36, no. 3, pp. 7011–7018, 2009.
- [8] B. M. Renting, H. H. Hoos, and C. M. Jonker, “Multi-Agent Meeting Scheduling: A Negotiation Perspective,” in *Proc. ALA Workshop at AAMAS*, 2024.
- [9] O. Wijerathne, A. Nimasha, D. Fernando, N. de Silva, and S. Perera, “ScheduleMe: Multi-Agent Calendar Assistant,” *arXiv preprint arXiv:2509.25693*, 2025.
- [10] J. Palanca, J. A. Rincon, C. Carrascosa, V. J. Julian, and A. Terrasa, “Flexible Agent Architecture: Mixing Reactive and Deliberative Behaviors in SPADE,” *Electronics*, vol. 12, no. 3, art. no. 659, 2023.
- [11] J. Gray and A. Reuter, *Transaction Processing: Concepts and Techniques*, Morgan Kaufmann, 1993.
- [12] OpenAI, “GPT-4 Technical Report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [13] A. Iliadis, “LLM Multi-Agent Scheduler,” GitHub repository, 2026.
[Online]. Available: <https://github.com/Alxiliadi/llm-multi-agent-scheduler>.

ΠΑΡΑΡΤΗΜΑ Α— Prompts προς το μεγάλο γλωσσικό μοντέλο και παραδείγματα κλήσεων API

Το παρόν παράρτημα παρουσιάζει ενδεικτικά αιτήματα φυσικής γλώσσας και αντίστοιχες κλήσεις προς το REST API του συστήματος. Τα παραδείγματα χρησιμοποιήθηκαν κατά τη διαδικασία δοκιμών και πειραματικής αξιολόγησης του προτεινόμενου πολυπρακτορικού συστήματος προγραμματισμού συναντήσεων.

7.6 A.1 System prompt προς το μεγάλο γλωσσικό μοντέλο

Η σημασιολογική επεξεργασία των αιτημάτων φυσικής γλώσσας βασίστηκε σε αυστηρά καθορισμένο system prompt. Το prompt αυτό καθορίζει το σχήμα JSON που πρέπει να επιστρέφει το μοντέλο, τους κανόνες μορφοποίησης, την αντιστοίχιση συμμετεχόντων σε XMPP identifiers και την ερμηνεία ελληνικών χρονικών εκφράσεων.

SYSTEM = ""

You are a helpful assistant that converts natural language meeting requests into a strict JSON spec.

OUTPUT JSON – NO PROSE, ONLY JSON:

```
{
  "peers": ["jid1", "jid2", ...],
  "window_start": "YYYY-MM-DDThh:mm:ss+TZ",
  "window_end": "YYYY-MM-DDThh:mm:ss+TZ",
  "quorum_threshold": 0.7,
  "duration_min": 60,
  "required_peers": []
}
```

Rules:

1. Output ONLY valid JSON.
2. If timezone missing → assume +02:00.
3. Convert all participant/resource names to XMPP JIDs (e.g. "member1" → "member1@xmpp", "room1" → "room1@xmpp").
4. Duration is in minutes.
5. Required/VIP users → required_peers.

6. Do not invent users.
 7. Always use 24-hour time (HH:MM).
 8. If the request includes a Greek time-of-day word, interpret hours as:
 - "πρωί": AM (no +12)
 - "μεσημέρι": if hour is 1..11 then add 12 (e.g., 3 -> 15:00)
 - "απόγευμα": if hour is 1..11 then add 12 (e.g., 6 -> 18:00)
 - "βράδυ"/"βραδάκι": if hour is 1..11 then add 12 (e.g., 9 -> 21:00)
- """

7.7 A.2 Πρότυπο εισόδου προς το μοντέλο

Το αίτημα του χρήστη ενσωματώνεται σε πρότυπο εισόδου, ώστε κάθε φυσική γλωσσική διατύπωση να μετατρέπεται με σταθερό τρόπο σε δομημένη προδιαγραφή προγραμματισμού.

USER_TEMPLATE = """Convert the following request to the JSON schema:

REQUEST:

```
{request}
"""
```

Η χρήση σταθερού system prompt και προτύπου εισόδου συνέβαλε στη μείωση μη έγκυρων αποκρίσεων και στη βελτίωση της συνέπειας των παραγόμενων JSON αποτελεσμάτων.

7.8 A.3 Βασικό αίτημα προγραμματισμού

```
curl -s http://localhost:8001/nl2schedule \
-H "Content-Type: application/json" \
-d '{"prompt": "Κάνε συνάντηση με member1 και member2 αύριο 10:00-14:00 διάρκεια 60"}'
```

Το αίτημα αυτό ελέγχει τη βασική λειτουργία του συστήματος με δύο συμμετέχοντες και χρονικό παράθυρο τεσσάρων ωρών.

Ενδεικτικό αποτέλεσμα:

```
{
  "status": "success",
  "slot": "2026-06-06T10:00:00+02:00",
  "duration_min": 60,
  "quorum": 1
}
```

7.9 A.4 Αίτημα με πολλούς συμμετέχοντες

```
curl -s http://localhost:8001/nl2schedule \  
-H "Content-Type: application/json" \  
-d '{"prompt":"Κάνε συνάντηση με member1 member2 member3 member4 member5  
στις 06/06/2026 10:00-14:00 διάρκεια 60"}'
```

Το αίτημα αυτό χρησιμοποιείται για αξιολόγηση της συμπεριφοράς του συστήματος με περισσότερους πράκτορες.

Ενδεικτικό αποτέλεσμα:

```
{  
"status": "success",  
"slot": "2026-06-06T10:00:00+02:00",  
"duration_min": 60,  
"quorum": 3  
}
```

7.10 A.5 Αίτημα κλιμάκωσης με 50 πράκτορες

```
curl -s http://localhost:8001/nl2schedule \  
-H "Content-Type: application/json" \  
-d '{"prompt":"Peers: member1, member2, member3, member4, member5, member6,  
member7, member8, member9, member10, member11, member12, member13, member14,  
member15, member16, member17, member18, member19, member20, member21,  
member22, member23, member24, member25, member26, member27, member28,  
member29, member30, member31, member32, member33, member34, member35,  
member36, member37, member38, member39, member40, member41, member42,  
member43, member44, member45, member46, member47, member48, member49,  
member50 αύριο στις 20:00 διάρκεια 60"}'
```

Το αίτημα αυτό χρησιμοποιήθηκε για την αξιολόγηση της κλιμάκωσης του συστήματος με μεγάλο αριθμό συμμετεχόντων.

7.11 A.6 Αίτημα περίπτωσης αποτυχίας

```
curl -s http://localhost:8001/nl2schedule \  
-H "Content-Type: application/json" \  
-d '{"prompt":"Κάνε συνάντηση με member1 member2 member3 member4 member5  
στις 04/04/2026 10:00-14:00 διάρκεια 60"}'
```

Το αίτημα αυτό χρησιμοποιείται σε σενάριο όπου οι συμμετέχοντες έχουν πλήρη σύγκρουση διαθεσιμότητας και το σύστημα αναμένεται να επιστρέψει αποτυχία τύπου NO_SLOT_FOUND.

Ενδεικτικό αποτέλεσμα:

```
{
  "status": "error",
  "result": {
    "status": "error",
    "reason": "NO_SLOT_FOUND"
  }
}
```

7.12 A.7 Μη έγκυρο χρονικό αίτημα

```
curl -s http://localhost:8001/nl2schedule \
-H "Content-Type: application/json" \
-d '{"prompt": "Κάνε συνάντηση με member1 member2 member3 member4 member5
στις 06/04/2026 10:00-14:00 διάρκεια 60"}'
```

Ενδεικτικό αποτέλεσμα:

```
{
  "detail": "LLM/parse error: Requested time window is in the past."
}
```

Το σύστημα απορρίπτει αιτήματα που αντιστοιχούν σε χρονικά διαστήματα του παρελθόντος πριν από την εκκίνηση του μηχανισμού προγραμματισμού. Με τον τρόπο αυτό αποφεύγεται η εκτέλεση μη έγκυρων αιτημάτων προγραμματισμού.

ΠΑΡΑΡΤΗΜΑ Β — Structured JSON και αρχεία εκτέλεσης

Το παρόν παράρτημα παρουσιάζει τα βασικά αρχεία που παράγονται κατά την εκτέλεση του συστήματος. Τα αρχεία αυτά χρησιμοποιούνται για την ενορχήστρωση της διαδικασίας προγραμματισμού, την ανταλλαγή παραμέτρων μεταξύ των επιμέρους υποσυστημάτων, την καταγραφή της εκτέλεσης και την πειραματική αξιολόγηση του προτεινόμενου πρωτοτύπου.

7.13 B.1 Δομημένο scheduling job (scheduler.json)

Το αρχείο scheduler.json περιλαμβάνει τη δομημένη αναπαράσταση του αιτήματος προγραμματισμού μετά τη σημασιολογική επεξεργασία της φυσικής γλώσσας.

```
{
  "peers": [
    "member1@xmpp",
    "member2@xmpp",
    "member3@xmpp",
    "member4@xmpp",
    "member5@xmpp"
  ],
  "window_start": "2026-06-06T10:00:00+02:00",
  "window_end": "2026-06-06T14:00:00+02:00",
  "duration_min": 60,
  "quorum_threshold": 0.7,
  "required_peers": [],
  "fixed_start": false
}
```

7.14 B.2 Αρχείο παραμέτρων εκτέλεσης (scheduler.env)

Το αρχείο αυτό δημιουργείται δυναμικά για κάθε εκτέλεση και χρησιμοποιείται για τη μεταφορά παραμέτρων προς τον πράκτορα συντονισμού.

Το αρχείο scheduler.env περιλαμβάνει μεταβλητές περιβάλλοντος και παραμέτρους που χρησιμοποιούνται κατά την εκτέλεση του μηχανισμού προγραμματισμού

```
ROLE=scheduler
JID=scheduler@xmpp
PASSWORD=secret

PEERS=member1@xmpp,member2@xmpp,member3@xmpp,member4@xmpp,member5@xmpp

WINDOW_START=2026-06-06T10:00:00+02:00
WINDOW_END=2026-06-06T14:00:00+02:00
```

DURATION_MIN=60

QUORUM_THRESHOLD=0.7

RUN_ID=1cc725ab46ab4c57ba6b77c490d1cd22

JOB_ID=job-20260601T201355Z-1cc725ab46ab4c57ba6b77c490d1cd22

7.15 B.3 Αρχείο αποτελέσματος (result.json)

Μετά την επιτυχή ολοκλήρωση της διαδικασίας προγραμματισμού παράγεται αρχείο αποτελέσματος, το οποίο περιλαμβάνει πληροφορίες σχετικά με το επιλεγμένο χρονικό διάστημα, τους συμμετέχοντες και τις επιβεβαιώσεις που συγκεντρώθηκαν.

```
{  
  "status": "success",  
  "slot": "2026-06-06T10:00:00+02:00",  
  "duration_min": 60,  
  "peers": [ "member1@xmpp", "member2@xmpp", "member3@xmpp", "member4@xmpp",  
    "member5@xmpp" ],  
  "acks": [ "member1@xmpp", "member2@xmpp", "member3@xmpp", "member4@xmpp",  
    "member5@xmpp" ],  
  "quorum": 3,  
  "required_peers": [],  
  "ics_path": "/app/out/meeting-fd86f6dd-c8cb-4c6d-9f62-a302da758a24.ics"  
}
```

7.16 B.4 Αρχείο κατάστασης εκτέλεσης (status.json)

Το αρχείο status.json χρησιμοποιείται για συνοπτική αναπαράσταση της κατάστασης της εκτέλεσης.

```
{  
  "status": "success",  
  "job_id":  
    "job-20260601T201355Z-1cc725ab46ab4c57ba6b77c490d1cd22",  
  "run_id": "1cc725ab46ab4c57ba6b77c490d1cd22" }
```

7.17 B.5 Αρχείο καταγραφής εκτέλεσης (log.txt)

Το αρχείο log.txt περιλαμβάνει αναλυτική καταγραφή της εκτέλεσης του πράκτορα συντονισμού και της επικοινωνίας μεταξύ των επιμέρους πρακτόρων.

Ενδεικτικά καταγράφονται:

- ανταλλαγή REQUEST_SUGGESTIONS,
- συλλογή SUGGESTIONS,
- PREPARE και COMMIT φάσεις,

- VOTE και ACK μηνύματα,
- τελική οριστικοποίηση της συνάντησης.

Παράδειγμα:

```
[scheduler@xmpp] → PREPARE attempt slot=2026-06-06T10:00:00+02:00
[scheduler@xmpp] ← VOTE from member1@xmpp: YES
[scheduler@xmpp] ← VOTE from member2@xmpp: YES
[scheduler@xmpp] → COMMIT slot=2026-06-06T10:00:00+02:00
[scheduler@xmpp] ✓ Meeting finalized
```

7.18 B.6 Δομή καταλόγου εκτέλεσης εργασίας

Κάθε κατάλογος ταυτοποιείται από μοναδικό αναγνωριστικό εργασίας (job_id), επιτρέποντας την ανεξάρτητη παρακολούθηση πολλαπλών εκτελέσεων.

job-20260601T201355Z-1cc725ab46ab4c57ba6b77c490d1cd22/

```
├── log.txt
├── result.json
├── scheduler.env
├── scheduler.json
└── status.json
```

Η δομή αυτή επιτρέπει την αναπαραγωγιμότητα των πειραμάτων, την αποσφαλμάτωση και τη μεταγενέστερη ανάλυση της συμπεριφοράς του συστήματος.

ΠΑΡΑΡΤΗΜΑ Γ — Οδηγίες εγκατάστασης και εκτέλεσης

7.19 Γ.1 Προαπαιτούμενα

Για την εκτέλεση του συστήματος απαιτούνται τα εξής εργαλεία στο περιβάλλον του χρήστη:

- Docker
- Docker Compose
- Ollama
- Git

Οι υπόλοιπες εξαρτήσεις (FastAPI, SPADE, XMPP server, agents κ.λπ.) εγκαθίστανται και εκτελούνται μέσω των Docker containers.

7.20 Γ.2 λήψη αποθετηρίου

Ο πηγαίος κώδικας του συστήματος μπορεί να ληφθεί από το αποθετήριο GitHub με τις ακόλουθες εντολές:

```
git clone https://github.com/Alxiliadi/llm-multi-agent-scheduler
cd llm-multi-agent-scheduler
```

7.21 Γ.3 Προετοιμασία γλωσσικού μοντέλου

Πριν από την εκτέλεση του συστήματος πρέπει να είναι διαθέσιμο το γλωσσικό μοντέλο που χρησιμοποιείται από το Ollama. Στην παρούσα εργασία χρησιμοποιήθηκε το μοντέλο qwen2.5:3b.

```
ollama pull qwen2.5:3b
```

7.22 Γ.4 Εκκίνηση συστήματος

Η εκκίνηση των υπηρεσιών πραγματοποιείται μέσω Docker Compose.

```
docker compose up -d
```

Με την εντολή αυτή εκκινούνται οι βασικές υπηρεσίες του συστήματος, όπως το REST API, ο XMPP server, οι πράκτορες και οι υποστηρικτικές υπηρεσίες.

7.23 Γ.5 Έλεγχος λειτουργίας API

Μετά την εκκίνηση μπορεί να ελεγχθεί η λειτουργία του API με την ακόλουθη εντολή:

```
curl http://localhost:8001/health
```

Ενδεικτικό αποτέλεσμα:

```
{
  "status": "ok"
}
```

7.24 Γ.6 Εκτέλεση ενδεικτικού αιτήματος

Το παρακάτω παράδειγμα παρουσιάζει την εκτέλεση ενός αιτήματος προγραμματισμού συνάντησης μέσω φυσικής γλώσσας.

```
curl -s http://localhost:8001/nl2schedule \  
-H "Content-Type: application/json" \  
-d '{"prompt": "Κάνε συνάντηση με member1 member2 αύριο στις 18:00 διάρκεια 60"}'
```

7.25 Γ.7 Τερματισμός συστήματος

Για τον τερματισμό των υπηρεσιών χρησιμοποιείται η εντολή:

```
docker compose down
```

7.26 Γ.7 Αποθετήριο πηγαίου κώδικα

Ο πηγαίος κώδικας, τα ενδεικτικά prompts και τα αρχεία παραμετροποίησης της εφαρμογής διατίθενται στο αποθετήριο:

<https://github.com/Alxiliadi/llm-multi-agent-scheduler>

Η διάθεση του αποθετηρίου επιτρέπει την αναπαραγωγή των πειραμάτων, την επιθεώρηση της υλοποίησης και την περαιτέρω επέκταση του προτεινόμενου πρωτοτύπου.