



Χαροκόπειο Πανεπιστήμιο

Τμήμα Πληροφορικής και Τηλεματικής

Πτυχιακή Εργασία

Μηχανισμός πρότασης σύνθετων εφαρμογών (gadgets) στο περιβάλλον του ακαδημαϊκού κοινωνικού δικτύου Unity με χρήση ευφυών τεχνικών

Τοπάλη Δέσποινα

AM : 20737

Επιβλέπων

Δημοσθένης Αναγνωστόπουλος, Καθηγητής

Μέλη εξεταστικής επιτροπής

Μάρα Νικολαΐδου, Καθηγήτρια

Ουρανία Χατζή, Διδάσκουσα βάσει Π.Δ 407/80

Φεβρουάριος 2013

Πίνακας περιεχομένων

Πρόλογος.....	6
Περίληψη.....	8
1. Εισαγωγή.....	10
1.1. Εννοιολογική θεμελίωση.....	10
1.2. Σύνθετη λειτουργικότητα	12
1.3. Σχεδιασμός Ενεργειών (AI Planning)	13
1.4. Αντικείμενο και στόχοι εργασίας	15
1.5. Δομή εργασίας	16
2. Βιβλιογραφική επισκόπηση	18
2.1. Εισαγωγή	18
2.2. Δυσκολίες στην υλοποίηση σύνθεσης.....	20
2.3. Σύνθεση υπηρεσιών ιστού με χρήση σχεδιασμού ενεργειών	21
2.3.1. Λογισμός καταστάσεων (Situation Calculus)	22
2.3.2. Planning μέσω προτύπου PDDL	23
2.3.3. Rule-based Planning (Σχεδιασμός βασισμένος στους κανόνες)	26
2.3.4. Case-based Planning (Σχεδιασμός βασισμένος στις περιπτώσεις)	28

2.3.5.	Άλλες τεχνικές.....	29
2.4.	Συνεισφορά της παρούσας εργασίας	30
3.	Περιγραφή, σχεδιασμός και υλοποίηση μηχανισμού σύνθεσης.....	32
3.1.	Περιγραφή Μηχανισμού Σύνθεσης	32
3.1.1.	Λειτουργικότητα μηχανισμού σύνθεσης.....	32
3.1.2.	Αναπαράσταση και Μετασχηματισμός Γνώσης	33
3.1.3.	Διασύνδεση με κοινωνικά δίκτυα	36
3.2.	Αλγόριθμος λειτουργίας	37
3.3.	Αρχιτεκτονική συστήματος	39
3.4.	Υλοποίηση Συστήματος	42
3.4.1.	Data Layer - Βάση Δεδομένων	42
3.4.2.	Business Logic Layer	43
3.4.2.1.	Web Service	43
3.4.2.2.	Εξωτερικό Σύστημα Σχεδιασμού - Planner	45
3.4.2.3.	Presentation Layer – User Interface.....	50
4.	Περιπτώσεις χρήσης	51
4.1.	Unity	54

4.1.1.	Αίτηση φοιτητή για κάρτα σίτισης.....	55
4.1.1.1.	Περιγραφή του προβλήματος.....	55
4.1.1.2.	Περιγραφή απαιτούμενων gadgets	57
4.1.1.3.	Περιγραφή λύσης.....	59
4.1.1.4.	Παρουσίαση αρχείων PDDL και SOL.....	61
4.1.1.5.	Screenshots	63
4.1.2.	Αίτηση αποφοίτησης Unity.....	64
4.1.2.1.	Περιγραφή του προβλήματος.....	64
4.1.2.2.	Περιγραφή απαιτούμενων gadgets	65
4.1.2.3.	Περιγραφή λύσης.....	67
4.1.2.4.	Παρουσίαση αρχείων PDDL και SOL.....	68
4.1.2.5.	Screenshots	70
4.2.	Εξωτερικό περιβάλλον	71
4.2.1.	Εργοδοτικές Εισφορές (ΙΚΑ).....	71
4.2.1.1.	Περιγραφή του προβλήματος.....	72
4.2.1.2.	Περιγραφή απαιτούμενων gadgets	73
4.2.1.3.	Περιγραφή λύσης.....	75

4.2.1.4.	Παρουσίαση αρχείων PDDL και SOL.....	77
4.2.1.5.	Screenshots	79
5.	Συμπεράσματα και Μελλοντικές κατευθύνσεις.....	80
	Βιβλιογραφία.....	82

Πρόλογος

Η εκπόνηση της παρούσας πτυχιακής εργασίας ξεκίνησε στο Τμήμα Πληροφορικής και Τηλεματικής του Χαροκοπείου Πανεπιστημίου τον Μάρτιο του 2012 και ολοκληρώθηκε τον Φεβρουάριο του 2013.

Αρχικά θα ήθελα να εκφράσω τις ευχαριστίες μου στον Πρύτανη του Χαροκοπείου Πανεπιστημίου και επιβλέπων της παρούσας πτυχιακής εργασίας, Καθηγητή Δημοσθένη Αναγνωστόπουλο.

Εν συνεχεία, θα ήθελα να εκφράσω τις ευχαριστίες μου προς την Καθηγήτρια Μάρα Νικολαΐδου, μέλος της εξεταστικής επιτροπής, για τις γνώσεις που μου παρείχε καθ' όλη τη διάρκεια των σπουδών μου. Ακόμα την ευχαριστώ θερμά για την κατανόηση που έδειξε σε όλα τα θέματα που με απασχόλησαν ως φοιτήτρια καθώς για την καθοδήγηση και τις συμβουλές που μου έδωσε τόσο σε επαγγελματικό όσο και σε προσωπικό επίπεδο.

Ακόμα, θα ήθελα να ευχαριστήσω τη διδάσκουσα βάσει Π.Δ. 407/80 Ουρανία Χατζή, για την αμέριστη προσοχή που μου έδωσε κατά την υλοποίηση της παρούσας εργασίας όπως και στα τόσα μαθήματα και εργασίες που απαιτούσε η σχολή. Θα ήθελα ακόμα να την ευχαριστήσω για το ενδιαφέρον που μου δημιούργησαν οι διαλέξεις της ώστε να ασχοληθώ περαιτέρω με τον τομέα της τεχνητής νοημοσύνης.

Ακόμα θα ήθελα να ευχαριστήσω τους Επίκουρους Καθηγητές Χρύσα Σοφianoπούλου και Θωμά Καμαλάκη, τους Λέκτορες Ηρακλή Βαρλάμη και Γεώργιο Δημητρακόπουλο καθώς και τους υποψήφιους διδάκτορες Αργυρή Τσαδήμα, Αλέξανδρο Νταή και Γεωργία Δέδε για την αναντικατάστατη συνεισφορά τους στις σπουδές μου.

Ακόμα θα ήθελα να εκφράσω τις ευχαριστίες μου στον εκλιπόντα πρώην Πρύτανη και Καθηγητή Γεώργιο Καραμπατζό, που δίδαξε σε όλους μας το ήθος, την επιμονή, τη συνεργασία, την κατανόηση και την υποστήριξη που χρειάζεται να δείχνουμε ακόμα και κάτω από ανταγωνιστικές συνθήκες. Τον ευχαριστώ ακόμα, για το κλίμα που δημιούργησε στο τμήμα μας και με έκανε να αισθανθώ τη σχολή και τους ανθρώπους της σαν οικογενειακό περιβάλλον

καθώς και για την υποστήριξη και συμπαράσταση που μου παρείχε τόσο σε μορφωτικό όσο και σε προσωπικό επίπεδο.

Επίσης, θα ήθελα να ευχαριστήσω όλους τους συμφοιτητές μου που συνεργαστήκαμε σε εργασίες και ιδιαίτερα τους Ιωάννη Μελετάκη και Στελίνα Λυμπεροπούλου, που αν και οι συνθήκες κάτω από πίεση δεν ήταν πάντα εύκολες καταφέραμε και τις ξεπεράσαμε.

Τέλος, θα ήθελα να ευχαριστήσω τη μητέρα μου για τη συνεχή συμπαράστασή της και τον τρόπο με τον οποίο με καθυσύχαιζε σε κάθε δύσκολη περίπτωση, όπως επίσης και τον αδερφό μου που αν και μικρότερος με βοηθούσε σε όποιο μάθημα μπορούσε. Ακόμα, θα ήθελα να ευχαριστήσω όλους τους φίλους μου που έκαναν υπομονή και μου συμπαραστάθηκαν τόσα χρόνια και ιδιαίτερα κατά την υλοποίηση της πτυχιακής.

Περίληψη

Τα κοινωνικά δίκτυα έχουν αναδειχθεί ως ένα νέο πρότυπο για την καθημερινή επικοινωνία, καθώς και για τη συνεργασία εντός των οργανισμών. Οι συμμετέχοντες σε ένα κοινωνικό δίκτυο, όχι μόνο ανταλλάσσουν πληροφορίες, αλλά επικαλούνται επίσης εξωτερικές εφαρμογές με τη μορφή των gadgets. Προκειμένου να παραχθεί σύνθετη λειτουργικότητα ο συνδυασμός διαφορετικών gadgets είναι απαραίτητος.

Η παρούσα εργασία αποσκοπεί στην αυτοματοποιημένη σύνθεση και πρόταση gadgets σε κοινωνικά δίκτυα ή άλλα δυναμικά συστήματα, με χρήση ευφών τεχνικών. Με αυτόν τον τρόπο, επιτυγχάνεται η προσφορά σύνθετης λειτουργικότητας στους χρήστες, χωρίς την υποχρέωση εκ μέρους τους να γνωρίζουν εκ των προτέρων τα απαιτούμενα gadgets και τον τρόπο με τον οποίον αυτά πρέπει να συνδυαστούν.

Αρχικά, μελετήθηκαν οι τεχνολογίες και προτάσεις που έχουν ήδη διατυπωθεί για την αυτοματοποιημένη σύνθεση διαδικτυακών υπηρεσιών καθώς και οι δυσκολίες που προκύπτουν κατά την υλοποίηση μηχανισμών που συνθέτουν δυναμικά λειτουργικότητα διαφορετικών εφαρμογών.

Στη συνέχεια, αναλύθηκαν οι απαιτήσεις που έχουν τεθεί για την παρούσα εργασία. Συγκεκριμένα, εξετάστηκε ο τρόπος με τον οποίο ορίζονται τα gadgets και πως αυτά μπορούν να συνδέονται μεταξύ τους. Αφού καταγράφηκαν οι λειτουργικές απαιτήσεις, στη συνέχεια παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του έργου.

Για την εφαρμογή του μηχανισμού υλοποιήθηκαν σενάρια χρήσης εντός και εκτός του ακαδημαϊκού κοινωνικού δικτύου Unity. Συγκεκριμένα, μελετήθηκαν οι περιπτώσεις που αφορούν στην αίτηση για Κάρτα Σίτισης ενός φοιτητή, στην αίτηση Αποφοίτησης καθώς και σε ένα παράδειγμα εκτός κοινωνικού δικτύου, στην καταβολή εργοδοτικών εισφορών στο ΙΚΑ.

Abstract

Social networks have emerged as a new paradigm for everyday communication, as well as for collaboration within organizations. Participants in a social network not only exchange information but also invoke external applications in the form of gadgets. In order to provide enhanced, complex functionality to participants enabling them to perform specific tasks, the combination of gadgets is required.

This thesis has implemented a recommendation mechanism for automated gadget composition in social networks or other dynamic environments, utilizing intelligent techniques. This achieves enhanced functionality, without requiring users to be aware in advance of the appropriate gadgets involved, or the way they should interact.

Initially, literature regarding automated web service composition has been studied, and difficulties occur during implementation of such mechanisms have been identified. Moreover, system requirements have been analyzed. Particularly, gadget definition has been examined and the way they can be combined has been explored. After identifying functional requirements, technologies that have been used in implementation are presented.

The applicability of the proposed system was validated using scenarios concerning the application of mechanism in the Unity academic social network. In particular, case studies that concern Student Meal Card application, Graduation application and Employer Insurance Tax Payment have been studied.

1. Εισαγωγή

Ένα από τα κυριότερα δομικά στοιχεία στην επιστήμη των υπολογιστών, είτε στο επίπεδο του υλικού (hardware) είτε του λογισμικού (software), είναι οι συνιστώσες (components). Στο επίπεδο του υλικού, ως συνιστώσες ορίζονται το σύνολο των φυσικών εξαρτημάτων ενός υπολογιστή όπως π.χ. ηλεκτρικά και ηλεκτρονικά στοιχεία, τα οποία συνθέτουν ολόκληρο το υπολογιστικό σύστημα και συνεργαζόμενα το καθιστούν λειτουργικό. Το επίπεδο του λογισμικού αποτελείται από ένα σύνολο προγραμμάτων, διαδικασιών και συνιστωσών που εκτελούν συγκεκριμένες εργασίες σε ένα υπολογιστικό σύστημα, και καλύπτουν το σύνολο της παρεχόμενης λειτουργικότητας. Παρατηρώντας και τους δύο κόσμους φαίνεται ότι οι συνιστώσες του καθενός, παρά τη διαφορετικότητά τους, είναι απαραίτητες για να λειτουργήσουν κάποια δεδομένα εισόδου (Inputs) και εξόδου (Outputs). Στην περίπτωση του υλικού ως Inputs και Outputs (I/O) ορίζονται οι συσκευές εισόδου: Πληκτρολόγιο, Ποντίκι, Σαρωτής (Scanner) και οι συσκευές εξόδου: Οθόνη, Ηχεία, Εκτυπωτής.

Στην περίπτωση του λογισμικού, οποιαδήποτε αλληλεπίδραση του χρήστη με μία εφαρμογή, ακολουθεί μια προβλέψιμη διαδικασία, της αίτησης μιας λειτουργικότητας και το αποτέλεσμα αυτής. Η παραπάνω αρχή εφαρμόζεται από την πιο απλή λειτουργικότητα που μπορεί να εκτελέσει ένας υπολογιστής μέχρι την πιο σύνθετη. Ένα απλό παράδειγμα είναι οι πράξεις μεταξύ δύο ή περισσότερων αριθμών, οι οποίες χαρακτηρίζονται ως inputs και το αποτέλεσμα αυτών το output, Παρόμοια διαδικασία ισχύει και σε μια πιο σύνθετη λειτουργικότητα, όπως η ανάγκη ενός χρήστη να κάνει μια αναζήτηση στο διαδίκτυο (input) και να δει τα αποτελέσματά της (output).

1.1. Εννοιολογική θεμελίωση

Ως τεχνολογία λογισμικού βασισμένη σε **components** (component-based software engineering), ορίζεται η προσέγγιση επαναχρησιμοποίησης του ορισμού, της υλοποίησης και της σύνθεσης ανεξάρτητων components σε ένα σύστημα. Ένα ανεξάρτητο component είναι ένα πακέτο λογισμικού που υλοποιεί μια σειρά από λειτουργίες, παρέχοντας στον τελικό χρήστη καλά

ορισμένη και συγκεκριμένη λειτουργικότητα, και χαρακτηρίζεται από καλά ορισμένες εισόδους και εξόδους. Ένα component μπορεί να απαρτίζεται από:

- ένα Web Service, θεωρείται ένα σύστημα λογισμικού σχεδιασμένο να υποστηρίζει διάδραση μεταξύ μηχανημάτων πάνω από τον ιστό και να προσφέρει μια συγκεκριμένη λειτουργικότητα στο χρήστη.
- ένα gadget, στο χώρο του λογισμικού, ορίζεται ένα πρόγραμμα που προσφέρει μια υπηρεσία, χωρίς να χρειάζεται να τρέχει μία ανεξάρτητη εφαρμογή για το κάθε gadget, αντίθετα μπορεί να τρέχει σε ένα περιβάλλον το οποίο μπορεί να διαχειριστεί όλα τα αυτά μαζί.
- ένα πρόσθετο (plugin), ορίζεται ένα σύνολο από components που προσθέτει συγκεκριμένες ικανότητες σε μια μεγαλύτερη εφαρμογή λογισμικού, αν υποστηρίζεται από αυτή.
- μια βιβλιοθήκη (library), ορίζεται μια συλλογή από έτοιμα υποπρογράμματα που χρησιμοποιείται για την ανάπτυξη λογισμικού. Οι βιβλιοθήκες περιέχουν υποβοηθητικό κώδικα και δεδομένα, παρέχοντας, με αυτόν τον τρόπο, υπηρεσίες σε προγράμματα.

Όλες οι διεργασίες που χρειάζονται σε ένα σύστημα χωρίζονται σε διαφορετικά components έτσι ώστε οι λειτουργίες και τα δεδομένα να σχετίζονται μεταξύ τους σημασιολογικά. Πολλές φορές ένα component για να λειτουργήσει, χρειάζεται τη λειτουργικότητα κάποιου άλλου component (Wikipedia, 2008).

Αναφορικά, ένας τύπος διαδικτυακών υπηρεσιών (web services) είναι αυτά που παρέχει το Amazon, γνωστά ως Amazon Web Services (AWS). Είναι μία συλλογή από ανεξάρτητες υπηρεσίες που παρέχονται στο ευρύ κοινό προκειμένου να καλύψουν λειτουργικότητες σε διάφορα συστήματα. Η συγκεκριμένη προσέγγιση του Amazon δείχνει, πως ανεξάρτητες υπηρεσίες μπορούν να ικανοποιήσουν λειτουργικότητα που επαναχρησιμοποιείται σε διαφορετικά συστήματα και από διαφορετικούς ανθρώπους.

Οι υπηρεσίες που αναφέρθηκαν παραπάνω εφαρμόζονται ολοένα και περισσότερο σε διαδικτυακούς τόπους, στα κοινωνικά δίκτυα και σε διάφορες διαδικτυακές πύλες. **Κοινωνικό δίκτυο** (social network) είναι μια δομή από κόμβους, κάθε ένας από τους οποίους

αντιπροσωπεύει ένα άτομο ή οργανισμό. Οι κόμβοι αυτοί είναι συνδεδεμένοι μεταξύ τους με μία σχέση αλληλεξάρτησης (interdependence), όπως για παράδειγμα φιλία, συγγένεια, σεξουαλικό ενδιαφέρον, οικονομική συναλλαγή, κοινά ενδιαφέροντα και πεποιθήσεις (Wikipedia, 2011). **Διαδικτυακή πύλη** (portal) είναι ένας ιστότοπος που προσφέρει ένα ευρύ φάσμα διαφορετικών υπηρεσιών ενοποιημένες σε μία, όπως ηλεκτρονικό ταχυδρομείο, μηχανές αναζήτησης, νέα, ψυχαγωγία κι άμεση σύνδεση σε καταστήματα (Wikipedia).

Η χρήση των κοινωνικών δικτύων αποτελεί την πιο δημοφιλή online δραστηριότητα των χρηστών που αλληλεπιδρούν με ηλεκτρονικούς υπολογιστές, κινητά, tablets. Η αρχική ανάγκη των χρηστών δημιουργήθηκε στο να γνωρίσουν άλλους ανθρώπους, να αναπτύξουν φιλίες ή επαγγελματικές συμμαχίες, να αναζητήσουν εργασία, να σχηματίσουν ομάδες και να μοιραστούν πληροφορίες και εμπειρίες σχετικά με τα χόμπι και τα ενδιαφέροντά τους. Σήμερα, εκτός από τις παραπάνω ανάγκες τους οι χρήστες, εκτελούν και λειτουργικότητα παρεχόμενη από ανεξάρτητα αλλά άρτια συνδεδεμένα components εντός του διαδικτυακού χώρου. Συγκεκριμένα, ζητούν άτομα, σελίδες, εφαρμογές, παιχνίδια, τα οποία είτε μπορεί να είναι απλά στη χρήση και τη λειτουργικότητα τους είτε να απαιτούν σύνθεση άλλων gadgets ώστε να επιτευχθεί η τελική λειτουργικότητα που αρχικά αιτήθηκε ο χρήστης.

1.2. Σύνθετη λειτουργικότητα

Στις περιπτώσεις που προαναφέρθηκαν, δηλαδή στα κοινωνικά δίκτυα και στις διαδικτυακές πύλες, υπάρχουν πολλά παραδείγματα που ανεξάρτητες εφαρμογές ενσωματώνονται προκειμένου να προσδώσουν μια λειτουργικότητα στο σύστημα, είτε με την μορφή gadgets είτε με την μορφή άλλου είδους components. Η λειτουργικότητα αυτή αντιπροσωπεύει την ενέργεια που ο χρήστης επιθυμεί να εκτελέσει. Σε συνδυασμό με τη λειτουργικότητα, πολλές από αυτές τις εφαρμογές απαιτούν συγκεκριμένα στοιχεία εισόδου για να μπορέσουν να αποδώσουν το τελικό επιθυμητό αποτέλεσμα, και σε αρκετές περιπτώσεις με συγκεκριμένη μορφή. Στην περίπτωση που κάποιος χρήστης αναζητά κάποια εφαρμογή, προκειμένου αυτή να επιτρέψει στο χρήστη να την εκτελέσει απαιτεί κάποια στοιχεία είτε αυθεντικοποίησης, όπως το όνομα χρήστη, τον κωδικό πρόσβασης είτε στοιχεία που προέρχονται από άλλες εξωτερικές εφαρμογές.

Στις περιπτώσεις που η τελική λειτουργικότητα προϋποθέτει ήδη εγκατεστημένες εφαρμογές, ο χρήστης χρειάζεται να αναζητήσει χειροκίνητα τις υπηρεσίες που χρειάζονται και στη συνέχεια να τις εκτελέσει ξεχωριστά, γεγονός που καθιστά το σύστημα χρονοβόρο και αναποτελεσματικό. Αν στο σύστημα του χρήστη, όπως προαναφέρθηκε δεν περιέχονται όλοι οι διαθέσιμοι πόροι ή εφαρμογές, ώστε να εκτελεστεί η λειτουργικότητα ενός ατομικού Web Service, υπάρχει η δυνατότητα να χρησιμοποιηθούν και να συντεθούν μαζί υπάρχοντες υπηρεσίες με σκοπό να ολοκληρωθεί η αίτηση του χρήστη (web service composition). Με τη σύνθεση υπηρεσιών γίνεται πιο εύκολη και προσιτή η αναζήτηση των χρηστών, έχοντας ως αποτέλεσμα την εξοικονόμηση χρόνου. Ωστόσο, όσο περισσότερα στοιχεία χρειάζεται να συντεθούν για την τελική απαίτηση του χρήστη, τόσο πιο πολύπλοκη γίνεται η διαδικασία σύνθεσης υπηρεσιών.

Για αυτό το λόγο, δημιουργείται η ανάγκη ενός μηχανισμού πρότασης, σύνθεσης και εκτέλεσης πολλαπλών υπηρεσιών, με τρόπο ώστε να συνδυάζονται για να επιτύχουν πιο σύνθετη και εξελιγμένη λειτουργικότητα. Η συγκεκριμένη ανάγκη είναι εφικτό να υλοποιηθεί και να τεθεί σε εφαρμογή με χρήση ευφυών τεχνικών, όπως ο σχεδιασμός ενεργειών. Αποτέλεσμα της υλοποίησης ενός μηχανισμού πρότασης components με τη λογική αλληλουχία που θα χρειαζόταν ο χρήστης να τα εκτελέσει, του εξασφαλίζει χρόνο και ενέργεια.

Ο παραπάνω μηχανισμός μπορεί να εγκατασταθεί σε οποιοδήποτε portal ή κοινωνικό δίκτυο, τα οποία για τη λειτουργία αυτού χρειάζεται να παρέχουν ένα τρόπο αναπαράστασης της λειτουργικότητας των components σε μια βάση δεδομένων, με πρότυπο τρόπο. Η αναπαράσταση χρειάζεται να αναφέρει τα δεδομένα εισόδου και εξόδου του κάθε component ή gadget, για να μπορεί ο μηχανισμός πρότασης να συνθέσει τη ροή των προαπαιτούμενων ώστε να δοθεί το τελικό αποτέλεσμα στο χρήστη.

1.3. Σχεδιασμός Ενεργειών (AI Planning)

Η ικανότητα επίλυσης προβλημάτων αποτελεί ένα από τα βασικά χαρακτηριστικά γνωρίσματα της νοημοσύνης. Συνεπώς πρέπει να αποτελεί πρωταρχική ιδιότητα οποιασδήποτε οντότητας που επιδεικνύει νοημοσύνη. Η έρευνα κατά τη διάρκεια των πρώτων χρόνων της Τεχνητής Νοημοσύνης έπρεπε να καταλήξει σε τυποποιημένες μεθοδολογίες με τις οποίες να είναι δυνατή αφενός η περιγραφή προβλημάτων και αφετέρου η επίλυσή τους. Η τομέας της Τεχνητής

Νοημοσύνης προσπαθεί να σχεδιάσει προγράμματα που, πρώτα, αναλύουν και επεξεργάζονται βήμα προς βήμα τη μέθοδο για την επίλυση ενός προβλήματος και στη συνέχεια θα εκτελούν τα βήματα (Durkin, 1994).

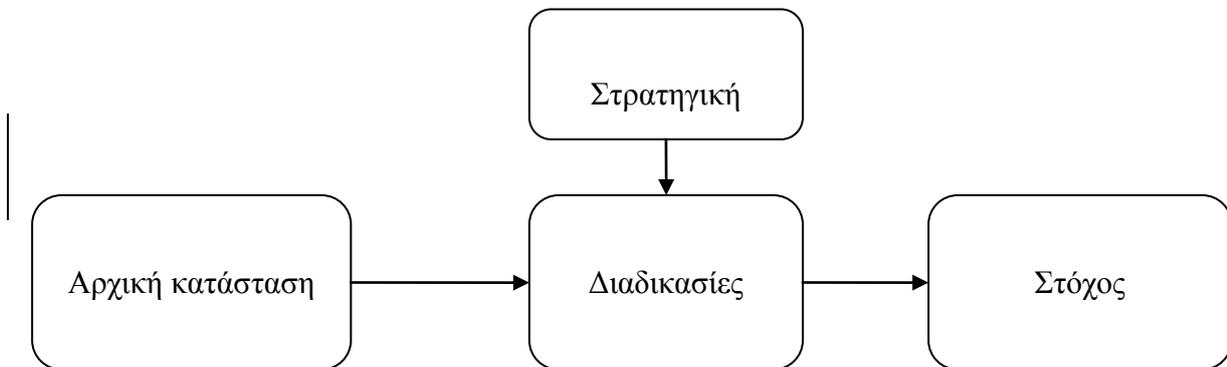
Ως αυτοματοποιημένος σχεδιασμός (automated planning) και προγραμματισμός ορίζεται ο κλάδος της τεχνητής νοημοσύνης, που αφορά στην υλοποίηση στρατηγικών και εκτελέσιμων ακολουθιών από ένα ευφυή και αυτόνομο πράκτορα (agent) ή ρομπότ. Αντίθετα, με τις κλασσικές περιπτώσεις, οι λύσεις είναι πολύπλοκες και χρειάζεται να ανακαλυφθούν και να βελτιστοποιηθούν (Wikipedia).

Ως προβλήματα σχεδιασμού ενεργειών (planning problems) ορίζονται εκείνα στα οποία η τελική κατάσταση είναι πλήρως γνωστή και επιδιώκεται η εύρεση μιας ακολουθίας ενεργειών, μέσω της διαδικασίας σχεδιασμού ενεργειών. Η ακολουθία των ενεργειών που αποτελεί τη λύση ενός προβλήματος σχεδιασμού, ονομάζεται πλάνο (plan) ενώ το πρόγραμμα που την παράγει ονομάζεται σχεδιαστής (planner). Οι κλασσικοί αλγόριθμοι αναζήτησης είναι ανεπαρκείς για την επίλυση τέτοιων προβλημάτων, για αυτό χρησιμοποιούνται ειδικές γλώσσες περιγραφής προβλημάτων, που μπορούν να υποστηρίξουν την εφαρμογή κατάλληλων αλγορίθμων.

Ένας χώρος απεικόνισης κατάστασης (state space representation) παρέχει ένα πλαίσιο μοντελοποίησης και επικοινωνίας προβλημάτων και των λύσεών του. Η αναπαράσταση σε ένα χώρο καταστάσεων είναι ένα μοντέλο του κόσμου η οποία μπορεί να ποικίλλει από απλό έως πολύπλοκο. Στην πραγματικότητα, ένα πρόβλημα θεωρείται πολύπλοκο ανάλογο με τον τρόπο αναπαράστασής του. Οι τελεστές και οι κανόνες είναι ενέργειες (actions) που περιγράφουν τις πιθανές μεταβολές μεταξύ των καταστάσεων. Η αρχική κατάσταση (initial state) ορίζεται από τις προδιαγραφές από τις οποίες η διαδικασία επίλυσης αρχίζει ενώ η κατάσταση στόχου (goal state) ορίζει τις προδιαγραφές για μια επιτυχή λύση.

Γενικά, ένα πρόβλημα σχεδιασμού μπορεί να περιγραφεί ως ένας συνδυασμός από πέντε πλειάδες $\{S, S_0, G, A, \Gamma\}$, όπου το S αναπαριστά όλες τις πιθανές καταστάσεις που μπορούν να παρατηρηθούν στο περιβάλλον του προβλήματος, το $S_0 \subset S$ υποδηλώνει την αρχική κατάσταση του περιβάλλοντος, το $G \subset S$ αναφέρεται στο στόχο ή στην τελική κατάσταση που το σύστημα προσπαθεί να φτάσει, το A υποδηλώνει ένα σύνολο ενεργειών που το σύστημα μπορεί να εφαρμόσει ώστε να περάσει από μία κατάσταση που βρίσκεται το σύστημα σε μία άλλη, και

τέλος η σημασία του $\Gamma \subseteq S \times A \times S$ ορίζει τις προϋποθέσεις και τα αποτελέσματα της εκτέλεσης για κάθε ενέργεια. Συγκεκριμένα, το S_0 και το G είναι η αρχική και τελική κατάσταση που ο χρήστης ζητά από την υπηρεσία. A είναι ένα σύνολο με τις διαθέσιμες υπηρεσίες και το Γ υποδηλώνει την αλλαγή της κατάστασης που παρέχει η κάθε υπηρεσία. Η διαδικασία επίλυσης ενός προβλήματος δίνεται στην Εικόνα 1.1.



Εικόνα 1.1 Διαδικασία επίλυσης (πηγή: Frenzel, 1987; Turban, 1993)

1.4. Αντικείμενο και στόχοι εργασίας

Στο πλαίσιο της ερευνητικής δραστηριότητας του τμήματος Πληροφορικής και Τηλεματικής του Χαροκοπέιου Πανεπιστημίου δημιουργήθηκε και αναπτύχθηκε το κοινωνικό δίκτυο Unity, με στόχο την διευκόλυνση της επικοινωνίας και της αλληλεπίδρασης μεταξύ των μελών της ακαδημαϊκής κοινότητας, παρέχοντας τις λειτουργικότητες που προσφέρουν όλα τα κοινωνικά δίκτυα. Στο κοινωνικό δίκτυο Unity δόθηκε η δυνατότητα επεκταμένης λειτουργικότητας μέσω gadgets, ώστε να καλυφθούν κι άλλες ανάγκες της ακαδημαϊκής κοινότητας. Κατά τη διάρκεια της πιλοτικής λειτουργίας και αξιολόγησης του Unity διαπιστώθηκε η ανάγκη για την δημιουργία σύνθετων ροών εργασιών (workflows), οι οποίες έχουν ως στόχο την ολοκλήρωση διαδικασιών της ακαδημαϊκής κοινότητας, που απαιτούν την εμπλοκή πολλών χρηστών με διαφορετικούς ρόλους, καθένας από τους οποίους εκτελεί gadgets μέσα στο κοινωνικό δίκτυο. Επομένως, προέκυψε η ανάγκη δημιουργίας ενός μηχανισμού για την αυτοματοποίηση της δημιουργίας τέτοιων ροών εργασιών κάτω από συγκεκριμένες προϋποθέσεις.

Η παρούσα εργασία έχει ως αντικείμενο τη δημιουργία ενός μηχανισμού πρότασης σύνθετων εφαρμογών (gadgets) με χρήση ευφών τεχνικών. Ο μηχανισμός αυτός ολοκληρώνεται στο

προαναφερθέν περιβάλλον του ακαδημαϊκού κοινωνικού δικτύου Unity. Το Unity απαρτίζεται από ποικίλα gadgets ανεξάρτητα μεταξύ τους, που το καθένα από αυτά παρέχει στο χρήστη μια συγκεκριμένη λειτουργία. Επιπλέον, τα gadgets διαθέτουν περιγραφή, τα δεδομένα εισόδου που απαιτούν καθώς και τα αποτελέσματα παράγουν μετά την εκτέλεσή τους. Η παραπάνω πληροφορία αναπαρίσταται σε σχεσιακή βάση δεδομένων. Το περιβάλλον εκτός από την εγκατάσταση και εκτέλεση των gadgets, υποστηρίζει και την ύπαρξη διαφορετικών ρόλων, ανάλογα με τη θέση που κατέχει ο κάθε χρήστης στο ίδρυμα, η οποία μπορεί να είναι φοιτητή, καθηγητή, διοικητικού προσωπικού κ.α. Ανάλογα, το ρόλο του που διαθέτει ο χρήστης έχει δικαιώματα να εκτελέσει και τα ανάλογα gadgets.

Συγκεκριμένα, οι διάφοροι τύποι χρηστών στο Unity μπορούν να ζητήσουν από το σύστημα μέσω κλήσεων gadgets μια απλή λειτουργικότητα. Όταν οι απαιτήσεις των χρηστών γίνουν πιο πολύπλοκες και δεν είναι δυνατόν να υλοποιηθούν από μία μόνο εφαρμογή, τότε είναι απαραίτητη η σύνθεση εφαρμογών. Με τον όρο σύνθεση εφαρμογών τόσο στο συγκεκριμένο περιβάλλον όσο και σε οποιοδήποτε άλλο, εννοείται ότι για να εκτελεστεί το τελικό component χρειάζεται να έχουν εγκατασταθεί στο σύστημα τα προαπαιτούμενα αυτού. Ως επακόλουθο, με το συνδυασμό των μεμονωμένων components επεκτείνεται η λειτουργικότητα προσφέροντας στο σύστημα νέες υπηρεσίες.

Στόχος της συγκεκριμένης εργασίας είναι να παρέχει στο χρήστη έναν έξυπνο μηχανισμό ώστε να γνωρίζει όλη τη ροή των gadgets που χρειάζεται να εκτελέσει ώστε να επιτύχει τον τελικό του στόχο, δηλαδή την εκτέλεση της τελικής λειτουργικότητας. Για σκοπούς διευκόλυνσης προς το χρήστη δόθηκε περαιτέρω λειτουργικότητα οπτικοποίησης των διαθέσιμων και προαπαιτούμενων υπηρεσιών. Η παρούσα αναπτύχθηκε σε παραδείγματα (gadgets) που έχουν ήδη υλοποιηθεί και είναι διαθέσιμα στο περιβάλλον του Unity.

1.5. Δομή εργασίας

Η παρούσα εργασία είναι διαρθρωμένη ως εξής:

Το 2^ο Κεφάλαιο εκθέτει όλες τις συσχετιζόμενες τεχνολογίες, υλοποιήσεις και προτάσεις σχετικά με την αυτοματοποιημένη ή ημι-αυτοματοποιημένη σύνθεση διαδικτυακών υπηρεσιών.

Γίνεται αναφορά στις δυσκολίες που προκύπτουν κατά την υλοποίηση μηχανισμών που συνθέτουν δυναμικά λειτουργικότητα διαφορετικών υπηρεσιών. Ακόμα, περιγράφονται οι τρόποι με τους οποίους επιχειρείται η αυτοματοποιημένη σύνθεση υπηρεσιών, όπως είναι ο λογισμός καταστάσεων (Situation Calculus), ο σχεδιασμός μέσω προτύπου PDDL, ο σχεδιασμός βασισμένος στους κανόνες, ο σχεδιασμός βασισμένος στις περιπτώσεις καθώς και διάφορες άλλες τεχνικές.

Στο 3^ο Κεφάλαιο περιγράφονται ο σχεδιασμός, η αρχιτεκτονική, η υλοποίηση και ο αλγόριθμος λειτουργίας του μηχανισμού σύνθεσης. Αναλύονται οι απαιτήσεις που έχουν τεθεί για την παρούσα εργασία και ο τρόπος με τον οποίο σχεδιάστηκε το σύστημα ώστε να ικανοποιεί τις απαιτήσεις αυτές. Παρουσιάζονται, ακόμα, οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του έργου. Τέλος, παρατίθεται η αλγοριθμική ακολουθία που εκτελούν οι διαφορετικές οντότητες του συστήματος καθώς και ο τρόπος με τον οποίο αυτές επικοινωνούν μεταξύ τους.

Το 4^ο Κεφάλαιο προβάλλονται σενάρια χρήσης εντός και εκτός του ακαδημαϊκού κοινωνικού δικτύου Unity. Συγκεκριμένα, περιγράφεται αναλυτικά το πρόβλημα του κάθε σεναρίου χρήσης καθώς και τα gadgets που απαιτούνται ώστε να ολοκληρωθεί το αίτημα του χρήστη. Παρουσιάζονται, ακόμα, η λύση που δίνεται καθώς και τα αρχεία του συστήματος που δημιουργούνται δυναμικά ανάλογα το πρόβλημα. Τέλος, για κάθε περίπτωση χρήσης παρουσιάζονται εικόνες (screenshots) που αναμένεται να δει ο χρήστης.

Στο 5^ο Κεφάλαιο παρουσιάζονται τα συμπεράσματα που προέκυψαν από την υλοποίηση της εργασίας. Προτείνονται, ακόμα, μελλοντικές υλοποιήσεις που θα μπορούσαν να εφαρμοστούν στο μηχανισμό και αναφέρονται τρόποι στους οποίους θα μπορούσαν να στηριχθούν οι μελλοντικές υλοποιήσεις.

2. Βιβλιογραφική επισκόπηση

2.1. Εισαγωγή

Σχετιζόμενη έρευνα με την παρούσα εργασία έχει πραγματοποιηθεί εκτενώς όσον αφορά την λειτουργικότητα με την μορφή διαδικτυακών υπηρεσιών (web services), αλλά δεν έχει μελετηθεί σε άλλους τομείς, όπως π.χ. gadgets. Το παρόν κεφάλαιο περιγράφει συστήματα που αφορούν σύνθεση διαδικτυακών υπηρεσιών (web service composition) μέσω σχεδιασμού ενεργειών. Οι ιδέες/τεχνικές αυτές μπορούν να εφαρμοστούν, και επιχειρούμε να τις εφαρμόσουμε στην παρούσα εργασία, για λειτουργικότητα με την μορφή gadgets σε ένα κοινωνικό δίκτυο

Οι υπηρεσίες Ιστού (Web Services) θεωρούνται αυτοτελείς και **αυτοπεριγραφικές** (self-describing) εφαρμογές που μπορούν να δημοσιευτούν, τοποθετηθούν και να κληθούν μέσω του διαδικτύου. Ως αυτοπεριγραφικές, ορίζονται οι εφαρμογές ή οι υπηρεσίες οι οποίες περιέχουν στην περιγραφή τους όλες τις απαραίτητες πληροφορίες και στοιχεία, για να περιγράψουν τη λειτουργικότητά τους, τον τρόπο κλήσης τους καθώς και τα αποτελέσματα που θα επιστρέψουν. Στις μέρες μας, ένας αυξανόμενος αριθμός εταιρειών και οργανισμών εφαρμόζουν αποκλειστικά τις δικές τους επιχειρηματικές δραστηριότητες στον πυρήνα της επιχείρησής τους και αναθέτουν άλλες λειτουργικότητες σε εξωτερικές εφαρμογές που λειτουργούν πάνω από το δίκτυο. Έτσι, η ικανότητα να επιλεχθεί αποτελεσματικά και να αλληλεπιδράσει με ετερογενή συστήματα μία υπηρεσία είναι ένα σημαντικό βήμα προς ανάπτυξη όσον αφορά στις εφαρμογές και υπηρεσίες Ιστού.

Η έρευνα στην περιοχή των Web Services, έχει καταλήξει σε πρότυπα τα οποία που διευκολύνουν την αλληλεπίδραση μεταξύ ετερογενών συστημάτων. Τέτοια πρότυπα για τον τρόπο ανακάλυψης υπηρεσιών, την περιγραφή και την επίκληση τους είναι:

1. Universal Description, Discovery, and Integration (UDDI) [2], θεωρείται μια πλατφόρμα ανεξάρτητη, βασισμένη Extensive Markup Language (XML), με την οποία οι επιχειρήσεις μπορούν παγκοσμίως να δηλώσουν την παρουσία τους στο διαδίκτυο, καθώς και να εγγράψουν αλλά και να αναζητήσουν εφαρμογές web service,

2. Web Services Description Language (WSDL) [5], θεωρείται μία γλώσσα βασισμένη σε Extensive Markup Language (XML), που χρησιμοποιείται για να περιγράψει τη λειτουργικότητα που προσφέρεται από ένα web service,
3. Simple Object Access Protocol (SOAP) [3], ορίζεται ως ένα πρότυπο πρωτόκολλου για την ανταλλαγή μηνυμάτων δομημένης πληροφορίας της υλοποίησης ενός web service,
4. OWL-S (παλαιότερα DAML-S) [6], ορίζεται ως μια οντολογία, ανεπτυγμένη βάση του OWL framework, για την περιγραφή των web services,
5. Semantic Annotations for WSDL and XML Schema (SAWSDL) [30], ορίζει ένα σύνολο από επεκτάσεις χαρακτηριστικών για την WSDL και την XML.

Κάποιες άλλες προτάσεις σύνθεσης υπηρεσιών σε ένα συγκεκριμένο πρόγραμμα ροής είναι οι Business Process Execution Language for Web Service (BPEL4WS) [1] και DAML-S Service Model. Οι δύο αυτοί τρόποι επιλογής σύνθεσης υπηρεσιών παρουσιάζουν μειονεκτήματα στη χρήση τους καθώς θεωρείται απαραίτητη προϋπόθεση για αυτούς η ροή της διαδικασίας και η σύνδεση μεταξύ των υπηρεσιών να είναι γνωστές εκ των προτέρων. Τα δεδομένα εισόδου και εξόδου, όπως και όλοι οι υπόλοιποι παράμετροι του προβλήματος χρειάζεται να είναι γνωστά από την αρχή της σύνθεσης. Ένα ακόμη μειονέκτημα θεωρείται η δυσκολία επαναχρησιμοποίησης του μοντέλου που δημιουργήθηκε, καθώς είναι σχεδόν ελάχιστες οι περιπτώσεις στις οποίες όλοι οι παράμετροι ενός προβλήματος θα είναι ίδιες με το αρχικό μοντέλο που αναπτύχθηκε. Ακόμα, εξαρτώνται από την παρέμβαση του ανθρώπου σε χαμηλού επιπέδου γλώσσα προγραμματισμού, κάτι που θεωρείται σύνθετο και είναι επιρρεπές σε λάθη.

Παρά τις προσπάθειες που έχουν γίνει για αυτόματη σύνθεση υπηρεσιών Ιστού, ώστε να προσφέρουν μία συγκεκριμένη λειτουργικότητα, παραμένει μία σύνθετη εργασία. Αναφορικά, πολλοί λόγοι συμβάλλουν στη δυσκολία της αυτόματης δημιουργίας σύνθεσης υπηρεσιών, όπως ο μεγάλος αριθμός web services που προσφέρονται στο διαδίκτυο, το πόσο ξεκάθαρα και περιγραφικά δίνεται η λειτουργικότητα της κάθε υπηρεσίας καθώς και η ποικιλομορφία των εννοιολογικών μοντέλων των υπηρεσιών σύμφωνα με τις ανάγκες του μοντέλου αλλά και την οπτική των προγραμματιστών.

2.2. Δυσκολίες στην υλοποίηση σύνθεσης

Σύμφωνα με τις παραπάνω αναφορές υπάρχουν αρκετές δυσκολίες στην εύρεση και σύνθεση υπηρεσιών (discovery and composition of services). Αναλυτικότερα, οι δυσκολίες που προκύπτουν εντοπίζονται στις ακόλουθες πηγές.

- Αρχικά, ο αριθμός των υπηρεσιών που είναι διαθέσιμος στο διαδίκτυο αυξάνεται δραματικά τα τελευταία χρόνια. Παρά αυτή τη σημαντική αύξηση, οι υπηρεσίες δεν βρίσκονται συγκεντρωμένες σε κάποιο κεντρικοποιημένο μητρώο. Με αυτό το μειονέκτημα της διαχείρισης των υπηρεσιών, παραμένει δύσκολη διαδικασία και η εύρεση του σωστού συνδυασμού υπηρεσιών που έχει ζητηθεί από το χρήστη και η καταχώρηση μοναδικά ορισμένων υπηρεσιών από επιχειρήσεις ή άτομα.
- Δεύτερον, οι υπηρεσίες Ιστού (Web Services) μπορούν να δημιουργηθούν και να επεξεργαστούν ή να αντικατασταθούν οποιαδήποτε στιγμή, με αποτέλεσμα το σύστημα σύνθεσης υπηρεσιών να χρειάζεται να ανιχνεύει τις αλλαγές που έχουν γίνει σε πραγματικό χρόνο και η απόφαση σύνθεσης να χρειαστεί να γίνει στην ανανεωμένη πληροφορία.
- Τέλος, οι υπηρεσίες Ιστού μπορούν να αναπτυχθούν από διαφορετικούς οργανισμούς και άτομα, οι οποίοι χρησιμοποιούν διαφορετικά μοντέλα για να περιγράψουν τις υπηρεσίες. Ωστόσο δεν υπάρχει κάποια μοναδική γλώσσα που να προσδιορίζει και να εκτιμά τις υπηρεσίες με τον ίδιο τρόπο.

Επομένως, η ανάπτυξη σύνθεσης υπηρεσιών με αυτόματο ή ημιαυτόματο τρόπο είναι κρίσιμη. Αρκετές μέθοδοι έχουν προταθεί για αυτό το σκοπό. Συγκεκριμένα, οι πιο πρόσφατες έρευνες που έχουν διατυπωθεί εμπίπτουν στο χώρο της σύνθεσης της ροής εργασίας (workflow) με χρήση ευφυών τεχνικών που εμπίπτουν στην ευρύτερη περιοχή του σχεδιασμού ενεργειών (AI Planning).

Οι υπηρεσίες Ιστού προσδιορίζονται από τις προϋποθέσεις και τα αποτελέσματα που θα προκύψουν στο πλαίσιο σχεδιασμού. Αρχικά, ένα web service μπορεί να θεωρηθεί ως ένα αυτόνομο τμήμα λογισμικού που δέχεται δεδομένα εισόδου και παράγει δεδομένα εξόδου. Έτσι, οι προϋποθέσεις και τα αποτελέσματα είναι η είσοδος και η έξοδος των παραμέτρων της

υπηρεσίας αντίστοιχα. Ένα επιπρόσθετο χαρακτηριστικό είναι ότι μετά την εκτέλεση η υπηρεσία μεταβάλλει την κατάσταση του περιβάλλοντος στο οποίο έχει διεξαχθεί.

2.3. Σύνθεση υπηρεσιών ιστού με χρήση σχεδιασμού ενεργειών

Όπως έχει προαναφερθεί πολλές μελέτες που έχουν διατυπωθεί διαχειρίζονται τη σύνθεση υπηρεσιών Ιστού μέσω σχεδιασμού ενεργειών. Γενικά, ένα πρόβλημα σχεδιασμού (planning) μπορεί να περιγραφεί ως ένας συνδυασμός από πέντε πλειάδες $\{S, S_0, G, A, \Gamma\}$, όπου το S αναπαριστά όλες τις πιθανές καταστάσεις που μπορούν να παρατηρηθούν στο περιβάλλον του προβλήματος, το $S_0 \subset S$ υποδηλώνει την αρχική κατάσταση του περιβάλλοντος, το $G \subset S$ αναφέρεται στο στόχο ή στην τελική κατάσταση που το σύστημα προσπαθεί να φτάσει, το A υποδηλώνει ένα σύνολο ενεργειών που το σύστημα μπορεί να εφαρμόσει ώστε να περάσει από μία κατάσταση που βρίσκεται το σύστημα σε μία άλλη, και τέλος η σημασία του $\Gamma \subseteq S \times A \times S$ ορίζει τις προϋποθέσεις και τα αποτελέσματα της εκτέλεσης για κάθε ενέργεια. Συγκεκριμένα για τις διαδικτυακές υπηρεσίες, το S_0 και το G είναι η αρχική και τελική κατάσταση που ο χρήστης ζητά από την υπηρεσία. A είναι ένα σύνολο με τις διαθέσιμες υπηρεσίες και το Γ υποδηλώνει την αλλαγή της κατάστασης που παρέχει η κάθε υπηρεσία.

Η OWL-S (γνωστή παλαιότερα και ως DAML-S) είναι η γλώσσα που χρησιμοποιείται κατεξοχήν σε τέτοιου είδους συστήματα, καθώς έχει άμεση σύνδεση με τεχνικές τεχνητής νοημοσύνης. Η αλλαγή μιας κατάστασης παράγεται από την εκτέλεση της υπηρεσίας και έχει οριστεί από τις ιδιότητες των προϋποθέσεων και των επιδράσεων στο Service Profile της DAML-S. Οι προϋποθέσεις αντιπροσωπεύουν λογικές συνθήκες που θα πληρούνται πριν ζητηθεί κάποια υπηρεσία. Οι επιδράσεις είναι το αποτέλεσμα της επιτυχούς εκτέλεσης μιας υπηρεσίας. Από τότε που η DAML+OIL, η γλώσσα κατασκευής της DAML-S, χρησιμοποιεί Description Logics ως βάση της, η DAML+OIL δίνει τη δυνατότητα χρήσης λογικών εκφράσεων.

2.3.1. Λογισμός καταστάσεων (Situation Calculus)

Η McPraith [9, 11, 8] υιοθετεί και επεκτείνει τη γλώσσα Golog για αυτόματη κατασκευή διαδικτυακών υπηρεσιών. Η Golog είναι μια γλώσσα λογικού προγραμματισμού που έχει χτιστεί-βασιστεί στην κορυφή του λογισμού καταστάσεων (situation calculus).

Η γενική ιδέα αυτής της μεθόδου, είναι ότι οι πράκτορες λογισμικού μπορούν να διακρίνουν υπηρεσίες Ιστού, για να παραχθεί αυτόματα η εύρεση, η εκτέλεση, η σύνθεση και η λειτουργικότητα μιας υπηρεσίας. Η αίτηση ενός χρήστη και οι περιορισμοί μπορούν να αναπαρασταθούν με λογική πρώτης τάξης (first-order logic) του λογισμού καταστάσεων. Πολλοί μελετητές θεωρούν τις υπηρεσίες σαν μία ενέργεια, είτε ως απλό ή σύνθετο τελεστή (PrimitiveAction ή ComplexAction). Οι αρχικές ενέργειες (Primitive actions) θεωρούνται είτε ως ενέργειες μεταβολής της κατάστασης του περιβάλλοντος είτε ως ενέργειες που συλλέγουν πληροφορίες και μεταβάλλουν τη γνώση που διαθέτει ένας πράκτορας (agent). Οι σύνθετες ενέργειες (Complex actions) δημιουργούνται από τη σύνθεση ατομικών ενεργειών. Η βάση πληροφοριών ενός πράκτορα διαθέτει μια λογική κωδικοποίηση των προϋποθέσεων και των επιδράσεων της υπηρεσίας στη γλώσσα του λογισμού καταστάσεων (situation calculus). Οι πράκτορες χρησιμοποιούν διαδικαστικές γλώσσες προγραμματισμού με χρήση επαγωγικών μηχανισμών (procedural programming language), δηλαδή καλώντας μεθόδους και διαδικασίες, που αποτελούνται από τις έννοιες και τους περιορισμούς που καθορίζονται από τις υπηρεσίες. Η σύνθεση μιας υπηρεσίας είναι ένα σύνολο από ατομικές υπηρεσίες που συνδέονται με δομές διαδικαστικών γλωσσών προγραμματισμού (if-then-else, while).

Οι μελετητές, επίσης, προτείνουν ένα τρόπο τροποποίησης των Golog προγραμμάτων με την ενσωμάτωση των περιορισμών στις αιτήσεις των χρηστών. Για παράδειγμα, οι χρήστες που αιτούνται μια υπηρεσία μπορούν να επιλέξουν ένα μη-ντετερμινιστικό τρόπο να παρουσιάσουν ποια ενέργεια έχει επιλεγεί δεδομένου μιας κατάστασης ή να χρησιμοποιήσουν μια σειρά κατασκευών για να αναγκάσουν την εκτέλεση μεταξύ δύο ενεργειών. Η δημιουργία του σχεδίου πρέπει να υπακούσει τους αρχικούς περιορισμούς που έχουν τεθεί.

2.3.2. Planning μέσω προτύπου PDDL

Ένα ιδιαίτερο ενδιαφέρον για τη σύνθεση υπηρεσιών Ιστού μέσω ευφών τεχνικών θα μπορούσε να εξηγηθεί με την ομοιότητα μεταξύ των αναπαραστάσεων της DAML-S και της PDDL. Ακόμα, αφού η DAML-S έχει επηρεαστεί από την PDDL, η χαρτογράφηση (mapping) από μία αναπαράσταση σε μία άλλη είναι απλή (αν ληφθούν υπόψη ότι μόνο οι δηλωτικές πληροφορίες). Όταν χρειάζεται ο σχεδιασμός για τη σύνθεση διαδικτυακών υπηρεσιών, οι περιγραφές από την DAML-S μπορούν να μετασχηματιστούν σε PDDL μορφή και στη συνέχεια να χρησιμοποιηθούν διαφορετικοί σχεδιαστές ενεργειών (planners) για τη σύνθεση περαιτέρω υπηρεσιών.

Αναπαριστώντας τη μέθοδο σύνθεσης υπηρεσιών Ιστού βασισμένη στην PDDL, ο McDermott [7] εισάγει ένα νέο τύπο γνώσης, τον οποίο ονομάζει 'τιμή μιας ενέργειας' (value of an action), η οποία εξακολουθεί να υφίσταται και δεν αντιμετωπίζεται ως κυριολεκτική αλήθεια. Από την οπτική της κατασκευής των διαδικτυακών υπηρεσιών, δίνεται η δυνατότητα να διαχωρίσουμε την πληροφορία του μετασχηματισμού και της αλλαγής των καταστάσεων μετά την εκτέλεση της υπηρεσίας. Η πληροφορία που αναπαριστάται ως είσοδος/έξοδος παραμέτρων μπορεί να επαναχρησιμοποιηθεί, συνεπώς οι τιμές των δεδομένων μπορεί να επαναχρησιμοποιηθούν για πολλαπλές υπηρεσίες. Αντίθετα, οι καταστάσεις του συστήματος αλλάζουν με την εκτέλεση της υπηρεσίας. Η αλλαγή ερμηνεύεται με τις παλαιές καταστάσεις να εξαφανίζονται και στη θέση τους παράγονται οι νέες.

Η αντιμετώπιση αυτού του ζητήματος είναι κρίσιμη για τη σύνθεση των υπηρεσιών Ιστού με χρήση ευφών τεχνικών γιατί συνήθως μέσω του σχεδιασμού με τεχνητή νοημοσύνη δημιουργούνται υποθέσεις κλειστού κόσμου (closed world assumption), δηλαδή αν δεν μπορεί να αποδειχθεί στο παρόν σύστημα ότι μια υπόθεση είναι αληθής τότε η πραγματική τιμή της θα είναι *ψευδής*. Στο λογικό προγραμματισμό αυτή η προσέγγιση ονομάζεται ως negation as failure. Το κύριο πρόβλημα των υποθέσεων του κλειστού κόσμου, από την οπτική της κατασκευής των διαδικτυακών υπηρεσιών είναι ότι δεν μπορούμε να εκφράσουμε τις νέες πληροφορίες που έχουν αποκτηθεί.

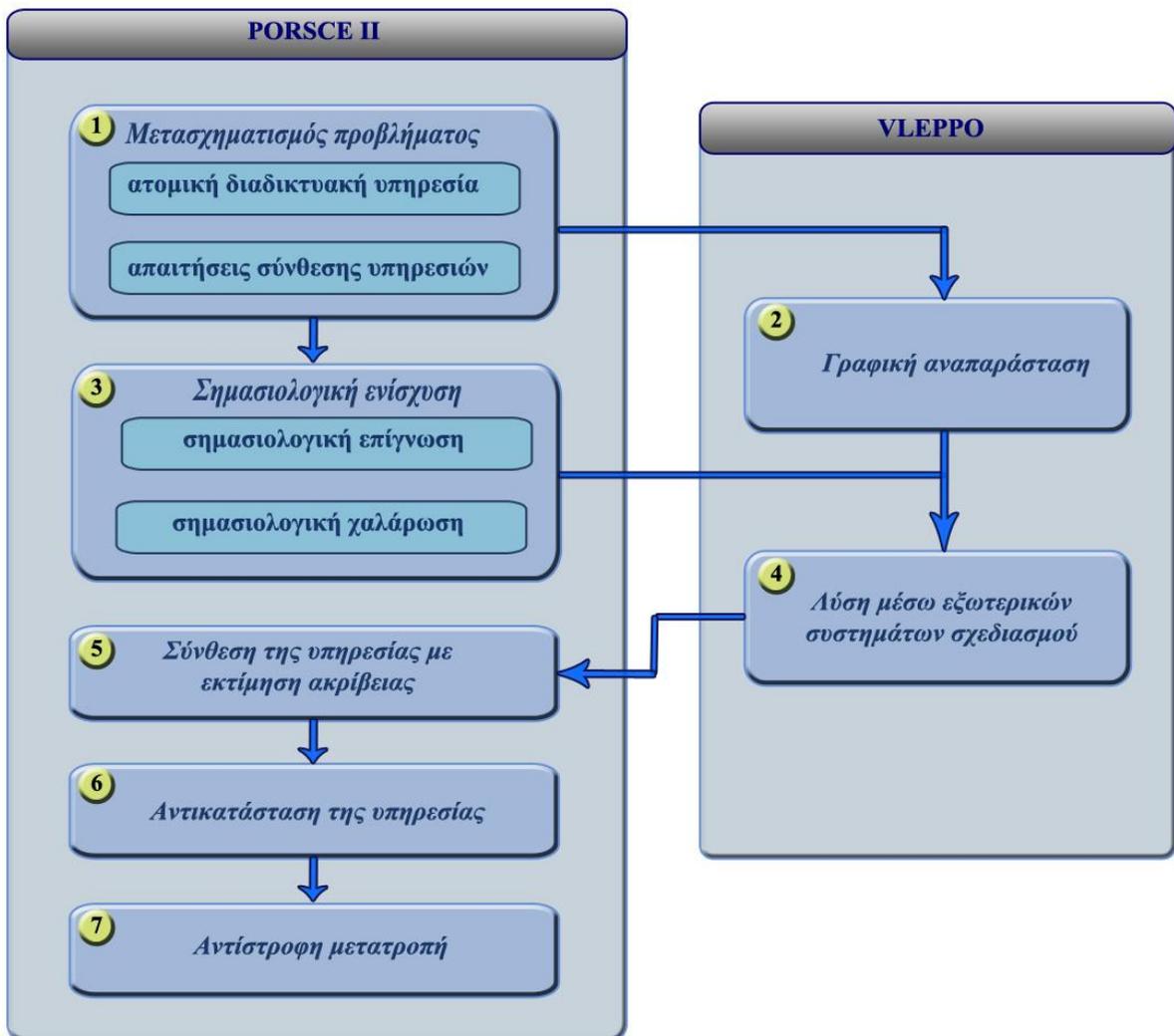
Άλλη μια προσέγγιση που έχει διατυπωθεί [16] για τη σύνθεση διαδικτυακών υπηρεσιών με τη χρήση ευφυών τεχνικών είναι ο μετασχηματισμός του προβλήματος σύνθεσης των υπηρεσιών σε πρόβλημα σχεδιασμού (planning problem). Με την παρούσα υλοποίηση επιτυγχάνονται:

- η φυσική αναπαράσταση των διαδικτυακών υπηρεσιών, με χαρτογράφηση του OWL-S σε PDDL στοιχεία
- η ευελιξία επιλογής εξωτερικών συστημάτων
- η εκτεταμένη αξιοποίηση των σημασιολογικών πληροφοριών
- η συμμόρφωση σε διακεκριμένα πρότυπα
- η πλήρης υποστήριξη σύνθεσης, δεδομένου ότι η διαδικασία αρχικοποιείται από τις περιγραφές των διαδικτυακών υπηρεσιών στο OWL-S και τα αποτελέσματα στην περιγραφή της σύνθετης υπηρεσίας περιγράφονται από το ίδιο πρότυπο, διευκολύνοντας την ανάπτυξη της παραγόμενης σύνθετης υπηρεσίας.

Ο τρόπος με τον οποίο επιτυγχάνεται η συγκεκριμένη προσέγγιση διακρίνεται στα ακόλουθα βήματα, όπως φαίνεται και στην Εικόνα 2.1:

1. Μετασχηματισμός του προβλήματος. Αρχικά χρειάζεται η μετάφραση από πρόβλημα σύνθεσης web service σε πρόβλημα planning. Λαμβάνονται υπόψη όλα τα διαθέσιμα OWL-S web services καθώς και οι απαιτήσεις του χρήστη. Παράγεται ένας κόσμος με όλες τις διαθέσιμες υπηρεσίες (domain) και μια περιγραφή του προβλήματος (problem).
2. Γραφική αναπαράσταση (προαιρετικό). Μέσω των αρχείων που έχει αναπαρασταθεί ο κόσμος (domain και problem), δίνεται η δυνατότητα γραφικής αναπαράστασης.
3. Σημασιολογική ενίσχυση. Σε αυτό το βήμα ενισχύονται σημασιολογικά οι διαθέσιμες υπηρεσίες και το πρόβλημα, που έχουν παραχθεί όπως αναφέρθηκε παραπάνω. Θεωρείται σημαντικό, αφού σε πολλές περιπτώσεις οι συντακτικές διαφορές μεταξύ των δεδομένων εισόδου και εξόδου, μπορούν να κάνουν το σύστημα να μην ολοκληρώσει τη διαδικασία

αντιστοίχισης επιτυχώς, ακόμα κι όταν εννοιολογικά κάποια υπηρεσία είναι παρόμοια με κάποια άλλη.



Εικόνα 2.1 Βήματα ολοκλήρωσης συστήματος PORSCCE II και VLEPPO

4. Λύση μέσω εξωτερικών συστημάτων σχεδιασμού (planners). Οι διαθέσιμες υπηρεσίες και το πρόβλημα, που πιθανώς έχουν ενισχυθεί και σημασιολογικά, εξάγονται ως PDDL, για να μπορούν να επιλυθούν από εξωτερικά συμβατά και ανεξάρτητα συστήματα σχεδιασμού (planners), χρησιμοποιώντας κλασσικές τεχνικές σχεδιασμού (classical planning techniques).

5. Σύνθεση της υπηρεσίας με εκτίμηση ακρίβειας. Το σχέδιο (plan) που παράχθηκε μέσω του συστήματος αποτελεί την επιθυμητή σύνθετη υπηρεσία. Η σημασιολογική ενίσχυση καθώς και η χρήση πολλαπλών εξωτερικών συστημάτων σχεδιασμού, μπορεί να παράγει περισσότερες από μία συνθέσεις υπηρεσιών.
6. Αντικατάσταση της υπηρεσίας (προαιρετικό). Διαχειρίζεται περιπτώσεις αποτυχίας ή μη διαθεσιμότητας κάποιας υπηρεσίας αντικαθιστώντας την με κάποια άλλη ατομική υπηρεσία ώστε να γίνει μέρος της σύνθετης υπηρεσίας.
7. Αντίστροφη μετατροπή. Τα παραγόμενα αποτελέσματα μεταφράζονται σε OWL-S.

Από τα παραπάνω βήματα όσα συνδέονται με υπηρεσίες έχουν υλοποιηθεί στο PORSCE II σύστημα, όπως ο μετασχηματισμός προβλήματος, η σύνθεση της υπηρεσία με εκτίμηση ακρίβειας, η αντικατάσταση της υπηρεσίας και η αντίστροφη μετατροπή [17]. Αντίστοιχα, όσα συνδέονται με σχεδιασμό (planning) έχουν υλοποιηθεί στο VLEPPO σύστημα [18, 19].

2.3.3. Rule-based Planning (Σχεδιασμός βασισμένος στους κανόνες)

Ο Medjahed [10] παρουσίασε μια τεχνική για τη δημιουργία σύνθεσης υπηρεσιών από υψηλού επιπέδου δηλωτική περιγραφή. Η μέθοδος χρησιμοποιεί κανόνες σύνθεσης, για να προσδιοριστεί κατά πόσο μπορούν να συντεθούν δύο υπηρεσίες. Η παρούσα προσέγγιση σύνθεσης αποτελείται από τέσσερις φάσεις. Αρχικά, η προσδιοριστική φάση που ενεργοποιεί την υψηλού επιπέδου περιγραφή των υπηρεσιών που θα συντελέσουν στη σύνθεση, χρησιμοποιώντας μια γλώσσα προγραμματισμού που ονομάζεται Composite Service Specification Language (CSSL). Δεύτερον, η φάση αντιστοιχίας χρησιμοποιεί κανόνες σύνθεσης για να παράγει το πλάνο που ταιριάζει στις απαιτήσεις του χρήστη. Τρίτον, η φάση επιλογής: αν δημιουργηθούν περισσότερα από ένα πλάνα στη φάση επιλογής, ο αιτών επιλέγει ένα από τα πλάνα σύμφωνα με την ποιότητα της σύνθεσης (QoC) των παραμέτρων, όπως το κόστος, η θέση του πλάνου κτλ. Η τελική φάση είναι η φάση παραγωγής, όπου μια λεπτομερής περιγραφή της σύνθεσης των υπηρεσιών παράγεται αυτόματα και παρέχεται στον τελικό χρήστη.

Σε αυτό το σημείο χρειάζεται να δοθεί περισσότερο προσοχή στους κανόνες σύνθεσης γιατί αποτελούν το κυριότερο ζήτημα που χρειάζεται να προσδιοριστεί για τον τρόπο με τον οποίο το πλάνο σύνθεσης δημιουργείται. Οι κανόνες σύνθεσης λαμβάνουν υπόψη τόσο τις συντακτικές όσο και τις σημασιολογικές ιδιότητες των διαδικτυακών υπηρεσιών. Οι συντακτικοί κανόνες περιλαμβάνουν τους κανόνες των τρόπων λειτουργίας και τους κανόνες για τα δεσμευτικά πρωτόκολλα των υπηρεσιών που θα αλληλεπιδράσουν. Οι σημασιολογικοί κανόνες περιλαμβάνουν τα ακόλουθα: (1) ένα μήνυμα σύνθεσης που προσδιορίζει ότι υπάρχει δυνατότητα σύνθεσης δύο διαδικτυακών υπηρεσιών μόνο υπό την προϋπόθεση ότι το μήνυμα εξόδου της μίας υπηρεσίας είναι συμβατό με το μήνυμα εισόδου της άλλης. (2) η σημασιολογική λειτουργία της σύνθεσης προσδιορίζει τη συμβατότητα μεταξύ των πεδίων ορισμού, των κατηγοριών και των σκοπών των δύο υπηρεσιών. (3) η ποιότητα σύνθεσης προσδιορίζει τις προτιμήσεις του αιτούντος όσον αφορά στην ποιότητα των εργασιών της σύνθεσης της υπηρεσίας και (4) η ορθότητα σύνθεσης λαμβάνει υπόψη αν η σύνθεση των υπηρεσιών είναι λογική. Για το σκοπό αυτό, εισάγεται η έννοια της σύνθεσης προτύπων που προσδιορίζουν την ανεξαρτησία μεταξύ διαφορετικών τύπων υπηρεσιών.

Η κύρια συνεισφορά αυτής της μεθόδου είναι οι κανόνες σύνθεσης, επειδή καθορίζουν τα πιθανά γνωρίσματα των διαδικτυακών υπηρεσιών που θα μπορούσαν να χρησιμοποιηθούν στη σύνθεση των υπηρεσιών. Αυτοί οι κανόνες θα μπορούσαν να χρησιμοποιηθούν σαν οδηγοί για άλλες μεθόδους υπηρεσιών ιστού βασισμένοι στο σχεδιασμό (planning).

Το SWORD [12] (Simple Web-Service Offering Repository Deposit) είναι ένα άλλο προγραμματιστικό εργαλείο ανάπτυξης σύνθετων υπηρεσιών Ιστού βασισμένο στη δημιουργία πλάνων με κανόνες. Το SWORD δεν αναπτύσσει τα πρότυπα των αναδυόμενων περιγραφικών υπηρεσιών όπως η WSDL και η DAML-S, αντιθέτως, χρησιμοποιεί το μοντέλο Οντοτήτων-Συσχετίσεων (ΟΣ) για να καθορίσει τις διαδικτυακές υπηρεσίες. Στο SWORD, μία υπηρεσία μοντελοποιείται από τις προϋποθέσεις και τα αποτελέσματα, τα οποία προσδιορίζονται στο μοντέλο του συστήματος από τις οντότητες και τις σχέσεις μεταξύ των οντοτήτων. Μία υπηρεσία ιστού αναπαρίσταται σε μορφή Horn που υποδηλώνει ότι τα αποτελέσματα μπορούν να πραγματοποιηθούν αν οι προϋποθέσεις είναι *αληθείς*. Για τη δημιουργία μιας σύνθετης υπηρεσίας, ο χρήστης χρειάζεται μόνο να προσδιορίσει την αρχική και τελική κατάσταση και τότε η παραγωγή του σχεδίου μπορεί να επιτευχθεί μέσω ενός προηγμένου συστήματος βασισμένο σε κανόνες (rule-based). Ως ένα μειονέκτημα του SWORD αναφέρεται το ότι

αρκετές φορές μπορεί να δημιουργηθεί ένα *αβέβαιο* αποτελέσματα αν η προϋπόθεση δεν ορίζει μοναδικά ένα αποτέλεσμα. Ωστόσο, διατυπώνεται η άποψη ότι αβέβαια αποτελέσματα μπορούν να αποφευχθούν μόνο όταν οι προϋποθέσεις εξαρτώνται λειτουργικά από τις προϋποθέσεις που ορίζονται στην ίδια υπηρεσία.

2.3.4. Case-based Planning (Σχεδιασμός βασισμένος στις περιπτώσεις)

Η παρούσα μέθοδος προσέγγισης [20] σύνθεσης υπηρεσιών Ιστού δημιουργεί ένα πλάνο για τη νέα αίτηση του χρήστη υιοθετώντας αυτόματα κάποιο, όσο πιο παρόμοιο πλάνο υπάρχει από μία παλαιότερη αίτηση. Το νέο πλάνο που έχει δημιουργηθεί μαζί με την νέα αίτηση μπορούν να αποθηκευτούν ως μία νέα περίπτωση για μελλοντική επαναχρησιμοποίηση αυτού του σεναρίου.

Στον κλασσικό σχεδιασμό με βάση τις περιπτώσεις (CBP), οι προηγούμενες λαμβανόμενες λύσεις και οι περιγραφές των προβλημάτων αποθηκεύονται ως περιπτώσεις. Δεδομένου του νέου προβλήματος, ανακτάται μία παρεμφερής περίπτωση και η λύση της υιοθετείται για να λυθεί το νέο πρόβλημα. Για τη σύνθεση υπηρεσιών Ιστού με βάση το σχεδιασμό στις περιπτώσεις, η κλασσική μέθοδος χρειάζεται να τροποποιηθεί στις ακόλουθες προοπτικές. Αναφορικά:

1. Στην αναπαράσταση περιπτώσεων
2. Στην ανάκτηση περιπτώσεων
 - 2.1. Υπολογισμός ομοιότητας
 - 2.2. Ρύθμιση σχεδίου
3. Στο σχέδιο προσαρμογής

Άλλες δύο προσεγγίσεις έχουν προταθεί με χρήση συλλογιστικής βασισμένη στις περιπτώσεις ή case-based reasoning (CBR), για τη λύση της αυτόματης σύνθεσης υπηρεσιών Ιστού. Στην προσέγγιση Web service composition with case-based reasoning [22], οι συγγραφείς παρουσίασαν ένα πλαίσιο για τη σύνθεση υπηρεσιών Ιστού χρησιμοποιώντας την τεχνική

συλλογιστικής βασισμένη στις περιπτώσεις για την ανακάλυψη των διεργασιών. Κάθε περίπτωση είναι μία σύνθετη υπηρεσία και έχει όνομα και περιγραφή. Ωστόσο, ο αλγόριθμος για την ανάκτηση της περίπτωσης είναι βασισμένος μόνο στο όνομα και αυτή η προσέγγιση δε χρησιμοποιεί τις σημασιολογικές περιγραφές, για να καταστεί δυνατή η αποτελεσματική αιτιολογία. Στην έρευνα [21], παρουσιάστηκε μία προσέγγιση που καλείται WeSCo_CBR για σύνθεση υπηρεσιών Ιστού με ημιαυτόματο τρόπο βασισμένο στην συλλογιστική τεχνική σύμφωνα με τις περιπτώσεις και στη σημασιολογική περιγραφή των υπηρεσιών. Ορίζεται ένα υψηλού επιπέδου αφαιρετικό επίπεδο, που χαρακτηρίζεται από ένα σύνολο δραστηριοτήτων, και οι οντολογίες έχουν αναπτυχθεί για να προσδιορίσουν τις σχέσεις μεταξύ των δραστηριοτήτων και των εννοιών. Αυτή η προσέγγιση, όμως μπορεί να επιστρέψει μόνο το πλάνο των πιο παρόμοιων περιπτώσεων και να αφήσει το πλάνο προσαρμογής στο χρήστη. Ο αλγόριθμος που υπολογίζει την ομοιότητα λαμβάνει υπόψη του μόνο τις υπογραφές εισόδου και εξόδου (I/O) των δραστηριοτήτων χωρίς να εξετάσει τις προϋποθέσεις και τις επιπτώσεις.

2.3.5. Άλλες τεχνικές

Προτείνονται και κάποιες άλλες ευφυείς τεχνικές για την αυτόματη σύνθεση διαδικτυακών υπηρεσιών. Το [14] σύστημα σχεδιασμού SHOP2 εφαρμόζεται για την αυτόματη σύνθεση υπηρεσιών ιστού, που παρέχονται από τις περιγραφές της DAML-S. Το σύστημα SHOP2 χρησιμοποιεί ένα ιεραρχικό δίκτυο εργασιών (Hierarchical Task Network - HTN). Οι συγγραφείς πιστεύουν ότι η εργασία αποσύνθεσης στο HTN σύστημα είναι παρόμοια με τη διαδικασία αποσύνθεσης στην οντολογία της DAML-S. Ισχυρίζονται, ακόμα, ότι ο σχεδιασμός με ένα HTN σύστημα είναι πιο αποδοτικός από άλλες γλώσσες προγραμματισμού, όπως η Golog. Στις δημοσιεύσεις τους, δίνουν αναλυτικές περιγραφές όσον αφορά στη διαδικασία μεταγλώττισης από την DAML-S σε SHOP-2.

Ο Sirin [13] παρουσίασε μία ημιαυτόματη μέθοδο για σύνθεση διαδικτυακών υπηρεσιών. Κάθε φορά που κάποιος χρήστης χρειάζεται να επιλέξει μία υπηρεσία, όλες οι πιθανές υπηρεσίες που ταιριάζουν με την επιλεγμένη, παρουσιάζονται σε αυτόν. Η επιλογή των πιθανών υπηρεσιών βασίζεται στις λειτουργίες και στα μη λειτουργικά χαρακτηριστικά. Οι λειτουργίες (παράμετροι) παρουσιάζονται από τις κλάσεις της OWL και προγράμματα συλλογιστικής της OWL (reasoners) εφαρμόζονται για να γίνει η αντιστοιχία μεταξύ των υπηρεσιών. Μια αντιστοιχία

μεταξύ δύο υπηρεσιών ορίζεται ως η παράμετρος εξόδου της μιας υπηρεσίας είναι η ίδια OWL κλάση ή υποκλάση της παραμέτρου εισόδου μιας άλλης υπηρεσίας. Η μηχανή συμπερασμάτων της OWL μπορεί να διατάξει τις υπηρεσίες που έχουν αντιστοιχιστεί, έτσι ώστε η προτεραιότητα να είναι χαμηλότερη όταν η απόσταση μεταξύ των δύο τύπων του οντολογικού δέντρου αυξάνεται. Αν βρεθούν παραπάνω από μία υπηρεσίες που αντιστοιχούν στα κριτήρια του χρήστη, το σύστημα φιλτράρει τις υπηρεσίες με βάση τα μη λειτουργικά χαρακτηριστικά που καθορίζονται από τις απαιτήσεις του χρήστη. Μόνο οι υπηρεσίες που πληρούν τις απαιτήσεις του χρήστη παρουσιάζονται σε αυτόν. Η ιδέα αυτή της σύνθεσης υπηρεσιών με ημιαυτόματο τρόπο είναι ενδιαφέρουσα γιατί θεωρείται δύσκολο να αποτυπωθεί η συμπεριφορά με επαρκείς λεπτομέρειες και να γίνει η σύνθεση των υπηρεσιών με πλήρως αυτόματο τρόπο, ειδικά για τις εμπορικές. Παρά όλα αυτά η προτεινόμενη μέθοδος είναι απλή, καθώς υποδεικνύει τον τρόπο με τον οποίο το σύστημα σχεδιασμού (planner) και ο άνθρωπος μπορούν να δουλέψουν μαζί, για να παραχθεί η σύνθεση των υπηρεσιών που ο χρήστης έχει αιτηθεί.

2.4. Συνεισφορά της παρούσας εργασίας

Στην παρούσα εργασία επιχειρήθηκε να υλοποιηθεί η αυτοματοποιημένη σύνθεση και πρόταση διαδικτυακών εφαρμογών (gadgets) για την απόδοση πολύπλοκης λειτουργικότητας σε κοινωνικά δίκτυα. Μέσα στις ποικίλες εφαρμογές που υπάρχουν σε ένα οποιοδήποτε κοινωνικό δίκτυο, οι χρήστες αναζητούν τρόπους με τους οποίους θα μπορέσουν να βρουν αυτές που ικανοποιούν τις ανάγκες τους. Πολλές φορές για να ικανοποιήσουν την αναζήτησή τους χρειάζονται περισσότερες από μία εφαρμογές, προκειμένου να καταλήξουν στην τελική ανάγκη. Η ανάγκη αυτή των χρηστών να εκτελέσουν σύνθετη λειτουργικότητα μπορεί είτε να απαιτεί στοιχεία που ο ίδιος ο χρήστης χρειάζεται να εισάγει ή να προκύψουν από την εκτέλεση άλλων εφαρμογών. Αυτές οι εφαρμογές μπορεί με τη σειρά τους να απαιτούν δεδομένα που παράγονται από την εκτέλεση άλλων υπηρεσιών κ.ο.κ. Στην περίπτωση που ο χρήστης δεν έχει συγκεντρώσει όλα τα απαραίτητα δεδομένα δεν μπορεί να φτάσει στον τελικό του στόχο. Ακόμα, πιο δύσκολη γίνεται η αναζήτησή του χρήστη καθώς ο αριθμός των διαθέσιμων υπηρεσιών αυξάνεται. Προκύπτει, λοιπόν, μία ροή διαδικασιών που ο χρήστης δεν είναι σε θέση να γνωρίζει εκ των προτέρων τα στοιχεία που θα του ζητηθούν, ώστε να προκύψει το τελικό αποτέλεσμα που επιθυμεί. Τέτοιες περιπτώσεις συναντώνται στα κοινωνικά δίκτυα καθώς και σε

δυναμικά περιβάλλοντα, όπου διατίθεται ένας μεγάλος αριθμός υπηρεσιών προς εκτέλεση και οι διαθέσιμες υπηρεσίες αλλάζουν ακόμα και κατά τη διάρκεια της αναζήτησης του χρήστη.

Σύμφωνα με τη διεθνή βιβλιογραφία, η αυτοματοποιημένη σύνθεση διαδικτυακών υπηρεσιών (web service) έχει αποδειχθεί χρήσιμη. Στην παρούσα εργασία προτείνεται και υλοποιείται ένας μηχανισμός πρότασης και σύνθεσης gadget στο περιβάλλον ενός κοινωνικού δικτύου, σύμφωνα με τις ιδέες για αυτοματοποιημένη σύνθεση διαδικτυακών υπηρεσιών που έχουν προταθεί παραπάνω. Δύο από τις υπηρεσίες που αναφέρθηκαν παραπάνω είναι τα web services και τα gadgets. Ο ορισμός των δύο τεχνολογιών, είναι παρόμοιος καθώς και τα δύο αναπαρίστανται σχεδόν με τον ίδιο τρόπο. Συγκεκριμένα, και στις δύο περιπτώσεις η κάθε υπηρεσία περιγράφει τη λειτουργία της, τα δεδομένα εισόδου που δέχεται και τα δεδομένα εξόδου που παράγει. Το σύστημα γνωρίζοντας εκ των προτέρων τις διαθέσιμες υπηρεσίες αλλά και τα αποτελέσματα που παράγει η κάθε μία μπορεί να παρουσιάσει στο χρήστη όλες τις απαιτούμενες εφαρμογές που χρειάζεται να γνωρίζει. Ως αποτέλεσμα, ο παρόν μηχανισμός θα επιτρέψει την παρουσίαση όλων των υπηρεσιών που χρειάζεται ο χρήστης, για να εκτελέσει τη λειτουργικότητα που επιθυμούσε εξ αρχής. Ο χρήστης χρειάζεται απλά να δηλώσει στο μηχανισμό τι στοιχεία χρειάζεται να έχει στο προφίλ του από την εκτέλεση μιας διεργασίας και ο μηχανισμός θα φροντίσει να του παρουσιάσει όλες τις υπηρεσίες που θα τον κάνουν να το πετύχει. Με αυτόν τον τρόπο, οι χρήστες απαλλάσσονται από την ανάγκη να γνωρίζουν ποια gadgets χρειάζονται καθώς και το τι λειτουργία εκτελεί το κάθε ένα από αυτά.

3. Περιγραφή, σχεδιασμός και υλοποίηση μηχανισμού σύνθεσης

Στο κεφάλαιο αυτό περιγράφονται οι αρχικές απαιτήσεις για την ολοκλήρωση του μηχανισμού πρότασης και σύνθεσης gadget, καθώς και η υλοποίησή του. Αναλύονται οι τεχνολογίες, η αρχιτεκτονική του συστήματος καθώς και τρόπος με τον οποίο επιτεύχθηκε η δημιουργία του περιβάλλοντος της εφαρμογής.

3.1. Περιγραφή Μηχανισμού Σύνθεσης

3.1.1. Λειτουργικότητα μηχανισμού σύνθεσης

Ο στόχος του συγκεκριμένου μηχανισμού είναι να παράγει λύση στην ανάγκη του χρήστη να ικανοποιήσει την λειτουργικότητα που επιθυμεί. Η λύση αυτή δίνεται από το μηχανισμό σύνθεσης και πρότασης gadget μέσω ευφών τεχνικών.

Ο κάθε συμμετέχοντας μέσα στο κοινωνικό δίκτυο μπορεί να λάβει υπηρεσίες, εγκαθιστώντας στο profile του και εκτελώντας συγκεκριμένα gadgets, τα οποία σε πολλές περιπτώσεις επικοινωνούν και αλληλεπιδρούν με εξωτερικά συστήματα. Τα gadgets αυτά, καθώς εκτελούνται, παράγουν και εγκαθιστούν στο προφίλ του κάποια δεδομένα.

Προκειμένου να μπορεί να ικανοποιηθεί η απαιτούμενη ανάγκη ενός συμμετέχοντα για την παροχή συγκεκριμένων υπηρεσιών, μέσω της εκτέλεσης κάποιου gadget, το προφίλ του χρειάζεται να είναι ενημερωμένο με όλα τα απαραίτητα στοιχεία που χρειάζεται η εκάστοτε εφαρμογή για να εκτελεστεί. Ο χρήστης, όμως, δεν μπορεί να γνωρίζει εκ των προτέρων που θα βρει αυτά τα στοιχεία, και από ποιες εφαρμογές είναι δυνατόν να παραχθούν.

Σε αυτή την περίπτωση, είναι αναγκαίος ένας μηχανισμός ο οποίος, έχοντας γνώση των gadgets που υπάρχουν διαθέσιμα στο κοινωνικό δίκτυο, μπορεί να συμβουλευσει τον χρήστη, προτείνοντας τα κατάλληλα gadgets. Στις περιπτώσεις μάλιστα που η λειτουργικότητα δεν είναι απλή, αλλά πιο σύνθετη, και ικανοποιείται από ένα σύνολο από gadgets, ο χρήστης πρέπει να

χρησιμοποιήσει τον εν λόγω μηχανισμό για να του προτείνει μια σύνθεση gadgets που θα παρέχει την απαιτούμενη λειτουργικότητα.

Για να είναι σε θέση να λειτουργήσει ο μηχανισμός και να μπορέσει ο χρήστης να ικανοποιήσει την ανάγκη του, χρειάζεται να δηλωθεί στο μηχανισμό η επιθυμητή λειτουργικότητα. Αυτό επιτυγχάνεται, όταν ο χρήστης επιλέξει μέσω της αναζήτησης την τελική απαίτηση που θέλει να εκτελέσει. Ο μηχανισμός είναι σε θέση να παράγει τη ροή των απαιτούμενων εφαρμογών προκειμένου να επιτευχθεί ο τελικός στόχος, όταν ο χρήστης εκφράσει στο μηχανισμό την λειτουργικότητα που χρειάζεται, με την μορφή απαιτούμενων δεδομένων που πρέπει να τρέχουν στο profile του. Πρακτικά, ο μηχανισμός είναι καθοδηγούμενος από τα δεδομένα που θα λάβει από το χρήστη (data / context – driven), και έτσι τα αποτελέσματα διαφοροποιούνται σύμφωνα με τα δεδομένα που θα του δοθούν.

Τα πλάνα σύνθεσης gadget που παράγονται από τον μηχανισμό μπορούν να είναι τμηματικά (partial). Αυτό σημαίνει ότι ο μηχανισμός, με την χρήση του σχεδιασμού ενεργειών, ξεκινάει την εκτέλεσή του από τα δεδομένα τα οποία ο χρήστης θέλει να έχει στο profile του και προχωράει «προς τα πίσω» μέχρι να βρει gadgets που παράγουν αυτά τα δεδομένα, και στη συνέχεια συνεχίζει με περαιτέρω βήματα «προς τα πίσω», σε πολλά επίπεδα, μέχρι να βρει τα απαραίτητα δεδομένα στο profile του χρήστη. Σε περίπτωση που κάτι τέτοιο δεν καθίσταται δυνατό, ο μηχανισμός ενημερώνει τον χρήστη για τα απαραίτητα δεδομένα που πρέπει να υπάρξουν στο profile του αλλά δεν μπορούν να παραχθούν αυτή τη στιγμή μέσα στο κοινωνικό δίκτυο από τα διαθέσιμα gadgets. Αυτό πρακτικά σημαίνει ότι ο χρήστης πρέπει είτε να τα παρέχει ως είσοδο, είτε πρέπει να φροντίσει να γίνουν διαθέσιμα στο κοινωνικό δίκτυο gadgets που παράγουν αυτά τα δεδομένα. Μέσω του μηχανισμού δίνεται η δυνατότητα παραγωγής και προβολής όλων των εναλλακτικών πλάνων που βρίσκονται στην πορεία εκτέλεσης, δηλαδή στην περίπτωση που το σύστημα σχεδιασμού έχει βρει περισσότερες από μία λύσεις ο μηχανισμός είναι σε θέση να τις εμφανίσει στο χρήστη.

3.1.2. Αναπαράσταση και Μετασχηματισμός Γνώσης

Σκοπός του συστήματος είναι να εξυπηρετήσει και να διευκολύνει του χρήστες στην αναζήτηση όλων των προαπαιτούμενων εφαρμογών, ώστε να ικανοποιηθεί η ανάγκη τους. Το σύστημα

χρειάζεται να είναι εύκολο στη χρήση και λειτουργικό ως προς τις απαιτήσεις του εκάστοτε χρήστη. Για το σκοπό αυτό χρειάζεται να δημιουργηθεί ένα γραφικό περιβάλλον μέσω του οποίου ο χρήστης θα μπορεί να αναζητά και να ανακαλύπτει δυναμικά όλες τις υπηρεσίες που συντίθενται ώστε να παραχθεί η τελική ανάγκη. Το γραφικό περιβάλλον αυτό μπορεί να ενσωματωθεί σε όποιο σύστημα περιηγείται ο χρήστης, είτε πρόκειται για κοινωνικό δίκτυο ή για άλλη δυναμική εφαρμογή.

Η αναζήτηση των εφαρμογών ως γραφική αναπαράσταση δε διαφέρει από τους κλασσικούς μηχανισμούς αναζήτησης. Ο χρήστης έχει μια ανάγκη που θέλει να ικανοποιήσει και απευθύνεται σε μία μηχανή αναζήτησης ώστε να του προτείνει αποτελέσματα που συμφωνούν με την ανάγκη του. Η διαφορά στο συγκεκριμένο μηχανισμό παρατηρείται στον τρόπο με τον οποίο παρατίθενται τα αποτελέσματα της αναζήτησης. Ο λόγος αυτής της διαφοροποίησης είναι γιατί κάθε φορά δημιουργούνται δυναμικά διαφορετικά αποτελέσματα, ανάλογα την αναζήτηση. Για αυτό το λόγο χρειάζεται ορθή απεικόνιση των διαφορετικών οντοτήτων ώστε να είναι κατανοητή από το χρήστη. Οι διαφορετικές οντότητες στο σύστημα είναι:

1. Τα gadgets
2. Τα δεδομένα εισόδου που απαρτίζονται από:
 - 2.1. κύρια πληροφορία
 - 2.2. πληροφορία από την έξοδο κάποιας άλλης υπηρεσίας
3. Τα δεδομένα εξόδου

Συγκεκριμένα, τα gadgets που είναι απαραίτητα να εκτελεστούν, χρειάζεται να αναπαρίστανται με διαφορετικό τρόπο από τα δεδομένα εισόδου και εξόδου. Ακόμα, χρειάζεται διαφορετική αναπαράσταση των δεδομένων ανάλογα με τον τύπο τους, δηλαδή αν είναι δεδομένα εισόδου ή εξόδου. Τα δεδομένα εισόδου διακρίνονται σε δύο βασικές κατηγορίες, αυτά που απαιτούν κύρια ή πρωτεύουσα πληροφορία ή αυτά που παράγονται με την εκτέλεση κάποιας άλλης υπηρεσίας. Με τον όρο κύρια ή πρωτεύουσα πληροφορία εννοείται όποια πληροφορία απαιτείται από το χρήστη και δεν μπορεί να παραχθεί από κάποιο gadget. Ο παραπάνω

μηχανισμός αλλά και ο τρόπος με τον οποίο αναπαρίστανται τα δεδομένα ενσωματώνεται στο κοινωνικό δίκτυο Unity, έχοντας πρόσβαση στις πληροφορίες των εφαρμογών και σε ό,τι αυτές προϋποθέτουν και παράγουν.

Για να μπορέσει να δοθεί στο χρήστη η γραφική απεικόνιση που περιγράφηκε παραπάνω, δηλαδή η λύση στο πρόβλημα που αναζητά, χρειάζεται να γίνουν κάποιοι απαραίτητοι μετασχηματισμοί της πληροφορίας. Χρειάζεται, αρχικά, ορισμένες ενέργειες του χρήστη να αποκωδικοποιηθούν και να μετατραπούν σε μετασχηματισμούς που δέχεται το σύστημα σχεδιασμού (planner). Στη συνέχεια να ακολουθηθεί η αντίστροφη πορεία, δηλαδή το αποτέλεσμα που έχει βγάλει ο planner ως λύση να μετατραπεί σε φιλική προς το χρήστη μορφή. Συγκεκριμένα χρειάζονται να γίνουν τα παρακάτω:

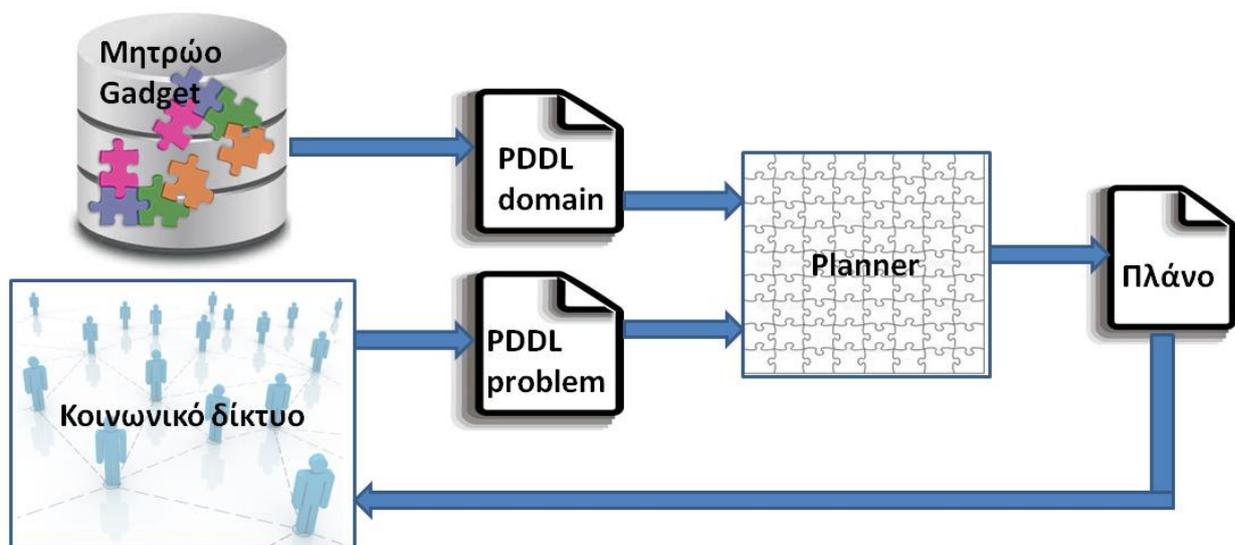
- Τα διαθέσιμα gadgets και δεδομένα εισόδου και εξόδου να μετασχηματιστούν στο αρχείο `domain.pddl`.
- Η επιθυμητή έξοδος που ζήτησε ο χρήστης να μετατραπεί στο αρχείο `problem.pddl`.
- Το αρχείο `solution.sol` που εξήγαγε η εκτέλεση του planner να μετασχηματιστεί στη γραφική απεικόνιση που δοθεί στο χρήστη.

Αναλυτικότερα, όπως φαίνεται στην Εικόνα 3.1, ο μηχανισμός σύνθεσης και πρότασης gadget δημιουργεί το αρχείο `domain.pddl`, σύμφωνα με τις διαθέσιμες εφαρμογές (gadgets) και τα διαθέσιμα δεδομένα εισόδου και εξόδου (concepts) που υπάρχουν στη βάση δεδομένων (registry). Όλα τα concepts μετασχηματίζονται σε preconditions και όλα τα gadgets σε actions. Ένα action περιγράφεται από το όνομά του, από τα δεδομένα εισόδου που δέχεται και από τα δεδομένα εξόδου που παράγει. Στο σημείο που περιγράφεται ένα action τα δεδομένα εισόδου ορίζονται ξανά ως preconditions, αλλά αυτή τη φορά για το συγκεκριμένο gadget, και τα δεδομένα εξόδου ως effect. Βασική προϋπόθεση είναι τα concepts που περιγράφουν ένα gadget να έχουν οριστεί στο αρχικό κομμάτι των preconditions.

Το δεύτερο τμήμα του μετασχηματισμού γίνεται όταν ο χρήστης εισέλθει στο κοινωνικό δίκτυο και στο μηχανισμό σύνθεσης. Αφού επιλέξει την τελική έξοδο που επιθυμεί να έχει στο profile του δημιουργείται το αρχείο `problem.pddl`. Το σύστημα μεταφράζει την έξοδο που επέλεξε ο

χρήστης σε ένα precondition από αυτά που έχουν ήδη οριστεί στο αρχείο domain.pddl, αλλά με την ετικέτα goal υποδηλώνοντας ότι αυτή είναι η τελική λειτουργικότητα που ζητείται.

Σύμφωνα με τους παραπάνω μετασχηματισμούς ο planner παράγει ένα αρχείο, το solution.SOL, το οποίο περιέχει το πλάνο λύσης που βρέθηκε. Το αρχείο αυτό περιέχει εγγραφές τα actions, δηλαδή τα gadgets, που οδηγούν στη λύση που ζητήθηκε. Ο μηχανισμός αποκωδικοποιεί τα actions βρίσκοντας σε ποια gadgets αναφέρεται ο planner και στη συνέχεια τα αναπαριστά στο χρήστη.



Εικόνα 3.1 Μετασχηματισμοί προβλήματος

3.1.3. Διασύνδεση με κοινωνικά δίκτυα

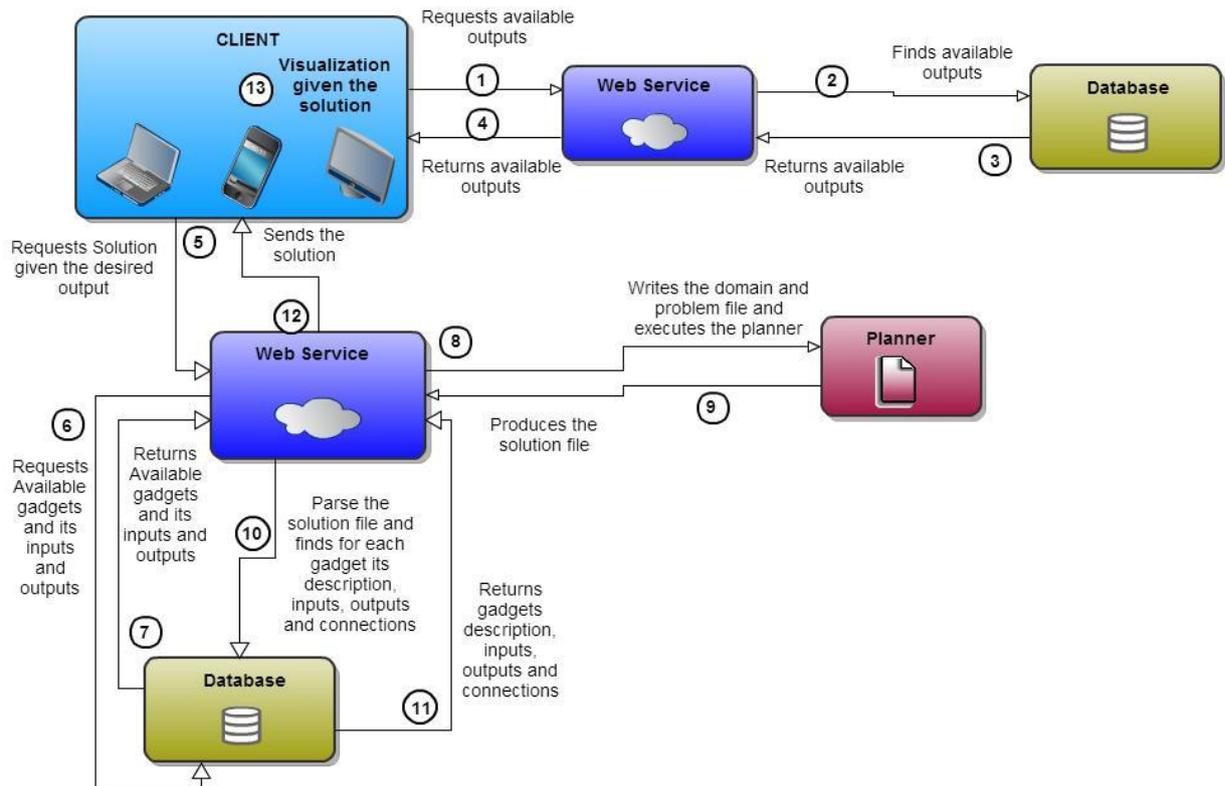
Ο μηχανισμός διατίθεται, όπως έχει ήδη αναφερθεί, για κοινωνικά δίκτυα ή οποιοδήποτε άλλο περιβάλλον συμφωνεί με τα τρέχοντα πρότυπα κοινωνικής δικτύωσης OpenSocial και Apache Shindig, ώστε να είναι εύκολο να ολοκληρωθεί σε αυτά. Συγκεκριμένα, ο μηχανισμός χρειάζεται ένα τρόπο ώστε να μπορεί να έχει πρόσβαση στις περιγραφές των εφαρμογών καθώς και στα δεδομένα εισόδου και εξόδου τις κάθε μίας. Ο μηχανισμός χρειάζεται να έχει άποψη της βάσης

δεδομένων, η οποία πρέπει να έχει δεχθεί τέτοιες επεκτάσεις ώστε να συμφωνεί με την δομή, όπως περιγράφηκε παραπάνω. Με αυτό τον τρόπο, δημιουργείται ένα σύνολο από τις διαθέσιμες εφαρμογές (gadgets) καθώς και ένα σύνολο από διαθέσιμα δεδομένα (concepts), ανεξάρτητα μεταξύ τους. Όταν τα παραπάνω συνδυαστούν προκύπτει η ολοκληρωμένη εικόνα που χρειάζεται να έχει ένα gadget. Συγκεκριμένα, τα δεδομένα χαρακτηρίζονται ως δεδομένα εισόδου ή εξόδου ανά εφαρμογή. Τα δεδομένα εισόδου μίας εφαρμογής μπορεί να είναι δεδομένα εξόδου για κάποια άλλη. Ο συγκεκριμένος τρόπος, εξυπηρετεί στην επαναχρησιμοποίηση των concepts χωρίς να δεσμεύεται το κάθε ένα από αυτά με στατικά δεδομένα εισόδου ή εξόδου. Με αυτόν τον τρόπο δημιουργείται ένα μητρώο (registry) από gadgets, που είναι διαθέσιμα προς χρησιμοποίηση από τους χρήστες.

3.2. Αλγόριθμος λειτουργίας

Στο συγκεκριμένο κεφάλαιο περιγράφεται ο αλγόριθμος λειτουργίας του μηχανισμού καθώς και ο τρόπος με τον οποίο εκτελούνται οι διαδικασίες του συστήματος και πως παράγεται το τελικό αποτέλεσμα στο χρήστη. Περιγράφεται ο τρόπος και η σειρά με την οποία καλούνται και εκτελούνται οι διαφορετικές οντότητες που απαρτίζουν το σύστημα. Ως οντότητες θεωρούνται το επίπεδο του client, οι χρήστες και οι φυλλομετρητές (browsers), το επίπεδο του application server, το web service και το εξωτερικό σύστημα σχεδιασμού (planner) και τέλος ο database server, δηλαδή η βάση δεδομένων. Ακόμα, περιγράφεται αναλυτικά η επικοινωνία του χρήστη με το σύστημα και η επικοινωνία των εσωτερικών διαφορετικών οντοτήτων μεταξύ τους.

Αρχικά, όπως φαίνεται στην παρακάτω Εικόνα 3.2 ο χρήστης εισέρχεται μέσω ενός browser στην αρχική σελίδα που είναι διαθέσιμος ο μηχανισμός σύνθεσης και πρότασης gadget (βήμα 1). Με την είσοδό του σε αυτή τη σελίδα, ο χρήστης ζητάει από το server όλες τις διαθέσιμες εξόδους που έχουν εγκατασταθεί στο σύστημα. Σύμφωνα με αυτές τις εξόδους ο χρήστης μπορεί να αναζητήσει την τελική λειτουργικότητα που χρειάζεται να εγκατασταθεί στο προφίλ του. Στη συνέχεια, όπως φαίνεται στο βήμα 2, ο server επικοινωνεί με τη βάση επιλέγοντας τα concepts εκείνα που είναι αποτελούν δεδομένα εξόδου (outputs) για το σύνολο των εγκατεστημένων gadgets. Η βάση επιστρέφει τα διαθέσιμα outputs στο server (βήμα 3) και αυτός τα στέλνει με τη σειρά του στο browser του client (βήμα 4), ώστε να μπορεί να επιλέξει την έξοδο που επιθυμεί.



Εικόνα 3.2 Ροή επικοινωνίας οντοτήτων συστήματος

Στη συνέχεια δίνεται η δυνατότητα στον client να επιλέξει μέσω των διαθέσιμων εξόδων ποια είναι η τελική λειτουργικότητα που θέλει να εκτελέσει. Με την επιλογή του (βήμα 5), ζητά από το web service το gadget ή το σύνολο των gadgets που πληρούν το κριτήριό του. Με αυτό τον τρόπο ο client ζητά από το server τη λύση σύμφωνα με την έξοδο που έχει επιλέξει.

Σε αυτό το σημείο γίνονται οι απαραίτητες επικοινωνίες μεταξύ web service και database προκειμένου να ικανοποιηθεί αργότερα το εξωτερικό σύστημα σχεδιασμού. Αρχικά, ο server ζητά από τη βάση όλα τα διαθέσιμα εγκατεστημένα gadgets καθώς και τα δεδομένα εισόδου και εξόδου του καθενός (βήμα 6). Η βάση επιστρέφει στο server το σύνολο των δεδομένων που της ζητήθηκαν (βήμα 7). Η επικοινωνία αυτή εξυπηρετεί στο να μπορέσει ο server να δημιουργήσει το αρχείο domain.pddl, το οποίο περιγράφει όλο τον διαθέσιμο κόσμο που υπάρχει στη βάση σε γλώσσα PDDL. Στα βήματα 6 και 7 ζητούνται ακόμα από τη βάση δεδομένων όλα τα δεδομένα εισόδου που αποτελούν κύρια πληροφορία. Η εκτέλεση αυτής της διεργασίας εξυπηρετεί το server ώστε να μπορέσει να δημιουργήσει το αρχείο problem.pddl, το οποίο περιγράφει το

πρόβλημα που τίθεται προς επίλυση σύμφωνα με την επιλογή του χρήστη. Αφού δημιουργηθούν τα δύο παραπάνω αρχεία, το web service εκτελεί τον planner (βήμα 8). Στη συνέχεια ο planner με τη σειρά παράγει το αρχείο λύσης .SOL (βήμα 9). Το web service αναλαμβάνει να διαβάσει το αρχείο που παράχθηκε (βήμα 10). Σύμφωνα με τα gadgets που διαβάστηκαν από το αρχείο, το web service επικοινωνεί ξανά με τη βάση (βήμα 11), για να μπορέσει να βρει πληροφορίες για το κάθε ένα από αυτά. Οι πληροφορίες που αναζητούνται είναι:

- Η περιγραφή του gadget
- Οι περιγραφές των δεδομένων εισόδου
- Η περιγραφή της εξόδου
- Οι συνδέσεις του συγκεκριμένου gadget με άλλα gadgets, δηλαδή ποια gadgets έχουν ως είσοδο την έξοδο που παράγει το πρώτο

Αφού το web service συλλέξει τις παραπάνω πληροφορίες και δημιουργήσει την κατάλληλη δομή (format), τις αποστέλλει στον client (βήμα 12). Εκεί με τη σειρά του ο browser του client δημιουργεί την αναπαράσταση των δεδομένων και συνθέτει τα αποτελέσματα, σύμφωνα με αυτά που έχει λάβει από την απάντηση του server (βήμα 13).

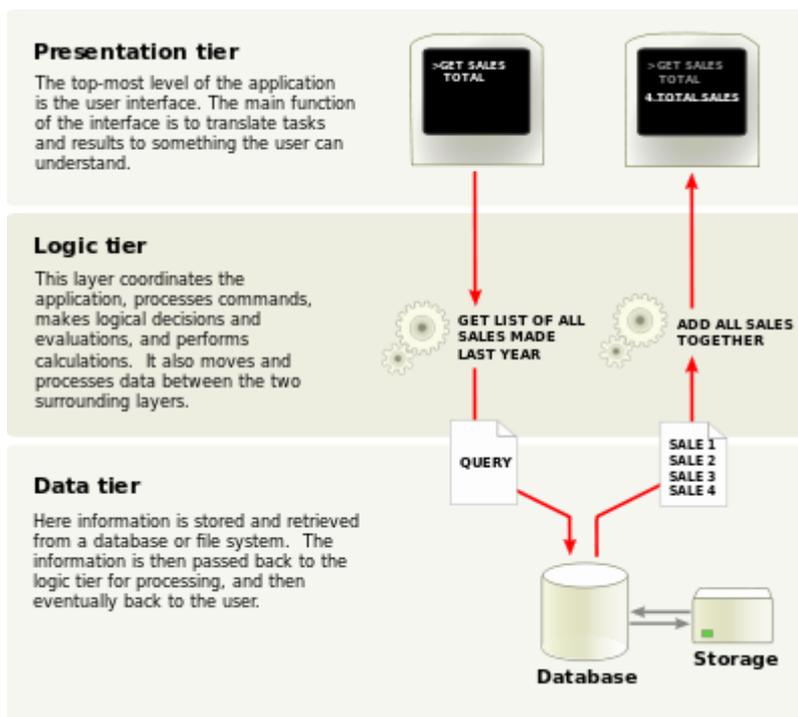
3.3. Αρχιτεκτονική συστήματος

Η αρχιτεκτονική του μηχανισμού σύνθεσης και πρότασης gadgets βασίζεται στην αρχιτεκτονική τριών στρωμάτων (3-tier architecture). Η αρχιτεκτονική 3-tier είναι ένας εξελιγμένος τύπος της αρχιτεκτονικής client-server. Σύμφωνα με αυτήν την αρχιτεκτονική, όπως φαίνεται στην Εικόνα 3.3 το σύστημα αποτελείται από τρία διακριτά επίπεδα:

1ο Επίπεδο – Data Layer – Database Server, αποτελεί το βασικότερο επίπεδο του συστήματος. Ο Database Server παρέχει όλες τις απαραίτητες λειτουργίες για την αποθήκευση, ανάκτηση, ενημέρωση και συντήρηση των δεδομένων του συστήματος καθώς επίσης και όλους τους απαραίτητους μηχανισμούς για την ακεραιότητα των δεδομένων (Data Integrity).

2ο Επίπεδο – Business Logic Layer – Application Server, Αποτελεί το κύριο τμήμα του λογισμικού, Στο επίπεδο αυτό υλοποιούνται όλες εκείνες οι διαδικασίες που είναι υπεύθυνες για την διαχείριση των δεδομένων που φυλάσσονται στη βάση του 1ο επιπέδου καθώς και οι κανόνες που χαρακτηρίζουν την επιχειρηματική λογική της εφαρμογής.

3ο Επίπεδο – Presentation Layer – User Interface, αποτελεί τη επαφή του χρήστη με το σύστημα (User Interface). Στο επίπεδο αυτό, πραγματοποιείται η διαχείριση των Οθονών Εργασίας (User Screens) καθώς επίσης και η μορφοποίηση των δεδομένων που εμφανίζονται.

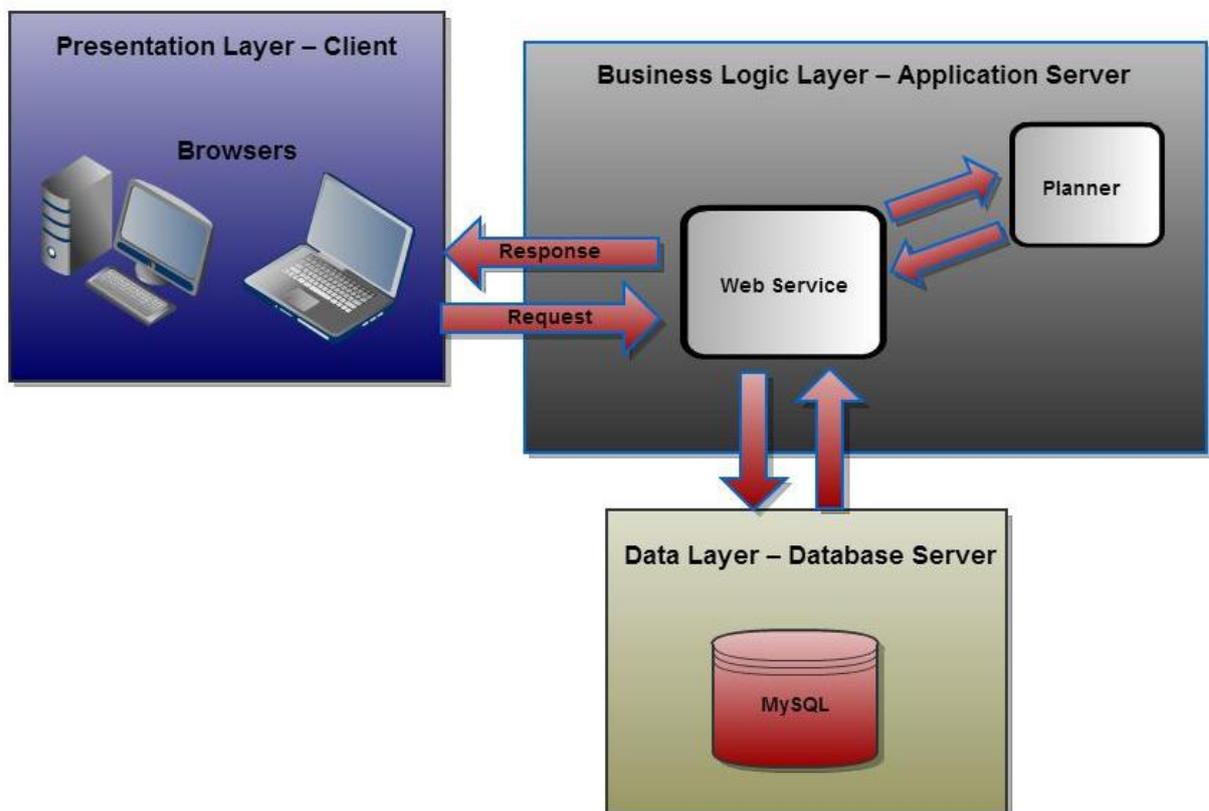


Εικόνα 3.3 Αρχιτεκτονική τριών επιπέδων (Πηγή: http://en.wikipedia.org/wiki/Multitier_architecture)

Η λογική της εφαρμογής, σχεδόν στο σύνολό της, τρέχει σε έναν ενδιάμεσο Server (τον Application Server) και απομένει ένα πολύ μικρό ποσοστό στον Client. Προς τον ενδιάμεσο Application Server απευθύνει τις αιτήσεις του για δεδομένα ο Client και ο Application Server τα διαβιβάζει στο RDBMS που χειρίζεται τα δεδομένα της βάσης. Στη συνέχεια, τα δεδομένα ακολουθούν αντίστροφη πορεία: αφού τα ανασύρει από τη βάση, το RDBMS τα στέλνει στον Application Server, όπου λαμβάνει χώρα η επεξεργασία τους και από εκεί καταλήγουν στον

client του τελικού χρήστη για μορφοποίηση και προβολή. Ο Client δεν επικοινωνεί ποτέ απευθείας με το RDBMS. Το σύνολο των δεδομένων που ανταλλάσσονται μεταξύ αυτών των δύο επιπέδων περνά από τον ενδιάμεσο Application Server (το λεγόμενο middleware).

Συγκεκριμένα, στην παρούσα εργασία δημιουργήθηκαν και χρησιμοποιήθηκαν μία διαδικτυακή εφαρμογή που περιέχει το γραφικό περιβάλλον (User Interface) για το χρήστη, ένα Web Service που περιέχει τη λογική και την υλοποίηση της σύνθεσης των gadget, ένας planner για τη λύση με του προβλήματος με ευφυείς τεχνικές και μια βάση δεδομένων, που περιέχει τις πληροφορίες των gadgets, των δεδομένων εισόδου και εξόδου, στο εξής concepts, καθώς και τον τρόπο που αυτά συνδέονται μεταξύ τους. Η αρχιτεκτονική που εφαρμόστηκε στην παρούσα εργασία, παρουσιάζεται στην παρακάτω Εικόνα 3.4.

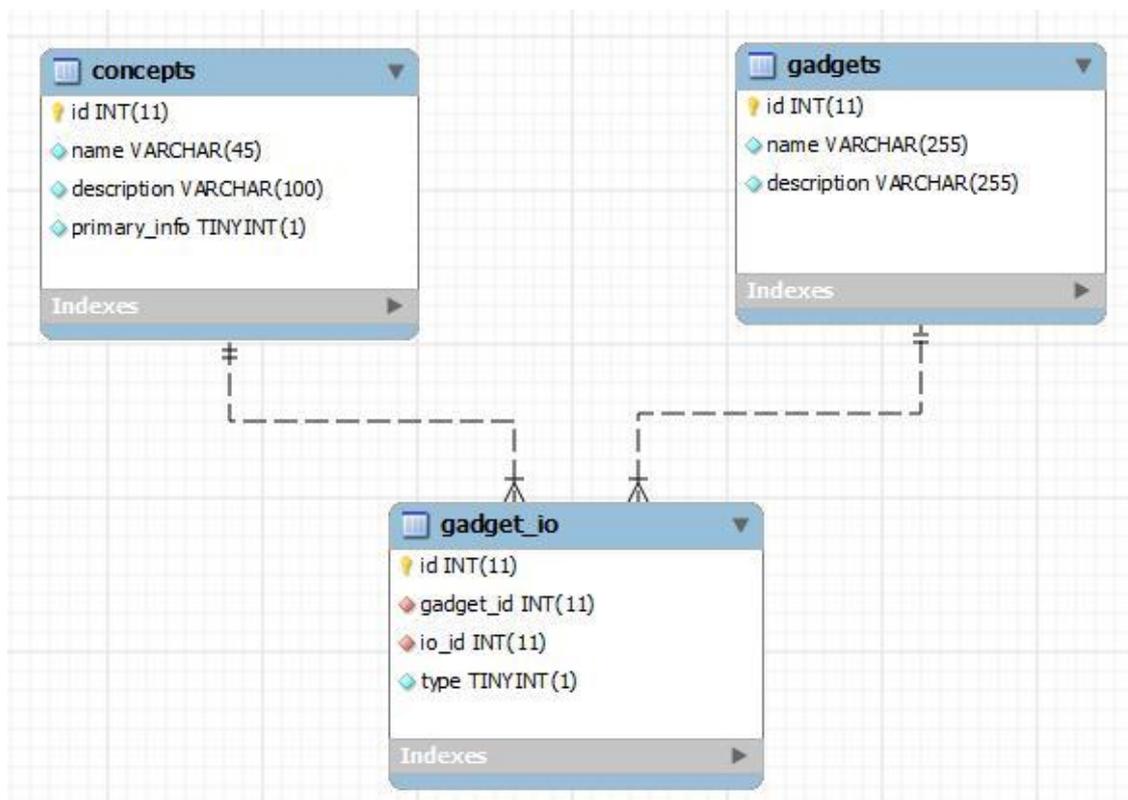


Εικόνα 3.4 Αρχιτεκτονική τριών επιπέδων για το μηχανισμό πρότασης και σύνθεσης gadget

3.4. Υλοποίηση Συστήματος

3.4.1. Data Layer - Βάση Δεδομένων

Για την αποθήκευση των πληροφοριών των gadgets και των concepts χρησιμοποιήθηκε η σχεσιακή βάση δεδομένων MySQL, και επεκτάθηκε η πρότυπη βάση δεδομένων που προτείνει το OpenSocial και το Shindig για κοινωνικά δίκτυα. Σκοπός της εν λόγω επέκτασης ήταν η δημιουργία ενός πρότυπου τρόπου περιγραφής των διαθέσιμων gadgets, ώστε να δημιουργηθεί ένα είδος μητρώου (registry), στο οποίο θα μπορεί να ανατρέξει ο μηχανισμός όταν αναζητεί τα κατάλληλα gadgets. Δημιουργήθηκαν οι παρακάτω πίνακες όπως φαίνονται στην παρακάτω Εικόνα 3.5.



Εικόνα 3.5 Σχήμα βάσης δεδομένων

Αναλυτικότερα, οι πληροφορίες που περιέχουν οι τρεις πίνακες είναι:

1. **Gadgets**, αποθηκεύει τις βασικές πληροφορίες ενός gadget, όπως το όνομά του (name) και μία περιγραφή (description) πιο φιλική προς το χρήστη.
2. **Concepts**, αποθηκεύει τα δεδομένα εισόδου και εξόδου, αποτελούμενα από το όνομά τους (name) και μία περιγραφή (description) πιο φιλική για το χρήστη. Σε αυτόν τον πίνακα ένα concept που είναι δεδομένο εισόδου αλλά και εξόδου ανάλογα το gadget, εγγράφεται μία φορά στη βάση. Υπάρχει ακόμα κι ένα πεδίο το *primary_info* το οποίο υποδεικνύει αν το συγκεκριμένο concept αποτελεί κύρια ή πρωτεύουσα πληροφορία, σύμφωνα με αυτά που έχουν περιγραφεί παραπάνω.
3. **Gadget_io**, αποθηκεύει τις σχέσεις μεταξύ των gadgets και των concepts. Περιέχει κι ένα πεδίο το οποίο υποδηλώνει αν για το συγκεκριμένο gadget το concept αποτελεί είσοδο ή έξοδο.

3.4.2. Business Logic Layer

Για τις ανάγκες της εργασίας στο επίπεδο του Application Server χρειάζεται να δημιουργηθεί ένα Web Service που διαχειρίζεται τις αιτήσεις του χρήστη και τις μεταβιβάζει στη βάση δεδομένων καθώς και ο εξωτερικός Planner που θα συνθέσει τη λύση του προβλήματος σύμφωνα με τις απαιτήσεις του χρήστη.

3.4.2.1. Web Service

Για τις ανάγκες εξυπηρέτησης του Client υλοποιήθηκε ένα Web Service και χρησιμοποιήθηκαν οι ακόλουθες τεχνολογίες:

- Apache Axis Framework
- Gson library της Google

Το συγκεκριμένο Web Service περιέχει δύο μεθόδους που εκτίθενται στον Client οι οποίες είναι:

1. Η `getOutputs()`, επιστρέφει όλα τα διαθέσιμα concepts τα οποία είναι outputs και ο χρήστης μπορεί να τα αναζητά ως λειτουργικότητα. Το Response που επιστρέφει είναι σε JSON form. Το κάθε output είναι ένα JSONArray που περιέχει το id και το description του concept, όπως ενδεικτικά φαίνεται στην παρακάτω Εικόνα 3.6.

```
[
  ["8", "Έγκυρη Καταβολή εισφορών ηλεκτρονικά μέσω web banking"],
  ["9", "Παραλαβή κωδικού απόδειξης πληρωμής για τις εισφορές"],
  ["10", "Παράδοση εγγράφων αποφοίτησης έγκυρη"],
  ["11", "Ημερομηνία ορκομωσίας"],
  ["12", "Υπολογισμός ποσού εισφορών"],
  ["13", "Έγκυρος αριθμός μητρώου εργοδότη"],
  ["14", "Έγκυρος αριθμός φορολογικού μητρώου"],
  ["15", "Έγκυρη δήλωση ποσού που καταβάλλεται"],
  ["16", "Έγκυρη δήλωση μισθολογικής περιόδου"],
  ["17", "Κανένα οφειλόμενο μάθημα"],
  ["18", "Υποβολή πτυχιακής έγκυρη"],
  ["19", "Κανένα οφειλόμενο βιβλίο"],
  ["24", "Αντίγραφο Φορολογικής Ενημερότητας"],
  ["26", "Αντίγραφο πιστοποιητικού οικογενειακής κατάστασης "],
  ["29", "Αντίγραφο Βεβαίωσης σπουδών αδελφού-ής"],
  ["31", "Αντίγραφο Βεβαίωσης Πολυτέκνων"],
  ["33", "Αριθμός κάρτας σίτισης"]
]
```

Εικόνα 3.6 Response Web Service για τις διαθέσιμες εξόδους σε JSON format

2. Η `getSolution(int output_id)`, είναι η βασική συνάρτηση του Web Service. Παίρνει ως όρισμα το id του concept (`output_id`) και επιστρέφει τη λύση που έχει βρεθεί σε JSON format. Η δομή του συγκεκριμένου Response περιέχει όλη την πληροφορία που χρειάζεται ο Client για να δημιουργηθεί το γραφικό περιβάλλον. Συγκεκριμένα, περιέχει ένα σύνολο από JsonArrays, που το κάθε ένα περιέχει με την εξής σειρά τα ακόλουθα, όπως φαίνεται στην παρακάτω Εικόνα 3.7:

- Τον μοναδικό αναγνωριστικό αριθμό (id) του gadget
- Την περιγραφή (description) του gadget
- Το επίπεδο (level) του gadget, σύμφωνα με τη λύση που δόθηκε, έτσι ώστε να μπορέσει να τοποθετηθεί σωστά σαν element στο γραφικό περιβάλλον
- Ένα jsonArray με τις συνδέσεις (connections) του gadget, δηλαδή με ποια άλλα gadgets θα συνδεθεί γραφικά
- Τα δεδομένα εισόδου (inputs) που δέχεται
- Τα δεδομένα εξόδου (outputs) που παράγει

```
[
  [
    "6", "Υπολογισμός εισφορών με βάση την ΑΠΔ που έχει καταθέσει ηλεκτρονικά", "0",
    [
      "7", "8", "9", "10"
    ],
    [
      "Όνομα", "Επώνυμο", "Αριθμός Δελτίου Ταυτότητας", "Αριθμός μητρώου ασφαλισμένου"
    ],
    [
      "Υπολογισμός ποσού εισφορών"
    ]
  ]
]
```

Εικόνα 3.7 Response Web Service για τη λύση που βρέθηκε σε JSON format

3.4.2.2. Εξωτερικό Σύστημα Σχεδιασμού - Planner

Για την εύρεση και σύνθεση των gadgets που εξυπηρετούν μία τελική λειτουργικότητα, μέσω ευφώνων τεχνικών χρησιμοποιήθηκε το σύστημα σχεδιασμού LPG-td (Local search for Planning Graphs) [15], το οποίο είναι ένα σύστημα βασισμένο σε γράφους σχεδιασμού και διαχειρίζεται πεδία ορισμού σύμφωνα με την PDDL2.1.

Στον Application Server, χρειάζεται να είναι εγκατεστημένο το εκτελέσιμο αρχείο lpg-td-1.0.exe. Το σύστημα μόλις δεχθεί αίτηση από το χρήστη (Client), δημιουργεί δυναμικά τα εξής αρχεία:

1. **gadgetDomain.pddl**, το οποίο περιέχει όλο το πεδίο ορισμού (domain) των gadgets, των concepts και τον τρόπο με τον οποίο αυτά συνδέονται μεταξύ τους. Κάθε φορά που τρέχει το σύστημα διαβάζει από τη βάση δεδομένων, όπως περιγράφηκε παραπάνω, τα διαθέσιμα gadgets και τα δεδομένα εισόδου και εξόδου του καθενός. Το συγκεκριμένο αρχείο, αν δεν έχει γίνει προσθήκη εγγραφών στη βάση παραμένει το ίδιο για κάθε πρόβλημα που δημιουργείται. Αρχικά, όπως ενδεικτικά φαίνεται στην παρακάτω Εικόνα 3.8, ορίζεται το όνομα του domain, gadgets. Στη συνέχεια δημιουργούνται όλα τα concepts, predicates σύμφωνα με τη γλώσσα της PDDL, που είναι διαθέσιμα στη βάση.

```
(define (domain gadgets)
  (:predicates (block ?b)
    (Name ?b)
    (Lastname ?b)
    (Library_Username ?b)
    (RN ?b)
    (No_Dept_Books ?b)
    (Thesis_Submitted ?b)
    (None_Remaining_Courses ?b)
    (Graduation_Documents_Submitted ?b)
    (Date_Graduation_Apply ?b)
  )
)
```

Εικόνα 3.8 predicates αρχείου domain.pddl

Ανεξάρτητα από τα gadgets σε αυτό το σημείο, τα δεδομένα εισόδου και εξόδου ορίζονται αυτόνομα και ανεξάρτητα. Στο ίδιο αρχείο ορίζονται και τα gadgets, actions σύμφωνα με την PDDL. Όπως παρατηρείται στην παρακάτω Εικόνα 3.9, για το κάθε gadget εγγράφονται όλα τα στοιχεία που το ορίζουν, δηλαδή το όνομά του, τι στοιχεία εισόδου δέχεται και τι παράγει ως αποτέλεσμα.

```

(:action Library_Book_Account_Student
  :parameters (?b)
  :precondition (and (block ?b) (Name ?b) (Lastname ?b) (Library_Username ?b) (RN ?b))
  :effect (No_Dept_Books ?b)
)
(:action Library_Thesis_Submission
  :parameters (?b)
  :precondition (and (block ?b) (Name ?b) (Lastname ?b) (Library_Username ?b) (RN ?b))
  :effect (Thesis_Submitted ?b)
)
(:action Remaining_Courses
  :parameters (?b)
  :precondition (and (block ?b) (Name ?b) (Lastname ?b) (RN ?b))
  :effect (None_Remaining_Courses ?b)
)
(:action Graduation_Documents
  :parameters (?b)
  :precondition (and (block ?b) (Name ?b) (Lastname ?b) (RN ?b))
  :effect (Graduation_Documents_Submitted ?b)
)
(:action Graduation_Apply
  :parameters (?b)
  :precondition (and (block ?b) (None_Remaining_Courses ?b) (No_Dept_Books ?b) (Thesis_Submitted ?b) (Graduation_Documents_Submitted ?b))
  :effect (Date_Graduation_Apply ?b)
)
)

```

Εικόνα 3.9 actions αρχείου domain.pddl

2. **gadgetProblem.pddl**, το οποίο περιέχει τον ορισμό του προβλήματος που τίθεται προς επίλυση. Το συγκεκριμένο αρχείο διαφοροποιείται και εγγράφεται εκ νέου κάθε φορά που ο χρήστης αναζητά τη λύση διαφορετικού προβλήματος. Όπως ενδεικτικά φαίνεται στην παρακάτω Εικόνα 3.10, ορίζεται το όνομα του προβλήματος (gadgetproblem) και στη συνέχεια ορίζεται μέσω ποιου domain (gadgets) θα γίνει η αναζήτηση της λύσης. Ορίζονται τα δεδομένα εισόδου, τα οποία αποτελούν πρωτεύουσα πληροφορία και στο τέλος ορίζεται το πρόβλημα που χρειάζεται λύση.

```

(define (problem gadgetproblem)
  (:domain gadgets)
  (:objects concept)
  (:init (block concept)
    (Name concept)
    (Lastname concept)
    (Library_Username concept)
    (RN concept)
    (Identification_Card_Number concept)
    (Social_Security_registration_number concept)
    (Financial_Id concept)
    (Personal_Id concept)
    (Aspe_Id concept)
    (Sibling_Study_Department concept)
    (Sibling_Password concept)
    (Password concept)
  )
  (:goal (Date_Graduation_Apply concept)
  )
)

```

Εικόνα 3.10 αρχείο problem.pddl

3. **plan_gadgetProblem.pddl_1.SOL**, το οποίο παράγεται όταν τελειώσει η εκτέλεση του συστήματος από το εξωτερικό σύστημα σχεδιασμού ενεργειών. Περιέχει τα ονόματα ενός ή περισσότερων gadgets στην περίπτωση που έχει βρεθεί λύση όπως φαίνεται στην Εικόνα 3.11.

```

; Version LPG-td-1.0
; Seed 71414320
; Command line: lpg-td-1.0 -o gadgetDomain.pddl -f gadgetProblem.pddl -speed
; Problem gadgetProblem.pddl
; Actions having STRIPS duration
; Time 0.01
; Search time 0.00
; Parsing time 0.00
; Mutex time 0.00
; Quality 7

```

Time 0.01

```

0: (CALCULATE_CONTRIBUTION CONCEPT) [1]
1: (STATEMENT_WAGE_PERIOD CONCEPT) [1]
1: (STATEMENT_INSURANCE_NUMBER CONCEPT) [1]
1: (STATEMENT_AMOUNT_PAID CONCEPT) [1]
1: (STATEMENT_TAXPAYER_IDENTIFICATION_NUMBER CONCEPT) [1]
2: (ELECTRONIC_CONTRIBUTION CONCEPT) [1]
3: (CODE_RETRIEVAL_PAYMENT_RECEIPT CONCEPT) [1]

```

Εικόνα 3.11 αρχείο SOL στο οποίο έχει βρεθεί λύση

Στη περίπτωση που το πρόβλημα δεν επιδέχεται λύση το αρχείο περιέχει την εγγραφή *'no solution found'*, όπως φαίνεται στην Εικόνα 3.12.

```

; Version LPG-td-1.0
; Seed 36784437
; Command line: lpg-td-1.0 -o gadgetDomain.pddl -f gadgetProblem.pddl -speed
; Problem gadgetProblem.pddl
; Actions having STRIPS duration
; Time 0.00
; Search time 0.00
; Parsing time 0.00
; Mutex time 0.00
; Quality 0

```

Time 0.00

no solution

Εικόνα 3.12 αρχείο SOL στο οποίο δεν έχει βρεθεί λύση

3.4.2.3. Presentation Layer – User Interface

Το επίπεδο αυτό αφορά στην γραφική απεικόνιση που θα έχει πρόσβαση ο χρήστης μέσω του περιηγητή (browser). Πρόκειται για την ιστοσελίδα που αποδίδεται από το τερματικό του χρήστη, αποτελούμενου από κώδικα που εκτελείται στον εξυπηρετούμενο (client). Για τις ανάγκες αυτού του επιπέδου χρησιμοποιήθηκαν οι παρακάτω τεχνολογίες:

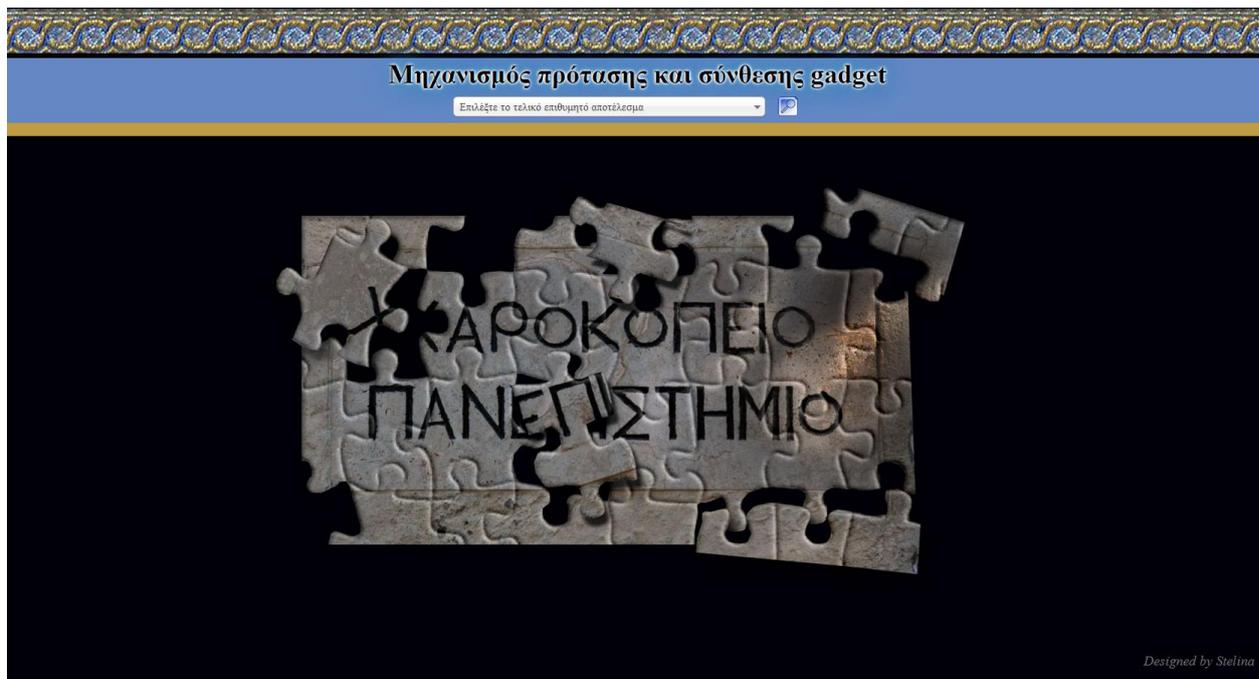
1. HTML, για τη δημιουργία και παρουσίαση της σελίδας
2. CSS, για τη μορφοποίηση της σελίδας
3. jQuery, για τη δυναμική προσθήκη στοιχείων στη σελίδα
4. jsPlumb, εξωτερική βιβλιοθήκη για την αναπαράσταση των gadgets, concepts και των συνδέσεων μεταξύ τους
5. JSP, το οποίο είναι υπεύθυνο για την κλήση του Web Service

4. Περιπτώσεις χρήσης

Ο μηχανισμός σύνθεσης και πρότασης εφαρμογών δημιουργήθηκε και αναπτύχθηκε στο πλαίσιο του κοινωνικού δικτύου Unity, σύμφωνα με τις ανάγκες που προκύπτουν στην ακαδημαϊκή κοινότητα. Ωστόσο, η εγκατάστασή του μπορεί να γίνει σε οποιοδήποτε κοινωνικό δίκτυο αρκεί να είναι συμβατός με τις τεχνολογίες και τα τρέχοντα πρότυπα.

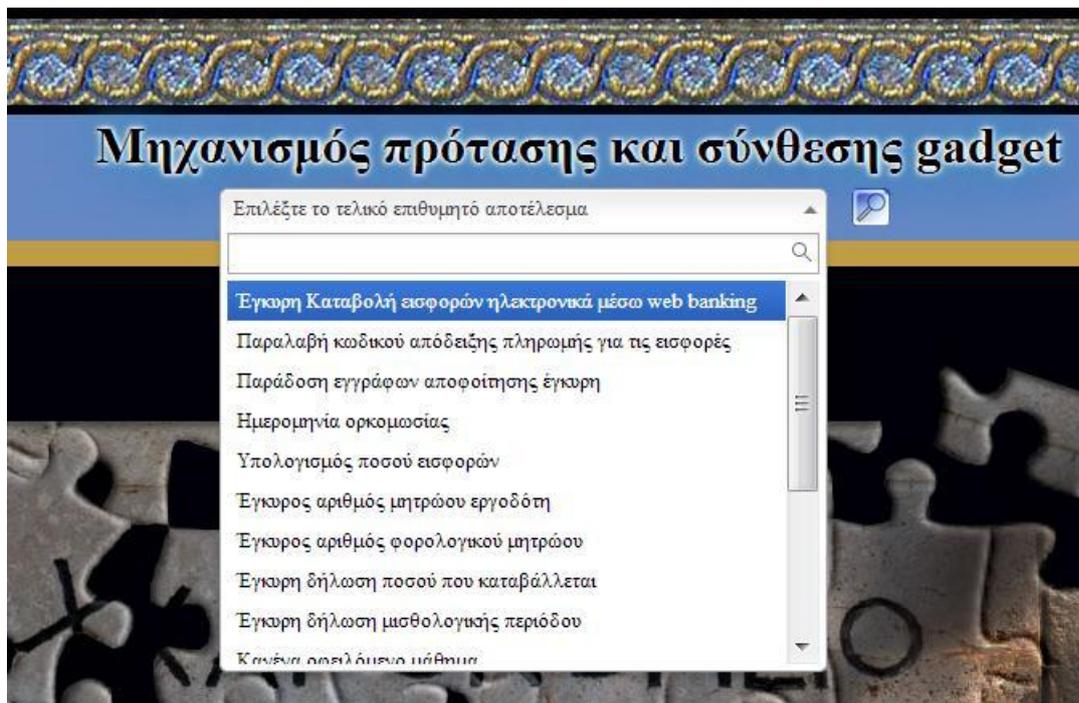
Ο μηχανισμός πρότασης σύνθετων εφαρμογών δοκιμάστηκε με εφαρμογές τόσο στο περιβάλλον του κοινωνικού δικτύου Unity όσο και σε περιπτώσεις εκτός αυτού. Στην πρώτη περίπτωση τα gadgets που τέθηκαν υπό δοκιμές είναι, η αίτηση ενός φοιτητή για κάρτα σίτισης καθώς και η αίτηση αποφοίτησης. Κάθε ένα από αυτά χρειάζεται ένα σύνολο από άλλα gadgets, για να μπορέσει να αποδώσει τη λειτουργικότητά του. Εκτός του κοινωνικού δικτύου Unity, δόθηκε έμφαση και παραδείγματα που αφορούν στις συναλλαγές επιχειρήσεων με τις δημόσιες υπηρεσίες, όπως η καταβολή εργοδοτικών εισφορών στο ΙΚΑ. Στο συγκεκριμένο παράδειγμα φαίνεται ξεκάθαρα η ανάγκη σύνθεσης διαφορετικών υπηρεσιών, καθώς για να επέλθει το τελικό αποτέλεσμα χρειάζεται η χρήση πολλών υπηρεσιών από διαφορετικά επίπεδα.

Σε κάθε περίπτωση χρήσης που περιγράφεται παρακάτω ο χρήστης χρειάζεται να εισέλθει στην αρχική οθόνη του συστήματος, ώστε να μπορέσει στη συνέχεια να αναζητήσει τις λειτουργικότητες που επιθυμεί. Στην παρακάτω Εικόνα 4.1 φαίνεται η αρχική σελίδα του μηχανισμού σύνθεσης και πρότασης gadget.



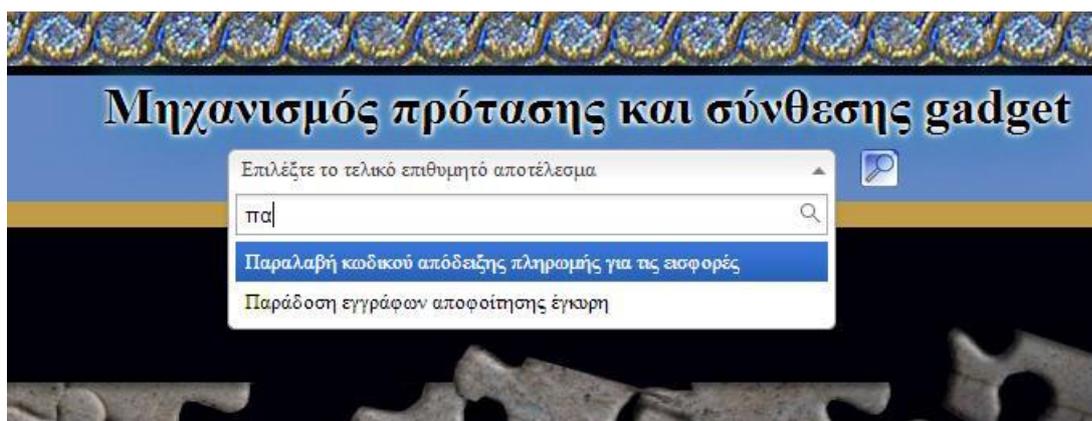
Εικόνα 4.1 Αρχική οθόνη μηχανισμού

Ο χρήστης καλείται να επιλέξει την τελική έξοδο που επιθυμεί ώστε ο μηχανισμός να του προτείνει τα απαραίτητα για την επιλογή του gadgets που θα τον βοηθήσουν ώστε να επιτύχει τον τελικό του στόχο. Ο μηχανισμός περιέχει ένα σύγχρονο τρόπο παρουσίασης των συνολικών διαθέσιμων εξόδων, ώστε να μπορεί ο χρήστης να αναζητά κάποια έξοδο όπως αυτό φαίνεται στην Εικόνα 4.2.



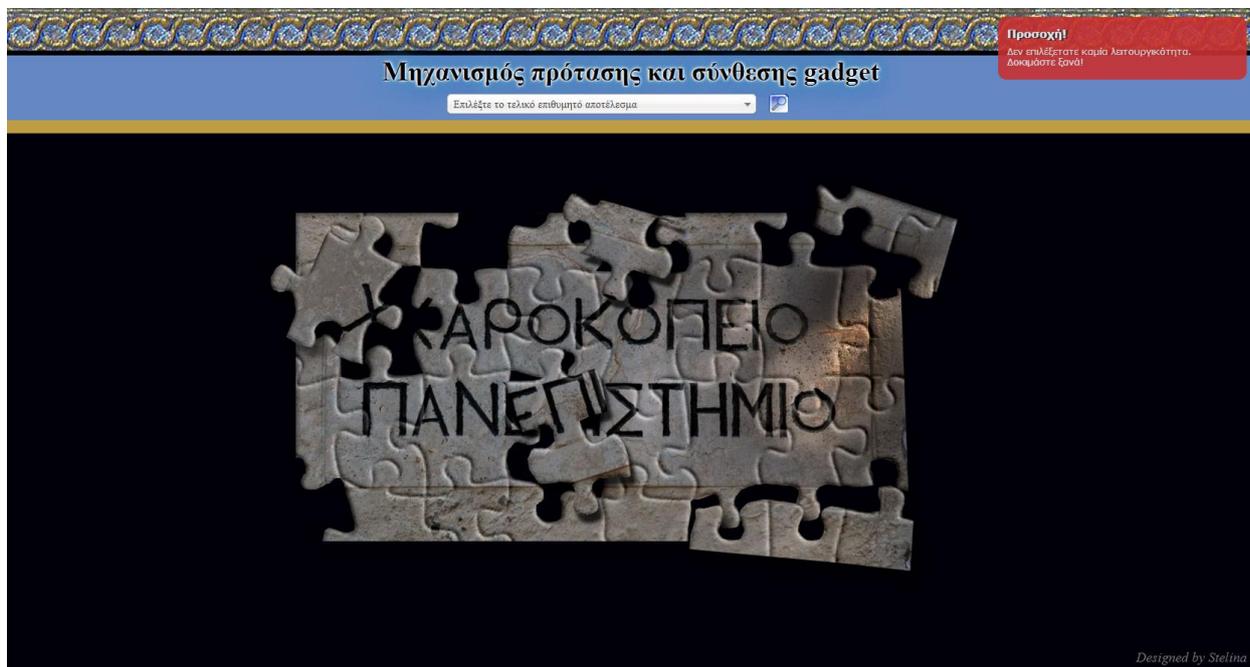
Εικόνα 4.2 Μηχανισμός αναζήτησης διαθέσιμων εξόδων

Ο χρήστης μπορεί να εισάγει το όνομα της εξόδου που επιθυμεί και ταυτόχρονα να τα αποτελέσματά του φιλτράρονται σύμφωνα με αυτά που έχει εισάγει, όπως φαίνεται στην Εικόνα 4.3.



Εικόνα 4.3 Φιλτραρισμένα αποτελέσματα σύμφωνα με την επιλογή του χρήστη

Στην περίπτωση που ο χρήστης δεν επιλέξει κάποια λειτουργικότητα και πατήσει αναζήτηση τότε ο μηχανισμός του εμφανίζει ένα προειδοποιητικό μήνυμα, όπως φαίνεται στην παρακάτω Εικόνα 4.4.



Εικόνα 4.4 Μήνυμα λάθους

Στα παρακάτω κεφάλαια παρατίθενται εικόνες από το κάθε σενάριο χρήσης που περιγράφεται αναλυτικά.

4.1. Unity

Όπως αναφέρεται και προηγουμένως η παρούσα χρησιμοποίησε ως έναυσμα τον Ιστότοπο Κοινωνικής Δικτύωσης Unity, ο οποίος έχει αναπτυχθεί στο Τμήμα Πληροφορικής και Τηλεματικής Χαροκόπειου Πανεπιστημίου στο πλαίσιο των πτυχιακών εργασιών των φοιτητών Χούδρα Βαλεντίνο (2011) , Κατσιβέλη-Περάκη Παναγιώτη (2011), Καπουράνη Χάντριολντ-Ανδρέα (2012), Μελετάκης Ιωάννης (2012). Ο συγκεκριμένος ΙΚΔ έχει αναπτυχθεί σύμφωνα με το πρότυπο της Google, το OpenSocial και έχει επεκταθεί ώστε να καλύπτει τις ανάγκες της ακαδημαϊκής κοινότητας.

Το Unity επιλέχθηκε καθώς καλύπτει ήδη κάποιες από τις βασικές λειτουργικές απαιτήσεις που αναφέρονται παραπάνω. Συγχρόνως, έχει δημιουργηθεί για λογαριασμό του Χαροκόπειου Πανεπιστημίου και για αυτό τον λόγο είναι προσβάσιμο και διαχειρίσιμο από φοιτητές του Πανεπιστημίου. Επιπλέον, παρέχεται η δυνατότητα επέμβασης στον πηγαίο κώδικα του Ιστότοπου καθώς και της βάσης δεδομένων, γεγονός το οποίο παρέχει τη δυνατότητα να υλοποιηθούν οι όποιες επιπλέον λειτουργίες θεωρηθούν αναγκαίες και δεν υποστηρίζονται ήδη.

4.1.1. Αίτηση φοιτητή για κάρτα σίτισης

4.1.1.1. Περιγραφή του προβλήματος

Η αίτηση ενός φοιτητή για κάρτα σίτισης είναι μια υπηρεσία, η οποία απαιτεί μια χρονοβόρα διαδικασία καθώς κρίνεται αναγκαία η επικοινωνία και η ανταλλαγή πληροφοριών με δημόσιες υπηρεσίες για την συλλογή δικαιολογητικών που είναι απαραίτητα για την αίτηση. Η διαδικασία συγκέντρωσης των απαιτούμενων εγγράφων, απαιτεί τη φυσική παρουσία του ατόμου σε πολλές αρμόδιες δημόσιες υπηρεσίες, κάτι που αποθαρρύνει ένα μεγάλο μέρος φοιτητών από το να αιτηθούν για κάρτα σίτισης. Η πολιτική κάθε Πανεπιστημίου ορίζει ποιοι φοιτητές δικαιούνται τη χορήγηση της κάρτας σίτισης βασιζόμενοι σε κριτήρια οικονομικά και οικογενειακής κατάστασης. Ανάλογα τα κριτήρια του κάθε φοιτητή η διαδικασία διαφέρει και διεκπεραιώνεται σύμφωνα με τα παρακάτω στάδια.

1. Κατά την έναρξη του ακαδημαϊκού εξαμήνου ανακοινώνονται από τη Γραμματεία κάθε τμήματος οι ημερομηνίες υποβολής της αίτησης. Σύμφωνα με την κατηγορία του κάθε φοιτητή τα απαραίτητα δικαιολογητικά που θα πρέπει να επισυναφτούν με την αίτηση ποικίλουν όπως φαίνεται στον παρακάτω Πίνακας 4.1.

Κατηγορίες Φοιτητών	Δικαιολογητικά που πρέπει να προσκομίσουν
Φοιτητές που είναι μέλη πολύτεκνης οικογένειας	1. Φορολογική Ενημερότητα 2. Βεβαίωση Ομοσπονδίας Πολυτέκνων

Φοιτητές που είναι μέλη πολύτεκνης οικογένειας και φοιτούν τουλάχιστον δύο προστατευόμενα μέλη της οικογένειας	1. Φορολογική Ενημερότητα 2. Βεβαίωση Ομοσπονδίας Πολυτέκνων 3. Βεβαίωση σπουδών Αδελφού-
Φοιτητές που είναι μέλη τρίτεκνης οικογένειας	1. Φορολογική Ενημερότητα 2. Πιστοποιητικό Οικογενειακής Κατάστασης
Φοιτητές που φοιτούν τουλάχιστον δύο προστατευόμενα μέλη της οικογένειας	1. Φορολογική Ενημερότητα 2. Πιστοποιητικό Οικογενειακής Κατάστασης 3. Βεβαίωση σπουδών Αδελφού-ης που φοιτά σε ΑΕΙ-ΤΕΙ
Φοιτητές που δεν ανήκουν σε καμία από τις παραπάνω κατηγορίες	1. Φορολογική Ενημερότητα

Πίνακας 4.1 Κατηγορίες φοιτητών (Πηγή: Μελετάκης 2012)

Παρατηρείται πως η Φορολογική Ενημερότητα είναι απαραίτητο δικαιολογητικό για όλες τις κατηγορίες φοιτητών καθώς είναι το μοναδικό επίσημο έγγραφο που δηλώνει την οικονομική κατάσταση του αιτούμενου. Όλα τα υπόλοιπα αποτυπώνουν την οικογενειακή κατάσταση του φοιτητή.

2. Ο φοιτητής χρειάζεται να συλλέξει όλα τα απαραίτητα δικαιολογητικά σύμφωνα με την κατηγορία στην οποία ανήκει. Η συλλογή αυτών εγγράφων απαιτεί την συναλλαγή με δημόσιες υπηρεσίες ή την χρήση υπηρεσιών Ηλεκτρονικής Διακυβέρνησης όπου διατίθενται.

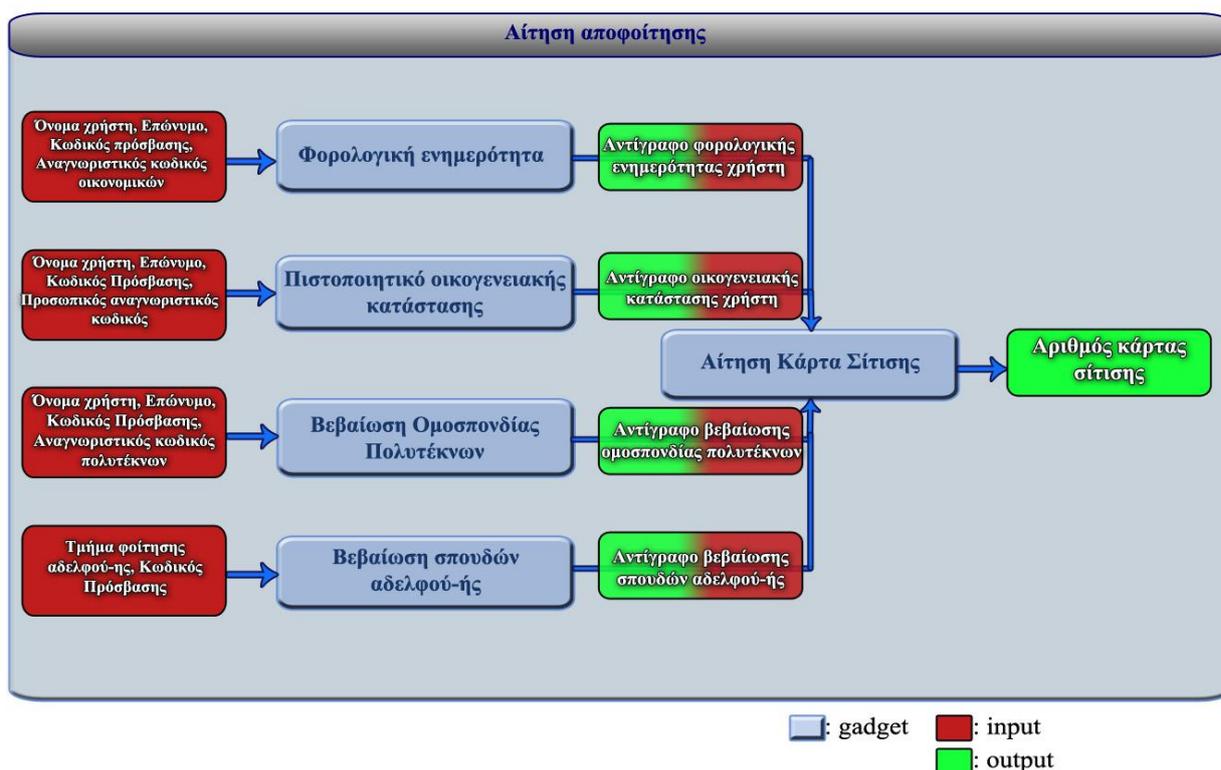
3. Ο φοιτητής προσκομίζει στη Γραμματεία του τμήματος του όλα τα έγγραφα και κάνει αίτηση για την κάρτα σίτισης.

4. Η Γραμματεία επεξεργάζεται τις αιτήσεις για κάρτα σίτισης και σύμφωνα με τα δικαιολογητικά που έχουν προσκομιθεί αποφασίζει αν θα χορηγήσει κάρτα σίτισης στον αιτούμενο φοιτητή λαμβάνοντας υπόψη την πολιτική του Πανεπιστημίου.

5. Ο φοιτητής ενημερώνεται σχετικά με το αποτέλεσμα της αίτησης του.

4.1.1.2. Περιγραφή απαιτούμενων gadgets

Ο φοιτητής χρειάζεται να έχει εγκατεστημένα τα απαιτούμενα gadgets για να μπορέσει να επικοινωνήσει με gadgets άλλων ρόλων, όπως της Γραμματείας ή της Βιβλιοθήκης, ή ακόμα και με άλλες εξωτερικές υπηρεσίες ώστε να μπορέσει να εξυπηρετηθεί. Στην παρακάτω Εικόνα 4.5 Gadgets και δεδομένα εισόδου/εξόδου για κάρτα σίτισης παρουσιάζονται αναλυτικά τα gadgets που απαιτούνται ώστε να λάβει ο φοιτητής την ημερομηνία ορκωμοσίας του. Εκτός από τα gadgets που χρειάζεται να εκτελεστούν παρατίθενται τόσα γραφικά όσο και περιγραφικά τα δεδομένα εισόδου και εξόδου του κάθε gadget.



Εικόνα 4.5 Gadgets και δεδομένα εισόδου/εξόδου για κάρτα σίτισης

1. Το **‘Φορολογική ενημερότητα (Request Financial Status)’**, είναι υπεύθυνο για την απόκτηση της φορολογικής ενημερότητας του χρήστη. Το στοιχείο αυτό είναι απαραίτητο για την διαδικασία της χορήγησης κάρτας σίτισης καθώς η φορολογική ενημερότητα αποτελεί αναγκαίο δικαιολογητικό σε όποια από τις παραπάνω κατηγορίες ανήκει ο φοιτητής. Ο χρήστης χρειάζεται να συμπληρώσει ορισμένα προσωπικά στοιχεία κατά την εκτέλεση του συγκεκριμένου gadget. Τα δεδομένα εισόδου αυτού του gadget είναι τα προσωπικά στοιχεία του φοιτητή, όπως το Όνομα χρήστη, το Επώνυμο, ο κωδικός πρόσβασης και έναν αναγνωριστικό κωδικό οικονομικών (financial Id). Ως δεδομένα εξόδου δίνει ένα αντίγραφο της φορολογικής ενημερότητας του χρήστη.
2. Το **‘Πιστοποιητικό οικογενειακής κατάστασης (Request Family Status)’**, αναλαμβάνει την απόκτηση πιστοποιητικού οικογενειακής κατάστασης του χρήστη. Χρησιμοποιείται από φοιτητές που ανήκουν σε τρίτεκνες οικογένειες και από φοιτητές που έχουν ένα τουλάχιστον αδερφό ή αδερφή που φοιτά σε διαφορετικό ΑΕΙ-ΤΕΙ. Ο χρήστης κατά την εκτέλεση του συγκεκριμένου gadget χρειάζεται να συμπληρώσει ορισμένα προσωπικά στοιχεία για να αποθηκευτεί ένα αντίγραφο της οικογενειακής κατάστασης στο προφίλ του. Τα δεδομένα εισόδου αυτού του gadget είναι τα προσωπικά στοιχεία του φοιτητή, όπως το Όνομα χρήστη, το Επώνυμο, ο κωδικός πρόσβασης και έναν προσωπικό αναγνωριστικό κωδικό (personal Id). Ως δεδομένα εξόδου δίνει ένα του χρήστη ένα αντίγραφο της οικογενειακής κατάστασης του χρήστη.
3. Το **‘Βεβαίωση Ομοσπονδίας Πολυτέκνων (Request Aspe Certification)’**, είναι υπεύθυνο για την απόκτηση βεβαίωσης της ομοσπονδίας πολυτέκνων. Χρησιμοποιείται από φοιτητές που ανήκουν σε πολύτεκνες οικογένειες. Ο χρήστης κατά την εκτέλεση του συγκεκριμένου gadget χρειάζεται να συμπληρώσει ορισμένα προσωπικά στοιχεία για να αποθηκευτεί ένα αντίγραφο της βεβαίωσης της ομοσπονδίας πολυτέκνων στο προφίλ του. Τα δεδομένα εισόδου αυτού του gadget είναι τα προσωπικά στοιχεία του φοιτητή, όπως το Όνομα χρήστη, το Επώνυμο, ο κωδικός πρόσβασης και ένας αναγνωριστικός κωδικός πολυτέκνων (aspe Id). Ως δεδομένα εξόδου δίνει ένα αντίγραφο της βεβαίωσης της ομοσπονδίας πολυτέκνων.
4. Το **‘Βεβαίωση σπουδών αδελφού-ής (Request Study Certification Sibling)’**, είναι υπεύθυνο για την απόκτηση βεβαίωσης σπουδών του αδελφού ή της αδελφής που φοιτά σε διαφορετικό ΑΕΙ-ΤΕΙ. Χρησιμοποιείται από φοιτητές που έχουν κάποιον αδερφό ή αδερφή

που φοιτά σε κάποιο άλλο Ανώτατο ή Τεχνολογικό Εκπαιδευτικό Ίδρυμα. Ο χρήστης κατά την εκτέλεση του συγκεκριμένου gadget χρειάζεται να συμπληρώσει ορισμένα προσωπικά στοιχεία για να αποθηκευτεί ένα αντίγραφο της βεβαίωσης της σπουδών στο προφίλ του. Τα δεδομένα εισόδου αυτού του gadget είναι το Τμήμα φοίτησης του αδελφού-ης και ο κωδικός πρόσβασης αυτού. Ως δεδομένα εξόδου δίνει ένα αντίγραφο της βεβαίωσης σπουδών του αδελφού ή της αδελφής.

5. Το **‘Αίτηση Κάρτα Σίτισης (Apply for Meal Card)**, είναι υπεύθυνο για τη συλλογή των δικαιολογητικών με τα οποία έχει ενημερωθεί το προφίλ του χρήστη από την εκτέλεση των προηγούμενων gadgets. Χρησιμοποιείται από τους φοιτητές που επιθυμούν να αποκτήσουν κάρτα σίτισης. Ο χρήστης κατά την εκτέλεση του συγκεκριμένου gadget χρειάζεται να επιβεβαιώσει τα δικαιολογητικά που έχει συλλέξει και να καταθέσει την αίτηση. Τα δεδομένα εισόδου αυτού του gadget είναι οι επιβεβαιωμένες θετικές απαντήσεις των προηγούμενων δικαιολογητικών. Ως δεδομένα εξόδου δίνει την απάντηση αν έλαβε κάρτα σίτισης μαζί με τα στοιχεία της.

4.1.1.3. Περιγραφή λύσης

Στην περίπτωση της αίτησης για κάρτα σίτισης ο χρήστης χρειάζεται να αναζητήσει το gadget ή το σύνολο των gadgets που χρειάζεται να συντεθούν ώστε να λάβει την τελική λειτουργικότητα που επιθυμεί, δηλαδή την έκδοση της κάρτας σίτισης. Ο χρήστης δεν μπορεί να γνωρίζει εκ των προτέρων όλα τις εφαρμογές και τα δεδομένα που θα του ζητηθούν ώστε να κάνει αίτηση για κάρτα σίτισης. Προφανώς, για να εκτελέσει την τελευταία χρειάζεται να υπάρχουν στο προφίλ του όλα τα απαραίτητα στοιχεία. Έτσι σκοπός του μηχανισμού είναι, να παρουσιάσει στο χρήστη όλα τα προαπαιτούμενα στοιχεία που θα του ζητηθούν καθώς και τις εξόδους άλλων gadgets που χρησιμοποιούνται για την αίτηση της κάρτας σίτισης. Με το μηχανισμό πρότασης και σύνθεσης gadget, του δίνεται η δυνατότητα να αναζητήσει τη λειτουργικότητα που επιθυμεί. Ανάλογα την επιλογή του χρήστη δημιουργούνται και συντίθενται δυναμικά τα απαραίτητα για αυτή την επιλογή gadgets.

Αναφορικά, για το συγκεκριμένο σενάριο χρήσης, τα τέσσερα αρχικά gadgets που χρειάζεται να εκτελέσει ο χρήστης είναι:

Προαπαιτούμενα gadgets

- Η Φορολογική ενημερότητα
- Το Πιστοποιητικό οικογενειακής κατάστασης
- Η Βεβαίωση Ομοσπονδίας Πολυτέκνων
- Η Βεβαίωση σπουδών αδελφού-ής

Τα παραπάνω απαιτούν κύρια πληροφορία από το χρήστη, όπως το όνομα, το επώνυμο, τον κωδικός πρόσβασής του, τον προσωπικό αναγνωριστικό κωδικό, τον αναγνωριστικό κωδικό οικονομικών του, τον αναγνωριστικό κωδικό πολυτέκνων, το τμήμα φοίτησης του αδελφού-ής, τον κωδικός πρόσβασης του αδελφού-ής. Από τη στιγμή που ο χρήστης συμπληρώνει τα απαιτούμενα πρωτεύοντα στοιχεία στη συνέχεια εκτελεί το κάθε gadget. Στην περίπτωση που όλες οι εφαρμογές εκτελεστούν ομαλά, δηλαδή χωρίς να προκύψει κάποιο σφάλμα, τότε ο χρήστης ειδοποιείται για την πορεία των αιτημάτων του.

Στη συνέχεια αφού διαθέτει πλέον τα στοιχεία που του προσέφερε η έξοδος, δηλαδή του κάθε gadget, μπορεί να εκτελέσει το τελικό gadget που διαθέτει την υπηρεσία που επιθυμεί, δηλαδή την αίτηση για κάρτα σίτισης.

Τελικό gadget

- Αίτηση Κάρτα Σίτισης

Το τελικό αυτό gadget προϋποθέτει ως εισόδους τα αποτελέσματα των εφαρμογών που έχει ήδη εκτελέσει, δηλαδή τα εξής: αντίγραφο φορολογικής ενημερότητας, αντίγραφο πιστοποιητικού οικογενειακής κατάστασης, αντίγραφο βεβαίωσης πολυτέκνων, αντίγραφο βεβαίωσης σπουδών αδελφού-ής.

Τέλος, η απαίτηση που έχει δώσει ο χρήστης στο σύστημα κωδικοποιείται σε πρόβλημα planning προς επίλυση. Προκειμένου να ολοκληρωθεί η αναζήτηση του χρήστη προωθείται σε

εξωτερικά συστήματα, τα οποία επιστρέφουν την απάντηση στο χρήστη σε φιλική προς αυτόν αναπαράσταση. Η τελική απαίτηση του χρήστη και έξοδος του συστήματος είναι ο αριθμός της κάρτας σίτισης του φοιτητή.

4.1.1.4. Παρουσίαση αρχείων PDDL και SOL

Στο κεφάλαιο αυτό παρουσιάζονται τα αρχεία `gadgetProblem.pddl` και `plan_gadgetProblem.pddl_1.SOL` που δημιουργούνται δυναμικά ανάλογα την επιλογή του χρήστη από το γραφικό περιβάλλον, δηλαδή την αίτηση για κάρτα σίτισης. Στο κεφάλαιο 3 παρουσιάστηκε το `gadgetDomain.pddl`, το οποίο παραμένει κοινό για όλα τα σενάρια χρήσης αφού περιέχει όλο το πεδίο ορισμού, δηλαδή όλα τα διαθέσιμα gadgets με τα δεδομένα εισόδου και εξόδου.

Αρχικά, όπως φαίνεται στην παρακάτω Εικόνα 4.6 παρουσιάζεται το αρχείο που είναι απαραίτητο για τον ορισμό του προβλήματος. Στο πρώτο μπλοκ ορίζονται όλα τα δεδομένα εισόδου που χαρακτηρίζονται ως κύρια πληροφορία, δηλαδή δεν παράγονται από την εκτέλεση κάποιας άλλης εφαρμογής και μπορεί μόνο ο χρήστης να τα εισάγει. Στη συνέχεια, στο μπλοκ με την ετικέτα `goal` φαίνεται η επιλογή του χρήστη. Συγκεκριμένα, έχει επιλέξει ότι χρειάζεται για τελική λειτουργικότητα την έκδοση της κάρτας σίτισης.

```

(define (problem gadgetproblem)
  (:domain gadgets)
  (:objects concept)
  (:init (block concept)
  (Name concept)
  (Lastname concept)
  (Library_Username concept)
  (RN concept)
  (Identification_Card_Number concept)
  (Social_Security_registration_number concept)
  (Financial_Id concept)
  (Personal_Id concept)
  (Aspe_Id concept)
  (Sibling_Study_Department concept)
  (Sibling_Password concept)
  (Password concept)
  )
  (:goal (Meal_Card_Number concept)
  ))

```

Εικόνα 4.6 Αρχείο problem.pddl για αίτηση κάρτας σίτισης

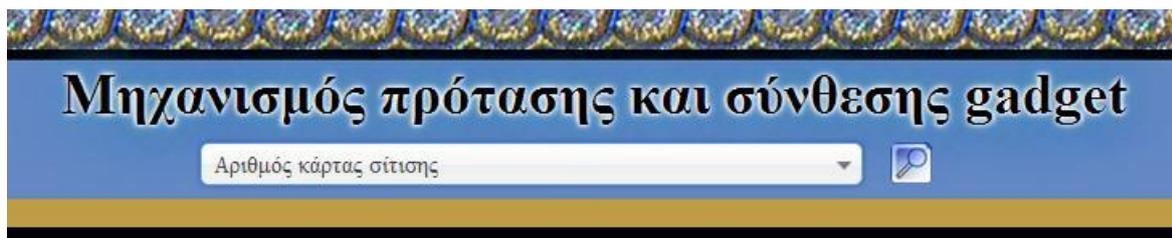
Το παρακάτω αρχείο που φαίνεται στην Εικόνα 4.7 είναι αυτό που παράγει το σύστημα σχεδιασμού (planner). Στο συγκεκριμένο σενάριο χρήσης φαίνεται ότι το σύστημα έχει εξάγει ότι για την αίτηση για κάρτα σίτισης ο χρήστης χρειάζεται να εκτελέσει πρώτα τα τέσσερα αρχικά gadgets, δηλαδή το ‘Φορολογική ενημερότητα’, το ‘Πιστοποιητικό οικογενειακής κατάστασης’, το ‘Βεβαίωση Ομοσπονδίας Πολυτέκνων’ και τέλος το ‘Βεβαίωση σπουδών αδελφού-ής’. Ο αριθμός που παρατηρείται στα αριστερά του κάθε gadget αφορά στη σειρά που πρέπει να εκτελεστούν τα gadgets, ώστε να μπορεί να εκτελεστεί αυτό με την τελική απαιτούμενη λειτουργικότητα. Ο αριθμός αυτός χρησιμοποιείται επίσης και στη γραφική απεικόνιση των gadget στο περιβάλλον του χρήστη. Σύμφωνα με αυτόν, τα στοιχεία τοποθετούνται δυναμικά στη σελίδα του χρήστη.

```
0: (REQUEST_STUDY_CERTIFICATION_SIBLING CONCEPT) [1]
0: (REQUEST_FINANCIAL_STATUS CONCEPT) [1]
0: (REQUEST_FAMILY_STATUS CONCEPT) [1]
0: (REQUEST_ASPE_CERTIFICATION CONCEPT) [1]
1: (APPLY_FOR_MEAL_CARD CONCEPT) [1]
```

Εικόνα 4.7 Αρχείο SOL για αίτηση κάρτας σίτισης

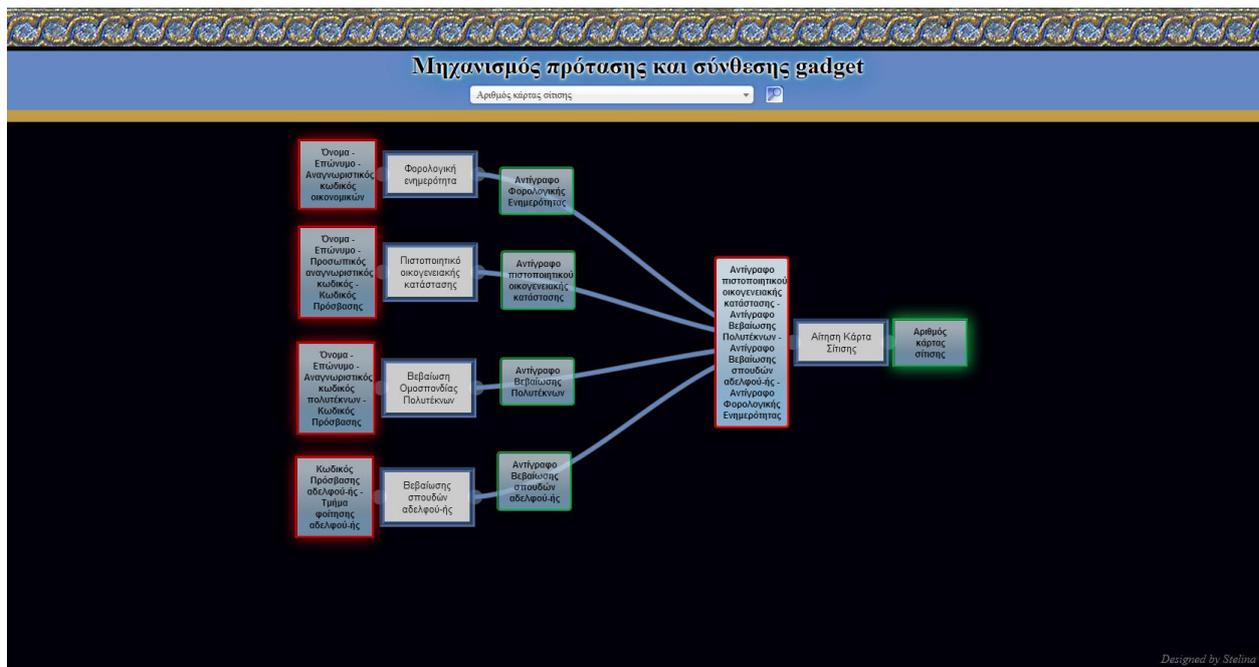
4.1.1.5. Screenshots

Στη συγκεκριμένη ενότητα παρουσιάζεται ένα σενάριο χρήσης για πρόταση gadgets έχοντας ως τελική λειτουργικότητα την αίτηση για κάρτα σίτισης. Ο χρήστης επιλέγει τη λειτουργικότητα αριθμός κάρτα σίτισης, όπως φαίνεται στην παρακάτω Εικόνα 4.8.



Εικόνα 4.8 Αναζήτηση για αίτηση κάρτας σίτισης

Αφού επιλέξει αναζήτηση, του παρουσιάζονται τα gadgets που είναι απαραίτητα για να αποκτήσει κάρτα σίτισης όπως παρατίθεται στην εικόνα Εικόνα 4.9 παρακάτω. Εκτός από τα gadgets φαίνονται αναλυτικά τα δεδομένα εισόδου και εξόδου του καθενός.



Εικόνα 4.9 Πρόταση απαιτούμενων gadgets για αίτηση κάρτας σίτισης

4.1.2. Αίτηση αποφοίτησης Unity

4.1.2.1. Περιγραφή του προβλήματος

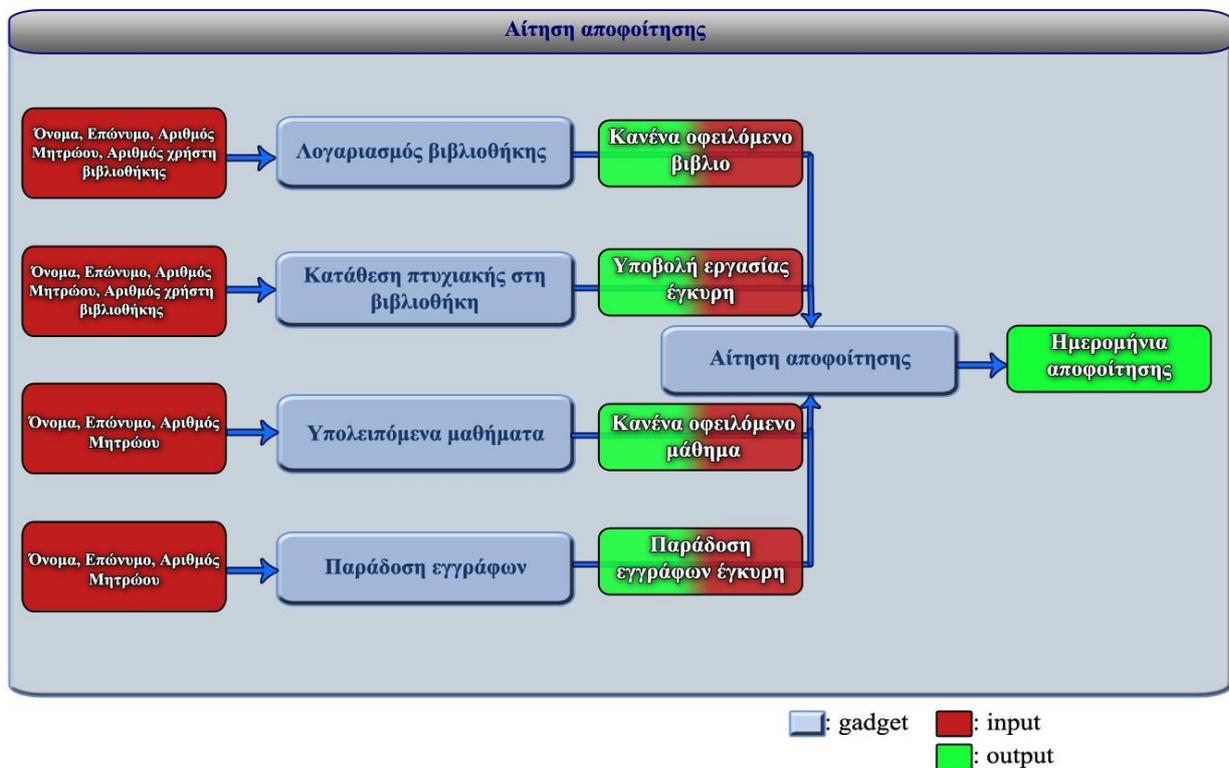
Η αίτηση για αποφοίτηση αποτελεί ένα gadget το οποίο μπορεί να εκτελέσει ο φοιτητής προκειμένου να υποβάλλει στο τμήμα της σχολής του αίτηση για αποφοίτηση. Η συγκεκριμένη διαδικασία περιλαμβάνει μια περίπλοκη και χρονοβόρα διαδικασία στην οποία συμμετέχουν όσοι χρήστες έχουν το ρόλο του φοιτητή μέσα στο δίκτυο. Τελική απαίτηση και λειτουργικότητα που θα λάβει ο φοιτητής μετά την εκτέλεση της υπηρεσίας, εφόσον πληροί όλες τις προϋποθέσεις, είναι η ημερομηνία ορκωμοσίας του. Για να μπορέσει να ολοκληρωθεί το αίτημα του φοιτητή χρειάζεται να μην έχει εκκρεμότητες με τη βιβλιοθήκη του Πανεπιστημίου και να έχει καταθέσει κάποια έγγραφα στη γραμματεία του Τμήματος. Αναλυτικότερα, ο φοιτητής χρειάζεται:

- Να μην οφείλει κανένα βιβλίο στη Βιβλιοθήκη.

- Να έχει παραδώσει την πτυχιακή του σε ηλεκτρονική μορφή στο σύστημα της Βιβλιοθήκης.
- Να έχει παραδώσει το πρακτικό παρουσίασης της πτυχιακής εργασίας στη γραμματεία.
- Να μην οφείλει κανένα μάθημα.
- Να έχει παραδώσει το τρίπτυχο και το πάσο του.

4.1.2.2. Περιγραφή απαιτούμενων gadgets

Ο φοιτητής χρειάζεται να έχει εγκατεστημένα τα απαιτούμενα gadgets για να μπορέσει να επικοινωνήσει με gadgets άλλων ρόλων, όπως της Γραμματείας ή της Βιβλιοθήκης, ή ακόμα και με άλλες εξωτερικές υπηρεσίες ώστε να μπορέσει να εξυπηρετηθεί. Στην παρακάτω Εικόνα 4.10 παρουσιάζονται αναλυτικά τα gadgets που απαιτούνται ώστε να λάβει ο φοιτητής την ημερομηνία ορκωμοσίας του. Εκτός από τα gadgets που χρειάζεται να εκτελεστούν παρατίθενται τόσα γραφικά όσο και περιγραφικά τα δεδομένα εισόδου και εξόδου του κάθε gadget.



1. Το **‘Λογαριασμός Βιβλιοθήκης (Library Book Account)’**, ελέγχει σύμφωνα με τα στοιχεία του φοιτητή που το εκτελεί τον αριθμό των βιβλίων που χρωστάει. Τα δεδομένα εισόδου αυτού του gadget είναι τα προσωπικά στοιχεία του φοιτητή, όπως το Όνομα, το Επώνυμο, ο Αριθμός Μητρώου και ο αριθμός χρήστη βιβλιοθήκης. Ως δεδομένα εξόδου δίνει τον αριθμό των χρωστούμενων βιβλίων.
2. Το **‘Κατάθεση πτυχιακής στη βιβλιοθήκη (Library Thesis Submission)’**, το οποίο σύμφωνα με τα στοιχεία του ελέγχει αν ο φοιτητής έχει παραδώσει την εργασία του ηλεκτρονικά. Τα δεδομένα εισόδου αυτού του gadget είναι τα προσωπικά στοιχεία του φοιτητή, όπως το Όνομα, το Επώνυμο, ο Αριθμός Μητρώου και ο αριθμός χρήστη βιβλιοθήκης. Ως δεδομένα εξόδου επιστρέφει ένα μήνυμα αν η υποβολή της εργασίας του είναι έγκυρη ή όχι.
3. Το **‘Υπολειπόμενα μαθήματα (Remaining Courses)’**, το οποίο ελέγχει τον αριθμό των υπολειπόμενων χρωστούμενων μαθημάτων. Τα δεδομένα εισόδου αυτού του gadget είναι τα προσωπικά στοιχεία του φοιτητή, όπως το Όνομα, το Επώνυμο και ο Αριθμός Μητρώου. Ως δεδομένα εξόδου επιστρέφει τον αριθμό των χρωστούμενων μαθημάτων.
4. Το **‘Εγγραφα αποφοίτησης (Graduation Documents)’**, το οποίο ελέγχει αν ο φοιτητής έχει παραδώσει το πρακτικό παρουσίασης της πτυχιακής του, το τρίπτυχό του και τέλος αν έχει παραδώσει το πάσο του. Τα δεδομένα εισόδου αυτού του gadget είναι τα προσωπικά στοιχεία του φοιτητή, όπως το Όνομα, το Επώνυμο και ο Αριθμός Μητρώου. Ως δεδομένα εξόδου ένα μήνυμα για το αν έχει παραδώσει το κάθε ένα από τα απαιτούμενα έγγραφα.
5. Το **‘Αίτηση αποφοίτησης (Graduation Apply)’**, το οποίο ελέγχει αν όλα τα προηγούμενα δεδομένα συμφωνούν ώστε να προχωρήσει η αίτηση του φοιτητή για ορκωμοσία. Τα δεδομένα εισόδου αυτού του gadget είναι οι έξοδοι των gadget στα βήματα 1, 2, 3, 4, δηλαδή κανένα οφειλόμενο βιβλίο, υποβολή πτυχιακής έγκυρη, κανένα οφειλόμενο μάθημα και παράδοση εγγράφων έγκυρη αντίστοιχα. Ως δεδομένα εξόδου επιστρέφει την ημερομηνία ορκωμοσίας του φοιτητή.

4.1.2.3. Περιγραφή λύσης

Στην περίπτωση της αίτησης αποφοίτησης ο χρήστης, όπως έχει ήδη αναφερθεί, αναζητά το gadget ή το σύνολο των gadgets που χρειάζονται για να λάβει την ημερομηνία ορκωμοσίας του. Ο χρήστης δεν μπορεί να γνωρίζει εκ των προτέρων όλα τις εφαρμογές και τα δεδομένα που θα του ζητηθούν ώστε να εκτελέσει την αίτηση για αποφοίτηση. Προφανώς, για να εκτελέσει την τελευταία χρειάζεται να υπάρχουν στο προφίλ του όλα τα απαραίτητα στοιχεία. Έτσι σκοπός του μηχανισμού είναι, να παρουσιάσει στο χρήστη όλα τα προαπαιτούμενα στοιχεία που θα του ζητηθούν καθώς και τις εξόδους άλλων gadgets που χρησιμοποιούνται για την αίτηση της αποφοίτησης. Σκοπός είναι, ο χρήστης να γνωρίζει εκ των προτέρων όλα τα προαπαιτούμενα στοιχεία και gadgets που θα του ζητηθούν, ώστε να κάνει αίτηση για την αποφοίτησή του. Με το μηχανισμό πρότασης και σύνθεσης gadget, του δίνεται η δυνατότητα να αναζητήσει τη λειτουργικότητα που επιθυμεί. Ανάλογα την επιλογή του χρήστη δημιουργούνται και συντίθενται δυναμικά τα απαραίτητα για αυτή την επιλογή gadgets. Αναφορικά, για το συγκεκριμένο σενάριο χρήσης, τα τέσσερα αρχικά gadgets που χρειάζεται να εκτελέσει ο χρήστης είναι:

Προαπαιτούμενα gadgets

- Ο Λογαριασμός Βιβλιοθήκης
- Η Κατάθεση πτυχιακής στη βιβλιοθήκη
- Τα Υπολειπόμενα μαθήματα
- Τα Έγγραφα αποφοίτησης

Τα παραπάνω απαιτούν κύρια πληροφορία από το χρήστη, όπως το όνομα, το επώνυμο, ο αριθμός μητρώου και αριθμός χρήστη βιβλιοθήκης. Από τη στιγμή που ο χρήστης συμπληρώνει τα απαιτούμενα πρωτεύοντα στοιχεία στη συνέχεια εκτελεί το κάθε gadget. Στην περίπτωση που όλες οι εφαρμογές εκτελεστούν ομαλά, δηλαδή χωρίς να προκύψει κάποιο σφάλμα, τότε ο χρήστης ειδοποιείται για την πορεία των αιτημάτων του.

Στη συνέχεια αφού διαθέτει πλέον τα στοιχεία που του προσέφερε η έξοδος του κάθε gadget, μπορεί να εκτελέσει το τελικό gadget που διαθέτει την υπηρεσία που επιθυμεί, δηλαδή την αίτηση αποφοίτησης.

Τελικό gadget

- Αίτηση αποφοίτησης

Το τελικό αυτό gadget προϋποθέτει ως εισόδους τα αποτελέσματα των εφαρμογών που έχει ήδη εκτελέσει, δηλαδή τα εξής: κανένα οφειλόμενο βιβλίο, υποβολή πτυχιακής έγκυρη, κανένα οφειλόμενο μάθημα και παράδοση εγγράφων έγκυρη.

Τέλος, η απαίτηση που έχει δώσει ο χρήστης στο σύστημα κωδικοποιείται σε πρόβλημα planning προς επίλυση. Προκειμένου να ολοκληρωθεί η αναζήτηση του χρήστη προωθείται σε εξωτερικά συστήματα, τα οποία επιστρέφουν την απάντηση στο χρήστη σε φιλική προς αυτόν αναπαράσταση. Η τελική απαίτηση του χρήστη και έξοδος του συστήματος είναι η ημερομηνία της ορκωμοσίας του φοιτητή.

4.1.2.4. Παρουσίαση αρχείων PDDL και SOL

Στο κεφάλαιο αυτό παρουσιάζονται τα αρχεία gadgetProblem.pddl και plan_gadgetProblem.pddl_1.SOL που δημιουργούνται δυναμικά ανάλογα την επιλογή του χρήστη από το γραφικό περιβάλλον, δηλαδή την αίτηση για αποφοίτηση. Στο κεφάλαιο 3 παρουσιάστηκε το gadgetDomain.pddl, το οποίο παραμένει κοινό για όλα τα σενάρια χρήσης αφού περιέχει όλο το πεδίο ορισμού, δηλαδή όλα τα διαθέσιμα gadgets με τα δεδομένα εισόδου και εξόδου.

Αρχικά, όπως φαίνεται στην παρακάτω Εικόνα 4.11 παρουσιάζεται το αρχείο που είναι απαραίτητο για τον ορισμό του προβλήματος. Στο πρώτο μπλοκ ορίζονται όλα τα δεδομένα εισόδου που χαρακτηρίζονται ως κύρια πληροφορία, δηλαδή δεν παράγονται από την εκτέλεση κάποιας άλλης εφαρμογής και μπορεί μόνο ο χρήστης να τα εισάγει. Στη συνέχεια, στο μπλοκ με την ετικέτα goal φαίνεται η επιλογή του χρήστη. Συγκεκριμένα, έχει επιλέξει ότι χρειάζεται για τελική λειτουργικότητα την ημερομηνία αποφοίτησης.

```

(define (problem gadgetproblem)
  (:domain gadgets)
  (:objects concept)
  (:init (block concept)
  (Name concept)
  (Lastname concept)
  (Library_Username concept)
  (RN concept)
  (Identification_Card_Number concept)
  (Social_Security_registration_number concept)
  (Financial_Id concept)
  (Personal_Id concept)
  (Aspe_Id concept)
  (Sibling_Study_Department concept)
  (Sibling_Password concept)
  (Password concept)
  )
  (:goal (Date_Graduation_Apply concept)
  ))

```

Εικόνα 4.11 Αρχείο problem.pddl για αίτηση αποφοίτησης

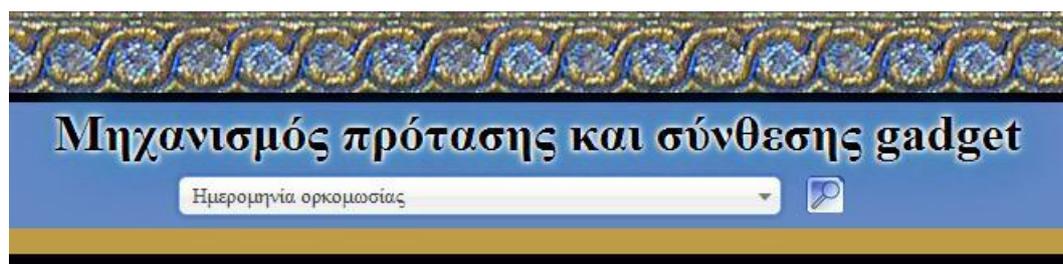
Το παρακάτω αρχείο που φαίνεται στην Εικόνα 4.12 είναι αυτό που παράγει το σύστημα σχεδιασμού (planner). Στο συγκεκριμένο σενάριο χρήσης φαίνεται ότι το σύστημα έχει εξάγει ότι για την ημερομηνία ορκωμοσίας ο χρήστης χρειάζεται να εκτελέσει πρώτα τα τέσσερα αρχικά gadgets, δηλαδή το ‘Λογαριασμός Βιβλιοθήκης’, το ‘Κατάθεση πτυχιακής στη βιβλιοθήκη’, το ‘Υπολειπόμενα μαθήματα’ και το ‘Εγγραφα αποφοίτησης’. Ο αριθμός που παρατηρείται στα αριστερά του κάθε gadget αφορά στη σειρά που πρέπει να εκτελεστούν τα gadgets, ώστε να μπορεί να εκτελεστεί αυτό με την τελική απαιτούμενη λειτουργικότητα. Ο αριθμός αυτός χρησιμοποιείται επίσης και στη γραφική απεικόνιση των gadget στο περιβάλλον του χρήστη. Σύμφωνα με αυτόν, τα στοιχεία τοποθετούνται δυναμικά στη σελίδα του χρήστη.

```
0: (REMAINING_COURSES CONCEPT) [1]
0: (LIBRARY_BOOK_ACCOUNT_STUDENT CONCEPT) [1]
0: (LIBRARY_THESIS_SUBMISSION CONCEPT) [1]
0: (GRADUATION_DOCUMENTS CONCEPT) [1]
1: (GRADUATION_APPLY CONCEPT) [1]
```

Εικόνα 4.12 Αρχείο SOL για αίτηση αποφοίτησης

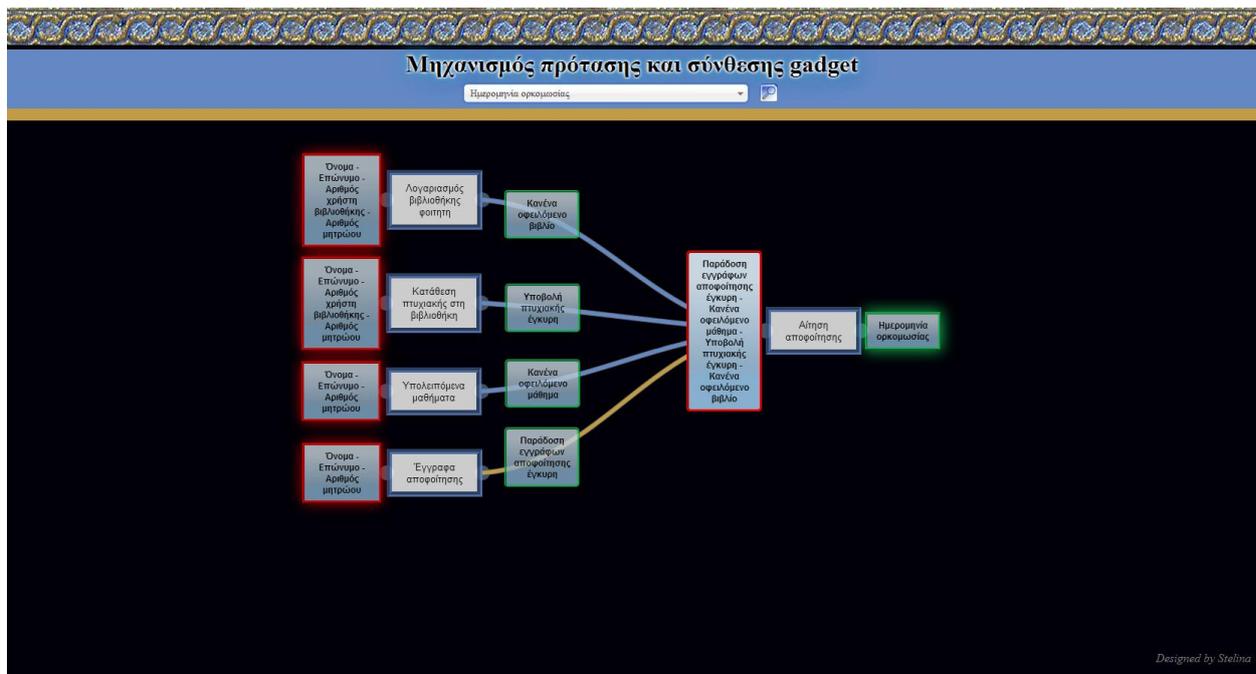
4.1.2.5. Screenshots

Στη συγκεκριμένη ενότητα παρουσιάζεται ένα σενάριο χρήσης για πρόταση gadgets έχοντας ως τελική λειτουργικότητα την αίτηση αποφοίτησης. Ο χρήστης επιλέγει τη λειτουργικότητα ημερομηνία ορκωμοσίας, όπως φαίνεται στην παρακάτω Εικόνα 4.13.



Εικόνα 4.13 Αναζήτηση για αίτηση αποφοίτησης

Αφού επιλέξει αναζήτηση, του παρουσιάζονται τα gadgets που είναι απαραίτητα για να κάνει αίτηση για αποφοίτηση, όπως παρατίθεται στην Εικόνα 4.14 παρακάτω. Εκτός από τα gadgets φαίνονται αναλυτικά τα δεδομένα εισόδου και εξόδου του καθενός.



Εικόνα 4.14 Πρόταση απαιτούμενων gadgets για αίτηση αποφοίτησης

4.2. Εξωτερικό περιβάλλον

Όπως έχει ήδη αναφερθεί ο συγκεκριμένος μηχανισμός μπορεί να εφαρμοστεί σε οποιοδήποτε κοινωνικό δίκτυο ή άλλο εξωτερικό περιβάλλον, στο οποίο πολλές υπηρεσίες μπορούν να συνδυαστούν και να συνθέσουν κάποια πιο πολύπλοκη δραστηριότητα. Μία τέτοια ανάγκη αναπτύχθηκε στα πλαίσια της Ηλεκτρονικής Διακυβέρνησης, που έχει ως στόχο τον εκσυγχρονισμό της Δημόσιας Διοίκησης.

4.2.1. Εργοδοτικές Εισφορές (ΙΚΑ)

Σύμφωνα με τη νομοθεσία του ΙΚΑ-ΕΤΑΜ οι εργοδότες είναι υπεύθυνοι για την καταβολή όχι μόνο των εισφορών που τους βαρύνουν (εργοδοτικές εισφορές), αλλά και των εισφορών που βαρύνουν τους μισθωτούς τους (εργατικές εισφορές).

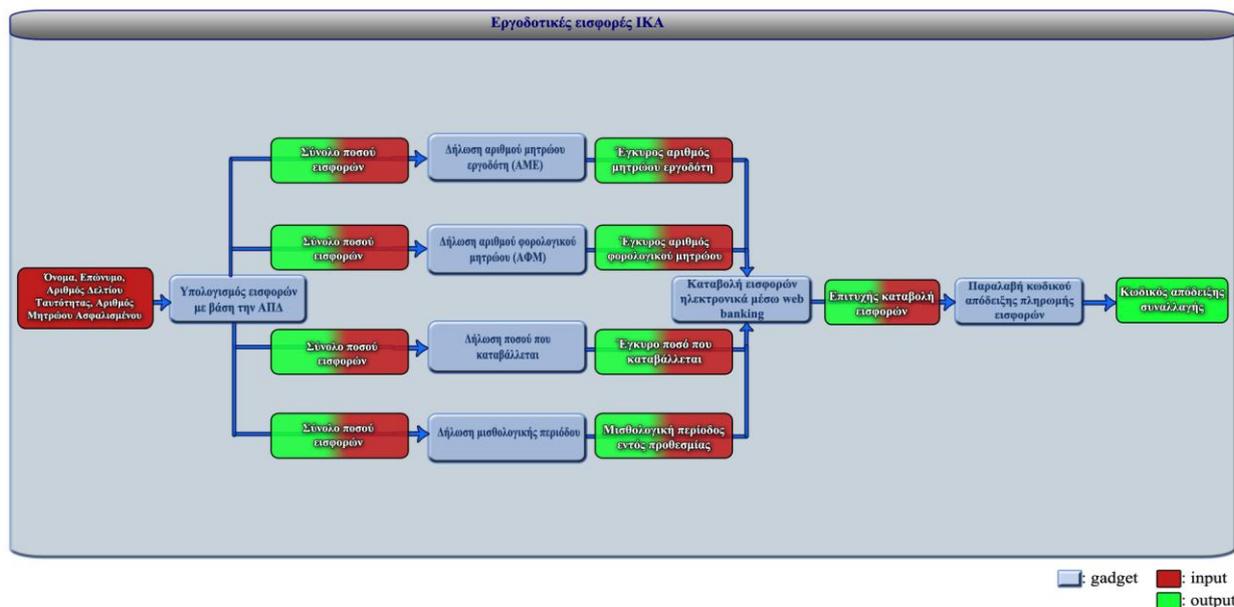
4.2.1.1. Περιγραφή του προβλήματος

Ο χρήστης χρειάζεται να έχει εγκατεστημένα τα απαιτούμενα gadgets για να μπορέσει να ολοκληρώσει το αίτημά του. Συγκεκριμένα τα gadgets αυτά είναι:

1. Η επιχείρηση υπολογίζει τις εισφορές με βάση την ΑΠΔ που έχει καταθέσει σε κάποιο Υποκατάστημα ή Παράρτημα του ΙΚΑ-ΕΤΑΜ.
2. Η επιχείρηση δηλώνει στον αρμόδιο οργανισμό που διενεργείται η πληρωμή (Υποκατάστημα ή Παράρτημα του ΙΚΑ-ΕΤΑΜ ή πιστωτικό ίδρυμα ή άλλο νομικό πρόσωπο που έχει συνάψει σύμβαση με το ΙΚΑ-ΕΤΑΜ για τη διενέργεια εισπράξεων και πληρωμών): Αριθμό Μητρώου Εργοδότη (ΑΜΕ), Αριθμό Φορολογικού Μητρώου (ΑΦΜ), το ποσό που καταβάλλεται και τη μισθολογική περίοδο, στην οποία αναφέρεται η καταβολή της εργοδοτικής εισφοράς.
3. Η επιχείρηση καταβάλλει τις εισφορές στον οργανισμό στον οποίο έχει απευθυνθεί (Υποκατάστημα ή Παράρτημα του ΙΚΑ-ΕΤΑΜ ή στο πιστωτικό ίδρυμα, ή σε άλλο νομικό πρόσωπο που έχει συνάψει σύμβαση με το ΙΚΑ-ΕΤΑΜ για τη διενέργεια εισπράξεων και πληρωμών) μέχρι την τελευταία εργάσιμη, για τις δημόσιες υπηρεσίες, του επόμενου μήνα από τον μήνα στον οποίο εντός του οποίου παρασχέθηκε η εργασία ή η υπηρεσία (φυσική παρουσία). Σημειώνεται ότι αν ο εργοδότης δεν έχει υποβάλλει στο ΙΚΑ-ΕΤΑΜ το Έντυπο Βεβαίωσης Στοιχείων Εργοδότη και δεν έχει λάβει δεκαημέριο Αριθμό Μητρώου Εργοδότη ή αν η καταβολή των εισφορών είναι εκπρόθεσμη, τότε πρέπει να απευθυνθεί αποκλειστικά στο Υποκατάστημα ή Παράρτημα του ΙΚΑ-ΕΤΑΜ, όπου υποβάλλεται η ΑΠΔ. Στα Υποκαταστήματα ΙΚΑ-ΕΤΑΜ καταβάλλουν υποχρεωτικά τις εισφορές τους οι Εργοδότες των οικοδομοτεχνικών έργων (ιδιωτικών και δημόσιων).
4. Τη χρονική στιγμή που πληρώνει, η επιχείρηση λαμβάνει απόδειξη πληρωμής για τις εισφορές από τον οργανισμό στον οποίο έχει απευθυνθεί (Υποκατάστημα ή Παράρτημα του ΙΚΑ-ΕΤΑΜ ή στο πιστωτικό ίδρυμα, ή σε άλλο νομικό πρόσωπο που έχει συνάψει σύμβαση με το ΙΚΑ-ΕΤΑΜ για τη διενέργεια εισπράξεων και πληρωμών).

4.2.1.2. Περιγραφή απαιτούμενων gadgets

Ο χρήστης χρειάζεται να έχει όλες τις απαραίτητες πληροφορίες ώστε να μπορέσει να ολοκληρώσει την αίτησή του. Στην παρακάτω Εικόνα 4.15 παρουσιάζονται αναλυτικά τα gadgets που απαιτούνται ώστε να λάβει ο χρήστης τον κωδικό απόδειξης της συναλλαγής. Εκτός από τα gadgets που χρειάζεται να εκτελεστούν παρατίθενται τόσα γραφικά όσο και περιγραφικά τα δεδομένα εισόδου και εξόδου του κάθε gadget.



Εικόνα 4.15 Gadgets και δεδομένα εισόδου/εξόδου για παραλαβή κωδικού απόδειξης από την καταβολή εργοδοτικών εισφορών στο ΙΚΑ

1. Το ‘Υπολογισμός εισφορών με βάση την ΑΠΔ¹ (Calculate Social Security Tax Amount)’ , επιτρέπει στην επιχείρηση να υπολογίζει τις εισφορές με βάση την ΑΠΔ που έχει καταθέσει ηλεκτρονικά. Τα δεδομένα εισόδου αυτού του gadget είναι τα προσωπικά στοιχεία του εργοδότη της επιχείρησης, όπως το Όνομα, το Επώνυμο, ο Αριθμός Δελτίου Ταυτότητας και ο Αριθμός Μητρώου Ασφαλισμένου. Ως δεδομένα εξόδου δίνει το σύνολο του ποσού των εισφορών.

2. Το **‘Δήλωση αριθμού μητρώου εργοδότη (AME) (State Insurance Number)’**, επιτρέπει στον εργοδότη να καταθέσει ηλεκτρονικά τον αριθμό μητρώου του. Τα δεδομένα εισόδου αυτού του gadget είναι το σύνολο του ποσού των εισφορών που έχει υπολογιστεί από το βήμα 1. Ως δεδομένα εξόδου επιστρέφει τον αριθμό μητρώου που έχει κατατεθεί στην περίπτωση που είναι έγκυρος.
3. Το **‘Δήλωση αριθμού φορολογικού μητρώου (ΑΦΜ) (State Social Security Number)’**, επιτρέπει στον εργοδότη να καταθέσει ηλεκτρονικά τον αριθμό φορολογικού μητρώου του. Τα δεδομένα εισόδου αυτού του gadget είναι το σύνολο του ποσού των εισφορών που έχει υπολογιστεί από το βήμα 1. Ως δεδομένα επιστρέφει τον αριθμό φορολογικού μητρώου που έχει κατατεθεί στην περίπτωση που είναι έγκυρος.
4. Το **‘Δήλωση ποσού που καταβάλλεται (State SS Tax Amount)’**, επιτρέπει στον εργοδότη να καταθέσει ηλεκτρονικά το ποσό που καταβάλλει. Τα δεδομένα εισόδου αυτού του gadget είναι το σύνολο του ποσού των εισφορών που έχει υπολογιστεί από το βήμα 1. Ως δεδομένα εξόδου επιστρέφει ότι το ποσό που καταβλήθηκε είναι έγκυρο.
5. Το **‘Δήλωση μισθολογικής περιόδου (State Payment Period)’**, επιτρέπει στον εργοδότη να δηλώσει τη μισθολογική περίοδο, στην οποία αναφέρεται η καταβολή της εργοδοτικής εισφοράς. Τα δεδομένα εισόδου αυτού του gadget είναι το σύνολο του ποσού των εισφορών που έχει υπολογιστεί από το βήμα 1. Ως δεδομένα εξόδου δίνει την απάντηση ότι το η μισθολογική περίοδος είναι έγκυρη, δηλαδή εντός προθεσμίας.
6. Το **‘Καταβολή εισφορών ηλεκτρονικά μέσω web banking (Place Electronic Payment)’**, επιτρέπει στην επιχείρηση να καταβάλλει τις εισφορές ηλεκτρονικά. Τα δεδομένα εισόδου αυτού του gadget είναι προϋποθέτουν τον Αριθμό Μητρώου Εργοδότη (AME), που δηλώθηκε στο βήμα 2, τον Αριθμό Φορολογικού Μητρώου (ΑΦΜ), που δηλώθηκε στο βήμα 3, το ποσό που καταβλήθηκε στο βήμα 4 και τέλος την μισθολογική περίοδο που δηλώθηκε στο βήμα 5. Ως δεδομένα εξόδου επιστρέφει αν πραγματοποιήθηκε επιτυχώς ή όχι η καταβολή εισφορών.
7. Το **‘Παραλαβή κωδικού απόδειξης πληρωμής για τις εισφορές (Retrieve Payment Receipt)’**, επιτρέπει στον εργοδότη να παραλάβει έναν κωδικό απόδειξης πληρωμής των

εισφορών που κατέθεσε . Τα δεδομένα εισόδου αυτού του gadget είναι η επιτυχημένη καταβολή των εισφορών από το βήμα 6. Ως δεδομένα εξόδου παράγει τον κωδικό απόδειξης της συναλλαγής.

4.2.1.3. Περιγραφή λύσης

Στην περίπτωση της παραλαβής κωδικού απόδειξης πληρωμής για τις εισφορές ο χρήστης, όπως έχει ήδη αναφερθεί, αναζητά το gadget ή το σύνολο των gadgets που χρειάζονται για να λάβει τον κωδικό απόδειξης της συναλλαγής του. Σκοπός είναι, ο χρήστης να γνωρίζει εκ των προτέρων όλα τα προαπαιτούμενα στοιχεία και gadgets που θα του ζητηθούν, ώστε να κάνει αίτηση για να παραλάβει την απόδειξη πληρωμής του. Με το μηχανισμό πρότασης και σύνθεσης gadget, του δίνεται η δυνατότητα να αναζητήσει τη λειτουργικότητα που επιθυμεί. Ανάλογα την επιλογή του χρήστη δημιουργούνται και συντίθενται δυναμικά τα απαραίτητα για αυτή την επιλογή gadgets. Αναφορικά, για το συγκεκριμένο σενάριο χρήσης, το αρχικό gadget που χρειάζεται να εκτελέσει ο χρήστης είναι:

Προαπαιτούμενο gadget

- Υπολογισμός εισφορών με βάση την ΑΠΔ που έχει καταθέσει

Το παραπάνω απαιτεί κύρια πληροφορία από το χρήστη, όπως το όνομα, το επώνυμο, ο αριθμός δελτίου ταυτότητας και ο αριθμός μητρώου ασφαλισμένου. Από τη στιγμή που ο χρήστης συμπληρώνει τα απαιτούμενα πρωτεύοντα στοιχεία στη συνέχεια εκτελεί το gadget. Στη συνέχεια, για το τρέχων σενάριο χρήσης όλες οι υπόλοιπες πληροφορίες ώστε να επιτευχθεί η τελική λειτουργικότητα απαιτούν εξόδους άλλων υπηρεσιών. Αναφορικά, τα εν συνεχεία gadgets που πρέπει να εκτελεστούν είναι:

Προαπαιτούμενα gadgets

- Δήλωση αριθμού μητρώου εργοδότη (ΑΜΕ)
- Δήλωση αριθμού φορολογικού μητρώου (ΑΦΜ)

- Δήλωση ποσού που καταβάλλεται
- Δήλωση μισθολογικής περιόδου

Τα παραπάνω απαιτούν ως είσοδο την έξοδο που παράγει η εκτέλεση του πρώτου gadget, δηλαδή τον υπολογισμό ποσού εισφορών. Στην περίπτωση που όλες οι εφαρμογές εκτελεστούν ομαλά, δηλαδή χωρίς να προκύψει κάποιο σφάλμα, τότε ο χρήστης ειδοποιείται για την πορεία των αιτημάτων του. Η επόμενη εφαρμογή που χρειάζεται να εκτελεστεί είναι:

Προαπαιτούμενο gadget

- Καταβολή εισφορών ηλεκτρονικά μέσω web banking

Η παραπάνω εφαρμογή απαιτεί ως είσοδο τις εξόδους που παράγει η εκτέλεση των προηγούμενων gadgets, δηλαδή τον έγκυρο αριθμό μητρώου εργοδότη, τον έγκυρο αριθμό φορολογικού μητρώου, την έγκυρη δήλωση ποσού που καταβάλλεται και την έγκυρη δήλωση μισθολογικής περιόδου. Στη συνέχεια αφού διαθέτει πλέον τα στοιχεία που του προσέφερε η έξοδος του κάθε gadget, μπορεί να εκτελέσει το τελικό gadget που διαθέτει την υπηρεσία που επιθυμεί, δηλαδή την αίτηση αποφοίτησης.

Τελικό gadget

- Παραλαβή κωδικού απόδειξης πληρωμής για τις εισφορές

Το τελικό αυτό gadget προϋποθέτει ως είσοδο τα αποτελέσματα της εφαρμογής που έχει ήδη εκτελέσει, δηλαδή έγκυρη καταβολή εισφορών ηλεκτρονικά μέσω web banking.

Τέλος, η απαίτηση που έχει δώσει ο χρήστης στο σύστημα κωδικοποιείται σε πρόβλημα planning προς επίλυση. Προκειμένου να ολοκληρωθεί η αναζήτηση του χρήστη προωθείται σε εξωτερικά συστήματα, τα οποία επιστρέφουν την απάντηση στο χρήστη σε φιλική προς αυτόν αναπαράσταση. Η τελική απαίτηση του χρήστη και έξοδος του συστήματος είναι η παραλαβή κωδικού απόδειξης πληρωμής για τις εισφορές.

4.2.1.4. Παρουσίαση αρχείων PDDL και SOL

Στο κεφάλαιο αυτό παρουσιάζονται τα αρχεία `gadgetProblem.pddl` και `plan_gadgetProblem.pddl_1.SOL` που δημιουργούνται δυναμικά ανάλογα την επιλογή του χρήστη από το γραφικό περιβάλλον, δηλαδή την παραλαβή κωδικού απόδειξης πληρωμής για τις εισφορές. Στο κεφάλαιο 3 παρουσιάστηκε το `gadgetDomain.pddl`, το οποίο παραμένει κοινό για όλα τα σενάρια χρήσης αφού περιέχει όλο το πεδίο ορισμού, δηλαδή όλα τα διαθέσιμα gadgets με τα δεδομένα εισόδου και εξόδου.

Αρχικά, όπως φαίνεται στην παρακάτω Εικόνα 4.16 παρουσιάζεται το αρχείο που είναι απαραίτητο για τον ορισμό του προβλήματος. Στο πρώτο μπλοκ ορίζονται όλα τα δεδομένα εισόδου που χαρακτηρίζονται ως κύρια πληροφορία, δηλαδή δεν παράγονται από την εκτέλεση κάποιας άλλης εφαρμογής και μπορεί μόνο ο χρήστης να τα εισάγει. Στη συνέχεια, στο μπλοκ με την ετικέτα `goal` φαίνεται η επιλογή του χρήστη. Συγκεκριμένα, έχει επιλέξει ότι χρειάζεται για τελική λειτουργικότητα την παραλαβή κωδικού απόδειξης πληρωμής για τις εισφορές..

```

(define (problem gadgetproblem)
  (:domain gadgets)
  (:objects concept)
  (:init (block concept)
    (Name concept)
    (Lastname concept)
    (Library_Username concept)
    (RN concept)
    (Identification_Card_Number concept)
    (Social_Security_registration_number concept)
    (Financial_Id concept)
    (Personal_Id concept)
    (Aspe_Id concept)
    (Sibling_Study_Department concept)
    (Sibling_Password concept)
    (Password concept)
  )
  (:goal (Code_Retrieval_Payment_Receipt concept)
  ))

```

Εικόνα 4.16 Αρχείο problem.pddl για παραλαβή κωδικού απόδειξης από την καταβολή εργοδοτικών εισφορών στο ΙΚΑ

Το παρακάτω αρχείο που φαίνεται στην Εικόνα 4.17 είναι αυτό που παράγει το σύστημα σχεδιασμού (planner). Στο συγκεκριμένο σενάριο χρήσης φαίνεται ότι το σύστημα έχει εξάγει ότι για την παραλαβή κωδικού απόδειξης πληρωμής για τις εισφορές ο χρήστης χρειάζεται να εκτελέσει πρώτο το αρχικό gadget, δηλαδή το ‘Υπολογισμός εισφορών με βάση την ΑΠΔ που έχει καταθέσει’. Στη συνέχεια, ο planner έχει εξάγει ότι τα επόμενα προς εκτέλεση gadgets είναι το ‘Δήλωση αριθμού μητρώου εργοδότη (ΑΜΕ)’, το ‘Δήλωση αριθμού φορολογικού μητρώου (ΑΦΜ)’, το ‘Δήλωση ποσού που καταβάλλεται’ και το ‘Δήλωση μισθολογικής περιόδου’. Το επόμενο gadget που χρειάζεται να εκτελέσει ο χρήστης σύμφωνα με τα αποτελέσματα του planner είναι το ‘Καταβολή εισφορών ηλεκτρονικά μέσω web banking’. Τέλος, στο τελευταίο επίπεδο του planner βρίσκεται το gadget που θα παράγει την τελική απαίτηση του χρήστη, δηλαδή την παραλαβή κωδικού απόδειξης. Ο αριθμός που παρατηρείται στα αριστερά του κάθε gadget αφορά στη σειρά που πρέπει να εκτελεστούν τα gadgets, ώστε να μπορεί να εκτελεστεί αυτό με την τελική απαιτούμενη λειτουργικότητα. Ο αριθμός αυτός χρησιμοποιείται επίσης και

στη γραφική απεικόνιση των gadget στο περιβάλλον του χρήστη. Σύμφωνα με αυτόν, τα στοιχεία τοποθετούνται δυναμικά στη σελίδα του χρήστη.

```
0: (CALCULATE_CONTRIBUTION CONCEPT) [1]
1: (STATEMENT_WAGE_PERIOD CONCEPT) [1]
1: (STATEMENT_INSURANCE_NUMBER CONCEPT) [1]
1: (STATEMENT_AMOUNT_PAID CONCEPT) [1]
1: (STATEMENT_TAXPAYER_IDENTIFICATION_NUMBER CONCEPT) [1]
2: (ELECTRONIC_CONTRIBUTION CONCEPT) [1]
3: (CODE_RETRIEVAL_PAYMENT_RECEIPT CONCEPT) [1]
```

Εικόνα 4.17 Αρχείο SOL για παραλαβή κωδικού απόδειξης από την καταβολή εργοδοτικών εισφορών στο ΙΚΑ

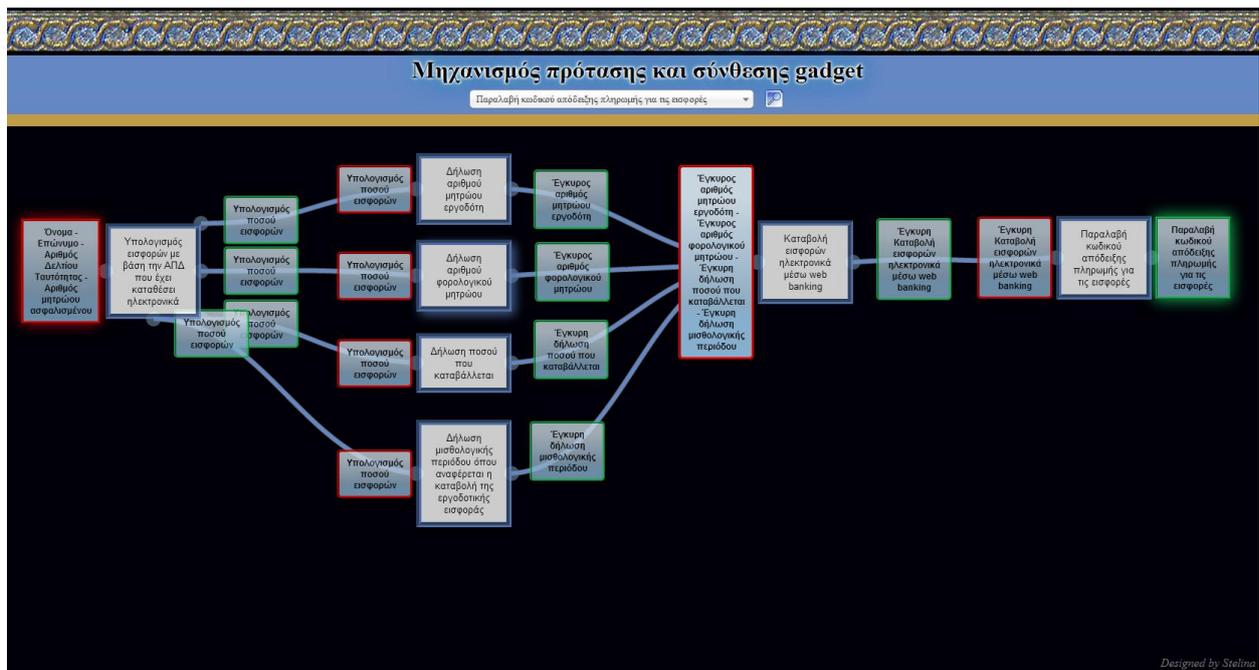
4.2.1.5. Screenshots

Στη συγκεκριμένη ενότητα παρουσιάζεται ένα σενάριο χρήσης για πρόταση gadgets έχοντας ως τελική λειτουργικότητα την παραλαβή κωδικού απόδειξης για τις εισφορές. Ο χρήστης επιλέγει την παραλαβή κωδικού απόδειξης για τις εισφορές, όπως φαίνεται στην παρακάτω Εικόνα 4.18.



Εικόνα 4.18 Αναζήτηση για παραλαβή κωδικού απόδειξης από την καταβολή εργοδοτικών εισφορών στο ΙΚΑ

Αφού πατήσει αναζήτηση, του παρουσιάζονται τα gadgets που είναι απαραίτητα για να παραλάβει την απόδειξή του, όπως παρατίθεται στην Εικόνα 4.19 παρακάτω. Εκτός από τα gadgets φαίνονται αναλυτικά τα δεδομένα εισόδου και εξόδου του καθενός.



Εικόνα 4.19 Πρόταση απαιτούμενων gadgets για παραλαβή κωδικού απόδειξης από την καταβολή εργοδοτικών εισφορών στο ΙΚΑ

5. Συμπεράσματα και Μελλοντικές κατευθύνσεις

Παρατηρώντας τις διαστάσεις που έχουν πάρει τα κοινωνικά δίκτυα στη σημερινή εποχή, καθώς και τις ποικίλες εφαρμογές που παρέχονται μέσω αυτών, αποτελεί επιτακτική ανάγκη η δημιουργία συστημάτων που θα μπορούν να επεξεργάζονται και να δημιουργούν δυναμικά λειτουργικότητα πολλαπλών εφαρμογών. Η αυτόματη δημιουργία και σύνθεση εφαρμογών, μπορεί να πραγματοποιηθεί μέσω ευφών τεχνικών. Ακόμα, εξετάζοντας και την ανάγκη των χρηστών να αναζητούν εφαρμογές, φαίνεται ξεκάθαρα ότι στην προσπάθειά τους αυτή η διαδικασία ανεύρεσης της ζητούμενης λειτουργικότητας γίνεται ολοένα και πιο δύσκολη.

Η παρούσα προσέγγιση προτείνει ένα δυναμικό μηχανισμό πρότασης και σύνθεσης εφαρμογών, που βοηθά την αναζήτηση των χρηστών. Μέσω αυτού, οι ίδιοι απαλλάσσονται από την υποχρέωση να γνωρίζουν εκ των προτέρων όλες τις διαθέσιμες εφαρμογές που έχουν εγκατασταθεί σε ένα κοινωνικό δίκτυο. Επίσης, δίνεται η δυνατότητα μέσω του μηχανισμού να

παρουσιαστεί ένας βέλτιστος τρόπος ώστε να φτάσουν στην τελική λειτουργικότητα που επιθυμούν.

Μία ενδιαφέρουσα μελλοντική προσέγγιση που χρειάζεται περαιτέρω υλοποίηση είναι η σημασιολογική ταύτιση των εισόδων και εξόδων μιας εφαρμογής. Αναλυτικότερα, έστω ότι υπάρχει η περίπτωση εγκατάστασης μίας εφαρμογής με τα δεδομένα εισόδου και εξόδου της και μετέπειτα εγκαθίσταται κάποια άλλη που περιέχει τα ίδια ή τουλάχιστον σημασιολογικά και εννοιολογικά τα ίδια δεδομένα. Όταν ο χρήστης αναζητήσει μία λειτουργικότητα θα ήταν επιθυμητό το σύστημα να αντιληφθεί μέσω ελέγχου ταύτισης των εννοιών αν υπάρχουν παρόμοιες εφαρμογές που ταιριάζουν στις ανάγκες του ώστε να του τις προτείνει. Ενοποίηση με συστήματα όπως το WordNet, θα μπορούσαν να φανούν χρήσιμα ώστε να επιτευχθεί η συγκεκριμένη υλοποίηση.

Μία ακόμα επέκταση που θα μπορούσε να υλοποιηθεί μελλοντικά είναι ο μηχανισμός να λαμβάνει υπόψη το ρόλο που κατέχει ο χρήστης στο σύστημα, έτσι ώστε όταν πραγματοποιείται μία αναζήτηση να του εμφανίζονται οι διαθέσιμες επιλογές βάση του ρόλου του. Συγκεκριμένα, αν έχει εισέλθει στο σύστημα χρήστης που είναι 'φοιτητής', να του παρουσιαστούν μόνο οι εφαρμογές που απευθύνονται σε φοιτητές. Ακόμα, θα μπορούσε να υλοποιηθεί μετά την έγκριση του χρήστη, η ταυτόχρονη και αυτόματη εγκατάσταση των εφαρμογών που του έχουν προταθεί ώστε να εκτελέσει την τελική λειτουργικότητα που επιθυμεί.

Βιβλιογραφία

1. T. Andrews et al. Business Process Execution Language for Web Services (BPEL4WS) 1.1. Online: <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel>, May 2003.
2. T. Bellwood et al. Universal Description, Discovery and Integration specification (UDDI) 3.0. Online: <http://uddi.org/pubs/uddi-v3.00-published-20020719.htm>.
3. D. Box et al. Simple Object Access Protocol (SOAP) 1.1. Online: <http://www.w3.org/TR/SOAP/>, 2001.
4. F. Casati, M. Sayal, and M.-C. Shan. Developing e-services for composing eservices. In Proceedings of 13th International Conference on Advanced Information Systems Engineering(CAiSE), Interlaken, Switzerland, June 2001. Springer Verlag.
5. R. Chinnici et al. Web Services Description Language (WSDL) 1.2. Online: <http://www.w3.org/TR/wsdl/>.
6. D. Martin et al. DAML-S(and OWL-S) 0.9 draft release. Online <http://www.daml.org/services/daml-s/0.9/> May 2003.
7. D. McDermott. Estimated-regression planning for interactions with Web services. In Proceedings of the 6th International Conference on AI Planning and Scheduling, Toulouse, France, 2002. AAAI Press.
8. S. McIlraith and T. C. Son. Adapting Golog for composition of Semantic Web services. In Proceedings of the 8th International Conference on Knowledge Representation and Reasoning (KR2002), Toulouse, France, April 2002.
9. S. McIlraith, T. C. Son, and H. Zeng. Semantic Web services. IEEE Intelligent Systems, 16(2):46–53, March/April 2001.

10. B. Medjahed, A. Bouguettaya, and A. K. Elmagarmid. Composing Web services on the Semantic Web. *The VLDB Journal*, 12(4), November 2003.
11. S. Narayanan and S. McIlraith. Simulation, verification and automated composition of Web service. In *Proceedings of the 11th International World Wide Web Conference*, Honolulu, Hawaii, USA, May 2002. ACM. Presentation available at <http://www2002.org/presentations/narayanan.pdf>.
12. S. R. Ponnekanti and A. Fox. SWORD: A developer toolkit for Web service composition. In *Proceedings of the 11th World Wide Web Conference*, Honolulu, HI, USA, 2002.
13. E. Sirin, J. Hendler, and B. Parsia. Semi-automatic composition of Web services using semantic descriptions. In *Proceedings of Web Services: Modeling, Architecture and Infrastructure workshop in conjunction with ICEIS2003*, 2002.
14. D. Wu, E. Sirin, J. Hendler, D. Nau, and B. Parsia. Automatic Web services composition using SHOP2. In *Workshop on Planning for Web Services*, Trento, Italy, June 2003.
15. Gerevini, A., Saetti, A., Serina, I., 2004. *LPG-td: a Fully Automated Planner for PDDL2.2 Domains*, in *IPC, 14th ICAPS*, 2004.
16. Ourania Hatzi, Dimitris Vrakas, Mara Nikolaidou, Nick Bassilades, Dimosthenis Anagnostopoulos, Ioannis Vlahavas, *An Integrated Approach to Automated Semantic Web Service Composition through Planning*, *IEEE Transactions on Services Computing*, 2011.
17. Hatzi, O., Meditskos, G., Vrakas, D., Bassiliades, N., Anagnostopoulos, D., Vlahavas, I., 2009b. *Semantic Web Service Composition using Planning and Ontology Concept Relevance with PORSCE II*, *The IEEE / WIC / ACM Conference on Web Intelligence 2009*.
18. O. Hatzi, D. Vrakas, N. Bassiliades, D. Anagnostopoulos, I. Vlahavas, "VLEPPO: A Visual Language for Problem Representation", *PlanSIG 07*, Roman Bartak (Ed.), pp. 60 - 66, Prague, Czech Republic, 2007.

19. O. Hatzi, D. Vrakas, N. Bassiliades, D. Anagnostopoulos, I. Vlahavas, "A Visual Programming System for Automated Problem Solving", *Expert Systems With Applications*, Elsevier, Vol. 37 (6), pp. 4611-4625, 2010.
20. Jia Hu Zhiyong Feng. *Automated Composition of Semantic Web Services Using Case-Based Planning*, China, 2009
21. Lajmi, S., Ghedira, C.: *Wesco cbr: How to compose web services via case based reasoning*. In *The IEEE International Symposium on Service-Oriented Applications, Integration and Collaboration held with the IEEE International Conference on e-Business Engineering (ICEBE 2006)*, Shanghai, China (October 2006)
22. B. Limthanmaphon and Y. Zhang. *Web service composition with case-based reasoning*. *Proceedings of the 14th Australasian Database Conference*, February 2003.
23. Durkin, J. (1994), *Expert Systems, Design and Development*, Macmillan
24. Frenzel, L. (1987), *Crash course in AI and expert systems*, Sams, Indianapolis
25. Turban, E. (1993), *Decision support and expert systems: Management support systems*, 3rd ed., Macmillan, New York.
26. Χούδρα Β. (2011), *Unity: Περιβάλλον Κοινωνικής Δικτύωσης για την Συνεργασία των μελών της Ακαδημαϊκής Κοινότητας – Συμμετέχοντες και Διαχείριση Σχέσεων*, Χαροκόπειο Πανεπιστήμιο, 2011.
27. Κατσιβέλης–Περάκης Π. (2011), *Unity: Περιβάλλον Κοινωνικής Δικτύωσης για την συνεργασία των Μελών της Ακαδημαϊκής Κοινότητας – Δημιουργία και Χρήση Εφαρμογών*, Χαροκόπειο Πανεπιστήμιο, 2011.
28. Καπουράνη–Χαντριόλντ Α. (2012), *Ολοκλήρωση Εξωτερικών Εφαρμογών στο Περιβάλλον του Ακαδημαϊκού Κοινωνικού Δικτύου Unity*, Χαροκόπειο Πανεπιστήμιο, 2012.

29. Μελετάκης Ιωάννης (2012), Συγχρονισμός σύνθετων υπηρεσιών Ηλεκτρονικής Διακυβέρνησης με τη χρήση Τεχνολογιών Κοινωνικού Δικτύου, 2012.
30. Semantic Web Standards W3C, <http://www.w3.org/2001/sw/wiki/SAWSDL>