



ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ - WEBINTELLIGENCE

**Ευφυείς Πράκτορες για την Υποστήριξη Κινητής
Μάθησης (Intelligent Agents for Mobile-Learning)**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ιωάννη Σταυρίδη

Επιβλέπων : Σταμάτης Δημοσθένης
Βαθμίδα, Ίδρυμα

Θεσσαλονίκη, Ιούνιος 2017

Η σελίδα αυτή είναι σκόπιμα λευκή.



ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ - WEBINTELLIGENCE

Τίτλος Μεταπτυχιακής Διπλωματικής Εργασίας

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ιωάννη Σταυρίδη

Επιβλέπων : Σταμάτης Δημοσθένης
Choose an item.ATEIΘ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις Choose a date.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Όνομα Επώνυμο
Choose an item.A.T.E.I.Θ.

.....
Όνομα Επώνυμο
Choose an item.A.T.E.I.Θ.

.....
Όνομα Επώνυμο
Choose an item.A.T.E.I.Θ.

Θεσσαλονίκη, Choose a date

(Υπογραφή)

.....

Click here to enter text.

Click here to enter text.

© Choose a date– Allrightsreserved

Περίληψη

Η μάθηση και κατ' επέκταση η εκπαίδευση, είναι παρούσα στην ζωή ενός ανθρώπου από τα πρώτα χρόνια της ζωής του, παίζοντας σημαντικό ρόλο. Με την πάροδο των χρόνων και την ανάπτυξη της τεχνολογίας η διαδικασία της μάθησης έχει εξελιχθεί από διαπροσωπική σε μαζικής κλίμακας προσωποκεντρικά στοχευμένη. Εκμεταλλευόμενοι την πρόοδο αυτή, γνώρισε μεγάλη άνθιση η διαδικασία εκμάθησης μέσω ψηφιακών μέσων, το γνωστό σε όλους μας e-learning. Καθώς τα τελευταία χρόνια η ανάπτυξη του τομέα των κινητών συσκευών είναι ραγδαία, έχει οδηγήσει σε μεγάλη άνοδο την εκμάθηση μέσω κινητών συσκευών, που ονομάζεται κινητή μάθηση ή αλλιώς Mobile Learning. Η παρακάτω εργασία πραγματεύεται τον συνδυασμό χρήσης ευφών πρακτόρων στην κινητή μάθηση. Οι ευφώνς πράκτορες αποτελούν προγράμματα που έχουν την ικανότητα αυτόνομης λειτουργίας στον περιβάλλον που ορίζονται. Κάνοντας χρήση της βιβλιοθήκης JADE δημιουργήσαμε ένα σύστημα ευφών πρακτόρων που παρουσιάζει ένα ερωτηματολόγιο στον κάτοχο-μαθητή κινητής συσκευής με σκοπό την ανατροφοδότηση του καθηγητή για την γνώση που μέχρι τώρα έχει αποκτήσει. Η βιβλιοθήκη JADE είναι βασισμένη στην γλώσσα προγραμματισμού Java οπότε κάνοντας χρήση του επιπρόσθετου της, JADE-LEAP επιτρέπει την μετανάστευσή ενός πράκτορα από ένα σταθερό σύστημα σε μια κινητή συσκευή. Στο κείμενο που ακολουθεί, στο κεφάλαιο 1,2 και 3 γίνεται αναφορά στην παραδοσιακή, στην ηλεκτρονική, στην κινητή μάθηση, και μεταφέρουμε τις κυριότερες αρχιτεκτονικές-μοντέλα κινητής μάθησης. Τέλος στο κεφάλαιο 4 παρουσιάζουμε την περίπτωση χρήσης με τα εργαλεία που επιλέξαμε για τη ανάπτυξη της και στο κεφάλαιο 5 παρουσιάζουμε το σύστημα που δημιουργήθηκε αναλύοντας ένα μεγάλο μέρος του κώδικα υλοποίησης.

Λέξεις Κλειδιά: Ηλεκτρονική μάθηση, Κινητή μάθηση, Ευφώνς πράκτορες, Jade, Jade-Leap, Android

Η σελίδα αυτή είναι σκόπιμα λευκή.

Abstract

Learning and education is present in a person's life from the early years of his life, playing an important role. Over the years and the development of technology the learning process has evolved from interpersonal to mass-scale person-centered targeting. Taking advantage of this progress, the learning process through digital media, E-learning, has flourished. As mobile devices development has been rapid in recent years, it has led to a big rise in learning through mobile devices, called Mobile Learning. The following work deals with the combination of intelligent agents in mobile learning. Intelligent agents are programs that have the ability to function autonomously in the defined environment. Using the JADE library, we created an intelligent agent system that presents a questionnaire to the owner-student of a mobile device in order to give the teacher feedback on the knowledge he has gained so far. The JADE library is based on Java programming language, while using its additional JADE-LEAP, allows an agent to migrate from a fixed system to a mobile device. In the following text, Chapters 1, 2 and 3 refer to traditional, electronic, mobile learning, and present the major mobile learning architectures. Finally in Chapter 4 we present the use case with the tools we chose for its development and in Chapter 5 we present the system created by analyzing a big part of the implementation code.

Keywords: E-learning, M-learning, Intelligent agents, Jade, Jade-Leap, Android

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1	Μάθηση σε κινητές και ακίνητες συσκευές.....	1
1.2	Ευφυής πράκτορες σε κίνηση.....	2
1.3	Οργάνωση κειμένου.....	2
2	Ηλεκτρονική μάθηση και πράκτορες.....	3
2.1	Εκπαίδευση.....	3
2.1.1	Μάθηση.....	3
2.1.2	Σύντομο Ιστορικό της Εξ Αποστάσεως Εκπαίδευσης.....	5
2.1.3	Πλεονεκτήματα-μειονεκτήματα εφαρμογής της εξ' αποστάσεως εκπαίδευσης.....	8
2.2	E-Learning.....	10
2.2.1	Εισαγωγή στο e-learning.....	10
2.2.2	Τι είναι το e-learning.....	11
2.2.3	Μορφές e-learning.....	12
2.3	Πράκτορες- Agents.....	13
2.3.1	Ευφυείς πράκτορες.....	13
2.3.2	Κινητοί πράκτορες.....	14
3	Mobile Learning.....	20
3.1	Ορισμός και χαρακτηριστικά mobile learning.....	20
3.1.1	Ορισμός mobile learning.....	20
3.1.2	Χαρακτηριστικά του mobile Learning.....	21
3.2	Μοντέλα και αρχιτεκτονικές mobile learning.....	25
3.2.1	Μοντέλα.....	25
3.2.1.1	Παιδαγωγικό μοντέλο ανεπτυγμένο για κινητή μάθηση.....	25
3.2.1.2	Ευφυές μοντέλο μάθησης βασισμένο στον Ιστό.....	26
3.2.1.3	Μοντέλο πολλαπλών πρακτόρων Bee-gent.....	27
3.2.1.4	Το μοντέλο κινητής μάθησης του Shih.....	29
3.2.2	Αρχιτεκτονικές.....	30
3.2.2.1	Αρχιτεκτονική από τους Ally, Mohamed - Lin, Fuhua - McGreal, Rory - Woo, Brian.....	30
3.2.2.2	Αρχιτεκτονική από τους L. Henry and S. Sankaranarayanan.....	33

3.2.2.3	Αρχιτεκτονική MX-Learn	37
3.2.2.4	Αρχιτεκτονική 3 Tier InfoStations	40
3.3	JADE.....	44
3.3.1	Μοντέλο Αρχιτεκτονικής Jade	46
4	Μελέτη περίπτωσης Χρήσης και εργαλεία.....	50
4.1	Μελέτη περίπτωσης Χρήσης - Jade_QA_Agents	50
4.2	Εργαλεία υλοποίησης Jade_QA_Agents	54
4.2.1	Java.....	54
4.2.2	Εγκατάσταση της JADE.....	54
4.2.3	Android Studio και Jade-LEAP	59
5	Jade_QA_Agents Υλοποίηση.....	61
5.1	Ανάλυση δημιουργίας εφαρμογής	61
5.1.1	Δημιουργία εφαρμογής- Android.....	63
5.1.2	Αποστολή μηνύματος από AndroidStudent.....	67
5.1.3	Εφαρμογή Jade QA Agents στον υπολογιστή.....	68
5.1.4	Δημιουργία και μετανάστευση πράκτορα student.....	71
5.1.5	Λήψη και εμφάνιση ερωτηματολογίου.....	75
5.1.6	Εμφάνιση και απάντηση ερωτήσεων	77
5.1.7	Επιστροφή πράκτορα student	78
6	Επίλογος	80
6.1	Σύνοψη και συμπεράσματα.....	80
6.2	Μελλοντικές επεκτάσεις	81
7	Βιβλιογραφία	82

1

Εισαγωγή

1.1 Μάθηση σε κινητές και ακίνητες συσκευές

Η διαδικασία της μάθησης αποτελεί θεμέλιο λίθο για την για την σωστή μετάδοση της γνώσης στο βάθος των αιώνων. Στις σύγχρονες κοινωνίες ακολουθείτε η παραδοσιακή διαπροσωπική εκπαίδευση από την δημιουργία τους. Από τα μέσα της δεκαετίας του 1990 η εκπαιδευτική κοινότητα προχώρησε σε επανεκτίμηση της εκπαιδευτικής διαδικασίας και στην σταδιακή ανάπτυξη και ενσωμάτωσή της εκπαίδευσης μέσω ηλεκτρονικής μάθησης. Η ηλεκτρονική μάθηση (e-learning) αποτελεί ένα μέρος των δυνατοτήτων εκμάθησης από απόσταση. Αναλογιζόμενοι τα πλεονεκτήματα της μεθόδου θα διακρίνει κανείς την δυνατότητα πρόσβασης στην γνώση από οποιαδήποτε σημείο διαθέτει ενεργή πρόσβαση στο διαδίκτυο, σε οποιοδήποτε χρονική στιγμή επιθυμεί, μείωση του κόστους μετακινήσεων του εκπαιδευόμενου - εκπαιδευτή καθώς και το χρονικού κόστους όπως επίσης και την δυνατότητα αναζήτησης επιπλέον πηγών και πληροφοριών από το μέσο που προσφέρει την ηλεκτρονική μάθηση. Η αποδοχή της ηλεκτρονικής μάθησης και η ενσωμάτωση της ξεκίνησε από την αρχή του 2000. Τα εκπαιδευτικά ιδρύματα αξιολογώντας θετικά την νέα αυτή μέθοδο ξεκίνησαν, παράλληλα με την παραδοσιακή μέθοδο διαπροσωπικής εκπαίδευσης, την δημιουργία και παροχή μαθημάτων ηλεκτρονικής μάθησης σε φοιτητές που δεν είχαν την δυνατότητα μετακίνησης από και προς τους χώρους του πανεπιστημίου. Η τεχνολογική ανάπτυξη στον τομέα των κινητών συσκευών έφερε την ιδέα για παροχή της υπηρεσίας ηλεκτρονικής μάθησης μέσω των νέων αυτών συσκευών. Κατά αυτό τον τρόπο

δημιουργήθηκε ο όρος κινητή μάθηση και ξεκίνησε σταδιακά η ενσωμάτωση του στην εκπαιδευτική διαδικασία. Στην κατηγορία των κινητών συσκευών που μπορούν να συμβάλουν στην μάθηση λογίζονται συσκευές όπως φορητοί υπολογιστές, pda, tablet και smartphone οι οποίες έχουν όλες την δυνατότητα για διασύνδεση σε εξ αποστάσεων δίκτυα, ανταλλαγή δεδομένων και αλληλεπίδραση μέσω της οθόνης που διαθέτουν.

1.2 Ευφυής πράκτορες σε κίνηση

Η λειτουργία των μοντέλων ηλεκτρονικής μάθησης και κατ' επέκταση των μοντέλων κινητής μάθησης προϋποθέτουν την δημιουργία πληροφοριακών συστημάτων πάνω στα οποία θα βασιστούν οι εκπαιδευτές και οι εκπαιδευόμενοι. Οι μεν εκπαιδευτές για να εισάγουν, να διαχειριστούν και να επεκτείνουν το μαθησιακό περιεχόμενο που επιθυμούν και οι δε εκπαιδευόμενοι για να το προσπελάσουν. Η αναφορά σε συγκεκριμένα συστήματα είναι πέρα από του σκοπούς της παρούσας διπλωματικής.

Στόχος της διπλωματικής εργασίας είναι η δημιουργία μιας περίπτωσης χρήσης όπου ο μαθητευόμενος θα λαμβάνει ένα ερωτηματολόγιο από τον εκπαιδευτή του, θα απαντάει στις ερωτήσεις που του παρουσιάζονται και στην συνέχεια θα το αποστέλλει στον εκπαιδευτή για ανατροφοδότηση και ενδεχόμενη αναπροσαρμογή του υλικού εκμάθησης που του δίνεται.

1.3 Οργάνωση κειμένου

Η υλοποίηση αυτής την περίπτωσης χρήσης γίνεται με την δημιουργία εφαρμογής σε περιβάλλον android κάνοντας χρήση της γλώσσας προγραμματισμού JAVA , της βιβλιοθήκης δημιουργίας και διαχείρισης ευφυών πρακτόρων JADE και του επιπρόσθετου αυτής JADE-LEAP που απαιτείται για την αλληλεπίδραση με συσκευές λειτουργικού android. Ξεκινώντας από το κεφάλαιο 2 παρουσιάζεται μια σύντομη αναφορά, για την μάθηση, το e-learning και στις εφαρμογές πρακτόρων. Στο κεφάλαιο 3 συναντάμε την κινητή μάθηση, τα χαρακτηριστικά της και τις αρχιτεκτονικές της με παράλληλη αναφορά στο πλαίσιο JADE. Στο κεφάλαιο 4 γίνεται αναφορά για την περίπτωση χρήσης και τις τεχνολογίες που χρησιμοποιούνται για την υλοποίηση της. Ακολουθεί το κεφάλαιο 5 όπου αναλύεται η δημιουργία και τα αποτελέσματα της εφαρμογής με το κεφάλαιο 6 να αναφέρει τα συμπεράσματα.

2

Ηλεκτρονική μάθηση και πράκτορες

2.1 Εκπαίδευση

2.1.1 Μάθηση

Η μάθηση είναι μια διαδικασία που ακολουθεί ο άνθρωπός από τις πρώτες μέρες της ζωής του. Ο καθένας μπορεί να την εφαρμόσει διαφορετικά, με τους ρυθμούς που αυτός επιθυμεί και ακολουθεί. Η γενική μαθησιακή διαδικασία, ανάλογα με τον τρόπο διεξαγωγής της, μπορεί να διακριθεί σε επιμέρους κατηγορίες. Μία βασική κατηγορία μάθησης είναι η εκπαίδευση, όπου η διαδικασία απόκτησης γνώσης δομείται και ελέγχεται από έναν εξωτερικό παράγοντα και έχει προκαθορισμένο σκοπό, ενώ μία άλλη μορφή είναι η ανοικτή μάθηση, όπου η διαδικασία κινείται και ελέγχεται σε κάποιο βαθμό από τον εκπαιδευόμενο. Ακόμα, μία βασική διάκριση που παρατηρείται, είναι αυτή ανάμεσα στην ατομική μάθηση και στην ομαδική μάθηση, που συνυπάρχουν αρμονικά σε περιβάλλοντα συνεργατικής μάθησης.

Στη συνέχεια αναλύονται κάποιοι βασικοί όροι που συνδέονται με τη μάθηση.

Μάθηση (learning) : Ως μάθηση ορίζεται οποιαδήποτε διαδικασία με την οποία ένα άτομο ή μια ομάδα ατόμων βρίσκονται σε μια διαδικασία αλλαγής σε ένα ή περισσότερα από τα

παρακάτω πεδία : το πεδίο συμπεριφοράς , το συναισθηματικό πεδίο , το γνωστικό πεδίο ή το πληροφοριακό πεδίο.

Εκπαίδευση: Ως εκπαίδευση ορίζεται μια διαδικασία προσχεδιασμένης μάθησης προερχόμενη, ελεγχόμενη και δομημένη από εξωτερικούς παράγοντες σε σχέση με τους μαθητές/εκπαιδευόμενους των οποίων τα επιθυμητά αποτελέσματα έχουν προαποφασιστεί από τον εξωτερικό παράγοντα.

Ανοικτή Μάθηση : Ως ανοικτή μάθηση ορίζεται μια διαδικασία που προέρχεται και/ή ελέγχεται σε κάποια έκταση από τον ίδιο τον μαθητή.

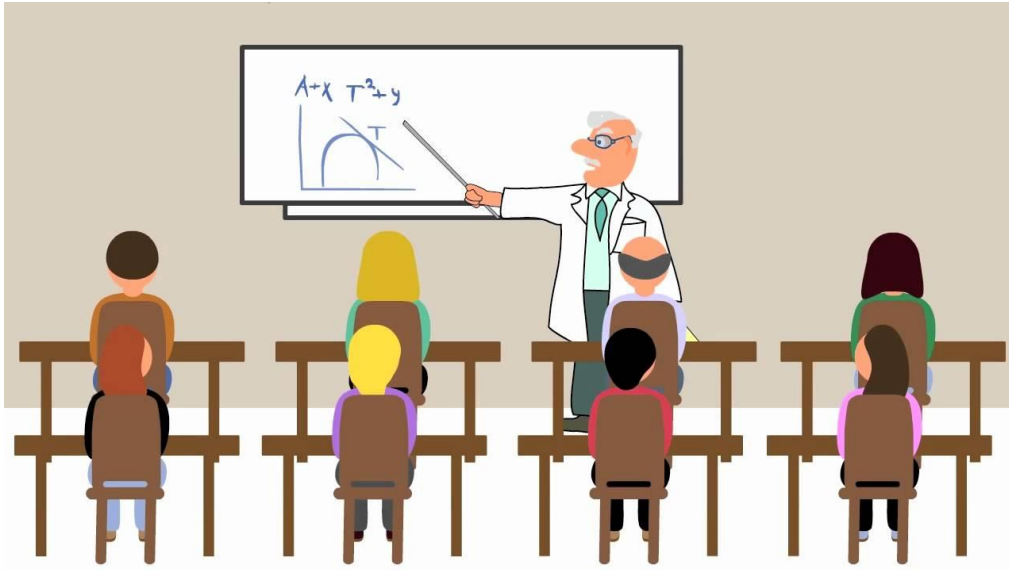
Ατομική Μάθηση: Ως ατομική (ή αυτόνομη) μάθηση ορίζεται οποιαδήποτε διαδικασία (σχετική με την εκπαίδευση) που ακολουθείται από ένα άτομο.

Ομαδική Μάθηση : Ως ομαδική μάθηση ορίζεται οποιαδήποτε διαδικασία (σχετική με την εκπαίδευση) που ακολουθείται από ένα ή περισσότερα άτομα.

Συνεργατική μάθηση : Ως συνεργατική μάθηση ορίζεται οποιαδήποτε διαδικασία ομαδικής μάθησης στην οποία λαμβάνουν χώρα τουλάχιστον κάποιες από τις σημαντικές μαθησιακές αλληλεπιδράσεις μεταξύ των μαθητών.

Στον τομέα της εκπαίδευσης παρατηρείται μια σημαντική αλλαγή στον τρόπο της διδασκαλίας. Η διδασκαλία, όπως αυτή προσδιορίζεται από την κλασική άποψη, αποτελούσε μια διαδικασία βασισμένη στην ομιλία και την παράδοση προγραμμάτων σπουδών από τον καθηγητή/ειδικό στους μαθητές / εκπαιδευόμενους. Η άποψη αυτή, για την διδασκαλία, τείνει να μετακινηθεί προς την διδασκαλία που βασίζεται στην προώθηση μαθησιακών συζητήσεων και την μοντελοποίηση της πρακτικής των ειδικών. Βασικός γνώμονας στην νέα άποψη είναι η προώθηση αλλαγών στις αντιλήψεις των μαθητών και στις ακολουθούμενες στρατηγικές, έτσι ώστε να επιτευχθεί βαθιά κατανόηση των διδασκόμενων αντικειμένων. Βασικό αποτέλεσμα αυτής της τάσης είναι η σταδιακή μετατροπή του ρόλου του διδάσκοντα από απλό παροχέα πληροφοριών σε σύμβουλο της εκπαιδευτικής διαδικασίας.

Η συνεργατική μάθηση μπορεί να εφαρμοστεί μεταξύ άλλων και από απόσταση. Η ιδέα της εφαρμογής της σε ένα εικονικό περιβάλλον οδήγησε στην ανάπτυξη ενός νέου τύπου διδασκαλίας η οποία πραγματοποιείται σε Ηλεκτρονικά Συστήματα Μάθησης και ονομάζεται ηλεκτρονική μάθηση ή αλλιώς e-learning. [1]



Εικόνα 1 Παραδοσιακή Μάθηση

2.1.2 Σύντομο Ιστορικό της Εξ Αποστάσεως Εκπαίδευσης

Την αρχή στην ιστορία της εξ αποστάσεως εκπαίδευσης μπορούμε να την προσδιορίσουμε χρονικά πριν από 150 χρόνια. Η εξ αποστάσεως εκπαίδευση θα ήταν αδύνατη χωρίς την ανάπτυξη της τεχνολογίας, των μεταφορών και των τηλεπικοινωνιών που πραγματοποιήθηκε με τη Βιομηχανική Επανάσταση. Η εξ αποστάσεως εκπαίδευση έχει ως βασικό γνώρισμα την απόσταση που χωρίζει το διδάσκοντα από τους διδασκόμενους αλλά και τους διδασκόμενους μεταξύ τους.

Κατά την συγκεκριμένη μορφή εκπαίδευσης, η διαπροσωπική επικοινωνία της συμβατικής εκπαίδευσης έχει υποκατασταθεί από μια απρόσωπη μορφή επικοινωνίας την οποία διεκπεραιώνει η τεχνολογία. Το είδος αυτό της εκπαίδευσης παρέχεται σήμερα από σχολεία καθώς και από ανοικτά πανεπιστήμια και από τμήματα παροχής εξ αποστάσεως εκπαίδευσης ή εξωτερικών σπουδών, τα οποία εντάσσονται στο πλαίσιο συμβατικών κολεγίων και πανεπιστημίων.

Ιστορικά, η ανάπτυξη της εξ αποστάσεως εκπαίδευσης οφείλεται κυρίως στις προσπάθειες της Μεγάλης Βρετανίας, του Καναδά και της Αυστραλίας. Παρόλο που έγιναν προσπάθειες και από άλλα κράτη, η πιο σημαντική ολοκληρωμένη κίνηση για την ανάπτυξη της εξ αποστάσεως εκπαίδευσης έγινε με την ίδρυση του Βρετανικού Ανοικτού Πανεπιστημίου το 1969 που παραμένει πρόδρομος όλων των εξελίξεων που αφορούν την εξ αποστάσεως εκπαίδευση. Το Ανοικτό Πανεπιστήμιο αποτελεί το μεγαλύτερο βρετανικό πανεπιστήμιο με πάνω από 200.000 φοιτητές που αντιπροσωπεύει πάνω από το 20% του συνόλου των μερικής απασχόλησης φοιτητών ανώτατης εκπαίδευσης στο Ηνωμένο Βασίλειο.[2]

Ίσως πιο σημαντική και από την κάθετη επιτυχία του Ανοικτού Πανεπιστημίου να ήταν η ίδρυση οκτώ παρομοίων πανεπιστημίων σε όλο τον κόσμο (Καναδάς, Κίνα, Κόστα Ρίκα, Γερμανία, Ισραήλ, Πακιστάν, Ισπανία και Βενεζουέλα) καθώς και παρόμοια τμήματα σε διάφορα άλλα πανεπιστήμια (Willis, 1994). Παγκοσμίως, υπολογίζεται ο αριθμός των φοιτητών ανώτατης εκπαίδευσης που παρακολουθούν ένα πρόγραμμα εξ αποστάσεως να κυμαίνεται από 5% έως 15% του συνόλου, ξεπερνώντας σε μερικές περιοχές το 25% (Harry & Perraton, 1999). Το e learning είναι μια από τις ταχέως αναπτυσσόμενες βιομηχανίες και συνεχίζει να αναπτύσσεται. Από το 2000 ο ρυθμός που έχει αναπτυχθεί είναι 900% [Πηγή: www.livetraining.gr]

Συνοπτικά γύρω στα τέλη της δεκαετίας του 1970 διευκρινίστηκε η ορολογία και σημειώθηκε πρόοδος ως προς τη θεωρία που υποστηρίζει την εξ αποστάσεως εκπαίδευση. Στη δεκαετία του 1980 αυξήθηκε ραγδαία η βιβλιογραφία της εξ αποστάσεως εκπαίδευσης, εκδόθηκαν επιστημονικά περιοδικά και αναπτύχθηκε η έρευνα σε κλάδους όπως ο σχεδιασμός, ο προγράμματος, η οικονομία της εξ αποστάσεως εκπαίδευσης, οι υπηρεσίες υποστήριξης των σπουδαστών και τα μέσα της εξ αποστάσεως εκπαίδευσης. Στα μέσα της δεκαετίας του 1980 έκαναν την εμφάνισή τους τα πρώτα προγράμματα στην εξ αποστάσεως εκπαίδευση που απένειμαν πανεπιστημιακό τίτλο.

Όσον αφορά την ελληνική πραγματικότητα αξίζει να σημειωθεί ότι τα ΑΕΙ έχουν ικανοποιητική υποδομή προκειμένου να πραγματοποιούν εκπαιδευτικά προγράμματα εξ αποστάσεως, η οποία όμως δεν έχει αξιοποιηθεί αποτελεσματικά. Μόλις το 1995 συστάθηκε το Εθνικό Δίκτυο Έρευνας και Τεχνολογίας (ΕΔΕΤ), ο φορέας δηλαδή που ανέλαβε την υποστήριξη μέσω διαδικτύου της ελληνικής ακαδημαϊκής και ερευνητικής κοινότητας. Ως το 2016 έχει πραγματοποιήσει τη διασύνδεση περισσότερων από 50 ιδρυμάτων 36 (ερευνητικά κέντρα του Υπουργείου Ανάπτυξης καθώς και ΑΕΙ και ΤΕΙ) με περίπου 150.000 χρήστες, δηλαδή το 40% της κοινότητας των Ελλήνων χρηστών του διαδικτύου. Στην μελέτη της Γενικής Γραμματείας Έρευνας και Τεχνολογίας επισημαίνεται ωστόσο η χρήση “low tech” εκπαιδευτικών εργαλείων από το Ελληνικό Ανοικτό Πανεπιστήμιο. [2]

Όπως έχει προαναφερθεί το Ανοικτό Πανεπιστήμιο λειτουργεί στην Ευρώπη εδώ και πολλές δεκαετίες. Στην Ελλάδα λειτουργεί μόλις από το 1999. Περίπου 5.000 φοιτητές είναι ήδη εγγεγραμμένοι ενώ υπολογίζεται ότι σε δυο χρόνια ο αριθμός των μαθητών θα αυξηθεί κατά 50%. Το εκπαιδευτικό πρόγραμμα υλοποιείται με την αποστολή υλικού προς τον φοιτητή, κυρίως εντύπου και οπτικοακουστικού, τις συμβουλευτικές συναντήσεις με τον καθηγητή, την συγγραφή εργασιών και τις εξετάσεις. Τα τελευταία χρόνια επίσης, γίνεται και διάθεση λογισμικού και ηλεκτρονικών βιβλίων προς τους φοιτητές του ΕΑΠ (<http://www.cap.gr>).

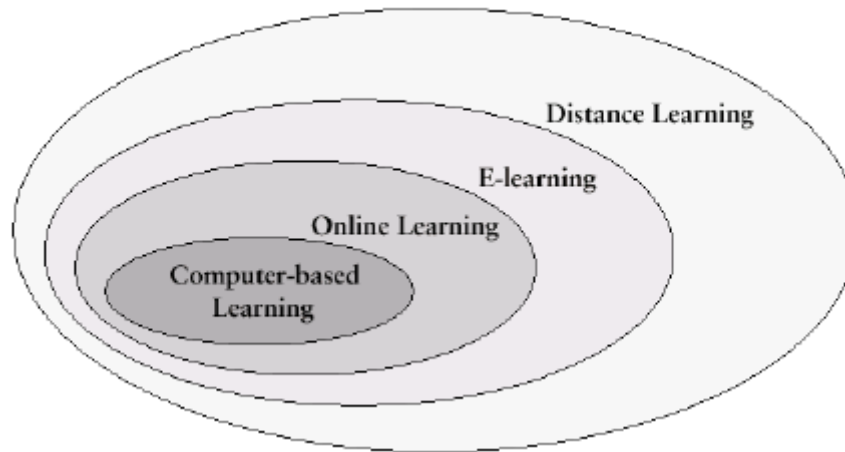
Η εξ αποστάσεως διδασκαλία και η εξ αποστάσεως μάθηση αποτελούν τις δυο διαστάσεις της διαδικασίας της εξ αποστάσεως εκπαίδευσης. Μετά από μελέτη διάφορων παλιών και νέων

ορισμών που δόθηκαν κατά καιρούς από πολλούς ερευνητές για την εξ αποστάσεως εκπαίδευση, παρακάτω παρατίθεται αυτός που συντάχθηκε το 1994 από τους Portway και Lane και επικεντρώνεται στο συνδυασμό εξ αποστάσεως μάθησης και διδασκαλίας:

Ο όρος «εξ αποστάσεως εκπαίδευση» αναφέρεται σε διδακτικές και μαθησιακές καταστάσεις κατά τις οποίες ο καθηγητής και ο διδασκόμενος ή οι διδασκόμενοι είναι απομακρυσμένοι γεωγραφικά και, ως εκ τούτου, η διδασκαλία του μαθήματος βασίζεται σε ηλεκτρονικά μέσα και έντυπο υλικό. Η εξ αποστάσεως εκπαίδευση περιλαμβάνει την εξ αποστάσεως διδασκαλία - τον ρόλο του διδάσκοντος στη διαδικασία και την εξ αποστάσεως μάθηση τον ρόλο του διδασκόμενου στη διαδικασία.

Παρακάτω ακολουθούν τα 6 βασικά συστατικά της εξ αποστάσεως εκπαίδευσης όπως διατυπώνονται από την κ. Βαμιαδάκη στο [2] :

- Την απόσταση που χωρίζει τον διδάσκοντα από το διδασκόμενο, στοιχείο το οποίο διαφοροποιεί την εξ αποστάσεως εκπαίδευση από την πρόσωπο με πρόσωπο διδασκαλία.
- Την παρεμβολή του εκπαιδευτικού οργανισμού στη μαθησιακή διαδικασία, στοιχείο που τη διαφοροποιεί από την προσωπική κατ' ιδίαν μελέτη.
- Τη χρήση τεχνικών μέσων για τη μεταφορά του περιεχομένου της εκπαίδευσης, συνήθως έντυπου υλικού, σημείο όπου συναντώνται διδάσκων και διδασκόμενος.
- Την εξασφάλιση αμφίδρομης επικοινωνίας, έτσι ώστε ο διδασκόμενος να επωφελείται και / ή ακόμα και από το άμεσο και ζωντανό διάλογο.
- Τη δυνατότητα συναντήσεων σε περιστασιακή βάση τόσο για διδακτικούς όσο και για κοινωνικούς σκοπούς.
- Το γεγονός ότι πρόκειται για βιομηχανοποιημένη μορφή εκπαίδευσης, πράγμα το οποίο, αν γίνει βέβαια αποδεκτό, διαφοροποιεί ριζικά την εξ αποστάσεως εκπαίδευση από τις άλλες μορφές του φάσματος της εκπαίδευσης.



Εικόνα. 2 Επίπεδα εκπαίδευσης

2.1.3 Πλεονεκτήματα-μειονεκτήματα εφαρμογής της εξ' αποστάσεως εκπαίδευσης

Τα κυριότερα πλεονεκτήματα από την εφαρμογή της εξ' αποστάσεως εκπαίδευσης μπορούν να συνοψιστούν στα παρακάτω σημεία:

- Δεν υπάρχουν φυσικά εμπόδια και γεωγραφικά σύνορα που μπορεί να επηρεάσουν στην μεταφορά των συμμετεχόντων.
- Υπάρχει ομοιόμορφη αντιμετώπιση του συνόλου των εκπαιδευομένων ως προς την απόστασή, την επιλογή του χρόνου εκπαίδευσης και το γνωστικό αντικείμενο.
- Συνεχής κατάρτιση και εκπαίδευση για τη βελτίωση των ικανοτήτων των εκπαιδευομένων.
- Αύξηση στην ταχύτητα της διαδικασίας μεταφοράς γνώσης από τους εκπαιδευτές στους εκπαιδευόμενους.
- Δυνατότητα άμεσης και επι 24ωρου βάσης διασύνδεση ατόμων και ομάδων με εξωτερικές πηγές γνώσεων (πχ. ηλεκτρονικές βιβλιοθήκες) και αυξημένες δυνατότητες συνεργασίας μεταξύ εκπαιδευόμενων και εκπαιδευτών, οι οποίοι είναι γεωγραφικά διασπαρμένοι.
- Δημιουργία μιας “ηλεκτρονικής τάξης” με εξομοίωση όλων των λειτουργιών μιας παραδοσιακής τάξης (παράδοση διαλέξεων επίλυση ασκήσεων, διόρθωση ασκήσεων, υποβολή ερωτήσεων κλπ).
- Εξοικονόμηση χρόνου, εκπαιδευτικού προσωπικού και πόρων και στις 2 πλευρές.

- Αύξηση του αριθμού των εκπαιδευόμενων με ταυτόχρονη μείωση των λειτουργικών αναγκών.
- Ευελιξία στο χρόνο, στο χώρο και στο ρυθμό μάθησης.
- Έλεγχος από την πλευρά του εκπαιδευόμενου για τον ρυθμό προόδου που σημειώνει κατά τη μαθησιακή διαδικασία.

Τα κυριότερα μειονεκτήματα από την εφαρμογή της εξ αποστάσεως εκπαίδευσης μπορούν να συνοψιστούν στα πιο κάτω:

- Μεγάλος χρόνος προετοιμασίας μαθημάτων (περίπου 50 φορές μεγαλύτερος σε σχέση με τα συμβατικά μαθήματα).
- Ικανότητα εκπαιδευόμενων να παρακολουθήσουν τα μαθήματα λόγω μιας απαίτησης για τεχνικές ικανότητες σχετιζόμενες με Η/Υ και με τηλεπικοινωνίες.
- Ισότητα όσον αφορά την ευκολία πρόσβασης στα ηλεκτρονικά μαθήματα (θέματα αποκλεισμού).
- Απαίτηση για ειδικές προγραμματιστικές ικανότητες (Dhtml, vrml, java applets, flash).
- Τεχνολογικοί περιορισμοί (π.χ. Bandwidth).[2]

2.2 E-Learning

2.2.1 Εισαγωγή στο e-learning

Οι ρυθμοί εξέλιξης στην σύγχρονη κοινωνία είναι καταγιστικοί προσφέροντας μέσα από την ραγδαία ανάπτυξη της τεχνολογίας ένα περιβάλλον ρευστό όπου η απόκτηση και διαχείριση της πληροφορίας συνεχώς μεταβάλλονται. Σε ένα τέτοιο περιβάλλον διαφαίνεται η ανάγκη μιας συνεχούς και δια βίου εκπαίδευσης. Η δια βίου εκπαίδευση έχει εξορισμού σαν στόχο την εξέλιξη του ατόμου στο εργασιακό του περιβάλλον αλλά μπορεί αν φτάσει μέχρι και τη ενημέρωσή για τις τεχνολογικές εξελίξεις στην καθημερινότητα του.

Το Internet έχει καταφέρει μέσα από την αλματώδη εξάπλωση του να προκαλέσει σημαντικές αλλαγές σε πολλούς κλάδους της οικονομία και της κοινωνίας. Τα αμέσως επόμενα χρόνια το τοπίο αναμένεται να αλλάξει δραματικά δημιουργώντας και στην χώρα μας, αυτό που όλοι σήμερα αναφέρεται ως "νέα οικονομία". Φυσικά από τις αλλαγές αυτές δεν θα μπορούσε να μείνει απέξω η εκπαίδευση.

Με την άνοδο της παγκοσμιοποίησης και την διενέργεια μελετών πάνω στην λειτουργία των επιχειρήσεων, έχει διαφανεί και ακολουθηθεί η πολιτική της συνεχούς εκπαίδευσης των εργαζομένων, των συνεργατών και των πελατών τους ώστε να επιτυγχάνεται το καλύτερο δυνατό αποτέλεσμα. Η αποκλειστική διανομή γνώσης μέσα από ξεχωριστά «κανάλια» όπως βίντεο, στατικές σελίδες και email έχει αποδειχθεί παρωχημένη. Η Ανάγκη για μια πιο ευέλικτη διαδικασία εκπαίδευσης έρχεται να καλυφθεί από την ηλεκτρονικής μάθησης ή e-learning όπως έχει επικρατήσει σαν όρος

Με τον όρο e-learning εννοούμε τη χρήση των νέων τεχνολογιών για την επιτάχυνση της διάχυσης της γνώσης από τον εκπαιδευτή προς τους εκπαιδευόμενους, μέσα σε ένα ευέλικτο περιβάλλον όπου οι ίδιοι οι εκπαιδευόμενοι επιλέγουν τον τρόπο, τον τόπο, τον χρόνο και την ποσότητα της γνώσης που θέλουν να πάρουν. Ο συνδυασμός του παραδοσιακού τρόπου εκπαίδευσης στη αίθουσα με τις τεχνολογίες e-learning, δίνει σήμερα τη δυνατότητα στις επιχειρήσεις να διαμορφώσουν ευέλικτα μοντέλα κατάρτισης του ανθρώπινου δυναμικού τους ανάλογα με τις εκπαιδευτικές ανάγκες του κάθε εργαζόμενου. [1]



Εικόνα 3 E-learning

2.2.2 Τι είναι το e-learning

Έχει γίνει κατανοητό και κοινά αποδεκτό ότι ο σύγχρονος άνθρωπος πρέπει να έχει: τη δυνατότητα να μαθαίνει με πολλαπλούς τρόπους καθώς τα οφέλη που απορρέουν είναι σημαντικά. Στα πλαίσια αυτής της αντίληψης, διεξάγεται σήμερα σε παγκόσμιο επίπεδο, έρευνα και ανάπτυξη στον τομέα της ηλεκτρονικής μάθησης καθώς και στον τομέα των προηγμένων μαθησιακών τεχνολογιών (όπως η κινητή μάθηση). Το e-learning είναι η διαδικασία μάθησης που πραγματοποιείται μέσα από σύγχρονες δικτυακές τεχνολογίες και προγράμματα υπολογιστών.

Ο μαθητής έχει την δυνατότητα για πλήρη έλεγχο του ρυθμού προόδου του, ενώ ταυτόχρονα υπάρχει διαθέσιμη η κατά βούληση υποστήριξη από τον καθηγητή ή ειδικό του θέματος. Η υποστήριξη είναι απαραίτητη, αφού στην αντίθετη περίπτωση θα μιλούσαμε μόνο για αυτοεκπαίδευση, η οποία θα μπορούσε να γίνει και με άλλα μέσα όπως με ένα βιβλίο ή με ένα εκπαιδευτικό βίντεο. Το e-learning εμπεριέχει την έννοια της συνεργατικής μάθησης και αλληλεπίδρασης μεταξύ των μαθητών. Επιπλέον δίνει την δυνατότητα διεύρυνσης της συχνότητας επικοινωνίας σε σχέση με αυτό που συμβαίνει στην κλασική εκπαίδευση, π.χ σε παραδοσιακή αίθουσα διδασκαλίας. Από την άλλη μεριά τα ηλεκτρονικά σεμινάρια και οι οποιεσδήποτε μαθησιακές δραστηριότητες γίνονται σε «τάξη». Ουσιαστικά ο εκπαιδευτικός και οι μαθητές βρίσκονται σε διαφορετικούς χώρους και η έννοια της τάξης δημιουργείται εικονικά. Έτσι η διδασκαλία και η μάθηση μπορεί να γίνεται με ασύγχρονη επικοινωνία, με σύγχρονη επικοινωνία ή να πραγματοποιείται σε εξατομικευμένο ρυθμό ανάλογα με το χρήστη. Σε αυτή την μορφή μάθησης το διαδίκτυο αποτελεί αναπόσπαστο μέρος της

υπηρεσίας που προσφέρεται στους εκπαιδευόμενους. Χωρίς αυτό, η νέου τύπου εκπαίδευση απλώς δεν μπορεί να προσφερθεί.

2.2.3 Μορφές e-learning

Αναλύοντας την εκπαίδευσή με βάση την χρονική στιγμή της επικοινωνίας μπορούμε να πούμε ότι υπάρχουν δύο μορφές, η σύγχρονη και η ασύγχρονη, στις οποίες όμως ο τόπος είναι ίδιος, αν και εικονικός ή πλασματικός:

Ασύγχρονη εκπαίδευση: Κατά την ασύγχρονη εκπαίδευση ο εκπαιδευόμενος έχει πρόσβαση σε πολύμορφες πηγές πληροφοριών. Στο υλικό αυτό έχει πρόσβαση οποιαδήποτε ημέρα και ώρα και πολλές φορές μπορεί να το μεταφέρει στον υπολογιστή του. Η χρήση του υλικού αυτού γίνεται συνήθως με την τεχνολογία των πολυμέσων σε χρόνο που αυτός επιθυμεί. Ο χρόνος αυτός δεν είναι απαραίτητο να συμπίπτει με το χρόνο των άλλων εκπαιδευόμενων που «συμμετέχουν» σε αυτήν την εικονική τάξη. Σε όλη την παραπάνω διαδικασία δεν απαιτείται η ταυτόχρονη on-line παρουσία του εκπαιδευτή. Ο τρόπος αυτός χρησιμοποιείται κυρίως από τα ανοικτά πανεπιστήμια και από άλλους φορείς που παρέχουν επιμόρφωση σε συγκεκριμένες ομάδες, π.χ. εργαζομένους.

Σύγχρονη εκπαίδευση: Κατά τη σύγχρονη εκπαίδευση, το μάθημα γίνεται σε προκαθορισμένο, πραγματικό χρόνο και εικονικό τόπο. Οι εκπαιδευόμενοι παρακολουθούν σε απευθείας σύνδεση τον εκπαιδευτή και μπορούν να συμμετέχουν σε όλες τις διαδικασίες του μαθήματος σε παραδοσιακού τρόπου διδασκαλίας στο μαθητοκεντρικό σύστημα, σε ηλεκτρονικές συνθήκες με τη διαφορά ότι ο εκπαιδευτής και η ονομαζόμενη τηλετάξη μπορούν να απέχουν μεταξύ τους εκατοντάδες ή χιλιάδες χιλιόμετρα.[3]

2.3 Πράκτορες- Agents

2.3.1 Ευφυείς πράκτορες

Η λέξη ‘πράκτορας’ (agent) προέρχεται από την ελληνική λέξη ‘άγω’ (άγειν/agein) που σημαίνει οδηγώ ή καθοδηγώ. Σήμερα ο όρος δηλώνει κάτι που παράγει ή είναι ικανό να παράγει αποτέλεσμα. Ένας γενικός Ορισμός ενός πράκτορα που επικρατεί είναι:

Μια αφηρημένη ή σωματική αυτόνομη οντότητα η οποία εκτελεί μια συγκεκριμένη εργασία χρησιμοποιώντας πληροφορίες προερχόμενες από το περιβάλλον του με στόχο να ενεργεί με έναν κατάλληλο τρόπο, έτσι ώστε να ολοκληρώνει το έργο του με επιτυχία. Ο πράκτορας πρέπει να είναι σε θέση να προσαρμοστεί με βάση τις αλλαγές που συμβαίνουν στο περιβάλλον του, έτσι ώστε μια αλλαγή στις συνθήκες περιβάλλοντος του να του επιτρέπει να εξακολουθεί να αποφέρει το επιδιωκόμενο αποτέλεσμα

Ο P.Maes, για τους αυτόνομους πράκτορες μας αναφέρει ότι: «Αυτόνομοι πράκτορες είναι υπολογιστικά συστήματα που κατοικούν σε ένα περίπλοκο δυναμικό περιβάλλον, έχοντας την δυνατότητα αλλά και την αίσθηση να ενεργούν αυτόνομα σε αυτό, ορίζοντας μια σειρά από στόχους ή καθήκοντα για τα οποία έχουν σχεδιαστεί». [Artificial Life Meets Entertainment: Lifelike Autonomous Agents] Επιπρόσθετα , η IBM ορίζει ένα κινητό πράκτορα ως: «Ευφυείς πράκτορες είναι οντότητες λογισμικού που διεξάγουν κάποιο σύνολο εργασιών για λογαριασμό ενός χρήστη ή άλλο πρόγραμμα με κάποιο βαθμό ανεξαρτησίας ή αυτονομίας, και με τον τρόπο αυτό, χρησιμοποιούν κάποια γνώση ή την εκπροσώπηση των στόχων ή τις επιθυμίες του χρήστη ».

Ο όρος “πράκτορας” δεν είναι μία σαφώς καθορισμένη έννοια αφού έχουν προταθεί από διάφορους ερευνητές πολλοί διαφορετικοί ορισμοί. Έτσι, αυτό που θα κάνουμε στη συνέχεια είναι να περιγράψουμε τις κυριότερες ιδιότητες ενός “πράκτορα” έτσι ώστε να δοθεί μία γενική εικόνα της έννοιας αυτής. Τις ιδιότητες αυτές μπορούμε να τις κατατάξουμε σε δύο κατηγορίες: στις ιδιότητες που εντάσσονται στην ασθενή θεώρηση της έννοιας “πράκτορας” και στις ιδιότητες που εντάσσονται στην ισχυρή θεώρηση της έννοιας “πράκτορας”. Οι ιδιότητες που ανήκουν στην ασθενή θεώρηση είναι αυτές που είναι γενικώς παραδεκτό από τους περισσότερους ερευνητές ότι πρέπει να έχει ένας πράκτορας. Αντιθέτως, οι ιδιότητες της ισχυρής θεώρησης δεν είναι γενικώς παραδεκτό ότι πρέπει να υπάρχουν σε έναν πράκτορα.

Η ασθενής θεώρηση του όρου “πράκτορας”: Οι ιδιότητες που κατά γενική παραδοχή πρέπει να έχει ένας πράκτορας είναι οι παρακάτω:

1. Αυτονομία (autonomy): οι πράκτορες λειτουργούν χωρίς την ανθρώπινη παρέμβαση ώστε να μπορούν να ελέγχουν τις πράξεις τους και την εσωτερική τους κατάσταση.
2. Κοινωνική ικανότητα (social ability): οι πράκτορες έχουν τη δυνατότητα να επικοινωνούν με άλλους πράκτορες μέσω μίας γλώσσας πρακτόρων.
3. Ικανότητα αντίδρασης (reactivity): οι πράκτορες είναι ικανοί να αντιλαμβάνονται το περιβάλλον τους και να αντιδρούν ανάλογα με τις αλλαγές που γίνονται σ’ αυτό.
4. Χρονική συνέχεια (temporal continuity): οι πράκτορες “τρέχουν” συνεχώς, είτε είναι ενεργοί στο προσκήνιο, είτε στο παρασκήνιο (background).
5. Προσανατολισμός σε στόχο (proactivity): οι πράκτορες δεν αντιδρούν απλώς στα ερεθίσματα του περιβάλλοντος αλλά μπορούν να δρουν προσανατολισμένοι σε έναν στόχο.

Η ισχυρή θεώρηση του όρου “πράκτορας”: Για πολλούς ερευνητές ο όρος “πράκτορας” έχει πιο σαφές και συγκεκριμένο νόημα. Αυτοί θεωρούν τον πράκτορα ως ένα υπολογιστικό σύστημα που, εκτός από τις ιδιότητες που δόθηκαν παραπάνω, έχει και ιδιότητες που συναντάμε στους ανθρώπους, όπως γνώση, πεποίθηση, υποχρέωση ή ακόμα και συναισθήματα. Μερικές από τις ιδιότητες που εντάσσονται στην ισχυρή θεώρηση του πράκτορα είναι οι παρακάτω:

1. Ικανότητα Μετακίνησης (mobility): Είναι η ικανότητα του πράκτορα να μετακινείται σε διάφορες τοποθεσίες μέσα σε ένα δίκτυο.
2. Καλοσύνη (benevolence): Οι πράκτορες δεν έχουν αλληλοσυγκρουόμενους στόχους, οπότε ο κάθε πράκτορας προσπαθεί να κάνει ό,τι του ζητηθεί.
3. Ορθολογικότητα (rationality): Ο πράκτορας ενεργεί έτσι ώστε να επιτύχει τους στόχους του.
4. Προσαρμοστικότητα (adaptivity): Η ικανότητα του πράκτορα να προσαρμόζεται στις συνθήκες του περιβάλλοντος. [4]

2.3.2 Κινητοί πράκτορες

Οι Κινητοί πράκτορες (Mobile agents) δεν έχουν δημιουργηθεί για να κάνουν κάτι καινούργιο αλλά η χρήση τους απλά έκανε τα πράγματα να δουλεύουν καλύτερα. Το κύριο κίνητρο για την ανάπτυξη των κινητών πρακτόρων ήταν από τα σημεία διαχείρισης συμφόρησης των δικτύων λόγω της άσκοπης χρήσης του εύρους ζώνης που γινόταν. Οι

κινητοί πράκτορες μπορούν να δουλέψουν σε χαμηλές ταχύτητες συνδέσεων, επειδή οι χρήστες επιτρέπουν στους πράκτορες να περιφέρονται για λογαριασμό τους και να περιμένουν τα αποτελέσματα τους. Εκτός αυτού, οι κινητοί πράκτορες έχουν ασύγχρονες δυνατότητες συναλλαγής, επιτρέποντάς τους να έχουν «νοημοσύνη», δηλαδή μπορούν να πάρουν αποφάσεις, να επεξεργάζονται πληροφορίες και να επικοινωνούν με άλλους παράγοντες.



Εικόνα 4 Κινητοί πράκτορες [5]

Όπως είναι φυσικό, πλεονεκτούν σε σχέση με στατικούς πράκτορες όσον αφορά ένα μεγάλο αριθμό -δικτυακών κυρίως- εργασιών. Οι κινητοί πράκτορες δεν μένουν προσκολλημένοι στο σύστημα απ' όπου ξεκίνησαν να εκτελούνται. Είναι ελεύθεροι να κινούνται ανάμεσα σε συστήματα του δικτύου. Το συγκριτικό τους πλεονέκτημα είναι ότι παρότι δημιουργούνται σε συγκεκριμένο περιβάλλον εκτέλεσης, έχουν τη δυνατότητα να μεταφέρουν την κατάσταση και τον κώδικα τους μαζί τους σε διαφορετικό περιβάλλον εκτέλεσης, στο οποίο συνεχίζουν την εκτέλεση τους. Η ικανότητα τους να «ταξιδεύουν», επιτρέπει στους κινητικούς πράκτορες να μεταπηδούν σε συστήματα που περιλαμβάνουν αντικείμενα με τα οποία οι πράκτορες επιθυμούν να έρθουν σε επαφή, και στη συνέχεια να επωφεληθούν του γεγονότος ότι βρίσκονται στο ίδιο σύστημα ή δίκτυο με το συγκεκριμένο αντικείμενο.

Πλεονεκτήματα Κινητών Πρακτόρων

Σημαντικά πλεονεκτήματα που προκύπτουν από τη χρήση των κινητών πρακτόρων είναι:

- Η μείωση του φόρτου δικτύου : Τα καταναμημένα συστήματα συχνά βασίζονται σε επικοινωνιακά πρωτόκολλα που εμπλέκουν πολλαπλές αλληλεπιδράσεις για να πετύχουν ένα δεδομένο σκοπό. Αυτό το γεγονός έχει ως αποτέλεσμα να προκαλείται

μεγάλη κυκλοφορία στο δίκτυο. Οι κινητοί πράκτορες επιτρέπουν τη μεταφορά ανάλογων αλληλεπιδράσεων καθώς και ακατέργαστων δεδομένων σε ένα απομακρυσμένο host όπου μπορεί να γίνει τοπική επεξεργασία αυτών των πληροφοριών. Έτσι επέρχεται τελικά μείωση του συνολικού δικτυακού φορτίου.

- Η αντιμετώπιση δικτυακών καθυστερήσεων : Τα συστήματα αληθινού χρόνου, όπως είναι τα ρομπότ που συμμετέχουν στην παραγωγική διαδικασία, οφείλουν να ανταποκρίνονται άμεσα στις αλλαγές του περιβάλλοντος. Ο έλεγχος τέτοιων συστημάτων με τη βοήθεια συμβατικών βιομηχανικών δικτύων μεγάλου μεγέθους εμπιέχει σημαντικές καθυστερήσεις, οι οποίες σε καμία περίπτωση δεν είναι επιτρεπτές. Οι κινητοί πράκτορες προσφέρουν λύση στο συγκεκριμένο πρόβλημα, καθώς μπορούν να αποσταλούν από ένα κεντρικό ελεγκτή και να δράσουν τοπικά, εκτελώντας αποκλειστικά τις οδηγίες του ελεγκτή.
- Η ενσωμάτωση πρωτοκόλλων : Για τη μεταφορά δεδομένων σε ένα κατανεμημένο σύστημα, ο κάθε host περιέχει κώδικα ο οποίος υλοποιεί τα αναγκαία πρωτόκολλα για εισερχόμενα και εξερχόμενα δεδομένα. Συχνά όμως αυτά τα πρωτόκολλα πρέπει να αναβαθμιστούν για να ανταποκριθούν σε νέες απαιτήσεις αναφορικά με θέματα ασφάλειας και αποδοτικότητας. Αυτή η διαδικασία αναβάθμισης είναι ιδιαίτερα επίπονη. Οι κινητοί πράκτορες λύνουν το συγκεκριμένο πρόβλημα «ταξιδεύοντας» σε απομακρυσμένους hosts με σκοπό να δημιουργήσουν «κανάλια επικοινωνίας» βασισμένα σε ιδιόκτητα πρωτόκολλα.
- Η ασύγχρονη και αυτόνομη εκτέλεση : Συχνά «κινητές» συσκευές είναι υποχρεωμένες να βασίζονται σε ακριβές και ευαίσθητες δικτυακές συνδέσεις. Αυτό έχει ως συνέπεια εργασίες που απαιτούν διαρκώς ανοικτή επικοινωνία ανάμεσα σε μια κινητή συσκευή και σε ένα σταθερό δίκτυο να μην είναι οικονομικά ή τεχνικά εφικτές. Τέτοιες εργασίες μπορούν να προσκολληθούν σε κινητούς πράκτορες, οι οποίοι στη συνέχεια θα αποσταλούν στο δίκτυο. Αφού ελευθερωθούν, οι κινητοί πράκτορες ανεξαρτητοποιούνται από τη διαδικασία που τους δημιούργησε και μπορούν να λειτουργήσουν ασύγχρονα και αυτόνομα. Η κινητή συσκευή πάντως έχει τη δυνατότητα σε κάποια μελλοντική χρονική στιγμή να συλλέξει τον πράκτορα, αφού πρώτα αυτός ολοκληρώσει την αποστολή του.
- Η δυναμική προσαρμοστικότητα : Οι κινητοί πράκτορες έχουν τη δυνατότητα να «αισθάνονται» το περιβάλλον εκτέλεσης στο οποίο κινούνται και να αντιδρούν αυτόνομα στις όποιες αλλαγές. Ακόμα, για τη βέλτιστη επίλυση ενός προβλήματος, μια ομάδα κινητών πρακτόρων μπορεί να διαμοιραστεί με τέτοιο τρόπο ανάμεσα στους διαφορετικούς host επιτυγχάνοντας τον καλύτερο καταμερισμό των ευθυνών.

- Η σταθερότητα και η ανοχή σε σφάλματα : Η ικανότητα των κινητών πρακτόρων να αντιμετωπίζουν δύσκολες καταστάσεις με δυναμική συμπεριφορά, επιτρέπει την εύκολη δημιουργία κατανεμημένων συστημάτων που είναι σταθερά και ανθεκτικά σε σφάλματα. Έτσι όταν ένας host καταρρεύσει, όλοι οι πράκτορες που εκτελούνται σε αυτόν θα δεχτούν τις ανάλογες προειδοποιήσεις και θα τους δοθεί ο απαραίτητος χρόνος ώστε να αποχωρίσουν και να συνεχίσουν την εκτέλεση τους σε διαφορετικό host του δικτύου
- Είναι από τη φύση τους ετερογενείς : Τα δίκτυα υπολογιστών είναι ετερογενή τόσο από πλευράς λογισμικού όσο και από πλευράς υλικού. Επειδή όμως οι κινητοί πράκτορες δεν επηρεάζονται από τις διαφορετικές επιλογές σε θέματα λογισμικού ή υλικού και εξαρτώνται μόνο από το περιβάλλον εκτέλεσης, παρέχουν τις ιδανικές συνθήκες για απρόσκοπτη ολοκλήρωση συστημάτων.

Μειονεκτήματα Κινητών Πρακτόρων

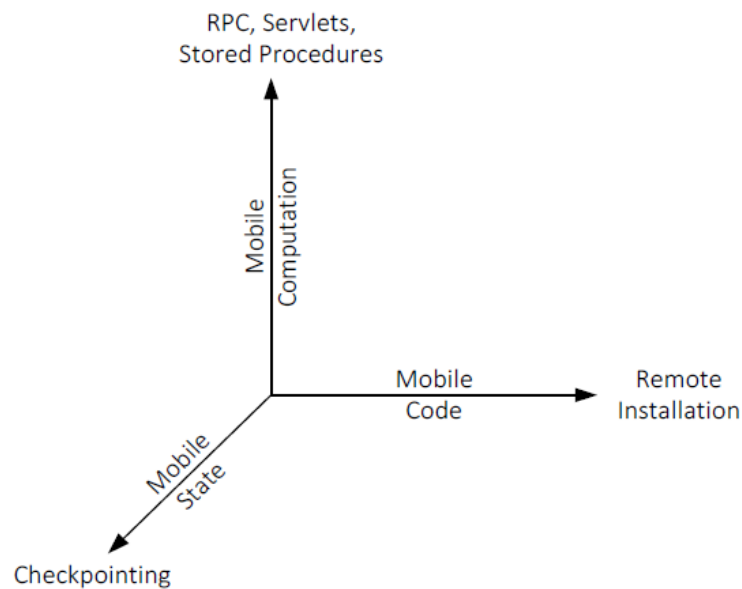
Παρόλα τα σημαντικά πλεονεκτήματα της τεχνολογίας των κινητών πρακτόρων υπάρχουν και κάποια προβλήματα που πρέπει να αντιμετωπιστούν ώστε να μπορέσει αυτή η τεχνολογία να εφαρμοστεί ευρέως. Τα σημαντικότερα μειονεκτήματα είναι:

- Η ελλιπής ασφάλεια: Ένας κινητός πράκτορας μπορεί να μεταφέρει εμπιστευτικές πληροφορίες που αφορούν το χρήστη του. Επίσης, ένας ιός που ταξιδεύει στο διαδίκτυο μπορεί να προσβάλλει ένα σύστημα πρακτόρων που θα τον έχει δεχτεί σαν κινητό πράκτορα.
- Η διαλειτουργικότητα: Ως τώρα τα πιο πολλά συστήματα λειτουργούσαν σε κλειστά περιβάλλοντα, όπου τα είδη των υπολογιστών και των πρακτόρων που τρέχουν σε αυτούς είναι γνωστά από τη φάση της κατασκευής. Για να χρησιμοποιηθεί η τεχνολογία των κινητών πρακτόρων, ο κώδικας πρέπει να μπορεί να εκτελεστεί και εκτός της πλατφόρμας για την οποία είναι σχεδιασμένος.
- Η ασύμφορη λειτουργία: Σε πολλές περιπτώσεις η χρήση κινητών πρακτόρων οδηγεί σε καλή απόδοση των συστημάτων τους, αλλά οι περισσότερες εφαρμογές που θα μπορούσαν να τους αξιοποιήσουν γίνονται σήμερα με αρκετά αποτελεσματικό τρόπο από τις παραδοσιακές τεχνικές. Είναι πιθανό το κόστος μεταφοράς του πράκτορα να είναι συγκρίσιμο ή ακόμα και να υπερβαίνει εκείνο της αλληλεπίδρασης από απόσταση.
- Τα προβλήματα χρήσης: Πολλοί δικτυακοί τόποι κερδίζουν αξιοσημείωτα ποσά μέσω των διαφημίσεων που φιλοξενούν. Αν αυτοί οι τόποι επιτρέψουν στους

κινητούς πράκτορες να έχουν πρόσβαση στα δεδομένα τους, ο αριθμός των χρηστών που τους επισκέπτονται θα μειωθεί. Επίσης, αν ο χρήστης χρησιμοποιεί μια υπηρεσία μέσω κινητών πρακτόρων, ο παροχέας της υπηρεσίας δεν έχει πλήρη έλεγχο πάνω σε αυτή. Ένας κακά σχεδιασμένος πράκτορας μπορεί να δυσφημίσει μία άμεμπτη υπηρεσία και να στερήσει οικονομικά έσοδα από τον παροχέα. [6]

Δυνατότητες κινητικότητας πρακτόρων

Η κινητικότητα των πρακτόρων, όπως παρατηρούμε και στο σχήμα μας προφέρει τις παρακάτω δυνατότητες που ξεχωρίζουν:



Εικόνα 5 Κινητικότητα

- Κινητικότητα κώδικα (Mobile Code)

Ο κινητός κώδικας επιτρέπει στον εκτελέσιμο κώδικα να μεταφερθεί σε ένα νέο σύστημα και μπορεί να χρησιμοποιήσει το μοντέλο push ή pull (ανάκτηση ή παροχή). Οι κινητοί πράκτορες που χρησιμοποιούν τη λειτουργία της ανάκτησης έχουν τη δυνατότητα να οδηγήσουν στην δημιουργία νέου κώδικα. Ο κώδικας μπορεί να είναι δυαδικός ή πηγαίος. Τα πλεονεκτήματα του κινητού κώδικα δίνουν την δυνατότητα δυναμικών αλλαγών όπως πρόσθεση/αλλαγή/ενημέρωση της πλατφόρμας ή κατάργηση κώδικα που δεν χρειάζεται πλέον. Η δυνατότητα όμως αυτή μπορεί να δημιουργήσει προβλήματα ασφάλειας λόγω της χρήσης μη έμπιστου ή ελεγχόμενου κώδικα (Ο κώδικας μπορεί να είναι ύποπτος ή προβληματικός).

- Κινητός υπολογισμός (Mobile Computation)

Η χρήση του κινητού υπολογισμού φέρνει την επανάσταση στην απομακρυσμένη επεξεργασία δεδομένων. Μερικά χαρακτηριστικά παραδείγματα είναι τα συστήματα RPC, RSH, RMI, Servlets, Stored Procedures, CORBA. Η δυνατότητα αυτή επιτρέπει σε ένα σύστημα να κάνει υπολογισμούς και σε ένα άλλο σύστημα χρησιμοποιώντας τους πόρους του απομακρυσμένου συστήματος (επεξεργαστή. μνήμη). Ακόμα, επιτρέπει να έχει πρόσβαση σε αρχεία ή βάσεις δεδομένων.

- Κινητή κατάσταση (Mobile State)

Η κινητή κατάσταση δίνει την δυνατότητα της παύσης μίας εκτελέσιμης διεργασίας, της μεταφοράς της σε έναν άλλο σύστημα και της συνέχισης της εκτέλεσης από το σημείο που έγινε η παύση.[5]

3

Mobile Learning

3.1 Ορισμός και χαρακτηριστικά mobile learning

3.1.1 Ορισμός mobile learning

Ο όρος mobile learning ή αλλιώς mlearning μεταφράζεται σαν κινητή μάθηση. Η κινητή μάθηση, αν και έχει συνδεθεί κυρίως με τα κινητά τηλέφωνα, πραγματικότητα καλύπτει ένα ευρύτερο φάσμα συσκευών όπως PDA, φορητούς υπολογιστές και tablet. Η κινητή μάθηση, στο πλαίσιο αυτό, μπορεί να οριστεί ως η *διαδικασία ηλεκτρονικής μάθησης που χρησιμοποιεί κινητές συσκευές για να επιτρέψει στον χρήστη να προσπελάσει το μαθησιακό περιβάλλον καθώς μετακινείται από σημείο σε σημείο*. Συμβαίνει αρκετά συχνά, ένας χρήστης να έχει πρόσβαση σε ηλεκτρονικές συσκευές που στην ταχύτητα, τη δύναμη επεξεργασίας, την δυνατότητα παρακολούθησης και άλλα χαρακτηριστικά εξόδου. Στο πλαίσιο αυτό, η κινητή μάθηση έχει πλέον ανοίξει πολλές δυνατότητες εξ αποστάσεως εκπαίδευσης και μάθησης γενικότερα για άτομα που βρίσκονται σε απομακρυσμένες περιοχές και ενδέχεται να μην έχουν πρόσβαση σε παραδοσιακά συστήματα επικοινωνιών, όπως ενσύρματα, για να συνδεθούν στο Διαδίκτυο. Όπως είναι αυτονόητο από την ορολογία, ένα από τα θεμελιώδη πλεονεκτήματα αυτής της μορφής μάθησης είναι η ικανότητα του χρήστη να διατηρεί την πρόσβαση όταν μετακινείται από ένα μέρος σε άλλο, δεδομένου ότι οι φορητές συσκευές μέσω της φορητότητάς τους επιτρέπουν να παραμένεις συνδεδεμένος.

3.1.2 Χαρακτηριστικά του *mobile Learning*

Τα συστήματα κινητής μάθησης περιγράφονται από τα παρακάτω χαρακτηριστικά:

- Επιτρέπουν τους χρήστες να έχουν πρόσβαση στους πόρους των μαθημάτων ανεξάρτητα από τον χώρο και τον χρόνο.
- Επιτρέπουν πιστοποιημένους χρήστες να έχουν πρόσβαση στο ίδιο το σύστημα.
- Επιτρέπει τους χρήστες να έχουν πρόσβαση στα δεδομένα με διαφορετικές μορφές(κείμενο, ήχος, βίντεο εικόνα) . Έχοντας αυτό στο μυαλό του και συνδυάζοντας το με ένα πραγματικό περιστατικό, πχ με την επίλυση ενός περίπλοκου κόμπου και με την αναζήτηση πληροφοριών για την επίτευξη, αντιλαμβάνεται κανείς την υποστήριξη στην πολυεπίπεδη σκέψη.
- Επιτρέπει τους χρήστες να επαναχρησιμοποιήσουν το μαθησιακό υλικό.
- Επιτρέπει τους χρήστες να εκτελούν τα καθήκοντά που τους ορίζονται ως συνιστώσα της εκπαίδευσης.
- Προσφέρει ευέλικτο περιβάλλον μέσα στο οποίο μπορούν να προστεθούν υπηρεσίες και συστατικά.[7]
- Πρωτοβουλία στην απόκτηση γνώσεων: Πολλές φορές η ανάγκη για γνώση μπορεί να προκύψει από τον ίδιο τον μαθητευόμενο. Στην κινητή μάθηση υπάρχει η δυνατότητα για άμεση κάλυψη της ανάγκης με ότι ποιο κοντινή στο ερώτημα πληροφορία υπάρχει ήδη αποθηκευμένη. [8].
- Μεγαλύτερος βαθμός διαδραστικότητας καθώς εκτός ότι ο χρήστης μπορεί να επιχειρήσει την επικοινωνία από οποιαδήποτε σημείο, ο συνομιλητής του, καθηγητής ή μαθητής, μπορεί και αυτός να απαντήσει στην επικοινωνία μέσα από κινητή συσκευή.
- Αμεσότητα στην εμπειρία: Τα προβλήματα και η ανάγκη γνώσεων για την λύση τους προκύπτουν σε μεγάλο βαθμό στη καθημερινή μας δραστηριότητα. Η παροχή της γνώσης και η άμεση λύση σύνδεσης με το πρόβλημα μπορεί να γίνει ευκολότερα μέσω της κινητής μάθησης.

<u>Ηλεκτρονική μάθηση</u>	<u>Κινητή μάθηση</u>
<u>Ορολογία</u>	
Υπολογιστής	Κινητή συσκευή
Εύρος Ζώνης	GPRS, 4G-LTE, Wi-Fi, Wi-Max
Διαδραστικό	Αυθόρμητο
Πολυμεσικό	Σκοπικό
Εξ αποστάσεως μάθηση	Κατά περίπτωση μάθηση
<u>Παιδαγωγικές αλλαγές</u>	
Οδηγίες που κυρίως βασίζονται στο κείμενο και στην γραφική απεικόνιση	Οδηγίες που βασίζονται στην φωνή, στην γραφική απεικόνιση, σε κινούμενη εικόνα και εκμετάλλευση αισθητήρων
Πρόσβαση στο υλικό μάθησης μέσω του internet	Πρόσβαση στο υλικό μέσω κινητής συσκευής ή την απαιτούμενη στιγμή
<u>Επικοινωνία με τον μαθητευόμενο</u>	
Ασύγχρονη επικοινωνία (απαιτείται περιοδικός έλεγχος για νέα email ή μηνύματα)	Άμεση ειδοποίηση για νέα SMS ή email
Παθητική	Άμεση
Προγραμματισμένη	Αυθόρμητη
<u>Επικοινωνία μεταξύ μαθητευόμενων</u>	
Πρόσωπο με πρόσωπο	Ευέλικτη
Συνηθισμένη μέσω τηλεδιασκέψεων	Δύναται και μέσω τηλεδιασκέψεων
Αποστολή και λήψη email	24/7 άμεση
Σε εξειδικευμένη τοποθεσία	Χωρίς περιορισμούς στην τοποθεσία
Πιθανή μετακίνηση για εύρεση σημείου πρόσβασης στο ίντερνετ	Καμία ανάγκη μετακίνησης καθώς παρέχεται ασύρματη πρόσβαση

Πίνακας 1 Ηλεκτρονική Μάθηση και Κινητή Μάθηση

Σύμφωνα με την διαχωρισμό που γίνεται στον πίνακα 1, η σύγκριση γίνεται σε 3 τομείς. Στην ορολογία, στις αλλαγές στην παιδαγωγική μέθοδο και στην επικοινωνία.

Η επικοινωνία χωρίζεται σε επικοινωνία των διδασκόντων με τους μαθητές και επικοινωνία των μαθητών μεταξύ τους. Υπάρχουν διάφοροι μηχανισμοί επικοινωνίας των κινητών συσκευών αλλά 3 από αυτούς εξυπηρετούν την κινητή μάθηση.

- Σαν πρώτος μπορεί να αναφερθεί η δυνατότητα επικοινωνίας μέσω φωνής που αποτελούσε το κύριο μηχανισμό για την επικοινωνία των μαθητευόμενων μεταξύ τους και με τον καθηγητή στην απόκτηση της γνώσης.
- Δεύτερος είναι η δυνατότητα πρόσβασης σε πληροφορίες που έχουν αναρτηθεί σε κάποιον διαδικτυακό τόπο από οποιοδήποτε σημείο μπορεί να επικοινωνήσει η κινητή συσκευή.
- Τρίτος είναι η δυνατότητα μάθησης μέσω από την αποστολή σύντομων γραπτών μηνυμάτων(SMS, υπηρεσίες άμεσων μηνυμάτων) ή email που μπορούν είτε να μεταδώσουν γνώση είτε να προβούν σε διαδικασίες υπενθύμισης.

Μια τέτοια διαδικασία είχε αναπτυχθεί από την εταιρία British Broadcast Company σε συνεργασία με την εταιρία Sina. Για να κατανοήσει κανείς τα κέρδη που μπορεί να έχει η κινητή μάθηση σε σχέση με την στην συμβατική ηλεκτρονική μάθηση θα αναφέρουμε ένα πολύ χαρακτηριστικό σενάριο που δανειζόμαστε από το [9] του Robert Yu-Liang Ting.

Έστω ότι είμαστε σε ένα πολυκατάστημα και πρέπει ένας νέος υπάλληλος να μάθει τα προϊόντα και τα χαρακτηριστικά τους. Στο συμβατικό τρόπο εκμάθησης με ηλεκτρονική μάθηση καλείτε να καθίσει μπροστά από μια οθόνη υπολογιστή σε κάποιο δωμάτιο και να αναγνώσει- αποστηθίσει τα απαραίτητα μέσα από φωτογραφίες των προϊόντων δίπλα από το ανάλογο κείμενο. Σε ένα σύστημα κινητής μάθησης, θα μπορούσε ο υπάλληλος, έχοντας μια κινητή συσκευή και ένα scanner ,να σκανάρει τα προϊόντα και να του εμφανίζονται στην οθόνη τα χαρακτηριστικά του προϊόντος μαζί με την απαραίτητη γνώση για την λειτουργία του. Έτσι ο μαθητευόμενος έχει άμεση γνώση για το αντικείμενο που καλείτε να αποστηθίσει. Επιπρόσθετα, θα μπορούσαμε να προσθέσουμε στο παραπάνω σενάριο και την δυνατότητα του συστήματος που έχει ο νέος υπάλληλος να κάνει συγκρίσεις των προϊόντων σκανάροντας τα διαδοχικά. Καταφέρνει έτσι συντομότερα και χωρίς μεγάλη εμπειρία να βοηθήσει έναν πελάτη στην επιλογή ενός προϊόντος αναμεσα σε 1 ή περισσότερα έχοντας το σύστημα κινητής μάθησης να του κάνει την σύγκρισή αυτών των προϊόντων μεταξύ τους προσφέροντας την δυνατότητα εκμάθησης παράλληλα με την εργασία. Αυτό μας δείχνει ότι η κινητή μάθηση μπορεί να είναι αρκετά ποιο ευέλικτη και άμεση από την παραδοσιακή ηλεκτρονική μάθηση.

Σε μια σύντομη αναφορά μας καταγράφουμε παρακάτω κάποιες απαιτήσεις που θα πρέπει να καλύψει ο δημιουργός ενός mobile learning συστήματος:

- Φορητότητα. Θα πρέπει η εφαρμογή που θα δημιουργηθεί να έχει την δυνατότητα λειτουργίας σε όσον το δυνατόν περισσότερες διαφορετικές πλατφόρμες λειτουργικών συστημάτων.
- Περιορισμένες απαιτήσεις σε κατανάλωση ενέργειας και χρήση πόρων. Στόχος μιας εφαρμογής κινητής συσκευής είναι να μην αναλαμβάνει η ίδια εργασίες επεξεργασίας

δεδομένων αλλά να αναθέτει αυτή την «δαπανηρή» υπολογιστικά διαδικασία σε εξειδικευμένους διακομιστές. Θα πρέπει να ενεργεί κυρίως ως συσκευή γραφικής απεικόνισης, χωρίς πολύπλοκες εργασίες.

- Ιδιότητες γραφικής απεικόνισης. Επιθυμητό είναι η εφαρμογή να διατηρεί ίδιο το γραφικό περιβάλλον μεταξύ εκτέλεσης σε διαφορετικές πλατφόρμες. Αυτό μπορεί να επιτευχθεί αν κατά την δημιουργία γίνεται σαφής διαχωρισμός του επιπέδου γραφικής απεικόνισης από την ανάλυση και επεξεργασία των δεδομένων.
- Ευέλικτο σχεδιαστικά και επεκτάσιμο. Το σύστημα που θα δημιουργηθεί θα πρέπει να σχεδιαστεί με τρόπο ώστε να αποτρέπεται η αποτυχία λειτουργίας αν κάποια πλευρά αποτύχει να επικοινωνήσει. Σημαντικό συστατικό αποτελεί και η δυνατότητα ενσωμάτωσης νέων χαρακτηριστικών χωρίς την ανάγκη εκ νέου δημιουργίας του όλου οικοδομήματος.
- Προσωρινή μνήμη δεδομένων. Η χρήση προσωρινών αποθηκών δεδομένων που ενδέχεται να χρησιμοποιηθούν εκ νέου σε σύντομο χρονικό διάστημα επιτυγχάνει την εξοικονόμηση ενέργειας, πόρων και μεταφερόμενων δεδομένων μέσω δικτύου.
- Αυτοματισμοί. Η πρόβλεψη αυτοματισμών σε συγκεκριμένες ενέργειες του χρήστη εξυπηρετεί το χαρακτηριστικό της εξοικονόμησης ενέργειας που αναφέραμε και προσφέρει ένα χαρακτήρα ευφυΐας στο σύστημα μας.

3.2 Μοντέλα και αρχιτεκτονικές *mobile learning*

3.2.1 Μοντέλα

Ακολουθεί μια καταγραφή των πιο βασικών μοντέλων κινητής μάθησης που προτείνονται στην βιβλιογραφία του *mobile learning*.

3.2.1.1 Παιδαγωγικό μοντέλο ανεπτυγμένο για κινητή μάθηση

Ένα θεωρητικό παιδαγωγικό μοντέλο για κινητή μάθηση με το όνομα AEFIRIP (Activation-Ενεργοποίηση, Externation-Εξωτερική ανάθεση, Focusing- Συγκέντρωση, Interpretations-Ερμηνίες, Reflection-Αντανάκλαση, Information Processing- Επεξεργασία πληροφορίας) αναπτύχθηκε στο [10]. Το μοντέλο αυτό βασίζεται στα σύγχρονα μοντέλα ηλεκτρονικής μάθησης και σε κοινωνικοπολιτικές θεωρίες όπως η προοδευτική ερώτηση, η ενεργοποίηση οδηγίων και στην μάθηση βασισμένη σε προβλήματα. Το μοντέλο εστιάζει στα χαρακτηριστικά της κινητής μάθησης και στις εξατομικευμένες πρακτικές που μπορεί να παρέχει αυτή η διαδικασία στο μοναδικό περιβάλλον του κάθε χρήστη . Τα βήματα του μοντέλου που παρουσιάζονται στην Πίνακα 2 , δημιουργήθηκαν για να δομήσουν την διαδικασία της μάθησης και να υποστηρίξουν τις εξατομικευμένες ενέργειες που απαιτούνται. Εκτός από την δόμηση της γνώσης, εξατομικευμένες και γνωστικές διαδικασίες, όπως ευριστικά και λογικά συμπεράσματα τονίζονται σε μεγάλο βαθμό στο AEFIRIP. Οι δημιουργοί του μοντέλου αυτού το χρησιμοποίησαν για να αναπτύξουν τα χαρακτηριστικά ενός ευφυούς εργαλείου διδασκαλίας που ονομάζεται *Älykkö* (“κεφάλι αυγού”). Το *Älykkö* αποτελείται από ευφυής πράκτορες διδασκαλίας που παρέχουν ημιαυτόματη και αυτόματη διδασκαλία, καθώς και δείκτες που βασίζονται στην πρόοδο του μαθητή οι οποίοι ενεργοποιούν εξατομικευμένες διαδικασίες μάθησης για τον κάθε μαθητή ακόμα και χωρίς την παρουσία εικονικού καθηγητή.

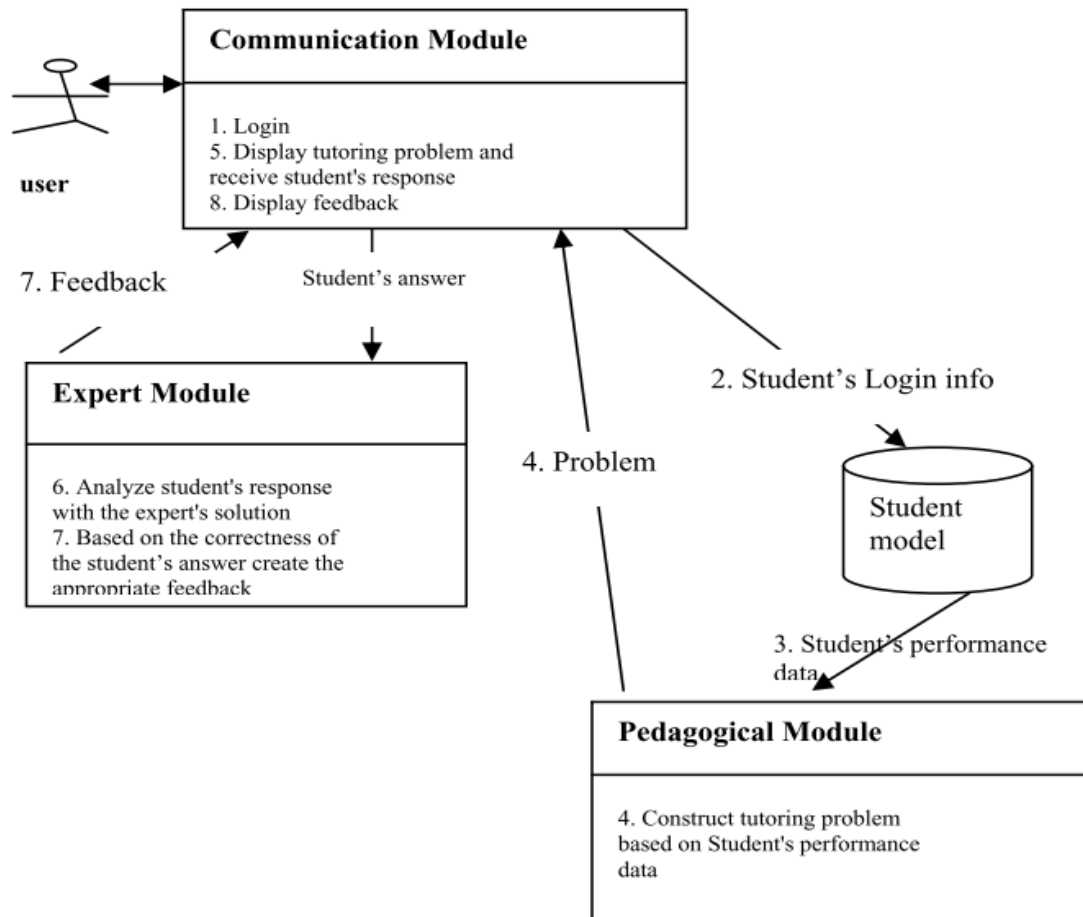
Στο μοντέλο AEFIRIP, η κινητή τεχνολογία δεν λογίζεται μόνο ως μεσολαβητής στις δραστηριότητες της μάθησης και της συνεργασίας αλλά επιπρόσθετα ως μία πλατφόρμα που παρέχει την δυνατότητα υποστήριξης και καθοδήγησής κατά την διαδικασία της μάθησης.

Φάση	Περιγραφή
1. Ενεργοποίηση	Ενεργοποίηση των προηγούμενων γνώσεων και γνωστικών στρατηγικών του μαθητή με τη δημιουργία πλαισίου ή π.χ. Παρουσιάζοντας τις αποκαλούμενες ερωτήσεις ενεργοποίησης
2. Εξωτερίκευση	Εξωτερίκευση των προηγούμενων γνώσεων και μοντέλων σκέψης του μαθητή. Οι μαθητές συνειδητοποιούν την προηγούμενη γνώση τους καθιστώντας την ορατή και εκθέτοντάς την σε προβληματισμό
3. Συγκέντρωση	Εστίαση της αντίληψης των μαθητών και της νοητικής επεξεργασίας σε ένα αυθεντικό μαθησιακό περιβάλλον σύμφωνα με τους στόχους της μαθησιακής κατάστασης (π.χ. εστιάζοντας ερωτήσεις ή αναθέσεις)
4. Ερμηνευση	Ρητές ερμηνείες που γίνονται από μαθητές με βάση την αντίληψη και τις προηγούμενες γνώσεις / γνωστικές στρατηγικές, καθώς και από τους κατά περίπτωση παράγοντες.
5. Αντανάκλαση	Αντανάκλαση των δικών τους ερμηνειών και των παραγόντων κατάστασης.
6. Πληροφορία	Η επεξεργασία πληροφοριών αποτελείται από υπορουτίνες όπως η επίλυση προβλημάτων, η ταξινόμηση, η σύγκριση κτλ.

Πίνακας 2 AEFIRIP [7]

3.2.1.2 Ευφυές μοντέλο μάθησης βασισμένο στον Ιστό

Μέσα από την εργασία του [11] και [12], ο συγγραφέας προτείνει ένα «ιδανικό ευφυές σύστημα διδασκαλίας»(Intelligent Tutoring System (ITS)) που έχει προκύψει από την έρευνα τεχνητής νοημοσύνης. Το μοντέλο βασίζεται στην βοήθεια στη μάθηση μέσω ηλεκτρονικού υπολογιστή(Computer Aided Learning (CAL)) όπου σύμφωνα με τον συγγραφέα χρησιμοποιείται στην διαδικασία της διδασκαλίας από το 1950. Παραδοσιακά η μάθηση μέσω ηλεκτρονικού υπολογιστή κάνει χρήση κυρίως οδηγών-μαθημάτων τα οποία ουσιαστικά είναι «προγραμματισμένες οδηγίες» που παρουσιάζονται σε φόρμες. Ο συγγραφέας περιγράφει το ITS ως νέου τύπου CAL. Το ITS δημιουργεί ένα μοντέλο που αποτελείται από την γνώση των ειδικών, την γνώση του μαθητευόμενου και τις αρχές διαπαιδαγώγησης. Όσο ο μαθητευόμενος εξελίσσεται, γίνεται σύγκριση την γνώσης των ειδικών με την τρέχουσα γνώση που έχει αποκτήσει ο μαθητευόμενος, με χρήση τεχνητής νοημοσύνης, παράγοντας δυναμικά την ακολουθία των οδηγιών που πρέπει να ακολουθήσει από εδώ και πέρα. Ακολουθεί σχηματικά η αναπαράσταση του μοντέλου.



Εικόνα 6 Μοντέλο ITS [7]

3.2.1.3 Μοντέλο πολλαπλών πρακτόρων Bee-gent

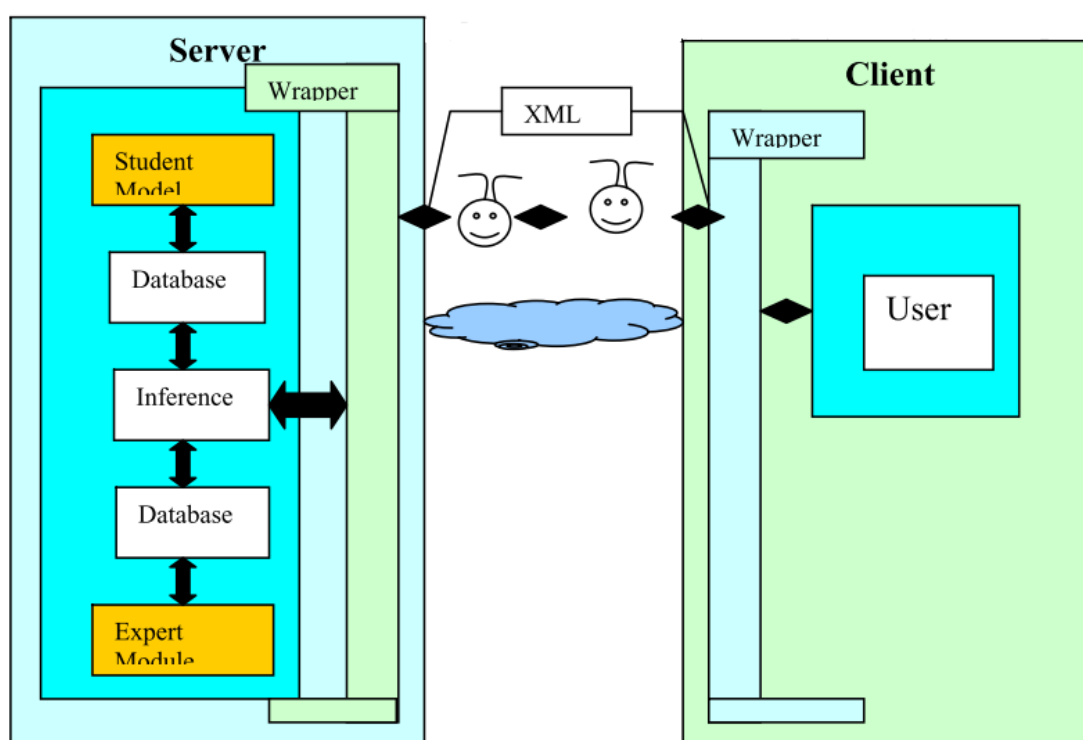
Ο Kinshuk στο έργο του στο σύγγραμμα [13] μας έδειξε πως οι ευφυείς πράκτορες μπορούν να χρησιμοποιηθούν στην βελτίωση του συστήματος κινητής μάθησης. Υποστηρίζει ότι το bee-gent framework μπορεί να χρησιμοποιηθεί για να υλοποιήσει τους κινητούς πράκτορες στο σύστημα. Σύμφωνα με τον συγγραφέα, η τεχνολογία του bee-gent έγινε αρχικά διαθέσιμη το 1999 από την Toshiba, ως ένα νέου τύπου framework καθαρής ανάπτυξης πράκτορα για την ανεπτυγμένου επιπέδου κοινωνία της οποίας το πλαίσιο επικοινωνίας είναι βασισμένο σε πολυπρακτορικό μοντέλο. Ο όρος bee-gent προκύπτει από το Bonding and Encapsulation Enhancement Agent που μεταφράζεται ελεύθερα ως η σύσφιξη και η ενθυλάκωση στην ενίσχυση του πράκτορα. Το bee-gent παρέχει εφαρμογές με αυτόνομη συμπεριφορά στο δίκτυο και το επιτυγχάνει δίνοντας σε ένα πράκτορα τις αρμοδιότητες αυτές. Επιπρόσθετα το μοντέλο bee-gent παρέχει επικοινωνία των εφαρμογών μέσω πρακτόρων που επιτυγχάνουν συνεργασία και επίλυση προβλημάτων. Το επιτυγχάνει αυτό μέσα από ευέλικτους και ανοιχτής δομής τρόπους, καθιστώντας το έτσι κατάλληλο για να προσφέρει συνεργατική

επεξεργασία σε μια ανεπτυγμένη δικτυακά κοινωνία. Το bee-gent framework αποτελείται από 2 τύπου πράκτορες: Τους πράκτορες περιτυλίγματος και τους πράκτορες μεσολάβησης.

Οι πράκτορες περιτυλίγματος χρησιμοποιούνται για να μετατρέψουν υπάρχουσες εφαρμογές σε πράκτορες. Είναι ικανοί να διαχειριστούν τις καταστάσεις μιας εφαρμογής καθώς «τυλίγουν» την εφαρμογή και την καλούν όταν κρίνουν ότι είναι απαραίτητο.

Οι πράκτορες-μεσολαβητές υποστηρίζουν την συνεργασία και την επικοινωνία μεταξύ εφαρμογών αναλαμβάνοντας την διαχείριση όλων των επικοινωνιών μεταξύ των εφαρμογών.

Οι πράκτορες μεσολάβησης μετακινούνται από ένα σημείο σε ένα άλλο όταν αλληλοεπιδρούν με απομακρυσμένους πράκτορες περιτυλίγματος.



Εικόνα 7 Bee-Gent Πράκτορες

Οι πράκτορες περιτυλίγματος χρησιμοποιούνται για να «περιτυλίξουν» τα συστήματά του διακομιστή και του πελάτη ενώ οι πράκτορες μεσολάβησης χρησιμοποιούνται για την επικοινωνία και την ανταλλαγή δεδομένων μεταξύ των πρακτόρων περιτυλίγματος. Τα κύρια σενάρια που λαμβάνου χώρα στο σύστημα είναι τα:

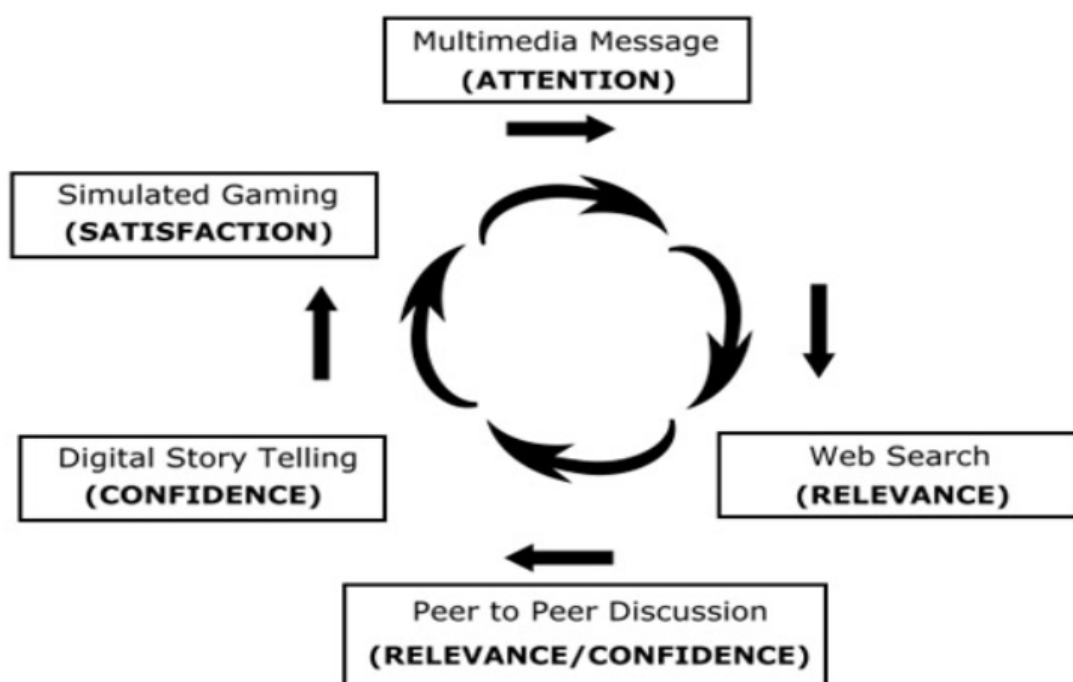
A. Από την πλευρά σου διακομιστή: η μηχανή συμπερασμού αλληλοεπιδρά με το μοντέλο του μαθητή και ειδικού μέσω της διεπαφής της βάσης δεδομένων. Αν κάτι πρέπει να αποσταλεί στην πλευρά του «πελάτη» τότε η μηχανή συμπερασμού ενημερώνει τον πράκτορα περιτυλίγματος με τις πληροφορίες που είναι προς αποστολή. Ο πράκτορας περιτυλίγματος δημιουργεί έναν ενδιάμεσο πράκτορα-μεσολαβητή, του μεταδίδει την απαραίτητη πληροφορία μαζί με ότι σχετικά προγράμματα απαιτούνται και τον αποστέλλει.

Αν ο ενδιαμέσος πράκτορας δεν μπορέσει να βρει τους επιθυμητούς στόχους- «πελάτες» τότε συνεχίζει την αναζήτηση ενημερώνοντας παράλληλα για την αποτυχία εύρεσης. Αν ο ενδιαμέσος πράκτορας εντοπίσει τον στόχο του τότε επικοινωνεί και ανταλλάσσει πληροφορίες με τον πράκτορα περιτυλίγματός του «πελάτη» και ολοκληρώνεται η διαδικασία ενημέρωσης της πλευράς του πελάτη.

Β.Η πλευρά του πελάτη: Αν απαιτείτε να αποσταλεί κάποιο αίτημα στην πλευρά του server, τότε ο πράκτορας περιτυλίγματος της πλευράς του πελάτη δημιουργεί έναν ενδιαμέσο πράκτορα που είναι ικανός να μεταφέρει την πληροφορία στον διακομιστή.

3.2.1.4 Το μοντέλο κινητής μάθησης του Shih

Ο συγγραφέας ανέπτυξε ένα μοντέλο κινητής μάθησης που βασίζεται στο μοντέλο υποκίνησης σχεδιασμού ARCS του Keller. Ο κύκλος μάθησης του ARCS περιλαμβάνει την προσοχή(Attention), την σχετικότητα(Relevance), την εμπιστευτικότητα(Confidence) και την ικανοποίηση(Satisfaction). Το μοντέλο κινητής μάθησης του Shih είναι μια παραλλαγή του ARCS μοντέλου, βασίζεται σε αυτό και στα χαρακτηριστικά που έχουν οι κινητές τεχνολογίες στην προώθηση και στην ενίσχυση στην επικοινωνία των ανθρώπων όπως φαίνεται στο παρακάτω σχήμα



Εικόνα 8 Κύκλος μάθησης του μοντέλου [7]

Το μοντέλο δημιουργήθηκε για να παρέχει υποστηρίξιξη εκπαιδευτικού σχεδιασμού στη κινητή μάθηση. Ο κύκλος μάθησης του Μοντέλου shih ακολουθεί τα παρακάτω βήματα:

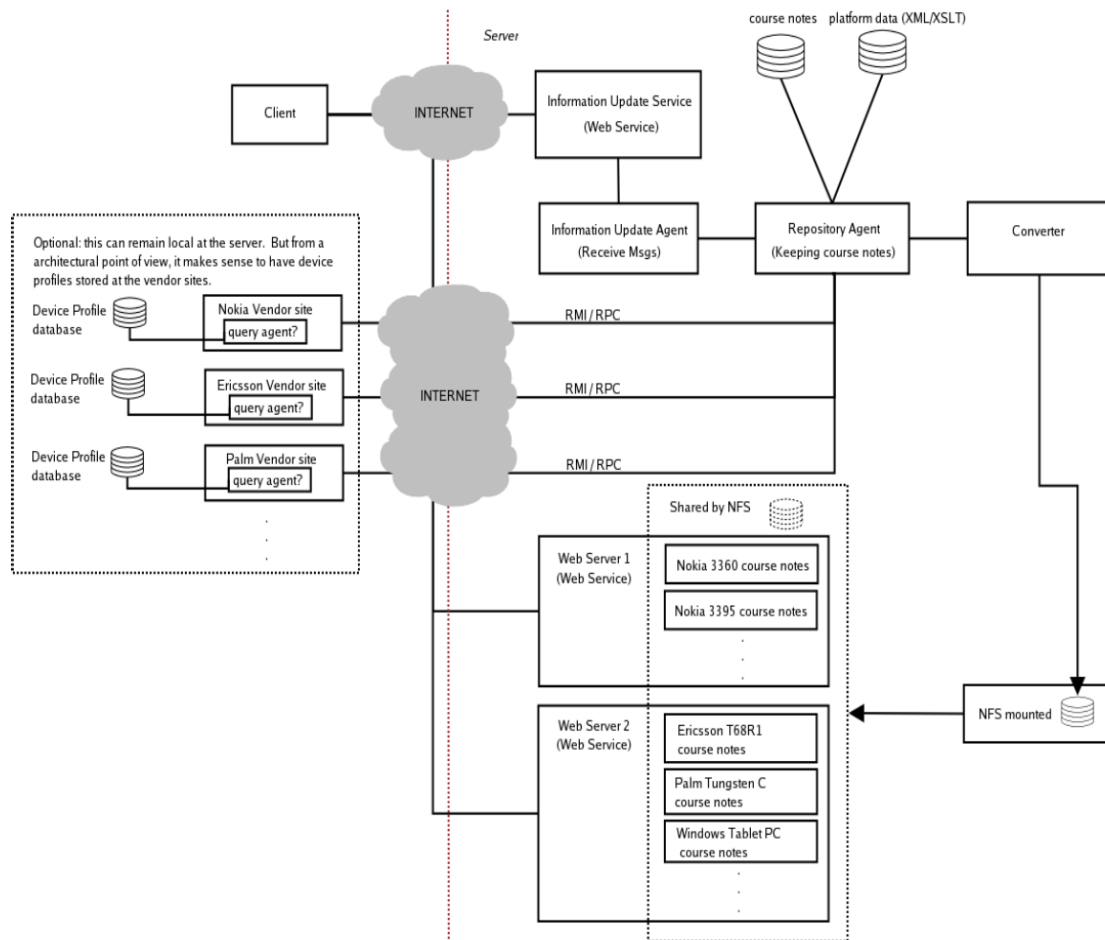
1. Αποστολή πολυμεσικών μηνυμάτων σε κινητές συσκευές με σκοπό την ενεργοποίηση και την παροχή κινήτρου στον μαθητευόμενο.
2. Αναζήτηση σχετικών πληροφοριών στο παγκόσμιο ιστό με την παροχή ενσωματωμένων υπερσυνδέσεων στα μηνύματα που λαμβάνουν στο τηλέφωνο οι μαθητευόμενοι.
3. Συζητώντας με συμμαθητές μέσω μηνυμάτων κειμένου, φωνής, εικόνων και βίντεο.
4. Παράγοντας μια ψηφιακή ιστορία με εικόνα ή ήχο εξιστορώντας το τι έμαθε ο καθένας.
5. Εφαρμογή αυτών που έμαθαν σε ένα περιβάλλον προσομοίωσης όπως ένα Online εκπαιδευτικό παιχνίδι.

Οι δημιουργοί του μοντέλου shih δηλώνουν ότι το μοντέλο κινητής μάθησης εμπνέεται από τη φιλοσοφία του κοινωνικού κονστρουκτιβισμού μέσω συλλογικών συζητήσεων και μέσω μιας θεωρίας εκμάθησης βασισμένη στην ψηφιακή αφήγηση μιας ιστορίας. Τελικώς μπορούμε να πούμε ότι το μοντέλο αυτό βιάζεται κυρίως στην υποδομή της επικοινωνίας κινητών συσκευών και θα ήταν περισσότερο ταιριαστό σε εφαρμογές με ανάμικτη μάθηση ή και σε περιβάλλοντα καθαρής μάθησης με κινητές συσκευές.

3.2.2 Αρχιτεκτονικές

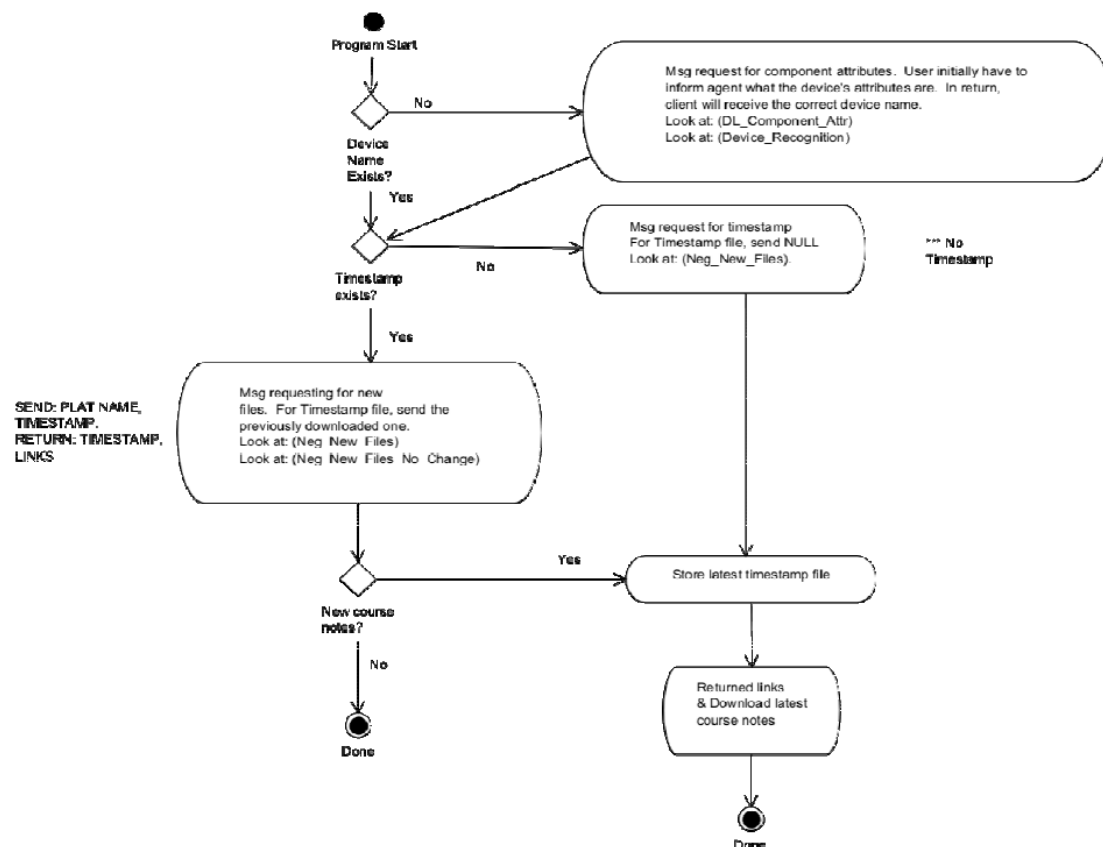
3.2.2.1 Αρχιτεκτονική από τους Ally, Mohamed - Lin, Fuhua - McGreal, Rory - Woo, Brian

Η αρχιτεκτονική που προτείνετε από τους Ally, Mohamed - Lin, Fuhua - McGreal, Rory - Woo, Brian στο [14] διαχωρίζεται σε 2 τομείς: Στον Πελάτη-Client και στον διακομιστή-Server.



Εικόνα 9 Προτεινόμενη αρχιτεκτονική Client Server

Όπως φαίνεται από την εικόνα 9 ο client εκτελείται στην πλευρά της κινητής συσκευής και ενώνει τον Information Update Client Agent μέσω από το Internet με τον server του πανεπιστημίου. Από την εικόνα 10 μπορούμε να δούμε τις ενέργειες που απαιτούνται ώστε να γίνει έλεγχος και λήψη των τελευταίων ενημερώσεων στην πλευρά του client από τον server. Μπορεί να γίνει διαχωρισμός μεταξύ δύο κύριων διαδικασιών που λαμβάνουν χώρα κατά την επικοινωνία: Η διαπραγμάτευση για την αναγνώριση της συσκευής και η διαπραγμάτευση για την ενημέρωση της πλευράς του client με τα τελευταία αρχεία που διαθέτει ο server.



Εικόνα 10 Client Server επικοινωνία

Διαπραγμάτευση για την αναγνώριση της συσκευής

Η διαδικασία αυτή είναι απαραίτητη και προηγείται της ενημέρωσης των αρχείων ώστε ο server να γνωρίζει τον τύπο και το περιεχόμενο των αρχείων που θα κληθεί να συγχρονίσει-αποστείλει στην συνέχεια στην συσκευή. Καλείτε λοιπόν ο client να μεταδώσει την πληροφορία για το ποιο υλικό και λογισμικό υποστηρίζει στον server. Σύμφωνα με το σύγγραμμα, το 2005, η πληροφορία αυτή δεν διατίθεται αρχικά στις κινητές συσκευές και ως εκ τούτου η λύση που προτείνεται είναι να χρησιμοποιηθεί το CC/PP(Composite Capabilities/Preferences Profile) που έχει υλοποιηθεί στο W3C. Το CC/PP συμπεριλαμβάνεται στο RDF(Resource Description Framework) και αναπαρίσταται με XML(Extensible Markup Language). Κάνοντας χρήση του Jena(2005) ως αναλυτή για την RDF μπορέσαν να γίνουν ερωτήματα και να συγκεντρωθούν πληροφορίες με την βοήθεια λίστας με γνωρίσματα που διαθέτει το CC/PP. Τα γνωρίσματα περιλαμβάνουν πληροφορίες όπως ο τύπος του υπολογιστή, το μέγεθος της μνήμης, το λειτουργικό σύστημα, η δυνατότητα αναπαραγωγής ήχου και εικόνας.

Αρχικά ο client αιτείται από τον server να του αποστείλει το σύνολο των χαρακτηριστικών που πρέπει να μεταδώσει στα οποία περιλαμβάνονται ερωτήσεις στις οποίες καλείτε ο χρήστης να απαντήσει. Με την χρήση των δεδομένων που θα λάβει στην συνέχεια ο server καλείτε να αναγνωρίσει την κινητή συσκευή. Όσο προστίθενται νέες συσκευές στην βάση

δεδομένων, τα χαρακτηριστικά μπορούν να μεταβάλλονται και αυτό αποτελεί ένα βασικό πλεονέκτημα της μεθόδου που επιλέχθηκε για την αναγνώριση της συσκευής καθώς οι αλλαγές μπορούν να γίνουν μόνο στην πλευρά του server και να υιοθετούνται χωρίς έξτρα ενέργειες από κάθε καινούργιο client.

Διαπραγμάτευση για την ανανέωση του υλικού των μαθημάτων

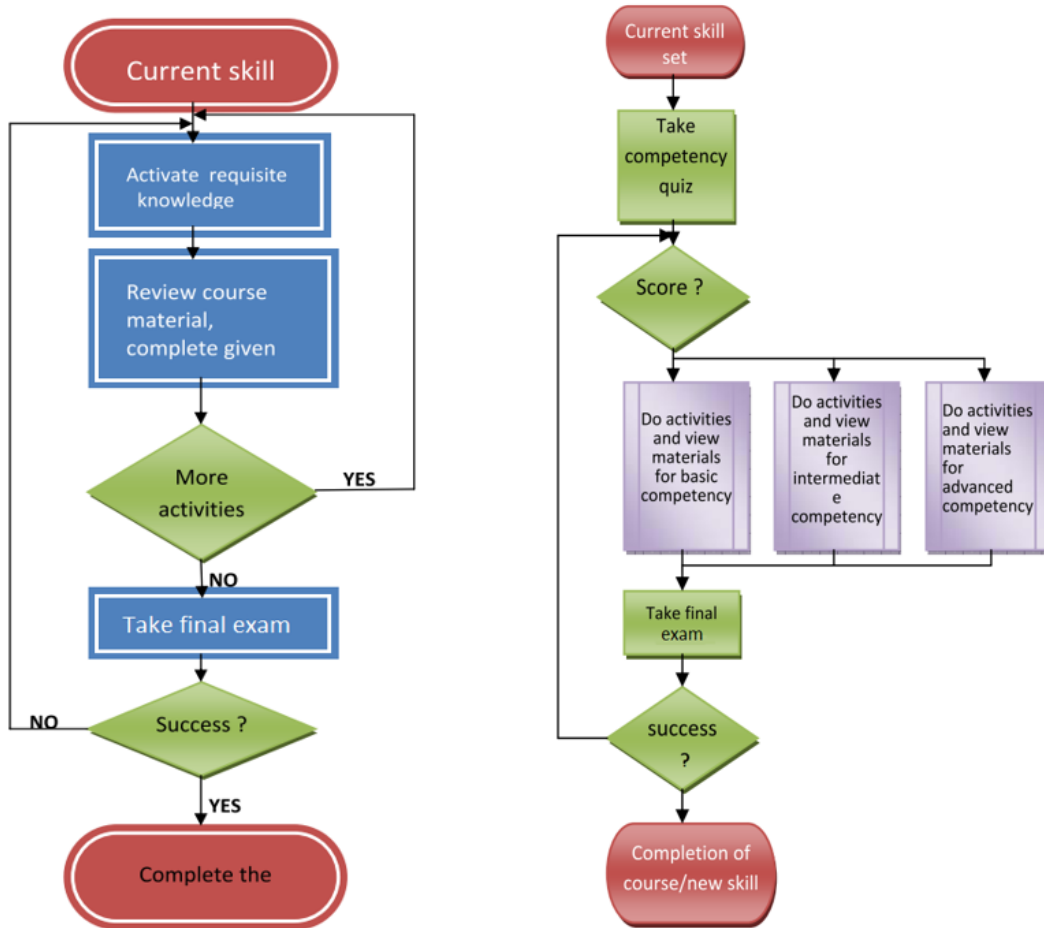
Η λειτουργία αυτή χρησιμοποιείται όταν πρέπει να επικοινωνήσει ο client με τον server για να ελέγξει ποια αρχεία έχουν ανανεωθεί. Αρχικά ο InformationUpdateClient στέλνει ένα ερώτημα στον InformationUpdateAgent ο οποίος με την σειρά του το ανακατευθύνει στον Repository Agent με αίτημα για ενημέρωση του νέου υλικού των μαθημάτων. Μέσα στο ερώτημα που αποστέλλει ο client περιλαμβάνονται το όνομα της πλατφόρμας και οι τελευταίες χρονοσφραγίδες, μαζί με τα ονόματα των αρχείων, που κατέχει. Η πρώτη ενέργεια του Repository Agent είναι να ανακαλέσει από την βάση όλα τα ονόματα των αρχείων και των χρονοσφραγίδων τους. Στην συνέχεια θα γίνει σύγκριση της λίστας αυτής με την λίστα που έχει αποσταλεί από τον client για να δημιουργηθεί μια νέα με τα αρχεία που πρέπει να κατεβάσει/ανανεώσει ο client. Στην λίστα αυτή συμπεριλαμβάνονται και τα αρχεία τα οποία έχουν σβήσει από τον server. Κατόπιν ο RepositoryAgent θα ζητήσει και θα λάβει από την κλάση RepositoryTimeUtil τον σύνδεσμο του κάθε αρχείου που πρέπει να αποθηκεύσει ο client. Σας τελικό αποτέλεσμα θα αποσταλούν στον client 3 δεδομένα. Οι σύνδεσμοι των αρχείων υλικού που πρέπει να αποθηκεύσει στην συσκευή, η λίστα των αρχείων που πρέπει να διαγράψει και οι αντίστοιχες χρονοσφραγίδες.

3.2.2.2 Αρχιτεκτονική από τους L. Henry and S. Sankaranarayanan

Η αρχιτεκτονική που παρουσιάζουν οι 2 συγγραφείς [15] έχει την βάση της στην παραδοσιακή διαδικασία της μάθησης στην οποία τα μέλη που συμμετέχουν είναι 3. Ο μαθητής, ο παιδαγωγός-καθηγητής και ο λέκτορας. Ο ρόλος που έχει ο κάθε ένας στην διαδικασία της μάθησης είναι του μαθητευόμενου, του διδάσκον και του δημιουργού του υλικού αντίστοιχα. Στην υλοποίηση που παρουσιάζεται εδώ, ο μαθητής παραδοσιακά παραμένει αμετάβλητος σαν οντότητα, ο καθηγητής μεταφέρεται στην οντότητα πράκτορα με το όνομα tutor agent (παιδαγωγός πράκτορας) και τον ρόλο του λέκτορα αναλαμβάνει ο Teacher agent (Δάσκαλος πράκτορας). Η προσομοίωση των 2 αυτών ρόλων από τους πράκτορες έχει ως στόχο την ομαλή λειτουργία της μάθησης σύμφωνα με την ως τώρα πρακτική. Ο παιδαγωγός πράκτορας θα βρίσκεται στην κινητή συσκευή και αναλαμβάνει την επικοινωνία με τους μαθητές, την καθημερινή εξυπηρέτησή τους, την λήψη αποφάσεων για το επίπεδο του μαθητευόμενου, την μεταφορά των δραστηριοτήτων στον μαθητευόμενο και την διαχείριση των αλλαγών στο περιβάλλον. Ο Δάσκαλος πράκτορας αναλαμβάνει την παράδοση του κατάλληλου υλικού του μαθήματος στον παιδαγωγό πράκτορα από την

«αποθήκη» λαμβάνοντας υπόψιν και το επίπεδο του εκάστοτε μαθητευόμενου. Με αυτόν τον τρόπο αναπαρίσταται η διαδικασία που ακολουθείτε παραδοσιακά στην μάθηση.

Στο παρακάτω σχήμα παρουσιάζεται η παραδοσιακή διαδικασία μάθησης και η διαδικασία κινητής μάθησης όπως υλοποιείτε στην παρούσα αρχιτεκτονική.

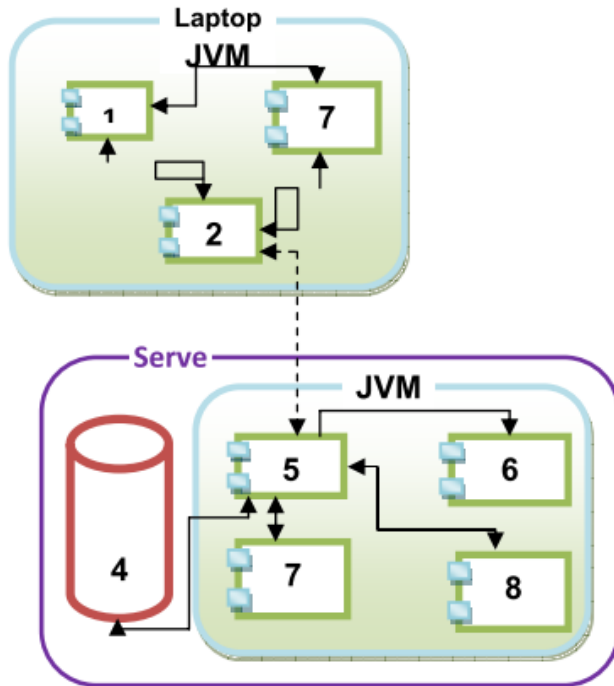


Εικόνα 11 Παραδοσιακή και προτεινόμενη διαδικασία

Κατά την δημιουργία των 2 πρακτόρων έχει δοθεί έμφαση στο σύστημα αποφάσεων και στην ευφυΐα που υλοποιεί ώστε να μπορέσουν να αναπαραστήσουν τον ρόλο των φυσικών προσώπων. Όσο πιο μεγάλο είναι το κοινό που απευθύνεται η διαδικασία μάθησης τότε πιο μεγάλη είναι η ανάγκη να οριστούν επίπεδα. Βλέποντας το αυτό οι συγγραφείς, έχουν ορίσει 3 επίπεδα γνώσεων, το βασικό, το ενδιάμεσο και το εξελιγμένο. Πριν ξεκινήσουν την διαδικασία εκμάθησης, οι χρήστες-μαθητές καλούνται να ακολουθήσουν κάποια προκαταρκτικά τεστ ώστε να κατηγοριοποιηθούν συμφωνά με τις βαθμίδες.

Βασικά συστατικά

Η αρχιτεκτονική έχει 8 κύρια συστατικά όπως φαίνεται και στο ΣΧΗΜΑ.



Εικόνα 12Τα 8 κύρια συστατικά της αρχιτεκτονικής

Για την υλοποίηση γίνεται συνδυαστική χρήση του γραφικού περιβάλλοντος της γλώσσας προγραμματισμού JAVA(Java GUI) και της βιβλιοθήκης JADE.

Περίληπτικά μπόυμε να απαριθμήσουμε τα συστατικά που είναι ορατά και στο σχήμα:

- 1) Γραφικό περιβάλλον παιδαγωγού (GUI Tutor): Αποτελεί την διεπαφή μέσα από την οποία επικοινωνεί ο χρήστης με τον παιδαγωγό πράκτορα. Ο βασικός σκοπός του είναι i) παρέχει την φόρμα σύνδεσης για την είσοδο του εκάστοτε μοναδικού χρήστη ii) διαχειρίζεται τα προκαταρκτικά τεστ για τον έλεγχο του επιπέδου του χρήστη iii) επιτρέπει την παροχή του κατάλληλου υλικού μαθήματος και iv) επιτρέπει τον μαθητή να προχωρήσει την διαδικασία μέχρι τα τελικά τεστ του κάθε επιπέδου
- 2) Διαχείριση προκαταρκτικών τεστ: Εδώ γίνεται διαχείριση των διάφορων τεστ που καλείτε να απαντήσει ο χρήστης στην διάρκεια της εκπαίδευσης. Στόχος είναι η επιτυχία μέσα από την συνεχή εκτίμηση των γνώσεων.
- 3) Παιδαγωγός πράκτορας: Δημιουργείται ένας πράκτορας με «νοημοσύνη» ο οποίος προσπαθεί να μιμηθεί και να προσημειώσει τις ενέργειες που θα έκανε ένας άνθρωπος στην θέση του. Ο πράκτορας αυτός λειτουργεί ως μεσάζον καθώς γίνεται δέκτης των αλληλεπιδράσεων με τον μαθητευόμενο, λαμβάνει όλες τις σχετικές πληροφορίες για τον μαθητή από τον δάσκαλο πράκτορα και τις μεταβιβάζει κατάλληλα σε αυτόν.
- 4) Τράπεζα ερωτήσεων: Περιέχει ερωτήσεις από τις οποίες αποστέλλει ο δάσκαλος πράκτορας στον παιδαγωγό πράκτορα για να απαντηθούν από τον μαθητευόμενο με σκοπό την διαπίστωση του επιπέδου γνώσης.

- 5) Δάσκαλος πράκτορας: Ο πράκτορας αυτός λειτουργεί ως το βασικό σημείο αναφοράς του μαθήματος αναλαμβάνοντας τον ίδιο ρόλο που έχει και ο λέκτορας μέσα σε μία αίθουσα διδασκαλίας. Παράλληλα είναι υπεύθυνος ως προς το ποιος έχει το δικαίωμα να παρακολουθήσει το μάθημα του, για την δημιουργία του υλικού μελέτης που θα έχει πρόσβαση ο μαθητευόμενος και την βαθμολόγηση του μέσω των εξετάσεων που του παρέχει.
- 6) Διανομή του υλικού μελέτης: Υπεύθυνος για την διαδικασία αυτή είναι ο πράκτορας παιδαγωγός. Εντοπίζει και διανέμει στον χρήστη το κατάλληλο υλικό ανάλογα με το επίπεδο γνώσεων που έχει ο μαθητευόμενος.
- 7) Δημιουργία τράπεζας ερωτήσεων: Σε αυτή την διαδικασία καλείτε ο ίδιος ο εισηγητής του μαθήματος να δημιουργήσει την τράπεζα ερωτήσεων από την οποία μετέπειτα θα διαλέξει δάσκαλος πράκτορας ερωτήσεις για την δημιουργία προκαταρκτικών τεστ και τελικών εξετάσεων για τους μαθητές.
- 8) Εγγραφή του μαθητευόμενου: Σε αυτό το σημείο γίνεται ο έλεγχος για την είσοδο του κάθε μαθητευόμενου στην βάση δεδομένων των μαθημάτων και στην εκχώρηση δικαιωμάτων πρόσβασης στο μάθημα που του αναλογεί.

Στην συνέχεια, στην αρχιτεκτονική που παρουσιάζουν οι Henry και Sankaranarayanan γίνεται χρήση της γλώσσας προγραμματισμού JAVA και της πλατφόρμας JADE για την υλοποίηση των πρακτόρων. Η πλατφόρμα JADE αποτελείται από πράκτορες και από τα «φορείς» που τους περιέχουν. Στην παρούσα υλοποίηση έχει επιλεγεί να δημιουργηθεί σε έναν βασικό υπολογιστή ο κύριος φορέας που περιέχει τον Δάσκαλο πράκτορα και σε κάθε κινητή συσκευή ένα ακόμα φορέα που περιέχει τον εκάστοτε παιδαγωγό πράκτορα για τον κάθε μαθητευόμενο. Κάθε παιδαγωγός πράκτορας δημιουργείται και συνδέεται με συγκεκριμένο Δάσκαλο πράκτορα.

Διατυπωμένος σε απλή γλώσσα ο αλγόριθμος που ακολουθείτε στην παρούσα αρχιτεκτονική είναι ο παρακάτω:

- Ο μαθητευόμενος ανοίγει την εφαρμογή ώστε να αποκτήσει πρόσβαση στο υλικό μαθήματος. Αν είναι η πρώτη φορά τότε πρέπει να ακολουθηθεί η διαδικασία της εγγραφής. Ζητείται από τον χρήστη να δηλώσει το επίπεδο γνώσης που έχει για το συγκεκριμένο μάθημα. Τα επίπεδα που ορίζονται είναι 3: Το βασικό, το ενδιάμεσο και το προχωρημένο.
- Ο πράκτορας παιδαγωγός λαμβάνοντας υπόψιν το επίπεδο που δηλώθηκε ζητάει από τον πράκτορα δάσκαλο ένα προκαταρκτικό τεστ για να γίνει επιβεβαίωση του επιπέδου που θέλει.
- Ο πράκτορας δάσκαλος ανακτά ερωτήσεις από την τράπεζα ερωτήσεων του μαθήματος για την δημιουργία του προκαταρκτικού τεστ. Το τεστ αυτό περιέχει

ερωτήσεις και από τα 3 επίπεδα. Στην συνέχεια αποστέλλεται στον πράκτορα παιδαγωγό. Κάθε επίπεδο του τεστ αποτελείται από 10 ερωτήσεις και ο μαθητευόμενος έχει στην διάθεση του για να απαντήσει 15 λεπτά για το πρώτο επίπεδο, 20 για το δεύτερο και 30 για το τρίτο.

- Ξεκινώντας ο μαθητευόμενος το τεστ ο χρόνος αρχίζει να μειώνεται
- Αρχικά ο πράκτορας παιδαγωγός ξεκινάει την επιλογή και την προώθηση ερωτήσεων στον χρήστη από την βασική κατηγορία. Αν το επίπεδο είναι ικανοποιητικό (θα πρέπει να επιτύχει 6 σωστές από τις 10 ερωτήσεις) τότε συνεχίζει το τεστ με ερωτήσεις από το επόμενο επίπεδο. Αντίστοιχα και στις ερωτήσεις από το ενδιάμεσο επίπεδο. Αν το επίπεδο δεν κριθεί ικανοποιητικό (οι σωστές απαντήσεις είναι λιγότερες από 6 στις 10) τότε προσφέρονται στον χρήστη ερωτήσεις του ίδιου επιπέδου. Κάθε μαθητής θα κληθεί να απαντήσει 30 ερωτήσεις σε μπλοκ των 10 εκτός αν κριθούν οι απαντήσεις μη ικανοποιητικές στο βασικό επίπεδο, όπου στην περίπτωση αυτή θα απαντήσει συνολικά 20. Στο τέλος του προκαταρκτικού τεστ τα αποτελέσματα παρουσιάζονται στην οθόνη του χρήστη κάνοντας σύγκριση του επιπέδου που δήλωσε και αυτού που προέκυψε μέσα από ερωτηματολόγιο.
- Με βάση το επίπεδο παρέχεται στον μαθητή πρόσβαση στο υλικό του μαθήματος από τον πράκτορα παιδαγωγό. Αυτό επιτυγχάνεται από τον πράκτορα στέλνοντας στον δάσκαλο πράκτορα τα αποτελέσματα και ζητώντας να του αποσταλεί το κατάλληλο υλικό μαθήματος.
- Μετά την ανάγνωση του υλικού αυτού από τον μαθητευόμενο, ζητάει να του αποσταλεί μέσω του παιδαγωγού πράκτορα νέο τεστ για το επίπεδο που ανήκει. Αν το αποτέλεσμα είναι επιτυχημένο τότε του δίνεται πρόσβαση στο επόμενο επίπεδο. Το τεστ κάθε επιπέδου αποτελείται και αυτό από 10 ερωτήσεις όπου δίνεται χρόνος 30 λεπτών για να απαντηθούν.
- Τέλος όταν ο χρήστης περάσει επιτυχημένα όλα τα επίπεδα του μαθήματος μπορεί μέσα από τον πράκτορα παιδαγωγό να ανακτήσει και να πραγματοποιήσει τις τελικές εξετάσεις για το μάθημα αυτό οι οποίες έχουν διάρκεια μιας ώρας για να ολοκληρωθούν.

3.2.2.3 Αρχιτεκτονική MX-Learn

Η αρχιτεκτονική MX-Learn προτάθηκε το 2006 από τους Casalino, Atri και Garro [16] με στόχο να επιτύχει τους περισσότερους από τους αντικειμενικούς σκοπούς της κινητής μάθησης εστιάζοντας στην χρήση σύγχρονων τεχνολογιών μέσω των οποίων οι συσκευές και τα δίκτυα επικοινωνίας θα προσφέρουν πολλαπλές υπηρεσίες και εφαρμογές με εύκολο στην χρήση περιβάλλον στον τελικό χρήστη. Έχει ως όραμα την περιβάλλουσα νοημοσύνη

τοποθετώντας ουσιαστικά τον χρήστη, δηλαδή τον μαθητή, στο κέντρο των ενεργειών. Η MX-Learn βασίζεται σε ανοικτά πρότυπα, κάνει χρήση ανοικτού λογισμικού που προσφέρει δια λειτουργικότητα και επιχειρεί μέσα από τα ασύρματα δίκτυα να δημιουργήσει βάσεις γνώσης που θα είναι πρόσβαση και ευκολά διαμοιραζόμενη. Σύμφωνα με τους εμπνευστές, αυτό θα φέρει κέρδη στους μαθητευόμενος, καθώς θα έχουν την δυνατότητα για μια ποιο προσωποκεντρική εμπειρία που θα προσφέρει περισσότερη νοημοσύνη στο τελικό αποτέλεσμα, θα είναι παραμετροποιήσιμη από τον ίδιο, στοχευμένη στον καθένα σύμφωνα με το προφίλ του και με μεγαλύτερη διαδραστικότητα. Επιπρόσθετα κέρδη θα έχει και ο εισηγητής καθώς μέσα από την βάση γνώσης θα αποκτήσει απρόσκοπτή γνώση σε δημόσιες και ιδιωτικές πηγές που θα του ενισχύσουν την δημιουργικότητα, την παραγωγικότητα και την πολυεπίπεδη παροχή γνώσης στον μαθητευόμενο.

Μέσα από την επιλογή για χρήση ανοικτών προτύπων οι δημιουργοί προσπαθούν να προσφέρουν καθοδήγηση καθώς και να δώσουν ώθηση στην κοινότητα των προγραμματιστών στην δημιουργία περισσότερων υπηρεσιών και εφαρμογών κινητής μάθησης. Η αρχιτεκτονική παρουσιάζει 4 χαρακτηριστικά που ορίζει ως εξαιρετικά χρίσμα σε ένα νέο σύστημα ηλεκτρονικής μάθησης:

- Είναι εξαιρετικά υποκειμενική καθώς διατηρεί ένα πολύ «πλούσιο» προφίλ με πληροφορίες για τον χρήστη, τις γνώσεις και τις δυνατότητες του.
- Είναι προσαρμόσιμη , λαμβάνοντας υπόψιν την συσκευή του χρήστη και τις δυνατότητες που έχει να αναπαράγει περιεχόμενο.
- Είναι γενικής χρήσης καθώς μπορεί να παρουσιάσει περιεχόμενο πολλών διαφορετικών τύπων
- Βασίζεται στην XML καθώς αποθηκεύει τα δεδομένα σε XML αρχεία και για την λειτουργία της κάνει χρήση των ACML(), XQuery και DOM (Document Object Model)

Ανάλυση

Η αρχιτεκτονική MX-Learn βασίζεται στο X-Learn, ένα XML πολυπρακτορικό σύστημα ηλεκτρονικής μάθησης και επεκτείνει τις δυνατότητες του στην κινητή μάθηση. Η MX-Learn διαχειρίζεται γνωστικά αντικείμενα τα οποία αποτελούν γνώση που οργανώνεται σε ξεχωριστές μονάδες και η οποία μπορεί να συνδυαστεί δυναμικά για να δημιουργήσει εξατομικευμένα προγράμματα μάθησης. Για να μπορέσει να γίνει οργάνωση των γνωστικών αντικειμένων χρησιμοποιούνται τα γνωστικά αντικείμενα μεταδεδομένων (Learning Object Metadata- LOM)

Μπαίνοντας βαθύτερα στην δομή της αρχιτεκτονικής βλέπουμε την ύπαρξη 4 πρακτόρων:

- i) Τον Πράκτορα διεπαφή συσκευής (User-Device Agent , UDA) ο οποίος διαχειρίζεται την συνεδρία ηλεκτρονικής μάθησης που είναι σε εξέλιξη μεταξύ του χρήστη U_j σε μια συσκευή D_i
- ii) Τον πράκτορα διαχείρισης των δεξιοτήτων του χρήστη (Skill Manager Agent , SKA). Ο πράκτορας αυτός σε συνδυασμό με το ιστορικό και τις δυνατότητες που ήδη κατέχει ο χρήστης τον βοηθάει στην επιλογή και στην πορεία που πρέπει να ακολουθήσει για κάνει κτήμα του την δεξιότητα-στόχο.
- iii) Τον πράκτορα προγράμματος εκμάθησης (Learning Program Agent, LPA) ο οποίος λαμβάνοντας υπόψιν το ιστορικό γνώσεων του χρήστη (U_j) , την συσκευή που έχει (D_i) και την δεξιότητα που πρέπει να αναπτύξει (Sk) διαμορφώνει ένα προσωποποιημένο πρόγραμμα εκμάθησης.
- iv) Τον πράκτορα συμπίεσης XML αρχείων (XML Compression Agent, XCA). Σκοπός αυτού είναι να μειώσει το δυνατόν τον χώρο που καταλαμβάνει ένα XML αρχείο, δείχνοντας κάποια σχετική ανοχή στην απώλεια ενός ποσοστού δεδομένων , ώστε να μεταφερθούν όσο γίνεται λιγότερα δεδομένα στην φορητή συσκευή του χρήστη.

Επιπρόσθετα αξίζει να σημειωθεί η ύπαρξη μιας αποθήκης όπου αποθηκεύονται όλα τα γνωστικά αντικείμενα που πραγματεύονται (Learning Object Repository, LOR).

Αναλυτικότερα τα βήματα που λαμβάνουν χώρα στο σενάριο χρήσης της αρχιτεκτονικής MX-Learn είναι:

- i) Ο χρήστης U_j που θέλει να αποκτήσει μια ακόμα δεξιότητα στην συσκευή του D_i , ενεργοποιεί τον User-Device Agent UDA_{ij}
- ii) Ο UDA_{ij} σε συνεργασία με τον Skill Manager Agent SMA εντοπίζει και παρουσιάζει, σε συνδυασμό με το ιστορικό του χρήστη, τις δεξιότητες που μπορεί να αποκτήσει σε συνέχεια αυτών που ήδη έχει.
- iii) Ο χρήστης U_j επιλέξει την δεξιότητα που επιθυμεί από αυτές που του προτείνονται
- iv) Ο UDA_{ij} συνεργάζεται με τον SMA για να ορίσει την λίστα των θεμάτων που οδηγήσουν στην απόκτηση της επιθυμητής δεξιότητας.
- v) Στην συνέχεια, για κάθε θέμα που έχει προκύψει από την προηγούμενη διαδικασία, ο UDA_{ij} συνεργάζεται με τον Learning Program Agent LPA για να οριστεί το καλύτερο δυνατό πρόγραμμα που θα οδηγήσει τον U_j στην εκμάθηση του Sk . Στην δημιουργία του κατάλληλου προγράμματος, λαμβάνονται υπόψιν πέρα από την γνώση που ήδη έχει ο χρήστης U_j και η συσκευή D_i την οποία χρησιμοποιεί. Για τον ορισμό των δυνατοτήτων της συσκευής λαμβάνεται υπόψιν και το χαρακτηριστικό της ταχύτητα συμπίεσης που διαπιστώνει ο XML Compression Agent XCA .

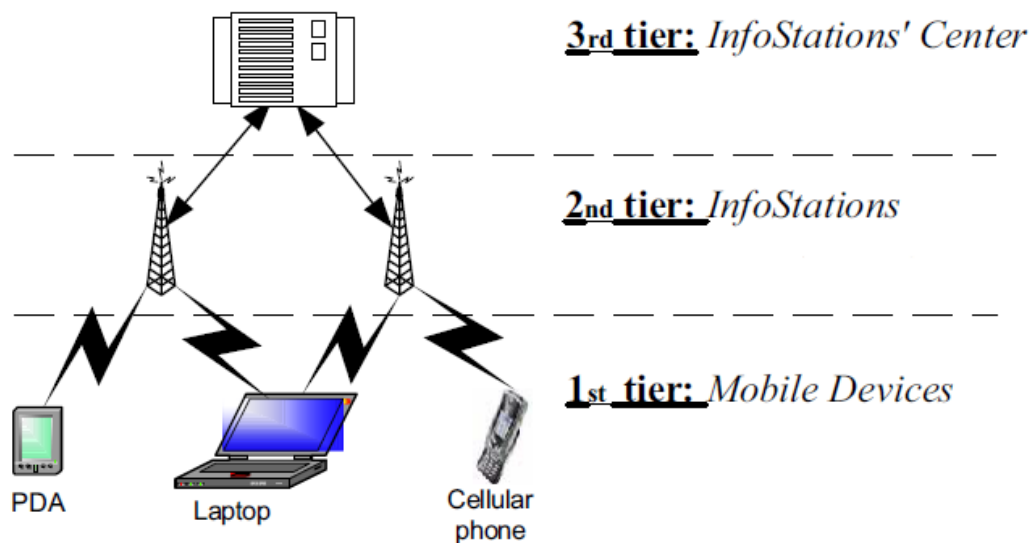
- vi) Ο UDAij παραλαμβάνει το κατάλληλο υλικό μαθήματος από το Learning Object Repository LOR και το παρουσιάζει στο χρήστη σύμφωνα με τον ρυθμό συμπίεσης που όρισε ο XCA.
- vii) Όταν ο χρήσης Uj ολοκληρώνει κάποιο κεφάλαιο για το Sk, ο UDAij ενημερώνει κατάλληλα το προφίλ του. Όταν ολοκληρώσει όλες τις ενότητες τότε ο πράκτορας προσθέτει την απόκτηση της νέας δεξιότητας για τον Uj. Στην διαδικασία αυτή παρέχετε η δυνατότητα στον χρήστη να «παγώσει» την μάθηση και να την ολοκληρώσει αργότερα ακόμα και με χρήση διαφορετικής συσκευής από την αρχική. Ο πράκτορας UDAij υποστηρίζει την συνέχιση της διαδικασίας μάθησης σε διαφορετική συσκευή λαμβάνοντας υπόψιν και τυχόν αυξημένες ή μειωμένες δυνατότητες συγκριτικά με την αρχική.

3.2.2.4 Αρχιτεκτονική 3 Tier InfoStations

Η αρχιτεκτονική αυτή παρουσιάστηκε από τους Ivan Ganchev, Stanimir Stojanov, Mairtin O'Droma, Damien Meere, Veselina Valkanova αρχικά στο [17] και μετέπειτα στο [18]. Η αρχιτεκτονική ορίζει ένα δίκτυο 3 επιπέδων, με βασικό χαρακτηριστικό την χρήση σταθμών πληροφορίας. Εμπνεύστηκε και προτείνεται για εφαρμογή σε ένα δίκτυο πανεπιστημιακής μονάδας. Βασίζεται στην χρήση τεχνολογιών Bluetooth, Wi-fi και Wi-Max.

Όπως φαίνεται και από το σχήμα, τα 3 επίπεδα που διαχωρίζει η αρχιτεκτονική είναι :

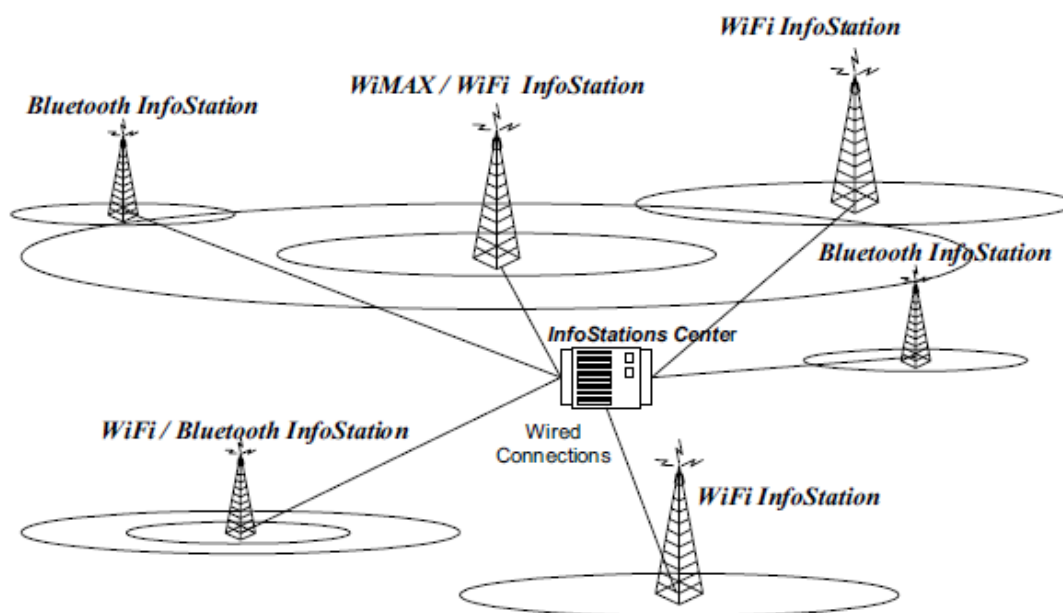
- Κινητές συσκευές (Mobile devices)
- Ενδιάμεσοι σταθμοί πληροφορίας (InfoStations)
- Κεντρικοί σταθμοί πληροφορίας (InfoStations Center)



Εικόνα 13 Αρχιτεκτονική 3 Tier InfoStations

Στο πρώτο επίπεδο τοποθετείται ο χρήστης έχοντας στην κατοχή του μια κινητή συσκευή. Παραδείγματα τέτοιων συσκευών είναι ο φορητός υπολογιστής και το κινητό τηλέφωνο. Μέσα από την λειτουργία ευφών πρακτόρων μεταφέρονται τα αιτήματα του χρήστη στο επόμενο επίπεδο.

Οι ενδιάμεσοι σταθμοί πληροφορίας λαμβάνουν τα αιτήματα του κάθε μαθητή και ανάλογα με την γνώση που έχουν απαντούν σε αυτά. Η δομή του συστήματος είναι τέτοια ώστε αν ο συγκεκριμένος ενδιάμεσος σταθμός που έλαβε το αίτημα, δεν έχει την πληροφορία που απαιτείται για να εξυπηρετήσει, να επικοινωνήσει αυτός με το 3ο επίπεδο αναζητώντας τα ανάλογα δεδομένα. Η επικοινωνία με το τρίτο επίπεδο γίνεται μέσα από δίκτυα μεγάλης ταχύτητας γεωγραφικά ταξινομημένα για όσο το δυνατόν ταχύτερη εξυπηρέτηση. Οι ενδιάμεσοι σταθμοί πληροφορίας υποστηρίζουν διάφορα πρωτόκολλα όπως φαίνεται από το παρακάτω σχήμα.



Εικόνα 14 Ενδιάμεσοι σταθμοί πληροφορίας

Το τρίτο επίπεδο αποτελείται από τους κεντρικούς σταθμούς πληροφορίας και αποτελεί τον πυρήνα του συστήματος. Η κύρια λειτουργία του είναι να ενημερώνει και να συγχρονίζει την πληροφορία ανάμεσα στα συστήματα. Βασική λειτουργία του είναι η δημιουργία και διαχείριση προφίλ χρηστών αλλά και υπηρεσιών που παρέχει. Τα προφίλ των χρηστών δίνουν την δυνατότητα παρακολούθησης της προόδου στον μαθητή καθώς και στον εκπαιδευτή διεκπεραιώνοντας έτσι διπλό ρόλο στην ενημέρωση του μαθητή και στην ανατροφοδότηση του εκπαιδευτή αντίστοιχα. Όπως είναι κατανοητό αυτό επιτρέπει στον εκπαιδευτή να μεταβάλει κατάλληλα την παρεχόμενη ύλη και τα απαιτούμενα τεστ του κάθε μαθητή ξεχωριστά. Αντίστοιχα, τα προφίλ των παρεχόμενων υπηρεσιών, ενημερώνονται για την παρεχόμενη ύλη, την ενότητα και τους χρήστες που τα παρακολουθούν.

Οι συγγραφείς προσδιορίζοντας σε μακροσκοπικό επίπεδο την αρχιτεκτονική, διαχωρίζουν τις υπηρεσίες που παρέχονται στις παρακάτω 3 κυριότερες:

Κινητή διάλεξη (mLecture) Σε αυτή την υπηρεσία συμπεριλαμβάνουμε την παρουσίαση των παρεχόμενων μαθημάτων, την επιλογή των επιθυμητών και την παρακολούθησή τους. Εδώ μπορούμε να πούμε ότι γίνεται προσομοίωση της επικοινωνίας που θα είχε ο εκπαιδευτής με τον μαθητευόμενο. Αξίζει να τονίσουμε ότι αν ένα μάθημα παρέχει πολυμεσικό υλικό πολλαπλών κατηγοριών, ανάλογα με την συσκευή του μαθητευόμενου θα του προβληθεί και αυτό που δύναται να παρουσιάσει η συσκευή του.

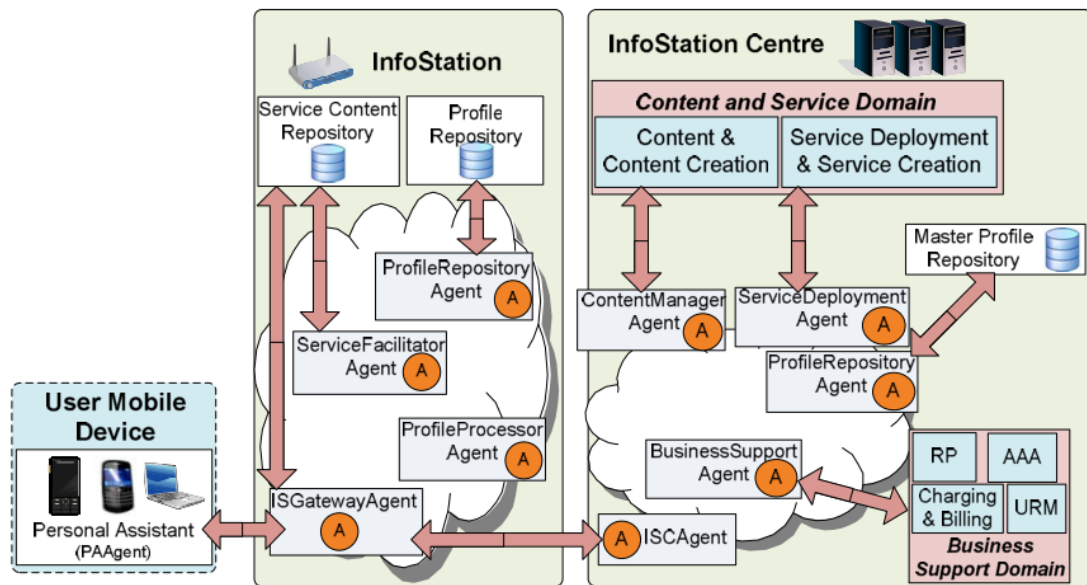
Κινητοί οδηγοί (mTutorial) Η υπηρεσία αυτή είναι υπεύθυνη να παρουσιάσει στον μαθητή τους εξατομικευμένους οδηγούς που έχει φτιάξει ο εκπαιδευτής. Σε αυτούς περιλαμβάνονται και ερωτήσεις.

Κινητή εξέταση (mTest) Η υπηρεσία αυτή παρέχει τα ανάλογα τεστ που πρέπει να ολοκληρώσει ο μαθητευόμενος για να πιστοποιηθεί η γνώση του πάνω στο μάθημα που παρακολούθησε μέσα από την διαδικασία.

Για την ορθή λειτουργία του συστήματος στην παραπάνω αρχιτεκτονική, είναι σημαντικό να συνεργάζονται επιτυχώς οι πράκτορες του κάθε επιπέδου ώστε να ενημερώνονται οι ενδιαμέσοι σταθμοί πληροφορίας αλλά και οι κεντρικοί σταθμοί πληροφορίας για τις ενέργειες που ολοκληρώνει ο κάθε μαθητής-χρήστης. Κάθε σύνδεση ενός χρήστη στο κοντινότερο ενδιαμέσο σταθμό πληροφορίας ακολουθείται από επικοινωνία του πράκτορα της κινητής συσκευής με τον εκάστοτε πράκτορα του ενδιαμέσου σταθμού ανταλλάσσοντας πληροφορίες αυθεντικοποίησης, συγχρονισμού των ενεργειών που έγιναν στην κινητή συσκευή όσο δεν ήταν σε ενεργή σύνδεση και ενημέρωση του προφίλ του χρήστη στον ενδιαμέσο σταθμό αλλά και στους κεντρικούς σταθμούς.

Αναλύοντας την αρχική παρουσίαση της αρχιτεκτονικής στο [17], παρατηρούμε ότι οι ευφυείς πράκτορες που έχουν αναλάβει την λειτουργία είναι 3. Ο πράκτορας προσωπικός βοηθός (personal assistant agent), ο πράκτορας ενδιαμέσος σταθμός (InfoStation agent) και ο πράκτορας κεντρικός σταθμός πληροφορίας (InfoStation Center agent).

Στην ανανεωμένη παρουσίαση της αρχιτεκτονικής που έγινε στο [18] παρατηρούμε ότι οι πράκτορες που ενεργοποιούνται στο σύστημα είναι αρκετά περισσότεροι. Παρουσιάζονται σχηματικά στην παρακάτω εικόνα.

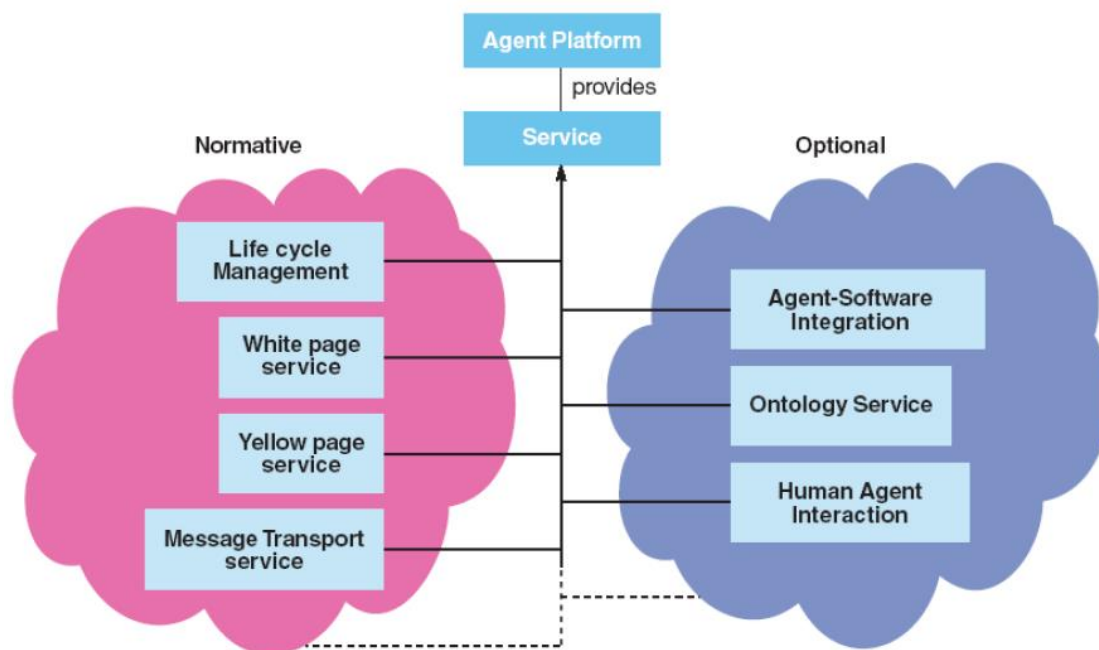


Εικόνα 15 Ενισχυμένη 3 Tier Αρχιτεκτονική

3.3 JADE

Η JADE (Java Agent Development Framework) είναι η βιβλιοθήκη-πλαίσιο, που έχει σκοπό την απλοποίηση της ανάπτυξης συστημάτων πολλαπλών πρακτόρων. Η Jade δημιουργήθηκε από την TILAB για την ανάπτυξη κατανεμημένων multi-agent εφαρμογών οι οποίες βασίζονται στην peer-to-peer αρχιτεκτονική επικοινωνίας. Η νοημοσύνη, η πρωτοβουλία, η πληροφορία, οι πόροι και ο έλεγχος μπορούν να κατανεμηθούν πλήρως σε κινητά τερματικά όπως επίσης και σε υπολογιστές σταθερού δικτύου. Το περιβάλλον μπορεί να αναπτύσσεται δυναμικά με οντότητες (peers), οι οποίες καλούνται στην Jade ως ευφυείς πράκτορες. Οι πράκτορες μπορούν να δημιουργούνται και να καταστρέφονται στο σύστημα ανάλογα με τις απαιτήσεις και τις ανάγκες του περιβάλλοντος της εφαρμογής. Η επικοινωνία ανάμεσα στις οντότητες, ανεξάρτητα αν τρέχουν σε ασύρματα ή ενσύρματα δίκτυα, είναι πλήρως συμμετρική και κάθε οντότητα μπορεί να παίζει το ρόλο και του δημιουργού ενός αιτήματος αλλά και αυτού που ανταποκρίνεται σε αυτό.

Το 1996 η TILAB προώθησε την δημιουργία του FIPA (Foundation for Intelligent Physical Agents), ενός διεθνούς μη κερδοσκοπικού συνεταιρισμού εταιρειών και οργανισμών. Ο FIPA είχε σαν σκοπό την παραγωγή ενός συγκεκριμένου προτύπου για την τεχνολογία των πρακτόρων. Η TILAB και πιο συγκεκριμένα η JADE ομάδα υποστήριξε και καθοδήγησε σε πολλαπλά επίπεδα αυτή την πρωτοβουλία. Στο τέλος του 2002 ο FIPA κυκλοφόρησε τελικά το πρότυπο.



Εικόνα 16 Πρότυπο FIPA: Οι υπηρεσίες που προσφέρονται. [1]

Το πρότυπο στοχεύει-εστιάζει στην εξωτερική συμπεριφορά των επιμέρους τμημάτων του συστήματος, αφήνοντας ανοιχτές τις λεπτομέρειες της υλοποίησης και τις εσωτερικές αρχιτεκτονικές. Στην πραγματικότητα η εσωτερική αρχιτεκτονική του Jade είναι μοναδική παρόλο που υπακούει πλήρως στο FIPA πρότυπο. Το FIPA πρότυπο εμπεριέχει το μοντέλο των πρακτόρων και συγκεκριμένα καθορίζει το σχετικό μοντέλο μιας πλατφόρμας που παρέχει ένα σύνολο απαιτούμενων υπηρεσιών. Η συλλογή αυτών των υπηρεσιών και οι διεπαφές τους, απεικονίζουν τους κανόνες που επιτρέπουν την ύπαρξη, λειτουργία και διαχείριση των πρακτόρων.

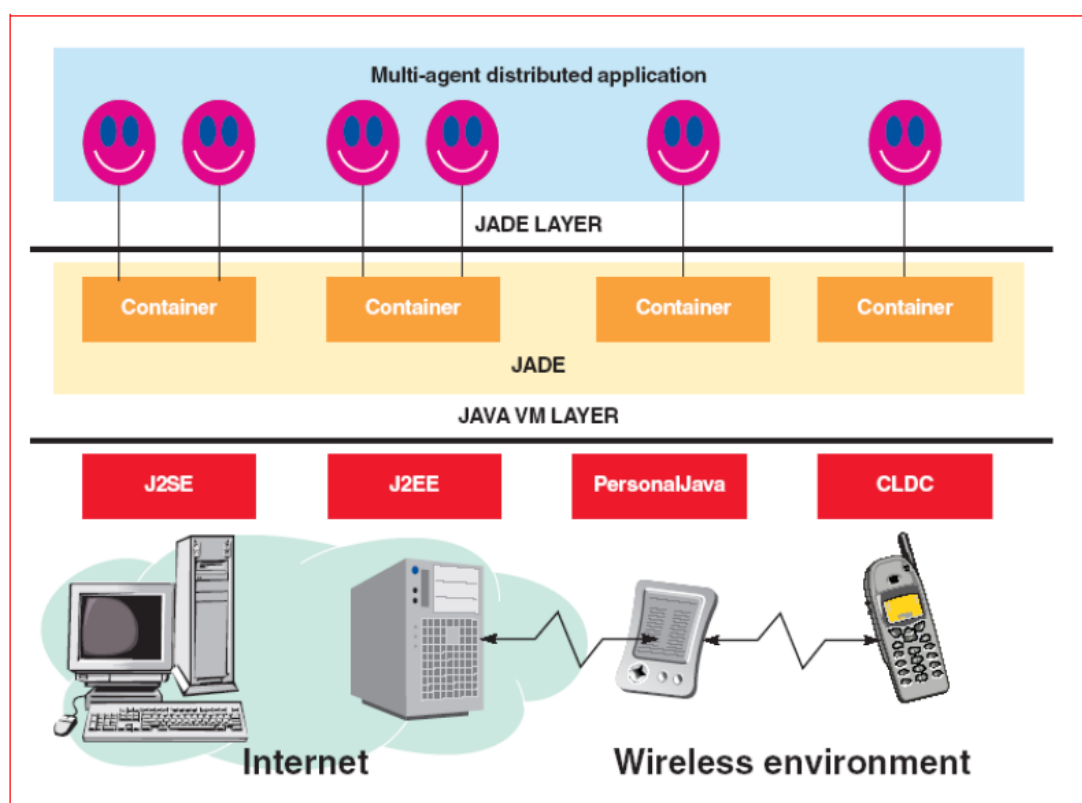
Για την επικοινωνία των πρακτόρων υιοθετήθηκε από την FIPA η Agent Communication Language (ACL). Η FIPA ACL βασίζεται στη θεωρία επικοινωνιακών ενεργειών και στις υποθέσεις και απαιτήσεις του πρακτορικού προτύπου. Η FIPA τυποποίησε μία επεκτάσιμη βιβλιοθήκη με 22 ενέργειες επικοινωνίας που επιτρέπουν την αναπαράσταση ποικίλων επικοινωνιακών προθέσεων (αιτήσεις, ενημερώσεις, προτάσεις, ερωτήσεις, αρνήσεις κ.α.). Ο οργανισμός FIPA καθόρισε τη δομή ενός μηνύματος, το περιεχόμενο του μηνύματος, τις ιδιότητές του (π.χ. κωδικοποιήσεις και γλώσσα αναπαράστασης) και προσδιόρισε χρήσιμη πληροφορία για την παρακολούθηση της επικοινωνίας ανάμεσα στους πράκτορες. Επιπρόσθετα ο FIPA περιέγραψε επίσης κοινά πρωτόκολλα επικοινωνίας τα οποία παρέχουν στους πράκτορες βιβλιοθήκες προτύπων για την υλοποίηση απλών εργασιών όπως μεταβίβαση μιας ενέργειας.

Η Jade έχει υλοποιηθεί εξολοκλήρου σε Java και βασίζεται στις παρακάτω κατευθυντήριες αρχές:

- Αλληλεπίδραση: Η Jade είναι συμβατή με τις FIPA προδιαγραφές. Κατά συνέπεια οι πράκτορες της Jade μπορούν να αλληλοεπιδρούν με άλλους πράκτορες με την προϋπόθεση ότι ακολουθούν το ίδιο πρότυπο.
- Ομοιομορφία και μεταφερσιμότητα: Η Jade παρέχει ένα ομοιογενές σύνολο διεπαφών οι οποίες είναι ανεξάρτητες του υποκείμενου δικτύου και της Java έκδοσης. Αναλυτικότερα παρέχει τις ίδιες διεπαφές για τα J2EE, J2SE και J2ME περιβάλλοντα. Θεωρητικά οι προγραμματιστές μιας εφαρμογής μπορούν να αποφασίσουν σχετικά με το Java περιβάλλον εκτέλεσης κατά τη διάρκεια της ανάπτυξης της εφαρμογής.
- Ευκολία στη χρήση: Η πολυπλοκότητα της κρύβεται πίσω από ένα απλό και διαισθητικό σύνολο διεπαφών.
- Φιλοσοφία «πληρωμής βάση χρήσης»: Οι προγραμματιστές δεν χρειάζεται να χρησιμοποιούν όλες τις υπηρεσίες που παρέχονται. Επιπλέον δεν χρειάζεται να γνωρίζουν τα πάντα σχετικά με υπηρεσίες που δεν χρησιμοποιούν. Αποτέλεσμα των παραπάνω είναι να γλιτώνουμε σε υπολογιστικό φόρτο.[1]

3.3.1 Μοντέλο Αρχιτεκτονικής Jade

Η Jade περιλαμβάνει εκτός από τις βιβλιοθήκες (Java κλάσεις) που απαιτούνται για την ανάπτυξη εφαρμογών πρακτόρων και το περιβάλλον εκτέλεσης το οποίο παρέχει τις βασικές υπηρεσίες, και πρέπει να είναι ενεργό στη συσκευή πριν την εκτέλεση των πρακτόρων. Κάθε στιγμιότυπο του Jade περιβάλλοντος εκτέλεσης ονομάζεται container (φορέας) . Το σύνολο όλων των containers καλείται πλατφόρμα και προσφέρει ένα ομοιογενές επίπεδο το οποίο αποκρύπτει από τους πράκτορες την πολυπλοκότητα και ανομοιομορφία του υποκείμενου κυκλώματος (υλικό, λειτουργικά συστήματα, είδη δικτύων, Java Virtual Machine).



Εικόνα 17 Αρχιτεκτονική Jade

Όπως απεικονίζεται και στο σχήμα η Jade είναι συμβατή με το J2ME CLDC/MIDP1.0 περιβάλλον. Επιπλέον έχουν ήδη δοκιμαστεί σε κινητά δίκτυα με διαφορετικά κινητά τερματικά. Η Jade είναι εξαιρετικά ευπροσάρμοστη και ως εκ τούτου δεν ταιριάζει μόνο σε περιβάλλοντα περιορισμένων πόρων αλλά έχει ήδη ενσωματωθεί σε πολύπλοκες αρχιτεκτονικές όπως .NET και J2EE.

Λειτουργικό Μοντέλο

Η Jade λειτουργικά παρέχει τις βασικές υπηρεσίες που είναι απαραίτητες για κατανεμημένες peer-to-peer εφαρμογές σε σταθερό και κινητό περιβάλλον. Παρέχει την δυνατότητα σε κάθε

πράκτορα να ανιχνεύει με δυναμικό τρόπο άλλους πράκτορες και να επικοινωνεί απευθείας μαζί τους. Από πλευράς εφαρμογών κάθε πράκτορας ταυτοποιείται από ένα μοναδικό όνομα και προσφέρει ένα σύνολο υπηρεσιών. Το όνομα του πράκτορα εξαρτάται εν μέρη από την πλατφόρμα στην οποία ανήκει ο container και από το όνομα που του δίνεται κατά την δημιουργία του. Ένα παράδειγμα ονόματος είναι το Teacher@192.168.1.1:7778/JADE όπου αριστερά από το σύμβολο @ είναι το όνομα που δίνεται κατά την δημιουργία του και δεξιά είναι το όνομα της πλατφόρμας. Ο πράκτορας μπορεί να καταγράψει και να τροποποιήσει τις υπηρεσίες του, να ψάξει για άλλους πράκτορες που παρέχουν συγκεκριμένες υπηρεσίες όπως και να ελέγξει τον κύκλο ζωής του. Οι πράκτορες επικοινωνούν ανταλλάσσοντας ασύγχρονα μηνύματα. Ο αποστολέας και ο παραλήπτης μπορεί να μην είναι διαθέσιμοι την ίδια στιγμή. Ο παραλήπτης μπορεί να μην υπάρχει καν (ή να μην υπάρχει ακόμα) ή μπορεί να μην είναι άμεσα γνωστό από τον αποστολέα ότι μπορεί να καθορίσει κάποια ιδιότητα σαν προορισμό.

Παρόλο που πραγματοποιείται ο παραπάνω τύπος επικοινωνίας, διατηρείται η ασφάλεια, αφού για εφαρμογές που την απαιτούν η Jade παρέχει κατάλληλους μηχανισμούς για πιστοποίηση και επαλήθευση των δικαιωμάτων των πρακτόρων. Ως εκ τούτου, όποτε απαιτείται, μια εφαρμογή μπορεί να πιστοποιήσει την ταυτότητα του αποστολέα ενός μηνύματος.

Για την αύξηση της κλιμάκωσης και για τη χρήση σε περιβάλλοντα περιορισμένων πόρων, η Jade παρέχει τη δυνατότητα εκτέλεσης πολλαπλών παράλληλων εργασιών στο ίδιο Java thread. Στο J2SE και σε προσωπικά περιβάλλοντα Java, η Jade υποστηρίζει ευκολία διακίνησης του κώδικα και της κατάστασης εκτέλεσής του. Αναλυτικότερα ένας πράκτορας μπορεί να σταματήσει να τρέχει σε ένα σύστημα και να μετακινηθεί σε διαφορετικό απομακρυσμένο σύστημα (χωρίς να απαιτείται εγκατάσταση του κώδικα στο νέο σύστημα, και να επανεκκινήσει από το σημείο στο οποίο είχε διακοπεί). Αυτή η λειτουργικότητα επιτρέπει, για παράδειγμα, την κατανομή του υπολογιστικού φόρτου κατά την εκτέλεση μετακινώντας τους πράκτορες σε μηχανές λιγότερου φόρτου χωρίς καμία επίδραση στην εφαρμογή.

Η πλατφόρμα περιέχει επίσης υπηρεσία ονοματολογίας (εξασφάλιση μοναδικότητας ονόματος κάθε πράκτορα) και υπηρεσίες yellow paging οι οποίες μπορούν να κατανεμηθούν σε πολλαπλά συστήματα. Κάθε πράκτορας έχει την δυνατότητα να ορίσει τις υπηρεσίες που μπορεί να παρέχει με σκοπό της αναζήτησης αυτών από άλλους πράκτορες. [1]

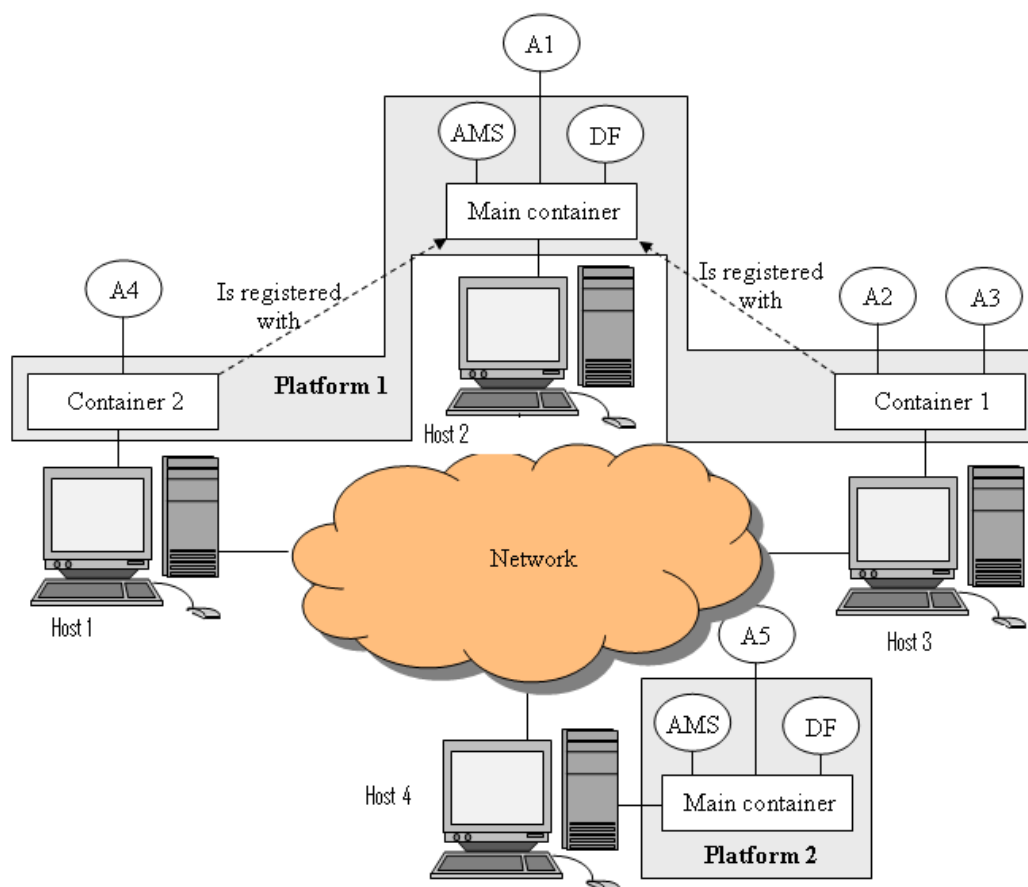
Βασικά στοιχεία της βιβλιοθήκης Jade

Σε κάθε εκκίνηση της βιβλιοθήκης δημιουργείτε ένα περιβάλλον εκτέλεσης όπου μπορούν να γεννηθούν Jade πράκτορες. Το περιβάλλον αυτό πρέπει να είναι πάντα ενεργό σε ένα συγκεκριμένο σύστημα πριν την εκτέλεση ενός ή περισσότερων πρακτόρων.

Containers, πλατφόρμες, AMS και DF

Κάθε runtime (στιγμιότυπο) που τρέχει στο Jade περιβάλλον εκτέλεσης ονομάζεται όπως έχουμε πει container και το σύνολο των ενεργών containers καλείται πλατφόρμα. Ένα μοναδικό βασικό Container (Main- Container) πρέπει να είναι πάντα ενεργό σε μία πλατφόρμα και τα υπόλοιπα containers εγγράφονται σε αυτό όταν γεννιούνται. Κατά συνέπεια το πρώτο container που ξεκινά σε μία πλατφόρμα πρέπει να είναι το βασικό container ενώ όλα τα υπόλοιπα πρέπει να είναι τα «κανονικά» container στα οποία πρέπει να είναι γνωστή η διεύθυνση του βασικού container.

Εάν ξεκινήσει ένα διαφορετικό βασικό container κάπου στο δίκτυο τότε συγκροτείται μια καινούρια πλατφόρμα. Σε αυτό το βασικό container μπορούν να εγγραφούν νέα κανονικά containers.



Εικόνα 18 Αρχιτεκτονική της JADE με παράδειγμα.

Στο σχήμα που δανειζόμαστε από το site της βιβλιοθήκης, παρουσιάζεται η παραπάνω ιδέα μέσω ενός σεναρίου όπου υπάρχουν δύο πλατφόρμες Jade και κάθε μία φέρει 3 και 1 containers αντίστοιχα. Οι agents στην Jade ταυτοποιούνται από ένα μοναδικό όνομα και γνωρίζοντας κάθε ένας τα ονόματα του άλλου μπορούν να επικοινωνούν με διαφάνεια

ασχέτως της πραγματικής τους τοποθεσίας. Μπορούν επομένως να επικοινωνούν είτε βρίσκονται στο ίδιο container (agents A2 και A3), είτε σε διαφορετικά containers στην ίδια πλατφόρμα (A1 και A2) είτε σε διαφορετικές πλατφόρμες (A4 και A5). Το βασικό πλεονέκτημα της υλοποίησης της Jade είναι ότι δεν χρειάζεται να γνωρίζει ο προγραμματιστής πώς ακριβώς δουλεύει η βιβλιοθήκη απλά πρέπει να ξεκινήσει την εκτέλεσή της πριν την γέννηση των πρακτόρων. Το main container εκτός του ότι έχει τη δυνατότητα να δέχεται εγγραφές άλλων containers διαφέρει επιπλέον από τα κανονικά containers καθώς περιέχει δύο ειδικού σκοπού πράκτορες: τον AMS (Agent Management System) και τον DF (Directory Facilitator). Οι agents αυτοί ενεργοποιούνται ταυτόχρονα με την έναρξη του main container.

Ο AMS παρέχει υπηρεσίες απόδοσης ονομάτων (διασφαλίζει τη μοναδικότητα του ονόματος κάθε πράκτορα σε μία πλατφόρμα) και αναπαριστά τις υπηρεσίες που μπορούν να υλοποιηθούν στην πλατφόρμα (π.χ. είναι υπεύθυνος να παρέχεται τη δυνατότητα της γέννησης/θανάτωσης agents σε απομακρυσμένους containers υλοποιώντας μια αίτηση που έλαβε). Ο πράκτορας DF παρέχει υπηρεσίες Yellow Pages δηλαδή προσφέρει τη δυνατότητα σε έναν agent να βρει τους κατάλληλους agents που μπορούν να φέρουν εις πέρας τις υπηρεσίες που αυτός επιθυμεί. Κάθε πράκτορας κατά την «γέννηση» του έχει την δυνατότητα να δηλώσει στον DF τις υπηρεσίες που μπορεί να παρέχει και έτσι τοποθετεί εαυτόν στην λίστα του προς εξυπηρέτηση άλλων πρακτόρων

4

Μελέτη περίπτωσης Χρήσης και εργαλεία

4.1 Μελέτη περίπτωσης Χρήσης - Jade_QA_Agents

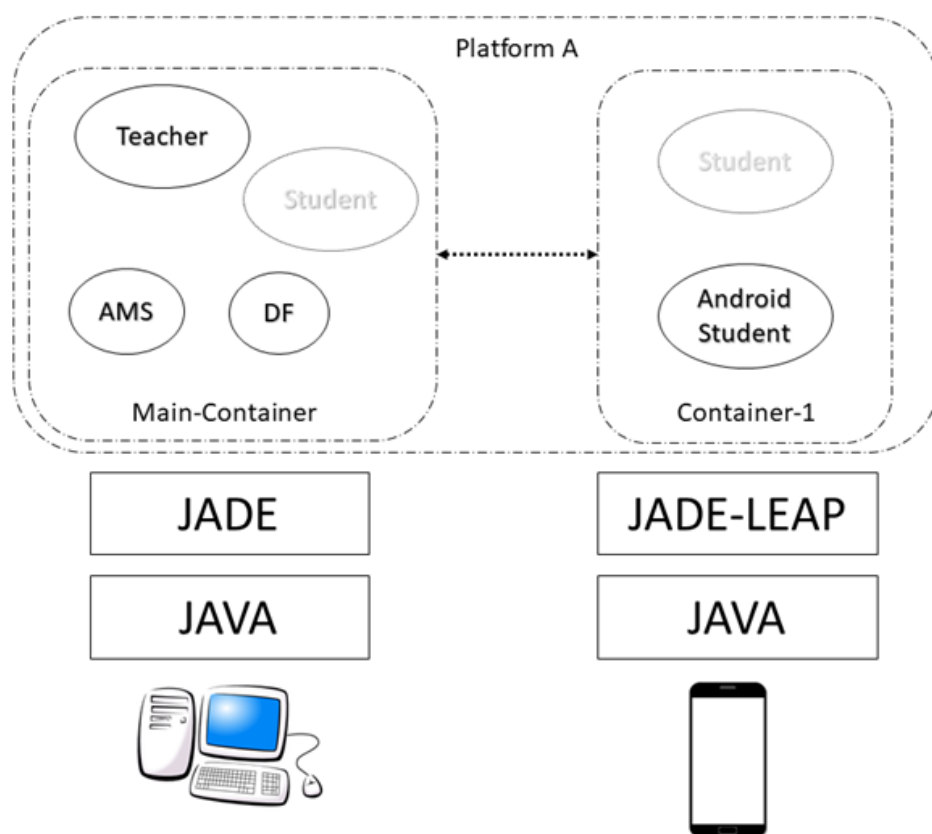
Στα πλαίσια της διπλωματικής εργασίας δημιουργήθηκε η περίπτωση χρήσης που ακολουθεί στο παρακάτω πλαίσιο, όπως ακριβώς αναφέρεται στην πρόσκληση της εργασίας.

Ανάπτυξη ενός συστήματος πρακτόρων μαθητή και εκπαίδευσης για την υποβοήθηση του εκπαιδευόμενου στον έλεγχο των γνώσεων του. Ο πράκτορας μαθητής θα έχει τη δυνατότητα να μεταναστεύει μεταξύ της κινητής συσκευής και του συστήματος μέσω του οποίου παρέχεται η εκπαίδευση και να επικοινωνεί με σχετικό πράκτορα εκπαίδευσης. Ο πράκτορας μαθητή θα έχει τη δυνατότητα να μεταφέρει διαγνωστικά τεστ δεξιοτήτων από το σύστημα στην κινητή συσκευή, έτσι ώστε να διεξάγεται τοπικά το τεστ από τον εκπαιδευόμενο και στη συνέχεια θα κοινοποιεί τις απαντήσεις στον πράκτορα εκπαίδευσης για έλεγχο και ανατροφοδότηση.

Από την μελέτη της εκφώνησης της περίπτωσης χρήσης γίνεται κατανοητό πως για την υλοποίηση θα γίνει παράλληλη χρήση 2 συστημάτων με το ένα να αναπτύσσεται σε περιβάλλον υπολογιστή και το δεύτερο σε περιβάλλον κινητής συσκευής. Η βιβλιοθήκη jade μέσω του προσθετού Jade-Leap απέκτησε υποστήριξη για εκτέλεση σε κινητές συσκευές. Το

Jade-Leap παρέχει την δυνατότητα εκτέλεσης σε συσκευές που υποστηρίζουν Java Micro Edition SDK, .NET framework και android περιβάλλον(από την έκδοση 2.1 και μετέπειτα) Για την υλοποίηση της εφαρμογής που θα εκτελείται σε κινητή συσκευή επιλέχθηκε να αναπτυχθεί στο δημοφιλέστερο λειτουργικό σύστημα υποστήριξης εφαρμογών Java που είναι το Android.

Στην παρακάτω εικόνα περιγράφεται σχηματικά το σύστημα που αναπτύχθηκε.



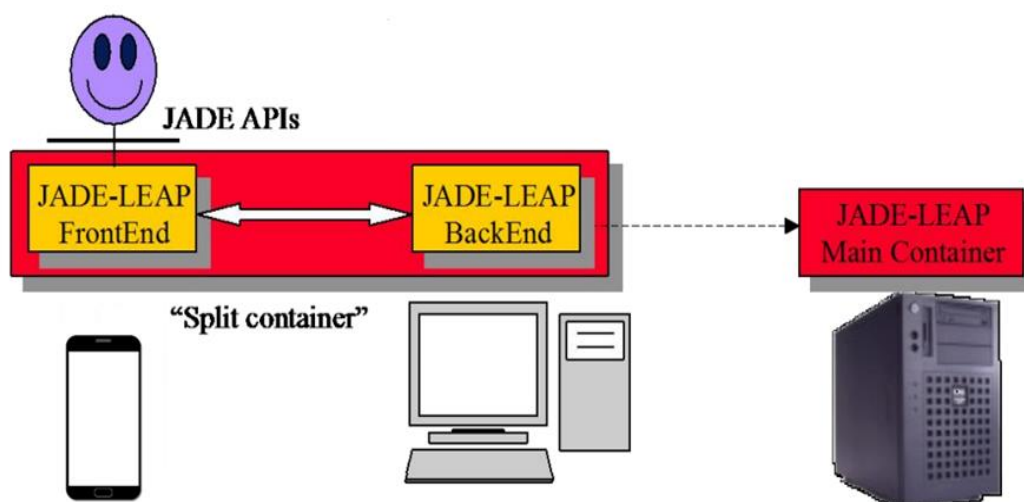
Εικόνα 19 Σύστημα Jade_QA_Agents

Αναλύοντας το σχήμα που απεικονίζει το σύστημα που δημιουργήθηκε για τις ανάγκες του project, αρχικά διακρίνεται ότι και τα δύο λειτουργικά συστήματα συνυπάρχουν κατά κάποιο τρόπο στην ίδια πλατφόρμα. Αυτός ο προσδιορισμός δημιουργήθηκε για να μας καταδείξει πως το Container-1 που δημιουργείται από την κινητή συσκευή συνδέεται με το main-container που έχει υπόσταση μέσα στο σύστημα του ηλεκτρονικού υπολογιστή.

Η κινητή συσκευή δημιουργεί το Container-1 σαν «αυτόνομο» container και το διατηρεί πλήρως στην κατοχή της. Αξίζει σε αυτό το σημείο, για να γίνει κατανοητή η ενέργεια αυτή, να καταγράψουμε τις επιλογές που δίνονται στην δημιουργία ενός Container από κινητή συσκευή.

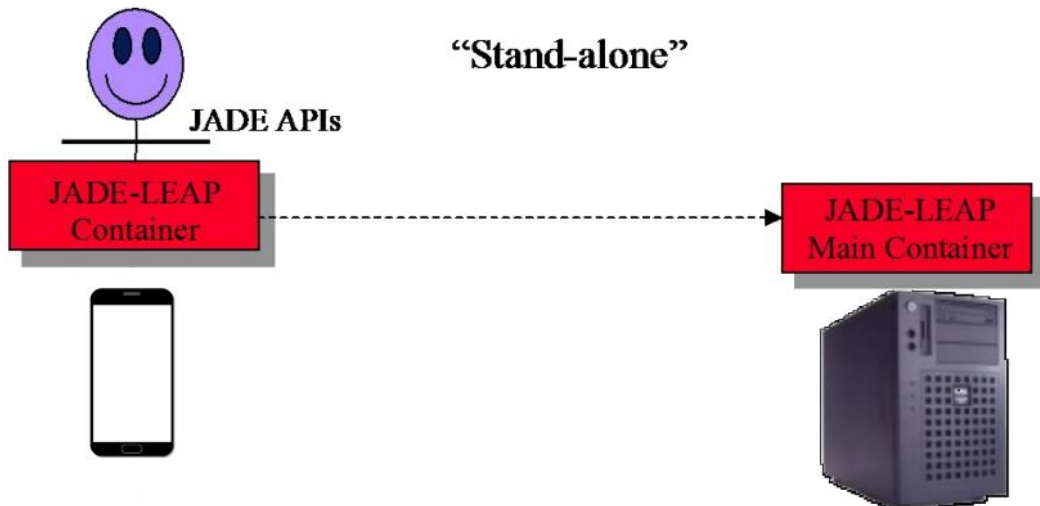
Το πρόσθετο Jade-Leap που έχει αναπτυχθεί δίνει την δυνατότητα δημιουργίας 2 ειδών container. Του split container (διαχωρισμένο) και του stand-alone container (αυτόνομο).

Το split container (βλέπε εικόνα 18) έχει δημιουργηθεί για να δώσει την δυνατότητα στις κινητές συσκευές να διατηρούν στην συσκευή μόνο ένα μέρος από την λειτουργικότητα (front-end, εμφανιζόμενη) και να μεταφέρουν τις ενεργοβόρες διαδικασίες σε μια άλλη ενδιάμεση συσκευή (back-end). Η συσκευή αυτή αναλαμβάνει να διατηρεί στην κατοχή της το container με τους ενεργούς πράκτορες με συνέπεια αυτή να είναι υπεύθυνη να εκτελέσει υπολογισμούς όπως και τις αποστολές-λήψεις μηνυμάτων με άλλες συσκευές. Αυτή η λειτουργικότητα έχει σαν πλεονέκτημα την ελάχιστη δυνατή κατανάλωση πόρων και δεδομένων δικτύου από την κινητή συσκευή. Το μεγάλο μειονέκτημα που επηρέασε και την επιλογή του project είναι ότι δεν υποστηρίζει την δυνατότητα μετανάστευσης και κλωνοποίησης των πρακτόρων που είναι ενεργοί στο container.



Εικόνα 20 JADE_LEAP Split Container

Η δεύτερη επιλογή που δίνεται από την jade-learn είναι η δυνατότητα δημιουργίας ενός stand-alone container (βλέπε εικόνα 21) το οποίο παραμένει στην κινητή συσκευή διατηρώντας τους πράκτορες. Σε αυτή την περίπτωση οι διαδικασίες υπολογισμού και επικοινωνίας με άλλα συστήματα γίνεται απευθείας από την συσκευή. Η μέθοδος αυτή επιτρέπει την δημιουργία κλώνων των πρακτόρων όπως και την μετανάστευση τους από και προς άλλη συσκευή με ενεργό runtime jade. Όπως είναι κατανοητό από τα παραπάνω αυτή είναι η μεθοδολογία που ακολουθήθηκε για την υλοποίηση του project καθώς στόχος μας είναι η μετανάστευσή ενός πράκτορα «μαθητή» μεταξύ των συσκευών. Να σημειώσουμε ότι το container αυτό που δημιουργείται θα πρέπει να συνδέεται με ένα main container για να μπορέσει να έχει υπόσταση, κάτι που από την πλευρά μας μας επιτρέπει να έχουμε καλύτερο έλεγχο και ανατροφοδότηση μέσα από το γραφικό περιβάλλον που μας παρέχει η jade.



Εικόνα 21 JADE-LEAP Stand-alone Container

Συνεχίζοντας τη ανάλυση, κατά την ενεργοποίηση του runtime της Jade στο σύστημα του υπολογιστή, δημιουργείται αυτόματα ένα container με το όνομα *Main-container* καθώς έχουμε προχωρήσει στην δημιουργία ενός αυτόνομου συστήματος που δεν προσκολλάται σε κάποιο υπάρχον. Στον *Main-container*, πέρα από τους πράκτορες AMS και DF που εξορισμού δημιουργούνται από την βιβλιοθήκη, συμπεριλαμβάνονται οι πράκτορες *Teacher* και *Student* με τον κάθε ένα να προσδιορίζει από το ονόματα του τον ρόλο που έχει αναλάβει στο σύστημα μας.

Ο πράκτορας *Teacher*, έχοντας αναλάβει να προσομοιώσει τον ρόλο του καθηγητή, έχει στην κατοχή του το ερωτηματολόγιο και είναι υπεύθυνος μόλις λάβει αίτημα από έναν μαθητή, να το παραδώσει σε έναν νέο πράκτορα με το όνομα *Student* που θα μεταναστεύσει στην κινητή συσκευή του μαθητή. Ο πράκτορας *Student* προσδιορίζεται από το σχήμα ότι έχει παρουσία και στα δυο container για να γίνει κατανοητό ότι αυτός είναι που μεταναστεύει από το σύστημα του υπολογιστή στην κινητή συσκευή μεταφέροντας το ερωτηματολόγιο και τελικά επιστρέφει πίσω με αυτό συμπληρωμένο από τον φυσικό μαθητή της συσκευής.

Ο ρόλος του πράκτορά *Android Student* που συναντάμε στην κινητή συσκευή είναι να ζητήσει από τον πράκτορα *Teacher* ένα νέο ερωτηματολόγιο και σε συνεργασία με τον πράκτορα *Student* να αλληλοεπιδράσει μέσω της κινητής συσκευής με τον τελικό χρήστη.

4.2 Εργαλεία υλοποίησης Jade_QA_Agents

4.2.1 Java

Θεμέλιος λίθος για να γίνει πράξη η περίπτωση χρήσης που περιγράψαμε παραπάνω είναι στο υπολογιστικό μας σύστημα να προβούμε στην εγκατάσταση του πακέτου ανάπτυξης (JDK, Java Development Kit) της γλώσσας προγραμματισμού java. Δανειζόμενοι μια σύντομη περιγραφή για τη γλώσσα αυτή μέσα από την βιβλιοθήκη Wikipedia μπορούμε να πούμε ότι «Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems. Η επίσημη εμφάνιση της Java αλλά και του HotJava (πλοηγός με υποστήριξη Java) στη βιομηχανία της πληροφορικής έγινε το Μάρτιο του 1995 όταν η Sun την ανακοίνωσε στο συνέδριο Sun World 1995. Η Java έκτοτε ακολουθεί μία ανοδική πορεία και είναι πλέον μία από τις πιο δημοφιλείς γλώσσες στον χώρο της πληροφορικής. Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Αυτό επιτυγχάνεται μέσω της εκτέλεσης του μεταγλωττισμένου πηγαίου κώδικα σε Εικονική Μηχανή (Virtual Machine) που μπορεί να λειτουργήσει σε κάθε υποστηριζόμενο λειτουργικό σύστημα».

Η εγκατάσταση της τελευταίας διαθέσιμης έκδοσης έγινε από τον παρακάτω σύνδεσμο <https://www.java.com/en/download> . Κατά την ανάπτυξη του συστήματος έγινε χρήση του JDK 1.8.0_60. Για την δημιουργία των προγραμμάτων στην γλώσσα αυτή έγινε χρήση των εργαλείων Notepad++ (έκδοση 7.3.3) και Eclipse (4.6.3).

4.2.2 Εγκατάσταση της JADE

Η βιβλιοθήκη που θα μας βοηθήσει στην δημιουργία και λειτουργία των πρακτόρων κινητής μάθησης είναι πρόσβαση μέσα από το site των δημιουργών της <http://jade.tilab.com>.

JAVA Agent DEvelopment Framework
is an open source platform for peer-to-peer agent based applications

JADE (Java Agent DEvelopment Framework) is a software Framework fully implemented in the Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the [FIPA specifications](#) and through a set of [graphical tools](#) that support the debugging and deployment phases. A JADE-based system can be distributed across machines (which not even need to share the same OS) and the configuration can be controlled via a [remote GUI](#). The configuration can be even changed at run-time by moving agents from one machine to another, as and when required. JADE is completely implemented in Java language and the minimal system requirement is the version 5 of JAVA (the run time environment or the JDK).

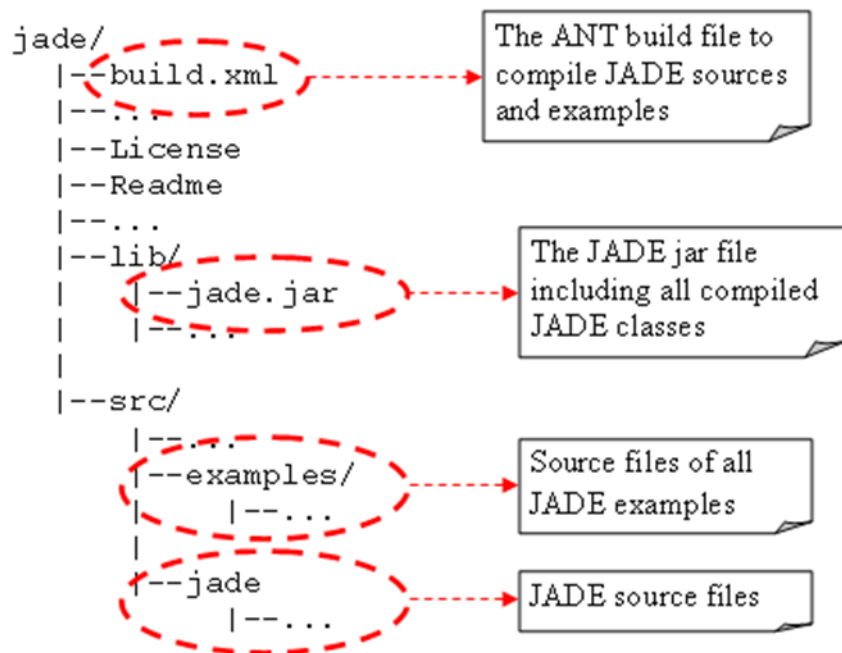
Latest news

- 23/12/2015
JADE 4.4, WADE 3.5 and AMUSE 1.5 have been released
- 11/12/2014
JADE 4.3.3, WADE 3.4 and AMUSE 1.0 have been released

Εικόνα 22 jade.tilab.com

Από την ενότητα download του site μπορεί η χρήστης να κατεβάσει την τελευταία διαθέσιμη έκδοση, που είναι η 4.5 με ημερομηνία 08/06/2017. Η εργασία βασίστηκε στην έκδοση 4.4.0 με ημερομηνία 23/12/2015 καθώς η νέα έκδοση δεν ήταν ακόμα διαθέσιμη την στιγμή της δημιουργίας του συστήματος. Όπως χαρακτηριστικά αναφέρεται έχει δημιουργηθεί η βιβλιοθήκη κάνοντας χρήση του μεταγλωττιστή (compiler) jdk 1.6.

Τα αρχεία που δίνονται είναι συμπιεσμένα και περιλαμβάνουν τα JADE-bin-4.4.0.zip, JADE-doc-4.4.0.zip, JADE-examples-4.4.0.zip, JADE-src-4.4.0.zip με τα ονόματα να είναι και προσδιοριστικά του τι περιέχει το κάθε ένα. Αρχικά ακολουθώντας τον οδηγό που είναι αναρτημένος στο site καλείτε να γίνει αποσυμπίεση του με τέτοιο τρόπο ώστε να δημιουργηθεί η δομή που παρουσιάζεται στην παρακάτω εικόνα

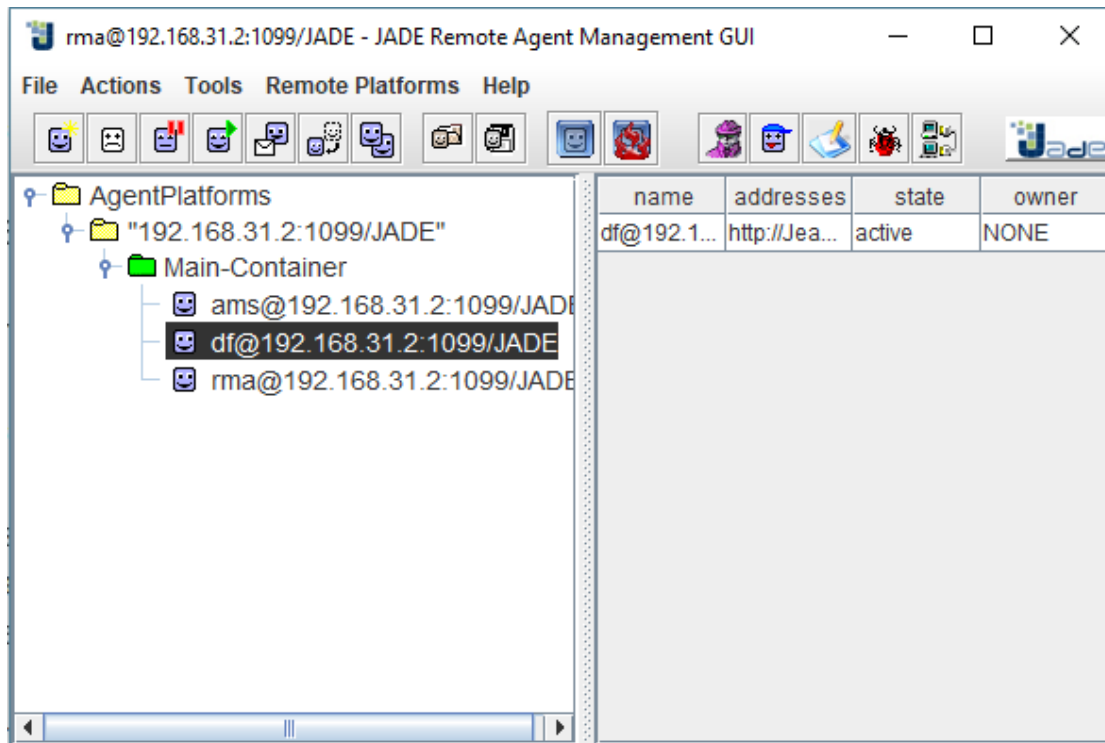


Εικόνα 23 Δομή αρχείων Jade

Το site διαθέτει ικανό οδηγό για να μπορέσει ένας αρχάριος χρήστης να ξεκινήσει την ενασχόληση του.[ΠΗΓΗ] Η βιβλιοθήκη jade έχει ενσωματωμένη την δυνατότητα γραφικής απεικόνισης των πρακτόρων που είναι ενεργοί σε ένα περιβάλλον που αποτελεί ένα χρήσιμο εργαλείο στην δημιουργία, εποπτεία και αποσφαλμάτωση του συστήματος που δημιουργήσαμε. Η αρχικοποίηση του περιβάλλοντος jade γίνεται μέσα από την γραμμή εντολών του εκάστοτε λειτουργικού.

Μια πρώτη εντολή για την εκκίνηση της βιβλιοθήκης με γραφική απεικόνιση είναι η παρακάτω:

```
java -cp D:\jade\lib\jade.jar jade.Boot -gui
```

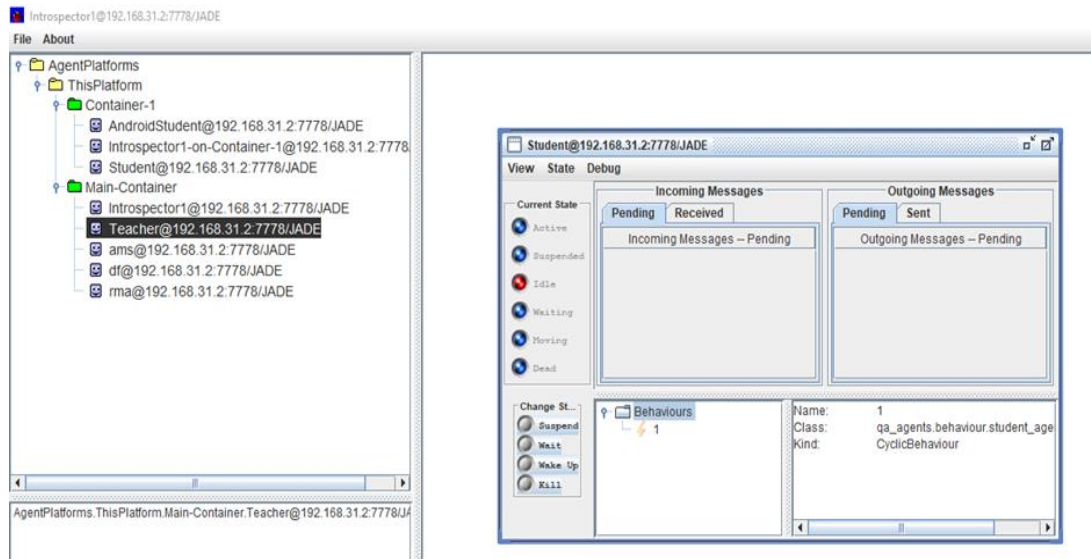


Εικόνα 24 Γραφικό περιβάλλον Jade

Μέσα από το γραφικό περιβάλλον δίνεται η δυνατότητα να προβούμε σε αρκετές από τις ενέργειες που επιθυμούμε οι οποίες μπορούν να πραγματοποιηθούν και μέσω κώδικα στις δημιουργημένες κλάσεις. Το κύριο χαρακτηριστικό που προσφέρει επιπρόσθετα το γραφικό περιβάλλον είναι ότι παρέχει στον χρήστη την δημιουργία πρακτόρων που λειτουργούν σαν εργαλεία υποβοήθησης για την παρακολούθηση των ενεργειών που λαμβάνουν χώρα, την παρακολούθηση μηνυμάτων, την δημιουργία αρχείων καταγραφής, τον εντοπισμό σφαλμάτων και την αποσφαλμάτωση τους.

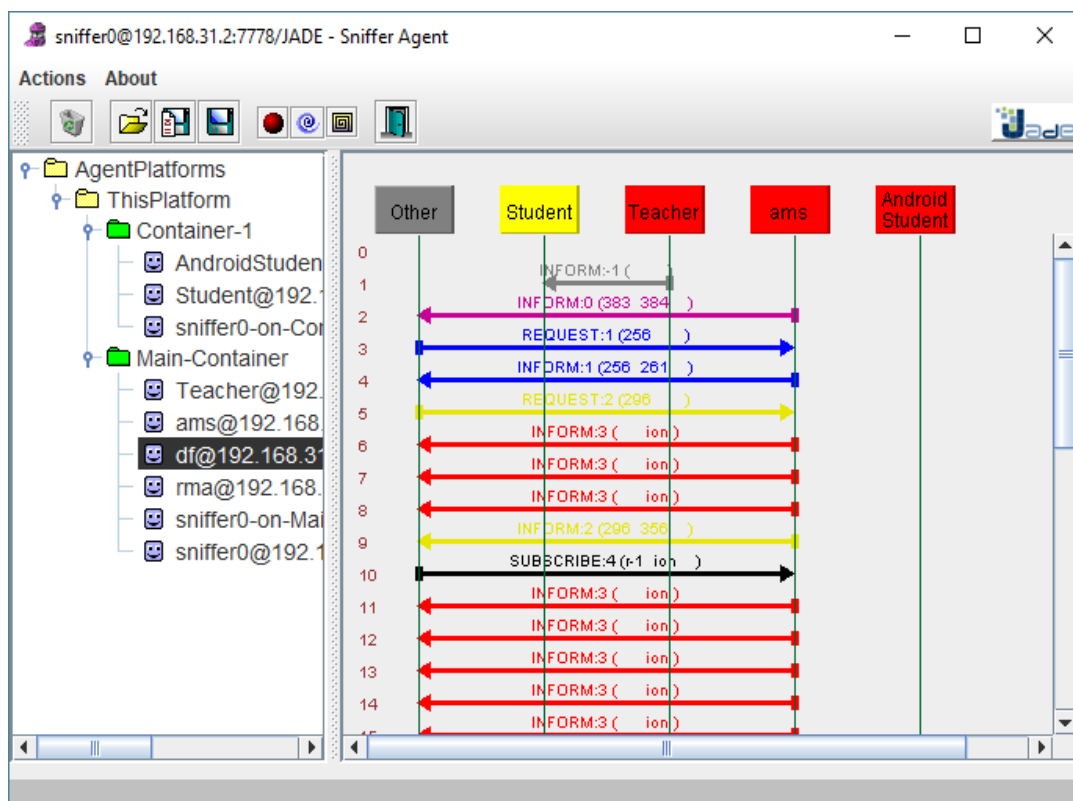
Ιδιαίτερη αναφορά χρήζει στον πράκτορα που Introspector και στον πράκτορα Sniffer

Ο πράκτορας introspector δίνει την δυνατότητα γραφικής απεικόνισης της κατάστασης οποιουδήποτε πράκτορα που είναι ενεργός μέσα σε κάποιο Container παρέχοντας παράλληλα πληροφορίες για την κατάσταση που βρίσκεται (ενεργή, σε αναμονή, σε παύση κ.α.) , για την συμπεριφορά που έχει ενεργή και για τα μηνύματα που έχει ανταλλάξει μέχρι τώρα.



Εικόνα 25 Πράκτορας Introspector

Ακόμα μέσα από το γραφικό περιβάλλον συναντάμε τον πράκτορα sniffer ο οποίος μας προσφέρει την δυνατότητα να εκμαιεύσουμε και να αναλύσουμε τα μηνύματα που ανταλλάσσονται μεταξύ των πρακτόρων με χρονολογική σειρά.



Εικόνα 26 Πράκτορας sniffer

Καθώς η συγγραφή του κώδικα της εργασίας έγινε στην γλώσσα java, όπως ήδη αναφέραμε, οι εφαρμογές που χρησιμοποιήσαμε ήταν το eclipse με παράλληλη χρήση του notepad++ ανάλογα με τις ανάγκες μας. Για μπορέσει να γίνει η μεταγλώττιση του κώδικα java απαιτείται να δηλωθεί και η βιβλιοθήκη jade που χρησιμοποιούμε ώστε να αναγνωρίσει και

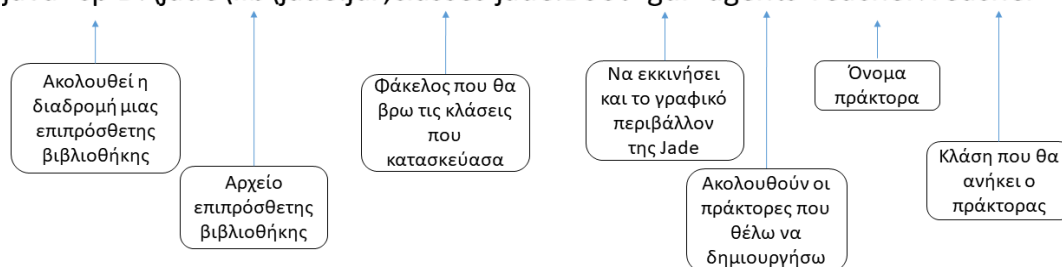
να επεξεργαστεί τον κώδικα μας συνδυαστικά με τις κλάσεις που περιλαμβάνονται στην βιβλιοθήκη μας. Η χρήση αυτής παρουσιάζεται στην παρακάτω εντολή στην γραμμή εντολών μας

```
javac -classpath D:\jade\lib\jade.jar -d classes D:\folder\*.java
```

Στην εντολή αυτή βλέπουμε ότι δηλώνεται η διαδρομή της jade και η διαδρομή που υπάρχουν τα αρχεία προς μεταγλώττιση.

Ένα παράδειγμα εκκίνησης πράκτορα στην jade παρουσιάζεται στην παρακάτω εικόνα:

```
java -cp D:\jade\lib\jade.jar;classes jade.Boot -gui -agents Teacher:Teacher
```

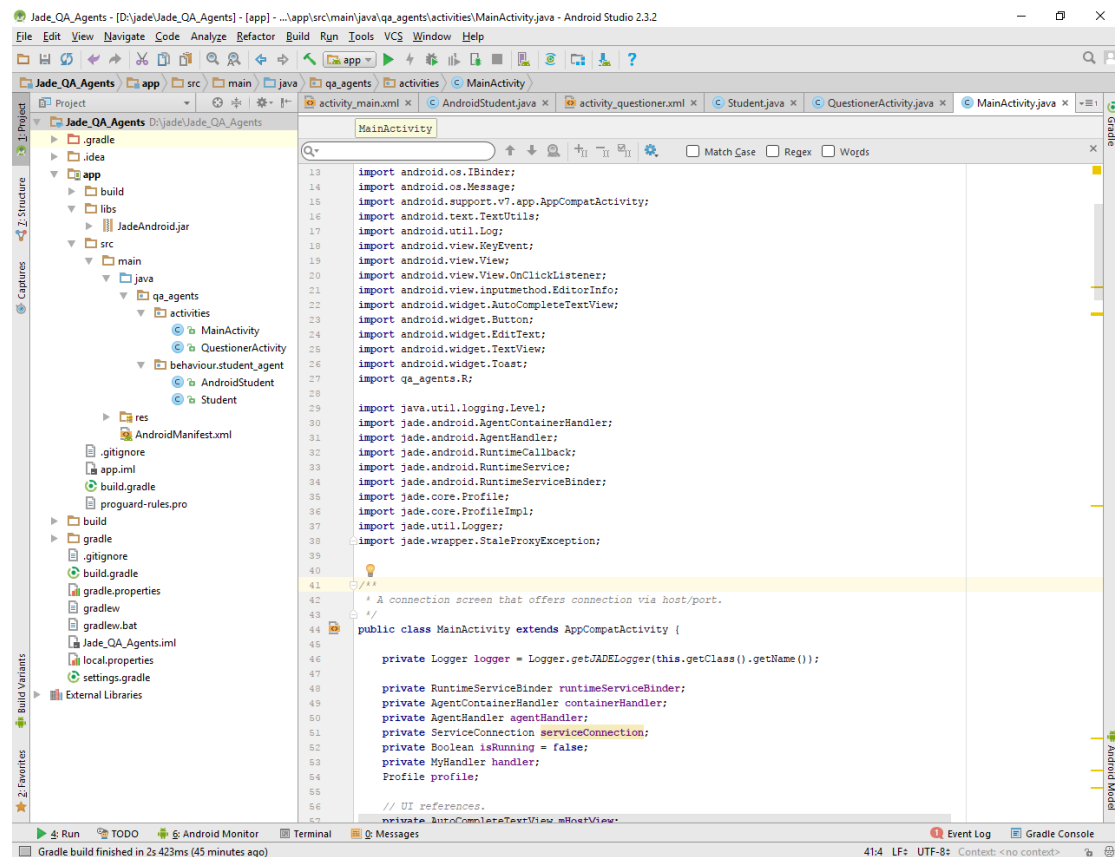


Εικόνα 27 Εκκίνηση πράκτορα jade

Το project που δημιουργήθηκε για την εργασία μας λαμβάνει χώρα σε 2 πλατφόρμες λειτουργικών συστημάτων. Σε υπολογιστή windows και σε κινητή συσκευή λειτουργικού Android.

4.2.3 Android Studio και Jade-LEAP

Για την δημιουργία της εφαρμογής στην συσκευή android επιλέχθηκε να γίνει χρήση του εργαλείου-προγράμματος που παρέχει ο κατασκευαστής του λειτουργικού συστήματος(Google) Android Studio. Η εγκατάσταση της εφαρμογής έγινε μέσα από τον παρακάτω σύνδεσμο <https://developer.android.com/studio/index.html>.



Εικόνα 28 Android Studio

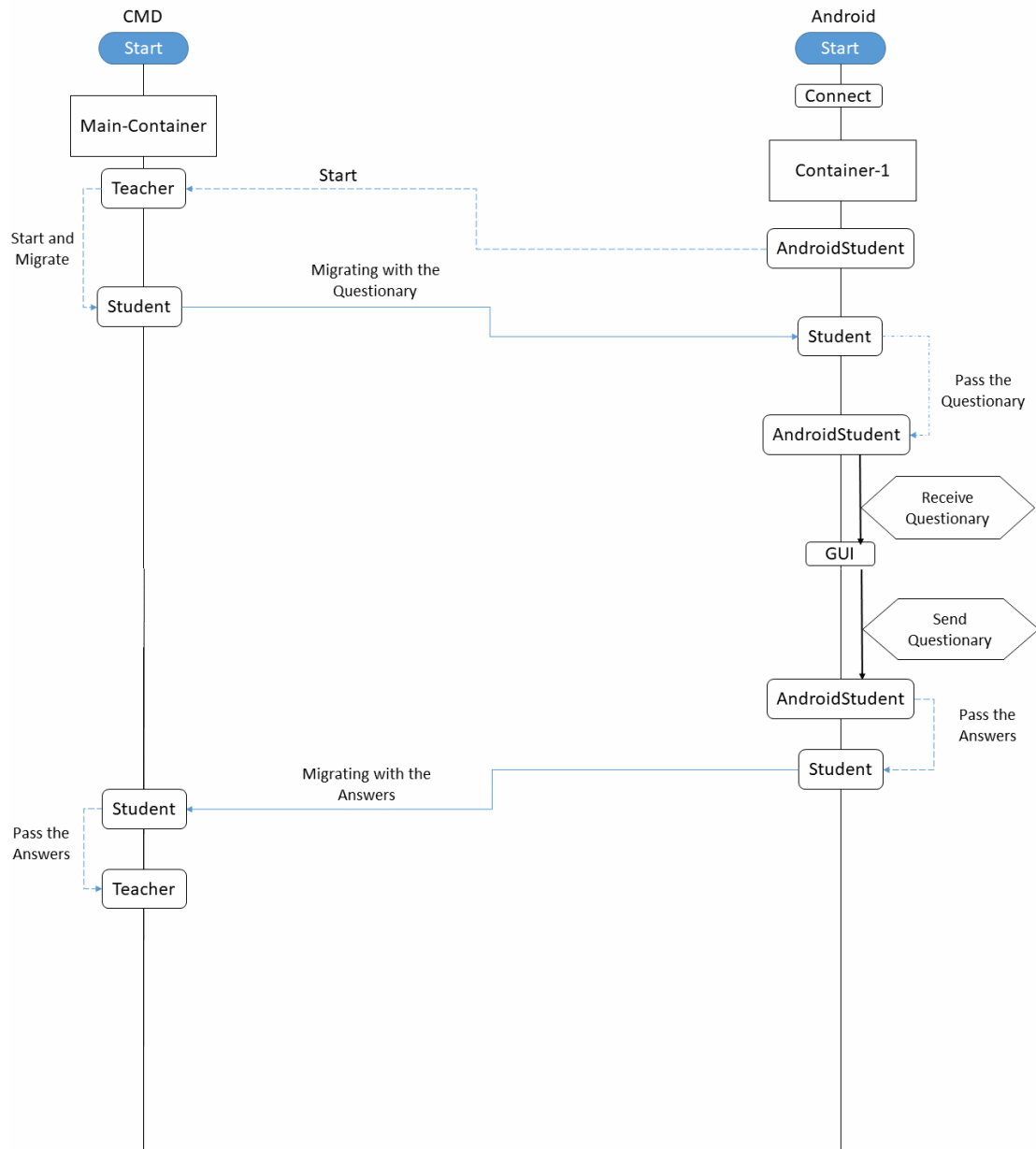
Για να μπορέσουμε να προσθέσουμε την λειτουργικότητα των πρακτόρων θα πρέπει να συμπεριλάβουμε στις βιβλιοθήκες της εφαρμογής την Jade-LEAP που όπως αναφέραμε είναι το πρόσθετο που έχει αναπτυχθεί για την υποστήριξη της Jade σε κινητές συσκευές υποστήριξης Java. Το αρχείο του πρόσθετου το βρίσκουμε και αυτό μέσα στο επίσημο site της jade. Η τελευταία διαθέσιμη έκδοση είναι η 4.5.0 . Αφού αποθηκεύσουμε το αρχείο στο υπολογιστή μας θα πρέπει ανοίγοντάς το να απομονώσουμε το αρχείο JadeAndroid.jar και να το προσθέσουμε τον φάκελο libs (Jade_QA_Agents\app\libs) του project που έχουμε δημιουργήσει. Κατόπιν αυτού το android studio θα έχει την δυνατότητα διαχείρισης των κλάσεων που ενσωματώνονται στην βιβλιοθήκη μας. Οι δοκιμές στην πορεία των διάφορων εκδόσεων της εφαρμογής Jade QA Agents έγιναν σε πραγματικές συσκευές που διαθέτουν εγκατεστημένο το λειτουργικό android 5.1 και το android 7.1.1.

5

Jade_QA_Agents Υλοποίηση

5.1 Ανάλυση δημιουργίας εφαρμογής

Η αρχή της ανάλυσης του συστήματος που δημιουργήσαμε ώστε να υλοποιήσουμε την περίπτωση χρήσης που περιεγράφηκε νωρίτερα οφείλει να γίνει παρατηρώντας το διάγραμμα ροής που ακολουθεί στην εικόνα.



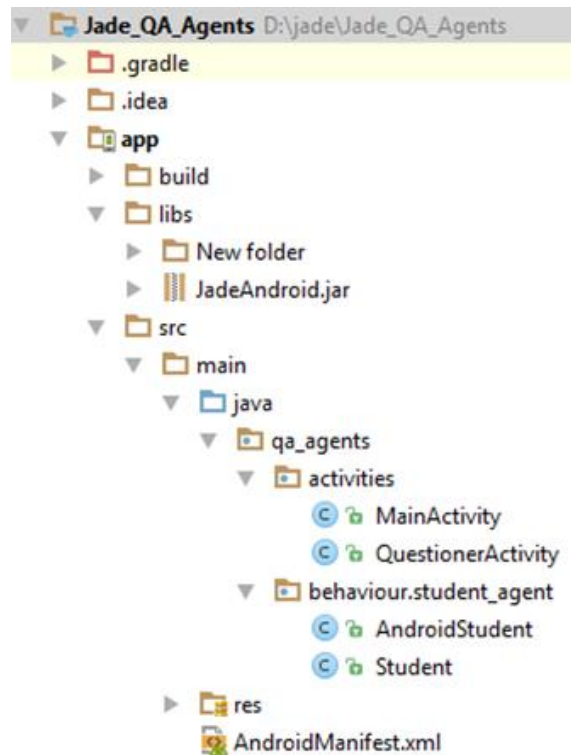
Εικόνα 29 Διάγραμμα Ροής

Στο διάγραμμά ροής παρουσιάζεται σχηματικά η συνύπαρξη των 2 συστημάτων, η ανταλλαγή μηνυμάτων μεταξύ των πρακτόρων, η μετανάστευση του πράκτορα *student*, η παρουσίαση και η αποστολή του ερωτηματολογίου καθώς και η τελική μετανάστευση του πράκτορα *student* στον *main-container* παραδίδοντας τα αποτελέσματα του ερωτηματολογίου στον πράκτορα *Teacher* για ανατροφοδότηση.

Καθώς γίνεται κατανοητό ότι η αλληλουχία ενεργειών έχει αρχή τον χρήστη της κινητής συσκευής θα ξεκινήσουμε την αναφορά μας στον κώδικα υλοποίησης από τα της κινητής συσκευής.

5.1.1 Δημιουργία εφαρμογής- Android

Το όνομα που δόθηκε στην εφαρμογή που εκτελείτε στην συσκευή android είναι Jade_QA_Agents και οι κλάσεις που περιλαμβάνει εμφανίζονται στην παρακάτω εικόνα:

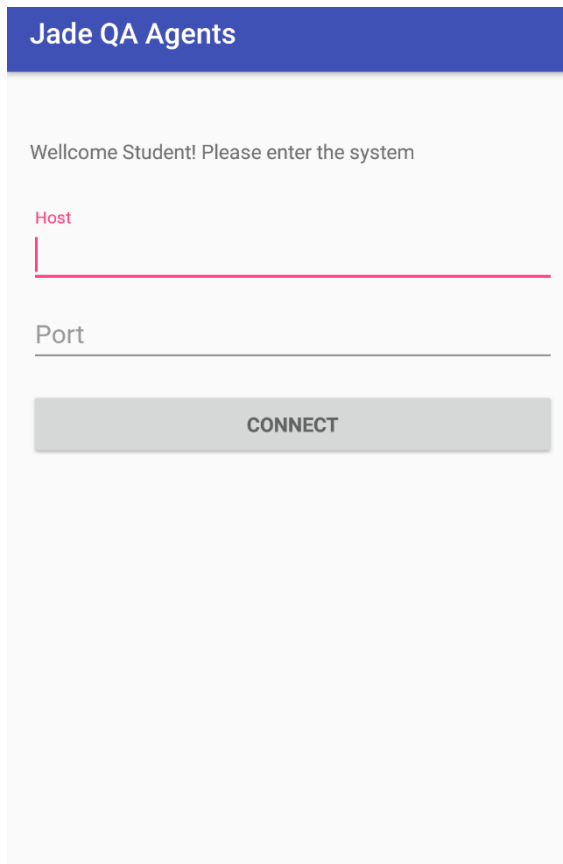


Εικόνα 30 Δομή στο Android

Οι κλάσεις MainActivity και QuestionerActivity, μαζί με τα αντίστοιχα xml αρχεία τους είναι υπεύθυνες για την δημιουργία της γραφικής διασύνδεσης και αλληλεπίδρασης της εφαρμογής με τον χρήστη.

Οι κλάσεις AndroidStudent και Student αποτελούν τους πράκτορες που ενεργοποιούνται στην εφαρμογή μας.

Η πρώτη εικόνα που θα συναντήσει ο χρήστης ενεργοποιώντας την εφαρμογή είναι η παρακάτω εικόνα:



Jade QA Agents

Wellcome Student! Please enter the system

Host

Port

CONNECT

Εικόνα 31 Πρώτη εικόνα εφαρμογής

Σε αυτό παρατηρούμε ότι η χρήστης καλείτε να βάλει τα στοιχεία σύνδεσης ώστε η εφαρμογή να αναζητήσει με βάση αυτά αν υπάρχει ενεργό runtime της jade στην δηλωμένη διεύθυνση. Ο χρήστης καλείτε να εισάγει την διεύθυνση ip που είναι ενεργός ο server και την «πόρτα» που έχει οριστεί ότι μπορεί να συνδεθεί μια εφαρμογή που υλοποιεί την jade. Να τονίσουμε ότι σαν προεπιλεγμένη «πόρτα» η βιβλιοθήκη έχει ορίσει την 1099 για πρόσβασή μέσω τοπικού δικτύου και την 7778 για πρόσβαση μέσω http. Για την υλοποίηση της δική μας εφαρμογής την έχουμε μεταβάλει σε 7778 και για την «τοπική» επικοινωνία.

Η διαδικασία εισαγωγής της διεύθυνσης σύνδεσης προσθέτει στο σύστημά μας το χαρακτηριστικό της ευελιξίας καθώς υποστηρίζει με αυτό τον τρόπο την δυνατότητα μετεγκατάστασης του σε διαφορετικό περιβάλλον χωρίς να χρειαστεί μεταβολή του κώδικα και εκ νέου εγκατάσταση της εφαρμογής στην κινητή συσκευή.

```

private String host = "";
private String port = "";
private void attemptLogin() {

boolean cancel = false;
// Check for a valid address.
if (TextUtils.isEmpty(host)) {
    mHostView.setError(getString(R.string.error_field_required));
    focusView = mHostView;
    cancel = true;
}
// Check for a valid password, if the user entered one.
if (TextUtils.isEmpty(port)) {
    mPortView.setError(getString(R.string.error_invalid_password));
    focusView = mPortView;
    cancel = true;
}}
start(host, port);

```

Κώδικας IMainActivity.java

Ξεκινώντας την ανάλυση του κώδικα της εφαρμογής, στο παραπάνω πρώτο απόσπασμα της MainActivity βλέπουμε τον έλεγχο που γίνεται ώστε να επιβεβαιώσουμε ότι δόθηκαν τιμές στα πεδία. Τα μηνύματα που αποστέλλει η setError() εμφανίζονται μέσα στο πεδίο του host ή του port αντίστοιχά.

```

private void start(final String host, final String port) {
    if (runtimeServiceBinder == null) {
        serviceConnection = new ServiceConnection() {
            @Override
            public void onServiceConnected(ComponentName name, IBinder service) {
                runtimeServiceBinder = (RuntimeServiceBinder) service;
                profile = new ProfileImpl(false);
                profile.setParameter(Profile.MAIN_HOST, host);
                profile.setParameter(Profile.MAIN_PORT, port);
                startJADE(profile);
            }
            @Override
            public void onServiceDisconnected(ComponentName name) {
                runtimeServiceBinder = null;
            }
        };
        bindService(new Intent(getApplicationContext(), RuntimeService.class),
            serviceConnection, Context.BIND_AUTO_CREATE);
    }
}

```

Κώδικας 2 MainActivity.java

Πριν ξεκινήσουμε την σύνδεση μας με το main-container πρέπει να γίνει η δημιουργία ενός service που θα συνδεθεί με την εφαρμογή μας ώστε να γίνει κατάλληλος χειρισμός της επικοινωνίας και παράλληλα να δοθεί κάποια προτεραιότητα καθώς στο λειτουργικό σύστημα android η δημιουργία ενός service και η σύνδεση του με κάποια εφαρμογή επιτρέπει αφενός να διατηρείτε ενεργή στο παρασκήνιο και αφετέρου να είναι περισσότερο «ανθεκτική» στην χαμηλή διαθεσιμότητα πόρων.

Παρατηρούμε στο πλαίσιο του κώδικα ότι υπάρχουν αντικείμενα `runtimeServiceBinder` (περιλαμβάνεται στην βιβλιοθήκη `jade` και αποτελεί επέκταση της `android.os.Binder`) και `serviceConnection` τα οποία μας βοηθούν στην δημιουργία και στην σύνδεση ενός `service` με την εφαρμογή μας. Υπεύθυνη για αυτό είναι η μέθοδος `bindService()`. Ορίζουμε μέσω του αντικειμένου `Intent` ότι η σύνδεση του νέου `service` θα πρέπει να γίνει με όλη την ενεργή εφαρμογή. Αυτό ορίζεται από την `getApplicationContext()` καθώς επιστρέφει όλα τα ενεργά μέρη της εφαρμογής κατευθύνοντας έτσι αυτή την σύνδεση προς όλη την εφαρμογή. Ακόμα, βλέπουμε ότι η μεταβίβαση των τιμών των πεδίων που εισήγαγε ο χρήστης γίνεται μέσω του αντικειμένου `profile` που δίνεται σαν όρισμα στην κλήση της μεθόδου `startJADE()`.

```
private void startJADE(Profile profile) {
    Log.i("Platform", "Creating Container!\n");
    runtimeServiceBinder.createAgentContainer(profile, new Runtime-
    Callback<AgentContainerHandler>() {
        @Override
        public void onSuccess(AgentContainerHandler agentContainerHandler) {
            Log.i("Platform", "Container successfully created!\n");
            containerHandler = agentContainerHandler;
            handler.sendSuccess();

            //Activity transition
            Intent intent = new Intent(MainActivity.this, QuestionerActiv-
            ity.class);
            startActivity(intent);

            startAgent("AndroidStudent");
        }
        @Override
        public void onFailure(Throwable throwable) {
            Log.i("Platform", "start jade container failed!\n");
            handler.sendError();
        }
    });
}
```

Κώδικας 3 MainActivity.java

Στην συνέχεια , μεταφέρουμε την προσοχή μας στην μέθοδο δημιουργίας του `Container-1`. Καλούμε την μέθοδο `createAgentContainer()` «πάνω» στο αντικείμενο `runtimeServiceBinder` που μόλις δημιουργήσαμε δίνοντας σαν όρισμα τα στοιχεία σύνδεσης που έχει εισάγει ο χρήστης. Το αντικείμενο `containerHandler` είναι ο διαχειριστής του `container` μέσα από τον οποίο μπορούμε να εκτελέσουμε ενέργειες πάνω στον εκάστοτε `container`.

Μέσω του αντικειμένου `intent` και της κλήσης της μεθόδου `startActivity()` μεταφέρουμε την ροή της εφαρμογής στην επόμενο `activity` που επιθυμούμε να παρουσιαστεί στον χρήστη. Πριν τελειώσει ο εκτελούμενος κώδικας της `startJADE()` γίνεται κλήση της μεθόδου `startAgent()` με παράμετρο το `string` που θα αποτελέσει το όνομα του νέου πράκτορα. Να σημειώσουμε εδώ ότι η κλήση της μεθόδου `Log.i()` που παρατηρείτε, δεν προσφέρει κάποιο μήνυμα για τον χρήστη στην οθόνη του αλλά χρησιμοποιείτε εσωτερικά για την αποσφαλμάτωση από τον

προγραμματιστή και ως εκ τούτου σε επόμενα αποσπάσματα του κεφαλαίου θα αφαιρεθεί για λόγους καλύτερης παρουσίασης του κώδικα.

```
private void startAgent(final String nickname) {
    containerHandler.createNewAgent(nickname, "qa_agents.behaviour.student_agent.AndroidStudent", null, new RuntimeCallback<AgentHandler>() {
        @Override
        public void onSuccess(AgentHandler ah) {
            agentHandler = ah;

            try {
                agentHandler.getAgentController().start();
            }
        }
    });
}
```

Κώδικας 4 MainActivity.java

Τελευταία μέθοδος της MainActivity.java που θα σταθούμε είναι η startAgent(). Εδώ βλέπουμε ότι επιχειρείται η δημιουργία πράκτορα με κλήση της μεθόδου createNewAgent() που περιέχει η βιβλιοθήκη μας και για να μπορέσει να την χειριστεί κατάλληλα καλείτε πάλι «πάνω» στο αντικείμενο containerHandler. Στα ορίσματα κατά την κλήση της μεθόδου παρατηρούμε το nickname, που είναι το όνομα που ορίσαμε κατά την κλήση της startAgent(), την κλάση AndroidStudent στην οποία ανήκει ο νέος πράκτορας (το κείμενο μπροστά από την κλάση αποτελεί το όνομα του πακέτου στο οποίο ανήκει) και ένα αντικείμενο AgentHandler που θα μας βοηθήσει στην διαχείριση του αποτελέσματος την δημιουργίας του. Τέλος, μετά την δημιουργία του πράκτορα, μέσα στο κώδικα της onSuccess(), προχωράμε στην ενεργοποίηση του μέσα από το αντικείμενο χειρισμού AgentHandler.

5.1.2 Αποστολή μηνύματος από AndroidStudent

Συνεχίζοντας της ανάλυση της εφαρμογής, έχοντας παράλληλα ως οδηγό το διάγραμμα ροής της εικόνας 29, παρατηρούμε ότι ο πράκτορας AndroidStudent αποστέλλει μήνυμα στον πράκτορα Teacher. Έχει επιλεγεί η διαδικασία αποστολής του μηνύματος αυτού να περιλαμβάνεται στην «συμπεριφορά» (Behaviour) του πράκτορα που είναι τύπου CyclicBehaviour (κυκλική συμπεριφορά) . Η «κυκλική συμπεριφορά» έχει την ιδιότητα να παραμένει σε κατάσταση αναμονής και να εκτελείτε εκ νέου κάθε φορά που συμβαίνει κάποιο γεγονός. Μια κοινή πρακτική για την ενεργοποίηση ενός CyclicBehaviour είναι να αποστείλουμε κάποιο μήνυμα στον πράκτορα που διαθέτει υλοποιημένη μια τέτοια.

```

protected void setup() {
    addBehaviour(new CyclicBehaviour(this)
    {
        public void action() {

            if(n==0) {
                ACLMessage tmsg = new ACLMessage(ACLMessage.INFORM);
                tmsg.setContent("Staring the process");
                tmsg.addReceiver(Teacher);
                send(tmsg);
                n++;
            }
        }
    }
}

```

Κώδικας 5 AndroidStudent.java

Στο κώδικα αυτό λοιπόν παρατηρούμε τον τρόπο αποστολής μηνύματος της Jade από πράκτορα σε πράκτορα. Αρχικά καλούμαστε να δημιουργήσουμε ένα αντικείμενο που θα ανήκει στην κλάση ACLMessage, το οποίο κατά την δημιουργία ορίζουμε και τον τύπο του μηνύματος που θα μεταφέρει. Οι τύποι μηνυμάτων έχουν περισσότερο σημασιολογική έννοια και ενδεικτικά αναφέρουμε ότι μπορεί να είναι INFORM, REQUEST, PROPOSE , AGREE, CONFIRM, CANCEL κ.α. Κατόπιν, προσομοιώνοντας την αποστολή μηνυμάτων όπως σε κάθε μορφή επικοινωνίας, πρέπει να ορίσουμε το περιεχόμενο του μηνύματος, με την μέθοδο setContent() και να προσθέσουμε τον παραλήπτη με την μέθοδο addReceiver(). Στην συνέχεια καλούμε την μέθοδο send() δίνοντάς σαν παράμετρο το αντικείμενο του μηνύματος μας ώστε να γίνει με κατάλληλο χειρισμό από την βιβλιοθήκη μας η αποστολή του μηνύματος.

Μετά την αποστολή του μηνύματος από τον πράκτορα AndroidStudent στον πράκτορα Teacher το ενδιαφέρον στρέφεται στην jade που εκτελείτε στον υπολογιστή μας. Πριν ξεκινήσουμε την ανάλυση για την λήψη αυτού του μηνύματος, θα πρέπει να παρουσιάσουμε την διαδικασία που ακολουθήσαμε ώστε να είναι ενεργός και σε κατάσταση αναμονής ο πράκτορας Teacher.

5.1.3 Εφαρμογή Jade QA Agents στον υπολογιστή.

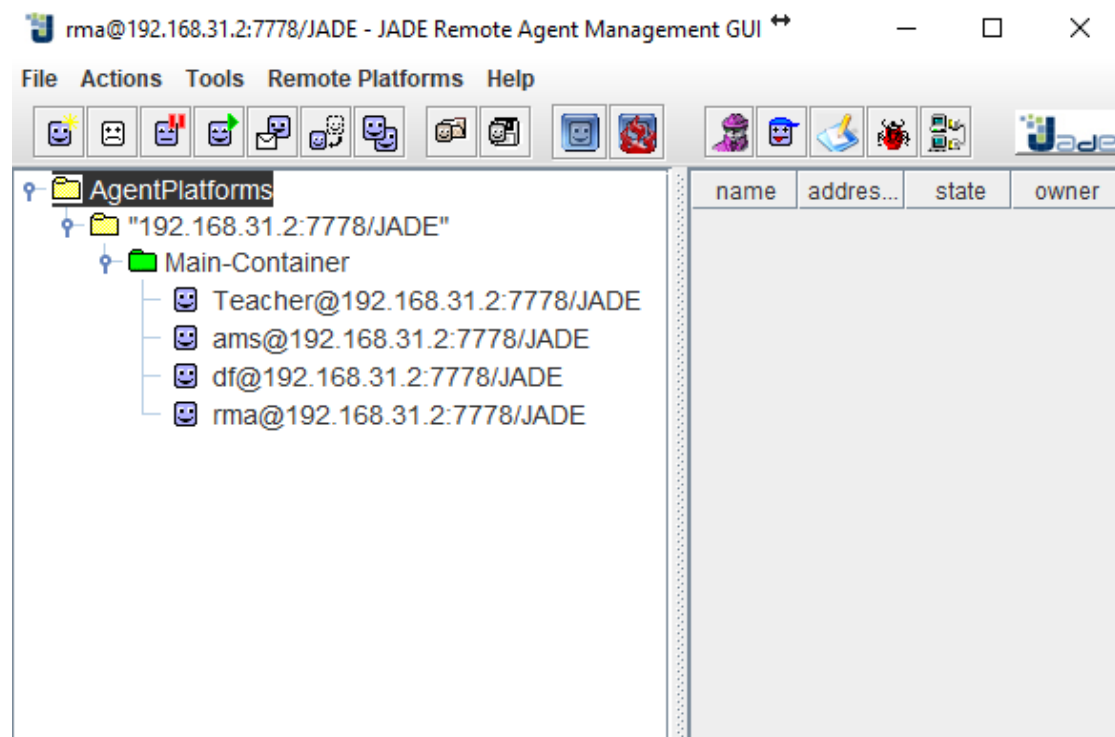
Ξεκινώντας την ανάλυση του project μας που εκτελείται στον υπολογιστή, θα πρέπει να αρχικοποιήσουμε και να ενεργοποιήσουμε τις κλάσεις μας. Την επιθυμητή αρχικοποίηση επιλέξαμε να την κάνουμε μέσα από την γραμμή εντολών των windows χωρίς αυτό να σημαίνει ότι το project δεν μπορεί να λειτουργήσει απaráλλακτο και σε περιβάλλον linux, εκτελώντας αντίστοιχα τις εντολές που χρειάζεται για το λειτουργικό σύστημα. Υπάρχει κατάλληλη αναφορά για αυτό στους οδηγούς χρήσης που είναι αναρτημένοι στο site της βιβλιοθήκης.[<http://jade.tilab.com/documentation/tutorials-guides>]

Στο παρακάτω κομμάτι κώδικα παρατηρούμε πως γίνεται η αρχικοποίηση. Θα πρέπει σε αυτή την εντολή να συμπεριλαμβανουμε το πλήθος των πρακτόρων που επιθυμούμε να δημιουργήσουμε, το όνομα που θα έχει ο κάθε πράκτορας και την κλάση στην οποία θα ανήκει. Παρατηρούμε ακόμα ότι ορίζουμε στην εντολή μας που βρίσκεται η βιβλιοθήκη μας (lib\jade.jar) καθώς και το όνομα του φακέλου στον οποίο θα αναζητήσει τις αναφερόμενες κλάσεις για την δημιουργία του πράκτορα.

```
java -cp lib\jade.jar;classes jade.Boot -gui -port 7778 -accept-foreign-  
agents true -jade_http_mtp_outPort 7778 -agents Teacher:Teacher
```

Κώδικας 6 Εκκίνηση jade

Με την εκτέλεση της παραπάνω εντολής ενεργοποιείτε η εφαρμογή μας και λαμβάνουμε το εξής γραφικό περιβάλλον.



Εικόνα 32 RAM GUI

Σε αυτό βλέπουμε με την βοήθεια του πράκτορα RMA τους εξορισμού δημιουργημένους ams, df, rma και τον Teacher που επιλέξαμε εμείς. Το container εξορισμού ονομάζεται Main-Container καθώς κατά την δημιουργία δεν ορίζουμε να προσκολληθεί σε άλλο container και έτσι λειτουργεί αυτόνομα.

Στο επόμενη βήμα μεταφερόμαστε στον κώδικα του πράκτορα Teacher. Σε αυτόν παρατηρούμε ότι επιλέξαμε η «συμπεριφορά» του πράκτορα να είναι τύπου CyclicBehaviour καθώς εξυπηρετεί την ασύγχρονη ενεργοποίηση της μετά από την αποστολή μηνύματος.

```

protected void setup() {
    addBehaviour(new CyclicBehaviour() {
        @Override
        public void action() {
            ACLMessage msg = receive();
            if (msg != null) {
                ...
            }
        }
    });
}

```

Κώδικας 7 Teacher.java

Επιπρόσθετα, στο προηγούμενο μέρος κώδικα μπορούμε να δούμε ότι γίνεται λήψη του μηνύματος που λαμβάνει ο πράκτορας, αναλύεται και αν διαπιστωθεί ότι ο αποστολέας ήταν ο πράκτορας με όνομα `AndroidStudent` τότε προχωράμε στην δημιουργία του πράκτορα `student`.

```

String name = "Student" ;
AID student = new AID( name, AID.ISLOCALNAME );

```

Κώδικας 8 Teacher.java

Πριν προβούμε στην δημιουργία του πράκτορα προηγείται το παραπάνω απαραίτητο μέρος κώδικα που καταγράφεται πριν την έναρξη της μεθόδου `setup()`. Σε αυτό δημιουργούμε ένα καινούργιο `AID` (Agent Identifier, αναγνωριστικό πράκτορα) δίνοντας σαν πρώτη παράμετρο το όνομα του πράκτορα που μελλοντικά θα δημιουργήσουμε και σαν δεύτερη παράμετρο προσδιορίζουμε ότι η ονοματολογία αυτή αφορά τοπική χρήση για τον `container` που αυτός δημιουργείται. Η ονοματολογία αυτή αποθηκεύεται από τον πράκτορα διαχείρισης συστήματος (AMS, Agent Management System) στο κατάλογο που διατηρεί εσωτερικά του συστήματος έχοντας έτσι γνώση όλων των πρακτόρων του συστήματος του. Η λειτουργία αυτή αφορά την πλατφόρμα που ανήκει ένας AMS συμπεριλαμβανομένων όλων των `container` που είναι ενεργά.

```

AgentContainer cur_container = getContainerController();
AgentController acontroller = cur_container.createNewAgent( name,
"qa_agents.behaviour.student_agent.Student",
new Object[] {question} );
acontroller.start();

```

Κώδικας 9 Teacher.java

Στην συνέχεια, έχοντας γνώση για το `Container` που βρίσκεται ο πράκτορας `Teacher` μέσω της μεθόδου `getContainerController()` ορίζουμε να δημιουργηθεί σε αυτό νέος πράκτορας. Υπεύθυνη μέθοδο για την δημιουργία είναι η `createNewAgent()` στην οποία δίνουμε με την μορφή παραμέτρων το όνομα του νέου πράκτορα, την κλάση στην οποία θα ανήκει ο

πράκτορας αυτός (η κλάση ανήκει στο πακέτο qa_agents.behaviour.student_agent) και ένα αντικείμενο που ονομάζεται question.

Το αντικείμενο question, ανήκει όπως θα δούμε στο επόμενο απόσπασμα κώδικα στην κλάση hasMap και περιέχει το ερωτηματολόγιο που έχει επιλέξει ο καθηγητής να προωθήσει στον μαθητή για να ελέγξει την γνώση και το επίπεδο που έχει καταφέρει να αναπτύξει.

```
Map <String, String> question = new HashMap<String, String>();  
question.put("q1", "What is the sum of 1 + 1 in the binary system");  
question.put("all", "1");
```

.....

```
question.put("a24", "Ruby");
```

Κώδικας 10 Teacher.java

Στο σημείο αυτό αξίζει να τονίζουμε πως ,όπως αναφέρθηκε και νωρίτερα, ο πράκτορας AMS είναι υπεύθυνος για δημιουργία και την διαχείριση των πρακτόρων στο/στα container στα οποία διαχειρίζεται. Πρακτικά αυτό σημαίνει ότι ο AMS λαμβάνει το αίτημα για δημιουργία του πράκτορα student και προχωράει στην δημιουργία του καταχωρώντας παράλληλα την παρουσία του στην λίστα που διατηρεί με όλους τους πράκτορες.

Όπως συμπεραίνουμε η αρχικοποίηση και η δημιουργία του πράκτορα student λαμβάνει χώρα στο εσωτερικό της μεθόδου setup() του Teacher καθώς εννοιολογικά υπεύθυνος για την δημιουργία πράκτορα με σκοπό την αποστολή ερωτηματολογίου είναι ο καθηγητής ενός μαθητή.

```
msg = new ACLMessage (ACLMessage.INFORM);  
msg.setContent(" ");  
msg.addReceiver(student);  
send(msg);
```

Κώδικας 11 Teacher.java

Τελευταία ενέργεια που λαμβάνει χώρα στην αλληλουχία διαδικασιών για την δημιουργία του πράκτορα, είναι η αποστολή μηνύματος σε αυτόν. Το περιεχόμενο του μηνύματος δεν είναι σημαντικό και επιλέγεται να είναι ο χαρακτήρας κενό.

5.1.4 Δημιουργία και μετανάστευση πράκτορα student

Συνεχίζοντας να ακολουθούμε την σειρά εκτέλεσης των ενεργειών, μεταφερόμαστε στην κλάση student.java που εκτελείται στο περιβάλλον του υπολογιστή. Αμέσως μετά την δημιουργία του πράκτορα student, που προκαλείτε από τον πράκτορα Teacher, σειρά παίρνουν οι πρώτες εντολές της κλάσης student.

```
protected void setup() {
    Object[] args = getArguments();
    if (args!=null){
        question = (HashMap <String, String>) args[0];
    }
    home = here();
}
```

Κώδικας 12 Student.java

Πριν εισχωρήσουμε στην «συμπεριφορά» της student ο παραπάνω κώδικας εκτελείται μέσα στην setup() και αποθηκεύει στο αντικείμενο question (είναι και αυτό τύπου hasMap) το ερωτηματολόγιο που έλαβε παραμετρικά ο πράκτορας κατά την δημιουργία του.

```
private static final long serialVersionUID = 435357658778L;
```

Κώδικας 13 Student.java

Στο παραπάνω σύντομο κομμάτι κώδικα παρατηρούμε την μεταβλητή serialVersionUID. Η χρήση αυτής της μεταβλητής απαιτείται από την jade καθώς ο πράκτορας της κλάσης αυτής θα μεταναστεύσει με την διαδικασία της σειριοποίησης (serialization) και ο σκοπός χρήσης της είναι για επιβεβαίωση σωστής λήψης της κλάσης κατά την αποσειριοποίηση (deserialization) από τον παραλήπτη.

Προχωρώντας στην εκτέλεση της κλάσης συναντάμε και εδώ την επιλογή για CyclicBehaviour() που μέχρι τώρα εξυπηρετεί τον τρόπο λειτουργίας που επιθυμούμε. Ακόμα θα συναντήσουμε την κλήση της μεθόδου receive() σε αντικείμενο ACLMessage ώστε να λάβουμε και να διαχειριστούμε στην συνέχεια το μήνυμα.

```
Location dest = null;
if (here().getName().equals("Main-Container") && msg.getSender().getName().contains("Teacher")) {
    // I'm at home --> move to the Main Container
    dest = new ContainerID("Container-1", null);
    doMove(dest);
}
```

Κώδικας 14 Student.java

Κατόπιν, παραπάνω παρατηρούμε ότι γίνεται ο έλεγχος αν ο πράκτορας την στιγμή της εκτέλεσης βρίσκεται στο Main-Container και αν το μήνυμα που έλαβε αποστέλλεται από τον πράκτορα Teacher. Αν και οι δύο συνθήκες είναι αληθείς τότε ο πράκτορας επιλέγει να μεταναστεύσει στο Container-1 που γνωρίζει ότι είναι ενεργός ο πράκτορας AndroidStudent ώστε να του παραδώσει το ερωτηματολόγιο. Καμία παραπάνω εντολή δεν απαιτείται για να μπορέσει ο πράκτορας να μεταναστεύσει επιτυχημένα στο Container-1 καθώς την διαχείριση αυτής την αναλαμβάνει εξολοκλήρου η βιβλιοθήκη Jade.

Η jade προσφέρει 2 δυνατότητες μετανάστευσης σε έναν πράκτορα. Την «αδύναμη μετανάστευση» (weak migration) και την «δυνατή μετανάστευση» (strong migration). Κατά την αδύναμη μετανάστευση μεταφέρεται ο πράκτορας και οι μέθοδοι του χωρίς όμως να έχει γνώση και τιμές της προηγούμενης του κατάστασης. Από την άλλη πλευρά η δυνατή μετανάστευση επιτρέπει τον πράκτορα να μεταφέρει και την κατάσταση στην οποία ήταν. Η μετανάστευση που θα λάβει χώρα σε αυτό το σημείο είναι η «δυνατή μετανάστευση» (strong migration). Στην περίπτωση μας αυτό μας εξυπηρετεί απόλυτα καθώς ο πράκτορας θα μεταφέρει μαζί του και το αντικείμενο question που περιέχει το ερωτηματολόγιο.

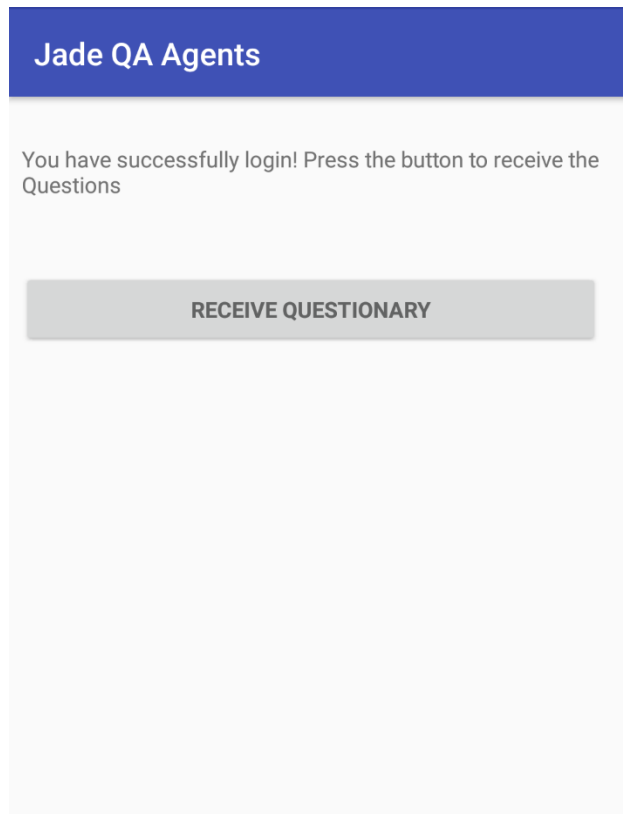
Διατηρώντας ακόμα την προσοχή μας στην κλάση student, μεταφερόμαστε στην μέθοδο afterMove(). Όπως μπορεί να συμπεράνει κάποιος από το όνομα της, ο κώδικας της εκτελείτε αμέσως μετά την μεταφορά του πράκτορα σε ένα νέο container.

```
@Override
public void afterMove()    {
    ....
    if (here().getName().equals("Container-1")) {
        ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
        msg.setContent(question.toString());
        msg.addReceiver(AndroidStudent);
        send(msg);
    }
}
```

Κώδικας 15 Student.java

Σε αυτό το μικρό απόσπασμά κώδικα λαμβάνει χώρα η ενέργεια που βλέπουμε να εκτελεί ο πράκτορας student αμέσως μετά την μετανάστευση του. Μετά τον έλεγχο αν το container που έφτασε είναι το Container-1, αποστέλλει μήνυμα στον πράκτορα AndroidStudent με τα δεδομένα του ερωτηματολογίου αφού τα μετατρέψει σε κείμενο με την toString().

Όση ώρα λαμβάνουν χώρα οι ενέργειες της δημιουργίας του πράκτορα Student, η μετανάστευσή του και η αποστολή του μηνύματος, ο χρήστης στην συσκευή android έχει μεταφερθεί από την εικόνα σύνδεσης στην παρακάτω.



Εικόνα 33 Επιτυχής σύνδεση

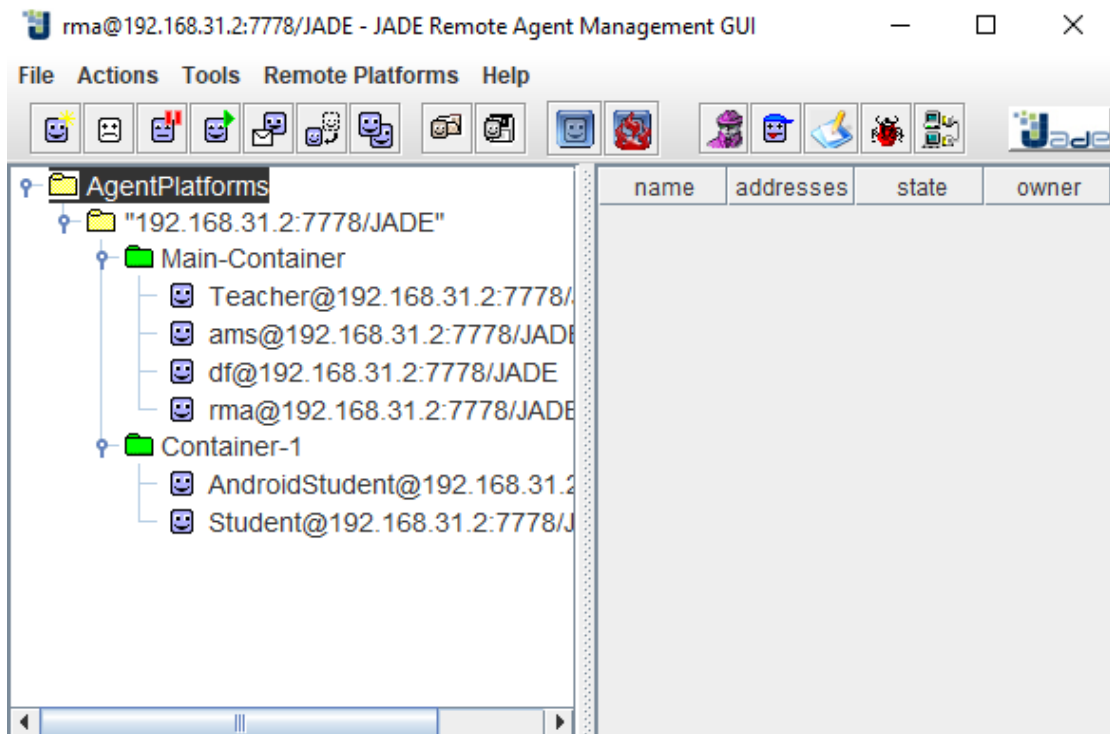
Παρατηρούμε ότι παρέχεται ενημέρωση πως συνδέθηκε επιτυχώς και αναμένει εντολή από τον ίδιο για να λάβει το ερωτηματολόγιο. Μια τελευταία ενέργεια που λαμβάνει χώρα πριν η εφαρμογή να είναι έτοιμη να παρουσιάσει το ερωτηματολόγιο είναι η λήψη του μηνύματος που αποστέλλεται από τον Teacher στον AndroidStudent.

```
ACLMessage msg = receive();
if (msg != null) {
    q = msg.getContent();
    student = msg.getSender();
    String[] vals = q.substring( 1, q.length()-1 ).split( "," );
    question = new HashMap<String, String>();
    for (int i=0; i<vals.length; i++){
        String[] str = vals[i].trim().split( "=" );
        question.put(str[0], str[1]);
    }
}
```

Κώδικας 16 AndroidStudent.java

Σε μια σύντομη περιγραφή του παραπάνω αποσπάσματος μπορούμε να αναφέρουμε ότι γίνεται η λήψη του μηνύματος από τον πράκτορα AndroidStudent, αποθηκεύεται ο αποστολέας για μελλοντική χρήση και μέσα από κατάλληλη επεξεργασία μεταφέρεται το ληφθέν μήνυμα σε ένα νέο αντικείμενο HashMap.

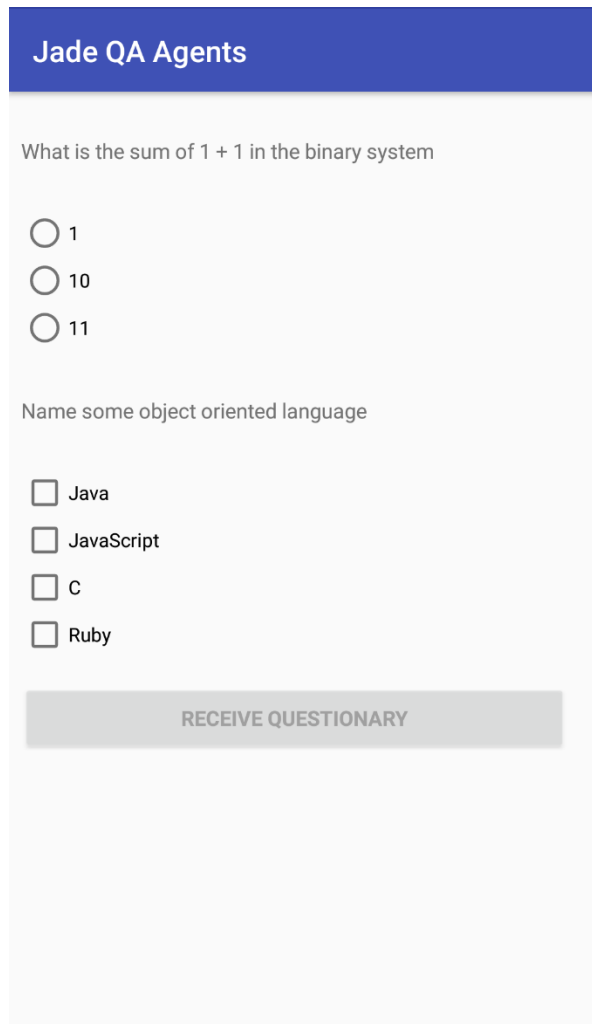
Την συγκεκριμένη χρονική στιγμή ενημερωνόμαστε από την παρακάτω εικόνα στον υπολογιστή για την τοποθεσία των πρακτόρων.



Εικόνα 34 RAM

5.1.5 Λήψη και εμφάνιση ερωτηματολογίου

Ο χρήστης λοιπόν, έχοντας πάρει την πληροφορία από την γραφική απεικόνιση της εφαρμογής ότι είναι έτοιμος για λήψη του ερωτηματολογίου, το επιλέγει πατώντας το αντίστοιχο πλήκτρο. Το activity που είναι ενεργό στην γραφική απεικόνιση είναι το QuestionerActivity που αναφέρθηκενωρίτερα. Κατά την δημιουργία του νέου activity η εφαρμογή μεταφέρεται στο περιβάλλον που βλέπουμε στην εικόνα 31 έχοντας παράλληλα δημιουργήσει έξτρα αντικείμενα που παραμένουν αφανή μέχρι ο χρήσης να επιλέξει για την λήψη του ερωτηματολογίου. Τα αντικείμενα αυτά είναι τύπου TextView, RadioGroup, RadioButton, CheckBox και Button. Ο τρόπος δημιουργία τους και η παρουσίαση τους είναι πέρα από τους σκοπούς της παρούσας διπλωματικής εργασίας.



Εικόνα 35 Jade QA Agents

Η παραπάνω εικόνα είναι αυτή που παρουσιάζεται στον χρήστη αφού επιλέξει το πλήκτρο Receive Questionary καθώς γίνονται διαθέσιμες οι ερωτήσεις που μεταφέρθηκαν από τον πράκτορά Teacher όπως και οι απαντήσεις που διατίθενται.

```
public void msg_receive() {  
    Map<String, String> q = question;  
    question1.setText( q.get("q1") );  
    question2.setText( q.get("q2") );  
    .  
    .  
    .  
}
```

Κώδικας 17 QuestionerActivity.java

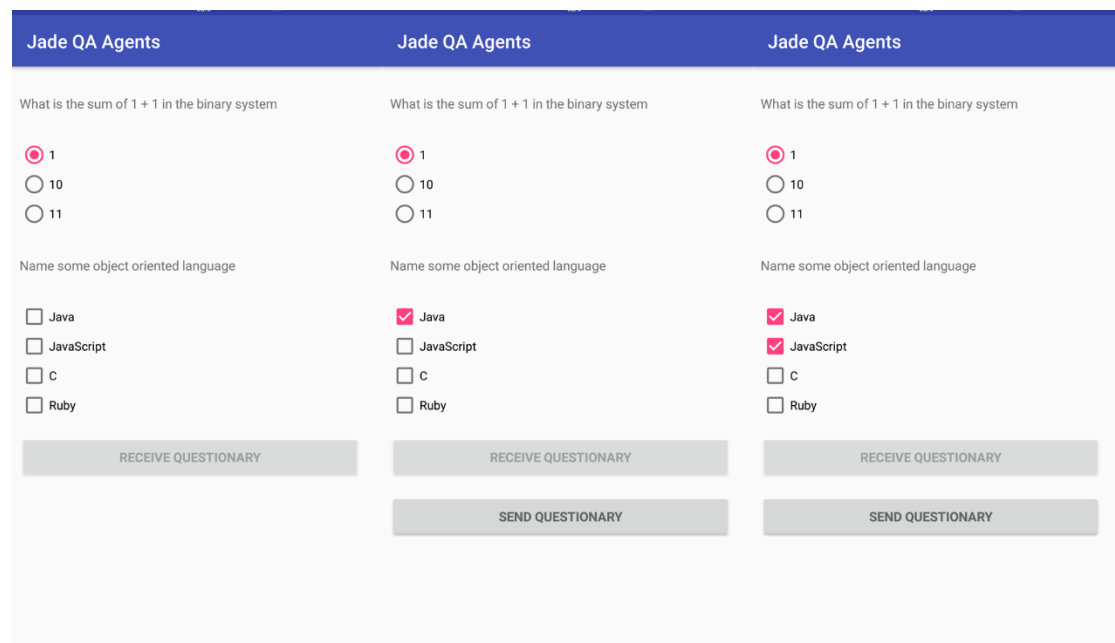
Για να αποκτήσει πρόσβαση το ενεργό Activity στα δεδομένα των ερωτήσεων πρέπει αρχικά να γίνει import η κλάση AndroidStudent που τα έχει στην κατοχή της και στην συνέχεια κάνοντας χρήση του αντικειμένου question που ανήκει στην AndroidStudent να περαστούν τα δεδομένα στα αντίστοιχα στοιχεία της xml ώστε να εμφανιστούν στον χρήστη.

Έχοντας ολοκληρωθεί τα παραπάνω, ο χρήσης καλείτε να επιλέξετε τις απαντήσεις και να προχωρήσει στην αποστολή τους. Στο στιγμιότυπο αυτό το πλήκτρο Send Questionary δεν είναι εμφανές καθώς δεν έχει επιλέξει ακόμα απαντήσεις στα ερωτήματα που εμφανίζονται. Έχουν προστεθεί κατάλληλοι έλεγχοι ώστε ο χρήστης να μπορεί να επιλέξει το αντίστοιχο κουμπί όταν έχει επιλέξει απαντήσεις σε όλες τις ερωτήσεις που διατίθεται. Ένα απόσπασμά από τον κώδικα των ελέγχων φαίνεται παρακάτω.

```
// Handle the Visibility of Send Button
ch1.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView,boolean isChecked) {
        if(r1.isChecked() || r2.isChecked() || r3.isChecked() ) {
            sendButton.setVisibility(View.VISIBLE);
        }
        if( !ch1.isChecked() && !ch2.isChecked() && !ch3.isChecked() &&
            !ch4.isChecked()) {
            sendButton.setVisibility(View.GONE);
        }
    }
});
```

Κώδικας 18 QuestionerActivity.java

5.1.6 Εμφάνιση και απάντηση ερωτήσεων



Εικόνα 36 Jade QA Agents Ερωτήσεις

Ακολουθώντας πάντα το διάγραμμα ροής που παρουσιάστηκε στην αρχή του κεφαλαίου, όταν ο χρήστης επιλέξει το κουμπί Send Questionary καλείτε η μέθοδος onClick() που αντιστοιχεί σε αυτό το κουμπί. Στο χρήστη παρουσιάζεται ένα σύντομο μήνυμα στο κέντρο της οθόνης που τον ενημερώνει για επιτυχή αποστολή των απαντήσεων του. Σε περίπτωση που το επιλέξει δεύτερη φορά εμφανίζεται κατάλληλο μήνυμα. Στο android αυτοί οι τύποι

μηνυμάτων ονομάζονται «Toast». Επιπρόσθετά στην μέθοδο αυτή γίνεται λήψη των απαντήσεων που δόθηκαν από τον κώδικα που ακολουθεί και καταχωρούνται στο αντικείμενο answer που είναι τύπου HashMap και ανήκει στην κλάση AndroidStudent.

```
public void onClick(View view) {
    int idx = radioButtonGroup.getCheckedRadioButtonId();
    RadioButton selected_radio = (RadioButton) findViewById(idx);
    AndroidStudent.answer = new HashMap<String, String>();
    //Put Values to Map answer
    AndroidStudent.answer.put("q1", AndroidStudent.question.get("q1"));
    AndroidStudent.answer.put("a1", selected_radio.getText().toString());
    AndroidStudent.answer.put("q2", AndroidStudent.question.get("q1"));
    if (ch1.isChecked()) AndroidStudent.answer.put("a21", AndroidStudent.question.get("a21"));
    if (ch2.isChecked()) AndroidStudent.answer.put("a22", AndroidStudent.question.get("a22"));
    if (ch3.isChecked()) AndroidStudent.answer.put("a23", AndroidStudent.question.get("a23"));
    if (ch4.isChecked()) AndroidStudent.answer.put("a24", AndroidStudent.question.get("a24")); }
    if (send==0) {
        Toast.makeText(QuestionerActivity.this, "You have Successfully send your answers to you Teacher", Toast.LENGTH_SHORT).show();
        send++;
    }
    else {
        Toast.makeText(QuestionerActivity.this, "You have already send your answers!", Toast.LENGTH_SHORT).show();
    }
}
```

Κώδικας 19 QuestionerActivity.java

Επόμενη ενέργεια στο διάγραμμα μας είναι η αποστολή των απαντήσεων από τον AndroidStudent στον Student με την μορφή μηνύματος. Για την επίτευξη αυτού του στόχου επιλέχθηκε η δημιουργία μιας νέας «συμπεριφοράς» στον AndroidStudent που είναι τύπου TickerBehaviour και εκτελείτε κάνε 500ms. Η «συμπεριφορά» αυτή περιλαμβάνει ένα έλεγχο που αφορά την επιτυχή καταχώρηση των απαντήσεων στο αντικείμενο answer και αν έχει αποσταλεί ήδη η απάντηση. Αν και οι δύο συνθήκες ελέγχου είναι αληθείς τότε δημιουργείτε ένα μήνυμα που αποστέλλει με την μορφή string την απάντηση στο πράκτορα student.

5.1.7 Επιστροφή πράκτορα student

```
addBehaviour(new TickerBehaviour( this, 500 ) {
    protected void onTick() {
        if (answer!=null && !isAnswerSend ){
            ACLMessage emsg = new ACLMessage(ACLMessage.INFORM);
            emsg.setContent(answer.toString());
            emsg.addReceiver(student);
            send(emsg);
            isAnswerSend = true;
        }
    }
});
```

Κώδικας 20 AndroidStudent.java

Ο πράκτορας student, καθώς όπως παρουσιάστηκε παραπάνω διαθέτει CyclicBehaviour, λαμβάνοντας το μήνυμα ενεργοποιεί εκ νέου την Behaviour. Στην action() μέθοδο αφού λάβει το μήνυμα, προχωράει σε έλεγχο του αποστολέα. Όταν επιβεβαιώνει πως ο αποστολέας είναι ο androidStudent εκτελεί διαδικασίες λήψης του μηνύματος, επεξεργασίας-καταχώρησης του σε ένα νέο αντικείμενο HashMap και τελικώς μεταναστεύει εκ νέου στο Main-Container από όπου αρχικά είχε προέλθει.

```
else if ( msg.getSender().getName().contains("AndroidStudent") ) {
    // I'm elsewhere --> go home
    dest = home;
    String ans = msg.getContent().toString();
    String[] vals = ans.substring( 1,ans.length()-1 ).split( "," );
    answer = new HashMap<String, String>();

    for (int i=0; i<vals.length; i++){
        String[] str = vals[i].trim().split( "=" );
        answer.put(str[0], str[1]);
    }
    doMove( dest );
}
```

Κώδικας 21 student.java

Εν κατακλείδι, ο πράκτορας student, μέσω της μεθόδου afterMove() διαθέτει κατάλληλο κώδικα που ανιχνεύει αν το Container στο οποίο μεταφέρθηκε είναι το Main-Container. Σε περίπτωση που ο έλεγχος αυτός είναι αληθής, τότε προχωράει στην αποστολή του μηνύματος που έχει στην κατοχή από τον AndroidStudent με παραλήπτη τον πράκτορα Teacher.

```
if (here().getName().equals("Main-Container")) {
    ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
    msg.setContent( answer.toString() );
    msg.addReceiver( Teacher );
    send( msg );
}
```

Κώδικας 22 student.java

Μετά την εκτέλεση του παραπάνω κώδικα, λαμβάνει σε μορφή μηνύματος ο πράκτορας Teacher τα αποτελέσματα για μελλοντική επεξεργασία και ανατροφοδότηση. Στην εφαρμογή μας δεν υλοποιείτε κάποια ενέργεια πάνω στα ληφθέντα δεδομένα.

Ο πλήρης κώδικας της εφαρμογής είναι προσβάσιμος από τον παρακάτω σύνδεσμο:

https://github.com/stavrinho/Jade_QA_Agents

6

Επίλογος

6.1 Σύνοψη και συμπεράσματα

Μέσα από την χρήση της βιβλιοθήκης Jade και του πρόσθετου JADE-LEAP παρέχεται ένα ισχυρό περιβάλλον δημιουργίας & διαχείρισης ευφών πρακτόρων που μπορούν με ευκολία να προστεθούν και να προσφέρουν σε ένα οποιοδήποτε σύστημα java. Το project που δημιουργήσαμε στην παρούσα εργασία παρέχει σε ένα εκπαιδευτικό σύστημα την δυνατότητα εκμετάλλευσης των κινητών συσκευών για μάθηση ακόμα σε ένα περιβάλλον χαμηλών ταχυτήτων διασύνδεσης και μεγάλων απαιτήσεων για χαμηλή κατανάλωση ενέργειας. Το εύρος της κινητικότητας του χρήστη ορίζεται σε επίπεδο πανεπιστημιούπολης. Το σύστημα κινητής μάθησης που δημιουργήθηκε μπορούμε να πούμε ότι καλύπτει το σύνολο σχεδόν των απαιτήσεων που ορίσαμε στο κεφάλαιο 3 πως πρέπει να έχει ένα Mobile learning. Συναντάμε τα χαρακτηριστικά της φορητότητας, της περιορισμένης κατανάλωσης πόρων, της γραφικής απεικόνισης, της ευελιξίας στην σχεδίαση και παράλληλά της ευκολίας στη επέκταση, καθώς και διαδικασίες αυτοματισμών. Λόγο της φύσης του συστήματος, δεν χρειάστηκε να υλοποιήσουμε κάποια προσωρινή μνήμη δεδομένων η οποία σε μια πιθανή αναβάθμιση του κρίνεται απαραίτητη.

Αξίζει να σημειώσουμε ότι η βιβλιοθήκη Jade παρόλα τα χρόνια ανάπτυξης που έχει συμπληρώσει, δεν έχει καταφέρει ακόμα να αποκτήσει κάποια ισχυρή κοινότητα υποστήριξης και οι πληροφορίες που παρέχονται επιπρόσθετά των οδηγιών και του επίσημου βιβλίου είναι

περιορισμένες. Τέλος συναντήσαμε περιορισμούς στην δυνατότητα υποστήριξης μετανάστευσης ενός πράκτορα σε κινητή συσκευή με χρήση δικτύων κινητής τηλεφωνίας καθώς όπως διαπιστώσαμε αυτή η λειτουργία δεν υποστηρίζεται προς το παρόν.

6.2 Μελλοντικές επεκτάσεις

Για να γίνει πιο έντονο το στοιχείο της ευφυΐας στο σύστημα που δημιουργήσαμε θα ήταν επιθυμητό να δημιουργηθούν κατάλληλες μέθοδοι που επιτρέπουν την δημιουργία «συζητήσεων» των πρακτόρων της κινητής συσκευής με τους πράκτορες των σταθερών συστημάτων ώστε η μετανάστευση να υλοποιείται σε χρόνο που η συσκευή του χρήστη είναι σε αναμονή. Ακόμα, η προσθήκη επικοινωνίας μεταξύ κινητών συσκευών θα μπορούσε να προσφέρει αποσυμφόρηση στην μονόδρομη επικοινωνία κινητών και σταθερών συστημάτων παρέχοντας έτσι μεγαλύτερες ταχύτητες σε συσκευές που είναι απομακρυσμένες από κεντρικά σημεία διασύνδεσης.

7

Βιβλιογραφία

- [1] Ελένη Θερμογιάννη, “Εφαρμογή τεχνικών Υπολογιστικής Νοημοσύνης για υποστήριξη Συστημάτων Ηλεκτρονικής Μάθησης βασισμένη σε αρχιτεκτονική ευφυών πρακτόρων,” *ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής*, 2006.
- [2] Μ. Βαμιαδάκη and Μ. Vamiadaki, “Η ανάπτυξη και χρήση των νέων μέσων ηλεκτρονικής μάθησης (e-learning) στην εκπαίδευση και η επίδραση της σε εκπαιδευτές και εκπαιδευόμενους,” 2016.
- [3] Λ. Ε. Ζουμπουλάκη Άννα, “Η ΑΣΥΓΧΡΟΝΗ ΤΗΛΕΚΠΑΙΔΕΥΣΗ ΚΑΙ Η ΕΦΑΡΜΟΓΗ ΤΗΣ ΣΤΗΝ ΕΞ ’ ΑΠΟΣΤΑΣΕΩΣ ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΣΤΗ ΔΙΑ ΒΙΟΥ,” *ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΙΟΝΙΩΝ ΝΗΣΩΝ ΤΜΗΜΑ ΕΦΑΡΜΟΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΗ ΔΙΟΙΚΗΣΗ ΚΑΙ ΤΗΝ ΟΙΚΟΝΟΜΙΑ*, 2011.
- [4] Π. Χουντής, “Χρήση ευφυών πρακτόρων σε εκπαιδευτικό σύστημα βασισμένο στον παγκόσμιο ιστό,” 2004.
- [5] Κ. Μ. Μελισίδης, “Κινητοί Πράκτορες με JADE.” 2014.
- [6] Κοκκαλογιάννη Σοφία, “Ηλεκτρονικό εμπόριο και συστήματα πολλαπλών πρακτόρων - Ιδρυματικό Αποθετήριο,” *Tech. Univ. Crete*, 2007.
- [7] C. N. Udanor and T. A. Nwodoh, “A REVIEW OF M-LEARNING MODELS,” vol. 1, no. 4, pp. 426–435, 2010.
- [8] Y. S. Chen, T. C. Kao, and J. P. Sheu, “A mobile learning system for scaffolding bird

- watching learning,” *J. Comput. Assist. Learn.*, vol. 19, no. 3, pp. 347–359, 2003.
- [9] R. Y. L. Ting, “Mobile learning: Current trend and future challenges,” in *Proceedings - 5th IEEE International Conference on Advanced Learning Technologies, ICALT 2005*, 2005, vol. 2005, pp. 603–607.
- [10] P. Silander and a. Rytkonen, “An intelligent mobile tutoring tool enabling individualization of students’ learning processes.,” *4th World Conf. mLearning*, pp. 25–28, 2005.
- [11] S. A. Kazi, “VocaTest: An Intelligent Tutoring System for Vocabulary Learning using the ‘mLearning’ Approach.”
- [12] A. A. Kassim, S. A. Kazi, and S. Ranganath, “A Web-based Intelligent Learning Environment for Digital Systems,” *Scopus*, 2004.
- [13] K. T. Lin, “Improving mobile learning environments by applying mobile agents technology,” *IEICE Trans. Commun.*, vol. 84, pp. 8–14, 2004.
- [14] M. Ally, F. Lin, R. McGreal, and B. Woo, “An Intelligent Agent for Adapting and Delivering Electronic Course Materials to Mobile Learners,” *Proc. MLearn 2005 - Mob. Technol. Futur. Learn. your hands*, 2005.
- [15] L. Henry and S. Sankaranarayanan, “Mobile Learning- Brief Review,” . *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, vol. 2, no. 2010, pp. 306–319, 2010.
- [16] N. Casalino, A. D’Atri, A. Garro, P. Rullo, D. Sacca, and D. Ursino, “An XML-based multi-agent system to support an adaptive cultural heritage learning,” in *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL’06)*, 2006, pp. 224–224.
- [17] I. Ganchev, S. Stojanov, M. O’Droma, and D. Meere, “An InfoStation-Based University Campus System for the Provision of mLearning Services,” in *Sixth IEEE International Conference on Advanced Learning Technologies (ICALT’06)*, 2006, pp. 195–199.
- [18] D. Meere, I. Ganchev, M. O’Dr??ma, S. Stojanov, and V. Valkanova, “Adaptation for assimilation: Shaping context-sensitive m-learning services within a multi-agent environment,” *6th Adv. Int. Conf. Telecommun. AICT 2010*, pp. 74–79, 2010.

