



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ**

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εκτέλεση Ροών Συνδυασμένων Υπηρεσιών σε BPEL4WS

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΠΑΝΑΓΙΩΤΗ Δ. ΖΩΓΡΑΦΟΥ

Επιβλέπων : ΤΙΜΟΛΕΩΝ ΣΕΛΛΗΣ
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2003



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ

Εκτέλεση Ροών Συνδυασμένων Υπηρεσιών σε BPEL4WS

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΠΑΝΑΓΙΩΤΗ Δ. ΖΩΓΡΑΦΟΥ

Επιβλέπων : ΤΙΜΟΛΕΩΝ ΣΕΛΛΗΣ
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 14^η Ιουλίου 2003.

(Υπογραφή)

.....
Τιμολέον Σελλής
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Παναγιώτης Τσανάκας
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Νεκτάριος Κοζύρης
Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2003

Πρόλογος

Η παρούσα διπλωματική εργασία ασχολείται με ένα από τα πιο ενδιαφέροντα πεδία της σύγχρονης επιχειρησιακής έρευνας. Μελετά την ανάπτυξη εφαρμογών κυρίως ηλεκτρονικού εμπορίου και ηλεκτρονικής διακυβέρνησης που περιλαμβάνουν έλεγχο ροών συνδυασμένων δικτυακών υπηρεσιών.

Η διπλωματική εργασία εκπονήθηκε στο εργαστήριο Βάσεων και Γνώσεων και Δεδομένων του Εθνικού Μετσόβιου Πολυτεχνείου.

Η εργασία έγινε υπό την επίβλεψη του Καθηγητή Ε.Μ.Π. κ. Τιμο Σελλή τον οποίο θα ήθελα να ευχαριστήσω προσωπικά για την μεγάλη βοήθεια που μου προσέφερε καθόλη την διάρκεια της προσπάθειάς μου με πολύτιμες συμβουλές αλλά και παρέχοντάς μου ιδανικές συνθήκες εργασίας και μελέτης.

Καθοριστική για την περάτωση του συγκεκριμένου έργου ήταν η συμβολή του Δρ. Ντίνου Αρκουμάνη ο οποίος καθοδήγησε την όλη προσπάθεια. Σε αυτόν οφείλεται τόσο η επιλογή του θέματος όσο και η υπέρβαση πολλών από τις δυσκολίες που συνάντησα. Θα ήθελα να τον ευχαριστήσω για την πολύτιμη βοήθειά του τόσο σε επιστημονικό επίπεδο όσο και στο επίπεδο της προσωπικής στήριξης και διαρκούς ενθάρρυνσης.

Επίσης σημαντική ήταν και η βοήθεια που δέχθηκα από όλα τα μέλη του εργαστηρίου Βάσεων και Γνώσεων και Δεδομένων, τόσο των μεταπτυχιακών όσο και των προπτυχιακών φοιτητών ..

Τέλος θεωρώ ιδιαίτερα πολύτιμη την στήριξη από την οικογένειά μου στο δύσκολο αυτό εγχείρημα. Θα ήθελα να ευχαριστήσω τον πατέρα μου Δημήτρη, την μητέρα μου Σοφία και τον αδερφό μου Λευτέρη.

Αθήνα, 18 Ιουλίου 2003,
Παναγιώτης Δ. Ζωγράφος

Περίληψη στα ελληνικά

Στόχος της συγκεκριμένης διπλωματικής είναι η μελέτη της τεχνολογίας των δικτυακών υπηρεσιών(web services) καθώς και των τεχνολογιών που χρησιμοποιούνται ή βρίσκονται υπό ανάπτυξη προκειμένου να ενταχθούν οι δικτυακές υπηρεσίες σε αυτόνομες ροές εκτέλεσης ή επιχειρησιακές διαδικασίες(business processes). Η υλοποίηση αφορά την δημιουργία ενός πρότυπου συστήματος ηλεκτρονικής διακυβέρνησης που θα χρησιμοποιεί τις ροές αυτές ως ανεξάρτητες διαδικασίες, προσβάσιμες από τους χρήστες του. Στόχοι έρευνας αποτέλεσαν οι υπάρχουσες πειραματικές γλώσσες που βρίσκονται υπό εξέλιξη και σκοπό έχουν την περιγραφή επιχειρησιακών διαδικασιών. Από αυτές επιλέχθηκε η γλώσσα BPEL4WS η οποία και χρησιμοποιήθηκε προκειμένου να περιγράψει και να υλοποιήσει τις διαδικασίες του συστήματός μας. Το καθαυτό σύστημα αποτελούσε μια εναλλακτική πρόταση υλοποίησης του ΚΕΠ(Κέντρο Εξυπηρέτησης Πολιτών) έτσι ώστε να μπορεί να προσφέρει τις υπηρεσίες του πάνω από το διαδίκτυο. Οι επιμέρους δημόσιες υπηρεσίες προσομοιώνονται με την μορφή δικτυακών υπηρεσιών και συνεπώς οι υπηρεσίες του ΚΕΠ δεν είναι παρά επιχειρησιακές διαδικασίες που τις χρησιμοποιούν.

Λέξεις Κλειδιά:

Δικτυακές Υπηρεσίες
Επιχειρησιακές Διαδικασίες
Ροές Εκτέλεσης
ΚΕΠ

Περίληψη στα αγγλικά

The goal of the diploma thesis is to study the web services technology as well as the technologies used or currently being developed in order to describe business processes that use web services. The system developed consists of an e-government application that uses business processes, creating independent procedures for its users. The experimental languages, designed to describe business processes, already under development have been the major subjects of the conducted research. Finally BPEL4WS was the language selected to describe and implement the business processes mentioned above. The system itself is an alternate way to design an already functional public service known as KEP (Citizen Support Center). The goal is to add to the system the potential of publishing the existing services over the internet. The public services used by KEP are emulated as web services and therefore the services of KEP (Citizen Support Center) are business processes that use them.

Keywords:

Web services

Business processes

BPEL4WS

Execution flows

Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1	Αντικείμενο της διπλωματικής.....	1
1.2	Οργάνωση του τόμου	2
2	Περιγραφή Θέματος.....	5
2.1	Βασικές Έννοιες.....	5
2.1.1	Επιχειρησιακές Διαδικασίες (Business Processes).....	5
2.1.2	Επιχειρησιακές Διαδικασίες και Δικτυακές Υπηρεσίες	6
2.1.3	Εισαγωγή στις δικτυακές υπηρεσίες.....	7
2.1.4	Δικτυακές Υπηρεσίες και Επιχειρησιακές Διαδικασίες.....	10
2.1.5	BPEL4WS.....	12
2.1.6	BPML.....	14
2.1.7	Σύγκριση των δύο γλωσσών	14
2.1.8	Συμπεράσματα - Επιλογή Συστήματος υλοποίησης.....	15
2.2	Κέντρα Εξυπηρέτησης Πολιτών.....	15
2.3	Στόχος.....	17
2.3.1	Γενικός προσδιορισμός του στόχου.....	17
2.3.2	Υλοποίηση των επιμέρους υπηρεσιών.....	17
2.3.3	Στοιχεία αρχιτεκτονικής	18
3	Αρχικές προσπάθειες Υλοποίησης.....	19

3.1	Εισαγωγικά.....	19
3.2	Προσπάθεια 1 ^η	19
3.3	Προσπάθεια 2 ^η	21

4

Ανάλυση και σχεδίαση.....	25
---------------------------	----

4.1	Περιγραφή Αρχιτεκτονικής.....	25
4.1.1	Αρχιτεκτονική Επιχειρησιακών Διαδικασιών	26
4.1.2	Αρχιτεκτονική κεντρικής εφαρμογής	27
4.2	Περιγραφή Λειτουργιών.....	28
4.3	Λειτουργίες Απλού Χρήστη.....	32
4.4	Λειτουργίες Διαχειριστή	35
4.4.1	Επίσκεψη της κεντρικής σελίδας του απλού χρήστη/ διαγραφή διαδικασίας.....	36
4.4.2	Προσθήκη Διαδικασίας.....	37
4.4.3	Προσθήκη Διαχειριστή	42
4.4.4	Διαγραφή Χρήστη.....	43
4.4.5	Logout.....	45
4.5	Υπερδιαχειριστές.....	45

5

Υλοποίηση.....	47
----------------	----

5.1	Πλατφόρμες και προγραμματιστικά εργαλεία.....	47
5.1.1	Επιλογή γλώσσας.....	48
5.1.2	Λειτουργικό σύστημα	49
5.1.3	Επιλογή τρόπου δημιουργίας δυναμικών ιστοσελίδων	49
5.1.4	Επιλογή τρόπου επικοινωνίας με τη Βάση Δεδομένων	50
5.1.5	Επιλογή εργαλείου ανάπτυξης κώδικα	50
5.1.6	Επιλογή Web Server	50
5.1.7	BPEL4WS engine	51
5.1.8	BPEL4WS Validator.....	51

5.1.9	BPEL4WS Editor Eclipse Plugin.....	51
5.2	Λεπτομέρειες υλοποίησης	51
5.2.1	Classes.....	52
5.2.2	Beans.....	55

6

Έλεγχος.....	61
--------------	----

6.1	Μεθοδολογία Ελέγχου.....	61
6.2	Αναλυτική παρουσίαση έλεγχου	61
6.2.1	Έλεγχος διαχωρισιμότητας χρηστών	61
6.2.2	Έλεγχος λειτουργιών	61
6.3	Συμπέρασμα	73

7

Επίλογος.....	75
---------------	----

7.1	Σύνοψη και συμπεράσματα	75
7.2	Μελλοντικές επεκτάσεις.....	75

ΠΑΡΑΡΤΗΜΑ: ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ.....	77
--------------------------------------	----

8

Βιβλιογραφία.....	79
-------------------	----

1

Εισαγωγή

Στην εποχή μας ο χώρος του ηλεκτρονικού εμπορίου και της ηλεκτρονικής διακυβέρνησης αποτελεί ένα τομέα που εξαπλώνεται συνεχώς και αποτελεί πεδίο εντατικότητας και συνεχούς έρευνας. Όλο και περισσότερες είναι οι νέες τεχνολογίες που αναπτύσσονται προκειμένου να βελτιώσουν τις ήδη υπάρχουσες εφαρμογές και να ορίσουν νέες προοπτικές, βελτιώνοντας τις υπηρεσίες που παρέχονται.

1.1 Αντικείμενο της διπλωματικής

Το αντικείμενο της παρούσας διπλωματικής έχει να κάνει με τις εξελίξεις που αναφέραμε παραπάνω αφού μελετάει μια από τις καινούργιες τεχνολογίες που αναπτύσσονται αυτή τη στιγμή.

Οι εφαρμογές που αναπτύσσονται στο χώρο του ηλεκτρονικού εμπορίου και της ηλεκτρονικής διακυβέρνησης καθημερινά αυξάνονται. Οι κύριοι λόγοι δεν είναι άλλοι από τις ευκολίες που παρέχει το διαδίκτυο στους χρήστες του και επομένως και στους χρήστες των παραπάνω δικτυακών εφαρμογών. Πιο συγκεκριμένα οι εφαρμογές αυτές:

- Είναι εύκολα προσβάσιμες
- Δίνουν στο χρήστη απεριόριστα περιθώρια χρόνου ώστε να τις χρησιμοποιήσει όπως αυτός επιθυμεί καλύτερα διευκολύνοντας την εξυπηρέτηση
- Η πρόσβαση σε αυτές γίνεται με αμελητέο κόστος
- Μπορούν να λειτουργούν αποδεδειγμένες από ωράριο
- Το κόστος συντήρησης είναι χαμηλό κ.α.

Η εξάπλωση όμως των ως άνω περιγραφόμενων υπηρεσιών, δημιούργησε μερικά προβλήματα κυρίως λόγω της αύξησης του μεγέθους των και της πολυπλοκότητάς τους. Έτσι δημιουργήθηκε η ανάγκη στο να περιοριστεί τόσο η πολυπλοκότητα όσο και το μέγεθος των εφαρμογών. Οι εξελίξεις όπως αυτές περιγράφονται περιληπτικά παρακάτω, οδήγησαν σε διάφορες νέες τεχνολογίες αλλά και στην προσαρμογή ήδη υπάρχοντων τεχνολογιών στα νέα δεδομένα.

Προφανής τρόπος αντιμετώπισης της πολυπλοκότητας και του μεγέθους ήταν ο καταμερισμός εργασίας. Έτσι δημιουργήθηκαν νέες αφαιρετικές δομές που έκαναν δυνατή τη χρήση εφαρμογών και των λειτουργιών τους από πολλούς χρήστες ταυτόχρονα, οι οποίοι μπορούν μάλιστα να τις χρησιμοποιούν από περιβάλλοντα ανεξάρτητα από αυτές. Με τον τρόπο αυτό επεκτάθηκε το ήδη υπάρχον μοντέλο χρήστη-εξυπηρετητή, αποκτώντας νέες προοπτικές και δυνατότητες. Η κυριότερη από τις παραπάνω τεχνολογίες είναι η τεχνολογία

των δικτυακών υπηρεσιών(web services). Η τεχνολογία αυτή θα περιγραφεί παρακάτω αφού αποτελεί κομβικό σημείο της μελέτης μας.

Παρόλες τις ευκολίες των νέων τεχνολογιών, οι σύγχρονες εξελίξεις στο χώρο μας δείχνουν ότι το πρόβλημα δεν έχει λυθεί. Πολλές εταιρίες και γενικότερα υπηρεσίες δημιούργησαν εφαρμογές το κυριότερο μέρος των οποίων ήταν ροές εκτελέσεις που χρησιμοποιούσαν μεταξύ άλλων και δικτυακές υπηρεσίες. Οι εν λόγω ροές εκτέλεσης με την πάροδο του χρόνου μεγάλωναν συνεχώς. Αποτέλεσμα ήταν να έχουμε σήμερα εφαρμογές με ροές εκτέλεσης πολύ μεγάλες. Το μέγεθος των ροών σε πολλές περιπτώσεις έχει γίνει τέτοιο ώστε να κάνει αδύνατο τον έλεγχο και την αλλαγή τους.

Σαν αποτέλεσμα, δημιουργήθηκε η ανάγκη εύρεσης ενός τρόπου περιγραφής των ροών εκτέλεσης και σχεδίασης των κατάλληλων εργαλείων για την δημιουργία εφαρμογών που θα τις εκτελούν. Τα εργαλεία αυτά θα πρέπει να δημιουργούν εφαρμογές οι οποίες θα είναι ανεξάρτητες των ροών. Για το σκοπό αυτό η σύγχρονη επιχειρησιακή έρευνα έχει στραφεί στην δημιουργία μιας περιγραφικής γλώσσας όσον αφορά τις ροές εκτέλεσης. Η έρευνα έχει και σαν σκοπό την δημιουργία προτύπων που θα αφορούν την αρχιτεκτονική των συστημάτων που θα κάνουν χρήση των ροών εκτέλεσης καθώς και των τρόπων υλοποίησής τους. Ο σχεδιασμός των κατάλληλων εργαλείων που θα υλοποιούν τα παραπάνω αποτελεί επίσης πολύ σημαντικό τομέα της έρευνας που διεξάγεται.

Οι απαραίτητες έννοιες που χρειάζονται για την λεπτομερή κατανόηση των παραπάνω περιγράφονται στα επόμενα κεφάλαια, μαζί με όλες τις λεπτομέρειες που αφορούν την υλοποίηση και τη συγχώνευσή τους .

Τέλος είναι πολύ σημαντικό να σημειώσουμε ότι τον όρο «ροές εκτέλεσης» θα τον εγκαταλείψουμε στην συνέχεια αφού είναι πολύ ειδικός. Αντ' αυτού θα χρησιμοποιήσουμε τον όρο **επιχειρησιακές διαδικασίες** αφού ο ίδιος όρος χρησιμοποιείται από τους μελετητές και σχεδιαστές των καινούργιων τεχνολογιών που αναφέραμε παραπάνω. Ο όρος είναι πολύ γενικότερος και αφορά οποιαδήποτε εσωτερική διαδικασία που μπορεί να εμπεριέχεται σε μια εφαρμογή και αποτελείται από την εκτέλεση επιμέρους λειτουργιών. Έτσι και η μελέτη μας γενικεύεται επιτρέποντας την εξαγωγή πολύ πιο χρήσιμων συμπερασμάτων.

1.2 Οργάνωση του τόμου

Παραπάνω κάναμε μια σύντομη εισαγωγή που αφορούσε κυρίως το αντικείμενο της διπλωματικής εργασίας γενικά καθώς και το γενικό πλαίσιο των αναγκών της σύγχρονης τεχνολογίας που μας οδήγησαν στο να μελετήσουμε και να εμβαθύνουμε όσο το δυνατό περισσότερο στο συγκεκριμένο θέμα.

Παρακάτω θα κάνουμε μια πολύ σύντομη περιγραφή στα κεφάλαια που ακολουθούν προκειμένου να διευκολυνθεί ο αναγνώστης στην καλύτερη κατανόηση της μελέτης μας.

Στο **κεφάλαιο 2** γίνεται μια σύντομη περιγραφή του θέματος καθώς και μια προσπάθεια αναλυτικής, περιεκτικής αλλά και σύντομης περιγραφής των βασικών εννοιών που θα μας απασχολήσουν στην συγκεκριμένη διπλωματική εργασία. Οι δύο βασικότερες από αυτές τις έννοιες είναι η έννοια της **δικτυακής υπηρεσίας** και η έννοια της **επιχειρησιακής διαδικασίας**. Επίσης γίνεται και μια προσπάθεια επακριβούς ορισμού του στόχου της διπλωματικής εργασίας αποφεύγοντας όσο γίνεται τις τεχνικές λεπτομέρειες που θα αναπτύξουμε εκτενώς σε άλλα κεφάλαια.

Στο **κεφάλαιο 3** υπάρχει μια εκτενής αναφορά στους διάφορους τρόπους υλοποίησης του συστήματος που θα ορίσουμε παρακάτω. Εφόσον το αντικείμενο είχε αρκετά ερευνητικό

περιεχόμενο, ο τρόπος υλοποίησης ήταν δύσκολο να προσδιοριστεί επακριβώς από την αρχή. Έτσι είχαμε πολλές ανεπιτυχείς υλοποιήσεις, οι οποίες όμως οδήγησαν στην εξαγωγή χρήσιμων συμπερασμάτων. Τα συμπεράσματα αυτά αναπτύσσονται εκτενώς στο εν λόγω κεφάλαιο.

Στο **κεφάλαιο 4** περιγράφουμε την αρχιτεκτονική του συστήματος που υλοποιήθηκε τελικά καθώς και τις επιμέρους λειτουργίες του. Είναι το πιο σημαντικό κεφάλαιο της διπλωματικής εργασίας μαζί με το κεφάλαιο της υλοποίησης αφού όλα τα τελικά συμπεράσματα πηγάζουν από αυτά.

Στο **κεφάλαιο 5** αναφέρονται αναλυτικά οι λεπτομέρειες υλοποίησης. Αναφερόμαστε στις διάφορες πλατφόρμες καθώς και στα εργαλεία που χρησιμοποιήσαμε. Επίσης αναφέρονται διάφορες άλλες λεπτομέρειες όπως κλάσεις κτλ οι οποίες είναι απαραίτητες γνώσεις όχι τόσο στο μελλοντικό χρήστη του συστήματος αλλά σε κάποιον που θα ήθελε να επέμβει σε αυτό και να το βελτιώσει στο μέλλον.

Το **κεφάλαιο 6** αφορά τον έλεγχο του συστήματος. Αναφέρεται αναλυτικά η μέθοδος ελέγχου που χρησιμοποιήθηκε καθώς και τα αποτελέσματά του. Στόχος είναι η απαρίθμηση των ελαττωματικών χαρακτηριστικών του συστήματος έτσι ώστε οι μελλοντικοί χρήστες του να μπορούν να το βελτιώσουν ανάλογα με τις ανάγκες τους.

Το **κεφάλαιο 7** είναι ο επίλογος της διπλωματικής εργασίας. Αφορά τα συμπεράσματα που προέκυψαν κατά την εκπόνησή της. Επίσης αναφέρεται στις μελλοντικές επεκτάσεις που μπορεί να γίνουν σε αυτή, θέμα πολύ ενδιαφέρον αφού το αντικείμενο της διπλωματικής είναι ένας τομέας συνεχώς εξελισσόμενος και σίγουρα θα χρειαστεί να γίνει επέκταση του συγκεκριμένου συστήματος στο μέλλον.

Τέλος το **κεφάλαιο 8** έχει να κάνει με την βιβλιογραφία που χρησιμοποιήθηκε για την εκπόνηση της συγκεκριμένης διπλωματικής εργασίας.

2

Περιγραφή Θέματος

Στο συγκεκριμένο κεφάλαιο προσδιορίζεται επακριβώς το αντικείμενο της διπλωματικής αλλά και ο στόχος της. Προκειμένου να γίνει σωστά κάτι τέτοιο αναφερόμαστε αρχικά όσο το δυνατό περιεκτικότερα και σύντομα στις βασικές έννοιες που θα μας απασχολήσουν. Στην συνέχεια ορίζουμε τις απαιτήσεις μας και τον στόχο μας.

2.1 Βασικές Έννοιες

Οι βασικές έννοιες που θα μας απασχολήσουν παρακάτω είναι το αντικείμενο του συγκεκριμένου υποκεφαλαίου. Ταυτόχρονα με την περιγραφή τους, όπου είναι δυνατόν, γίνεται και μια μικρή ιστορική αναδρομή στις εξελίξεις που οδήγησαν στην ανάπτυξη των εκάστοτε τεχνολογιών. Έτσι ο αναγνώστης θα μπορεί να κατανοήσει πιο εύκολα τις σύγχρονες απαιτήσεις της τεχνολογίας και το στόχο της παρούσας μελέτης. Η πρώτη βασική έννοια με την οποία θα ξεκινήσουμε θα είναι η έννοια της **επιχειρησιακής διαδικασίας** (business process).

2.1.1 Επιχειρησιακές Διαδικασίες (*Business Processes*)

Οι **επιχειρησιακές διαδικασίες** γενικά είναι μια έννοια που χρησιμοποιείται στον χώρο των επιχειρήσεων εδώ και αρκετές δεκαετίες. Το συγκεκριμένο ζήτημα αποτελεί πεδίο διαρκούς έρευνας και αφορά τόσο τις οικονομικές όσο και τις τεχνολογικές επιστήμες γενικότερα. Στόχος της συγκεκριμένης διπλωματικής δεν είναι να ερευνήσει γενικά το αντικείμενο των επιχειρησιακών διαδικασιών αλλά τις επιχειρησιακές διαδικασίες στο χώρο του ηλεκτρονικού εμπορίου και της ηλεκτρονικής διακυβέρνησης. Η έρευνά μας επικεντρώνεται κυρίως σε επιχειρησιακές διαδικασίες που εντάσσονται σε αυτές την τεχνολογία των δικτυακών υπηρεσιών. Το συγκεκριμένο ζήτημα αφορά πολλές μεγάλες εταιρίες παγκοσμίως και αποτελεί πεδίο εντατικής έρευνας η οποία όμως δεν έχει φθάσει σε σημείο να ορίσει κάποιο συγκεκριμένο πρότυπο που θα επέτρεπε και την επέκταση των επιχειρησιακών διαδικασιών σε ένα πολύ χρήσιμο εργαλείο στα χέρια των εταιριών και γενικά όσων χρησιμοποιούν δικτυακές υπηρεσίες και θα ήθελαν να δημιουργήσουν εφαρμογές συνδυάζοντας τις.

Δυστυχώς ο μεγάλος ανταγωνισμός είναι η κύρια αιτία που έχει καθυστερήσει αρκετά την δημιουργία του προτύπου αν και οι πρόσφατες εξελίξεις είναι ενθαρρυντικές και μάλλον πολύ σύντομα θα υπάρξει το πρότυπο στο οποίο αναφερόμαστε. Παρακάτω θα προχωρήσουμε σε μια σύντομη αναφορά των διάφορων εξελίξεων όπως αυτές συνέβησαν χρονολογικά στο χώρο του ηλεκτρονικού εμπορίου και της ηλεκτρονικής διακυβέρνησης και

αφορούσαν τις επιχειρησιακές διαδικασίες. Στη συνέχεια θα επικεντρωθούμε στα σύγχρονα πρότυπα που προτείνονται για τη δημιουργία επιχειρησιακών διαδικασιών που χρησιμοποιούν δικτυακές υπηρεσίες και θα πραγματοποιήσουμε μια σύντομη αλλά αναλυτική περιγραφή τους.

Το ενδιαφέρον για τις επιχειρησιακές διαδικασίες εκφράστηκε στο χώρο των ηλεκτρονικών επιχειρήσεων σε διάφορες μορφές.

Το UN/CEFACT - Ηνωμένα Έθνη/ Κέντρο διευκόλυνσης εμπορίου και ηλεκτρονικών επιχειρήσεων, επικέντρωσε στα τέλη της δεκαετίας του 90 το μεγαλύτερο μέρος των εργασιών του στο Open-edi Reference Model. Το Open-edi Reference Model, που στη συνέχεια έγινε πρότυπο (ISO standard 14662), καθόριζε ένα τρόπο επικοινωνίας και αλληλεπίδρασης εταιριών με εμπορικούς συνεργάτες. Το συγκεκριμένο πρότυπο, σε αντίθεση με το έως τότε χρησιμοποιούμενο μοντέλο που περιέγραφε λεπτομερώς την τεχνολογία που οι επιμέρους συνεργάτες σε μια διαδικασία χρησιμοποιούσαν και αναγνώριζε τις επιχειρησιακές διαδικασίες εμπεριεχομένου και των μελών που τις χρησιμοποιούσαν. Αναγνώριζε επίσης τις διενέργειες και τα μηνύματα προς και από τις διαδικασίες αυτές που μπορούσαν να πραγματοποιηθούν, καθώς και την ερμηνεία αυτών των μηνυμάτων.

Η πρωτοποριακή εργασία του UN/CEFACT συνέβαλε δραστικά στη δημιουργία του Business Process Specification Schema ή BPSS που αναπτύχθηκε σαν πρωτοβουλία της ebXML. Η BPSS συμπεριέλαβε και τα δύο πρότυπα που αναφέρονται παρακάτω σε ένα νέο πλαίσιο. Το πλαίσιο αυτό περιελάμβανε μεταξύ άλλων και ευρετήρια, ανταλλαγή μηνυμάτων, συμφωνίες μεταξύ των εμπορικών συνεργατών, επικοινωνία μέσω μηνυμάτων και ενιαίο τρόπο ερμηνείας των. Το συγκεκριμένο πλαίσιο υλοποιήθηκε τόσο σε XML όσο και σε UML και στηρίχθηκε στην αρχιτεκτονική του ebXML.

Το ως άνω περιγραφόμενο συγκεντρωτικό ενδιαφέρον στις επιχειρησιακές διαδικασίες επηρέασε σε μεγάλο βαθμό τις εργασίες του RosettaNet . Ένα νέο πλαίσιο για τις ηλεκτρονικές υπηρεσίες σχεδιάστηκε και μάλιστα έγινε άμεσα αποδεκτό ως μοντέλο από πολλές επιχειρήσεις. Οι αρχιτέκτονες του RosettaNet αντιστάθηκαν στην παρόρμηση να αναπτύξουν ένα λεξιλόγιο επικοινωνίας αποκλειστικά βασισμένο σε XML και συγκεντρώθηκαν στον ορισμό επιχειρησιακών διαδικασιών για την ηλεκτρονική βιομηχανία που αυτοί περιέγραψαν ως Partner Interface Processes ή PIPs. Το κάθε PIP αναπαριστά μία ή περισσότερες διαδράσεις και περιλαμβάνει τα ηλεκτρονικά επιχειρησιακά έγγραφα που ανταλλάσσονται, το λεξιλόγιο που χρησιμοποιείται για το καθένα. Το RosettaNet δημοσίευσε 6 συλλογές από PIPs, τα οποία καλύπτουν ισάριθμες επιχειρησιακές λειτουργίες, μία εκ των οποίων αφορά την διαχείριση.

Ανάμεσα στα ήδη υπάρχοντα XML λεξικά για ηλεκτρονικές επιχειρήσεις το RosettaNet αποτέλεσε μια από τις πιο επιτυχημένες πρωτοβουλίες, χάρη στο ότι ένα μεγάλο μέρος του αφιερώθηκε στο να περιγράψει business processes.

2.1.2 Επιχειρησιακές Διαδικασίες και Δικτυακές Υπηρεσίες

Η ανάγκη για αναπαράσταση επιχειρησιακών διαδικασιών σε εφαρμογές με δικτυακές υπηρεσίες, όπως φάνηκε από την επιτυχία του RosettaNet, οδήγησε στη δημιουργία πολλών άλλων XML λεξιλογίων και προτύπων δικτυακών υπηρεσιών που αφορούν αυτά τα ζητήματα. Οι δύο πιο σημαντικές συμβολές είναι the Business Process Modeling Language and the Business Process Execution Language for Web Services.

Η **Business Processes Modeling Language** ή **BPML** αναπαριστά επιχειρησιακές διαδικασίες με τη βοήθεια μιας μεταγλώσσας βασισμένης στην XML. Η BPML ορίζει πάνω στις επιχειρησιακές διαδικασίες τις εξής λειτουργίες: ροή ελέγχου, ροή δεδομένων και ροή γεγονότων, ενώ παράλληλα προσθέτει και δυνατότητες για τον ορισμό επιχειρησιακών ρόλων, ρόλων ασφαλείας και ανταλλαγής μηνυμάτων. Σκοπός της BPML είναι επίσης να αναπτύξει κατάλληλο γραφικό περιβάλλον και γλώσσα ερωτήσεων.

Μια συνεργασία μεταξύ της IBM, της Microsoft και της BEA, ανέπτυξε την **Business Process Execution Language for Web Services**, με το ακρωνύμιο **BPEL4WS**. Η συγκεκριμένη γλώσσα έχει αναπτυχθεί ειδικά για να δουλεύει με δικτυακές υπηρεσίες και έχει ενσωματώσει τις εργασίες της Microsoft's XLANG and IBM's Web Services Flow Language. Η BPEL4WS διαχώρισε τις επιχειρησιακές διαδικασίες με τα επιχειρησιακά πρωτόκολλα. Οι επιχειρησιακές διαδικασίες αναπαριστούν έναν συμμετέχοντα σε μία επιχειρησιακή συναλλαγή. Τα επιχειρησιακά πρωτόκολλα περιγράφουν την συναλλαγή, έτσι ώστε να είναι ορατή σε όλα τα εμπλεκόμενα μέρη, και εφόσον δεν αποκαλύπτουν την εσωτερική συμπεριφορά των τελευταίων, ονομάζονται αφηρημένες διαδικασίες.

Για να κατανοήσουμε καλύτερα τις απαιτήσεις μιας περιγραφικής γλώσσας επιχειρησιακών διαδικασιών που εντάσσει σε αυτή και την τεχνολογία των δικτυακών υπηρεσιών παραθέτουμε στη συνέχεια μια σύντομη ανάλυση των τελευταίων, καθώς και των απαιτήσεων που προκύπτουν όσον αφορά τη γλώσσα στην οποία αναφερθήκαμε.

2.1.3 Εισαγωγή στις δικτυακές υπηρεσίες

Ο όρος **δικτυακές υπηρεσίες** όπως υπονοεί το όνομά του αναφέρεται σε υπηρεσίες προσβάσιμες μέσω του διαδικτύου. Στον όρο όμως εμπεριέχονται πολύ περισσότερες έννοιες απ' ό,τι υπονοεί η ονομασία του. Με την έννοια δικτυακές υπηρεσίες αναφερόμαστε και στην αρχιτεκτονική, τα πρότυπα, την τεχνολογία και τα επιχειρησιακά μοντέλα που απαιτούνται για να υπάρξουν και να υλοποιηθούν οι δικτυακές υπηρεσίες. Παρακάτω παρατίθεται ο ορισμός των δικτυακών υπηρεσιών, όπως αυτός δίνεται από την IBM

Οι δικτυακές υπηρεσίες αποτελεί την καινούργια γενιά δικτυακής εφαρμογής. Είναι αυτοπεριεχόμενες, αυτοπεριγραφόμενες εφαρμογές που δημοσιεύονται και τοποθετούνται στο διαδίκτυο αλλά και καλούνται μέσα από αυτό. Οι δικτυακές υπηρεσίες υλοποιούν λειτουργίες που μπορεί να είναι από απλές συναρτήσεις μέχρι σύνθετες επιχειρησιακές διαδικασίες.

Με άλλα λόγια οι δικτυακές υπηρεσίες είναι κατασκευαστικές μονάδες για την δημιουργία εφαρμογών. Σαν παράδειγμα θα μπορούσαμε να θεωρήσουμε μια κατανεμημένη ψηφιακή βιβλιοθήκη κατασκευασμένη από δικτυακές υπηρεσίες που η κάθε μια τους υλοποιεί λειτουργίες όπως κατανεμημένη αναζήτηση, πιστοποίηση, αίτημα ενοικίασης, πληρωμή, μετάφραση κ.α.. Αυτές οι δικτυακές υπηρεσίες θα συνδυαστούν με την βοήθεια μιας εφαρμογής ηλεκτρονική βιβλιοθήκης, η οποία δημιουργεί το απαραίτητο περιβάλλον για να γίνουν προσβάσιμες οι διάφορες πληροφοριακές πηγές από μια κλειστή εξουσιοδοτημένη κοινότητα χρηστών.

Η σύγχρονη μορφή του διαδικτύου επιτρέπει στους διάφορους χρήστες να συνδεθούν σε διάφορες εφαρμογές στο διαδίκτυο. Οι δικτυακές υπηρεσίες επιτρέπουν σε εφαρμογές να συνδεθούν με άλλες εφαρμογές. Με άλλα λόγια από το πρότυπο επιχείρηση προς πελάτη, περνάμε στο πρότυπο επιχείρηση προς επιχείρηση.

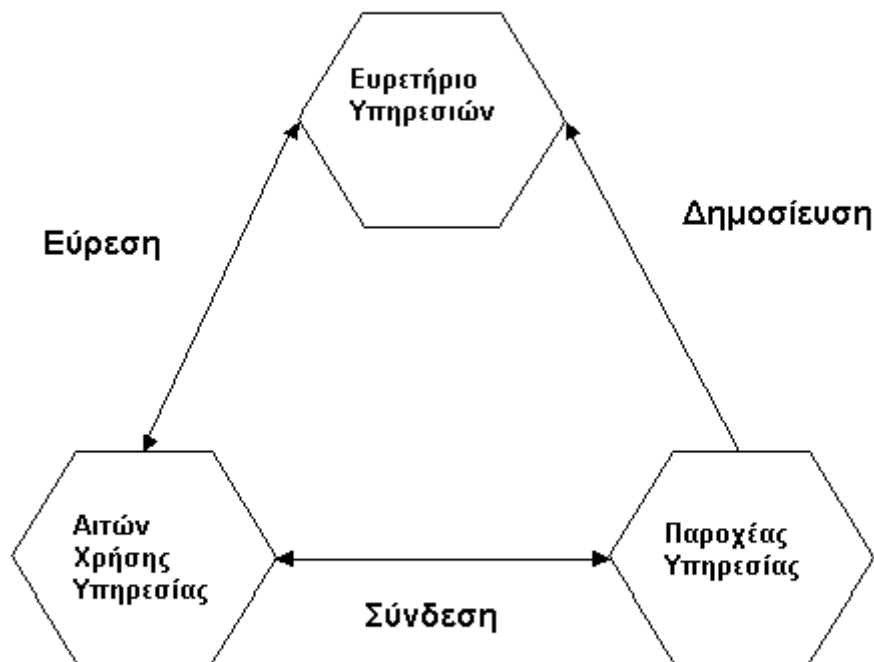
2.1.3.1 Αρχιτεκτονική των δικτυακών υπηρεσιών.

Επιλέξαμε πιο πάνω το παράδειγμα της ηλεκτρονικής βιβλιοθήκης για να περιγράψουμε τις λειτουργίες των δικτυακών υπηρεσιών. Η επιλογή του συγκεκριμένου παραδείγματος θα μας βοηθήσει να καταλάβουμε την αρχιτεκτονική που απαιτεί η τεχνολογία των δικτυακών υπηρεσιών για την υλοποίησή της αφού αυτή είναι αρκετά όμοια με την αρχιτεκτονική του παραδείγματος της βιβλιοθήκης. Για την ακρίβεια, τα μεταδεδομένα που χρησιμοποιεί η εφαρμογή της ηλεκτρονικής βιβλιοθήκης για την πιστοποίηση, αναζήτηση κλπ είναι επίσης εφαρμόσιμα στην τεχνολογία των δικτυακών υπηρεσιών.

Η τεχνολογία των δικτυακών υπηρεσιών έχει το πρόβλημα της επικοινωνίας με τις υπηρεσίες αφού αποκτηθεί πρόσβαση σε αυτές. Με βάση τα σύγχρονα δεδομένα το πρόβλημα της επικοινωνίας δεν υφίσταται αφού π.χ. στην περίπτωση που η επικοινωνία γίνεται μέσω κειμένων, τα κείμενα μπορούν να σταλούν μέσω του πρωτοκόλλου MIME και αφού πραγματοποιηθεί η λήψη τους, να επιλεγεί η κατάλληλη εφαρμογή προκειμένου να γίνει η ανάγνωσή τους. Το πρόβλημα έχει να κάνει με τις μελλοντικές επεκτάσεις των δικτυακών υπηρεσιών που η επικοινωνία θα πρέπει να γίνεται όχι με κείμενα αλλά με τύπους δεδομένων που δεν έχουν οριστεί ακόμα. Χρειάζεται λοιπόν ο client να γνωρίζει εξ' αρχής τον τύπο της υπηρεσίας στην οποία προσπαθεί να αποκτήσει πρόσβαση. Έτσι μια εφαρμογή που δημιουργήθηκε πριν από τη δικτυακή υπηρεσία να μπορεί να την χρησιμοποιεί.

Η IBM έχει δημοσιεύσει μια πιθανή αρχιτεκτονική όσον αφορά τις δικτυακές υπηρεσίες. Θα αναπτύξουμε σύντομα την αρχιτεκτονική αυτή ως μια ενδεικτική αρχιτεκτονική για δικτυακές υπηρεσίες χωρίς να σημαίνει ότι είναι και η μοναδική επιλογή.

Η αρχιτεκτονική της IBM ορίζει την εσωτερική δομή που απαιτείται προκειμένου να υποστηριχθούν οι δικτυακές υπηρεσίες στα πλαίσια τριών ρόλων: παροχέας υπηρεσίας, αιτών χρήσης υπηρεσίας και ευρετηρίων υπηρεσιών. Οι τρεις πράξεις που ορίζονται προκειμένου να υλοποιηθεί η επικοινωνία μεταξύ των τριών ρόλων είναι οι δημοσίευση, εύρεση και σύνδεση. Η σύνδεση είναι η λειτουργία εκείνη που επιτρέπει στις διάφορες εφαρμογές να συνδεθούν σε μια υπηρεσία που είναι δημοσιευμένη στο διαδίκτυο και να αρχίσουν να ανταλλάσσουν μηνύματα μαζί της.



Σ' αυτή την αρχιτεκτονική η υπηρεσία είναι η υλοποίηση της περιγραφής της υπηρεσίας που είναι τα μεταδεδομένα που την περιγράφουν. Αυτά τα μεταδεδομένα πρέπει να περιέχουν αρκετά πληροφοριακά στοιχεία ώστε ο πιθανός χρήστης να μπορεί να έχει πρόσβαση σε αυτή, περιλαμβανομένης τόσο της διεπαφής όσο και της τοποθεσίας της υπηρεσίας.

Ο παροχέας της υπηρεσίας δημοσιεύει την περιγραφή της σε ένα ευρετήριο. Ο πιθανός χρήστης στη συνέχεια, βρίσκει την περιγραφή της υπηρεσίας μέσω του ευρετηρίου. Η περιγραφή περιέχει αρκετά στοιχεία για τον πιθανό χρήστη προκειμένου να συνδεθεί στον παροχέα και να χρησιμοποιήσει την υπηρεσία.

2.1.3.2 Πρότυπα για δικτυακές υπηρεσίες

Παρόλο που η αρχιτεκτονική που περιγράψαμε παραπάνω μπορεί να θεωρηθεί ανεξάρτητα από κάποιο πρότυπο είναι σαφές ότι η ανεξαρτησία εφαρμογών είναι απαραίτητη για να γίνει η παραπάνω (και η οποιαδήποτε άλλη) αρχιτεκτονική ευρέως αποδεκτή. Για να επιτευχθεί ο συγκεκριμένος σκοπός ένα μεγάλο ποσοστό εκπροσώπων σημαντικών επιχειρήσεων προσπαθούν να ορίσουν ένα πρότυπο στηριγμένο στη γλώσσα XML που θα κάνει εφικτή την υλοποίηση της αρχιτεκτονικής των δικτυακών υπηρεσιών.

Πρωταρχικής σημασίας θεωρείται ο ορισμός μια συγκεκριμένης μεθόδου περιγραφής της κάθε δικτυακής υπηρεσίας. Για το σκοπό αυτό δημιουργήθηκε η περιγραφική γλώσσα **WSDL**. Η WSDL περιγράφει κάθε δικτυακή υπηρεσία σαν ένα σύνολο από «θύρες», οι οποίες κατηγοριοποιούν μια σειρά από διεργασίες που μπορούν να πραγματοποιηθούν ανάμεσα σε ένα χρήστη και στη δικτυακή υπηρεσία. Οι διεργασίες αυτές προαιρετικά ονομάζονται «λειτουργίες» και έχουν ένα μήνυμα εισόδου και ενίοτε και κάποιο μήνυμα

εξόδου. Η κάθε λειτουργία περιγράφει μια διεργασία η οποία μπορεί να πραγματοποιηθεί μεταξύ χρήστη και δικτυακής υπηρεσίας. Αυτό μπορεί να σημαίνει πως η εφαρμογή που την χρησιμοποιεί ζητά από την δικτυακή υπηρεσία να κάνει κάτι και να επιστρέψει το αποτέλεσμα, αλλά και ότι η δικτυακή υπηρεσία μπορεί να εκκινήσει μια διαδικασία για την οποία η εφαρμογή πρέπει να αντιδράσει αναλόγως.

Υπάρχουν δύο είδη χρήσεων για ένα WSDL αρχείο:

- Κατά την διάρκεια του σχεδιασμού μιας εφαρμογής ο δημιουργός της πρέπει να ξέρει την διεπαφή την οποία θα χρησιμοποιήσει για να επικοινωνήσει με την δικτυακή υπηρεσία και να εντοπίσει τις διάφορες διεργασίες που αυτή μπορεί να του προσφέρει.
- Κατά την διάρκεια της υλοποίησης της εφαρμογής θα πρέπει να είναι γνωστή ακριβώς και η υλοποίηση της κάθε λειτουργίας της δικτυακής υπηρεσίας ώστε να μπορεί η εφαρμογή να συνδεθεί κάθε μια από τις λειτουργίες που του παρέχει η δικτυακή υπηρεσία.

Η WSDL μπορεί να πραγματοποιήσει και τα δύο αυτά στάδια.

Η WSDL περιγράφει, όπως είδαμε, μία δικτυακή υπηρεσία στα πλαίσια των λειτουργιών που αυτή μπορεί να υλοποιήσει. Με τον τρόπο αυτό δεν περιγράφεται όμως η ακριβής μέθοδος που θα χρησιμοποιήσει μια εφαρμογή για να επικοινωνήσει μια εφαρμογή με τη δικτυακή υπηρεσία. Η WSDL επιτρέπει το προσδιορισμό ενός συγκεκριμένου προτύπου 'σύνδεσης' με την δικτυακή υπηρεσία. Στην πράξη αυτό οδηγεί σε ένα πρωτόκολλο επικοινωνίας επίσης στηριγμένο στη γλώσσα XML, το SOAP.

Το **SOAP**(Simple Object Access Protocol) αποτελεί ένα πρότυπο για επικοινωνία μέσω ανταλλαγής πληροφοριών (κωδικοποιημένες με τρόπο που έχει ως βάση του την XML) ανάμεσα σε εφαρμογές.

Το SOAP επιτρέπει να εκκινήσει μια λειτουργία με τη συγκεκριμένη ακολουθία χαρακτήρων (web service) στέλνοντας σε αυτή ένα SOAP μήνυμα. Το SOAP ορίζει τη μορφή των XML κειμένων που στέλνονται πάνω από το HTTP είτε σαν ερωτήσεις είτε σαν απαντήσεις από μια δικτυακή υπηρεσία. Επίσης έχει προβλεφθεί συγκεκριμένος τρόπος κωδικοποίησης WSDL μηνυμάτων.

Με την βοήθεια του SOAP και της WSDL μπορούμε να περιγράψουμε μια δικτυακή υπηρεσία και να χρησιμοποιήσουμε δικτυακές υπηρεσίες μέσα από εφαρμογές. Το μόνο που μένει πλέον να καθοριστεί είναι ένας τρόπος ανεύρεσης δικτυακών υπηρεσιών. Για το σκοπό αυτό δημιουργήθηκε ένα πρότυπο όσον αφορά τη δημιουργία ευρετηρίων για την ανεύρεση των εκάστοτε δικτυακών υπηρεσιών που διατίθενται προς χρήση, το Universal Discovery, Description and Integration (UDDI).

Ένα **UDDI** ευρετήριο είναι και αυτό μία δικτυακή υπηρεσία, το οποίο μπορεί να χρησιμοποιήσει μια εφαρμογή προκειμένου να βρει τις δικτυακές υπηρεσίες που έχουν δημοσιευτεί κατά την συγκεκριμένη χρονική περίοδο. Στην πραγματικότητα το UDDI μπορεί να περιέχει και πολλά ακόμα μεταδεδομένα που αφορούν λέξεις κλειδιά, τρόπους πληρωμής, ασφάλεια κλπ.

2.1.4 Δικτυακές Υπηρεσίες και Επιχειρησιακές Διαδικασίες

Στόχος των δικτυακών υπηρεσιών είναι η επίτευξη καθολικής ανεξαρτησίας εφαρμογών μεταξύ των διαφόρων δικτυακών προτύπων. Σε πρώτη φάση, η ανεξαρτησία των εφαρμογών εξασφαλίζεται πλήρως χάρη στα τα τρία αυτά πρότυπα. Η χρήση πολλών διαφορετικών συστημάτων, όμως, απαιτεί περισσότερα από τη δυνατότητα της διεξαγωγής απλών

συναλλαγών μεταξύ συγκεκριμένων και ανεξάρτητων πρωτοκόλλων. Η πλήρης δυναμική των δικτυακών υπηρεσιών ως πλατφόρμα χρήσης ανεξάρτητων εφαρμογών θα επιτευχθεί μόνο αν οι εφαρμογές και οι επιχειρησιακές διαδικασίες γίνουν ικανές να υλοποιήσουν τις πολύπλοκες συναλλαγές τους με ένα πρότυπο ανεξάρτητο από τις ίδιες.

Γενικά τα πρότυπα για επιχειρησιακές διαδράσεις χρησιμοποιούν peer to peer ανταλλαγές μηνυμάτων τόσο σύγχρονων όσο και ασύγχρονων. Για την περιγραφή αυτών των επιχειρησιακών διαδράσεων χρειάζεται μια επίσημη περιγραφή των πρωτοκόλλων ανταλλαγής μηνυμάτων. Η περιγραφή τέτοιων επιχειρησιακών πρωτοκόλλων περιλαμβάνει τον ακριβή προσδιορισμό της συμπεριφοράς των μηνυμάτων που ανταλλάσσονται έτσι ώστε να μην αποκαλύπτεται ο ακριβής τρόπος υλοποίησης των εσωτερικών διαδικασιών που χρησιμοποιεί η μια επιχείρηση από την άλλη.

Υπάρχουν δύο καλοί λόγοι για να διαχωρίσουμε το δημόσιο από το ιδιωτικό μέρος μιας επιχειρησιακής διαδικασίας. Ο πρώτος είναι ότι οι επιχειρήσεις προφανώς δεν θέλουν να αποκαλύπτουν όλη την εσωτερική διαδικασία λήψης αποφάσεων και διαχειρίσεις των δεδομένων τους στους συνεργάτες τους. Ο άλλος είναι ότι ο διαχωρισμός μεταξύ του δημόσιου και του ιδιωτικού μέρους παρέχει την ελευθερία της αλλαγής του ιδιωτικού μέρους της υλοποίησης χωρίς να επηρεάζεται το πρωτόκολλο του δημοσίου μέρους.

Τα επιχειρησιακά πρωτόκολλα πρέπει να περιγράφονται με σαφήνεια με τρόπο ανεξάρτητο της εκάστοτε πλατφόρμας και να περιλαμβάνουν όλες τις πτυχές της συμπεριφοράς που έχουν διεπιχειρησιακή σημασία. Κάθε συμμετέχων μπορεί να σχεδιάζει επιχειρησιακές διαδικασίες συμβατές με το παραπάνω πρωτόκολλο χωρίς να εντάσσεται στο σχεδιασμό η οποιαδήποτε ανθρώπινη συμφωνία. Με τον τρόπο αυτό αποφεύγεται η κύρια δυσκολία που υπάρχει σήμερα στη δημιουργία διεπιχειρησιακών διαδικασιών.

Προκειμένου να προσδιορίσουμε τις κύριες έννοιες που απαιτούνται για να περιγράψουμε ένα επιχειρησιακό πρωτόκολλο όπως το περιγράψαμε παραπάνω, πρέπει να θεωρήσουμε τα εξής:

- Τα επιχειρησιακά πρωτόκολλα περιλαμβάνουν συμπεριφορά εξαρτώμενη από τα δεδομένα που χρησιμοποιούν.
- Η ικανότητα να προσδιορίζουμε οριακές συνθήκες εκτός του πλαισίου λειτουργίας, περιλαμβανομένων και τρόπων ανάνηψης του συστήματος από αυτές είναι εξίσου, σημαντική με την υπό φυσιολογικές συνθήκες καλή λειτουργία του.
- Οι μακράς διάρκειας διαδράσεις περιλαμβάνουν πολλαπλές, συχνά εμφωλευμένες, μονάδες εργασίας, η καθεμιά με τις δικές της απαιτήσεις δεδομένων. Για το λόγο αυτό τα επιχειρησιακά πρωτόκολλα απαιτούν την ευκολία στον προσδιορισμό των συντεταγμένων, στις οποίες βρίσκονται τα αποτελέσματα της εκάστοτε μονάδας εργασίας (διαδικασίας).

Αν θέλαμε, λοιπόν, να ορίσουμε ακριβείς περιγραφές των διαφόρων υπηρεσιών για κάποιο διεπιχειρησιακό Business Process πρωτόκολλο, χρειαζόμαστε μια πλούσια περιγραφικά γλώσσα, η οποία θα περιέχει και πολλά στοιχεία γλώσσας εκτέλεσης. Το κλειδί για την επιτυχημένη υλοποίηση μιας τέτοιας γλώσσας είναι ο διαχωρισμός μεταξύ των δημόσιων μηνυμάτων που ανταλλάσσονται και των εσωτερικών διαδικασιών που εκτελούνται έτσι ώστε οι διαδικασίες αυτές να μπορούν να χειρίζονται δεδομένα, ώστε να μη χρειάζεται να περιγραφούν από δημόσιο πρωτόκολλο.

Παρακάτω παραθέτουμε μια αναλυτική περιγραφή των δύο γλωσσών που ήδη υπάρχουν στο συγκεκριμένο τομέα: της BPEL4WS και της BPML.

2.1.5 BPEL4WS

Η BPEL4WS ακολουθεί πλήρως το πρότυπο που περιγράψαμε παραπάνω. Όσον αφορά τα μηνύματα, τα ανταλλάσσόμενα μεταξύ διαδικασιών και επιχειρήσεων, χρησιμοποιεί τη λογική των ιδιοτήτων μηνύματος προκειμένου να αναγνωρίσει τα σχετικά με το πρωτόκολλο δεδομένα που εμπεριέχονται σ' αυτά. Οι ιδιότητες αυτές μπορεί να θεωρηθούν ότι είναι διαφανή δεδομένα σε μια προοπτική δημόσιας ανταλλαγής δεδομένων, η οποία αντιτίθεται στα αδιαφανή δεδομένα που ανταλλάσσονται μεταξύ εσωτερικών / ιδιωτικών συναρτήσεων που χρησιμοποιούνται. Τα διαφανή δεδομένα επηρεάζουν το επιχειρησιακό πρωτόκολλο με έναν άμεσο τρόπο, ενώ τα αδιαφανή δεδομένα το επηρεάζουν μόνο δημιουργώντας μη-ντετερμινισμό, γιατί ο τρόπος, με τον οποίο επηρεάζουν τις αποφάσεις είναι αδιαφανής. Η BPEL4WS θεωρεί ότι κάθε δεδομένο πρέπει να είναι διαφανές και γι' αυτό το λόγο το βλέπει σαν ιδιότητα του μηνύματος.

Οι έννοιες που χρειάζονται για να ορίσουμε ένα επιχειρησιακό πρωτόκολλο, καθώς και οι έννοιες που χρειάζονται για να ορίσουμε ένα εκτελέσιμο Business Process δημιουργούν μια εννοιολογική αλυσίδα και η BPEL4WS είναι σχεδιασμένη για να καλύψει όλο το εύρος της. Η γλώσσα αυτή ορίζει ένα μοντέλο και μια γραμματική προκειμένου να περιγράψει τη συμπεριφορά μιας Business Process βασισμένη σε συναλλαγές μεταξύ της διαδικασίας και των επιμέρους χρηστών της. Η αλληλεπίδραση με τον κάθε χρήστη υλοποιείται μέσω διεπαφών που αντιστοιχούν σε δικτυακές υπηρεσίες, και η δομή της σχέσης στο επίπεδο της διεπαφής είναι ενθυλακωμένη σε αυτό που ονομάζουμε σύνδεσμο υπηρεσιών (Service Link). Η BPEL4WS διαδικασία ορίζει το πως διαφορετικές υπηρεσίες μπορούν να αλληλεπιδράσουν με τους χρήστες της προκειμένου να πετύχουν έναν επιχειρησιακό στόχο, καθώς και την λογική και την κατάσταση που απαιτείται για να πραγματοποιηθεί ένας τέτοιος προσανατολισμός. Η γλώσσα αυτή, επίσης, εισάγει συστηματικούς μηχανισμούς για την επίτευξη του ελέγχου πιθανών επιχειρησιακών απρόβλεπτων οριακών συνθηκών και σφαλμάτων της διαδικασίας. Τέλος, η BPEL4WS εισάγει ένα μηχανισμό για τον καθορισμό του πως ατομικές ή σύνθετες λειτουργίες εντός μιας διαδικασίας, μπορούν να αποζημιωθούν στην περίπτωση που οι παραπάνω εξαιρέσεις εμφανιστούν ή ένας συνεργάτης / χρήστης ζητήσει ακύρωση της συμμετοχής του στην διαδικασία.

Η βασική ιδέα της BPEL4WS μπορεί να εφαρμοστεί με δύο τρόπους. Μια διαδικασία της μπορεί να ορίσει ένα ρόλο επιχειρησιακού πρωτοκόλλου χρησιμοποιώντας την ιδέα της αφηρημένης διαδικασίας (abstract process). Π.χ. σε ένα πρωτόκολλο εμπορικής συναλλαγής, ο πωλητής και ο αγοραστής είναι δύο διακριτοί ρόλοι, ο καθένας εκ των οποίων αποτελεί μια τέτοια αφηρημένη διαδικασία. Η σχέση τους είναι τυπικά μοντελοποιημένη σαν ένας σύνδεσμος υπηρεσίας (Service Link). Οι αφηρημένες διαδικασίες χρησιμοποιούν όλο το εύρος των ιδεών της BPEL4WS, αλλά προσεγγίζουν τον χειρισμό των δεδομένων με τρόπο που αντανάκλα το επίπεδο της αφαίρεσης που χρειάζεται για να περιγράψουμε τις διάφορες προοπτικές ενός επιχειρησιακού πρωτοκόλλου. Πιο συγκεκριμένα, οι αφηρημένες διαδικασίες διαχειρίζονται μόνο δεδομένα που σχετίζονται με το επιχειρησιακό πρωτόκολλο. Επιπρόσθετα, οι αφηρημένες διαδικασίες χρησιμοποιούν μη-ντετερμινιστικές τιμές δεδομένων προκειμένου να κρύψουν τα ιδιωτικά μέρη της συμπεριφοράς τους.

Είναι επίσης δυνατό να χρησιμοποιήσουμε την BPEL4WS για να ορίσουμε μια εκτελέσιμη Business Process. Η λογική και η κατάσταση της διαδικασίας ορίζουν τη φύση και την ακολουθία των αλληλεπιδράσεων μεταξύ δικτυακές υπηρεσίες που διεξάγονται σε κάθε χρήστη / συνεργάτη της και άρα και στα πρωτόκολλα αλληλεπίδρασης. Όσο ένας ορισμός μιας διαδικασίας σε BPEL4WS δε χρειάζεται να είναι ολοκληρωμένος όσον αφορά το ιδιωτικό μέρος της υλοποίησης, η γλώσσα αποτελεσματικά ορίζει μια μεταφέρσιμη μορφή εκτέλεσης για Business Processes, η οποία στηρίζεται αποκλειστικά στους πόρους της δικτυακής υπηρεσίας και στα δεδομένα σε μορφή XML. Επιπλέον, τέτοιες διαδικασίες εκτελούνται και αλληλεπιδρούν με τους χρήστες / συνεργάτες με ένα συνεπή τρόπο

ανεξάρτητα από την πλατφόρμα που τις υποστηρίζει ή το προγραμματιστικό μοντέλο που χρησιμοποιείται για την υλοποίηση του περιβάλλοντος που τη φιλοξενεί.

Ακόμα και όταν τα ιδιωτικά μέρη της υλοποίησης χρησιμοποιούν λειτουργίες που εξαρτώνται από τις πλατφόρμες στις οποίες έχουν δημιουργηθεί, κάτι που είναι πιθανό στις περισσότερες πραγματικές περιπτώσεις, η λογική συνέχεια του βασικού μοντέλου μεταξύ αφηρημένων και επιχειρησιακών διαδικασιών στην BPEL4WS κάνει δυνατή την εξαγωγή και εισαγωγή των δημόσιων παραμέτρων, που είναι ενσωματωμένες στο επιχειρησιακό πρωτόκολλο σαν διαδικασίες, ή σαν ρόλοι, διατηρώντας ταυτόχρονα το σκοπό και τη δομή των πρωτοκόλλων. Αυτό είναι κατά γενική ομολογία η πιο ελκυστική προοπτική για να χρησιμοποιήσει κανείς την BPEL4WS από την οπτική ότι απελευθερώνει τις δυνατότητες των δικτυακών υπηρεσιών εφόσον επιτρέπει την ανάπτυξη εργαλείων και άλλων τεχνολογιών που αυξάνουν το επίπεδο της αυτοματοποίησης και συνεπώς μειώνουν το κόστος εγκατάστασης διεπιχειρησιακών αυτόματων επιχειρησιακών διαδικασιών.

Η BPEL4WS έχει διαστρωματωθεί στην κορυφή διάφορων XML προτύπων: WSDL 1.1, XML Schema και XPath 1.0. Τα WSDL μηνύματα και οι ορισμοί τύπων στο XML Schema παρέχουν το μοντέλο δεδομένων που χρησιμοποιείται από τις διαδικασίες της BPEL4WS. Το XPath παρέχει υποστήριξη στη διαχείριση των δεδομένων. Όλοι οι εξωτερικοί πόροι και συνεργάτες αναπαρίστανται σαν WSDL υπηρεσίες. Η BPEL4WS παρέχει επεκτασιμότητα προκειμένου να συμπεριλάβει στο μέλλον πιθανές επεκτάσεις των συγκεκριμένων προτύπων και πιο ειδικά το XPath και τα συσχετιζόμενα πρότυπα που χρησιμοποιούνται στη δημιουργία αρχείων XML.

Μεταξύ των παραπάνω προτύπων η WSDL έχει την καλύτερη επιρροή πάνω στην γλώσσα BPEL4WS. Το μοντέλο μιας BPEL4WS διαστρωματώνεται στην κορυφή του υπηρεσιακού μοντέλου που ορίζει η WSDL 1.1. Στο κέντρο του BPEL4WS μοντέλου διαδικασίας είναι η peer-to-peer αλληλεπίδραση μεταξύ υπηρεσιών που περιγράφονται από την WSDL 1.1. Τόσο οι διαδικασίες όσο και οι χρήστες / συνεργάτες μοντελοποιούνται σαν WSDL υπηρεσίες. Μια Business Process ορίζει τον τρόπο με τον οποίο μπορεί να γίνει ο προσανατολισμός των αλληλεπιδράσεων μεταξύ ενός στιγμιότυπου μιας διαδικασίας και των χρηστών / συνεργατών της. Ένας ορισμός μιας BPEL4WS διαδικασίας παρέχει και / ή χρησιμοποιεί μία ή περισσότερες υπηρεσίες, και χρησιμοποιεί την περιγραφή της συμπεριφοράς και της αλληλεπίδρασης ενός στιγμιότυπου μιας διαδικασίας με τους συνεργάτες της και τους πόρους μέσω διεπαφών δικτυακών υπηρεσιών. Δηλαδή η BPEL4WS ορίζει πρωτόκολλα ανταλλαγής μηνυμάτων ακολουθούμενα από μία επιχειρησιακή διαδικασία ενός συγκεκριμένου ρόλου στη διάδραση.

Ο ορισμός μιας BPEL4WS διαδικασίας ακολουθεί επίσης και το μοντέλο WSDL όσον αφορά το διαχωρισμό ανάμεσα σε αφηρημένα περιεχόμενα μηνυμάτων, όπως αυτά χρησιμοποιούνται σε μια επιχειρησιακή διαδικασία, καθώς πληροφορίες για την υλοποίησή της (μηνύματα και τύπους θυρών ενάντια σε πληροφορίες διεύθυνσης και σύνδεσης). Συγκεκριμένα, μια BPEL4WS αναπαριστά τους συνεργάτες / χρήστες, καθώς και τις αλληλεπιδράσεις μεταξύ αυτών στα πλαίσια των WSDL διεπαφών (τύπους θυρών και λειτουργιών). Καμιά αναφορά δε γίνεται στις πραγματικές υπηρεσίες που χρησιμοποιούνται από το συγκεκριμένο στιγμιότυπο της διαδικασίας. Μια BPEL4WS διαδικασία είναι ένας επαναχρησιμοποιούμενος ορισμός που μπορεί να υλοποιηθεί με πολλούς τρόπους και πολλά σενάρια, ενώ διατηρεί μια ενιαία συμπεριφορά στο επίπεδο εφαρμογής σε καθένα από αυτά τα σενάρια. Αξίζει να σημειώσουμε ότι η περιγραφή της υλοποίησης μιας BPEL4WS διαδικασίας ξεφεύγει από τους σκοπούς του συγκεκριμένου κεφαλαίου.

2.1.6 *BPML*

Η BPML έχει παρόμοια λογική με την BPEL4WS. Το σημείο στο οποίο διαφέρουν κυρίως είναι ο τρόπος υλοποίησης.

Σκοπός λοιπόν της BPML είναι να ορίσει ένα αφαιρετικό μοντέλο και ένα συντακτικό βασισμένο σε XML προκειμένου να περιγράψει εκτελέσιμες επιχειρησιακές διαδικασίες και τις επιμέρους οντότητες που αυτές υποστηρίζουν. Η ίδια η BPML δεν ορίζει κάποια ειδική εφαρμογή που θα αποτελείται από επιχειρησιακές διαδικασίες αλλά περιορίζεται μόνο στη περιγραφή τους. Πιο συγκεκριμένα ορίζει το αφαιρετικό μοντέλο και τη γραμματική που απαιτείται για να περιγράψει κανείς μια εκτελέσιμη επιχειρησιακή διαδικασία. Το γεγονός αυτό δίνει στη BPML τη δυνατότητα να χρησιμοποιείται για πολλούς σκοπούς πέραν της περιγραφής απλών επιχειρησιακών διαδικασιών, όπως σύνθετες δικτυακές υπηρεσίες(επιχειρησιακές διαδικασίες αποτελούμενες από δικτυακές υπηρεσίες) και διαδικασίες που περιλαμβάνουν την συνεργασία πολλών διαφορετικών και ανεξάρτητα μεταξύ τους μερών.

Η BPML στηρίζει το πρότυπό της σε μια σειρά άλλων προτύπων. Τα πρότυπα αυτά αναφέρονται παρακάτω.

- XML 1.0
- XML – namespaces
- XML- schema
- Xpath 1.0
- WSDL 1.1

Η BPML προβλέπει για την υλοποίησή της τους εξής τύπους οντοτήτων:

- BPML constructs: Είναι οι βασικές κατασκευαστικές μονάδες που περιγράφουν μια διαδικασία σε BPML. Οι μονάδες αυτές περιγράφονται με την βοήθεια ειδικής γραμματικής στηριγμένη σε XML.
- BPML definitions: Είναι μια συλλογή από τα ονόματα των BPML constructs που χρησιμοποιούνται για την περιγραφή της συγκεκριμένης διαδικασίας. Τα ίδια αποτελούν και αυτά ένα BPML construct.
- BPML package: Είναι μια συλλογή από BPML definitions που μπορεί να περιέχουν και definitions από άλλες γλώσσες όπως XML schema ή WSDL 1.1
- BPML documents: Είναι XML αναπαραστάσεις ενός BPML package βασισμένα στο συντακτικό που ορίζει η ίδια η BPML.

Όπως είδαμε η BPML μοιάζει πολύ με την BPEL4WS τόσο στη σχεδίαση όσο και στην υλοποίηση. Δεν κρίνεται σκόπιμο να εμβαθύνουμε περισσότερο στις δυο γλώσσες. Αυτό όμως που επιβάλλεται να γίνει είναι μια προσπάθεια σύγκρισής τους.

2.1.7 *Σύγκριση των δύο γλωσσών*

Ο ανταγωνισμός μεταξύ των δύο γλωσσών είναι πολύ μεγάλος αφού όλες οι μεγάλες εταιρίες στο χώρο της ανάπτυξης λογισμικού ενδιαφέρονται για την ενιαία περιγραφή των επιχειρησιακών διαδικασιών και έχουν πολωθεί προς τη μια ή την άλλη κατεύθυνση. Εμείς θα προσπαθήσουμε μια σύγκριση μεταξύ των όσο το δυνατό πιο αντικειμενικά και αμερόληπτα.

Προκειμένου να πετύχουμε κάτι τέτοιο θα απαριθμήσουμε ονομαστικά κάποια κοινά χαρακτηριστικά και διαφορές που θεωρούνται αποδεκτά και από τις δύο πλευρές. Τα χαρακτηριστικά αυτά είναι τα εξής

- Η BPML περιλαμβάνει πιο γενικευμένες οντότητες από την BPEL4WS και μπορεί γενικώς να περιγράψει περισσότερες μορφές επιχειρησιακών διαδικασιών.
- Και οι δύο γλώσσες χρησιμοποιούν τα ίδια ιδιώματα και παρόμοιο συντακτικό.
- Και οι δύο έχουν την ικανότητα να περιγράφουν πλήρως σύνθετες όσο και απλές επιχειρησιακές διαδικασίες.
- Και οι δύο γλώσσες στηρίζουν το συντακτικό τους σε επιμέρους προγραμματιστικά σύνολα το καθένα από τα οποία περιγράφει όμοιες οντότητες.
- Και οι δύο γλώσσες παρέχουν δυνατότητες για επαναληπτικές, συγχρονισμένες και δυναμικές διαδικασίες.
- Ενώ η BPML περιγράφει πιο γενικευμένες οντότητες από την BPEL4WS, η BPEL4WS περιγράφει με πιο πλήρη τρόπο διαδικασίες που περιλαμβάνουν δικτυακές υπηρεσίες.

2.1.8 Συμπεράσματα - Επιλογή Συστήματος υλοποίησης

Παρατηρούμε από όσα αναγράφονται από τα παραπάνω ότι οι δυο γλώσσες δεν έχουν ουσιαστικές διαφορές. Εφόσον όμως το σύστημα μας απαιτεί εκτενή χρήση των δικτυακών υπηρεσιών επιλέχθηκε τελικά η BPEL4WS. Αυτό έγινε διότι η συγκεκριμένη γλώσσα εμφανίζεται να είναι πιο πλήρης όσον αφορά την περιγραφή επιχειρησιακών διαδικασιών που χρησιμοποιούν δικτυακές υπηρεσίες. Αυτό δεν σημαίνει ότι θεωρούμε την BPEL λιγότερο ικανή γλώσσα. Αντίθετα είναι πιο γενικευμένη και έχει δυνατότητες περιγραφής διαδικασιών που η BPEL4Ws δεν διαθέτει.

2.2 Κέντρα Εξυπηρέτησης Πολιτών

Η ελληνική δημόσια διοίκηση έχει αναπτύξει ένα δίκτυο από πολύπλοκες, δαιδαλώδεις και αρκετές φορές αντιφατικές διαδικασίες, οι οποίες λειτουργούν ανασχετικά ως προς την παραγωγικότητα και αποτελεσματικότητα των υπηρεσιών και ταλαιπωρούν το συναλλασσόμενο πολίτη με μεγάλες χρονοτριβές και πιθανόν οικονομικές επιβαρύνσεις, δημιουργώντας ταυτόχρονα κρίση εμπιστοσύνης στο κράτος .

Ωστόσο, η αναπόδραστη ανάγκη της βέλτιστης διαχείρισης των ανεπαρκών διαθέσιμων οικονομικών πόρων από τα σύγχρονα κράτη, οδηγεί στη μετάβαση από μια δημόσια διοίκηση που είχε ως κυρίαρχο στόχο τη λήψη αποφάσεων και τον έλεγχο εφαρμογής τους, σε μια σύγχρονη, ανταποκριτική δημόσια διοίκηση, η οργάνωση της οποίας έχει ως κύριο στόχο τη διάθεση ολοκληρωμένων υπηρεσιών, έτσι ώστε ο πολίτης να εξυπηρετείται εύκολα και γρήγορα .

Στο πλαίσιο αυτό και για να ελεγχθεί από τους πολίτες / χρήστες η συνθετότητα του δικτύου των δημοσίων υπηρεσιών-φορέων, προκειμένου να διεκπεραιώσουν τις καθημερινές ατομικές τους υποθέσεις και να περιορίσουν τις διαδρομές από γραφείο σε γραφείο, απαιτείται η εφαρμογή διαφόρων τεχνικών, όπως:

Οι υπηρεσίες περιορισμού των διαδρομών (μιας στάσης, one stop shop ή one step service ή guichet unique).

Η αναζήτηση ορισμένων δικαιολογητικών από την αρμόδια υπηρεσία ("οίκοθεν").

Η κατάρτιση φακέλου διοικητικής διαδικασίας, με όλα τα σχετικά στοιχεία / έντυπα που σχετίζονται με την εν λόγω διαδικασία.

Τρόπος λειτουργίας των ΚΕΠ

Είναι υπηρεσιακές μονάδες που έχουν ως σκοπό να περιορίσουν τις μετακινήσεις των συναλλασσομένων με τη διοίκηση πολιτών, από γραφείο σε γραφείο και από υπηρεσία σε υπηρεσία. Οι υπηρεσίες περιορισμού των διαδρομών μπορεί να επιλαμβάνονται υποθέσεων από τη υποβολή του σχετικού αιτήματος (αίτησης) μέχρι την ικανοποίησή του, για τις οποίες υποθέσεις συναρμόδιες είναι υπηρεσιακές μονάδες (Δ/νσεις, Τμήματα) ενός φορέα (π.χ. Δήμος, Νομαρχία) ή δικτύου φορέων (π.χ. ένα Υπουργείο, μία Νομαρχία, ένα Ν.Π.Δ.Δ.).

Αρμοδιότητες των ΚΕΠ

Οι μορφές που μπορεί να λάβει μια υπηρεσία Περιορισμού των Διαδρομών είναι οι παρακάτω:

- **ΕΞΥΠΗΡΕΤΗΣΗ ΑΠΟ ΕΝΑ ΚΤΙΡΙΟ** (Κέντρο δημοσίων Υπηρεσιών).
Σ' ένα κτίριο συστεγάζονται πολλές διαφορετικές δημόσιες υπηρεσίες. Ο πολίτης με μία "στάση" στο κέντρο των δημοσίων υπηρεσιών μπορεί να απευθυνθεί σε πολλές υπηρεσίες.
- **ΕΞΥΠΗΡΕΤΗΣΗ ΑΠΟ ΕΝΑ ΣΗΜΕΙΟ**
Ο πολίτης εξυπηρετείται πλήρως από ένα μόνο σημείο (π.χ. μία υπηρεσία, ένα γραφείο, ένα τηλεφωνικό κέντρο, κ.λ.π.). Αντί ο πολίτης να απευθύνεται σε διαφορετικές υπηρεσίες για διαφορετικές πληροφορίες (π.χ. για άδειες οδήγησης, διαβατήρια, κοινωνικά βοηθήματα) απευθύνεται σε ένα σημείο από το οποίο μπορεί να λαμβάνει πληροφορίες που αφορούν πολλές υπηρεσίες.
- **ΕΞΥΠΗΡΕΤΗΣΗ ΑΠΟ ΕΝΑ ΠΡΟΣΩΠΟ** (Επαφή με ένα πρόσωπο)
Ο πολίτης συνεργάζεται με έναν μόνο υπάλληλο (ένα πρόσωπο επαφής), ο οποίος μεριμνά για την ολοκλήρωση της διαδικασίας, όταν το αιτούμενο "διοικητικό προϊόν" ολοκληρώνεται τμηματικά με μια σειρά από διαδοχικές ενέργειες που γίνονται στις διάφορες οργανικές μονάδες και ο πολίτης είναι υποχρεωμένος να απευθύνεται διαδοχικά σε διαφορετικά τμήματα και να συνεργάζεται με διαφορετικούς υπαλλήλους μέχρι να ολοκληρωθεί όλη η διαδικασία.
- **ΕΞΥΠΗΡΕΤΗΣΗ ΚΑΤΑ ΟΜΑΔΕΣ**
Μια κατηγορία πολιτών (π.χ. άστεγοι) ομαδοποιείται ανάλογα με τις ανάγκες που έχει και εξυπηρετείται από αντίστοιχες εξειδικευμένες, υπηρεσιακές μονάδες. Δημιουργείται ειδική υπηρεσία που εξυπηρετεί τα μέλη της ομάδας (π.χ. υπηρεσία αστέγων, που εξυπηρετεί για όλα τα θέματα τους αστέγους).
- **ΕΞΥΠΗΡΕΤΗΣΗ ΑΠΟ ΚΑΘΕ ΣΗΜΕΙΟ** (Κάθε γραφείο το σωστό γραφείο)
Ο πολίτης εξυπηρετείται από το πλησιέστερο σημείο (γραφείο-γκισέ, θυρίδα-γκισέ) για όλες τις ζητούμενες υπηρεσίες, αντί να είναι αναγκασμένος να αναζητά κάθε φορά το "σωστό" υπάλληλο ή το "σωστό" γραφείο. Η εφαρμογή αυτή ευνοείται από την αποκέντρωση αρμοδιοτήτων σε μονάδες που είναι χωροθετημένες σε διάφορα σημεία της γεωγραφικής έκτασης ευθύνης της περιφέρειας, του νομού ή του δήμου.
- **ΕΞΥΠΗΡΕΤΗΣΗ ΜΕΣΩ ΣΥΝΕΡΓΑΣΙΑΣ ΔΗΜΟΣΙΩΝ ΥΠΗΡΕΣΙΩΝ**
Ο πολίτης με μία επίσκεψη σε μία υπηρεσία μπορεί να απολαμβάνει και τα διοικητικά προϊόντα της άλλης υπηρεσίας, αφού οι υπηρεσίες αυτές έχουν συμφωνήσει μεταξύ τους,

ώστε η καθεμιά, εκτός από τους δικούς της "πελάτες", εξυπηρετεί και τους "πελάτες" της άλλης . Η μορφή αυτή εφαρμόζεται κυρίως σε απομακρυσμένες περιοχές, όπου η παροχή υπηρεσιών είναι δύσκολη ή έχει ψηλό κόστος .

2.3 Στόχος

2.3.1 Γενικός προσδιορισμός του στόχου.

Έχοντας ορίσει τόσο τις αρμοδιότητες των Κέντρων Εξυπηρέτησης Πολιτών όσο και τον τρόπο λειτουργίας τους μπορούμε να προσδιορίσουμε πλέον τον τρόπο που θα χρησιμοποιηθούν σε αυτή την διπλωματική ως μέσο για την μελέτη των business processes που χρησιμοποιούν δικτυακές υπηρεσίες

Κάθε υπουργείο ή γενικά κάθε δημόσιος παροχέας υπηρεσιών, οι υπηρεσίες του οποίου αποτελούν μέρος των διαδικασιών που υλοποιεί το ΚΕΠ, θα μπορούσαν, προς όφελος του πολίτη, να παρέχονται και μέσω του διαδικτύου διευκολύνοντας έτσι με προφανές τρόπο τους πολίτες που τις χρησιμοποιούν. Προκειμένου να υλοποιηθεί κάτι τέτοιο θα αρκούσε απλά μια δικτυακή φόρμα για κάθε υπηρεσία. Έτσι ακόμα και αν κάποιος πολίτης δεν είχε πρόσβαση στο διαδίκτυο θα μπορούσε να επισκεφθεί ένα Κέντρο Εξυπηρέτησης Πολιτών και από εκεί πάλι μέσω διαδικτύου να εξυπηρετηθεί σε πολύ μικρό χρονικό διάστημα.

Η παροχή όμως υπηρεσιών μέσω διαδικτύου θα μπορούσε να βελτιώσει δραστικά και τις διαδικασίες που απαιτούν περισσότερες από μια δημόσιες υπηρεσίες για να υλοποιηθούν. Αυτό μπορεί να γίνει όχι απλά επισπεύδοντας τις επιμέρους διαδικασίες αλλά δημιουργώντας μία ενιαία διαδικασία η οποία θα εκτελείται από το διαδίκτυο αποκρύπτοντας από το χρήστη τις εσωτερικές ενέργειες που πραγματοποιούνται. Μια τέτοια διαδικασία θα δημοσιεύονταν σε ένα ηλεκτρονικό ΚΕΠ και θα αποτελούσε μια μορφή επιχειρησιακής διαδικασίας. Μάλιστα προκειμένου να έχουμε μια απόλυτα αφαιρετική, εύκολα επεκτάσιμη και εύκολη στην χρήση αρχιτεκτονική θα ήταν ιδανικό οι επιμέρους υπηρεσίες να δημοσιεύονταν στην μορφή των δικτυακών υπηρεσιών. Έτσι το ΚΕΠ θα κατασκεύασε επιχειρησιακές διαδικασίες που θα χρησιμοποιούσαν τις παραπάνω δικτυακές υπηρεσίες αναμειγνύοντας τις.

Στην συγκεκριμένη διπλωματική προσπαθήσαμε να σχεδιάσουμε ένα τέτοιο ΚΕΠ. Οι δημόσιες υπηρεσίες είχαν ήδη σχεδιαστεί με την μορφή των δικτυακών υπηρεσιών και άρα ο στόχος ήταν η υλοποίηση των επιχειρησιακών διαδικασιών που με τη σειρά τους θα υλοποιούσαν τις όποιες διαδικασίες που τις χρησιμοποιούσαν.

Σαν γενικότερη ιδέα, όπως θα δείξουμε παρακάτω αυτό θα μπορούσε να εφαρμοστεί όχι μόνο στο ΚΕΠ που χρησιμοποιεί δημόσιες δικτυακές υπηρεσίες, αλλά οπουδήποτε χρησιμοποιούνται δικτυακές υπηρεσίες με συγκεκριμένη σειρά προκειμένου να ικανοποιηθούν ένα συγκεκριμένο αίτημα του πελάτη / χρήστη.

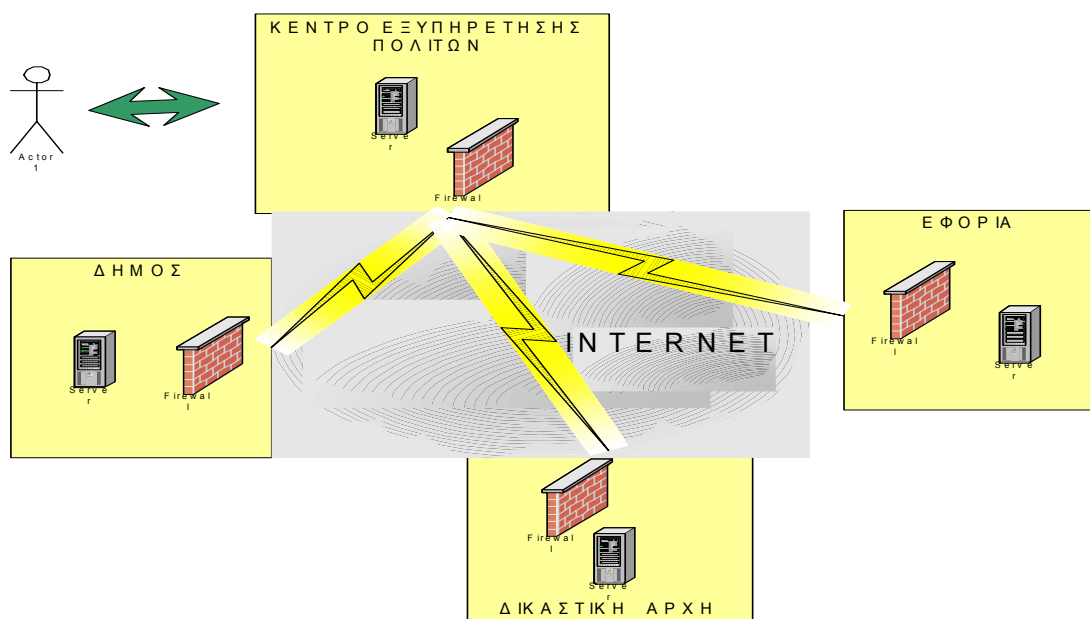
2.3.2 Υλοποίηση των επιμέρους υπηρεσιών

Πριν προχωρήσουμε σε περισσότερες λεπτομέρειες καλό θα ήταν να αναφέρουμε όσον αφορά την υλοποίηση των δημοσίων υπηρεσιών, πως αυτή πραγματοποιήθηκε σε μια άλλη διπλωματική εργασία, η οποία δεν μας είναι ακόμα διαθέσιμη πλήρως. Επομένως στην συγκεκριμένη περίπτωση οι δικτυακές υπηρεσίες που χρησιμοποιούνται είναι εικονικές και υπεραπλουστευμένες. Αυτό όμως δεν αποτελεί πρόβλημα αφού σε μια πραγματική

περίπτωση το μόνο που θα άλλαζε είναι η υλοποίηση των διαδικασιών που είναι όμως ανεξάρτητες του συστήματος.

2.3.3 Στοιχεία αρχιτεκτονικής

Από τα παραπάνω εύκολα μπορεί να συμπεράνει κανείς πως η αρχιτεκτονική του ΚΕΠ θα είναι κάπως έτσι:



Το ΚΕΠ θα πρέπει να παρέχει στον πολίτη τις δυνατότητες να επικοινωνεί μεμονωμένα με το κάθε υπουργείο και να χρησιμοποιεί μέσα από κατάλληλες διεπαφές τις υπηρεσίες του αλλά κυρίως θα πρέπει να δίνει στον χρήστη την επιλογή να εκτελέσει τις οποίες αποθηκευμένες διαδικασίες που υπάρχουν σε αυτό. Οι διαδικασίες αυτές αποτελούν πολύπλοκες διακυβερνητικές διαδικασίες που ακόμα και στην περίπτωση που όλοι οι εμπλεκόμενες υπηρεσίες μπορούσαν να προσεγγιστούν ηλεκτρονικά θα απαιτούσαν μια πολύπλοκη για τον χρήστη διαδικασία που θα έπρεπε να ακολουθηθεί, δυσκολεύοντας έτσι τον απώτερο στόχο του ΚΕΠ, που είναι φυσικά η διευκόλυνση του πολίτη.

Προκειμένου λοιπόν να πετύχουμε την σωστή λειτουργία του ΚΕΠ θα πρέπει να υπάρχει ένας μηχανισμός που να εκτελεί τις αποθηκευμένες διαδικασίες. Ο μηχανισμός αυτός, προκειμένου να είναι εύκολα διαχειρίσιμος, θα πρέπει να είναι ανεξάρτητος των επιμέρους διαδικασιών. Επίσης είναι προφανές πως ένας διαχωρισμός των χρηστών του ΚΕΠ είναι απαραίτητος. Έτσι θα υπάρχουν οι κατάλληλοι υπάλληλοι που θα διαχειρίζονται το ΚΕΠ προσθέτοντας και αφαιρώντας διαδικασίες από αυτό, και οι κυρίως χρήστες (οι απλοί πολίτες) που θα μπορούν απλά να επιλέγουν μια διαδικασία από τις ήδη υπάρχουσες και να την εκτελούν.

3

Αρχικές προσπάθειες

Υλοποίησης

3.1 Εισαγωγικά

Στην προσπάθειά μας να υλοποιήσουμε το σύστημά που θέσαμε σαν στόχο συναντήσαμε πολλά εμπόδια και δυσκολίες. Αυτό είχε σαν αποτέλεσμα να οδηγηθούμε σε πολλές αποτυχημένες υλοποιήσεις. Η αιτία των προβλημάτων και συνεπώς και των αποτυχημένων προσπαθειών υλοποίησης ήταν το γεγονός ότι η γλώσσες περιγραφής επιχειρησιακών διαδικασιών στις οποίες αναφερθήκαμε παραπάνω βρίσκονται ακόμα σε πειραματικό στάδιο και δεν παρέχουν πλήρη υποστήριξη σε πλατφόρμες και εργαλεία που προσπαθήσαμε να υλοποιήσουμε. Η έλλειψη εμπειρίας έκανε τα προβλήματα που εμφανίζονταν ανά περιόδους ακόμα πιο δύσκολη.

Έτσι πριν φθάσουμε τελικά στην τελική υλοποίηση περάσαμε από διάφορες αποτυχημένες προσπάθειες σχεδίασης. Οι προσπάθειες αυτές παρόλα αυτά δεν υπήρξαν απόλυτα άκαρπες αφού μας οδήγησαν σε χρήσιμα συμπεράσματα. Παρακάτω θα περιγράψουμε σύντομα τις υλοποιήσεις αυτές και στο τέλος θα παραθέσουμε τα χρήσιμα συμπεράσματα που αποκομίσαμε από αυτές.

3.2 Προσπάθεια 1^η

Η αρχική ιδέα σχεδιασμού και υλοποίησης του ΚΕΠ προέβλεπε οι επιμέρους διαδικασίες να είναι ενσωματωμένες στη μηχανή υλοποίησής τους. Για να είμαστε πιο ακριβής το ΚΕΠ θα χωρίζονταν αρχιτεκτονικά σε δύο τμήματα.

Το πρώτο τμήμα θα αποτελούσε τον κύριο μηχανισμό του ΚΕΠ. Οι επιμέρους διαδικασίες θα αποτελούσαν μέρος του κυρίως προγράμματος όπως και η διεπαφή μέσω της οποίας θα εκτελούνταν. Οι διαδικασίες θα αναπαριστάνω λοιπόν μέσω αντικειμένων. Τα αντικείμενα αυτά θα περιγράφονταν μέσω κατάλληλων κλάσεων, η κατασκευάστρια συνάρτηση των

οποίων θα εκτελούσε την αντίστοιχη διαδικασία. Έτσι όταν ο χρήστης μέσω της διεπαφής καλούσε μία διαδικασία, ο κώδικας θα δημιουργούσε το κατάλληλο στιγμιότυπο της διαδικασίας που επελέγη. Η δημιουργία και μόνο του στιγμιότυπου αρκούσε για να εκτελεστεί η διαδικασία και να οδηγηθεί η διεπαφή του χρήστη στην κατάλληλη φόρμα επίδειξης του αποτελέσματος.

Το δεύτερο τμήμα θα αποτελούνταν από μια γραφική απεικόνιση της διαδικασίας η οποία θα παρεμβάλλονταν της εκτέλεσής της. Η γραφική απεικόνιση θα έδινε στο χρήστη τη δυνατότητα να παρακολουθεί την διαδικασία και συνεπώς να ελέγχει πιθανά δικά του και όχι μόνο λάθη σε αυτή. Έτσι οποιοδήποτε λάθος στην διαδικασία θα μπορούσε να διορθωθεί ταχύτερα και με μεγαλύτερη αποτελεσματικότητα.

Το δεύτερο τμήμα του ΚΕΠ είχε και σαν εκπαιδευτικό σκοπό την προσωπική εξοικειώσει με ζητήματα γραφικής απεικόνισης σε δικτυακές εφαρμογές, αντικείμενο ιδιαίτερα ενδιαφέρον και χρήσιμο αφού στις επαγγελματικές εφαρμογές τέτοιες απεικονίσεις χρησιμοποιούνται πολύ συχνά.

Η υλοποίηση του πρώτου τμήματος είχε σχεδιαστεί να πραγματοποιηθεί στο προγραμματιστικό περιβάλλον VISUAL STUDIO.NET και σε γλώσσα VISUAL BASIC όσον αφορά το πρώτο τμήμα. Όσον αφορά το δεύτερο θα γινόταν χρήση της γλώσσας προγραμματισμού JAVA και σε κατάλληλο προγραμματιστικό περιβάλλον που να υποστηρίζεται από το λειτουργικό σύστημα UNIX.

Η υλοποίηση όπως περιγράφεται παραπάνω φαίνεται αρκετά ελκυστική. Σκοπό έχει να τονίσει την ανεξαρτησία των web services από το λειτουργικό σύστημα του χρήστη και του παροχέα υπηρεσίας, όπως και του τρόπου υλοποίησης των.

Η συγκεκριμένη σχεδίαση του ΚΕΠ είχε κάποια πλεονεκτήματα και αρκετά περισσότερα μειονεκτήματα.

Το κυριότερο πλεονέκτημά της είναι η ασφάλεια των παρεχόμενων υπηρεσιών. Δεν είναι απαραίτητος ο διαχωρισμός χρηστών και διαχειριστών αφού οι διαδικασίες προστίθενται και αφαιρούνται προγραμματιστικά. Επίσης η συγκεκριμένη εφαρμογή θα ήταν και αρκετά ταχύς αφού το σύστημα μας θα ήταν ενιαίο. Τέλος ο έλεγχος λαθών από την πλευρά του χρήστη θα ήταν πολύ αποτελεσματικός αφού η διαδικασία μπορεί να ελέγχεται σε κάθε σημείο της.

Το μεγαλύτερο μειονέκτημα που παρουσιάζει ο εν λόγω σχεδιασμός είναι βέβαια η ανάγκη για συνεχή προγραμματιστική παρέμβαση. Αυτό συνεπάγεται μεγάλες δυσκολίες στην αναπροσαρμογή του συστήματος σε ενδεχόμενες αλλαγές στη διαδικασία αλλά και ανάγκη για εξαιρετικά ειδικευμένο προσωπικό που θα ήξερε τον κώδικα της εφαρμογής αρκετά καλά ώστε να μπορεί να τον αλλάζει εύκολα και γρήγορα. Θα είχαμε λοιπόν το ίδιο πρόβλημα που έχουν όλες οι εταιρίες ανά τον κόσμο που ασχολούνται με ηλεκτρονικές επιχειρήσεις και όχι μόνο, την ανάγκη ανεξαρτησίας των επιχειρησιακών διαδικασιών από το πρόγραμμα που τις εκτελεί. Είναι η ανάγκη εκείνη που γέννησε τις γλώσσες περιγραφής των εν λόγω διαδικασιών και με τις οποίες ασχοληθήκαμε στην συνέχεια.

Ύστερα από κατάλληλη προτροπή του επιβλέποντος της διπλωματικής Δρ Ντίνου Αρκουμάνη, αποφασίσαμε να μελετήσουμε τις γλώσσες περιγραφής των διαδικασιών προκειμένου να σχεδιαστεί ένα σύστημα ευέλικτο και εύκολα ανανεώσιμο όντας ανεξάρτητο των διαδικασιών που περιέχει.

3.3 Προσπάθεια 2^η

Το σύστημα λοιπόν ξανασχεδιάστηκε, αυτή τη φορά με την απαίτηση της ανεξαρτησίας του από τις διαδικασίες που χρησιμοποιούσε. Προκειμένου να περιγράψουν οι διαδικασίες με τρόπο ανεξάρτητο επελέγη η γλώσσα BPEL4WS. Η κύρια διεπαφή σχεδιάστηκε και πάλι σε περιβάλλον VISUAL STUDIO.NET και με γλώσσα VISUAL BASIC. Το δεύτερο μέρος από την πρώτη υλοποίηση αποφασίστηκε να μην υλοποιηθεί.

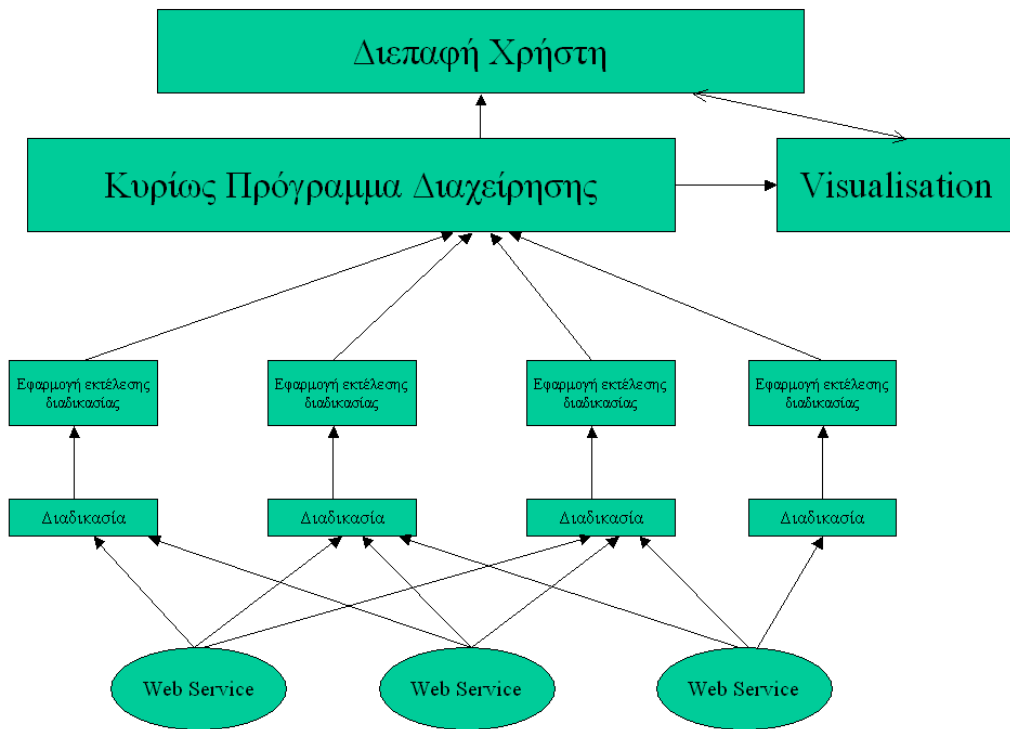
Στην συνέχεια έπρεπε να αποφασισθεί για τον τρόπο σχεδίασης αν για κάθε διαδικασία θα δημιουργούσαμε ξεχωριστή εφαρμογή που να την εκτελεί ή κάποια νέα εφαρμογή που θα εντάσσονταν στο κώδικα τις διεπαφές θα δημιουργούνταν με σκοπό να εκτελεί οποιαδήποτε διαδικασία με παραμέτρους που θα εισάγονταν από το χρήστη ή θα ήταν ήδη αποθηκευμένες στο σύστημα σε κάποια βάση.

Ο δεύτερος τρόπος αν και πραγματοποιούσε πλήρως την επιθυμητή αφαίρεση απορρίφθηκε νωρίς. Η αιτία ήταν πως δεν υπήρχαν έτοιμες βιβλιοθήκες για το συγκεκριμένο περιβάλλον υλοποίησης που θα μας βοηθούσαν να εντάξουμε στο σύστημά μας μια διαδικασία που θα εκτελούσε κάθε διαδικασία περιγραφόμενη σε BPEL4WS δεδομένων των ξεχωριστών παραμέτρων της. Προκειμένου να υλοποιηθεί κάτι τέτοιο θα έπρεπε πρώτα να υλοποιήσουμε ένα μεταφραστή που θα δέχονταν σαν είσοδο μια διαδικασία γραμμένη σε BPEL4WS και σαν έξοδο θα είχε κάποια συνάρτηση σε VISUAL BASIC που θα την εκτελούσε δεδομένων των παραμέτρων που είχε θέσει ο σχεδιαστής της. Κάτι τέτοιο ξέφευγε από τους σκοπούς της διπλωματικής. Επιπλέον η έρευνά μας στο συγκεκριμένο χώρο θα ήταν πολύ σύντομα ξεπερασμένη από τις εξελίξεις στο χώρο.

Έτσι επιλέχθηκε να σχεδιαστεί η εφαρμογή μας που να εκτελεί εφαρμογές οι οποίες με την σειρά τους θα εκτελούσαν τις εν λόγω διαδικασίες. Μπορεί σε πρώτη φάση αυτό να φαίνεται λίγο τραβηγμένο αλλά με δεδομένες τις εξελίξεις που είχαν ήδη ανακοινωθεί, σε πολύ λίγο καιρό η εφαρμογή μας θα μπορούσε εύκολα να αναβαθμιστεί παρακάμπτοντας το προφανώς άχρηστο στάδιο εκτέλεσης. Άλλωστε και με το συγκεκριμένο τρόπο σχεδιασμού η εφαρμογή μας ήταν ανεξάρτητη των διαδικασιών. Απλά αντί να αποθηκεύει και να διαχειρίζεται διαδικασίες αποθήκευσε και διαχειρίζονταν εφαρμογές που εκτελούσαν τις διαδικασίες που ζητάμε.

Ο πιθανός διαχειριστής του συστήματος εκτός από τις διαδικασίες θα έπρεπε να σχεδιάζει ταυτόχρονα και τις εφαρμογές που τις χρησιμοποιούν. Ο συγκεκριμένος σχεδιασμός όμως μπορούσε να διευκολυνθεί πολύ με τα ήδη υπάρχοντα εργαλεία που οι διαχειριστές της BPEL4WS μας παρείχαν. Οι εφαρμογές αυτές θα υλοποιούνταν σε γλώσσα JAVA και με την βοήθεια του εργαλείου APACHE SOAP. Φυσικά στην συγκεκριμένη ανάλυση υλοποίησης παρακάμπτουμε τον κυρίως τρόπο σχεδίασης των διαδικασιών σε BPEL4WS ο οποίος είναι ίδιος με αυτόν που χρησιμοποιήθηκε στο τελικό σχεδιασμό του συστήματος και θα αναλυθεί παρακάτω.

Παρακάτω σχεδιάζουμε μια σχηματική αναπαράσταση του συστήματος όπως αυτό σχεδιάστηκε κατά τον δεύτερο σχεδιασμό.



Τα προβλήματα που εμφανίστηκαν ήταν πολλά. Τα κυριότερα απαριθμούνται παρακάτω.

Εφόσον η εφαρμογή ήταν δικτυακή δεν μπορούσαμε να εκτελέσουμε εφαρμογές του λειτουργικού συστήματος εξαιτίας των περιορισμών που επιβάλλει ο κάθε server για προφανείς λόγους. Οι συγκεκριμένοι περιορισμοί μπορούσαν να παρακαμφθούν αλλά κάτι τέτοιο θα έκανε το σύστημά μας ευάλωτο όσον αφορά την ασφάλειά του. Επομένως η εφαρμογή μας δεν θα ήταν λειτουργική και η υλοποίηση ενός τέτοιου συστήματος δεν θα είχε νόημα.

Επιπλέον η εκτέλεση μιας εφαρμογής από μέσω του λειτουργικού συστήματος θα έκανε την κυρίως δικτυακή εφαρμογή πολύ αργή και θα είχαμε πρόβλημα με τον καθορισμένο χρόνο που επιβάλλει ο κάθε server για την φόρτωση μιας σελίδας από τον χρήστη. Ο εκάστοτε διαχειριστής του συστήματος θα έπρεπε να αλλάξει τον χρόνο αυτό επιβάλλοντας στο σύστημα να περιμένει σε ενδεχόμενη καθυστέρηση που οφείλεται στους χρήστες και με τον τρόπο αυτό να προκαλέσει επιπλοκές στη ίδια του τη λειτουργία.

Στην προσπάθειά μας να λύσουμε το συγκεκριμένο πρόβλημα οδηγηθήκαμε σε μια λύση η οποία θα μπορούσε να εφαρμοστεί γενικά σε οποιοδήποτε σύστημα αντιμετωπίζει πρόβλημα συγχρονισμού ιστοσελίδων. Εφαρμόζοντας την τεχνική του multithreading μπορούμε να οδηγήσουμε τον χρήστη σε μια κενή ιστοσελίδα που απλά θα περιέχει ένα μήνυμα αναμονής. Εν τω μεταξύ δημιουργούμε ένα καινούργιο thread το οποίο εκτελεί την χρονοβόρα διαδικασία και θέτουμε το κυρίως thread σε αναμονή έως ότου να τελειώσει το δευτερεύον, ή απλά ορίζουμε πως το σύστημα πρέπει να περιμένει όλα τα threads προκειμένου να σταματήσει. Μέσα στην κύρια διαδικασία του δευτερεύοντος μηνύματος προβλέπουμε την μεταφορά στην κατάλληλη σελίδα μόλις αυτή τελειώσει. Έτσι ο χρήστης έχει την πλήρη αίσθηση του τι συμβαίνει και έχει τη σωστή συμπεριφορά προκειμένου να ολοκληρώσει το σύστημα τη λειτουργία του.

Η παραπάνω τεχνική είναι αρκετά ενδιαφέρουσα και κάνει το σύστημα πολύ ευέλικτο όταν έχει να εκτελέσει χρονοβόρες διαδικασίες, διατηρώντας ταυτόχρονα όλα τα πλεονεκτήματα του μικρού παραθύρου αναμονής για την φόρτωση μιας σελίδας.

Συνεχίζοντας την προσπάθειά μας προκειμένου να εκτελέσουμε εφαρμογές από το λειτουργικό σύστημα διατηρώντας ταυτόχρονα τους περιορισμούς ασφαλείας του server δοκιμάσαμε μια αρκετά πρωτότυπη τεχνική. Δημιουργήσαμε μια βάση δεδομένων η οποία περιείχε «εικονικά» πεδία. Δημιουργήσαμε τους κατάλληλους αυτοματισμούς στη βάση έτσι ώστε η αλλαγή των κατάλληλων πεδίων να σημαίνει και ταυτόχρονη εφαρμογή του αντιστοίχου σε κάθε διαδικασία προγράμματός. Εφόσον η ενημέρωση μιας βάσης δεδομένων είναι μια ενέργεια που ο server επιτρέπει το σύστημά μας θα λειτουργούσε χωρίς να επηρεάζονται οι περιορισμοί. Μάλιστα δεδομένου ότι η βάση μπορούσε να πυροδοτήσει μόνο μία εφαρμογή η ασφάλεια επί της ουσίας δεν επηρεάζονταν.

Το κυριότερο πρόβλημα με την συγκεκριμένη τεχνική ήταν ότι η δημιουργία και εκτέλεση των αυτοματισμών επιβάρυνε το σύστημά μας και το έκανε πολύ αργό ακόμα και για τον καλύτερο υπολογιστή να το χρησιμοποιήσει.

Παρόλα αυτά δοκιμάστηκε να τοποθετήσουμε την βάση δεδομένων σε ένα υπολογιστή με όσο το δυνατό μεγαλύτερη υπολογιστική ισχύ. Το γεγονός όμως παρέμενε ότι οι διαδικασίες και η διεπαφή έπρεπε να αναπτυχθούν σε άλλο μηχάνημα μικρότερης υπολογιστικής ισχύος. Αυτή τη φορά ήταν οι περιορισμοί του τοπικού δικτύου που μας επέβαλλαν να μην μπορούμε να εκτελέσουμε τις εφαρμογές μας. Δυστυχώς οι περιορισμοί αυτοί δεν μπορούσαν να αρθούν. Αναγκαστικά λοιπόν οδηγηθήκαμε σε αδιέξοδο.

Τα διαδοχικά αδιέξοδα μας οδήγησαν στο να αλλάξουμε για τρίτη φορά την υλοποίηση. Αυτή τη φορά είχαμε τη δυνατότητα να υλοποιήσουμε το σύστημά μας όπως ακριβώς επιθυμούσαμε χωρίς περιορισμούς. Η αρχιτεκτονική και υλοποίηση που περιγράφεται παρακάτω είναι η καλύτερη που μπορούσε να γίνει με τα έως τώρα δεδομένα. Κλειδί στο να επιτευχθεί το παραπάνω ήταν η αλλαγή του προγραμματιστικού περιβάλλοντος και γλώσσας προγραμματισμού από VISUAL STUDIO.NET και VISUAL BASIC σε JBUILD8 και JAVA.

4

Ανάλυση και σχεδίαση

4.1 Περιγραφή Αρχιτεκτονικής

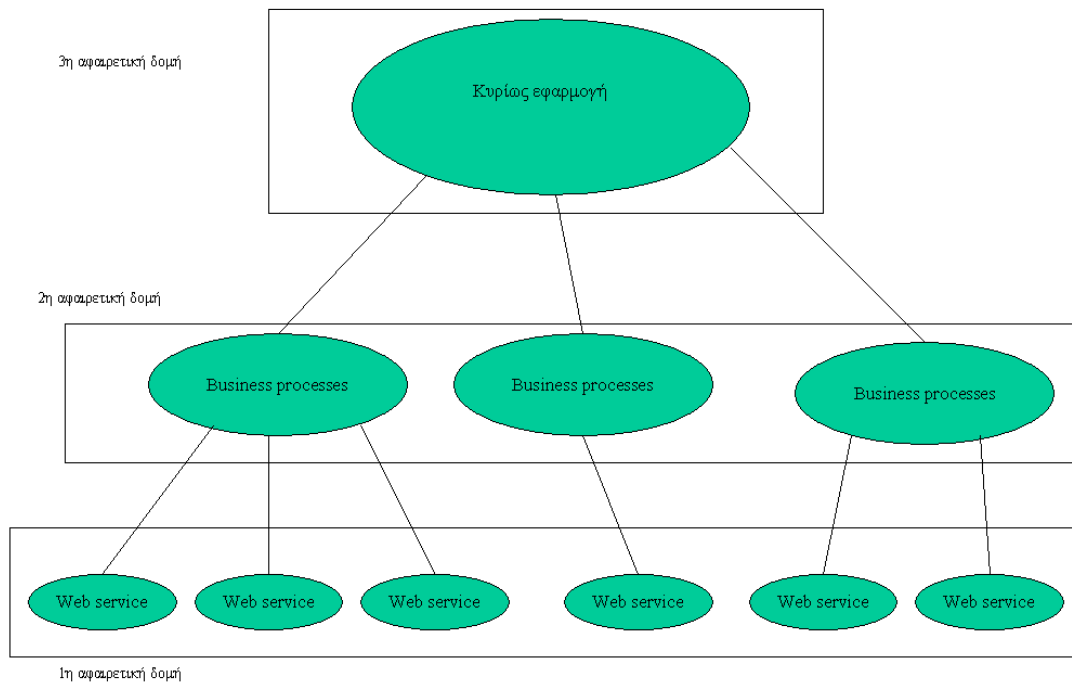
Η τελική αρχιτεκτονική στην οποία στηρίχθηκε το σύστημά μας δεν διέφερε και πολύ από αυτές που περιγράψαμε συνοπτικά στο προηγούμενο κεφάλαιο. Παρόλαυτά στο συγκεκριμένο σημείο είμαστε αναγκασμένοι να την περιγράψουμε αναλυτικά για την καλύτερη κατανόησή της.

Αρχική απαίτηση είναι η δημιουργία ενός συστήματος που θα υλοποιεί τρία επίπεδα αφαίρεσης. Το πρώτο επίπεδο αφαίρεσης αφορά τις επιμέρους υπηρεσίες που χρησιμοποιεί το ΚΕΠ στις διαδικασίες του. Το επίπεδο αυτό αφαίρεσης υλοποιείται με την βοήθεια της τεχνολογίας των δικτυακών υπηρεσιών. Ο κάθε δημόσιος ή ιδιωτικός φορέας που εμπλέκεται στο ΚΕΠ, προσφέρει τις υπηρεσίες του με την μορφή των συνδυαστικών υπηρεσιών που είναι προσβάσιμες μέσω του διαδικτύου από οποιαδήποτε πλατφόρμα και εφαρμογή.

Το δεύτερο επίπεδο αφαίρεσης είναι οι διαδικασίες που χρησιμοποιούν τις δικτυακές υπηρεσίες όπως αναφέρθηκαν παραπάνω. Οι διαδικασίες αυτές περιγράφουν, υλοποιούν τις υπηρεσίες του ΚΕΠ προς τους χρήστες του. Το επίπεδο αυτό αφαίρεσης υλοποιείται μέσω της BPEL4WS. Οι διαδικασίες περιγράφονται σαν BPEL4WS και μπορούν θεωρητικά να γίνουν προσβάσιμες σε κάθε σύστημα και κάθε πλατφόρμα. Οι αποτυχημένες προσπάθειες υλοποίησης πάνω από την πλατφόρμα σχεδίασης λογισμικού MICROSOFT VISUAL STUDIO.NET μας έδειξαν πως η ανεξαρτησία εφαρμογών όπως την περιγράψαμε παραπάνω είναι μόνο θεωρητική. Αυτό οφείλεται μόνο στο γεγονός ότι η BPEL4WS είναι ακόμα σε πειραματικό στάδιο και οι επιμέρους πλατφόρμες σχεδίασης και υλοποίησης λογισμικού δεν έχουν ακόμα υλοποιήσει τα κατάλληλα εργαλεία που χρειάζονται για την εύκολη διαχείριση των διαδικασιών BPEL4WS. Παρ' όλα αυτά οι διαδικασίες BPEL4WS έχουν περιγραφεί με βάση την XML οπότε μπορούν να αυτοπεριγρέφονται και επομένως σε αρχιτεκτονικό επίπεδο θεωρούνται αυτόνομες και ανεξάρτητες πλατφόρμες. Σε επίπεδο υλοποίησης η ανάπτυξη εργαλείων είναι μόνο θέμα χρόνου, όπως ελπίζουμε.

Το τρίτο επίπεδο είναι η κύρια εφαρμογή που εκτελεί τις διαδικασίες του ΚΕΠ. Η εφαρμογή αυτή αποτελείται κυρίως από δύο διεπαφές. Η μία διεπαφή αφορά τους απλούς χρήστες του ΚΕΠ και η άλλη τους διαχειριστές της. Η εσωτερική αρχιτεκτονική και των δύο θα αναλυθεί στη συνέχεια.

Παρακάτω ακολουθεί ένα σχήμα που περιγράφει τη γενική αρχιτεκτονική που περιγράψαμε.



Στη συγκεκριμένη διπλωματική εργασία ασχοληθήκαμε με τα δύο επίπεδα αρχιτεκτονικής που αφορούν τις επιχειρησιακές διαδικασίες και το κυρίως πρόγραμμα. Τα δύο αυτά επίπεδα αρχιτεκτονικής περιγράφονται παρακάτω.

4.1.1 Αρχιτεκτονική Επιχειρησιακών Διαδικασιών

Η αρχιτεκτονική των επιχειρησιακών διαδικασιών δεν έχει κάποιο ιδιαίτερο χαρακτηριστικό. Γενικά θα μπορούσαμε να πούμε πως αποτελείται κυρίως από κάποια περιγραφικά αρχεία απαραίτητα για την υλοποίηση της διαδικασίας.

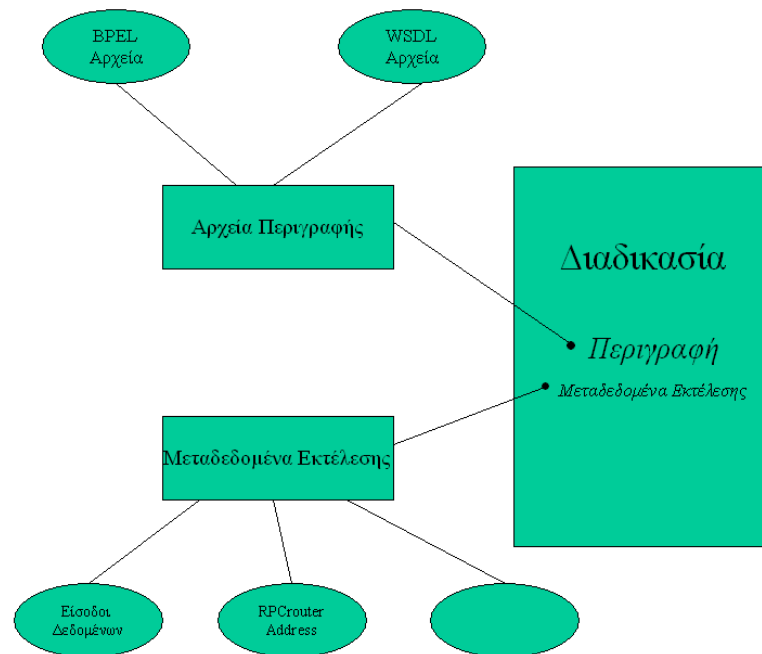
Τα αρχεία αυτά αποθηκεύονται σε κάποιο κατάλογο του διακομιστή που χρησιμοποιείται για να τις αναπτύξουμε. Στον ίδιο διακομιστή πρέπει να αναπτυχθούν και δύο βασικά εργαλεία: Μία εφαρμογή που θα δημιουργεί τα κατάλληλα μεταδεδομένα που αφορούν την διαδικασία και είναι απαραίτητα για την εφαρμογή της και μια εφαρμογή που αναπτύσσεται πάνω στον διακομιστή και εκτελεί το ρόλο του RPCrouter. Η ακριβής διεύθυνση του RPCrouter αποτελεί και αυτή μέρος των μεταδεδομένων που αναφέραμε. Οι εφαρμογές που αναφέρουμε παραπάνω μας παρέχονται δωρεάν από την BPEL4WS και δεν χρειάζονται περισσότερη ανάλυση.

Όσον αφορά τα υπόλοιπα αρχεία, πρόκειται για τα αρχεία περιγραφής της κάθε διαδικασίας και τα WSDL αρχεία που περιγράφουν τις δικτυακές υπηρεσίες που εμπλέκονται. Συνήθως τα αρχεία που περιγράφουν την διαδικασία χωρίζονται σε δύο κατηγορίες: WSDL που ορίζουν τις πηγές, τις διασυνδέσεις, τα μηνύματα επικοινωνίας που ανταλλάσσονται μεταξύ των συνεργατών της διαδικασίας και αυτής και τις θύρες επικοινωνίας τις διαδικασίας με τους χρήστες της, BPEL αρχεία που περιγράφουν τις ροές εκτέλεσης και τις μεθόδους κάθε διαδικασίας.

Τέλος τα μεταδεδομένα που χρειάζονται για την εκτέλεση της κάθε διαδικασίας είναι τα εξής:

- Οι είσοδοι δεδομένων που εισάγει ο χρήστης για κάθε διαδικασία.
- Το όνομα της μεθόδου που εκτελείται
- Η διεύθυνση του RPCrouter.

Γενικά κάθε διαδικασία θα μπορούσε να μοντελοποιηθεί όπως φαίνεται στο σχήμα που ακολουθεί.



4.1.2 Αρχιτεκτονική κεντρικής εφαρμογής

Η αρχιτεκτονική της κυρίας εφαρμογής είναι αρκετά απλή.

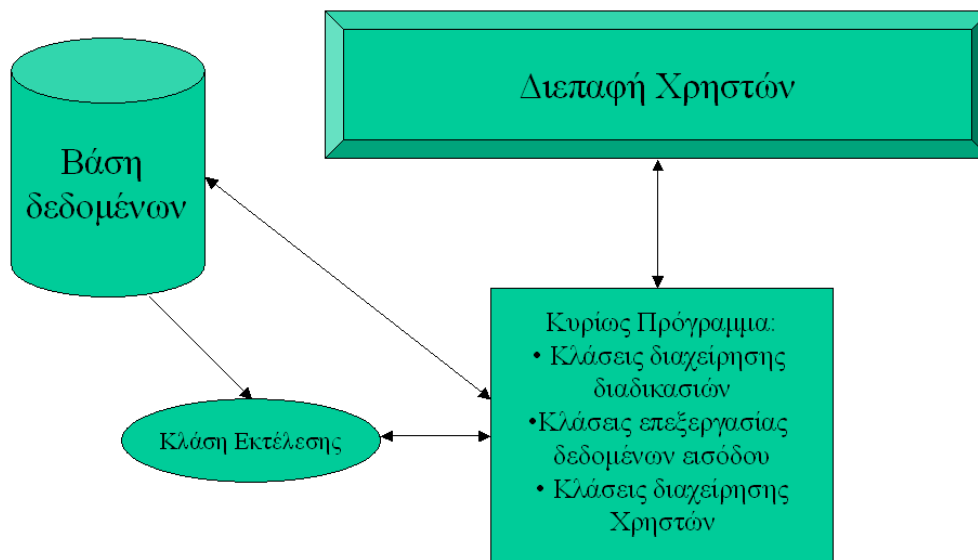
Οι διάφορες διαδικασίες προς εκτέλεση αποθηκεύονται σε μια βάση δεδομένων μαζί με όλα τα απαραίτητα μεταδεδομένα για την εκτέλεσή τους.

Υπάρχει μια εσωτερική διαδικασία η οποία εκτελεί οποια διαδικασία της ζητηθεί παίρνοντας τα απαραίτητα δεδομένα από την βάση και εκτελώντας την πάνω στο διακομιστή όπου έχει δημιουργηθεί.

Υπάρχει το κυρίως πρόγραμμα που παρέχει δυνατότητες προσθήκης, διαγραφής και εκτέλεσης διαδικασιών καθώς και μια διεπαφή που χρησιμεύει για την παρουσίαση των αποτελεσμάτων διαφόρων λειτουργιών καθώς και για την επιλογή της λειτουργίας που θα εκτελεστεί.

Εδώ πρέπει να αναφέρουμε ότι στο ΚΕΠ υπάρχει διαχωρισμός των χρηστών σε διαχειριστές και απλούς χρήστες. Καθένας από αυτούς μπορεί να εκτελέσει μόνο ορισμένες από τις διαδικασίες του συστήματος. Οι επιμέρους λειτουργίες και τα δικαιώματα εκτέλεσης θα αναφερθούν παρακάτω σε επόμενο υποκεφάλαιο.

Στο σχήμα που ακολουθεί φαίνεται μια γραφική αναπαράσταση της αρχιτεκτονικής της κύριας εφαρμογής του ΚΕΠ



4.2 Περιγραφή Λειτουργιών

Όπως περιγράψαμε και στην αρχή του παρόντος κειμένου, το σύστημα που επιδιώκουμε να σχεδιάσουμε απευθύνεται σε δύο ήδη χρηστών: τους απλούς χρήστες του ΚΕΠ και τους διαχειριστές. Καθένας από αυτούς τους χρήστες έχουν διαφορετικές δυνατότητες και αρμοδιότητες. Έτσι οι απλοί χρήστες του ΚΕΠ μπορούν να ενημερωθούν για τις λειτουργίες που τους παρέχονται και να τις χρησιμοποιήσουν. Επίσης έχουν πρόσβαση σε διάφορα υπουργεία και υπηρεσίες σε περίπτωση που χρειαστούν κάποια από τις διαδικασίες που αυτά προσφέρουν ανεξαρτήτως του ΚΕΠ.

Θα πρέπει, για λόγους ασφαλείας να διασφαλίσουμε ότι ένας απλός χρήστης δεν θα μπορεί να επεμβαίνει στο σύστημα αλλάζοντας τις καταχωρήσεις που αφορούν τις διαδικασίες. Δεν θα μπορεί να αφαιρεί διαδικασίες από το σύστημα ούτε να προσθέτει. Πολύ περισσότερο δεν θα μπορεί να αλλάζει τις υπάρχουσες διαδικασίες. Παρ' όλ' αυτά οι παραπάνω λειτουργίες θα πρέπει να μπορούν να υλοποιηθούν αφού το ΚΕΠ είναι μια υπηρεσία δυναμικά εξελισσόμενη και συνεχώς ανανεώσιμη. Για τον λόγο αυτό ορίζουμε και ένα νέο τύπο χρηστών τους διαχειριστές.

Οι διαχειριστές έχουν πολύ περισσότερες αρμοδιότητες. Μπορούν να σχεδιάζουν και να προσθέτουν διαδικασίες από το σύστημα. Μπορούν να διαγράφουν διαδικασίες αν θεωρήσουν πως αυτές δεν χρειάζονται ποια ή αν απλά σκοπεύουν να τις αναβαθμίσουν. Επίσης χρειάζεται να τους παρέχεται η δυνατότητα να διαχειρίζονται τις καταχωρήσεις που αφορούν τους υπόλοιπους διαχειριστές. Μπορούν προσθέτουν διαχειριστές στο σύστημα και

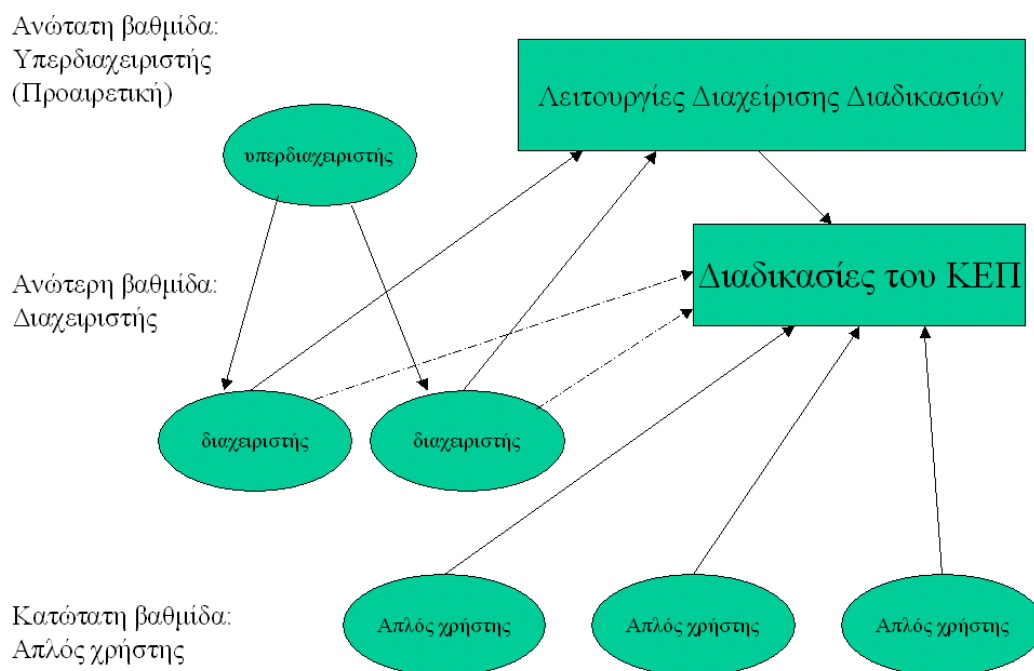
να αφαιρούν τον εαυτό τους. Οι παραπάνω λειτουργίες έχουν άμεση σχέση και με την ασφάλεια του συστήματος.

Κανένας διαχειριστής δεν θα μπορεί να διαγράψει κάποιον άλλον από το σύστημα χωρίς την έγκριση του ιδίου. Επίσης οι ελευθερίες που παρέχονται στους διαχειριστές θα μπορούν να παραχωρούνται μόνο σε άτομα που εγκρίνουν οι ίδιοι αναλαμβάνοντας ταυτόχρονα και την ευθύνη. Ο απόλυτος έλεγχος χρηστών επιτυγχάνεται αν στους χρήστες προστεθεί ένας νέος τύπος αυτός του υπερδιαχειριστή.

Οι αρμοδιότητες ενός υπερδιαχειριστή θα είναι να ελέγχει τους υπόλοιπους διαχειριστές και να μπορεί να τους αποβάλλει από το σύστημα αν αυτός κρίνει σκόπιμο. Αυτό γίνεται εάν έχει πρόσβαση στα ιδιωτικά στοιχεία που προσδιορίζουν κάθε διαχειριστή.

Τέλος αναφέρουμε πως ο κάθε διαχειριστής θα μπορεί να λειτουργεί και σαν χρήστης ενώ και κάθε υπερδιαχειριστής θα μπορεί να λειτουργεί και σαν απλός διαχειριστής.

Παρακάτω ακολουθεί ένα σχήμα που περιγράφει την ιεραρχία που περιγράψαμε παραπάνω.




Ο διαχωρισμός των χρηστών γίνεται στην αρχική σελίδα της εφαρμογής μας

New Page 2 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites Media Print Mail News RSS

Address http://localhost:8083/main.jsp Go Links



ΚΕΝΤΡΟ ΕΞΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ

[Original KEP Page](#)

Process Description


[Loan Application](#)

[Echo](#)

Useful Links

[Ministries](#)

[Other Public Services](#)



Your Options:

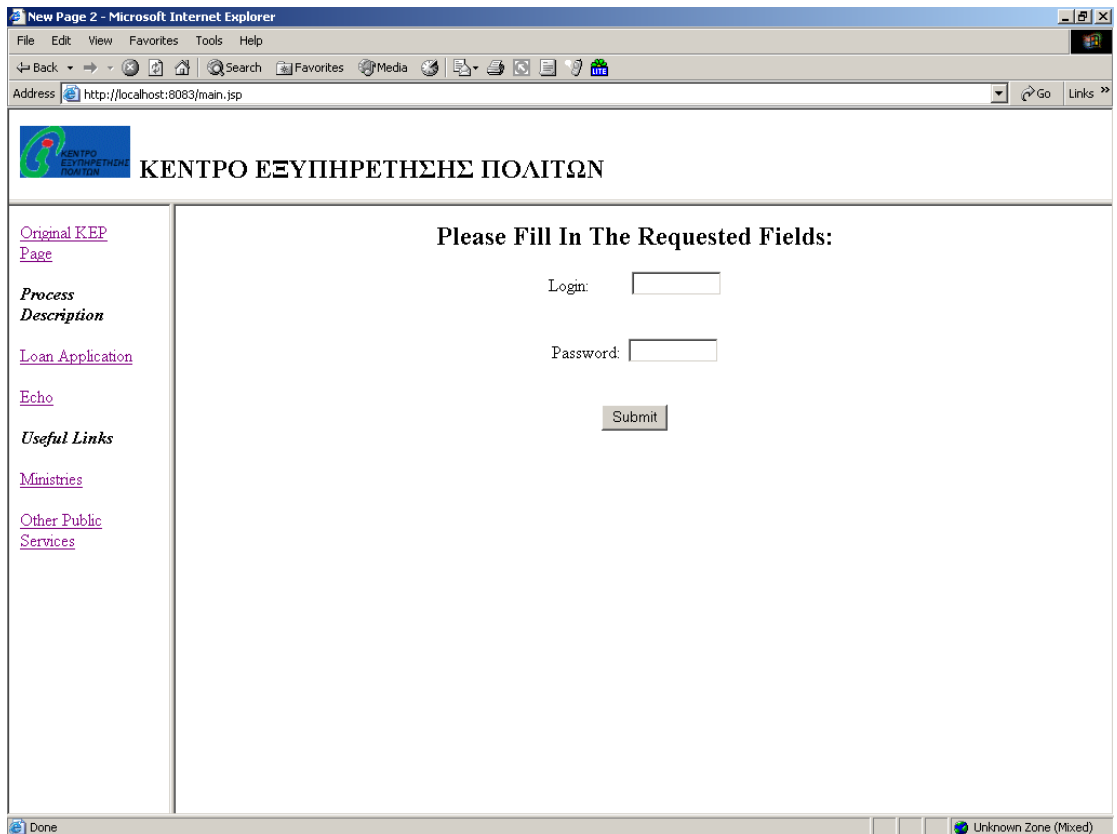
Use The Services Of Kep [click here](#)

Login As An Administrator of Kep [click here](#)

Done Unknown Zone (Mixed)

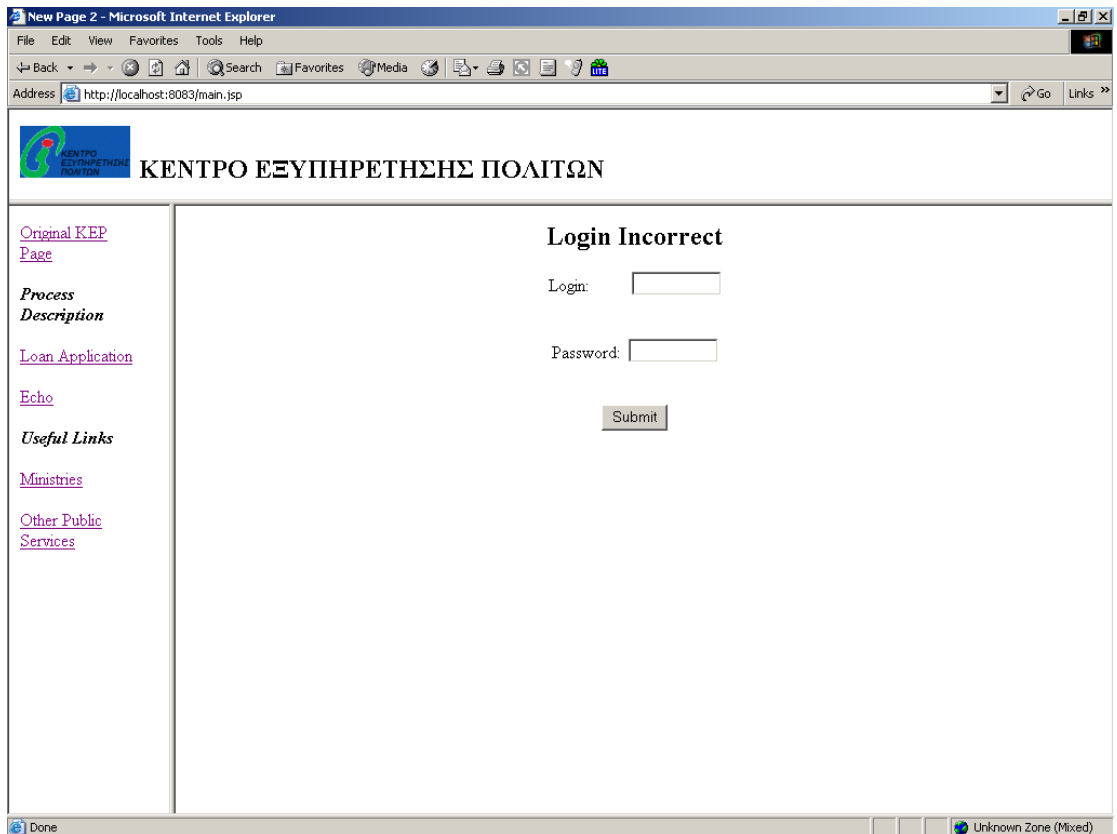
Στη σελίδα αυτή ο επισκέπτης του ΚΕΠ υποχρεούται να αναγνωρίσει το ρόλο του στο σύστημα. Στην περίπτωση που αυτοπροσδιοριστεί σαν απλός χρήστης δεν του ζητείται κάποια περαιτέρω πιστοποίηση και μπορεί να συνεχίσει από την κεντρική σελίδα των απλών χρηστών που θα περιγράψουμε αναλυτικά στην συνέχεια.

Αν ο χρήστης αυτοπροσδιοριστεί σαν διαχειριστής θα του ζητηθεί να πιστοποιήσει το δικαίωμά του αυτό παρέχοντας στο σύστημα τα προσωπικά του στοιχεία username και password. Αυτό γίνεται στη σελίδα που ακολουθεί.



Σε περίπτωση που η πιστοποίηση είναι επιτυχής το σύστημα θα τον οδηγήσει στην κεντρική σελίδα των χρηστών. Οι λειτουργίες που του παρέχονται από εκεί και πέρα θα περιγραφούν στην συνέχεια.

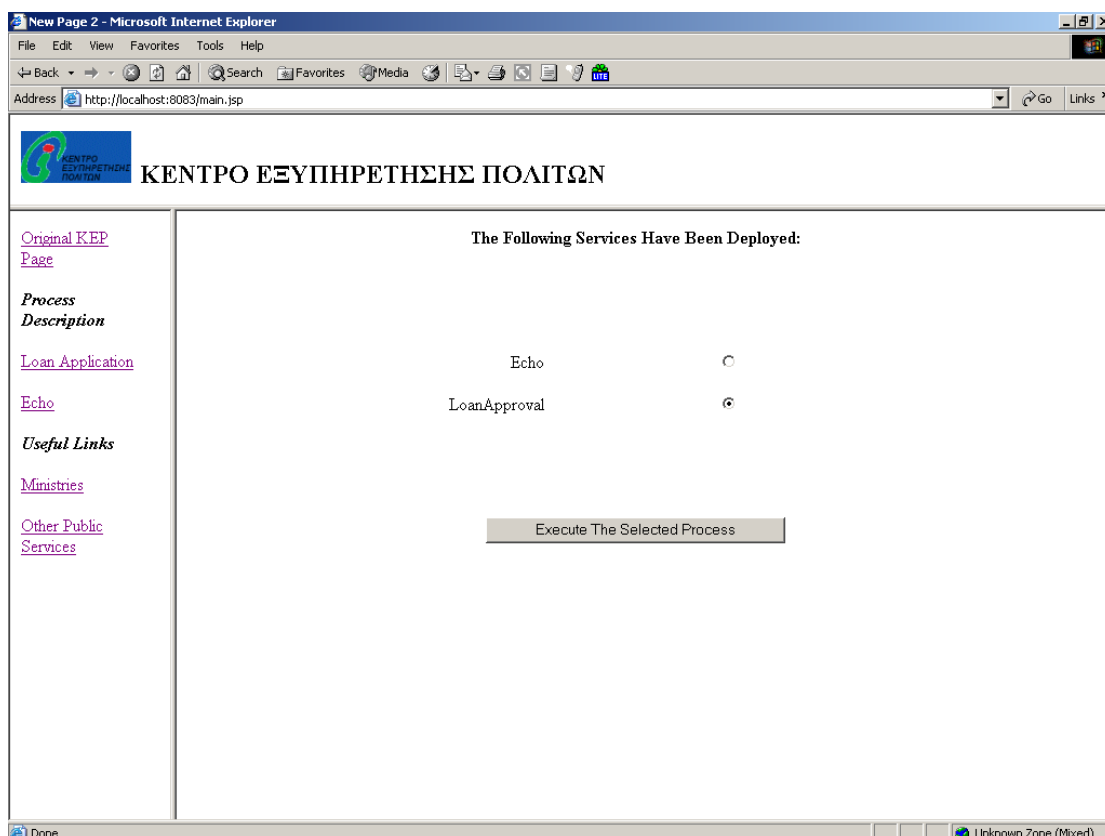
Σε περίπτωση που η πιστοποίηση αποτύχει, το σύστημα θα του ζητήσει να την επαναλάβει επανεισάγοντας τα προσωπικά του στοιχεία όπως φαίνεται στην σελίδα που ακολουθεί. Αυτό θα γίνεται μέχρις ότου η πιστοποίηση επιτύχει.



Παρακάτω αναλύονται ξεχωριστά οι επιμέρους λειτουργίες που αφορούν κάθε είδος χρήστη.

4.3 Λειτουργίες Απλού Χρήστη.

Όπως αναφέρθηκε και παραπάνω, όταν ο χρήστης αυτοπροσδιορίζεται σαν απλός χρήστης, δεν του ζητάτε κάποια περαιτέρω πιστοποίηση. Αμέσως το σύστημα αποδέχεται την παρουσία του και τον οδηγεί στην κεντρική σελίδα των χρηστών η οποία φαίνεται παρακάτω.



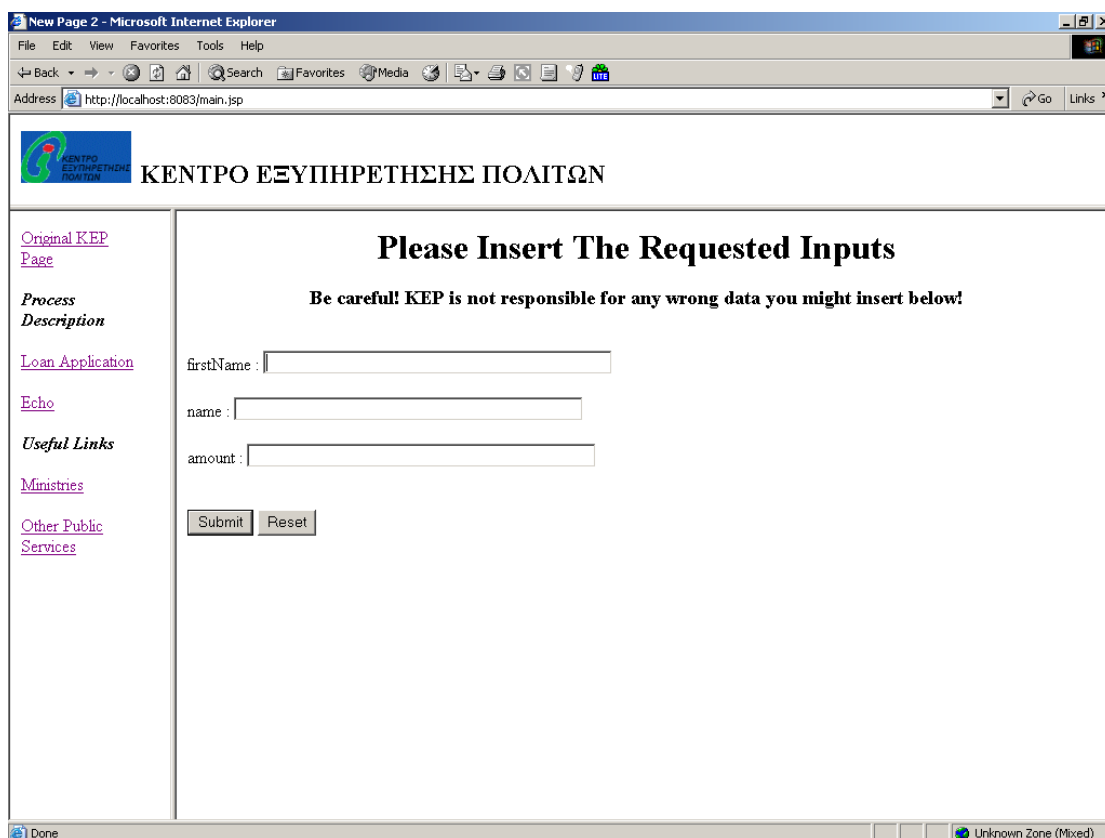
Εδώ ο χρήστης μπορεί να επιλέξει μία από τις διαδικασίες που έχουν ήδη ενταχθεί στο σύστημα και να την εκτελέσει.

Μπορεί επίσης επιλέγοντας τον κατάλληλο σύνδεσμο στα αριστερά της σελίδας να επιλέξει να ενημερωθεί για κάποιο από τα θέματα που περιγράφονται (π.χ. τις λεπτομέρειες μίας διαδικασίας), ή να επισκεφθεί κάποιον κατάλληλο σύνδεσμο(π.χ. Την κεντρική σελίδα κάποιου υπουργείου ή κάποιας υπηρεσίας). Οι λειτουργίες αυτές που δεν αφορούν άμεσα το ΚΕΠ αφού δεν είναι απαραίτητες για την εκτέλεση των διαδικασιών παρέχονται, όπως θα δούμε παρακάτω, σε κάθε βήμα που κάνει ο χρήστης μέσα στο ΚΕΠ, ώστε να είναι όσο το δυνατό καλύτερα πληροφορημένος.

Μόλις ο χρήστης επιλέξει μια διαδικασία για εκτέλεση, αμέσως το ΚΕΠ δημιουργεί εσωτερικά όλες τις κατάλληλες προϋποθέσεις για την εκτέλεσή της, αντλώντας από την βάση δεδομένων όλες εκείνες τις πληροφορίες που την αφορούν και έχουν ζωτικό ρόλο στην υλοποίησή της.

Με βάση τις πληροφορίες αυτές ο χρήστης οδηγείται στο επόμενο στάδιο που αφορά την απόδοση συγκεκριμένων τιμών στις μεταβλητές που η διαδικασία δέχεται σαν είσοδο. Αυτό γίνεται στην σελίδα που φαίνεται παρακάτω όπου το σύστημα επιδεικνύει τα ονόματα των επιμέρους μεταβλητών στο χρήστη δίπλα από τα αντίστοιχα textboxes όπου του ζητείται να εισάγει τις αντίστοιχες τιμές.

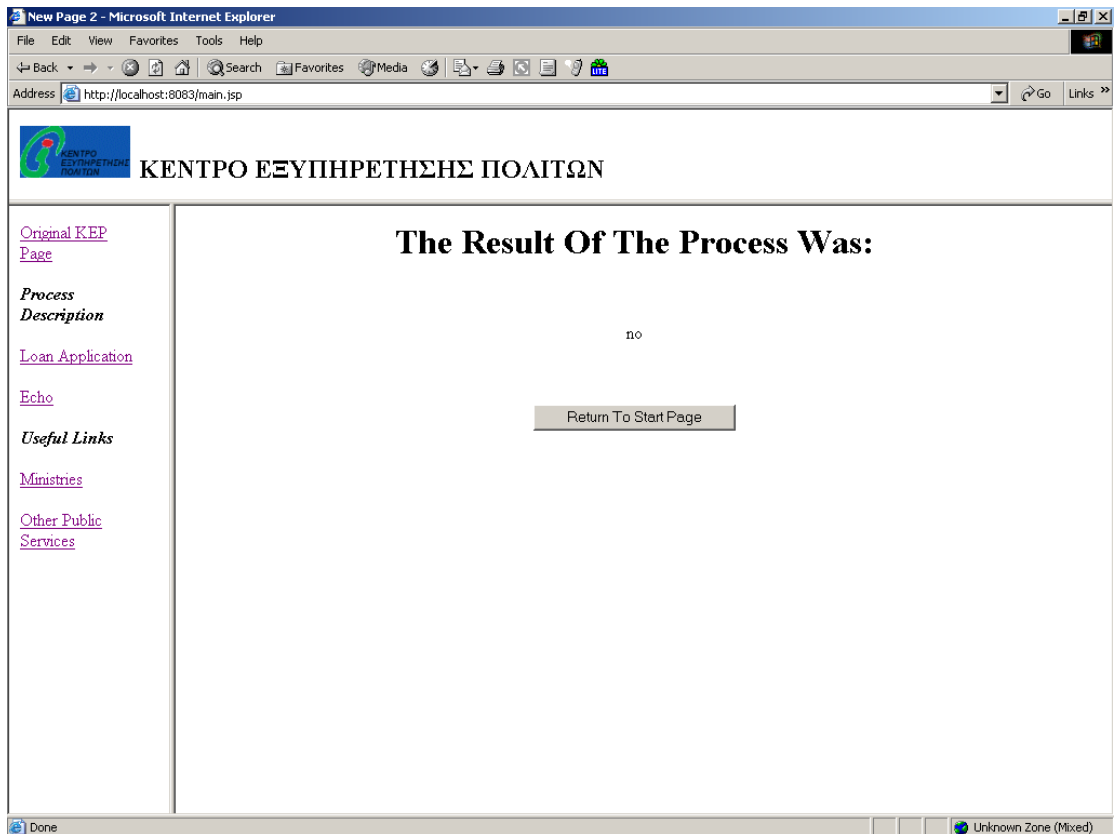
Με το κουμπί Submit ο χρήστης υποβάλλει τις τιμές που έχει εισάγει στο σύστημα ενώ με το κουμπί Reset του δίνεται η ευκαιρία να σβήσει τις ήδη υπάρχουσες τιμές και να εισάγει νέες αν θεωρεί ότι έχει κάνει λάθος.



Μετά την υποβολή των τιμών των εισόδων για την διαδικασία στο σύστημα , έχουμε όλα τα απαραίτητα δεδομένα προκειμένου να την εκτελέσουμε. Με το κουμπί submit δίνουμε ταυτόχρονα στο σύστημα την εντολή να εκτελέσει την επιλεγμένη διαδικασία.

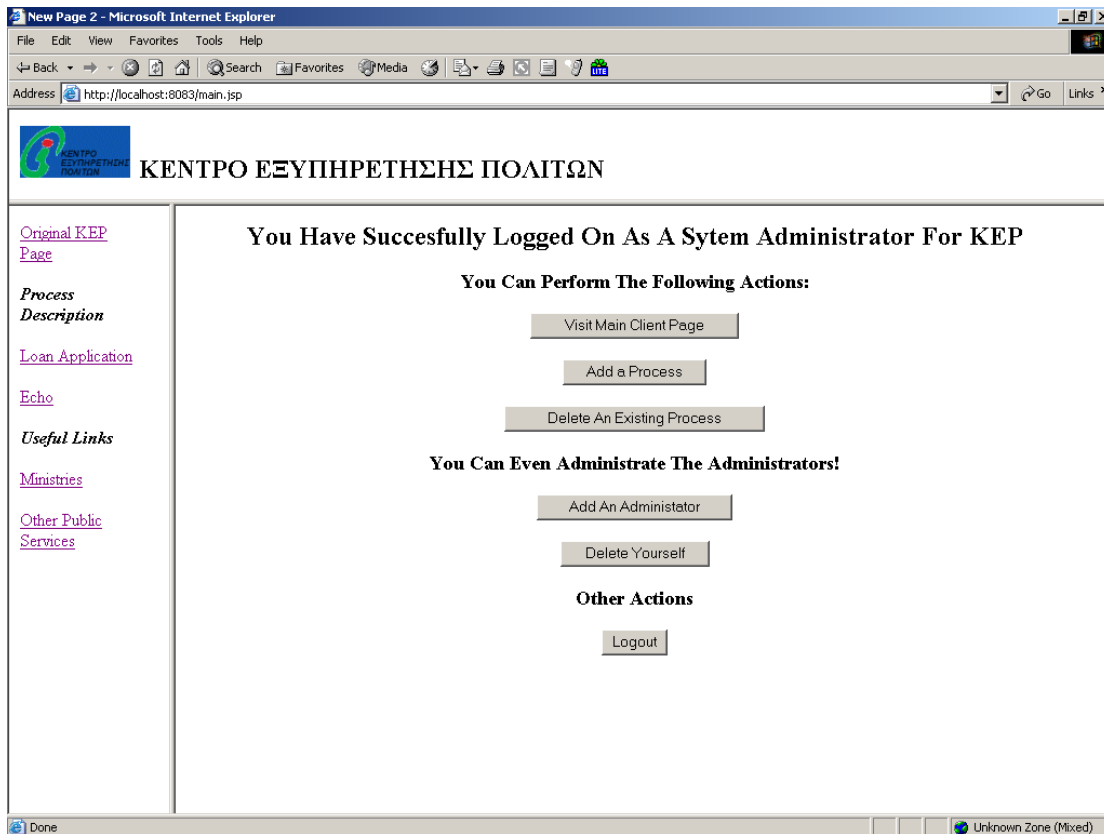
Η σελίδα που ακολουθεί έχει σαν σκοπό την επίδειξη στο χρήστη των αποτελεσμάτων εκτέλεσης της διαδικασίας. Σε περίπτωση που η διαδικασία δεν εκτελεστεί σωστά, το σύστημα στη θέση του αποτελέσματος θα εμφανίσει το προβλεπόμενο από αυτό μήνυμα λάθους. Από εκεί και πέρα είναι ο διαχειριστής υπεύθυνος για τον εντοπισμό του λάθους αυτού και την διόρθωση της δυσλειτουργίας που το προκάλεσε.

Παρακάτω επιδεικνύουμε το αποτέλεσμα εκτέλεσης μιας διαδικασίας που σκοπό της έχει την επιλεκτική έγκριση παροχής δανείου προς τον χρήστη ανάλογα με κριτήρια που αφορούν και γνωρίζει μόνο η ίδια. Στην συγκεκριμένη περίπτωση το αποτέλεσμα της αίτησης ήταν αρνητικό όπως φαίνεται και παρακάτω.



4.4 Λειτουργίες Διαχειριστή

Ο διαχειριστής του ΚΕΠ όπως αναφέραμε και παραπάνω αφού επιβεβαιώσει τα δικαιώματα που έχει πάνω στο σύστημα., εισάγοντας σωστό κωδικό όπως έχουμε ήδη περιγράψει, οδηγείται από το σύστημα στην κεντρική σελίδα που αντιστοιχεί στους διαχειριστές και όπου εμφανίζονται οι λειτουργίες που είναι προβλεπόμενες. Η σελίδα αυτή φαίνεται παρακάτω ενώ οι λειτουργίες που παρουσιάζονται σε αυτή θα αναλυθούν στη συνέχεια.

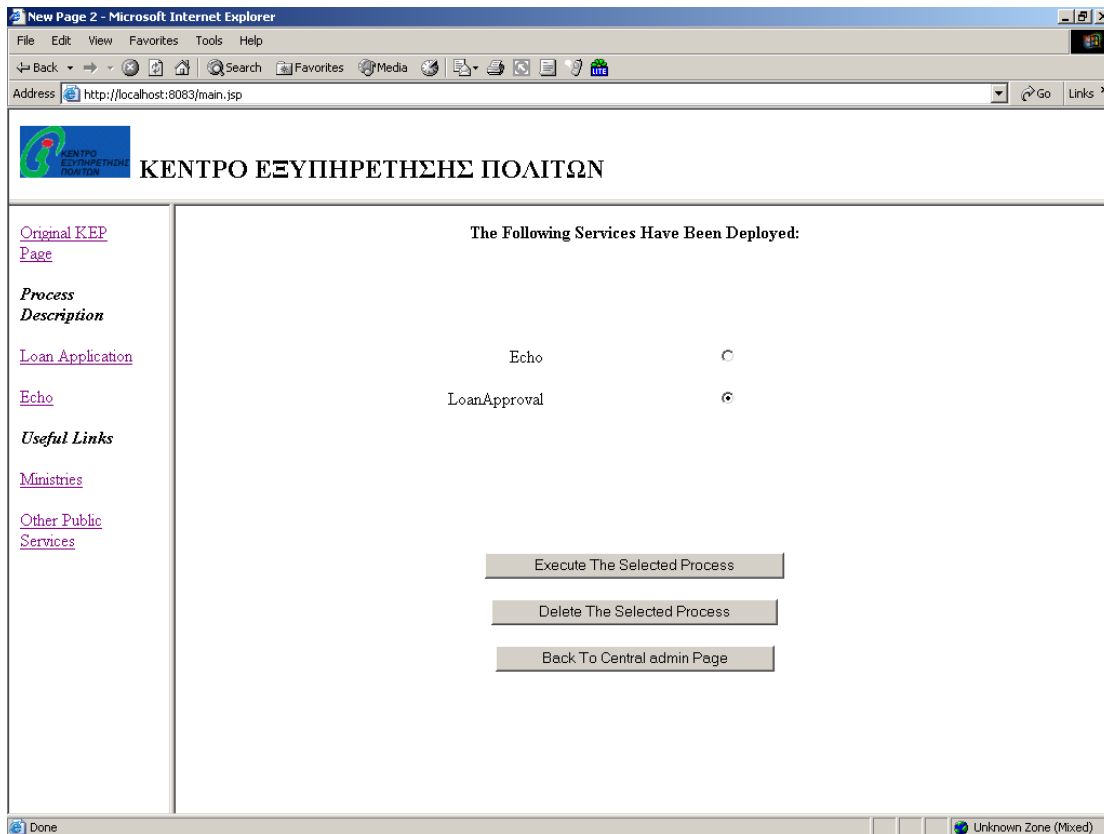


Αναλυτικά οι ενέργειες που μπορεί να κάνει ο διαχειριστής περιγράφονται στα υποκεφάλαια που ακολουθούν.

4.4.1 Επίσκεψη της κεντρικής σελίδας του απλού χρήστη/ διαγραφή διαδικασίας

Οι δύο αυτές διαδικασίες αν και ξεχωριστές μπορούν να υλοποιηθούν ταυτόχρονα. Αυτό μπορεί να γίνει ως εξής. Παρέχεται η δυνατότητα στον διαχειριστή να επισκεφτεί την κύρια σελίδα των απλών χρηστών. Αν η σελίδα καταλάβει (μέσα από κατάλληλες διαδικασίες στο εσωτερικό του κώδικα) πως ο χρήστης που την προσεγγίζει είναι διαχειριστής, θα εμφανίσει σε αυτόν δύο επιπλέον δυνατότητες. Την επιστροφή στην κεντρική σελίδα των χρηστών, για λόγους εύκολης μετακίνησης μέσα στο σύστημα, σε περίπτωση που ο διαχειριστής θέλει απλά να δει τις διαδικασίες που έχουν αναπτυχθεί σε αυτό, και την δυνατότητα διαγραφής της διαδικασίας που θα επιλέξει. Φυσικά ο διαχειριστής διατηρεί το δικαίωμά του να εκτελέσει κάποια διαδικασία σαν απλός χρήστης.

Στην σελίδα αυτή (η οποία φαίνεται παρακάτω) ο διαχειριστής μπορεί να οδηγηθεί είτε επιλέγοντας να επισκεφθεί την κεντρική σελίδα χρηστών, είτε επιλέξει να διαγράψει μια διαδικασία μέσω της κεντρικής σελίδας των διαχειριστών



Σε περίπτωση που ο διαχειριστής επιλέξει να διαγράψει την επιλεγμένη διαδικασία το σύστημα εκτελεί αμέσως την αίτηση του και επιβεβαιώνει την διαγραφή με το κατάλληλο μήνυμα.

4.4.2 Προσθήκη Διαδικασίας


Η εφαρμογή αυτή είναι σίγουρα η σημαντικότερη του ΚΕΠ αφού ζητά από τον διαχειριστή να περιγράψει πλήρως μια νέα διαδικασία ώστε να μπορεί το σύστημα να την παρέχει στο χρήστη του προς εκτέλεση χωρίς να γίνονται σφάλματα κατά την διάρκειά της.

Ο χρήστης επιλέγει από τη κεντρική σελίδα διαχειριστών την επιλογή που αντιστοιχεί στην πρόσθεση μιας επιπλέον διαδικασίας. Το σύστημα τον οδηγεί αμέσως στη παρακάτω σελίδα όπου του ζητείται να επιλέξει και να εισάγει διάφορες παραμέτρους στο σύστημα τις οποίες και θα εξηγήσουμε ευθύς αμέσως.

New Page 2 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://localhost:8083/main.jsp> Go Links

 ΚΕΝΤΡΟ ΕΞΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ

[Original KEP Page](#)

Process Description

[Loan Application](#)

[Echo](#)

Useful Links

[Ministries](#)

[Other Public Services](#)

To add a new process you need to follow the steps listed below:

- **Deploy The Process**

You can deploy the new process by clicking [here](#)

IMPORTANT: Make sure you keep notice of the **urn** value as well as the exact **name** of the process.
- **Fill In The Requested Fields**

Process Name:

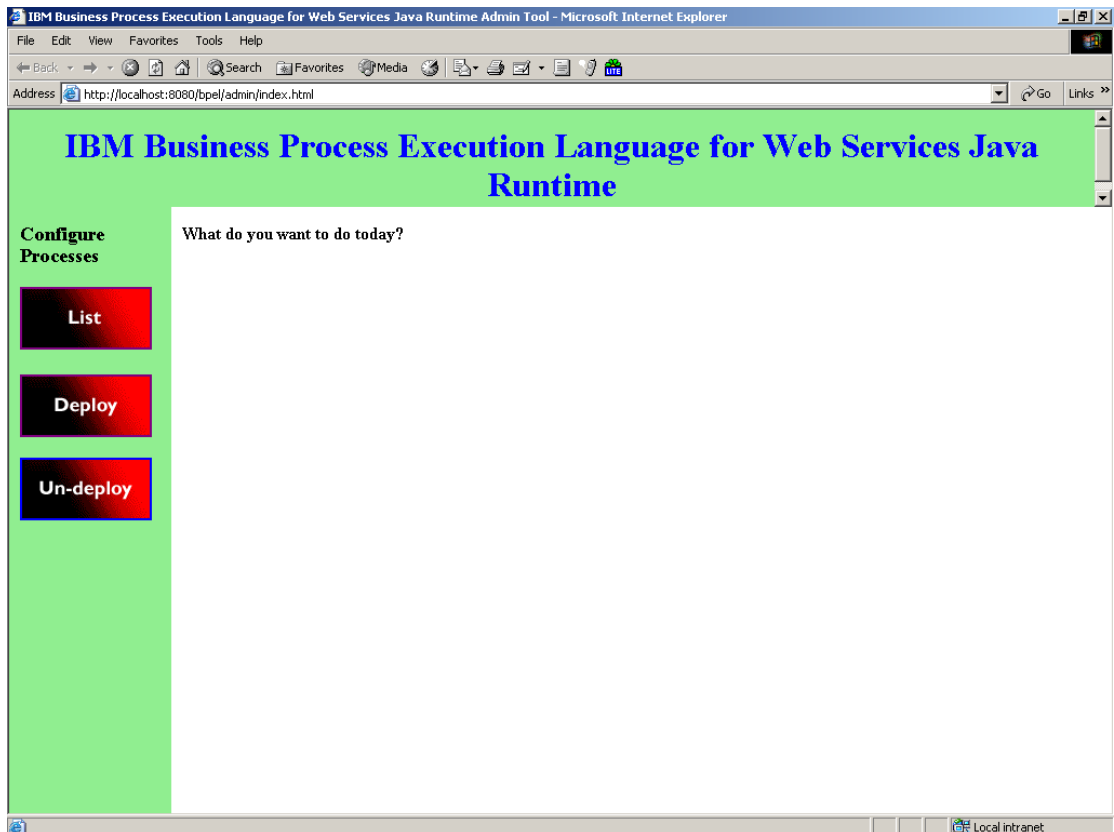
Process URN:

Number Of Process Inputs:

Method name:

Done Unknown Zone (Mixed)

Σε πρώτη φάση του ζητείται να δημιουργήσει την διαδικασία. εφόσον δεν το έχει κάνει ήδη. Η λειτουργία αυτή υλοποιείται επιλέγοντας τον υποδεικνυόμενο σύνδεσμο στην συγκεκριμένη σελίδα. Η επιλογή του εν λόγω συνδέσμου οδηγεί το σύστημα στο να εκκινήσει σε ξεχωριστό παράθυρο την εφαρμογή που φαίνεται παρακάτω.



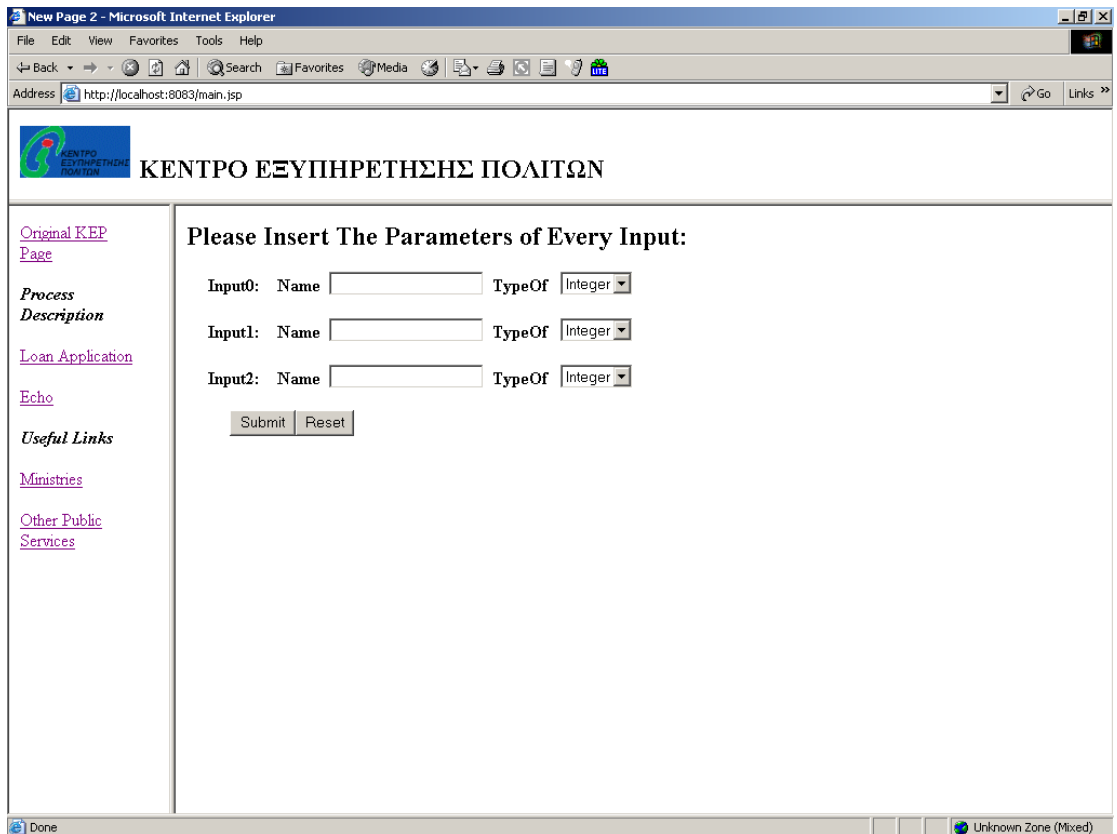
Η εφαρμογή αυτή μας παρέχεται από τους σχεδιαστές της BPEL4WS και ο κώδικας για την υλοποίησή της δεν μπορεί να μας γίνει γνωστός. Μπορούμε απλά να υποθέσουμε πως δημιουργεί κάποιο αρχείο στο κεντρικό φάκελο του εξυπηρετητή πάνω στον οποίο έχει θα γίνει η εκτέλεση της διαδικασίας. Η εφαρμογή αυτή αναπτύσσεται στα πλαίσια του εν λόγω εξυπηρετητή γεγονός που μας δεσμεύει να χρησιμοποιήσουμε δύο εξυπηρετητές για την εφαρμογή μας. Το συγκεκριμένο ζήτημα αφορά την υλοποίηση του συστήματος και θα αναλυθεί σε άλλο κεφάλαιο.

Ακολουθώντας τα προβλεπόμενα βήματα ο διαχειριστής χρησιμοποιεί το παραπάνω εργαλείο της BPEL4WS προκειμένου να δημιουργήσει την διαδικασία που θέλει να εισάγει στο σύστημα. Το εργαλείο ζητά από τον χρήστη να δώσει πληροφορίες για την τοποθεσία στην οποία βρίσκονται τα WSDL αρχεία τόσο του WSDL αρχείου που περιγράφει την ίδια την BPEL4WS διαδικασία. Επίσης του ζητείται να προσδιορίσει και την τοποθεσία του BPEL4WS αρχείου που περιέχει την καθεαυτό διαδικασία.

Μόλις ο διαχειριστής εισάγει τα δεδομένα που περιγράψαμε παραπάνω η εφαρμογή της BPEL4WS έχει στη διάθεσή της όλα τα στοιχεία που χρειάζεται προκειμένου να δημιουργήσει τα απαραίτητα μεταδεδομένα στον εξυπηρετητή και να γίνει δυνατή η εκτέλεση της διαδικασίας.

Ο χρήστης έρχεται και πάλι πίσω στην σελίδα που αφορά τις εισόδους τις διαδικασίας αφού πρώτα σημειώσει τα απαραίτητα δεδομένα που τον προτρέπει το σύστημα έτσι ώστε να τα εισάγει σαν τιμές στις στατικές μεταβλητές μιας διαδικασίας. Ανάμεσα σε αυτά όπως είναι γνωστό είναι και ο αριθμός των μη στατικών εισόδων τις τιμές των οποίων εισάγει κάθε φορά που εκτελείται η διαδικασία ο χρήστης της.

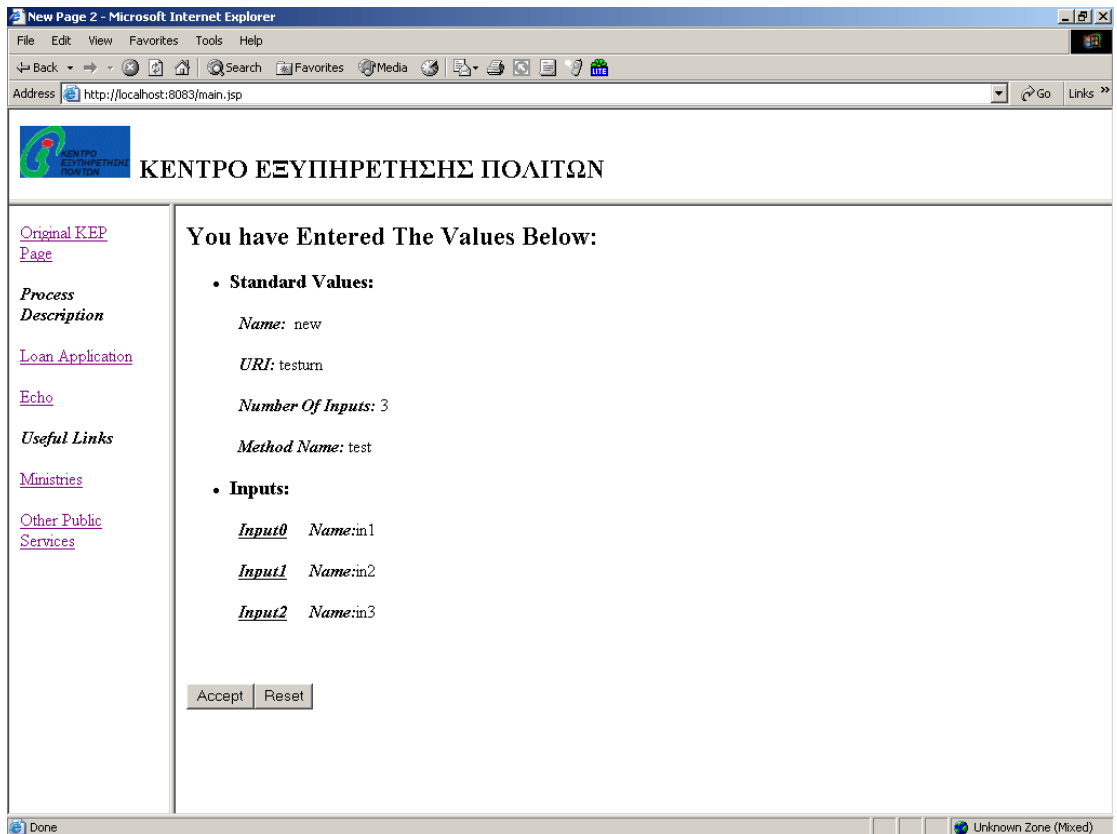
Ο αριθμός των εισόδων είναι απαραίτητος για να δημιουργήσει το σύστημα την επόμενη σελίδα στην οποία θα οδηγήσει τον διαχειριστή. Η σελίδα αυτή φαίνεται παρακάτω και είναι το στιγμιότυπο που αντιστοιχεί σε 3 εισόδους.



Ο διαχειριστής καλείται από το σύστημα να εισάγει τα ονόματα που αντιστοιχούν στις μη στατικές μεταβλητές καθώς και τους τύπους των δεδομένων που δέχεται η κάθε μία σαν τιμές.

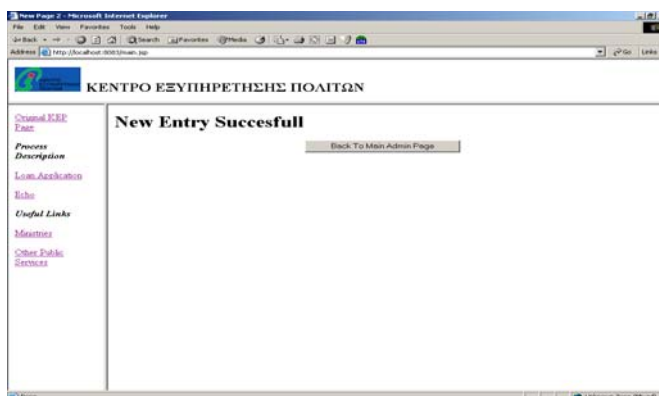
Με τα δεδομένα αυτά το σύστημα είναι έτοιμο να δημιουργήσει την διαδικασία και να εισάγει τις απαραίτητες πληροφορίες για την εκτέλεσή της στην βάση δεδομένων που χρησιμοποιείται για τον εν λόγω σκοπό.

Πριν ολοκληρωθεί όμως η διαδικασία το σύστημα ζητά από τον διαχειριστή να επιβεβαιώσει τα δεδομένα που έχει εισάγει παρουσιάζοντάς τα του συγκεντρωτικά στην παρακάτω σελίδα.



Η επιβεβαίωση θεωρείται αναγκαία αφού ένα λάθος είναι πιθανό και ταυτόχρονα κάνει την διαδικασία αδύνατη να εκτελεστεί

Εάν ο χρήστης επιλέξει να αποδεχθεί τις τιμές που έχει εισάγει η εισαγωγή ολοκληρώνεται. Αν όχι το σύστημα οδηγεί τον χρήστη στο κατάλληλο σημείο προκειμένου να επανεκκινήσει η διαδικασία εισαγωγής. Η ολοκλήρωση μιας εγγραφής επιβεβαιώνεται με τον τρόπο που φαίνεται παρακάτω.



Εδώ ολοκληρώνονται οι λειτουργίες που αφορούν την διαχείριση των διαδικασιών. Ο διαχειριστής όμως του ΚΕΠ έχει στην διάθεσή του και άλλες δυνατότητες. Οι δυνατότητες αυτές αφορούν την διαχείριση πλέον των εγγραφών στη βάση που αφορούν τους υπόλοιπους διαχειριστές και τον εαυτό του.

Έτσι ο κάθε διαχειριστής μπορεί να εκτελέσει οποιαδήποτε από τις παρακάτω λειτουργίες.

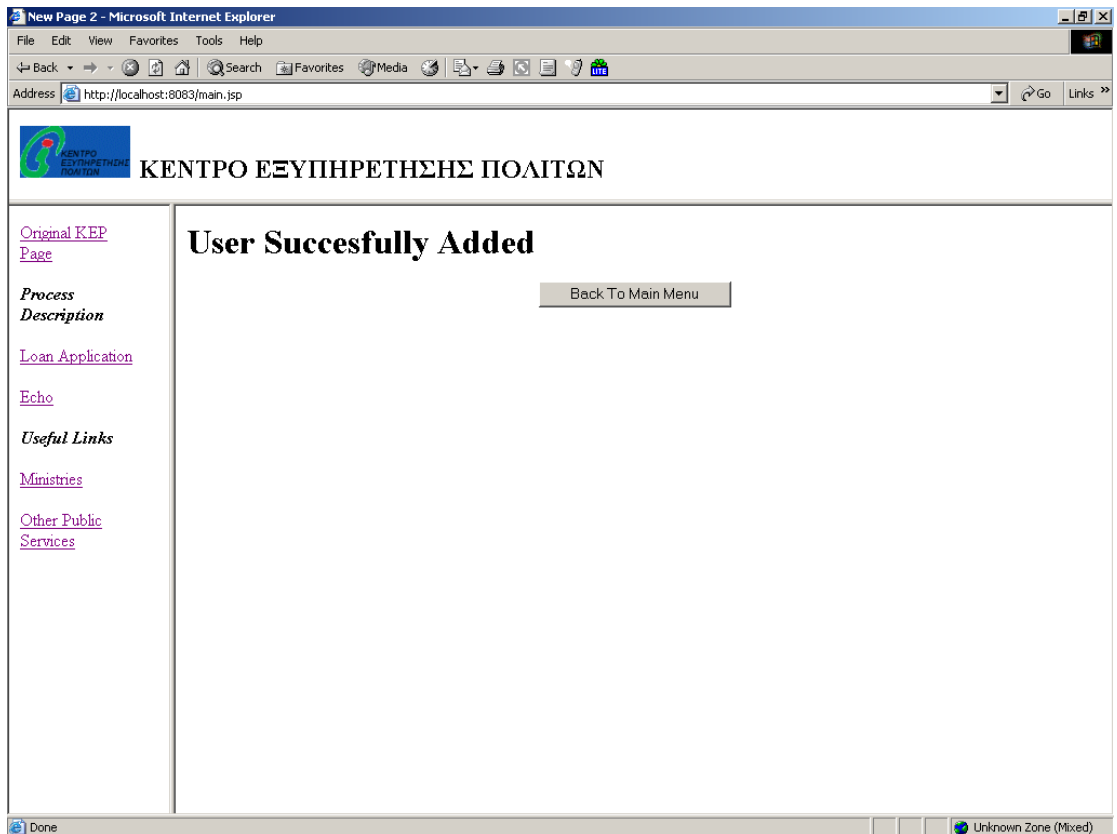
4.4.3 Προσθήκη Διαχειριστή

Εφόσον ο χρήστης καταγραφεί στο σύστημα σαν διαχειριστής του παρέχεται η δυνατότητα να προσθέσει σε αυτό και άλλους διαχειριστές. Η λειτουργία αυτή υλοποιείται μέσω της παρακάτω σελίδας.

The screenshot shows a web browser window with the address bar displaying 'http://localhost:8083/main.jsp'. The page header features the logo of the 'ΚΕΝΤΡΟ ΕΞΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ' (Citizen Service Center). The main content area is titled 'Add A New Administrator' and includes the instruction 'Please Fill The Requested Fields'. The form consists of three input fields: 'Login', 'Password', and 'Password'. An 'Add' button is positioned to the right of the second password field. On the left side of the page, there are several links: 'Original KEP Page', 'Process Description', 'Loan Application', 'Echo', 'Useful Links', 'Ministries', and 'Other Public Services'. The browser's status bar at the bottom shows 'Done' and 'Unknown Zone (Mixed)'.

Στη παραπάνω σελίδα ζητείται από τον χρήστη να εισάγει το νέο όνομα με το οποίο θα αναγνωρίζεται ο νέος χρήστης από το σύστημα και το μυστικό κωδικό που θα χρησιμοποιείται κάθε φορά για επικύρωση. Ο μυστικός κωδικός ζητείται να εισαχθεί δύο φορές για να ελεγχθεί κάποιο πιθανό λάθος πληκτρολόγησης συγκρίνοντας τις δυο καταχωρήσεις αν είναι ίσες. Η ανάγκη για τον έλεγχο αυτό είναι προφανής και κοινή σε όλα τα συστήματα που απαιτούν διαχωρισμό χρηστών.

Σε περίπτωση που το σύστημα αποδεχθεί το νέο χρήστη ενημερώνει για την σωστή εγγραφή με το ακόλουθο μήνυμα αποδοχής

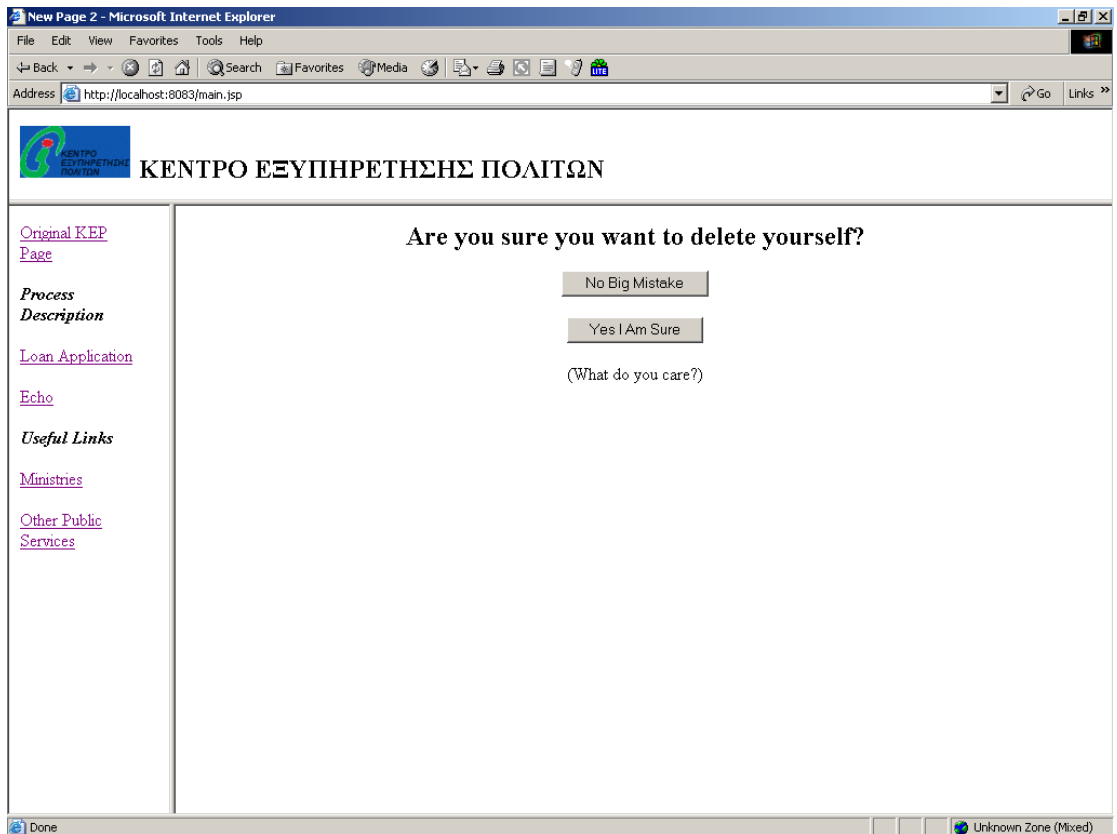


Σε περίπτωση απόρριψης ζητείται από τον χρήστη να επανεισάγει τα στοιχεία εγγραφής.

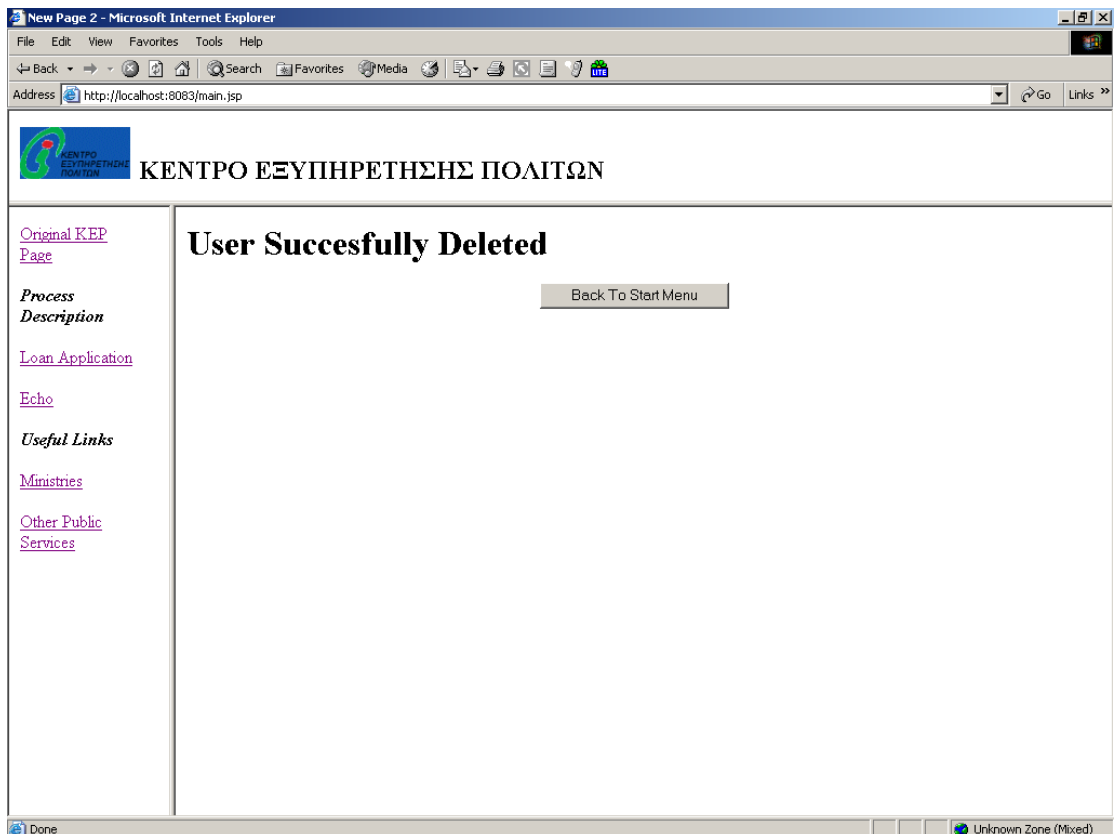
4.4.4 Διαγραφή Χρήστη

Μια ακόμη δυνατότητα που παρέχεται στο διαχειριστή του συστήματος είναι και η διαγραφή κάποιου χρήστη από αυτό. Προκειμένου όμως να υπάρχει διαύγεια ενεργειών και έλεγχος όσον αφορά τη συγκεκριμένη λειτουργία ο κάθε χρήστης μπορεί να διαγράψει μόνο τον εαυτό του.

Η λειτουργία αυτή υλοποιείται πατώντας το κατάλληλο κουμπί από την κεντρική σελίδα των διαχειριστών. Στη συνέχεια το σύστημα επιβεβαιώνει την επιθυμία του διαχειριστή να διαγραφεί, δίνοντάς του την επιλογή να ακύρωση την εκτέλεση της διαγραφής.



Εάν ο χρήστης αποδεχθεί τις συνέπειες της αυτοδιαγραφής του, το σύστημα εκτελεί την συγκεκριμένη λειτουργία και επιβεβαιώνει την διαγραφή με το κατάλληλο μήνυμα



4.4.5 Logout

Τέλος το σύστημα παρέχει στους διαχειριστές την επιλογή να αποσυνδεθούν από αυτό με τη λειτουργία logout. Με την επιλογή αυτή το σύστημα παύει να αντιμετωπίζει το χρήστη σαν διαχειριστή έως ότου αυτός επιλέξει να επανεπικυρώσει τα στοιχεία του. Η εφαρμογή της συγκεκριμένης λειτουργίας οδηγεί το χρήστη στην αρχική σελίδα της εφαρμογής απ' όπου μπορεί να επανεκκίνηση την πλοήγησή του σε αυτό.

Η λειτουργία αυτή είναι καλό να εκτελείται κάθε φορά που κάποιος διαχειριστής επιλέξει να αποχωρήσει από το σύστημα ώστε αυτό να ενημερώνεται για την επιλογή του. Η ενημέρωση για την πληροφορία αυτή δεν γίνεται να υλοποιηθεί με άλλο τρόπο. Με λίγα λόγια το σύστημα μπορεί μόνο να υποθέσει αν κάποιος διαχειριστής είναι ακόμα συνδεδεμένος με αυτό και δεν έχει αποχωρήσει. Σε συστήματα που η κίνηση εντός μιας δικτυακής εφαρμογής είναι αρκετά μεγάλη και χρειάζεται να ελεγχθεί το σύστημα αναγκάζεται να μαντέψει αν ο χρήστης έχει αποχωρήσει εκτός και αν έχει εφαρμόσει την αντίστοιχη λειτουργία logout.

4.5 Υπερδιαχειριστές.

Η κατηγορία αυτή των χρηστών είναι προαιρετική και δεν χρησιμοποιείται στο συγκεκριμένο σύστημα. Οι λειτουργίες που θα χρησιμοποιούσε θα ήταν ίδιες με αυτές των απλών διαχειριστών με την διαφορά ότι θα είχαν περισσότερες ελευθερίες. Θα μπορούσαν για παράδειγμα να διαγράψουν και άλλους χρήστε εκτός του εαυτού τους ή να ορίσουν μια διαδικασία ως μη διαγράψιμη. Κρίθηκε πως ένας τέτοιος τύπος χρήστη δεν είναι απαραίτητος για το σύστημά μας. Θα ήταν απαραίτητος εάν είχαμε πολλούς διαχειριστές και χρειαζόνταν περισσότερες δυνατότητες ελέγχου πάνω σε αυτούς. Εφόσον κάτι τέτοιο δεν ισχύει δεν επεκτάθηκαν οι λειτουργίες διαχείρισης αλλά παρέμειναν ως έχουν

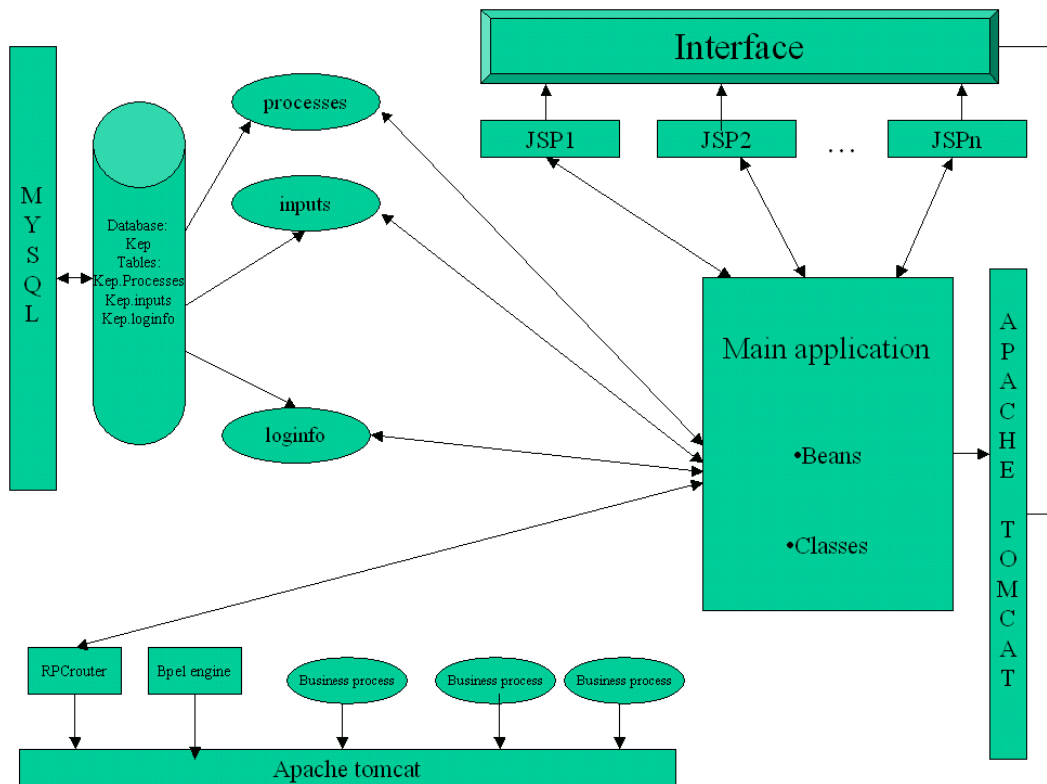
5

Υλοποίηση

5.1 Πλατφόρμες και προγραμματιστικά εργαλεία

Στην ενότητα αυτή περιγράφονται τα χαρακτηριστικά της συγκεκριμένης υλοποίησης, όπως η πλατφόρμα ανάπτυξης και εκτέλεσης, τα προγραμματιστικά εργαλεία, οι απαιτήσεις της εφαρμογής, κ.λπ..

Μια σχηματική απεικόνιση του συστήματος όπως αυτό υλοποιήθηκε φαίνεται παρακάτω



Όπως φαίνεται και από το παραπάνω σχήμα η υλοποίηση αποτελείται από πολλά επιμέρους τμήματα, ανεξάρτητες οντότητες που αλληλεπιδρούν με συγκεκριμένο τρόπο. Γενικά προσπαθώντας να αναλύσουμε το παραπάνω σχήμα θα περιγράψουμε όλες τις επιμέρους επιλογές που κάναμε προκειμένου να το υλοποιήσουμε και θα περιγράψουμε ένα ένα το κάθε προγραμματιστικό εργαλείο. Κάτι τέτοιο σε συνδυασμό με την περιγραφή της αρχιτεκτονικής στο προηγούμενο κεφάλαιο αρκεί για να καταλάβει ο αναγνώστης όλες τις λεπτομέρειες που του χρειάζονται όχι μόνο για να χρησιμοποιήσει αλλά και να επέμβει προγραμματιστικά στο σύστημα προκειμένου να το προσαρμόσει στις ανάγκες του.

5.1.1 Επιλογή γλώσσας

Στο συγκεκριμένο κεφάλαιο θα μιλήσουμε για την γλώσσα προγραμματισμού που χρησιμοποιήσαμε προκειμένου να υλοποιηθεί η ζητούμενη εφαρμογή. Εξαιτίας των πολλαπλών προσπαθειών υλοποίησης έγινε χρήση δύο γλωσσών προγραμματισμού με σκοπό τη σχεδίαση και δημιουργία του ΚΕΠ. Παρακάτω θα γίνει μια σύντομη αναφορά και στις δύο ενώ στη συνέχεια θα επιχειρήσουμε μια μικρή σύγκριση όσον αφορά την εφαρμογή μας και γενικά τις παραμέτρους των γλωσσών αυτών που αφορούν παρόμοιες εφαρμογές.

VISUAL BASIC.NET

Όπως αναφέραμε και στο σχετικό κεφάλαιο, οι πρώτες προσπάθειες υλοποίησης έγιναν με τη βοήθεια της γλώσσας προγραμματισμού VISUAL BASIC 7.0 ή VISUAL BASIC.NET. Η γλώσσα αυτή αναπτύχθηκε σχετικά πρόσφατα από τον εμπορικό οργανισμό MICROSOFT και αποτελεί μέρος του προγραμματιστικού πακέτου MICROSOFT VISUAL STUDIO .NET. Η περαιτέρω περιγραφή της δεν έχει νόημα αφού ναι μεν ασχοληθήκαμε με αυτή αρκετά αλλά η τελική υλοποίηση έγινε με βάση την γλώσσα JAVA.

JAVA

Η υλοποίηση του συστήματος έγινε τελικά αποκλειστικά σε JAVA 2. Συγκεκριμένα, χρησιμοποιήθηκε το Java 2 SDK Standard Edition v1.3 και το Java 2 SDK Enterprise Edition v1.2.1. Η επιλογή της συγκεκριμένης γλώσσας στηρίχθηκε στα παρακάτω χαρακτηριστικά της:

- Είναι αντικειμενοστραφής. Η Java – σε αντίθεση με τη C++ που περιέχει και μη αντικειμενοστραφή χαρακτηριστικά – είναι μια καθαρά αντικειμενοστραφής γλώσσα προγραμματισμού. Το γεγονός αυτό διευκολύνει την ανάπτυξη επαναχρησιμοποιήσιμου κώδικα. Η ίδια η Java παρέχει ένα σημαντικό αριθμό κλάσεων που επιταχύνουν την ανάπτυξη κώδικα.
- Παράγει κώδικα μεταφέρσιμο σε οποιαδήποτε αρχιτεκτονική. Ο μεταγλωττιστής της Java δεν παράγει κώδικα για την αρχιτεκτονική ενός συγκεκριμένου συστήματος, αλλά για την εικονική μηχανή της Java (Java Virtual Machine). Μια εφαρμογή λοιπόν αυτής της γλώσσας μπορεί να τρέξει σε οποιαδήποτε πλατφόρμα, χωρίς καμιά μετατροπή. Επομένως, ο κώδικας δημιουργείται αλλά και μεταγλωττίζεται μονάχα μια φορά.
- Παράγει σταθερό αλλά και αξιόπιστο κώδικα. Η Java περιορίζει τα προγραμματιστικά λάθη που μπορεί να οδηγήσουν σε εφαρμογές ασταθείς και αναξιόπιστες. Παρέχει μηχανισμούς ελέγχου λαθών και παραβάσεων μνήμης κατά την εκτέλεση (run-time errors), που θα μπορούσαν να οδηγήσουν σε αστάθεια το σύστημα, φροντίζει για την αποδέσμευση της μνήμης (ύπαρξη garbage collector), αφαιρώντας από τον προγραμματιστή αυτή την ευθύνη, ενώ τέλος ο χειρισμός των εξαιρέσεων απλοποιεί τη διαχείριση λαθών και την ανάληψη από αυτά. Η έλλειψη δεικτών, χωρίς να μειώνει τη δύναμή της, απλοποιεί την ανάπτυξη του κώδικα και

απαλλάσσει τις εφαρμογές από τις απρόβλεπτες καταστάσεις που εμφανίζεται σε προγράμματα C και C++.

- Είναι δυναμική και καταναμημένη. Χαρακτηρίζοντας την Java δυναμική, αναφερόμαστε στη δυνατότητα δυναμικής φόρτωσης μιας κλάσης στον Java μεταφραστή οποιαδήποτε χρονική στιγμή κατά την εκτέλεση. Επίσης, μας δίνει τη δυνατότητα να αποκτήσουμε δυναμικά πληροφορίες για μια κλάση στο χρόνο εκτέλεσης. Η Java χαρακτηρίζεται, επίσης, καταναμημένη, χάρη στην υποστήριξη που παρέχει σε επίπεδο δικτύου. Εδώ πρέπει να αναφερθούμε στο Remote Method Invocation (RMI) Application Interface (API), το οποίο επιτρέπει σε ένα Java πρόγραμμα να καλέσει μεθόδους απομακρυσμένων αντικειμένων σαν να ήταν τοπικά.
- Περιλαμβάνει ή υποστηρίζει όλα τα απαραίτητα APIs για την πραγματοποίηση των λειτουργιών που χρειάστηκαν (JDBC, Swing, Servlets, JAXP). Το γεγονός αυτό αφενός διευκολύνει τον προγραμματιστή, αφού του παρέχει ένα βασικό υπόβαθρο, πάνω στο οποίο θα δημιουργήσει τις δικές του εφαρμογές, αφετέρου ελαχιστοποιεί τις πιθανότητες κακής επικοινωνίας μεταξύ των τμημάτων του κώδικα. Ταυτόχρονα, οδηγεί σε ευανάγνωστο κώδικα και σίγουρα στην καλύτερη και βαθύτερη εκμάθηση του προγραμματιστικού περιβάλλοντος.
- Συνοδεύεται από πλήρη και κατατοπιστική τεκμηρίωση (documentation), η οποία παρέχεται δωρεάν μέσω του World Wide Web.

5.1.2 Λειτουργικό σύστημα

Έχοντας να δουλέψουμε με την γλώσσα Java είχαμε την επιλογή να αναπτύξουμε τον κώδικά μας σε οποιοδήποτε περιβάλλον και λειτουργικό σύστημα θέλαμε. Επιλέχθηκαν τα Windows 2000 της εταιρίας Microsoft λόγω της συμβατότητας που προσφέρουν, αφού αποτελούν μακράν την πιο διαδεδομένη πλατφόρμα, αλλά και λόγω της πληθώρας των προγραμμάτων και εργαλείων για ανάπτυξη κώδικα που διατίθενται σε αυτά.

Ο παραγόμενος κώδικας βέβαια είναι πολύ εύκολο να χρησιμοποιηθεί και σε άλλα λειτουργικά συστήματα, αρκεί βέβαια να είναι εγκατεστημένο το Java Runtime Environment 1.3. Για την λειτουργία του ΚΕΠ ήταν απαραίτητη η και ύπαρξη ενός Java enabled Web Server, ενός διακομιστή δηλαδή που έχει τη δυνατότητα να εκτελεί κλάσεις της γλώσσας Java.

5.1.3 Επιλογή τρόπου δημιουργίας δυναμικών ιστοσελίδων

Για τη δημιουργία των διαπροσωπειών της εφαρμογής με τον χρήστη του ΚΕΠ χρησιμοποιήθηκε η τεχνολογία των JSP (Java Server Pages), που περιλαμβάνει ένα σύνολο πλεονεκτημάτων σε σχέση με άλλες παρόμοιες τεχνολογίες και επιτρέπει στους προγραμματιστές να δημιουργήσουν δυναμικές ιστοσελίδες, οι οποίες μεταφράζονται κατά τη διάρκεια της εκτέλεσής τους. Οι JSPs έχουν τα παρακάτω πλεονεκτήματα:

- Είναι μέρος του κώδικα που συνοδεύει την πλατφόρμα της Java.
- Είναι ανεξάρτητες της αρχιτεκτονικής του συστήματος στο οποίο εκτελούνται.
- Μπορούν να «επικοινωνήσουν» πολύ εύκολα με άλλες κλάσεις ή οντότητες της Java.
- Μπορούν να εξυπηρετούν ταυτόχρονα πολλαπλές αιτήσεις διαφορετικών χρηστών και επιπλέον έχουν τη δυνατότητα να τις συγχρονίζουν.
- Έχουν τη δυνατότητα να διαβιβάζουν τις αιτήσεις των χρηστών σε άλλους servers ή JSPs.

- Παρέχουν όλες τις δυνατότητες των φαινομενικά ισχυρότερων servlets, αφού αυτόματα μεταφράζονται σε servlets που αναλαμβάνουν να φέρουν σε πέρας την απαίτηση.
- Δίνουν τη δυνατότητα στον προγραμματιστή να δημιουργήσει πολύ εύκολα τόσο στατικές όσο και δυναμικές οντότητες, με μια πιο φυσική προσέγγιση από άλλες τεχνολογίες.
- Συνοδεύεται από πλήρη και κατατοπιστική τεκμηρίωση (documentation), η οποία παρέχεται δωρεάν μέσω του World Wide Web.
- Υποστηρίζονται από τη μεγάλη πλειονότητα των servers που κυκλοφορούν στην αγορά λογισμικού.

5.1.4 Επιλογή τρόπου επικοινωνίας με τη Βάση Δεδομένων

Για την επικοινωνία του συστήματος **ΚΕΠ** με τη βάση δεδομένων χρησιμοποιήθηκε το πρωτόκολλο JDBC (Java DataBase Connectivity). Το JDBC είναι ιδιαίτερα απλό, ευέλικτο και αποτελείται από κλάσεις και διαπροσωπίες (interfaces) που επιτρέπουν την επεξεργασία των δεδομένων μιας οποιασδήποτε Βάσης Δεδομένων με τη χρήση ενσωματωμένης SQL, καθώς επίσης και στην επισκόπηση των μετά-δεδομένων της. Για να μπορέσει μια συγκεκριμένη Βάση Δεδομένων να επικοινωνήσει μέσω JDBC θα πρέπει να υλοποιηθούν οι κατάλληλοι οδηγοί (drivers), κλάσεις δηλαδή, που υλοποιούν τα interfaces και τις αφηρημένες κλάσεις που δηλώνονται από το JDBC API. Στην παρούσα διπλωματική επιλέχθηκαν οδηγοί τύπου 4. Ένας οδηγός τύπου 4 χαρακτηρίζεται από το γεγονός ότι μετατρέπει τις καλούμενες μεθόδους στο πρωτόκολλο δικτύου (network protocol) το οποίο χρησιμοποιείται άμεσα από το εκάστοτε ΣΔΒΔ. Αυτό επιτρέπει την πραγματοποίηση μιας άμεσης κλήσης από το μηχάνημα-πελάτη (client machine) στον εξυπηρετητή του ΣΔΒΔ και είναι μια πρακτική λύση για πρόσβαση μέσω κάποιου δικτύου ή ακόμα και μέσω του Internet, κάτι που συμβαίνει στην πλειονοψηφία των περιπτώσεων χρησιμοποίησης του συστήματος. Καθώς πολλά από αυτά τα πρωτόκολλα είναι εξειδικευμένα ανά σύστημα, η πρωταρχική πηγή για αυτούς τους οδηγούς ήταν ο κατασκευαστής του συγκεκριμένου ΣΔΒΔ

5.1.5 Επιλογή εργαλείου ανάπτυξης κώδικα

Ο κώδικας αναπτύχθηκε με την βοήθεια του εργαλείου JBUILDER8. Οι κύριοι λόγοι για την επιλογή του συγκεκριμένου εργαλείου ήταν οι εξής:

1. Παρέχει δυνατότητες σύνδεσης με οποιαδήποτε τεχνολογία σχετική με την γλώσσα προγραμματισμού JAVA
2. Πολλές από τις επιμέρους διαδικασίες που είναι απαραίτητες για την ανάπτυξη του κώδικα μιας συγκεκριμένης εφαρμογής όπως υλοποίηση της σύνδεσης με μια βάση, γίνονται με τρόπο σχεδόν αυτοματοποιημένο.
3. Γίνεται εύκολη η δημιουργία διεπαφών χρήστη χάρη στις αυτοματοποιημένες διαδικασίες που παρέχει το εν λόγω εργαλείο.
4. Είναι η πλέον διαδεδομένη πλατφόρμα υλοποίησης κώδικα και κάνει την εξικοίωση με αυτή πολύ χρήσιμη εμπειρία.
5. Υπάρχει άμεση και πλούσια υποστήριξη για την αντιμετώπιση πιθανών προβλημάτων.

5.1.6 Επιλογή Web Server

Όπως αναφέρθηκε πιο πάνω χρησιμοποιήσαμε τον apache tomcat προκειμένου να αναπτύξουμε την εφαρμογή της BPWS4J engine. Χρησιμοποιήσαμε επίσης τον apache

tomcat σαν διακομιστή πάνω στον οποίον θα εκτελούνταν η κυρίως εφαρμογή. Οι λόγοι που μας οδήγησαν σε αυτή την επιλογή είναι οι εξής:

- Παρέχει υποστήριξη για εφαρμογές που κάνουν χρήση της τεχνολογίας των Java Server Pages (JSP).
- Αποτελεί έναν αρκετά σταθερό Web Server.
- Συνοδεύεται από πλήρη και κατατοπιστική τεκμηρίωση (documentation), η οποία παρέχεται δωρεάν μέσω του World Wide Web.
- Είναι ένας από τους πιο διαδεδομένους Web Servers που κυκλοφορούν στην αγορά λογισμικού.
- Διανέμεται δωρεάν μέσω του World Wide Web.

5.1.7 BPEL4WS engine

Η BPWS4J είναι το κυρίως εργαλείο που χρησιμοποιείται για την εκτέλεση διαδικασιών γραμμένες σε BPEL4WS. Η BPWS4J δέχεται σαν δεδομένα εισόδου τα αρχεία περιγραφής των διαδικασιών και των δικτυακών υπηρεσιών που εμπλέκονται σε αυτές και δημιουργεί τις κατάλληλα δεδομένα, διεπαφές για να γίνει εφικτή η εκτέλεση τους. Η εν λόγω εφαρμογή αναπτύσσεται πάνω σε apache tomcat αλλά έχει δοκιμαστεί επιτυχώς και σε άλλες εφαρμογές διακομιστή όπως το WebSphere Application Server 4.0. Επίσης έχει δοκιμαστεί τόσο σε λειτουργικό σύστημα windows όσο και σε linux.

5.1.8 BPEL4WS Validator

Το συγκεκριμένο εργαλείο έχει σαν κύριο λειτουργικό σκοπό τον έλεγχο ενός κειμένου περιγραφής μιας διαδικασίας γραμμένο σε BPEL4WS για συντακτικά λάθη. Πιο συγκεκριμένα ελέγχει αν όλες οι απαραίτητες παράμετροι είναι ορισμένες και αν οι οντότητες που αναφέρονται και χρησιμοποιούνται στο κείμενο έχουν όντως ορισθεί. Είναι ένα πολύ χρήσιμο εργαλείο στα χέρια του σχεδιαστή των διαδικασιών αφού είναι το μόνο που μπορεί να τον βοηθήσει να ελέγξει έστω και στοιχειωδώς την ορθότητα των διαδικασιών που γράφει.

5.1.9 BPEL4WS Editor Eclipse Plugin

Το συγκεκριμένο εργαλείο είναι ένα Plugin για το εργαλείο ανάπτυξης κώδικα Eclipse της IBM. Σκοπός του είναι να διευκολύνει τη συγγραφή κειμένων περιγραφής BPEL4WS διαδικασιών. Ακολουθώντας τις οδηγίες εγκατάστασης και χρήσης μπορεί κάποιος να διευκολυνθεί αρκετά στην σύνθεση ενός κειμένου BPEL.

5.2 Λεπτομέρειες υλοποίησης

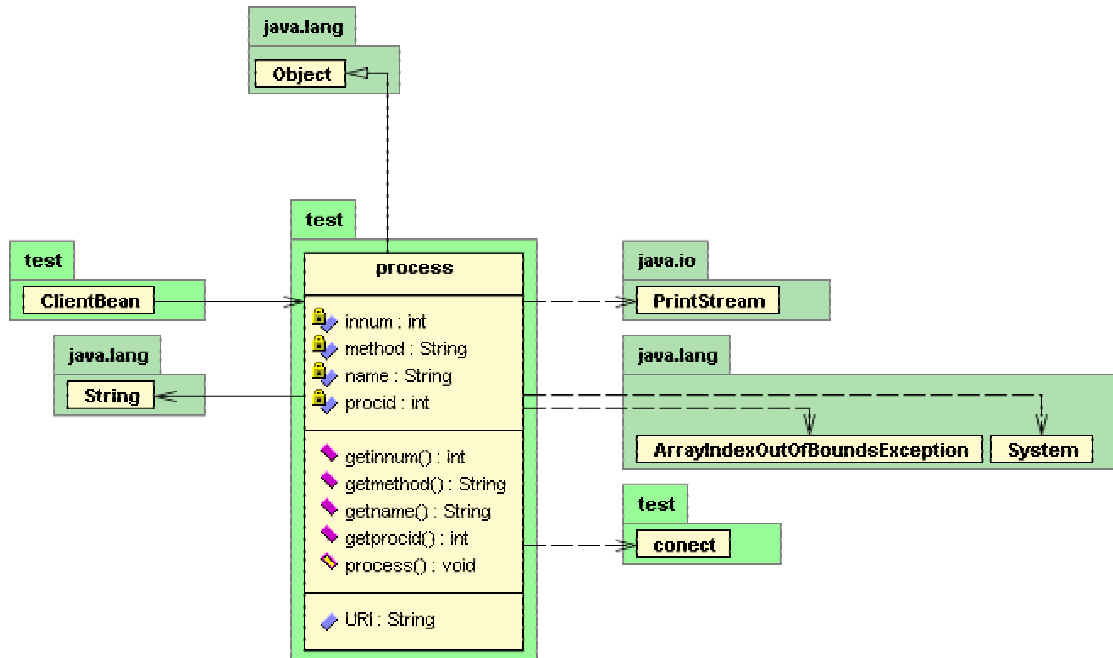
Παρακάτω θα προσπαθήσουμε να κάνουμε μια σύντομη περιγραφή των μονάδων προγραμματισμού που χρησιμοποιήθηκαν στην υλοποίηση του κώδικα της διπλωματικής.

Χωρίζουμε τις μονάδες αυτές στις εξής κατηγορίες. Απλές κλάσεις, JavaBeans και JSP

5.2.1 Classes

5.2.1.1 Class process

Αρχικά παραθέτουμε το UML διάγραμμα της κλάσης process.



Η εν λόγω κλάση περιγράφει μια διαδικασία σύμφωνα με το μοντέλο που παρουσιάσαμε παραπάνω. Η λεπτομερής περιγραφή της κλάσης ακολουθεί παρακάτω χωρίς να περιλαμβάνει λεπτομέρειες υλοποίησης της κάθε συνάρτησης.

- **Ιδιότητες**

Οι παρακάτω μεταβλητές αποτελούν τις ιδιότητες της κλάσης και είναι όλες ιδιωτικές που σημαίνει ότι μόνο τα μέλη της κλάσης έχουν πρόσβαση σε αυτές.

```
private String name; // το όνομα της διαδικασίας
private String URI; // URI της διαδικασίας
private int innum; // αριθμός εισόδων στην διαδικασία
private int procid; // αριθμός ταυτότητας της διαδικασίας
private String method; // μέθοδος που εκτελεί η διαδικασία
```

- **Συναρτήσεις**

Παρακάτω παρουσιάζονται οι συναρτήσεις της κλάσης με σύντομα σχόλια στην κάθε μία για την λειτουργία τους.

```
public process(int id,boolean byid){ }
// κατασκευαστής: επιλέγει κάποια διαδικασία από την βάση δοθέντος του //αριθμού
// ταυτότητάς της ή του αύξοντα αριθμού καταχώρησης

// ακολουθούν συναρτήσεις που επιστρέφουν τις τιμές κάθε μιας ιδιότητας
// η αρχικοποίηση των τιμών αυτών γίνεται από τη συνάρτηση κατασκευαστή που
// αφού επιλέξει την διαδικασία από την βάση διαβάζει και τις αντίστοιχες τιμές.
// Οι τιμές αυτές αρχικοποιούν τις ιδιότητες της κλάσης.
public String getname(){ }
```

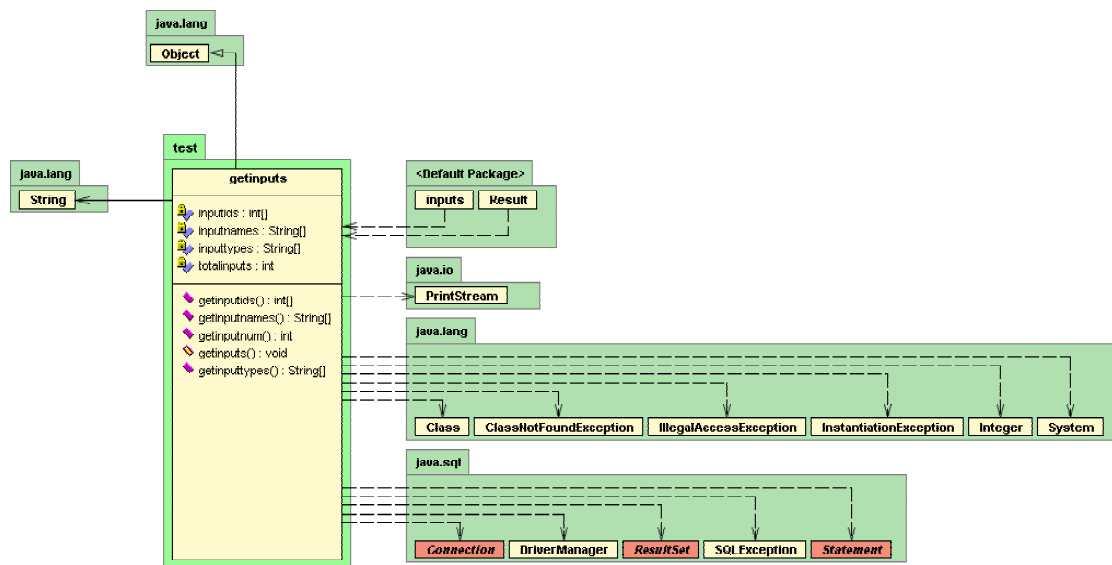
```

public int getprocid(){}
public String getURI(){}
public String getmethod(){}
public int getinum(){}

```

5.2.1.2 Getinputs

Παραθέτουμε παρακάτω το UML διάγραμμα της κλάσης getinputs. Σκοπός της κλάσης είναι να αποσπάσει από την βάση δεδομένων όλα τα απαραίτητα στοιχεία που αφορούν τις εισόδους της επιλεγμένης διαδικασίας.



- **Ιδιότητες**

Οι παρακάτω μεταβλητές αποτελούν τις ιδιότητες της κλάσης και είναι όλες ιδιωτικές που σημαίνει ότι μόνο τα μέλη της κλάσης έχουν πρόσβαση σε αυτές

```

private int totalinputs=0; // Ο συνολικός αριθμός εισόδων της διαδικασίας
private String[] inputnames; // Πίνακας που περιέχει τα ονόματα των εισόδων
private String[] inputtypes; // Πίνακας που περιέχει τους τύπους των εισόδων
private int[] inputids; // Πίνακας που περιέχει τους αριθμούς ταυτότητας των εισόδων

```

- **Συναρτήσεις**

Παρακάτω παρουσιάζονται οι συναρτήσεις της κλάσης με σύντομα σχόλια στην κάθε μία για την λειτουργία τους.

```

// Η συνάρτηση που ακολουθεί είναι ο κατασκευαστής της κλάσης
// Συνδέεται στην βάση και στη συνέχεια αποσπά τις ζητούμενες τιμές
// Οι τιμές αυτές αποδίδονται στην συνέχεια στις εκάστοτε μεταβλητές της κλάσης

```

```

public getinputs(int id) { }

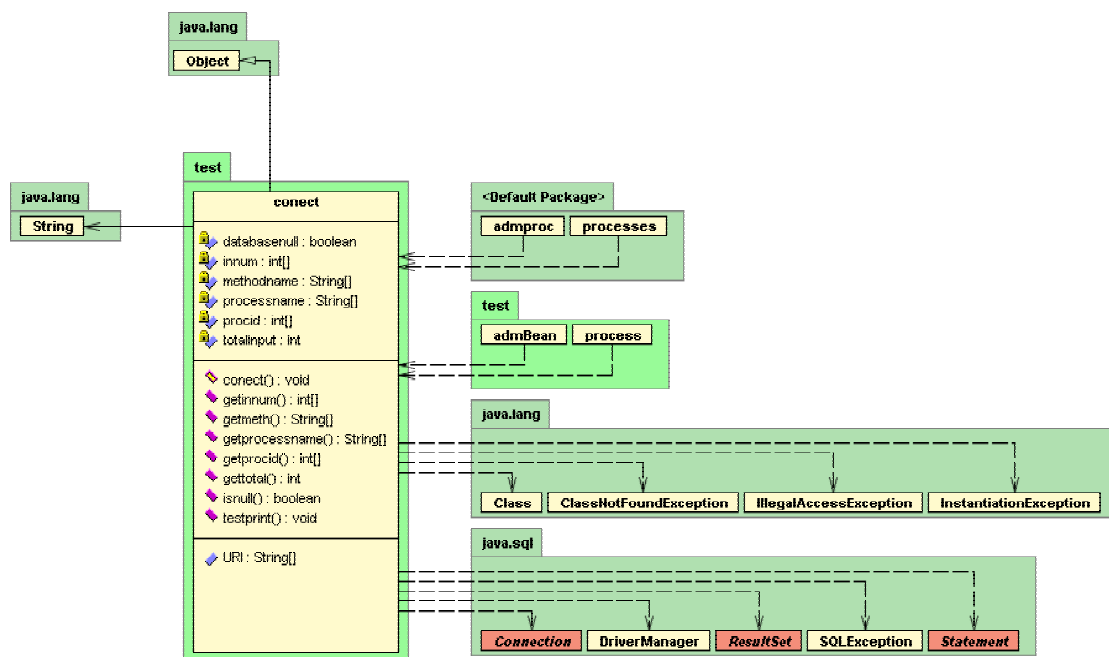
```

//Οι παρακάτω συναρτήσεις επιστρέφουν τις τιμές των ιδιοτήτων της κλάσης και //σκοπό έχουν να ελέγξουν την πρόσβαση στις ιδιότητες αυτές δίνοντας δικαιώματα //ανάγνωσης και μόνο στα αντικείμενα που δεν ανήκουν στην κλάση.

```
public int getinputnum(){}
public int[] getinputids(){}
public String[] getinputnames(){}
public String[] getinputtypes(){}
```

5.2.1.3 Conect

Παρακάτω ακολουθεί το UML διάγραμμα της κλάσης conect. Σκοπός της κλάσης είναι να πραγματοποιεί συνδέσεις στην βάση δεδομένων και να αποθηκεύει όλες τις καταχωρήσεις σε πίνακες. Τους πίνακες αυτού στη συνέχεια τους χρησιμοποιούν οι συναρτήσεις των κλάσεων που περιγράψαμε πιο πάνω για να εντοπίσουν τα ειδικά στοιχεία για κάθε διαδικασία που επιθυμούν.



- **Ιδιότητες**

Οι παρακάτω μεταβλητές αποτελούν τις ιδιότητες της κλάσης και είναι όλες ιδιωτικές που σημαίνει ότι μόνο τα μέλη της κλάσης έχουν πρόσβαση σε αυτές

```
private String[] processname;
// Πίνακας όπου αποθηκεύονται τα ονόματα των διαδικασιών
private String[] URI;
// Πίνακας που αποθηκεύει τις τιμές της URN που αντιστοιχούν σε κάθε διαδικασία
private int[] procid;
// Πίνακας που περιέχει τους αριθμούς ταυτότητας της κάθε διαδικασίας
private String[] methodname;
```

```
// Πίνακας με τα ονόματα των μεθόδων που αντιστοιχούν σε κάθε διαδικασία
private int[] innum;
//Πίνακας με τον αριθμό εισόδων σε κάθε διαδικασία
private int totalinput=0;
//Περιγράφει τον συνολικό αριθμό των διαδικασιών
private boolean databasenull=true;
//ελέγχει αν η βάση είναι κενή
```

- **Συναρτήσεις**

Παρακάτω παρουσιάζονται οι συναρτήσεις της κλάσης με σύντομα σχόλια στην κάθε μία για την λειτουργία τους.

```
// Η συνάρτηση που ακολουθεί είναι ο κατασκευαστής της κλάσης
// Συνδέεται στην βάση και στη συνέχεια αποσπά όλες τις τιμές που είναι
// αποθηκευμένες σε αυτή
// Οι τιμές αυτές αποδίδονται στην συνέχεια στις εκάστοτε μεταβλητές της κλάσης
```

```
public connect() { }
```

//Οι παρακάτω συναρτήσεις επιστρέφουν τις τιμές των ιδιοτήτων της κλάσης και //σκοπό έχουν να ελέγξουν την πρόσβαση στις ιδιότητες αυτές δίνοντας δικαιώματα //ανάγνωσης και μόνο στα αντικείμενα που δεν ανήκουν στην κλάση.

```
public String[] getprocessname(){}
public int[] getprocid(){}
public String[] getURI(){}
public String[] getmeth(){}
public int[] getinum(){}
public int gettotal (){}
public boolean isnull(){}
```

// Η παρακάτω διαδικασία τυπώνει όλες τις τιμές που αποθηκεύονται τελικά στους πίνακες. Έτσι ελέγχουμε τη λειτουργία των διαδικασιών της κλάσης, η ορθότητα των οποίων είναι απαραίτητο στοιχείο για την ορθή λειτουργία όλων των υπόλοιπων διεργασιών στο σύστημα.

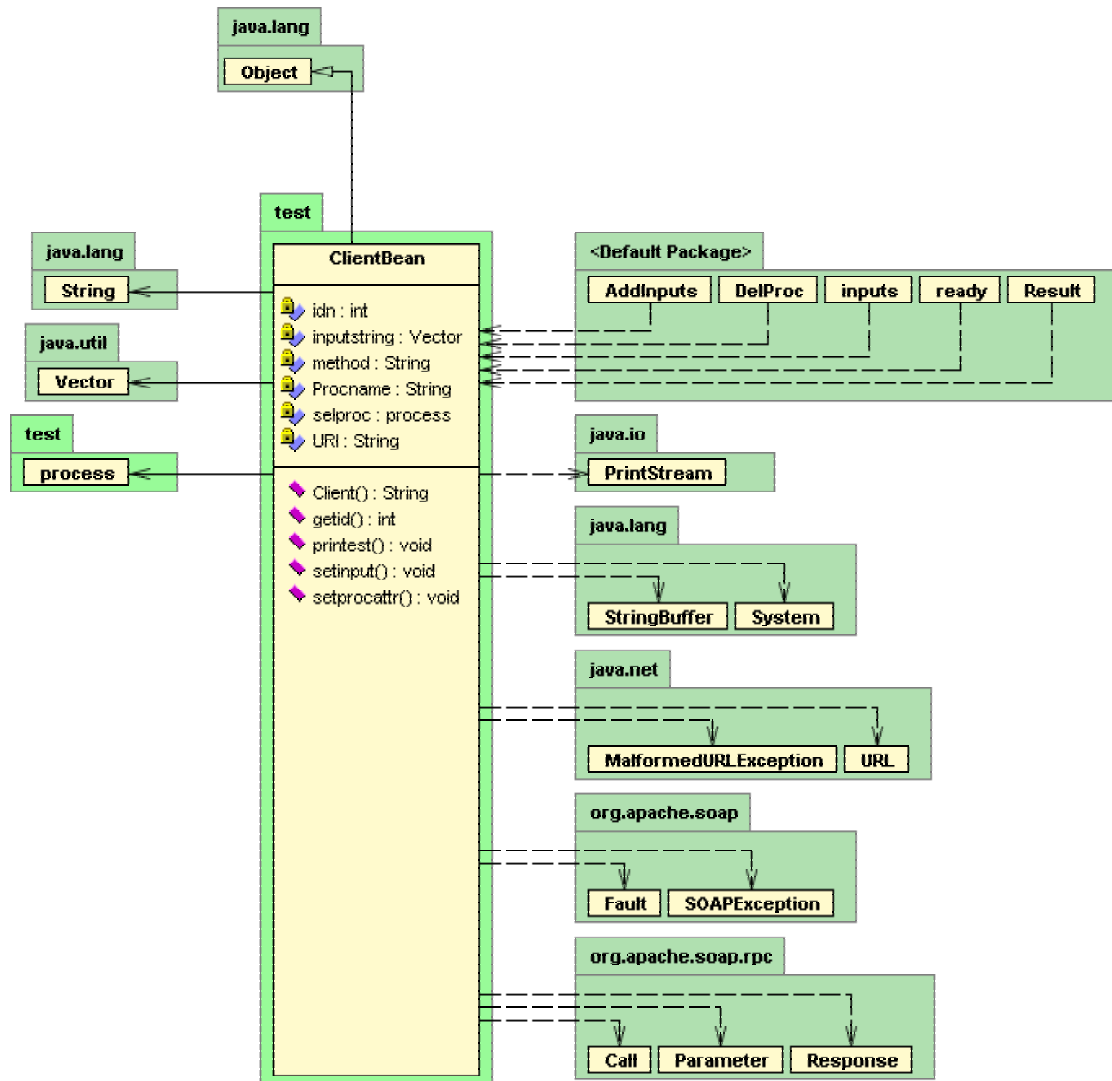
```
public void testprint(){}
```

5.2.2 Beans

Τα Enterprise Java Beans είναι μια τεχνολογία που παρέχεται από την JAVA και αποτελεί κομβικό σημείο στην υλοποίηση του συστήματός μας. Πρόκειται για απλές κλάσεις που παρέχουν όμως μια πολύ ενδιαφέρουσα ιδιότητα όσον αφορά τα Java Server Pages (JSP) που χρησιμοποιούνται για την δημιουργία διεπαφής μέσω διαδικτύου. Εισάγοντας σε ένα JSP τις κατάλληλες γραμμές εντολών, μπορούμε να ορίσουμε ένα σημείο δημιουργίας ενός στιγμιότυπου της κλάσης. Το συγκεκριμένο στιγμιότυπο θα έχει χρόνο ζωής τον χρόνο ζωής του συστήματος διευκολύνοντας έτσι την αποθήκευση και μεταφορά προσωρινών δεδομένων μέσω αυτού. Έτσι η πλοήγηση ανάμεσα στα JSPs γίνεται πολύ εύκολη και το σύστημά μας πολύ ευέλικτο.

Παρακάτω παραθέτουμε μια μικρή αναφορά των Java Beans που χρησιμοποιήσαμε πλαισιωμένη από μια σύντομη περιγραφή για το καθένα.

5.2.2.1 ClientBean



Σκοπός του συγκεκριμένου Bean είναι η εκτέλεση της διαδικασίας που επιλέγουμε κάθε φορά. Παραπάνω φαίνεται το UML διάγραμμα που αντιστοιχεί στο συγκεκριμένο BEAN ενώ ακολουθεί σύντομη περιγραφή του.

- **Ιδιότητες**

Οι παρακάτω μεταβλητές αποτελούν τις ιδιότητες της κλάσης και είναι όλες ιδιωτικές που σημαίνει ότι μόνο τα μέλη της κλάσης έχουν πρόσβαση σε αυτές

```
private String URI="";
// Αποθηκεύει την τιμή του ζητούμενου από την bpel URN της διαδικασίας
private int idn=1000;
// Ο αριθμός ταυτότητας της διαδικασίας
private String Procname="Failure";
// Το όνομα της διαδικασίας.
private Vector inputstring=null;
```

```
// Οι τιμές εισόδου της διαδικασίας τοποθετημένες σε μια μεταβλητή τύπου
//vector(διάνυσμα).
private String method;
// η μέθοδος που καλεί εσωτερικά η BPEL και η οποία ουσιαστικά εκτελεί την //διαδικασία
private process selproc;
// Η ίδια η διαδικασία όπως περιγράφεται από ένα στιγμιότυπο της κλάσης process
// που περιγράψαμε πιο πάνω.
```

- **Συναρτήσεις**

Παρακάτω παρουσιάζονται οι συναρτήσεις της κλάσης με σύντομα σχόλια στην κάθε μία για την λειτουργία τους.

```
//Δεδομένου ενός διανύσματος εισόδου, η συνάρτηση Client εκτελεί στην ουσία την
// επιλεγμένη διαδικασία. Προσοχή πρέπει να δοθεί ότι όλες οι παράμετροι της κλάσης //που
αφορούν την συνάρτηση πρέπει να έχουν αρχικοποιηθεί πριν αρχίσει η //εκτέλεση.
```

```
public String Client() {}
```

```
// Η επόμενη συνάρτηση παίρνει σαν όρισμα τον αριθμό ταυτότητας της διαδικασίας //και με
την βοήθεια προηγούμενων κλάσεων αρχικοποιεί τις κατάλληλες μεταβλητές //της κλάσης
προκειμένου να είναι σε θέση να εκτελέσει την διαδικασία.
```

```
public void setprocattr(int id){}
```

```
// Η συνάρτηση που ακολουθεί αρχικοποιεί την παράμετρο inputstring που //αντιστοιχεί στο
διάνυσμα εισόδου που στο οποίο αναφερθήκαμε πιο πάνω
```

```
public void setinput(Vector input){}
```

```
// Η συνάρτηση που ακολουθεί χρησιμοποιείται για έλεγχο
```

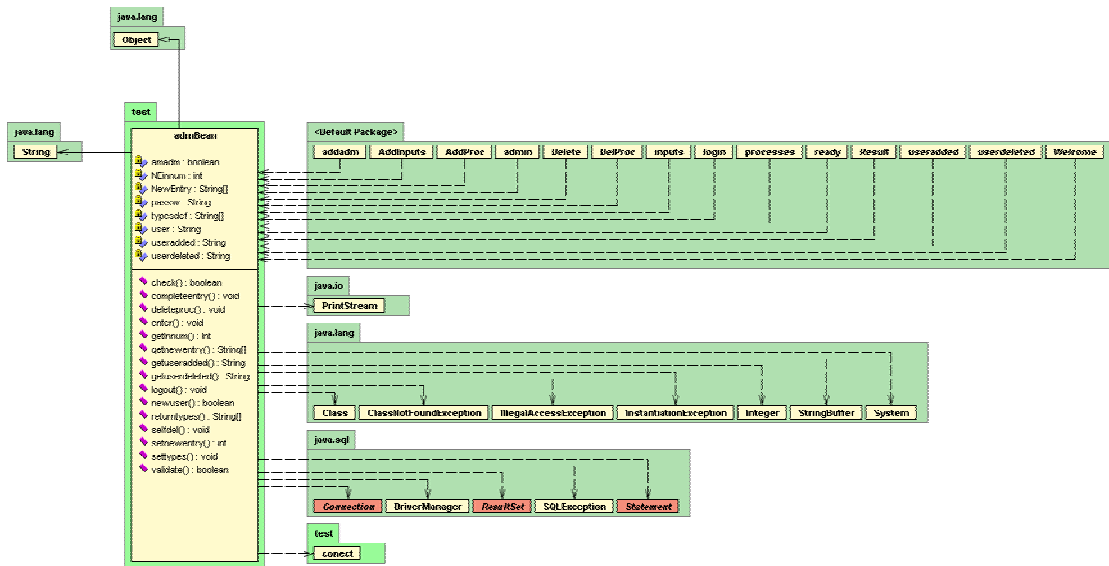
```
public void printest(){}
```

```
// Η παρακάτω συνάρτηση επιστρέφει τον αριθμό ταυτότητας της προς εκτέλεσης
// διαδικασίας
```

```
public int getid(){}
```

5.2.2.2 AdmBean

Το συγκεκριμένο bean υλοποιεί λειτουργίες διαχείρισης που αφορούν τόσο τις αποθηκευμένες διαδικασίες όσο και τους χρήστες του ΚΕΠ. Ακολουθεί το UML διάγραμμα της κλάσης και στη συνέχεια σύντομη περιγραφή των λειτουργιών που υλοποιεί.



- **Ιδιότητες**

Οι παρακάτω μεταβλητές αποτελούν τις ιδιότητες της κλάσης και είναι όλες ιδιωτικές που σημαίνει ότι μόνο τα μέλη της κλάσης έχουν πρόσβαση σε αυτές

// Οι μεταβλητές που ακολουθούν αναφέρονται σε στοιχεία που έχουν σχέση με τους // διαχειριστές του συστήματος και δεν αφορούν τις διαδικασίες.

```
private boolean amadm=false;
// Εφόσον η τιμή της μεταβλητής αυτής είναι true το σύστημα ανανωρίζει τον χρήστη // ως
// διαχειριστή
private String user="";
//το όνομα που περιγράφει το χρήστη στο σύστημα
private String passw="";
// Ο προσωπικός κωδικός του χρήστη
private String useradded="";
// το όνομα του νέου διαχειριστή που θα προστεθεί στο σύστημα
private String userdeleted="";
// το όνομα του διαχειριστή που ζητά να αυτοδιαγραφεί
```

// Οι μεταβλητές που ακολουθούν αφορούν τη νέα διαδικασία που θα προστεθεί στο // σύστημα.

```
private String[] NewEntry;
// πίνακας που περιέχει όλα τα χαρακτηριστικά της νέας διαδικασίας
private int NEinum=0;
// μεταβλητή που αναφέρει τον αριθμό των εισόδων της νέας διαδικασίας
private String[] typesdef;
// πίνακας που καταχωρεί τους τύπους δεδομένων της νέας διαδικασίας
```

- **Συναρτήσεις**

Παρακάτω παρουσιάζονται οι συναρτήσεις της κλάσης με σύντομα σχόλια στην κάθε μία για την λειτουργία τους.

Παρακάτω ακολουθούν διαδικασίες που αφορούν την διαχείριση χρηστών

// Η ακόλουθη συνάρτηση ελέγχει αν τα στοιχεία που δίνει ο χρήστης κατά την
// λειτουργία πιστοποίησης είναι σωστά συνδεδεμένη με την βάση και συγκρίνοντας τα // με
// τις αντίστοιχες καταχωρήσεις.

```
public boolean validate(String usr,String pwd){}
```

// Η παρακάτω συνάρτηση προσθέτει ένα διαχειριστή στο σύστημα

```
public boolean newuser(String u,String p1,String p2){}
```

// Η επόμενη συνάρτηση επιστρέφει την τιμή της παραμέτρου amadmin μέσω της
// οποίας γίνεται ο έλεγχος χρηστών

```
public boolean check(){}
```

// επιστρέφει το όνομα του εκάστοτε διαχειριστή

```
public String getuseradded(){}
```

// Υλοποιεί λειτουργίες αποχώρησης από το σύστημα για ένα χρήστη

```
public void logout(){}
```

// Υλοποιεί την λειτουργία αυτοδιαγραφής

```
public void selfdel(){}
```

// Επιστρέφει το όνομα του αυτοδιαγραφέντος χρήστη

```
public String getuserdeleted(){}
```

Οι συναρτήσεις που ακολουθούν αφορούν την διαχείριση των διαδικασιών.

// Η παρακάτω συνάρτηση υλοποιεί την λειτουργία διαγραφής μιας διαδικασίας

```
public void deleteproc(int pid){}
```

// Θέτει τις τιμές για τις στατικές μεταβλητές μιας διαδικασίας

```
public int setnewentry(String name,String uri,String innum,String met){}
```

// Προσθέτει στον πίνακα των μεταβλητών τα στοιχεία που αφορούν τις εισόδους της //
// διαδικασίας

```
public void completeentry(int tot,String[] inputs){}
```

// Επιστρέφει τον αριθμό των εισόδων της νέας διαδικασίας

```
public int getinnum(){}
```

// Επιστρέφει τον πίνακα με τις μεταβλητές της νέας διαδικασίας

```
public String[] getnewentry(){
```

```
// υλοποιεί την εισαγωγή της διαδικασίας στην βάση δεδομένων
```

```
public void enter(){
```

```
// αποθηκεύει στην κατάλληλη μεταβλητή της κλάσης τους τύπους των εισόδων
```

```
public void settypes(String[] pame){
```

```
// επιστρέφει ένα πίνακα με τους τύπου εισόδου της νέας διαδικασίας
```

```
public String[] returntypes(){
```

6

Έλεγχος

6.1 Μεθοδολογία Ελέγχου

Η μεθοδολογία που έχει χρησιμοποιηθεί είναι η μέθοδος του μαύρου κουτιού. Θα παρουσιαστούν συγκεκριμένα παραδείγματα εκτέλεσης των εφαρμογών του συστήματος ΚΕΠ, ώστε να αποδειχτεί η ορθότητα των λειτουργιών που υλοποιήθηκαν.

6.2 Αναλυτική παρουσίαση έλεγχου

6.2.1 Έλεγχος διαχωρισιμότητας χρηστών

Στην ενότητα αυτή ελέγχουμε το κατά πόσο οι λειτουργίες των διαχειριστών δεν είναι προσβάσιμες στους απλούς χρήστες. Αυτό γίνεται με διαδοχικές προσπάθειες προσέγγισης σελίδων που αφορούν τους διαχειριστές χωρίς να περάσουμε από την διαδικασία πιστοποίηση στοιχείων.

Οι κατάλληλες συναρτήσεις που εφαρμόζονται όμως κάθε φορά σε κάθε σελίδα του συστήματος των διαχειριστών μας παραπέμπουν συνεχώς στην σελίδα πιστοποίηση στοιχείων. Έτσι σε πρώτη φάση ο έλεγχος αποδεικνύει την ύπαρξη μιας πρώτης μορφής ασφάλειας.

Θα μπορούσαμε να έχουμε καλύτερη ασφάλεια αν επιβάλαμε και κρυπτογράφηση στα προσωπικά στοιχεία του κάθε διαχειριστή. Κάτι τέτοιο όμως είναι αρκετά εξεζητημένο για μια εφαρμογή σαν την δική μας.

6.2.2 Έλεγχος λειτουργιών

Παρακάτω παρουσιάζεται ένα σενάριο χρήσης του ΚΕΠ. Ένας διαχειριστής αφού πιστοποιήσει τα στοιχεία του ακολουθεί τις κατάλληλες διαδικασίες και εισάγει μια διαδικασία στο ΚΕΠ. Στην συνέχεια εφαρμόζει την λειτουργία logout και επανέρχεται στο σύστημα σαν απλός χρήστης. Ακολουθώντας επιλέγει την διαδικασία που προστέθηκε και την εκτελεί.


Ξεκινώντας βήμα προς βήμα την παραπάνω διαδικασία ανοίγουμε την αρχική σελίδα του ΚΕΠ.

New Page 2 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Refresh Stop

Address <http://localhost:8083/main.jsp>

 ΚΕΝΤΡΟ ΕΞΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ

[Original KEP Page](#)

[Process Description](#)


[Loan Application](#)

[Echo](#)

[Useful Links](#)

[Ministries](#)

[Other Public Services](#)



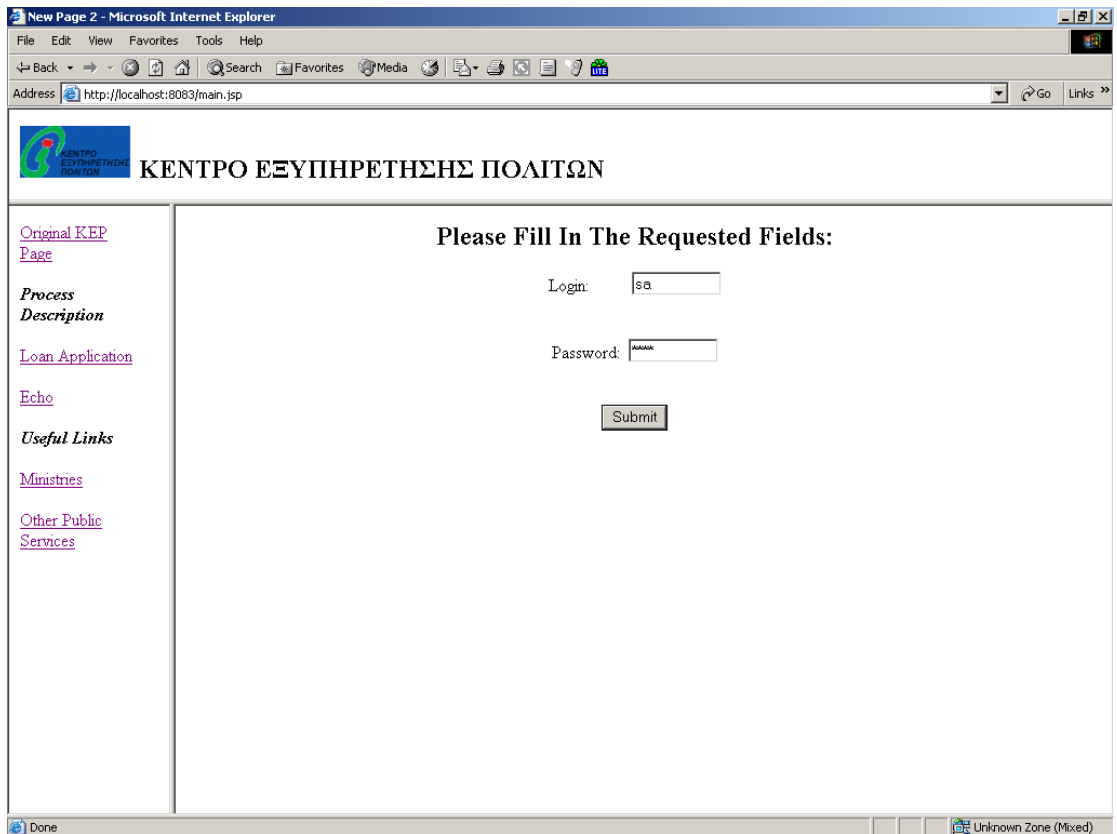
Your Options:

Use The Services Of Kep [click here](#)

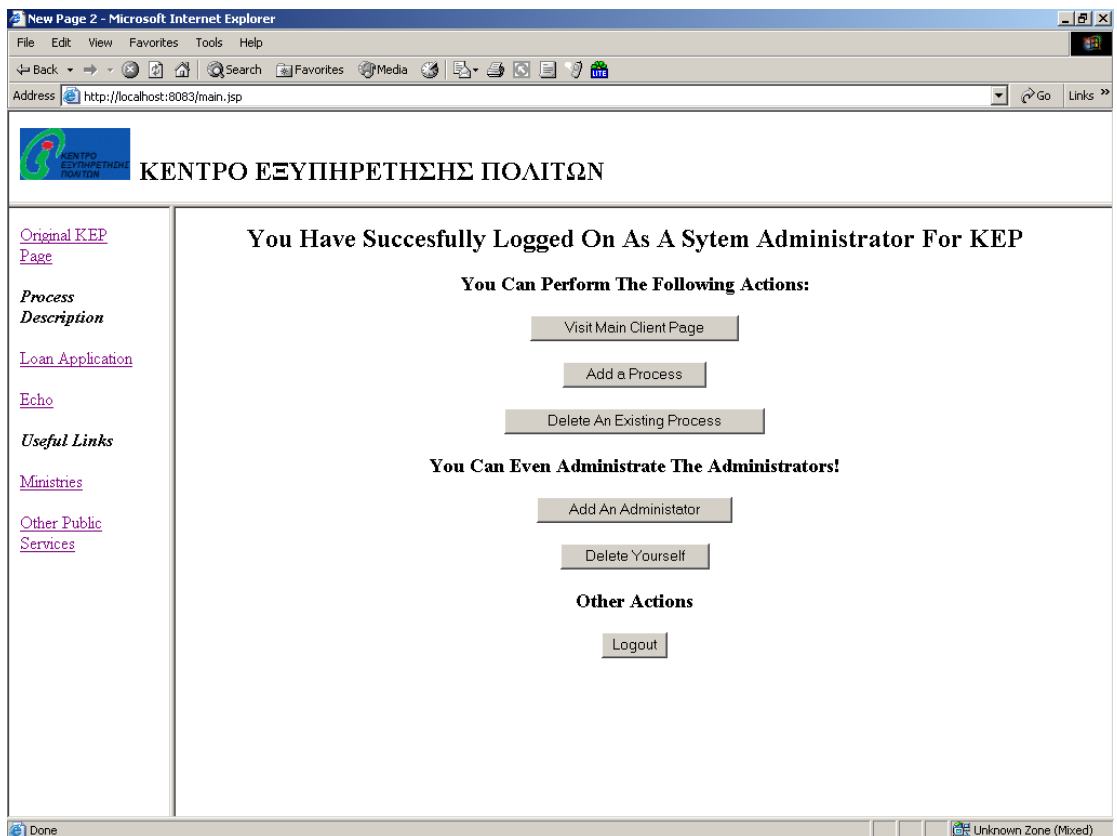
Login As An Administrator of Kep [click here](#)

Done Unknown Zone

Επιλέγουμε την δεύτερη επιλογή που μας παρέχεται και ζητάμε να πιστοποιήσουμε την ταυτότητά μας ως διαχειριστές του ΚΕΠ



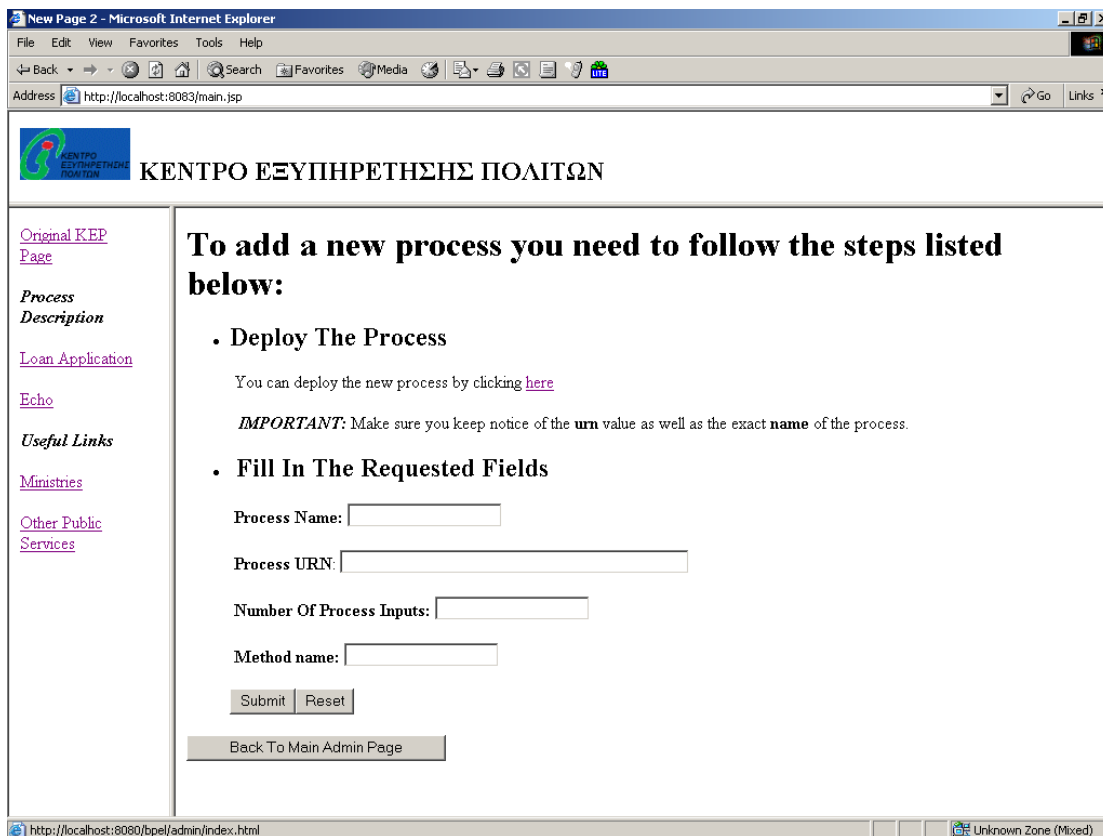
Αφού εισάγουμε τα ζητούμενα πεδία στη σελίδα ζητάμε να ολοκληρωθεί η πιστοποίηση πατώντας το κουμπί Submit.



Η πιστοποίηση ήταν επιτυχής και το σύστημα μας οδηγεί στην κεντρική σελίδα των διαχειριστών που φαίνεται παραπάνω.

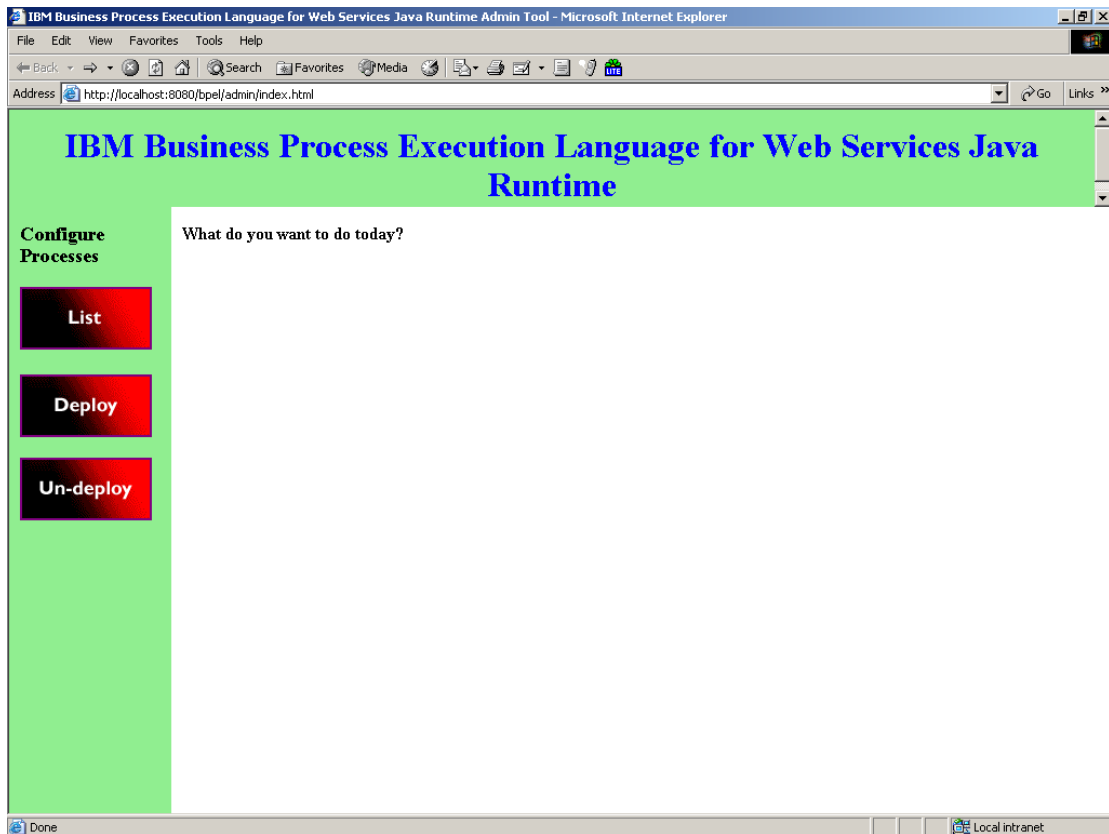
Από την σελίδα αυτή επιλέγουμε την επιλογή «Add A Process» που αντιστοιχεί στη λειτουργία προσθήκης

Το σύστημα μας οδηγεί ευθύς αμέσως στην ακόλουθη σελίδα όπου μας ζητείται να εισάγουμε τα πεδία που περιγράφουν την νέα διαδικασία

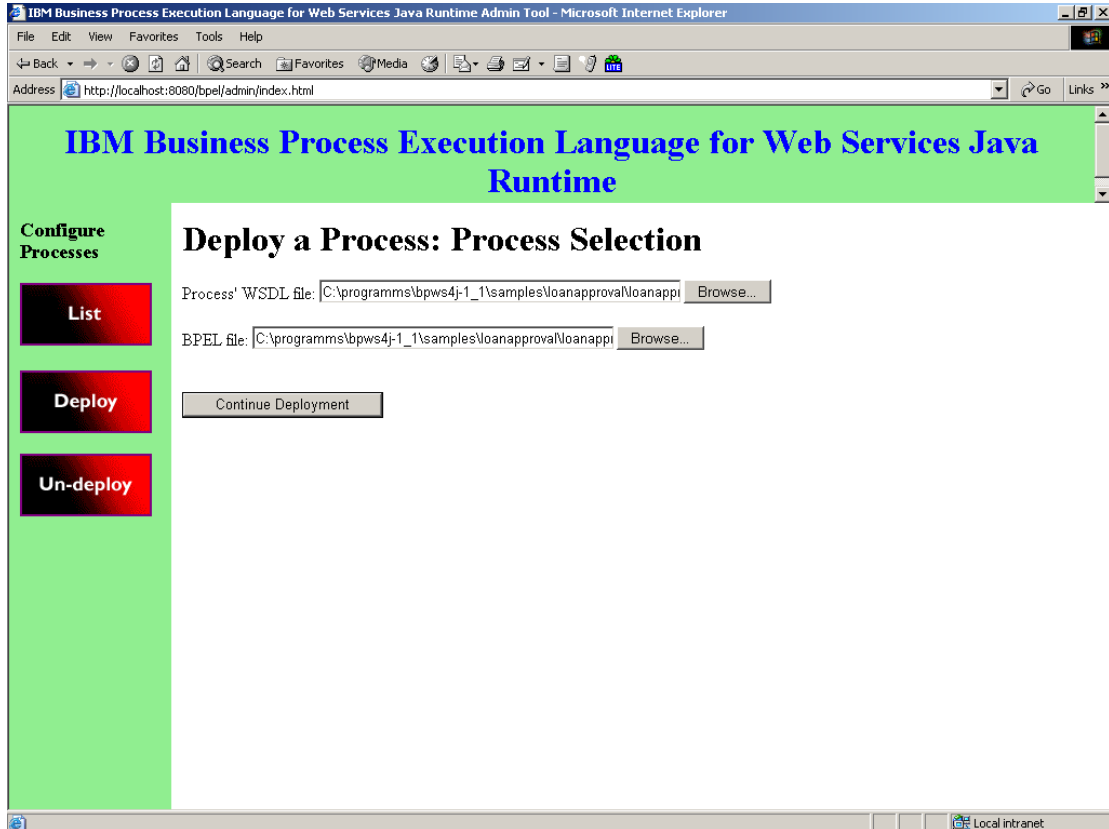


The screenshot shows a Microsoft Internet Explorer browser window displaying a web application. The address bar shows 'http://localhost:8083/main.jsp'. The page title is 'ΚΕΝΤΡΟ ΕΞΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ'. The main content area is titled 'To add a new process you need to follow the steps listed below:'. It contains two main sections: 'Deploy The Process' and 'Fill In The Requested Fields'. The 'Fill In The Requested Fields' section includes input fields for 'Process Name', 'Process URN', 'Number Of Process Inputs', and 'Method name'. There are 'Submit' and 'Reset' buttons, and a 'Back To Main Admin Page' button. A sidebar on the left contains links for 'Original KEP Page', 'Process Description', 'Loan Application', 'Echo', 'Useful Links', 'Ministries', and 'Other Public Services'. The status bar at the bottom shows 'http://localhost:8080/bpel/admin/index.html' and 'Unknown Zone (Mixed)'.

Πριν όμως γίνει η εισαγωγή των πεδίων πρέπει πρώτα να δημιουργήσουμε την εφαρμογή στον διακομιστή όπου έχουμε τοποθετήσει το RPCrouter. Για το σκοπό αυτό ακολουθούμε τον υποδεικνυόμενο σύνδεσμο και δημιουργείται αμέσως το ακόλουθο παράθυρο:

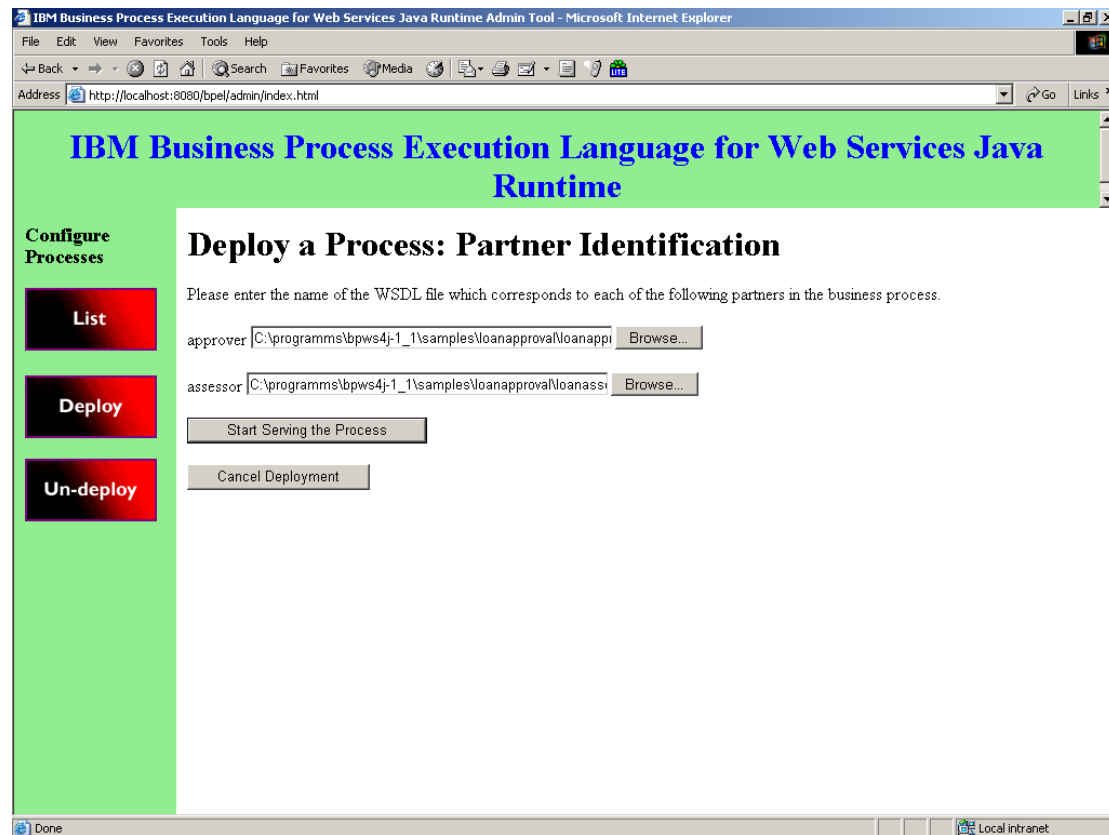


Επιλέγουμε την επιλογή Deploy και συνεχίζουμε.

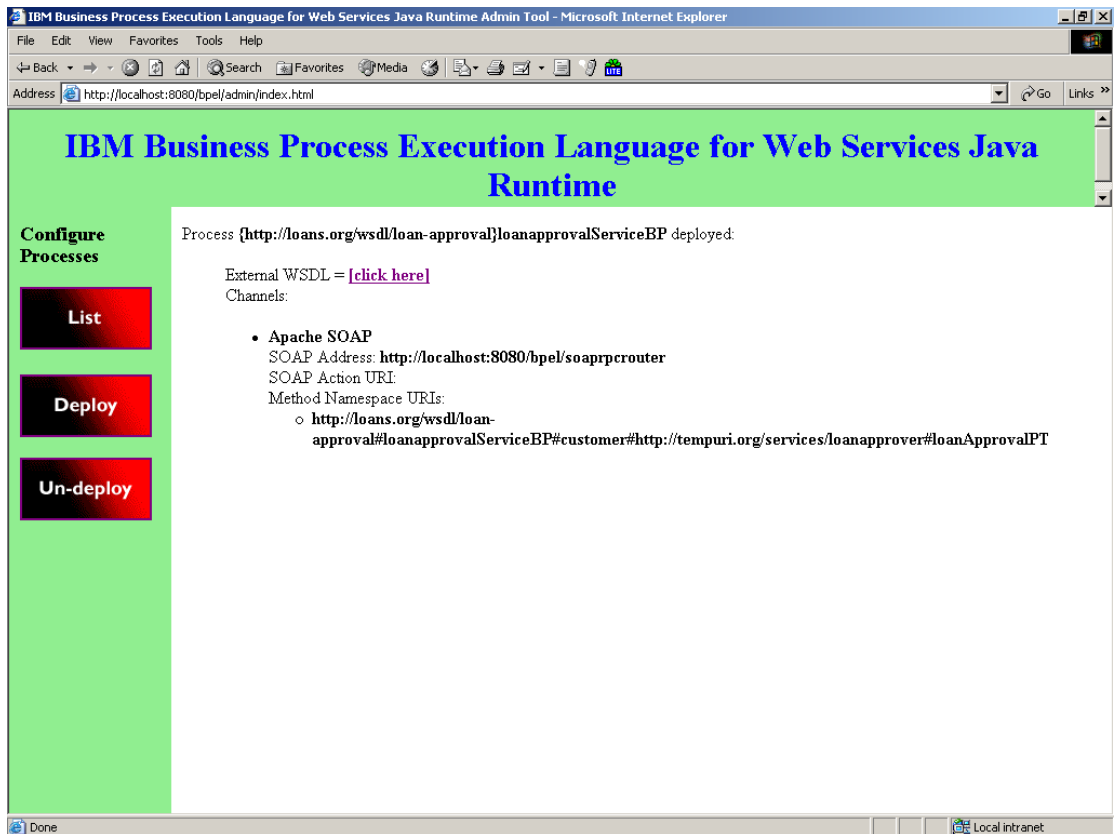


Αμέσως μας ζητούνται δύο πεδία. Η διεύθυνση του WSDL αρχείου που περιγράφει την διαδικασία και η διεύθυνση του BPEL αρχείου που περιγράφει τις μεθόδους της και την υλοποίησή τους.

Εφόσον η εισαγωγή γίνεται σωστά η BPEL4WS engine μας ζητά τα WSDL αρχεία των δικτυακών υπηρεσιών που εμπλέκονται στην διαδικασία την διεύθυνση των οποίων και παρέχουμε αμέσως.



Συνεχίζοντας με την επιλογή «Start Deploying the Process» η εφαρμογή δημιουργείται αμέσως και σαν αποτέλεσμα έχουμε την τιμή του URN που θα χρειαστούν στην συνέχεια.



Τώρα μπορούμε να συνεχίσουμε την διαδικασία προσθήκης, εισάγοντας το URN που πήραμε από την BPEL4WS engine καθώς και τα πεδία που αντιστοιχούν στο όνομα με το οποίο θέλουμε να φαίνεται η διαδικασία στο σύστημα, των αριθμό εισόδων και την μέθοδο της διαδικασίας που θα εκτελείται.

New Page 2 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://localhost:8083/main.jsp>

ΚΕΝΤΡΟ ΕΞΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ

[Original KEP Page](#)

Process Description

[Loan Application](#)

[Echo](#)

Useful Links

[Ministries](#)

[Other Public Services](#)

To add a new process you need to follow the steps listed below:

- **Deploy The Process**

You can deploy the new process by clicking [here](#)

IMPORTANT: Make sure you keep notice of the **urn** value as well as the exact **name** of the process.

- **Fill In The Requested Fields**

Process Name:

Process URN:

Number Of Process Inputs:

Method name:

Start | photos.doc ... | Tomcat | Builder 8 - ... | New Page ... | C:\program... | my.ntua.gr ... | Customer.js... | 3:41 πμ

Με βάση τα νέα δεδομένα το σύστημα μας ζητά όνομα και τύπο δεδομένων για κάθε είσοδο.

New Page 2 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://localhost:8083/main.jsp>

ΚΕΝΤΡΟ ΕΞΥΠΗΡΕΤΗΣΗΣ ΠΟΛΙΤΩΝ

[Original KEP Page](#)

Process Description

[Loan Application](#)

[Echo](#)

Useful Links

[Ministries](#)

[Other Public Services](#)

Please Insert The Parameters of Every Input:

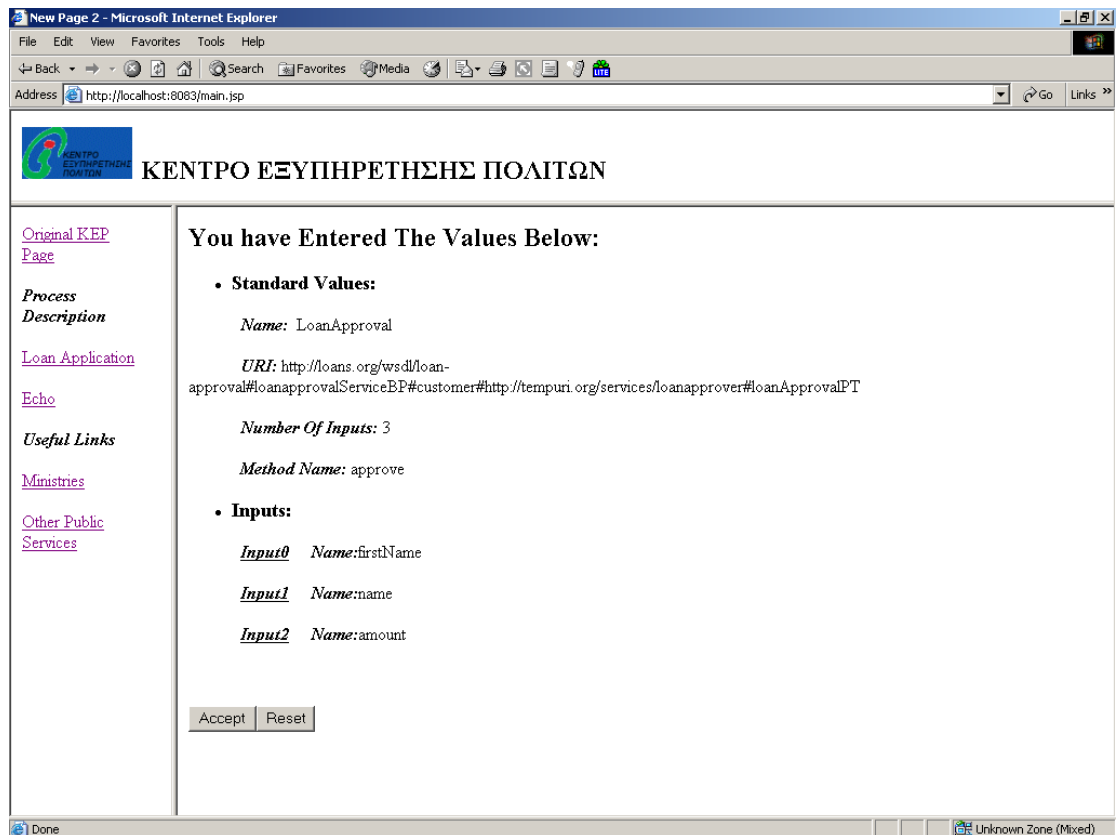
Input0: Name TypeOf

Input1: Name TypeOf

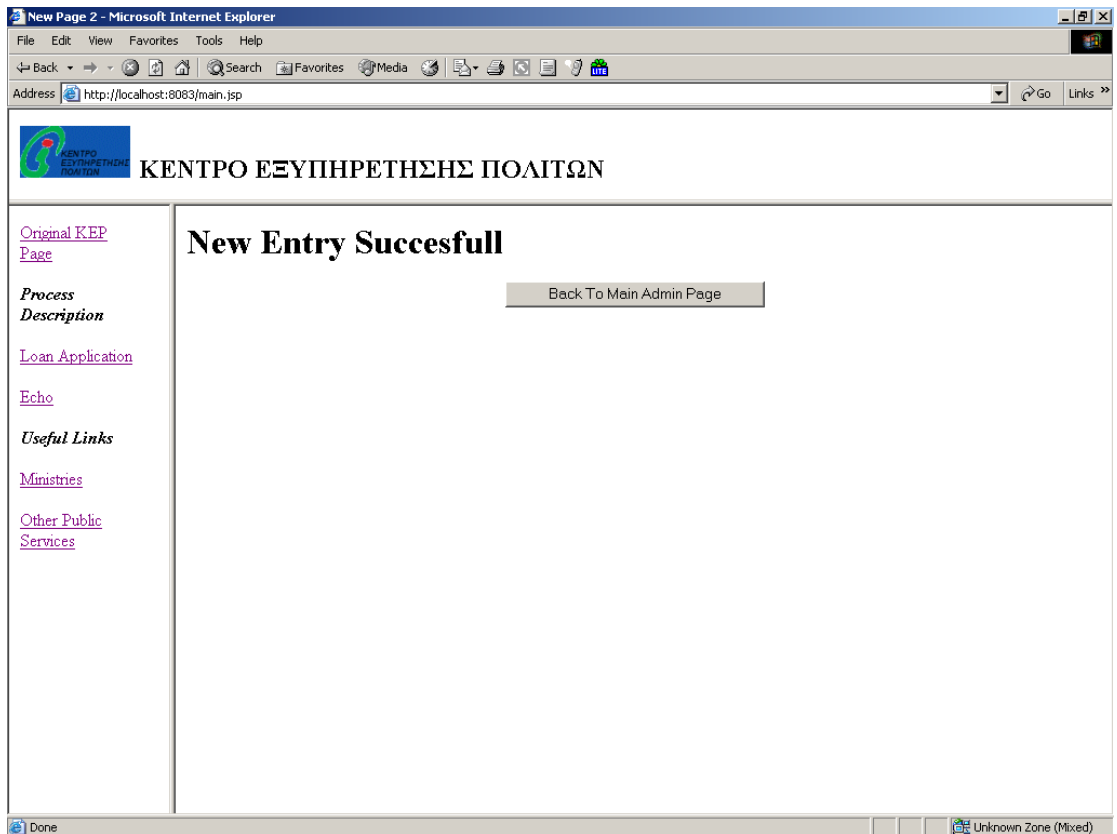
Input2: Name TypeOf

Done | Unknown Zone (Mixed)

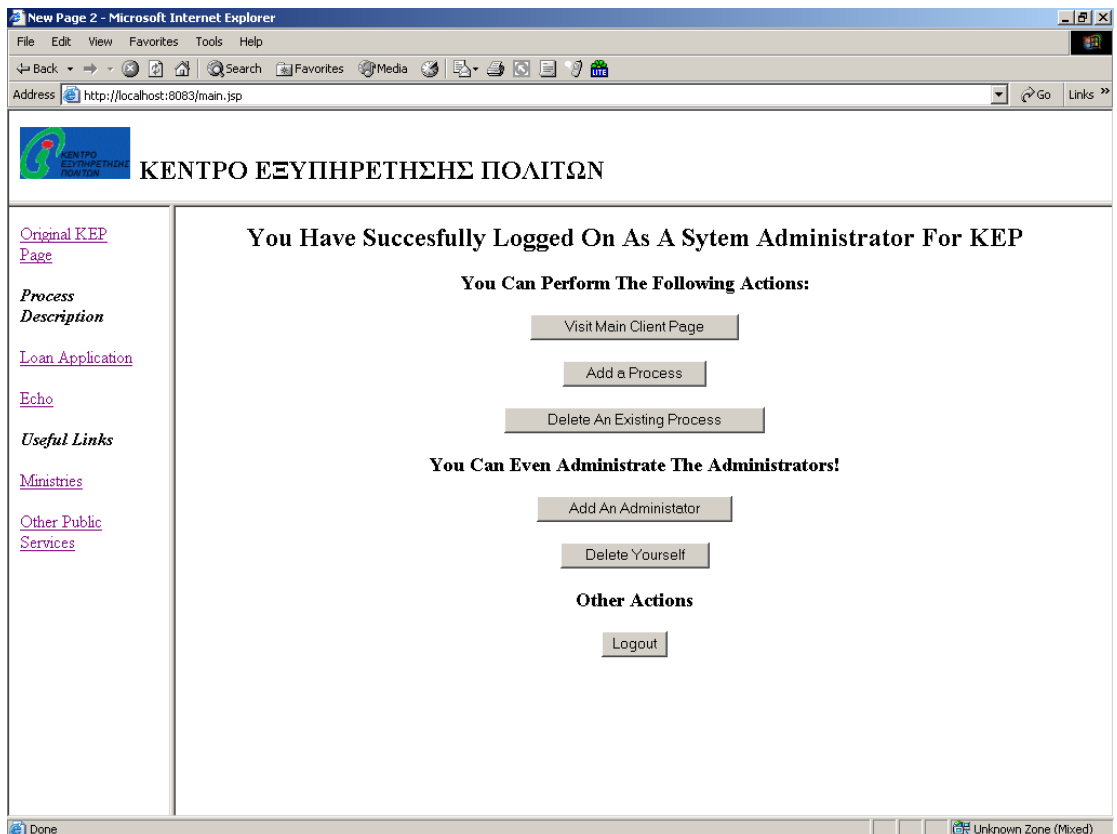
Πριν προχωρήσει στην καταχώρηση της διαδικασίας μας ζητάτε να ελέγξουμε αν οι τιμές που βάλουμε είναι σωστές παρουσιάζοντας όλες τις νεοεισελθούσες τιμές.



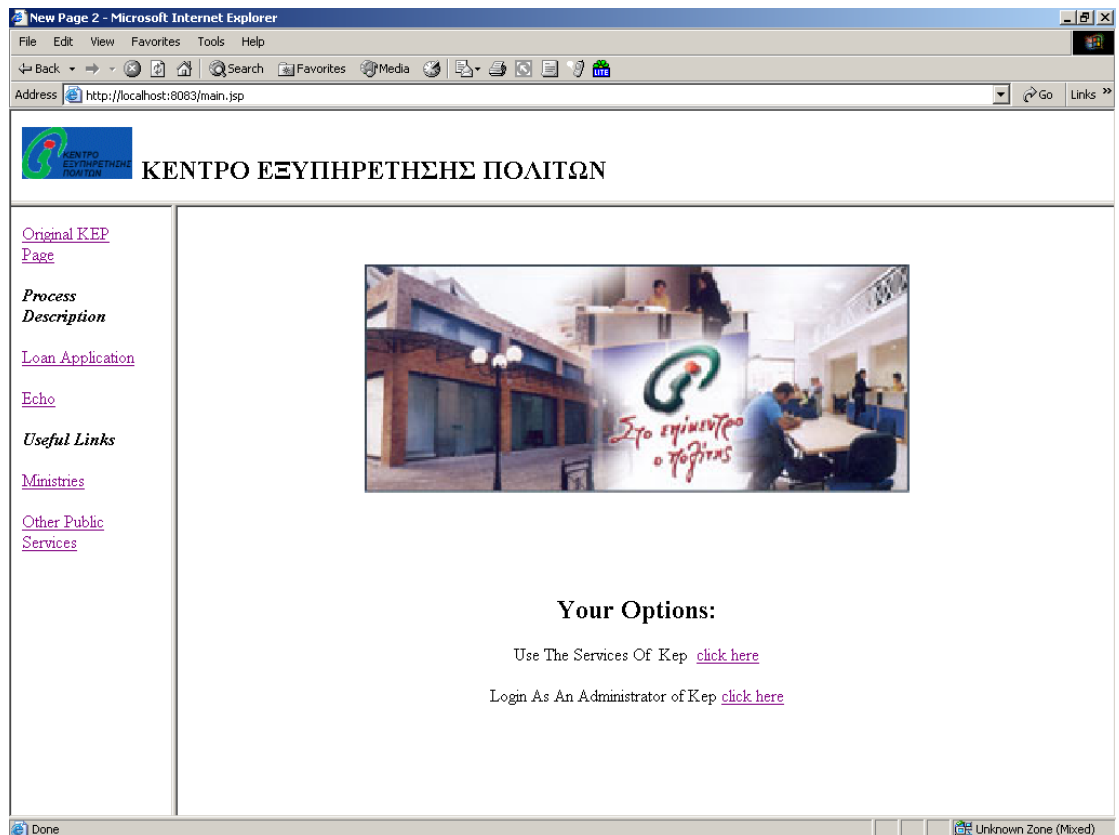
Ακολουθεί η επιβεβαίωση της ολοκλήρωσης της εγγραφής.



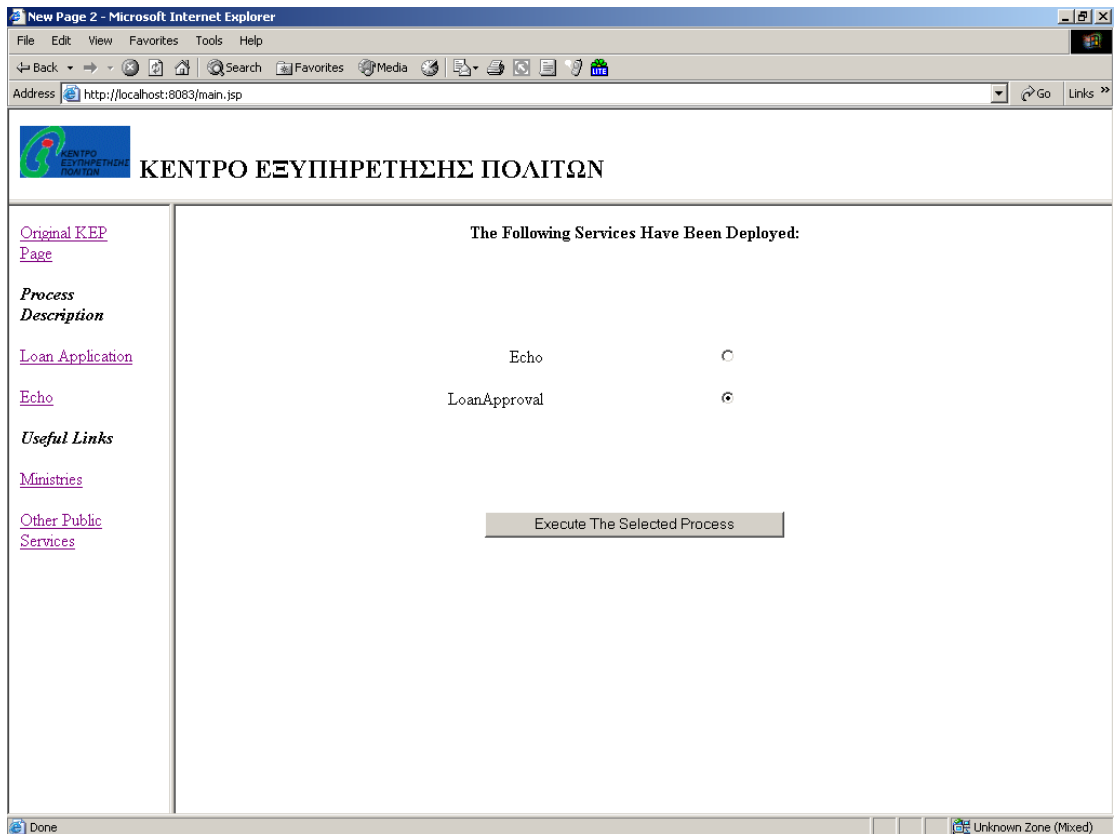
Ακολουθώντας τον κατάλληλο σύνδεσμο οδηγούμαστε και πάλι στην αρχική σελίδα των διαχειριστών.



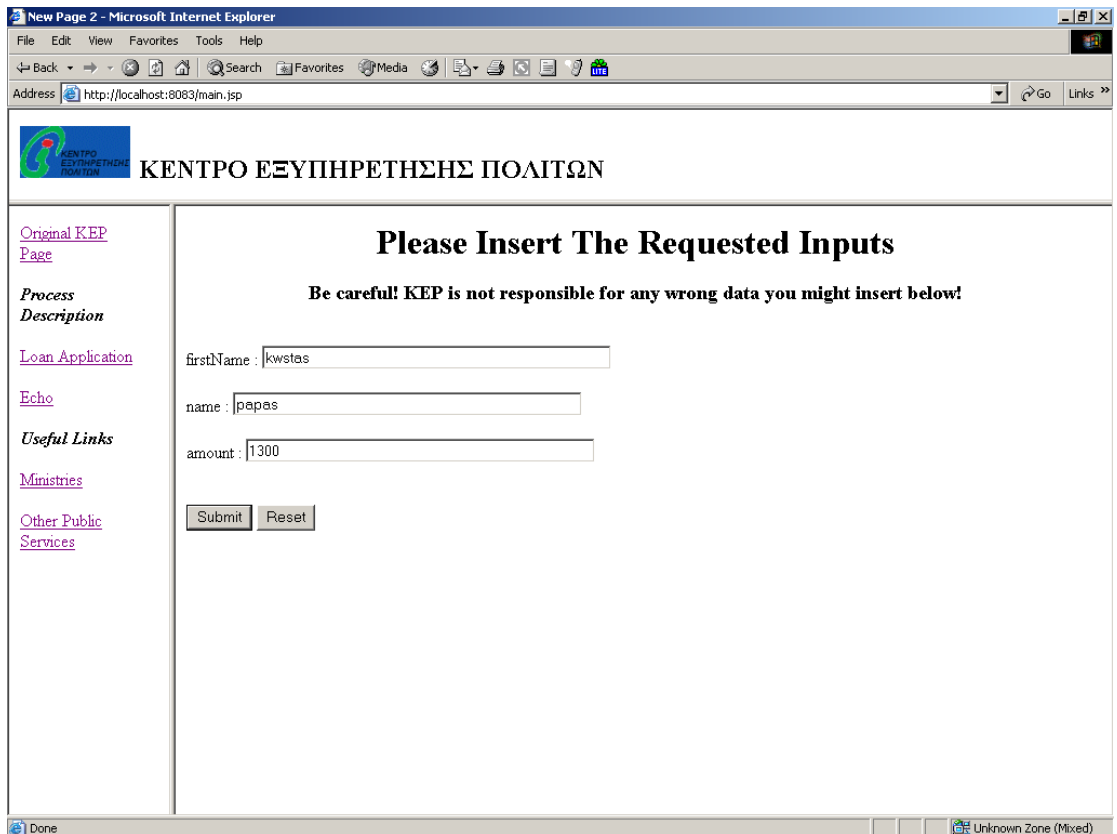
Με την επιλογή «Logout» οδηγούμαστε στην αρχική σελίδα απ' όπου επιλέγουμε να συνεχίσουμε σαν απλοί χρήστες.



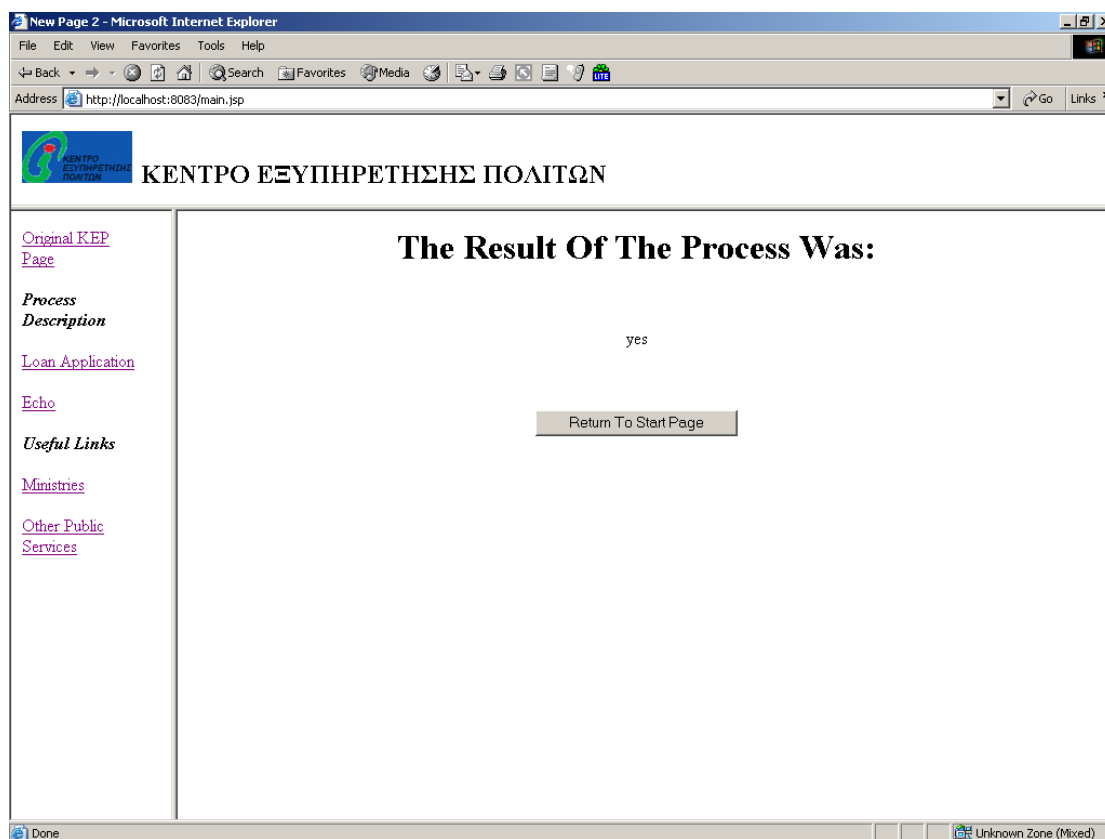
Το σύστημα μας οδηγεί στην κεντρική σελίδα των απλών χρηστών όπου βλέπουμε την καινούργια διαδικασία στην διάθεσή μας προς εκτέλεση.



Επιλέγοντάς την και ζητώντας την εκτέλεσή της το σύστημα ζητά από εμάς να εισάγουμε τιμές για κάθε είσοδο της διαδικασίας.



Εφόσον όλες οι επιμέρους ενέργειες έγιναν σωστά, δεν αργεί να εμφανιστεί και το αντίστοιχο με τις τιμές εισόδου αποτέλεσμα.



6.3 Συμπέρασμα

Όπως βλέπουμε οι λειτουργίες που εξετάσαμε λειτουργούν άψογα και δεδομένου ότι περιλαμβάνουν τα πιο επικίνδυνα για λάθος σημεία της εκτέλεσης μπορούμε να συμπεράνουμε πως ο έλεγχος ολοκληρώθηκε επιτυχώς.

7

Επίλογος

7.1 Σύννοψη και συμπεράσματα

Παρατηρώντας το σύστημα του Κ.Ε.Π. όπως υλοποιήθηκε στην συγκεκριμένη διπλωματική μπορούμε να εξάγουμε πολλά συμπεράσματα. Οι αρχικές διαπιστώσεις είναι ότι το σύστημά μας έγινε αρκετά ευέλικτο, εύκολα αναβαθμίσιμο και αρκετά εύκολο στην χρήση του. Γενικά μπορεί να πει κανείς ότι το Κέντρο Εξυπηρέτησης Πολιτών μπορεί πλέον να προσφέρει στον πολίτη γρήγορη και άμεση εξυπηρέτηση παρακάμπτοντας πολλές χρονοβόρες γραφειοκρατικές διαδικασίες, αντικαθιστώντας τες με νέες άμεσες και αυτοματοποιημένες πολύ πιο γρήγορες και ακριβείς. Η θεαματική αυτή βελτίωση οφείλεται σε δύο κυρίως λόγους.

Πρώτος παράγοντας βελτίωσης είναι αναμφισβήτητα η χρήση της τεχνολογίας των δικτυακών υπηρεσιών. Χάρη στην τεχνολογία αυτή, απομονώσαμε τις επιμέρους διαδικασίες, τις κάναμε αυτόνομες, γρήγορες στην υλοποίηση και ποιο ακριβείς. Ο ανθρώπινος παράγοντας που αποτελούσε αιτία ανακρίβειας, καθυστέρησης στην εξυπηρέτηση και επιβράδυνσης των διαδικασιών περιορίστηκε σε ρόλο επίβλεψης και μόνο. Μόνο η χρήση της τεχνολογίας αυτής αρκούσε σε πρώτο βαθμό για την υλοποίηση ενός γρήγορου και ευέλικτου συστήματος.

Σε δεύτερο επίπεδο η ενιαία περιγραφή των ροών εκτέλεσης δημιούργησε ένα σύστημα εύκολο στο σχεδιασμό, στην αναβάθμιση και στην αντιμετώπιση των δυσλειτουργιών. Το σύστημα εκτέλεσης είναι ανεξάρτητο των διαδικασιών που εκτελεί και έτσι αναβαθμίζοντας τις διαδικασίες αναβαθμίζονται και οι παρεχόμενες υπηρεσίες.

Είναι προφανές ότι ο συνδυασμός των δύο τεχνολογιών μπορεί να οδηγήσει σε συστήματα με δυνατότητα παροχής υπηρεσιών άμεσων και επεκτάσιμων. Πρόκειται για τεχνολογίες που μπορεί να αλλάξουν τα δεδομένα στο χώρο των ηλεκτρονικών επιχειρήσεων και υπηρεσιών.

7.2 Μελλοντικές επεκτάσεις

Όπως μπορούμε να παρατηρήσουμε από το σύστημα που υλοποιήσαμε, το σύστημα δεν είναι άμεσα συσχετισμένο με την λειτουργία του ΚΕΠ. Η άμεση συσχέτιση αφορά τις διαδικασίες μόνο που εκτελούνται από το σύστημα. Έτσι μπορούμε να θεωρήσουμε σαν μελλοντική

επέκταση τη δημιουργία ενός συστήματος εντελώς αποδεδουλευμένου των προς εκτέλεση διαδικασιών. Ο χρήστης θα μπορεί να εισάγει τα στοιχεία του στο σύστημα και στη συνέχεια να αναπτύσσει διαδικασίες που αποθηκεύονται σε αυτό και γίνονται έτοιμες προς εκτέλεση από όποιον χρήστη επιλέξει αυτός. Οι δυνατότητες ενός τέτοιου συστήματος είναι απεριόριστες αφού το σύστημα θα προσαρμόζεται στα δεδομένα του κάθε χρήστη όχι μόνο σε επίπεδο εκτέλεσης αλλά και σχεδίασης.

ΠΑΡΑΡΤΗΜΑ: ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ

Στο συγκεκριμένο παράρτημα δίνονται συγκεκριμένες οδηγίες για την εγκατάσταση του προγράμματός μας. Παρόλο που η εφαρμογή μπορεί να εκτελεστεί αυτόνομα, εμείς δίνουμε τις κατάλληλες οδηγίες προκειμένου να εκτελεστεί μέσω του JBUILDER. Αυτό γίνεται προκειμένου να ελέγχεται καλύτερα η εκτέλεση η οποία γίνεται πάνω σε δύο στιγμιότυπα του apache tomcat. Ο JBUILDER διαχειρίζεται τα δύο αυτά στιγμιότυπα με αρκετή ευκολία. Προκειμένου να εκτελεστεί η εφαρμογή ακολουθούμε τα εξής βήματα:

1. Εγκατάσταση του JBUILDER ο οποίος δίνεται και στο cd που συνοδεύει την διπλωματική εργασία.
2. Από το ίδιο cd εγκαθιστούμε τον Apache Tomcat 1.4
3. Εγκαθιστούμε το BPEL4J engine από το cd ακολουθώντας πιστά τις οδηγίες που εμπεριέχονται στο zip αρχείο. Σαν server επιλέγουμε τον apache tomcat που εγκαταστήσαμε στο προηγούμενο βήμα.
4. Προαιρετικά εγκαθιστούμε και τα samples της BPEL4J engine στον ίδιο server
5. Εγκαθιστούμε και το jbproject που περιέχεται στον φάκελο source σε σημείο όπου να είναι «ορατό» από τον JBUILDER.
6. Εγκαθιστούμε την MySql και εισάγουμε το SQL script περιέχεται στον φάκελο source.
7. Εκτελούμε το jbproject από τον JBUILDER σε διαφορετικό tomcat από αυτόν που εγκαταστήσαμε την BPEL4J engine.
8. Από ένα browser ανοίγουμε την κεντρική σελίδα main.jsp ή Welcome.jsp

8

Βιβλιογραφία

[url1]	http://www.unece.org/cefact/
[url2]	http://www.ebxml.org/specs/ebBPSS.pdf
[url3]	http://www.rosettanet.org/
[url4]	Specification: http://www-106.ibm.com/developerworks/library/ws-bpel/
[url5]	Specification Web Services Description Language (WSDL) 1.1
[url6]	uddi.org
[FC02]	FRANK P. COYLE, XML, 'WEB SERVICES, AND THE DATA REVOLUTION', BOSTON, MARCH 2002
[GSB+03]	GRAHAM, SIMEONOV, BOUDEZ, DAVIS, DANIELS, NAKAMURA, NEYAMA, 'BUILDING WEB SERVICES WITH JAVA', 2003