

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
Π.Μ.Σ. ΔΙΔΑΚΤΙΚΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΨΗΦΙΑΚΑ
ΣΥΣΤΗΜΑΤΑ
ΚΑΤΕΥΘΥΝΣΗ ΔΙΚΤΥΟΚΕΝΤΡΙΚΑ ΣΥΣΤΗΜΑΤΑ

**Ανάλυση, Σχεδιασμός και Ανάπτυξη
Ηλεκτρονικής Υπηρεσίας**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Χαϊντίνη Ασημίνας

Επιβλέπων: Μ. Θεμιστοκλέους
Επίκουρος Καθηγητής

Πειραιάς, 2011

РАНЕЕЗНАМО ПЕРПАА

.....
ΧΑΪΝΤΙΝΗ ΑΣΗΜΙΝΑ

Διπλωματούχος Π.Μ.Σ. Διδακτική της Τεχνολογίας και Ψηφιακά
Συστήματα Πανεπιστημίου Πειραιώς

Copyright © Χαϊντίνη Ασημίνα, 2011

Με επιφύλαξη παντός δικαιώματος. All rights reserved

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Πρόλογος

Η διπλωματική εργασία που τεκμηριώνεται στο παρόν έγγραφο έχει ως αντικείμενο τη σχεδίαση, ανάπτυξη και ολοκλήρωση ενός πληροφοριακού συστήματος με χρήση ηλεκτρονικών υπηρεσιών (web services).

Ολοκληρώνοντας τη διπλωματική εργασία, θα ήθελα να ευχαριστήσω θερμά τον κ. Μαρίνο Θεμιστοκλέους, επιβλέποντα καθηγητή, για την ανάθεση της εργασίας και τη συνεχή του επίβλεψη. Επίσης θα ήθελα να ευχαριστήσω τους συναδέλφους μου Γεώργιο Λαμπράκη, Βαγγέλη Μαρκέτη, Χρήστο Μπούρα και Σοφία Αναγνωστοπούλου καθώς και τον προϊστάμενό μου Αντώνη Αλεξάκη. Τέλος, γιώθω την ανάγκη να ευχαριστήσω την οικογένειά μου και τους φίλους μου για τη στήριξη και τη συμπαράστασή τους τόσο στη διάρκεια εκπόνησης της διπλωματικής εργασίας όσο και γενικότερα καθ' όλη τη διάρκεια των σπουδών μου.

Δεκέμβριος 2011

Χαϊντίνη Ασημίνα

Περιεχόμενα

Κεφάλαιο 1. Εισαγωγή.....	- 6 -
1.1. Αντικείμενο της Διπλωματικής	- 6 -
1.2. Πως προέκυψε η ανάγκη δημιουργίας αυτού του συστήματος;.....	- 6 -
1.3. Θεσμικό Πλαίσιο	- 8 -
1.4. Ιστορικό Ελέγχων	- 8 -
1.5. Σκοπός και Αντικειμενικοί Στόχοι	- 11 -
1.5.1. Λειτουργικές Περιοχές - Υποσυστήματα	- 12 -
1.5.1.1. Πρώτη Λειτουργική Περιοχή	- 12 -
1.5.1.2. Δεύτερη Λειτουργική Περιοχή.....	- 12 -
1.6. Σχεδιαζόμενες Επεκτάσεις	- 12 -
1.7. Δομή εργασίας.....	- 13 -
Κεφάλαιο 2. Ανασκόπηση Βιβλιογραφίας.....	- 14 -
2.1. Διαδικτυακές Υπηρεσίες (Web Services)	- 14 -
2.1.1 Πλεονεκτήματα των Web Services.....	- 16 -
2.2. Υπηρεσιοστρεφής Αρχιτεκτονική (Service Oriented Architecture ...	- 17 -
2.3. XML (eXtensible Markup Language)	- 19 -
2.3.1. Ορθότητα XML εγγράφου	- 21 -
2.3.2. Ετικέτες - Tags	- 21 -
2.3.3. Μεταβλητές Ιδιοτήτων - Attributes	- 22 -
2.3.4. XML - Namespaces	- 22 -
2.3.5. XML - Shema	- 22 -
2.3.6. XML Parsers (λεκτικοί αναλυτές XML).....	- 23 -
2.4. SOAP (Simple Object Access Protocol)	- 24 -
2.4.1. SOAP over HTTP.....	- 25 -
2.4.2. SOAP Exceptions and Errors	- 25 -
2.4.3. SOAP Extensions.....	- 26 -
2.4.4. SOAP Versioning	- 27 -
2.4.5. Χειρισμός λαθών (Error Handling)	- 27 -
2.5. REST (Representational State Transfer).....	- 27 -
2.6. WSDL (Web Services Description Language).....	- 28 -
2.7. UDDI (Universal Description, Discovery and Integration)	- 32 -
Κεφάλαιο 3. Ερευνητική Μεθοδολογία	- 33 -
3.1. Ποιοτική Έρευνα	- 33 -
3.1.1. Πλεονεκτήματα και μειονεκτήματα ποιοτικής έρευνας	- 34 -
3.1.2. Τύποι ποιοτικής έρευνας.....	- 35 -
3.2. Έρευνα Δράσης (Action Research)	- 35 -
3.2.1. Στόχοι της Έρευνας Δράσης.....	- 36 -
3.2.2. Τα βασικά χαρακτηριστικά της Έρευνας Δράσης.....	- 36 -
3.2.3. Μεθοδολογία	- 37 -
3.2.4. Πότε χρησιμοποιείται η Έρευνα Δράσης;.....	- 38 -
Κεφάλαιο 4. Ανάλυση και Σχεδίαση	- 39 -
4.1. Εισαγωγή επενδυτικού σχεδίου.....	- 39 -
4.2. Δομή βάσης δεδομένων	- 46 -
Κεφάλαιο 5. Εργαλεία Υλοποίησης.....	- 59 -
5.1. Spring	- 59 -
5.1.1. Τα module της Spring	- 60 -
5.1.2. Java EE Connector API (JCA)	- 62 -
5.1.3. Dependency Injection (DI).....	- 63 -

5.1.4.	Aspect – Oriented Programming	- 63 -
5.1.5.	Model – View - Controller	- 64 -
5.2.	Apache Axis	- 66 -
5.2.1.	Τα γενικά χαρακτηριστικά του AXIS2	- 68 -
5.2.2.	Tomcat	- 71 -
5.2.3.	Apache HTTP Server	- 71 -
5.3.	ZK Framework	- 72 -
5.3.1.	Η αρχιτεκτονική του ZK	- 72 -
5.3.2.	Η γλώσσα σήμανσης ZUML	- 73 -
5.3.3.	Τα Components του ZK και ο χειρισμός γεγονότων	- 74 -
5.4.	Ant	- 77 -
Κεφάλαιο 6.	Λεπτομέρειες Υλοποίησης	- 78 -
6.1.	Υλοποίηση Web Forms	- 78 -
6.1.1.	Υλοποίηση του στρώματος Μοντέλου (View Layer)	- 78 -
6.1.1.1.	Αντικείμενα μεταφοράς δεδομένων (DTO)	- 78 -
6.1.1.2.	Αντικείμενα πρόσβασης δεδομένων (DAO)	- 80 -
6.1.1.3.	Διαχειριστής (Manager)	- 81 -
6.1.2.	Υλοποίηση του στρώματος της Προβολής (View Layer)	- 85 -
6.1.3.	Υλοποίηση του στρώματος Ελεγκτή (Controller Layer)	- 88 -
6.2.	Υλοποίηση Web Service	- 90 -
6.2.1.	Μέθοδος checkAccumulation	- 90 -
6.2.2.	Μέθοδος insertIPlan	- 91 -
6.3.	Ασφάλεια Συστήματος	- 92 -
Κεφάλαιο 7.	Σενάριο χρήσης εφαρμογής	- 94 -
Κεφάλαιο 8.	Επίλογος	- 99 -
8.1.	Σύνοψη και συμπεράσματα	- 99 -
8.2.	Μελλοντικές Επεκτάσεις	- 99 -
Βιβλιογραφία		- 100 -
Ιστότοποι		- 101 -
Κεφάλαιο 9.	Ευρετήριο Όρων	- 102 -
Κεφάλαιο 10.	Παράρτημα	- 107 -
10.1.	AccumulationService.wsdl	- 107 -
10.2.	InsertIPlan.xsd	- 108 -
10.3.	Request CheckAccumulation.xml	- 118 -
10.4.	Request InsertIPlan.xml	- 119 -
10.5.	CompanyDaoImpl.java	- 122 -
10.6.	CompanyController.java	- 126 -
10.7.	CompanyDaoImpl.java	- 127 -
10.8.	panelCompany.zul	- 134 -

Κεφάλαιο 1. Εισαγωγή

Η παρούσα εργασία αποτελεί την τεκμηρίωση της σχεδίασης και της ανάπτυξης ενός πληροφοριακού συστήματος το οποίο αυτοματοποιεί την διαδικασία ελέγχου σώρευσης. Στο εισαγωγικό αυτό κεφάλαιο παρουσιάζεται η γνωστική περιοχή και το θέμα στο οποίο στηρίζεται η διπλωματική εργασία, καθώς και η οργάνωση του τόμου σε κεφάλαια.

1.1. Αντικείμενο της Διπλωματικής

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η ανάπτυξη μιας εφαρμογής την οποία θα μπορεί να χρησιμοποιήσει ο χρήστης μέσω κάποιων ηλεκτρονικών σελίδων στο διαδίκτυο (web forms) ή μέσω ηλεκτρονικών υπηρεσιών (web services). Σκοπός είναι η εφαρμογή του θεωρητικού υπόβαθρου ώστε να υλοποιηθεί ένα πληροφοριακό σύστημα από την αρχή, με τον βέλτιστο τρόπο και η ανάπτυξη web services με τα οποία θα γίνεται εισαγωγή στοιχείων σε μια βάση δεδομένων, εφόσον ικανοποιούνται κάποιοι έλεγχοι. Με τον όρο Web Service εννοούμε μια υπηρεσία, η οποία βρίσκεται σε έναν διακομιστή Web και την οποία μπορεί να χρησιμοποιήσει ο οποιοσδήποτε έχει πρόσβαση σε αυτόν. Το μόνο που χρειάζεται για να χρησιμοποιήσουμε την υπηρεσία είναι να ξέρουμε τον τρόπο κλήσης και τη μορφή των μεταβλητών που απαιτούνται και να μπορούμε να επικοινωνήσουμε με τέτοιο τρόπο, ώστε η υπηρεσία να καταλάβει την κλήση μας και να επιστρέψει ένα κατανοητό, για εμάς, αποτέλεσμα.

Η προσπάθειά μας τέλος επικεντρώθηκε στο να υλοποιήσουμε ένα αυτοματοποιημένο και πλήρες σύστημα, το οποίο θα παρέχει όσο το δυνατό πιο εύχρηστες και λειτουργικές υπηρεσίες.

1.2. Πως προέκυψε η ανάγκη δημιουργίας αυτού του συστήματος;

Σύμφωνα με τη συνθήκη της Ευρωπαϊκής Ένωσης και το κανονιστικό πλαίσιο εξειδίκευσης, όπως αυτό διαμορφώνεται από τους σχετικούς κανονισμούς και τις κατευθυντήριες γραμμές της Ευρωπαϊκής Επιτροπής, τα Κράτη-Μέλη υπόκεινται σε περιορισμούς όσον αφορά στη χορήγηση κρατικών ενισχύσεων ώστε να διασφαλίζεται ότι δε δημιουργούν στρέβλωση του ανταγωνισμού.

Ειδική κατηγορία περιορισμών, που διέπει το σύνολο του κανονιστικού πλαισίου κρατικών ενισχύσεων είναι οι μέγιστες εντάσεις και τα ανώτατα ποσά των κρατικών ενισχύσεων που μπορούν κατά περίπτωση να χορηγούνται από τα κράτη μέλη.

Η ιδιαιτερότητα στη διαχείριση των περιορισμών αυτών έγκειται στο γεγονός ότι η τήρηση των ανώτατων κατά περίπτωση ορίων, δεν μπορεί να διασφαλισθεί μεμονωμένα στο πλαίσιο κάθε καθεστώτος αλλά πρέπει να λαμβάνεται υπόψη σωρευτικά το συνολικό ποσό των δημόσιων μέτρων στήριξης ανεξάρτητα από το αν η στήριξη χρηματοδοτείται από τοπικές, περιφερειακές, εθνικές ή κοινοτικές πηγές.

Ειδικότερα, για τις ενισχύσεις ήσσονος σημασίας ο κανονισμός (ΕΕ) 1998/06-De minimis, ορίζει ότι τα κράτη μέλη οφείλουν να εξασφαλίζουν ότι το συνολικό ποσό

ενισχύσεων ήσσονος σημασίας που χορηγείται βάσει του κανόνα για τις ενισχύσεις ήσσονος σημασίας στην ίδια οικονομική μονάδα, δεν υπερβαίνει το ανώτατο όριο των 200.000 ευρώ σε περίοδο τριών οικονομικών ετών (100.000 για τον τομέα των μεταφορών).

Επιπλέον, η ανακοίνωση της ΕΕ «προσωρινό κοινοτικό πλαίσιο για τη λήψη μέτρων κρατικής ενίσχυσης με σκοπό να στηριχθεί η πρόσβαση στη χρηματοδότηση κατά τη διάρκεια της τρέχουσας χρηματοπιστωτικής και οικονομικής κρίσης» (2009/C 83/01), αύξησε το ανώτατο όριο de minimis ενισχύσεων σε 500.000 ευρώ για τα έτη 2008-2010.

Την περίοδο 2000-2006, η Ελλάδα ενισχύεται με σημαντικούς πόρους από τα διαρθρωτικά ταμεία της Ευρωπαϊκής Ένωσης στα πλαίσια της Ευρωπαϊκής Περιφερειακής Πολιτικής. Οι πόροι αυτοί δημιουργούν τις προϋποθέσεις για υψηλότερη μακροπρόθεσμη ανάπτυξη, οδηγούν σε πραγματική σύγκλιση με τις υπόλοιπες οικονομίες των κρατών μελών και συμβάλουν στην αντιμετώπιση των βασικών ελλείψεων της οικονομικής και κοινωνικής ζωής, κυρίως στους τομείς των υποδομών, της ανάπτυξης των ανθρώπινων πόρων και της παραγωγικότητας. Η στήριξη που παρέχει στην Ελλάδα η Ευρωπαϊκή Ένωση πραγματοποιείται μέσω α) του Κοινοτικού Πλαισίου Στήριξης β) των Κοινοτικών Πρωτοβουλιών και γ) του Ταμείου Συνοχής.

Στο πλαίσιο του Γ΄ ΚΠΣ χορηγήθηκαν ενισχύσεις de minimis σε μικρές και μεσαίες επιχειρήσεις όλων των τομέων.

Σε συμμόρφωση προς την απαίτηση του κανονισμού σχετικά με τη σώρευση, οι οδηγοί εφαρμογής των προγραμμάτων για την χορήγηση αυτών των ενισχύσεων προέβλεπαν την υποβολή σχετικής υπεύθυνης δήλωσης – βεβαίωσης από τον δικαιούχο της ενίσχυσης σχετικά με τις de minimis ενισχύσεις που έχει λάβει την προηγούμενη τριετία.

Στο πλαίσιο της διαχειριστικής τακτοποίησης για το κλείσιμο των επιχειρησιακών προγραμμάτων διενεργήθηκαν στην συγκεκριμένη φάση απολογιστικά διοικητικοί διασταυρωτικοί έλεγχοι σχετικά με τη σώρευση των ενισχύσεων de minimis, βάση των πληροφοριών που είναι καταχωρημένες στο ΟΠΣ ΚΠΣ, σε συνδυασμό με στοιχεία που συλλέχθηκαν σχετικά με τις μη συγχρηματοδοτούμενες ενισχύσεις που δεν καταχωρούνται στο ΟΠΣ. Τα αποτελέσματα αυτής της διαδικασίας έδειξαν ότι εκατοντάδες επιχειρήσεις είχαν υπερβεί το ανώτατο όριο χρηματοδότησης που τους αναλογούσε.

Στόχος αυτής της διπλωματικής εργασίας είναι η ανάπτυξη ενός πληροφοριακού συστήματος το οποίο θα υποστηρίζει τη διαδικασία ελέγχου σώρευσης ΑΦΜ σε επίπεδο ΑΦΜ Επιχείρησης κι ΑΦΜ μετόχων Επιχείρησης, σε συνεργαζόμενες και συνδεδεμένες επιχειρήσεις. Η υλοποίηση της παρούσας εργασίας θεωρείται η πρώτη φάση ανάπτυξης ενός συστήματος που με μελλοντικές αναβαθμίσεις θα περιλαμβάνει ακόμα περισσότερα χαρακτηριστικά.

Οι βασικές εργασίες για την ολοκλήρωση της εργασίας είναι:

Επίπεδο 1: online έλεγχος κατά την στιγμή έγκρισης της Κρατικής Ενίσχυσης

Επίπεδο 2: Επικαιροποίηση του ποσού της Δημόσιας Δαπάνης κατά την ολοκλήρωση της επένδυσης

Επίπεδο 3: Απολογιστικός (ex post) έλεγχος.

Τα κύρια παραδοτέα της εργασίας είναι:

- ✓ Η σχεσιακή βάση δεδομένων
- ✓ Ο κώδικας της διαδικτυακής εφαρμογής και των ηλεκτρονικών υπηρεσιών
- ✓ Οδηγός χρήσης συστήματος

Τα εργαλεία στα οποία βασίστηκε η υλοποίηση ήταν ο Tomcat ως server, η Oracle Database 11g ως Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS) και η γλώσσα προγραμματισμού JAVA ως πλατφόρμα υλοποίησης σε συνδυασμό με Spring Framework και ZK σελίδες.

1.3. Θεσμικό Πλαίσιο

Το 1992, η Ευρωπαϊκή Επιτροπή ζήτησε να της ανακοινώνονται ενισχύσεις μεγαλύτερες των 100.000 ευρώ (στην ίδια επιχείρηση, μέσα σε 3 έτη). Επειδή το 1992 δεν υπήρχε πρόβλεψη για ελέγχους σώρευσης επενδύσεων, η διαδικασία εύρεσης των απαιτούμενων στοιχείων ήταν δύσκολο να πραγματοποιηθεί.

1992

Ο κανόνας de minimis εισήχθη το 1992 και αφορά στην πολιτική της Ευρωπαϊκής Επιτροπής για τις κρατικές ενισχύσεις των μικρομεσαίων επιχειρήσεων. Συγκεκριμένα, ενισχύσεις μεγαλύτερες των 100.000 ΕΥΡΩ, μέσα σε χρονικό διάστημα μιας τριετίας, ανά επιχείρηση, έπρεπε να ανακοινώνονται στην Επιτροπή για λόγους ελέγχου της τήρησης των κανόνων του ανταγωνισμού.

Μερικά χρόνια αργότερα, ο κανονισμός ανανεώνεται, διπλασιάζοντας το ανώτατο όριο από 100.000 ευρώ σε 200.000 και συμπεριλαμβάνοντας περισσότερες περιπτώσεις που θα έπρεπε να αναφερθούν.

2006

Ένας νέος κανονισμός (ΕΚ 1998/2006) ενημερώνει τον κανόνα de minimis, με διπλασιασμό ιδίως του ανώτατου ορίου του και διεύρυνση του πεδίου εφαρμογής του.

Ο νέος κανονισμός de minimis εγγράφεται, εξάλλου, στο πλαίσιο του σχεδίου δράσης της Επιτροπής για τις κρατικές ενισχύσεις και συμπληρώνει τις κατευθυντήριες γραμμές για το επιχειρηματικό κεφάλαιο και την πλαισίωση των ενισχύσεων στην έρευνα, στην ανάπτυξη και στην καινοτομία.

1.4. Ιστορικό Ελέγχων

Η ανάγκη για έλεγχο της σώρευσης de minimis ξεκίνησε ουσιαστικά από την Τρίτη Προγραμματική Περίοδο, δηλαδή από το Γ' ΚΠΣ. Τότε έγιναν επιτακτικές οι απαιτήσεις από την Επιτροπή. Αρχικά ενεργοποιήθηκε η Διαχειριστική Αρχή του Επιχειρησιακού Προγράμματος (ΕΠ) της Ανταγωνιστικότητας. Το ΕΠ

Ανταγωνιστικότητα ήταν και το κατεξοχήν υπεύθυνο για κρατικές ενισχύσεις σε ιδιωτικές εταιρίες.

Η διαδικασία συλλογής και ελέγχου των δεδομένων πραγματοποιήθηκε σε 2 φάσεις οι οποίες περιγράφονται παρακάτω και έδωσαν το έναυσμα για την δημιουργία μίας εφαρμογής ελέγχου σώρευσης σε ευρύτερο επίπεδο που θα ελέγχει απολογιστικά όλες τις δράσεις που εμπίπτουν στα καθεστώτα De minimis, ή/και σε άλλα καθεστώτα ενίσχυσης.

Πρώτο Βήμα: Έλεγχος δεδομένων για επιχειρήσεις που ενισχύθηκαν με βάση τον κανονισμό 69/2001 (De minimis)

Τον Φεβρουάριο του 2008 αποφασίστηκε να γίνουν οι πρώτοι έλεγχοι, κάνοντας χρήση των στοιχείων που ήταν καταχωρημένα στο Ολοκληρωμένο Πληροφοριακό Σύστημα του Υπουργείου Οικονομικών (ΟΠΣ). Στόχος ήταν να διασταυρωθούν τα στοιχεία που αφορούν σε ενισχύσεις de minimis που χορηγήθηκαν από το Επιχειρησιακό Πρόγραμμα «Ανταγωνιστικότητα».

Για το εγχείρημα αυτό, επειδή στο ΟΠΣ δεν ήταν καταχωρημένες όλες οι απαιτούμενες πληροφορίες για την παρακολούθηση της σώρευσης των ενισχύσεων αυτών, η Διαχειριστική Αρχή του ΕΠ «Ανταγωνιστικότητα» χρειάστηκε να προβεί σε υποστηρικτικές ενέργειες καταγραφής επιπλέον πληροφοριών, που αντλήθηκαν από τους φακέλους των συγκεκριμένων επενδύσεων για τις οποίες υπήρχαν ενδείξεις ότι μπορεί να υπάρχει πρόβλημα.

Τα στοιχεία αυτά διασταυρώθηκαν με τα στοιχεία του ΟΠΣ για ενισχύσεις που έχουν χορηγηθεί στο πλαίσιο των λοιπών προγραμμάτων του Γ΄ ΚΠΣ.

Τα ευρήματα της διαδικασίας αυτής ήταν:

1. 27 Επιχειρήσεις (ΑΦΜ) του Επιχειρησιακού Προγράμματος «Ανταγωνιστικότητα» με υπέρβαση De minimis η οποία δημιουργήθηκε από τελική ενίσχυση υπέρβασης σε έργα αυτού του Επιχειρησιακού Προγράμματος. Τα 27 αυτά ΑΦΜ αντιστοιχούσαν σε 28 υποέργα τα οποία είχαν συνολικά 876.803,96 € δημόσια δαπάνη.
2. Αντίστοιχα εντοπίστηκαν 13 περιπτώσεις επιχειρήσεων (ΑΦΜ) με υπέρβαση De minimis, η οποία δημιουργήθηκε σε άλλο επιχειρησιακό πρόγραμμα (Κοινωνία της Πληροφορίας). Αυτά αφορούσαν 13 υποέργα του επιχειρησιακού προγράμματος Κοινωνία της Πληροφορίας.

Δεύτερο Βήμα: Έλεγχος δεδομένων για επιχειρήσεις που ενισχύθηκαν με βάση τον κανονισμό 69/2001 και τον κανονισμό 1998/2006

Στην δεύτερη φάση του ελέγχου που διεξήγαγε η Διαχειριστική Αρχή του Επιχειρησιακού Προγράμματος «Ανταγωνιστικότητα» στα στοιχεία των κρατικών ενισχύσεων που την αφορούν, χρησιμοποίησε τα παρακάτω:

- Δεδομένα όλων των Επιχειρησιακών Προγραμμάτων του Γ΄ ΚΠΣ και του ΕΣΠΑ, όπως αυτά έχουν καταχωρηθεί στο ΟΠΣ.
- Δεδομένα που χορηγήθηκαν από το ΤΕΜΠΜΕ και καταχωρήθηκαν σε ειδική βάση δεδομένων.
- Δεδομένα που είχαν καταγραφεί κατά το Πρώτο Βήμα ελέγχου.
- Επιπλέον δεδομένα που αναζητήθηκαν ανά περίπτωση.

Ο στόχος του ελέγχου ήταν να εντοπιστούν και να διορθωθούν πιθανά προβλήματα υπέρβασης του συνολικού ορίου των 100.000€ δημόσιας δαπάνης υπαγωγής που προβλέπεται¹ για τριετία από τον κανονισμό 69/2001 και του συνολικού ορίου των 200.000€ που προβλέπεται² από τον κανονισμό 1998/2006. Εξαιρέθηκαν φορείς που είναι Δημόσιο ή Ευρύτερος Δημόσιος Τομέας, Ιδρύματα και άλλες τέτοιες περιπτώσεις.

Τα ευρήματα της διαδικασίας αυτής ήταν:

1. 7 Επιχειρήσεις (ΑΦΜ) του Επιχειρησιακού Προγράμματος «Ανταγωνιστικότητα» με υπέρβαση De minimis.
2. 10 Επιχειρήσεις (ΑΦΜ) του Επιχειρησιακού Προγράμματος «Κοινωνίας της Πληροφορίας».

Τα βήματα βάση των οποίων έγινε ο έλεγχος είναι τα εξής:

1. Δημιουργία βάσης δεδομένων με όλες τις πληρωμές κρατικών ενισχύσεων που υπάγονταν στα δύο συγκεκριμένα καθεστάτα ενισχύσεων οι οποίες αντλήθηκαν από το ΟΠΣ.
2. Αποστολή των στοιχείων στους φορείς. Εκεί διορθώθηκαν (παρουσίαζαν αρκετά λάθη) και ομογενοποιήθηκαν σε ένα ενιαίο format στην λογική δομή της βάσης δεδομένων και εν συνεχεία εισαγωγή σε αυτήν.
3. Υλοποίηση πληροφοριακού συστήματος για την συγκέντρωση, τήρηση, επεξεργασία, έλεγχο, εξαγωγή και παρουσίαση των δεδομένων. Πάνω σε υποσυστήματα του συστήματος αυτού, βασίστηκε και η παρούσα Διπλωματική Εργασία.
4. Εκτέλεση ελέγχου σώρευσης για τα ΑΦΜ των επιχειρήσεων που έχουν χρηματοδοτηθεί με βάση το καθεστώς ΕΚ 69/2001 με βάση την ημερομηνία έγκρισης και τις πληρωμές. Εξαγωγή αναφορών με τα αποτελέσματα αυτών που ξεπερνούν το όριο των 100.000€ στην τριετία.
5. Εκτέλεση ελέγχου σώρευσης για τα ΑΦΜ των επιχειρήσεων που έχουν χρηματοδοτηθεί με βάση το καθεστώς ΕΚ 1998/2006 με βάση την ημερομηνία έγκρισης και τις πληρωμές. Εξαγωγή αρχείου με αυτές που ξεπερνούν το όριο των 200.000€ στην τριετία.

¹ Η έναρξη και η λήξη της τριετίας για κανονισμό 69/2001 ταυτοποιούνται με τις ημερομηνίες των Υπουργικών Αποφάσεων υπαγωγής οι οποίες παρέχουν την ενίσχυση. Παραβίαση του κανόνα συμβαίνει, όταν χορηγείται η τελευταία ενίσχυση με την οποία γίνεται η υπέρβαση των 100.000 αθροίζοντας τις χορηγηθείσες ενισχύσεις σε ημερολογιακό διάστημα των 3 προηγούμενων ετών από την τελευταία υπαγωγή.

² Αναφορικά με τον κανονισμό 1998/2006 ο έλεγχος πραγματοποιείται με βάση τα 3 τελευταία οικονομικά έτη (δηλαδή του τρέχοντος έτους έγκρισης-υπαγωγής της ενίσχυσης και των 2 προηγούμενων ετών).

1.5. Σκοπός και Αντικειμενικοί Στόχοι

Βάσει των παραπάνω απαιτείται η ύπαρξη πληροφοριακού συστήματος για την τήρηση και τον έλεγχο των δεδομένων. Οι αντικειμενικοί στόχοι της παρούσας εργασίας είναι οι εξής:

- ✓ Μελέτη και εφαρμογή θεωρητικού υπόβαθρου
- ✓ Σχεδίαση και υλοποίηση πληροφοριακού συστήματος
- ✓ Ικανοποίηση των απαιτήσεων και προδιαγραφών

Στόχος του συστήματος είναι η επαλήθευση της νομιμότητας των αντίστοιχων δαπανών των ενισχύσεων σύμφωνα με τους κοινοτικούς και εθνικούς κανόνες και την αρχή της χρηστής δημοσιονομικής διαχείρισης.

- Η εν λόγω εφαρμογή **δίνει** τη δυνατότητα on line ελέγχου κάθε επενδυτικής πρότασης και αναφοράς του de minimis ποσού που δικαιούται η επιχείρηση πριν την έγκριση. Δημιουργείται **μοναδικός αριθμός σώρευσης** για κάθε επενδυτική πρόταση, που θα πιστοποιεί ότι έχει περάσει από τον έλεγχο σώρευσης, καθώς καμία πρόταση δεν θα εγκρίνεται αν δεν φέρει αυτόν τον αριθμό. Η χρησιμοποίηση του μοναδιαίου κωδικού «θωρακίζεται» θεσμικά προκειμένου να μην είναι δυνατή η παροχή κρατικής ενίσχυσης, χωρίς να έχει γίνει ο έλεγχος από το Σύστημα Σώρευσης.
- Κατά την ολοκλήρωση της επένδυσης θα γίνεται η επικαιροποίηση με βάση το ποσό της ενίσχυσης που τελικά χορηγήθηκε. Η ενημέρωση του Συστήματος είναι υποχρέωση του Φορέα που χορηγεί τις εν λόγω ενισχύσεις.

Επιπλέον υπάρχει η δυνατότητα εκ των υστέρων ελέγχων ώστε να καλύπτεται οποιαδήποτε τυχόν καθυστέρηση στην εισαγωγή των δεδομένων στο σύστημα σώρευσης.

Οι χρήστες στους οποίους απευθύνεται το σύστημα θα είναι οι φορείς που χορηγούν κρατικές ενισχύσεις (δεν θα έχει πρόσβαση το ευρύ κοινό).

Στο σύστημα θα εισαχθούν αναδρομικά όλα τα διαθέσιμα στοιχεία που είναι απαραίτητα για τον έλεγχο της σώρευσης για το ΕΣΠΑ.

Το σύστημα ελέγχει τη σώρευση ανά ΑΦΜ λαμβάνοντας υπόψη και τις κατά δήλωση συνδεδεμένες και με κοινό έλεγχο επιχειρήσεις και τρείς στοιχεία μέχρι 1ου βαθμού μετοχικής σύνθεσης (πάνω από ένα ποσοστό που θα ορίζεται παραμετρικά). Επίσης τηρούνται στοιχεία συνδεδεμένων επιχειρήσεων και των ασκούντων τον έλεγχο (με βάση το ΑΦΜ της επιχείρησης που υποβάλει την επενδυτική πρόταση).

Τα παραπάνω στοιχεία υποβάλλονται ανά επιχείρηση με την αίτηση για λήψη Κρατικής Ενίσχυσης καλύπτοντας όλους τους φορείς που χορηγούν Κρατικές Ενισχύσεις:

- Φορείς που έχουν δικό τους Πληροφοριακό Σύστημα (ΠΣ)
- Φορείς που δεν έχουν δικό τους Πληροφοριακό Σύστημα (ΠΣ)

1.5.1. Λειτουργικές Περιοχές - Υποσυστήματα

Το σύστημα αποτελείται από δύο λειτουργικές περιοχές έτσι ώστε να καλύψει τους φορείς που έχουν δικό τους ΠΣ αλλά και αυτούς που δεν έχουν οι οποίες περιγράφονται στις επόμενες παραγράφους.

1.5.1.1. Πρώτη Λειτουργική Περιοχή

Η πρώτη Λειτουργική Περιοχή του συστήματος αφορά σε φορείς με δικά τους πληροφοριακά συστήματα. Θα αναπτυχθεί σειρά αυτομάτων διεπαφών (web services):

- Έλεγχος ΑΦΜ, για το ποσόν που μπορεί να απορροφήσει
- Online έλεγχος την στιγμή τις έγκρισης
- Σε περίπτωση που δικαιούται μικρότερο ποσό από το ζητούμενο, απάντηση με το ποσό αυτό – σε αυτή την περίπτωση θα δίνεται και πληροφόρηση για τον λόγο (πχ κάποιος μέτοχος έχει υπερβεί το όριο)
- Προσδιορισμός της έγκρισης της Κρατικής Ενίσχυσης με μοναδιαίο κωδικό
- Ενημέρωση της κατάστασης της Κρατικής Ενίσχυσης στο σύστημα σε περίπτωση αλλαγών
- Περάτωση της Κρατικής Ενίσχυσης στο σύστημα

Η ενημέρωση του Πληροφοριακού Συστήματος Σώρευσης είναι υποχρέωση του Φορέα που παρέχει τις Κρατικές Ενισχύσεις.

1.5.1.2. Δεύτερη Λειτουργική Περιοχή

Η δεύτερη Λειτουργική Περιοχή του συστήματος αφορά σε φορείς που δεν διαθέτουν Πληροφοριακό Σύστημα ή/και δεν θα μπορούν να στείλουν δεδομένα μέσω διαδικτυακών διεπαφών (web services). Για τους χρήστες αυτούς θα αναπτυχθεί διαδικτυακή εφαρμογή για να παρέχονται οι ίδιες δυνατότητες όπως και στην 1^η Λειτουργική Περιοχή – αλλά αυτή τη φορά μέσω web forms.

Ο φορέας καταχωρεί τις απαραίτητες πληροφορίες (όπως στην 1η ΛΠ) και το σύστημα θα ανταποκρίνεται με τις ίδιες πληροφορίες (όπως και στην 1η ΛΠ).

1.6. Σχεδιαζόμενες Επεκτάσεις

Οι διαδικτυακές υπηρεσίες που παρουσιάζονται στην παρούσα εργασία, καλύπτουν ένα μόνο τμήμα των υφιστάμενων απαιτήσεων της ΕΕ, της χώρας και των χρηστών που διαχειρίζονται και χρησιμοποιούν τα προγράμματα αυτά. Στο πλαίσιο αυτό, προγραμματίζονται και σχεδιάζονται οι παρακάτω επεκτάσεις του συστήματος:

1. Επέκταση του συστήματος για την κάλυψη περισσότερων επενδυτικών καθεστώτων σώρευσης.
2. Υλοποίηση διαδικτυακών υπηρεσιών για την εξυπηρέτηση εξωτερικών πληροφοριακών συστημάτων. Στόχος είναι η παροχή πληροφόρησης και/ή έγκρισης.

3. Κάλυψη υπηρεσιών/δεδομένων με ψηφιακή υπογραφή.

Η σειρά με την οποία παρουσιάζονται οι παραπάνω σχεδιαζόμενες επεκτάσεις είναι τυχαία, κι όχι ενδεικτική της σειράς με την οποία θα υλοποιηθούν.

1.7. Δομή εργασίας

Η παρούσα εργασία αποτελείται από 10 κεφάλαια. Στις ενότητες των κεφαλαίων αυτών παρουσιάζεται ουσιαστικά με αναλυτικό τρόπο η διαδικασία ανάπτυξης της εφαρμογής που περιγράφηκε στην προηγούμενη ενότητα.

Το κεφάλαιο 2 έχει τίτλο «Ανασκόπηση Βιβλιογραφίας» και τοποθετεί το θέμα ως προς την επιστημονική περιοχή στην οποία ανήκει.

Το κεφάλαιο 3 έχει τίτλο «Ερευνητική Μεθοδολογία» και παρουσιάζει τη μέθοδο έρευνας που χρησιμοποιήθηκε στην παρούσα εργασία.

Το κεφάλαιο 4 έχει τίτλο «Ανάλυση και Σχεδίαση» και μας παρουσιάζει αναλυτικά τη δομή του συστήματος, τις λειτουργίες του και τη σχεδίαση της βάσης δεδομένων.

Το κεφάλαιο 5 έχει τίτλο «Εργαλεία Υλοποίησης» και παρουσιάζει τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της εργασίας.

Το κεφάλαιο 6 έχει τίτλο «Λεπτομέρειες Υλοποίησης» και παρουσιάζει τα δύο τμήματα από τα οποία αποτελείται η εργασία (Web Forms & Web Services) και τον τρόπο υλοποίησής τους.

Το κεφάλαιο 7 έχει τίτλο «Σενάριο Χρήσης Εφαρμογής» και περιλαμβάνει ένα παράδειγμα χρήσης του συστήματος.

Το κεφάλαιο 8 έχει τίτλο «Επίλογος» και περιλαμβάνει τη σύνοψη της διπλωματικής εργασίας καθώς και τα συμπεράσματα που εξήχθησαν με την ολοκλήρωσή της. Επιπλέον γίνεται λόγος στις μελλοντικές επεκτάσεις που η παρούσα εργασία επιδέχεται.

Έπειτα ακολουθούν τα κεφάλαια της Βιβλιογραφίας και των Ιστότοπων.

Το κεφάλαιο 9 έχει τίτλο «Ευρετήριο Όρων» και παρέχει επεξηγήσεις για τους όρους που χρησιμοποιήθηκαν στην εργασία.

Το κεφάλαιο 10 έχει τίτλο «Παράρτημα» όπου δίνονται τα στοιχεία που συμπληρώνουν το κεφάλαιο 6 όσον αφορά την υλοποίηση του συστήματος.

Κεφάλαιο 2. Ανασκόπηση Βιβλιογραφίας

Στο κεφάλαιο αυτό παρουσιάζεται το θεωρητικό υπόβαθρο για την υλοποίηση μιας Διαδικτυακής υπηρεσίας με την ανάλυση των τεχνολογιών και των προτύπων που απαιτούνται. Στη συνέχεια, δίνεται η γενική περιγραφή του προς ανάπτυξη συστήματος και προσδιορίζονται οι στόχοι αυτού.

2.1. Διαδικτυακές Υπηρεσίες (Web Services)

Οι Διαδικτυακές Υπηρεσίες αποτελούν τεχνολογία αιχμής στο χώρο του λογισμικού και της ολοκλήρωσης συστημάτων και εφαρμογών (System Application Integration) και συνεχώς όλο και περισσότεροι οργανισμοί παγκοσμίως ασχολούνται ενεργά με την τεχνολογία αυτή.

Τα Web Services παρέχουν ένα τυποποιημένο τρόπο αλληλεπίδρασης μεταξύ διαφορετικών εφαρμογών λογισμικού, ανεξάρτητα από την πλατφόρμα και το πλαίσιο εργασίας (framework), στα οποία έχουν υλοποιηθεί, και χωρίς να απαιτούν καμία αλλαγή στον μηχανισμό του συστήματος. Για το σκοπό αυτό, χρησιμοποιούν ανοικτά πρότυπα και ευρέως διαδεδομένα πρωτόκολλα, όπως XML (eXtensible Markup Language), SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), HTTP (HyperText Transfer Protocol), SMTP (Simple Mail Transport Protocol) ή FTP (File Transfer Protocol). Οι πληροφορίες που αφορούν τις διαδικτυακές υπηρεσίες δημοσιεύονται, οπότε η εύρεση και η χρήση τους γίνονται ταχύτατα, ενώ μπορούν να δημιουργηθούν και να ανανεωθούν με εύκολο και οικονομικό τρόπο. Με άλλα λόγια, οι διαδικτυακές υπηρεσίες αποτελούν σήμερα το επόμενο βήμα στον αντικειμενοστραφή προγραμματισμό: αντί της ανάπτυξης λογισμικού από ένα μικρό πλήθος βιβλιοθηκών κλάσεων που παρέχονται σε μια θέση, οι προγραμματιστές μπορούν να προσπελάσουν αμέτρητες βιβλιοθήκες μέσω ενός δικτύου υπολογιστών, το οποίο μπορεί να είναι το internet, κάποιο intranet ή οποιασδήποτε άλλης μορφής δίκτυο.

Ένας κοινά αποδεκτός ορισμός για τα web services δεν έχει πάντως ακόμη επικρατήσει. Ακολουθεί μια απόπειρα απόδοσης ενός ορισμού η οποία βασίζεται σε σχετικούς ορισμούς που παρέχονται από μεγάλες εταιρείες του χώρου, όπως η IBM και η Microsoft, οργανισμούς όπως ο W3C όπως και απόπειρες από τη βιβλιογραφία:

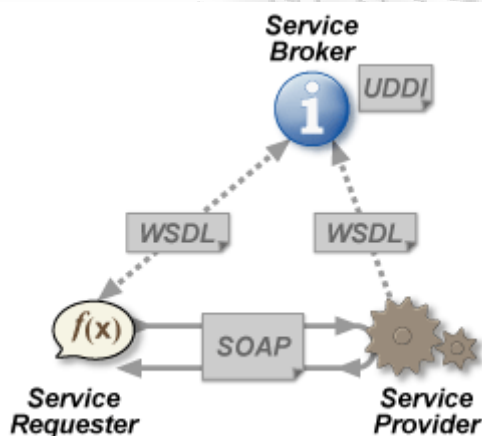
«Ένα web service είναι μια εφαρμογή λογισμικού προσβάσιμη από δίκτυο μέσω τυποποιημένης αποστολής μηνυμάτων σε XML. Ένα Web Service μπορεί να περιγράφεται χρησιμοποιώντας τυποποιημένη περιγραφή σε XML (service description) που παρέχει όλες τις απαραίτητες λεπτομέρειες για αλληλεπίδραση με την υπηρεσία, συμπεριλαμβανομένων της μορφής των μηνυμάτων, των πρωτοκόλλων μεταφοράς και της τοποθεσίας. Επιπλέον μπορεί να είναι δημοσιευμένο σε κάποιο επίσημο οδηγό καταγραφής από όπου και να αναζητείται».

Η κλήση της υπηρεσίας γίνεται μέσω κάποιας διεπαφής, η οποία κρύβει τις λεπτομέρειες υλοποίησής της. Έτσι το web service μπορεί να χρησιμοποιείται ανεξάρτητα από την πλατφόρμα υλικού ή λογισμικού στην οποία υλοποιείται και τη γλώσσα προγραμματισμού στην οποία η υπηρεσία είναι γραμμένη. Τα web services είναι προσβάσιμα μέσω ευρέως διαδεδομένων πρωτοκόλλων όπως το HTTP.

Οι βασισμένες σε web services εφαρμογές είναι συνεπώς υλοποιήσεις χαλαρά συνδεδεμένες, αξιοποιούν την τεχνολογία των components λογισμικού (component-oriented) και είναι διατεχνολογικές (cross-technology). Τα web services μπορούν να χρησιμοποιηθούν αυτόνομα ή σε συνδυασμό με άλλα web services για τις ανάγκες μιας εφαρμογής.

Αξίζει να σημειωθεί ότι ο όρος «web» είναι ίσως ατυχής ή περιοριστικός. Τα web services δεν είναι υποχρεωτικό να υπάρχουν μόνο στο διαδίκτυο (web) αλλά σε οποιοδήποτε είδος δικτύου (π.χ. Intranet). Επίσης σημειώνουμε ότι δεν αποτελεί διαδικτυακή υπηρεσία κάθε υπηρεσία που παρέχεται μέσω διαδικτύου, όπως για παράδειγμα η on-line υπηρεσία κράτησης δωματίων που παρέχεται από μια ξενοδοχειακή επιχείρηση μέσω του εξυπηρετητή της εταιρείας στην επίσημη ιστοσελίδα της. Απαραίτητη προϋπόθεση για να θεωρηθεί μια υπηρεσία διαδικτυακή είναι η ύπαρξη μιας εφαρμογής λογισμικού με σταθερή, γνωστή σε όλους διεπαφή, που κρύβει τις λεπτομέρειες υλοποίησης της υπηρεσίας και μέσω της οποίας γίνεται η κλήση της. Για την περίπτωση της ξενοδοχειακής μονάδας, η ύπαρξη μιας διαδικτυακής υπηρεσίας κράτησης δωματίων θα έδινε το δικαίωμα σε όλα τα ενημερωτικά – τουριστικά sites, αλλά και συνεργαζόμενες επιχειρήσεις να κλείνουν αυτόματα δωμάτιά της.

Μια γενική εικόνα που παρουσιάζει μια διαδικτυακή υπηρεσία και τα πρότυπα που χρησιμοποιούνται για την υλοποίησή της απεικονίζονται στο επόμενο σχήμα :

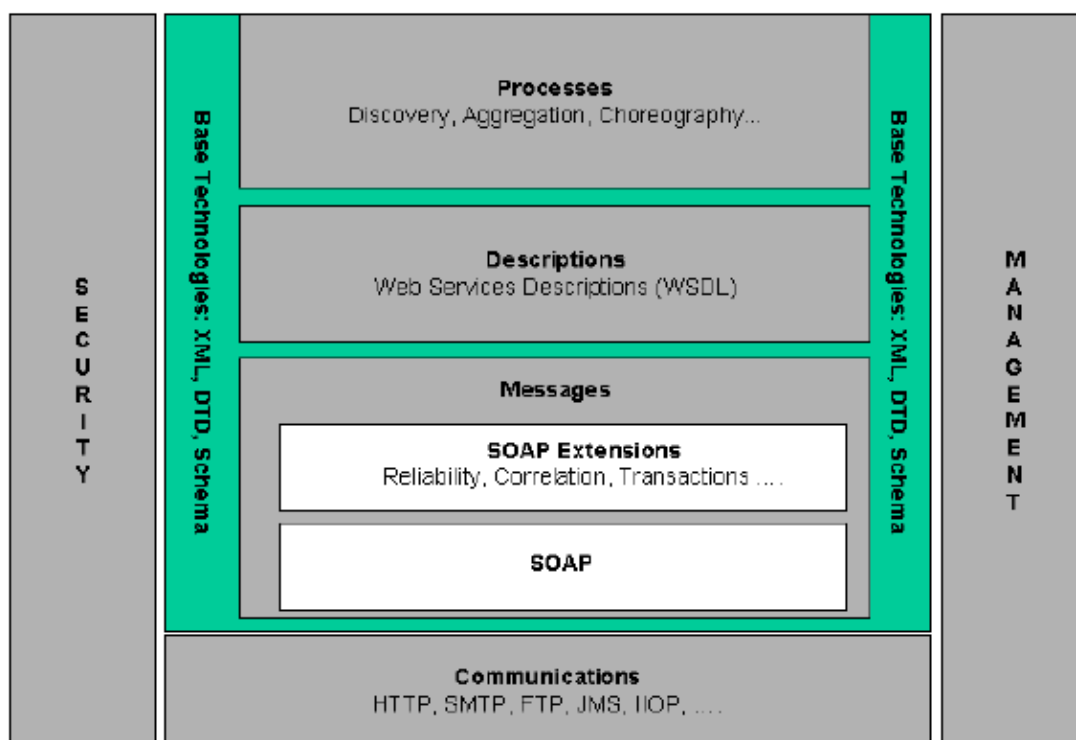


Εικόνα 2.1.1 Δομή των Web Services

(Πηγή:<http://upload.wikimedia.org/wikipedia/commons/4/4a/Webservices.png>)

Μια διαδικτυακή υπηρεσία αποτελείται από πολλές συσχετιζόμενες τεχνολογίες που τοποθετούνται σε διαφορετικό επίπεδο, όπως φαίνεται στο σχήμα 2.1.1. Ξεκινώντας από κάτω προς τα πάνω, αναφέρουμε τα πρότυπα που χρησιμοποιούνται και τα αναλύουμε εκτενέστερα στις επόμενες ενότητες. Αρχικά, απαιτείται ένα πρωτόκολλο για μεταφορά πληροφοριών μέσω δικτύου, όπως το HTTP (HyperText Transfer Protocol), το SMTP (Simple Mail Transport Protocol) ή το FTP (File Transfer Protocol). Κάθε κλήση και απόκριση της υπηρεσίας θα πρέπει να «συσκευαστεί» σε ένα μήνυμα SOAP (Simple Object Access Protocol), το οποίο μπορεί να υφίσταται επεξεργασία από κάποιες επεκτάσεις SOAP (SOAP Extensions) πριν αποσταλεί από τον αιτούντα υπηρεσία (request agent) και παραδοθεί στον πάροχο υπηρεσίας (provider agent) και αντίστροφα. Τα μηνύματα που ανταλλάσσονται και ο τρόπος που γίνεται η ανταλλαγή περιγράφονται λεπτομερώς στο αρχείο WSDL (Web Services Description Language).

Επόμενο βήμα είναι η ανακάλυψη των υπηρεσιών, για την οποία υπάρχουν τρεις προσεγγίσεις: η ύπαρξη ενός καταλόγου καταγραφής και δημοσίευσης των υπηρεσιών με τη μορφή υπηρεσίας καταγραφής (registry), όπως το UDDI (Universal Description, Definition and Integration) και το DISCO (Microsoft Discovery), που αποτελεί την πιο διαδεδομένη λύση, η ύπαρξη ιστοσελίδων παραπομπής σε υπηρεσίες, τύπου <http://www.amazon.com> και <http://www.google.com>, και η ανακάλυψη υπηρεσιών peer-to-peer, όπου οι υπηρεσίες ανακαλύπτουν η μια την άλλη δυναμικά μέσα σε ένα δίκτυο ομότιμων. Σημειώνουμε ότι τα πρότυπα SOAP, WSDL, UDDI και DISCO έχουν γραφεί σε γλώσσα επίσημησης XML (eXtensible Markup Language) και ενδεχομένως υπακούουν σε κάποιο έγγραφο DTD (Document Type Definition) ή XML Schema.



Εικόνα 2.1.2 Αρχιτεκτονική των Web Services
(Πηγή: <http://www.w3.org/TR/2003/WD-ws-arch-20030514/>)

2.1.1 Πλεονεκτήματα των Web Services

Η αρχιτεκτονική των web services παρέχει αρκετά πλεονεκτήματα μερικά από τα οποία αναφέρονται παρακάτω:

Διαλειτουργικότητα: Ένα web service παρέχει ανεξαρτησία τόσο από λειτουργικό σύστημα όσο και από το hardware. Οποιοδήποτε πρόγραμμα που συμβαδίζει με αυτή τη τεχνολογία μπορεί πολύ εύκολα να προσπελάσει μία τέτοια υπηρεσία.

Ενσωμάτωση: Σε ένα υπάρχον λογισμικό σύστημα που λειτουργεί μέσα στο Internet η δημιουργία ενός web service δεν απαιτεί αλλαγές στον μηχανισμό του συστήματος.

Διαθεσιμότητα και δημοσίευση: Οι πληροφορίες για τα web services δημοσιεύονται οπότε η εύρεση και η χρήση τους μπορεί να είναι ταχύτατες.

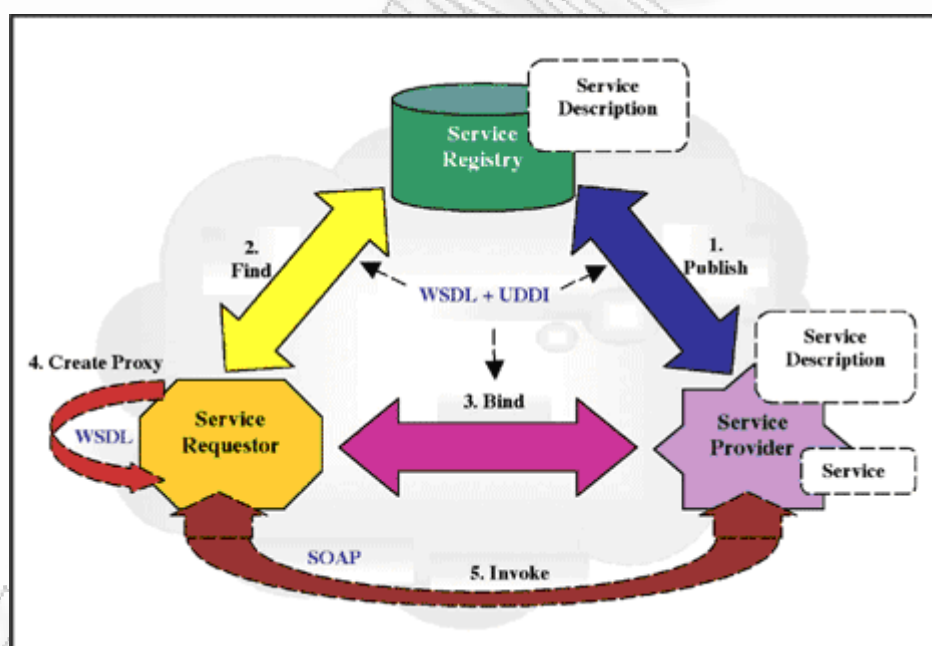
Επέκταση: Ένα έτοιμο web service είναι δυνατό να ανανεωθεί με εύκολο τρόπο παρέχοντας έτσι επιπρόσθετες υπηρεσίες στους χρήστες του.

Μικρό κόστος δημιουργίας και χρήσης: Εφόσον σε ένα λογισμικό σύστημα υπάρχει ήδη κάποια διαδικασία που χρειάζεται να επεκταθεί σε on-line υπηρεσία, η δημιουργία του web service κοστίζει ελάχιστα. Επίσης το κόστος ενσωμάτωσης ενός web service σε κάποιο website ή σε δικτυακή εφαρμογή είναι πάρα πολύ μικρό. Ακόμα και στις περιπτώσεις που η χρήση κάποιου web service γίνεται με ενοικίαση σίγουρα το συνολικό κόστος της χρήσης είναι αρκετά πιο μικρό από το κόστος δημιουργίας της υπηρεσίας αυτής.

Χρήση λογισμικών συστημάτων: Όλα τα λογισμικά συστήματα και ειδικότερα τα websites που χρησιμοποιούν έτοιμες υπηρεσίες γίνονται πιο λειτουργικά και πιο φιλικά αφού παρέχουν περισσότερες υπηρεσίες στους χρήστες.

2.2. Υπηρεσιοστρεφής Αρχιτεκτονική (Service Oriented Architecture)

Σε επίπεδο σχεδιασμού συστημάτων, η χρήση της τεχνολογίας των web services οδηγεί στην υιοθέτηση της λεγόμενης service-oriented αρχιτεκτονικής. Οι βασικοί ρόλοι και λειτουργίες στην αρχιτεκτονική αυτή παρουσιάζονται στο Σχήμα 2.2.1.



2.2.1. Αρχιτεκτονική γύρω από την υπηρεσία

(Πηγή: <http://soa.sys-con.com/node/121937>)

Από το παραπάνω σχήμα μπορούμε να διακρίνουμε τις οντότητες που απαρτίζουν την τρέχουσα αρχιτεκτονική των υπηρεσιών διαδικτύου. Αυτές είναι η οντότητα που ζητάει την υπηρεσία (Service Requestor), η οντότητα που παρέχει την υπηρεσία (Service Provider) και τέλος η οντότητα του καταλόγου υπηρεσιών (Service Registry).

Με περισσότερη λεπτομέρεια, κάθε μία οντότητα έχει τους ακόλουθους ρόλους:

- Η οντότητα, η οποία ζητάει την υπηρεσία (Service Requestor), είναι στην ουσία ο «αιτούμενος» (Client) της υπηρεσίας και πυροδοτεί την έναρξη της όλης διαδικασίας. Αρχικά επιφορτίζεται με το έργο της αναζήτησης της κατάλληλης περιγραφής μιας υπηρεσίας τις καταχωρήσεις της υπηρεσίας καταλόγου. Στη συνέχεια, και ενώ έχει εντοπίσει τον επιθυμητό παροχέα, δημιουργεί σύνδεση με αυτόν, καλεί την υπηρεσία και λαμβάνει τα αποτελέσματα.

- Η οντότητα που παρέχει την υπηρεσία (Service Provider), δημοσιοποιεί την περιγραφή της υπηρεσίας ιστού στην οντότητα του καταλόγου υπηρεσιών (Service Registry) και ουσιαστικά παρέχει την υπηρεσία στο διαδίκτυο. Επιπλέον, της ανατίθεται ο ρόλος της λήψης των μηνυμάτων κλήσης για την υπηρεσία από έναν ή περισσότερους αιτούμενους και της παροχής των απαραίτητων αποτελεσμάτων.

- Τέλος, η οντότητα του καταλόγου υπηρεσιών (Service Registry), περιέχει καταχωρήσεις με τις περιγραφές των ήδη δημοσιοποιημένων υπηρεσιών διαδικτύου. Για τις υπηρεσίες αυτές παρέχονται τρόποι αναζήτησης ανάμεσα στις διάφορες περιγραφές.

Οι λειτουργίες που παρέχονται μέσω αυτής της αρχιτεκτονικής είναι οι ακόλουθες:

- Η καταχώρηση των απαραίτητων πληροφοριών για μια υπηρεσία από τη μεριά του παρόχου (publishing), η οποία επιτυγχάνεται μέσω της χρήσης της γλώσσας περιγραφής WSDL.

- Η αναζήτηση και η εύρεση στους καταλόγους της κατάλληλης περιγραφής μιας «Υπηρεσίας Διαδικτύου» (finding). Η επικρατούσα τεχνολογία καταλόγου ονομάζεται UDDI.

- Τέλος, έχουμε την εγκαθίδρυση σύνδεσης μεταξύ αιτούμενου και παροχέα (binding), κλήση της κατάλληλης υπηρεσίας και αποστολή των αποτελεσμάτων.

Όλα τα παραπάνω επιτυγχάνονται με την ανταλλαγή μηνυμάτων SOAP μεταξύ των τριών εμπλεκόμενων οντοτήτων, η οποία βασίζεται στη γλώσσα μεταδεδομένων XML. Το κλειδί της παραπάνω αρχιτεκτονικής είναι η περιγραφή της «Υπηρεσίας Διαδικτύου». Η περιγραφή είναι εκείνη που δημοσιοποιείται από τον παροχέα της υπηρεσίας στον κατάλογο υπηρεσιών. Η περιγραφή είναι το αποτέλεσμα της αναζήτησης του αιτούμενου στον κατάλογο υπηρεσιών. Η περιγραφή είναι εκείνη που λέει στον αιτούμενο όσα ακριβώς πρέπει να γνωρίζει προκειμένου να κάνει κλήση της υπηρεσίας που τον ενδιαφέρει. Τέλος, η περιγραφή της υπηρεσίας μπορεί να περιέχει πληροφορία για το τι είδους είναι το αποτέλεσμα που αναμένεται να επιστραφεί ύστερα από την κλήση της υπηρεσίας.

Με την παραπάνω αρχιτεκτονική ήδη υπάρχουσες εφαρμογές μπορούν εύκολα να μετασχηματισθούν σε υπηρεσίες που να εξυπηρετούν άλλες υπάρχουσες – ή και νέες – εφαρμογές. Πολλές υπηρεσίες μπορούν να συνδυαστούν και να συνεργασθούν για την παραγωγή ενός αποτελέσματος με τρόπο διαφανή προς τον τελικό χρήστη, στον οποίο δίνεται η αίσθηση ότι κλήθηκε μία και μόνο υπηρεσία. Επιπρόσθετα, οργανισμοί και επιχειρήσεις μπορούν ευκολότερα να κατασκευάσουν λογισμικό, που να αλληλεπιδρά με επιχειρησιακές διαδικασίες και να ανταποκρίνεται γρήγορα στις αλλαγές του επιχειρησιακού περιβάλλοντος.

Στη συνέχεια γίνεται μια εισαγωγή στις σχετικές με τα web services τεχνολογίες (XML, SOAP, WSDL, UDDI).

2.3. XML (eXtensible Markup Language)

Σε ένα κόσμο όπου οι πληροφορίες παρέχονται μέσω του παγκόσμιου διαδικτύου, τα έγγραφα πρέπει να είναι εύκολα προσβάσιμα, μεταφέρσιμα και ευέλικτα. Πρέπει επίσης να είναι ανεξάρτητα οποιουδήποτε συστήματος και περιεχομένου. Οι γενικευμένες γλώσσες έχουν τέτοια χαρακτηριστικά, παρέχοντας στα έγγραφα αυτά μια δυνατότητα η οποία δεν υπάρχει σε άλλες γλώσσες περιγραφής εγγράφων. Η HTML είναι προβληματική και περιοριστική γλώσσα. Για την υλοποίηση των παραπάνω είναι απαραίτητη μια κοινή σύνταξη και γλώσσα για την περιγραφή των προδιαγραφών και τη μεταφορά των δεδομένων. Αυτό το ρόλο καλείται να παίξει η γλώσσα XML (Extensible Markup Language).

Η XML (eXtensible Markup Language) αναπτύχθηκε από το W3C's XML Working Group το 1996 και συνδυάζει την ισχύ και την επεκτασιμότητα της SGML (Standard Generalized Markup Language), από την οποία προέρχεται, με την απλότητα που απαιτεί η κοινότητα του Διαδικτύου. Είναι μια μεταφερτή, ευρέως υποστηριζόμενη, ανοικτή τεχνολογία για την περιγραφή δεδομένων.

Η XML περιγράφει μια κατηγορία αντικειμένων δεδομένων που ονομάζονται XML έγγραφα και εν μέρει τη συμπεριφορά των προγραμμάτων υπολογιστών που τα επεξεργάζονται. Τα έγγραφα XML αποτελούνται από μονάδες αποθήκευσης, τις οντότητες (entities), οι οποίες περιέχουν αναλυμένα λεκτικά (parsed) ή μη αναλυμένα λεκτικά (unparsed) δεδομένα. Τα αναλυμένα λεκτικά δεδομένα (parsed) συνίστανται από χαρακτήρες, ορισμένοι από τους οποίους αποτελούν δεδομένα χαρακτήρων και κάποιοι άλλοι δημιουργούν markup, δηλαδή κωδικοποιούν μια περιγραφή της διάταξης και της λογικής δομής του εγγράφου. Για τα μη αναλυμένα λεκτικά δεδομένα (unparsed) δεν είναι γνωστό εάν είναι κείμενο ή όχι, αλλά ακόμη κι αν είναι κείμενο, ενδέχεται να μην είναι XML, οπότε αγνοούνται κατά τη διαδικασία της ανάλυσης λεκτικών δεδομένων (parsing), στην οποία θα αναφερθούμε εκτενέστερα στη συνέχεια, και διοχετεύονται όπου προβλέπει η εφαρμογή.

Η XML έλυσε πολλά από τα προβλήματα που αντιμετώπισαν οι σχεδιαστές του web και είναι υπεύθυνη για την XHTML, μια ανασχεδιασμένη HTML. Η XML σχεδιάστηκε να ικανοποιήσει πολλές ανάγκες δίνοντας στα έγγραφα ένα μεγαλύτερο επίπεδο προσαρμοστικότητας στο στυλ και τη δομή από αυτό που υπήρχε παλαιότερα στην HTML. Η XML προσφέρει στους σχεδιαστές της HTML τη δυνατότητα να προσθέτουν περισσότερα στοιχεία στη γλώσσα. Δεν αναφέρεται μονάχα στους σχεδιαστές του web αλλά σε οποιονδήποτε ασχολείται με εκδόσεις.

Στην πραγματικότητα, η XML είναι markup γλώσσα για έγγραφα που περιέχουν δομημένες πληροφορίες. Markup γλώσσα είναι ένας μηχανισμός που καθορίζει δομές σε ένα έγγραφο. Οι δομημένες πληροφορίες περιλαμβάνουν περιεχόμενο και κάποιες διευκρινίσεις για το ρόλο που παίζει το περιεχόμενο. Σχεδόν όλα τα έγγραφα έχουν την ίδια δομή.

Η XML αποτελεί σήμερα το πρότυπο για την αποθήκευση δεδομένων που ανταλλάσσονται μεταξύ των εφαρμογών χάρη στα ακόλουθα χαρακτηριστικά που παρουσιάζει :

- Υποστηρίζει ανεξαρτησία από τα δεδομένα και διαχωρίζει τα περιεχόμενα από τον τρόπο εμφάνισής τους και τον χειρισμό τους, οπότε διευκολύνεται η λεκτική ανάλυσή τους (parsing).

- Διατίθενται έτοιμοι τρόποι σύνδεσης των κειμένων XML με τα πλέον σύγχρονα προγραμματιστικά περιβάλλοντα, όπως το Document Object Model (DOM) και το Simple API for XML (SAX).
- Είναι επεκτάσιμη και ανεξάρτητη από πλατφόρμες, γεγονός που την καθιστά απρόσβλητη σε τεχνολογικές αλλαγές.
- Τα έγγραφα XML είναι αναγνώσιμα από ανθρώπους και μηχανές και παρότι δεν προορίζονται για ανάγνωση προσφέρουν αυτή τη δυνατότητα στο χρήστη εάν κριθεί αναγκαίο.
- Είναι πλήρως συμβατή με Unicode, οπότε μπορεί να χειριστεί την πληροφορία που έχει γραφεί σε οποιαδήποτε ανθρώπινη γλώσσα. Παράλληλα, υποστηρίζει διεθνείς και τοπικές προσαρμογές.
- Χρησιμοποιώντας την XML, οι δημιουργοί των εγγράφων μπορούν να αναπαραστήσουν πολύπλοκες δομές δεδομένων, όπως λίστες, εγγραφές και δένδρα και να περιγράψουν οποιονδήποτε τύπο δεδομένων, συμπεριλαμβάνοντας μαθηματικούς τύπους, οδηγίες διαμόρφωσης λογισμικού, μουσική, αποδείξεις και οικονομικές αναφορές.

Επιπλέον απαιτείται - όπως είναι προφανές - ένας μηχανισμός για την επικοινωνία μεταξύ των επιμέρους τμημάτων του δικτύου. Μπορούν να χρησιμοποιηθούν διάφορα πρωτόκολλα όπως HTTP, TCP/IP, SMTP. Η αλληλεπίδραση μεταξύ των συμμετεχόντων τμημάτων βασίζονται στο πρωτόκολλο SOAP (Simple Object Access Protocol). Με το SOAP τα services μπορούν να ανταλλάσσουν απαραίτητες πληροφορίες με τη μορφή XML μηνυμάτων. Το SOAP αποτελεί ουσιαστικά τον φάκελο ο οποίος περιβάλλει τα XML μηνύματα για να μεταφερθούν. Το σχήμα 2.3.1 που παρατίθεται στη συνέχεια παρουσιάζει τις τεχνολογίες που καθορίζουν τον τρόπο με τον οποίο ένα μήνυμα μεταφέρεται από τον service requestor στον service provider.



2.3.1 Στοιβά τεχνολογιών για τη μεταφορά μηνυμάτων

(Πηγή:<http://www.rudibelotti.com/doc/projects/webservices/webservices.pdf>)

Ένα ακόμα ζήτημα είναι η περιγραφή των services και ιδίως των αντίστοιχων interfaces ώστε να γνωστοποιείται ο τρόπος κλήσης τους, με ένα τυποποιημένο τρόπο. Το σκοπό αυτό εξυπηρετεί η WSDL (Web Service Description Language).

Η WSDL είναι μια γλώσσα για την περιγραφή interfaces που βασίζεται στην XML. Τέλος αυτό που απομένει είναι μια υπηρεσία καταγραφής των διαθέσιμων services

και των περιγραφών τους, ένα είδος καταλόγου-οδηγού για web services. Απαιτείται συνεπώς ένας τυποποιημένος τρόπος για τη δημοσίευση και τον εντοπισμό των services. Το ρόλο αυτό καλύπτει το (UDDI) Universal Description, Discovery and Integration project. Ακολουθούν αναλυτικότερα στοιχεία για τις αναφερόμενες στο κεφάλαιο αυτό τεχνολογίες και πρότυπα.

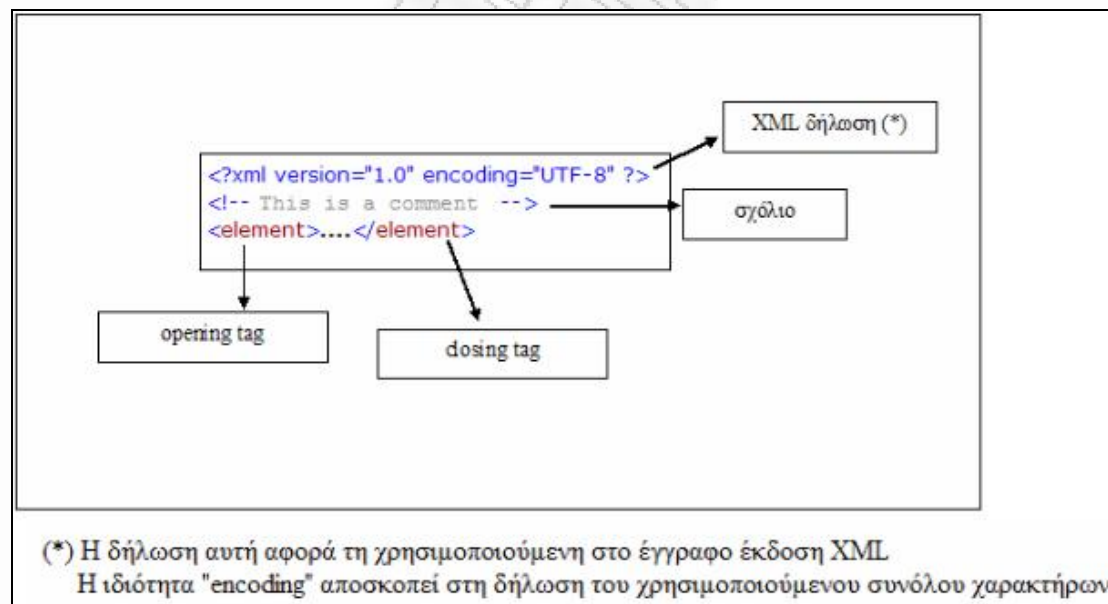
2.3.1. Ορθότητα XML εγγράφου

Για να είναι ένα έγγραφο XML σωστό, θα πρέπει να είναι συγχρόνως :

- Καλοσχηματισμένο (well-formed), δηλαδή να υπακούει σε όλους τους συντακτικούς κανόνες της XML. Εάν ένα έγγραφο δεν είναι συντακτικά σωστό, τότε δεν θεωρείται έγγραφο XML, η ανάλυση (parsing) σταματά και ο αναλυτής (parser) δίνει ένα σφάλμα.
- Έγκυρο (valid), δηλαδή να ικανοποιεί ένα σύνολο κανόνων που έχει ορίσει ο χρήστης και περιλαμβάνεται σε προαιρετικά έγγραφα XML, όπως τους ορισμούς Τύπων Εγγράφων (Document Type Definitions – DTDs) και τα Σχήματα XML (XML Schemas).

2.3.2. Ετικέτες - Tags

Στην XML η πληροφορία προσδιορίζεται με την βοήθεια ετικετών (tags), δηλ. με ένα αναγνωριστικό περικλειόμενο από <>. Με τα tags επιτυγχάνεται το “markup”. Στο 2.3.2 παρουσιάζονται κάποια βασικά στοιχεία ενός XML εγγράφου.



Σχήμα 2.3.2 Βασικά στοιχεία XML εγγράφου

(Πηγή: <http://www.dblab.ntua.gr/pubs/uploads/DIPL-2005-2.pdf>)

Με την ενθυλάκωση (nesting) των tags δίνεται η δυνατότητα να κατασκευασθούν στοιχεία με πολύπλοκη εσωτερική δενδρική δομή, π.χ.

```
<message>
  <to>you@yourAddress.com</to>
  <from>me@myAddress.com</from>
```

```
<subject>XML Is Really Cool</subject>
<text>
  How many ways is XML cool? Let me count
the ways...
</text>
<another_text/>
</message>
```

2.3.3. Μεταβλητές Ιδιοτήτων - Attributes

Κάθε στοιχείο (element) είναι δυνατόν να έχει και ιδιότητες (attributes) οι μεταβλητές των οποίων μαζί με τις τιμές τους συντάσσονται σύμφωνα με το παράδειγμα

```
<movie type="comedy" rating="for adults" year="1998"> .... </movie>
```

2.3.4. XML - Namespaces

Μια σημαντική ιδιότητα των XML-εγγράφων είναι ότι μπορούν να χρησιμοποιηθούν ώστε να συνθέσουν ένα νέο έγγραφο. Ο μηχανισμός των namespaces έχει ως σκοπό την αντιμετώπιση του προβλήματος της σύγκρουσης elements με κοινά ονόματα, τα οποία μπορεί να προέρχονται από διαφορετικά έγγραφα. Συγκεκριμένα ένα element συμπληρώνεται με κάποιο επιπρόσθετο αναγνωριστικό που να του αποδίδει τη μοναδικότητα. Ως αναγνωριστικό χρησιμοποιείται κάποιο Uniform Resource Identifier (URI). Για παράδειγμα στην αρχή ενός XML εγγράφου μπορεί να οριστούν τα namespaces ως εξής :

```
xmlns:it= "http://www.italian_teams"
xmlns:ru= "http://www.russian_teams.com"
```

και στο εξής να γίνεται αναφορά στο έγγραφο μέσω it:teams, ru:teams .

2.3.5. XML - Shema

Το XML-Schema είναι μια προδιαγραφή στην οποία κατέληξε η απόπειρα της κοινότητας της XML υπό την ηγεσία του W3C για τη δημιουργία μιας μεταγλώσσας για την περιγραφή της δομής των XML εγγράφων και των τύπων δεδομένων σε αυτά. Τα XML-schemas είναι και αυτά εκφρασμένα σε XML. Με τον τρόπο αυτό εξαλείφεται η ανάγκη ύπαρξης ειδικών parsers που να τα διαβάζουν. Η ορολογία-λεξιλόγιο του XML-Schema καθορίζεται και αυτή χρησιμοποιώντας κάποιο schema. Το XML-Schema αποτελεί εκτός από τρόπο περιγραφής και τρόπο επικύρωσης της καλής μορφής (well-formedness) και εγκυρότητας (validity) ενός XML εγγράφου.

Στο σχήμα 2.3.3 που ακολουθεί παρατίθενται ένα απλό XML έγγραφο και το αντίστοιχο του XMLSchema έγγραφο.

XML έγγραφο

```
<?xml version="1.0" encoding="UTF-8" ?>
- <example xmlns:example="http://www.example.com/example"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.example.com/schema/example.xsd">
  <element1>Aa</element1>
  <element2>11</element2>
</example>
```

Αντίστοιχο XML-Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <xsd:element name="example">
- <xsd:complexType>
- <xsd:sequence>
  <xsd:element name="element1" type="xsd:string" />
  <xsd:element name="element2" type="xsd:integer" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

Σχήμα 2.3.3 Παράδειγμα XML εγγράφου και του αντίστοιχου XML – Schema

(Πηγή: <http://www.dblab.ntua.gr/pubs/uploads/DIPL-2005-2.pdf>)

2.3.6. XML Parsers (λεκτικοί αναλυτές XML)

Μια εφαρμογή που χρησιμοποιεί έγγραφα XML πρέπει να είναι σε θέση να αναλύει το έγγραφο στα στοιχεία που το συνθέτουν ώστε να αποσπά από αυτό και την απαραίτητη από το έγγραφο πληροφορία. Αυτό γίνεται μέσω της διαδικασίας του parsing. Το parsing αναλύει το κείμενο σε επιμέρους τμήματα τα οποία είναι αναγνωρίσιμα (start tags, end tags, attributes και τιμές αυτών, κείμενο, σχόλια). Η εφαρμογή μπορεί να κάνει χρήση κάποιου από τα καθορισμένα Application Programming Interfaces (APIs) που υλοποιούν κάποιο μοντέλο parsing. Έχουν αναπτυχθεί δύο APIs για το parsing: το Document Object Module (DOM) και το Simple API for XML (SAX).

Στο DOM ο parser διαβάζει όλο το έγγραφο και παράγει μια δομή δεδομένων (parse tree) που περιγράφει τα περιεχόμενα του εγγράφου (elements, attributes κλπ). Στο SAX ο parser ενημερώνει την εφαρμογή για τα στοιχεία που συναντά καθώς διαβάζει σειριακά ένα έγγραφο. Αυτό το είδος parsing είναι γνωστό και ως βασισμένο στα γεγονότα (event-based).

Τα APIs αυτά έχουν υλοποιηθεί και σε Java. Η κοινότητα της Java έχει αναπτύξει και την προδιαγραφή JDOM η οποία βασίζεται στη φιλοσοφία του DOM και έχει το πλεονέκτημα ότι χειρίζεται τα parse trees με αντικείμενα περισσότερο συμβατά με τις δομές δεδομένων της Java.

Με τη χρήση APIs και άλλων προδιαγραφών που έχουν υλοποιηθεί σε γλώσσες προγραμματισμού όπως η Java, οι εφαρμογές είναι σε θέση εκτός από το parsing, να δημιουργούν και να επεξεργάζονται έγγραφα XML για τις ανάγκες τους.

2.4. SOAP (Simple Object Access Protocol)

Το Simple Object Access Protocol (SOAP) είναι η τεχνολογία για την ανταλλαγή μηνυμάτων που διατυπώνονται ως XML μέσω του διαδικτύου. Σαν πρωτόκολλο επικοινωνίας μπορεί να χαρακτηριστεί ως “stateless” και μονόδρομο, αγνοεί τη σημασιολογία των μηνυμάτων που μεταφέρει και σχεδιάστηκε για να υποστηρίξει χαλαρά συνδεδεμένες εφαρμογές που αλληλεπιδρούν ανταλλάσσοντας ασύγχρονα μηνύματα. Συγκεκριμένα, το πρωτόκολλο SOAP καθορίζει τα ακόλουθα :

- Μια μορφή μηνύματος για μονόδρομη επικοινωνία, που περιγράφει πώς η πληροφορία μπορεί να ενσωματωθεί σε ένα έγγραφο XML. Ουσιαστικά, είναι ένας φάκελος (envelope), στον οποίο η εφαρμογή περικλείει οποιαδήποτε πληροφορία πρέπει να σταλεί.
- Ένα σύνολο συμβάσεων για τη χρήση μηνυμάτων SOAP στην υλοποίηση του προτύπου αλληλεπίδρασης RPC (Remote Procedure Call), που προσδιορίζουν πώς οι πελάτες μπορούν να καλέσουν μια απομονωμένη διαδικασία στέλνοντας ένα μήνυμα SOAP και πώς τα προγράμματα υπηρεσιών μπορούν να ανταποκριθούν στέλνοντας ένα άλλο SOAP μήνυμα σε αυτόν που τα κάλεσε.
- Ένα σύνολο κανόνων, τους κανόνες κωδικοποίησης SOAP (SOAP encoding rules), τους οποίους κάθε οντότητα που επεξεργάζεται ένα μήνυμα SOAP πρέπει να ακολουθεί. Συγκεκριμένα, οι κανόνες καθορίζουν τα στοιχεία XML που μια οντότητα πρέπει να διαβάσει και να καταλάβει, καθώς και τις ενέργειες στις οποίες μια οντότητα πρέπει να προβεί εάν δεν καταλάβει το περιεχόμενο.
- Μια περιγραφή του τρόπου με τον οποίο ένα μήνυμα SOAP θα πρέπει να μεταφέρεται πάνω από πρωτόκολλα μεταφοράς, όπως το HTTP και το SMTP.

Ένα μήνυμα SOAP συνίσταται από το φάκελο SOAP <Envelope>, ο οποίος περιέχει δύο μέρη : την επικεφαλίδα <Header> και το κυρίως τμήμα <Body>. Η επικεφαλίδα (header) είναι προαιρετική και χρησιμοποιείται για να επεκτείνει το μήνυμα προσθέτοντας μεταδεδομένα, τα οποία θα μπορούν εναλλακτικά να επεξεργαστούν και οι ενδιαμέσοι κόμβοι. Γενικά, περιλαμβάνει πληροφορίες σχετικές με το μήνυμα, όπως την ημερομηνία που στάλθηκε, το χρόνο που εκπνέει ή δεδομένα ταυτοποίησης (authentication), εξουσιοδότησης (authorization) και χειρισμού δοσοληψιών (transactional interaction). Το κυρίως σώμα (body) είναι υποχρεωτικό και περιέχει την πληροφορία που μεταφέρεται σε XML. Τόσο η επικεφαλίδα όσο το κυρίως τμήμα ενδέχεται να περιέχουν πολλαπλά υπομέρη στη μορφή «μπλοκ» (header blocks ή body blocks ανάλογα).

Η δομή ενός μηνύματος SOAP παρουσιάζεται στο επόμενο σχήμα 2.4.1 :

```

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Header>
  ...
  </soap:Header>

  <soap:Body>
  ...
  <soap:Fault>
  ...
  </soap:Fault>
  </soap:Body>

</soap:Envelope>

```

Σχήμα 2.4.1 Δομή μηνύματος SOAP

(Πηγή: http://www.w3schools.com/soap/soap_syntax.asp)

2.4.1. SOAP over HTTP

Τα μηνύματα SOAP μπορούν να μεταφερθούν μέσα στο δίκτυο με διάφορα πρωτόκολλα, όπως το SMTP και το FTP, αλλά κατά κανόνα προτιμάται το HTTP (HyperText Transfer Protocol). Ο λόγος είναι ότι το HTTP είναι το πρωτόκολλο που χρησιμοποιείται ευρέως στο Διαδίκτυο (internet), καθώς χρησιμοποιείται από τους browsers για την πλοήγηση στον παγκόσμιο ιστό. Τα δεδομένα μεταφέρονται ακριβώς όπως οι ιστοσελίδες και δεν εμποδίζονται από τείχη ασφαλείας (firewalls) που ενδέχεται να διαθέτουν οι υπολογιστές.

Το μήνυμα SOAP αποστέλλεται σαν μέρος του μηνύματος αίτησης ή απόκρισης HTTP (HTTP request ή HTTP response, αντίστοιχα) και προστίθενται στο μήνυμα και οι επικεφαλίδες HTTP (HTTP headers). Γενικά, το πρωτόκολλο HTTP χρησιμοποιεί δύο τρόπους για να στέλνει δεδομένα : HTTP – POST και HTTP – GET, με τον πρώτο να προτιμάται για τις Διαδικτυακές υπηρεσίες, διότι σύμφωνα με αυτόν το μήνυμα SOAP μεταφέρεται στο κυρίως σώμα (body) του μηνύματος HTTP.

2.4.2. SOAP Exceptions and Errors

Οι κανόνες κωδικοποίησης SOAP (SOAP encoding rules) παρέχουν στο πρότυπο SOAP τη δυνατότητα χειρισμού λαθών, τα οποία μπορεί να προκύψουν τόσο κατά τη μεταφορά όσο κατά την επεξεργασία του μηνύματος. Στην πρώτη κατηγορία περιλαμβάνονται η μη αναγνωσιμότητά του μηνύματος και η αλλοίωση κάποιων χαρακτήρων του μηνύματος, ενώ στη δεύτερη κατατάσσονται οι λανθασμένες τιμές σε κάποια πεδία της επικεφαλίδας (SOAP header) ή των δεδομένων που αποστέλλονται, η μη ταυτοποίηση του αποστολέα και τα λάθη κατά τη σύνδεση με τη βάση δεδομένων.

Οι πληροφορίες λάθους περιέχονται εντός του κυρίως σώματος (SOAP Body) στο στοιχείο XML <fault>, το οποίο αποτελείται από τα υποχρεωτικά στοιχεία : <faultcode> που προσδιορίζει πού συνέβη το λάθος, <faultstring> το οποίο περιγράφει πώς δημιουργήθηκε λάθος, και τα προαιρετικά στοιχεία : <faultactor>

και <detail> που παρέχουν περαιτέρω λεπτομέρειες. Πιο συγκεκριμένα, ανάλογα με την τιμή του <faultcode> διακρίνονται δύο γενικές κατηγορίες σφαλμάτων :

- <soap:Server> που αναφέρεται σε λάθη για τα οποία δεν ευθύνεται το μήνυμα που έστειλε ο πελάτης της διαδικτυακής υπηρεσίας και δεν μπορούν να διορθωθούν με ενέργειες του client. Εάν αποσταλεί το ίδιο μήνυμα αργότερα, ενδέχεται να επιτύχει η υπηρεσία.
- <soap:Client> δηλαδή λάθη τα οποία οφείλονται στο περιεχόμενο του μηνύματος που έστειλε ο client και μπορεί να βρίσκονται στην επικεφαλίδα ή / και στο κυρίως μήνυμα.

```
<soap:Body>
  <soap:Fault>
    <faultcode>soap:Server</faultcode>
    <faultstring>Server Error</faultstring>
  </soap:Fault>
</soap:Body>
```

Σχήμα 2.4.2 Το στοιχείο Fault

(Πηγή: http://www.ebusiness-lab.gr/files/dmdocuments/Ptyxiakes-old/PDFs/Delis_Kiriakoulis_%CE%A4%CE%B5%CF%87%CE%BD%CE%BF%CE%BB%CE%BF%CE%B3%CE%AF%CE%B1%20Web%20Services.pdf)

2.4.3. SOAP Extensions

Οι προεκτάσεις SOAP (SOAP Extensions) είναι ένα είδος component που επισυνάπτεται σε μια διαδικτυακή εφαρμογή που ανταλλάσσει μηνύματα SOAP με κάποια εξωτερική οντότητα και μπορεί να εφαρμόζεται σε κάθε SOAP μήνυμα που λαμβάνεται και αποστέλλεται ή σε κάποια SOAP μηνύματα επιλεκτικά. Ανάλογα με τη λειτουργία που επιτελεί η προέκταση καθορίζεται εάν απαιτείται να τοποθετηθεί και από την πλευρά της εξωτερικής οντότητας. Η διαδικασία που ακολουθείται έχει ως εξής : οι προεκτάσεις SOAP (SOAP Extensions) παραλαμβάνουν τα εισερχόμενα μηνύματα SOAP και τα προ-επεξεργάζονται προτού τα παραδώσουν στον χειριστή ASMX για να τα επεξεργαστεί. Μόλις δημιουργηθεί μια απόκριση στην αίτηση – εξερχόμενο μήνυμα SOAP προωθείται στην επέκταση SOAP για περαιτέρω μετεπεξεργασία.

Η χρήση των προεκτάσεων SOAP (SOAP Extensions) συνοψίζεται στους ακόλουθους λόγους:

- Ενίσχυση της ασφάλειας με κρυπτογράφηση και αποκρυπτογράφηση του μηνύματος SOAP.
- Συμπίεση και αποσυμπίεση του μηνύματος SOAP.
- Μετατροπή των μηνυμάτων που προορίζονται για παλιότερες εκδόσεις στην τρέχουσα έκδοση της υπηρεσίας.
- Λεκτική ανάλυση των επικεφαλίδων SOAP για ταυτοποίηση και εξουσιοδότηση του πελάτη.

- Έλεγχος εάν τα μηνύματα SOAP ικανοποιούν τα κατάλληλα XML σχήματα.
- Επιβολή επιχειρηματικών κανόνων στις τιμές εισόδου και εξόδου.

2.4.4. SOAP Versioning

Ένα σημείο άξιο αναφοράς σχετικά με το SOAP είναι ότι το στοιχείο Envelope των μηνυμάτων SOAP δεν αποκαλύπτει ρητά κάποια έκδοση (version) του πρωτοκόλλου. Το versioning στην περίπτωση των μηνυμάτων SOAP γίνεται μέσω του URI του namespace του φακέλου του SOAP.

Συνεπώς για τις εκδόσεις του SOAP δεν τίθεται ζήτημα συμβατών ή μη αλλαγών στο πρωτόκολλο, αλλά μόνο αν δύο εκδόσεις είναι ίδιες ή διαφορετικές.

2.4.5. Χειρισμός λαθών (Error Handling)

Το SOAP δίνει τη δυνατότητα χειρισμού λαθών που προκύπτουν κατά τη μεταφορά των μηνυμάτων. Λάθη μπορούν να προκύψουν σε περίπτωση που ένα μήνυμα δεν είναι σε μορφή κατάλληλη για να «διαβαστεί» από κάποιον παραλήπτη, ο παραλήπτης δεν είναι σε θέση να «διαβάσει» ή να δέχεται μηνύματα, υπάρχει πρόβλημα στο authentication του αποστολέα. Το SOAP δίνει τη δυνατότητα ενημέρωσης για την εμφάνιση κάποιου error και την παροχή κάποιας περιγραφής για αυτό. Σε περίπτωση λάθους, αποστέλλεται από τον παραλήπτη του μηνύματος που περιέχει το λάθος, ένα μήνυμα απάντησης (επίσης SOAP message). Η πληροφορία σχετικά με το λάθος μεταφέρεται στο τυποποιημένο element SOAP Fault. Το element αυτό στο οποίο ενθυλακώνονται και άλλα τυποποιημένα elements (faultcode, faultstring, faultactor) περιέχει πληροφορία σχετικά με το λάθος, η οποία μπορεί να είναι και βοηθητική για την αντιμετώπισή του και για μια νέα απόπειρα επιτυχημένης αποστολής.

2.5. REST (Representational State Transfer)

Το Representational State Transfer (REST) είναι ένα αρχιτεκτονικό στυλ για κατανεμημένα συστήματα υπερμέσων (hypermedia), όπως ο Παγκόσμιος Ιστός. Οι όροι «Representational State Transfer» και «REST» παρουσιάστηκαν το 2000 στη διδακτορική διατριβή του Roy Fielding.

Η αρχιτεκτονική REST αποτελείται από εξυπηρετητές (servers) και πελάτες (clients). Οι clients εκκινούν την αποστολή αιτημάτων στους servers οι οποίοι με τη σειρά τους τα επεξεργάζονται προκειμένου να επιστρέψουν σε κατάλληλες απαντήσεις. Κάθε αίτημα ή μήνυμα προς τους εξυπηρετητές αποστέλλεται σε συγκεκριμένο αντικείμενο της εφαρμογής το οποίο ονομάζεται πόρος (resource). Κάθε πόρος έχει μοναδική διεύθυνση στην οποία αντιστοιχίζεται το κάθε αίτημα. Οι απαντήσεις εμφανίζουν στον πελάτη αναπαραστάσεις συγκεκριμένων πόρων για τη δεδομένη χρονική στιγμή του αιτήματος, αποτελούν δηλαδή ένα στιγμιότυπό τους. Η ονομασία της αρχιτεκτονικής REST προέρχεται από την έννοια της μεταφοράς κατάστασης στον πελάτη. Με κάθε νέο αίτημα επιστρέφεται μια ανάλογη απόκριση στον πελάτη με τη μορφή αναπαράστασης με αποτέλεσμα να μεταβαίνει αυτός σε μια νέα κατάσταση.

Βασικές αρχές του REST είναι:

- Μια υλοποίηση REST αποτελείται από συστατικά δικτύου (components) τα οποία διαιρούνται σε πόρους.
- Κάθε πόρος διευθυνσιοδοτείται μοναδικά με καθολικό ταυτοποιητή (URI)
- Η μεταφορά κατάστασης μεταξύ του πελάτη και του πόρου πραγματοποιείται μέσα από μια ομοιόμορφη διεπαφή (uniform interface), η οποία χαρακτηρίζεται από:
 - Ένα καθορισμένο σύνολο αυστηρά ορισμένων λειτουργιών
 - Ένα σύνολο από τύπους περιεχομένου

Τα δύο παραπάνω συστατικά αναδεικνύουν τη σημασιολογία της επικοινωνίας καθώς επίσης αποτελούν το πρότυπο για την ανταλλαγή πληροφορίας.

Η εφαρμογή της αρχιτεκτονικής γίνεται σε πρωτόκολλα τα οποία:

- Είναι Πελάτη – Εξυπηρετητή (client - server)
- Είναι άνευ κατάστασης (stateless)
- Δίνουν δυνατότητα αποθήκευσης περιεχομένου σε κρυφές μνήμες (Cacheable)
- Υποστηρίζουν διαστρωμάτωση (Layered)

Η δυνατότητα αποθήκευσης σε κρυφές μνήμες παρέχεται μέσα από το ίδιο το μήνυμα απόκρισης που αποστέλλει ο εξυπηρετητής στο περιεχόμενο του οποίου αναφέρεται εάν θα πρέπει να αποθηκευτεί προσωρινά.

Αναπτύχθηκε παράλληλα με το πρωτόκολλο HTTP / 1.1 ενώ η ύπαρξη του παγκόσμιου ιστού αποτελεί την μεγαλύτερη υλοποίηση συστήματος που ακολουθεί τις αρχές της χωρίς ωστόσο να αποτελεί το μοναδικό πεδίο εφαρμογής.

2.6. WSDL (Web Services Description Language)

Η Web Services Description Language (WSDL) είναι μια γλώσσα που βασίζεται στην XML για να καθορίσει τις Διαδικτυακές υπηρεσίες και να περιγράψει τους μηχανισμούς που πρέπει να τεθούν σε λειτουργία για να έχουν τα προγράμματα πελάτες πρόσβαση σε αυτές. Πιο συγκεκριμένα, στο έγγραφο WSDL καταγράφονται οι λειτουργίες που η διαδικτυακή υπηρεσία προσφέρει, τα πρωτόκολλα που μπορούν να χρησιμοποιηθούν για την κλήση της, η τοποθεσία της στο δίκτυο και το πώς δέχεται και επιστρέφει τα δεδομένα.

Προκειμένου να γνωρίζει ο πελάτης όλες τις απαραίτητες πληροφορίες για την κλήση μιας υπηρεσίας διαδικτύου, πρέπει να έχει στην κατοχή του μια περιγραφή αυτής. Για να γίνει η όλη διαδικασία ακόμη πιο εύκολη και πιο τυποποιημένη χρειάζεται ένας καθορισμένος μηχανισμός για την περιγραφή των υπηρεσιών διαδικτύου. Αυτός ο μηχανισμός θα παρέχει πληροφορίες για ακριβώς αυτά που χρειάζεται να ξέρει ο χρήστης που θέλει να καλέσει την υπηρεσία. Επιπρόσθετα, χάρη σε αυτόν το μηχανισμό είναι δυνατή η ανάπτυξη εφαρμογών που θα βοηθούν τους προγραμματιστές στην ανάπτυξη των υπηρεσιών μέσω των περιγραφών τους.

Όπως έχει αναφερθεί, ο ρόλος της υπηρεσίας καταλόγου (Service Registry) σε μια SOA αρχιτεκτονική είναι πολύ σημαντικός. Ο πελάτης, προκειμένου να ανακαλύψει

την υπηρεσία που τον ενδιαφέρει, ψάχνει πρώτα στον κατάλογο αυτό. Το αποτέλεσμα της αναζήτησης δεν είναι τίποτε άλλο από την περιγραφή της υπηρεσίας που ζήτησε. Ο πάροχος της υπηρεσίας διαδικτύου δημοσιεύει την περιγραφή της υπηρεσίας του γι' αυτό το λόγο. Για να ξέρει ο κάθε πιθανός πελάτης πώς να εγκαθιδρύσει επικοινωνιακή σύνδεση μαζί του και πώς ακριβώς να καλέσει την υπηρεσία που του παρέχει.

Η περιγραφή μιας υπηρεσίας διαδικτύου πρέπει να αποτελείται από δύο μέρη:

- Τη λειτουργική περιγραφή – Η περιγραφή αυτή περιγράφει ποιες ακριβώς λειτουργίες είναι διαθέσιμες από την υπηρεσία διαδικτύου και ποια είναι η απαιτούμενη σύνταξη που θα πρέπει να έχει το SOAP μήνυμα προκειμένου να γίνει με επιτυχία η κλήση. Επίσης, πληροφορίες σχετικές με την τοποθεσία της υπηρεσίας είναι διαθέσιμες ώστε ο χρήστης να έχει το ακριβές URL στο οποίο θα γίνει η κλήση. Είναι χρέος της περιγραφής αυτής να δώσει όλες τις απαραίτητες πληροφορίες στον πελάτη ώστε να καλέσει την υπηρεσία.
- Τη μη-λειτουργική περιγραφή – Η λειτουργική περιγραφή είναι πολύ σημαντική, αφού χωρίς αυτή δεν είναι δυνατόν να επιτευχθεί η κλήση της υπηρεσίας. Η μη-λειτουργική περιγραφή όμως είναι εξίσου σημαντική. Όπως υποδηλώνει και το όνομά της, προσφέρει πληροφορίες που σχετίζονται με μη λειτουργικές απαιτήσεις της υπηρεσίας. Τέτοιες πληροφορίες μπορεί να περιγράφουν το λόγο που κάποιος πελάτης να επιθυμεί να καλέσει την υπηρεσία. Επίσης, μπορεί να περιγράφουν τον παροχέα της υπηρεσίας με αρκετή λεπτομέρεια. Μπορεί να προσφέρουν προσωπικά του στοιχεία και τηλέφωνα επικοινωνίας. Τέλος, πληροφορίες για την ασφάλεια γύρω από την υπηρεσία και την πολιτική του παροχέα μπορεί να είναι διαθέσιμες μέσω της μη λειτουργικής περιγραφής της υπηρεσίας.

Η είναι μια γλώσσα βασισμένη στην XML και περιγράφει τρεις σημαντικές ιδιότητες της υπηρεσίας:

- Τι κάνει η υπηρεσία – Εδώ περιγράφονται οι μέθοδοι της υπηρεσίας, οι παράμετροί της και τα αποτελέσματα που επιστρέφει.
- Πώς γίνεται η πρόσβαση στην υπηρεσία – Εδώ παρέχονται όλες οι απαραίτητες πληροφορίες για τη σύνταξη των SOAP μηνυμάτων που θα ανταλλάγουν, καθώς και τα πρωτόκολλα που υποστηρίζονται από τον παροχέα.
- Που βρίσκεται η υπηρεσία – Εδώ παρέχονται πληροφορίες για τη δικτυακή διεύθυνση της υπηρεσίας. Συνήθως είναι ένα URL.

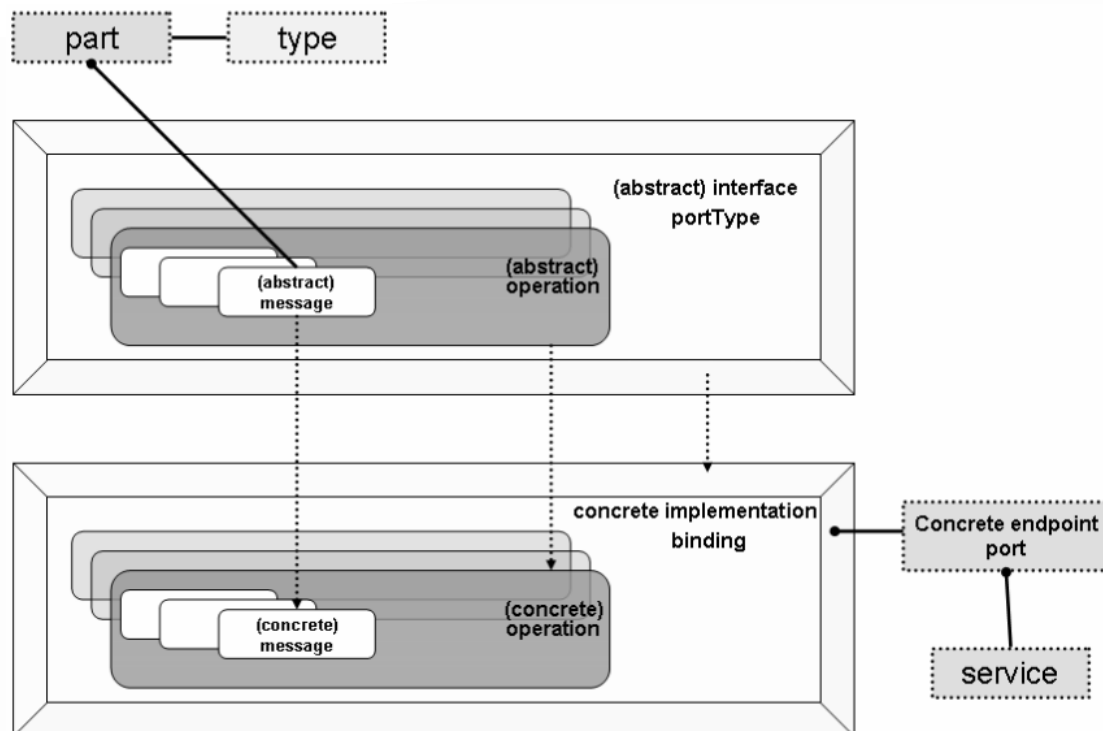
Όπως κάθε καλά ορισμένη γλώσσα βασισμένη σε XML, έτσι και η WSDL ορίζει κάποια βασικά στοιχεία που συναντώνται σε ένα WSDL έγγραφο. Τα κυριότερα από αυτά είναι:

- portType: Είναι ένας αφαιρετικός ορισμός της διασύνδεσης της υπηρεσίας διαδικτύου. Ουσιαστικά, σκοπός του είναι η περιγραφή αυτής της διασύνδεσης. Όπως γίνεται σε έναν κώδικα Java, που η διασύνδεση αποτελείται μόνο από τις υπογραφές των μεθόδων, έτσι κι εδώ ο ορισμός του portType είναι η συλλογή των λειτουργιών που παρέχει η υπηρεσία διαδικτύου. Ένα τέτοιο στοιχείο μπορεί να έχει ένα απλό όνομα, το οποίο πιθανόν να σχετίζεται με τη λειτουργία που προσφέρει η υπηρεσία. Εάν ένα

έγγραφο WSDL περιέχει πολλά τέτοια στοιχεία, τότε το κάθε ένα οφείλει να έχει διαφορετικό όνομα.

- **Message:** Καθορίζει τη μορφή του μηνύματος, καθώς και το σύνολο των παραμέτρων που σχετίζονται με τις μεθόδους της υπηρεσίας. Το στοιχείο αυτό μπορεί να αποσυντεθεί σε πολλά μέρη, που ονομάζονται parts. Κάθε στοιχείο message μέσα σε ένα WSDL έγγραφο πρέπει να έχει ένα μοναδικό όνομα ώστε να ξεχωρίζει από τα υπόλοιπα. Ουσιαστικά, τα στοιχεία αυτά δεν παρουσιάζουν ιδιαίτερο ενδιαφέρον αφού δεν είναι τίποτε άλλο από μια συλλογή από parts και κάθε part έχει ένα όνομα, που συνήθως αντικατοπτρίζει την πληροφορία που περιέχεται στο part.
- **Types:** Ορίζει τη συλλογή όλων των τύπων δεδομένων που χρησιμοποιούνται από την υπηρεσία διαδικτύου. Το εξ ορισμού σύστημα των τύπων δεδομένων στην WSDL είναι το XML σχήμα (XML Schema – XSD). Το σχήμα αυτό μπορεί να χρησιμοποιηθεί για να περιγράψει όλους τους τύπους δεδομένων που μπορεί να χρησιμοποιηθούν στα διάφορα μηνύματα που ανταλλάσσονται για την κλήση της υπηρεσίας. Το στοιχείο types είναι ένα μέρος για το WSDL έγγραφο στο οποίο ο χρήστης ορίζει τύπους δεδομένων που θα χρησιμοποιηθούν σε άλλα στοιχεία του εγγράφου.
- **Binding:** Περιέχει λεπτομέρειες για το πώς τα στοιχεία σε μια αφαιρετική δομή (όπως το portType) μπορούν να μετασχηματισθούν σε μια συμπαγή αναπαράσταση των δεδομένων και πρωτοκόλλων που θα χρησιμοποιηθούν για την επικοινωνία μεταξύ του πελάτη και της υπηρεσίας. Ουσιαστικά είναι το στοιχείο που θα πει στον πελάτη πώς ακριβώς να μορφοποιήσει το μήνυμα που θα στείλει για την κλήση της υπηρεσίας. Κάθε στοιχείο portType μπορεί να έχει ένα ή περισσότερα στοιχεία binding συσχετισμένα με αυτό. Για ένα συγκεκριμένο στοιχείο portType, ένα στοιχείο binding περιγράφει τον τρόπο που πρέπει να γίνει η κλήση των μεθόδων της υπηρεσίας χρησιμοποιώντας κάποιο καθορισμένο πρωτόκολλο, όπως το SOAP, πάνω σε κάποιο επίσης καθορισμένο πρωτόκολλο μεταφοράς, όπως το HTTP. Το όνομα του binding πρέπει να είναι μοναδικό καθώς πολλά τέτοια στοιχεία μπορεί να υπάρχουν σε ένα WSDL έγγραφο.
- **Port –** Περιγράφει πώς ακριβώς εκφράζεται ένα binding σε μια φυσική θύρα του δικτύου. Ο ρόλος της είναι πολύ απλός αφού δεν κάνει τίποτε άλλο από μια απλή αντιστοιχία. Τα στοιχεία αυτά, όπως και τα προηγούμενα, οφείλουν να φέρουν ένα όνομα, μοναδικό μέσα στο WSDL έγγραφο. Συνήθως, τα στοιχεία αυτά υποδεικνύουν το URL στο οποίο πρέπει να σταλούν τα μηνύματα SOAP ώστε να γίνει η κλήση της υπηρεσίας. Τα στοιχεία αυτά δε συναντώνται μόνα τους μέσα στο έγγραφο της περιγραφής της υπηρεσίας. Είναι στοιχεία – παιδιά του συστατικού service.
- **Service –** Ένα τέτοιο στοιχείο είναι μια συλλογή από ports. Το στοιχείο αυτό μπορεί να έχει ένα μοναδικό όνομα μέσα στο έγγραφο. Παρόλο που ο ρόλος του δεν έχει ιδιαίτερη σημασία, είναι καλό τα στοιχεία port να ομαδοποιούνται κάτω από ένα στοιχείο service. Συνήθως όμως ένα μόνο στοιχείο port συναντάται και έτσι καταλήγουμε στο να έχουμε ένα στοιχείο service, που να περιέχει ένα στοιχείο port.

Το σχήμα που ακολουθεί απεικονίζει τις συσχετίσεις μεταξύ των συστατικών της WSDL.



Σχήμα 2.6.1 Το πληροφοριακό μοντέλο της WSDL

(Πηγή: http://www.ebusiness-lab.gr/files/dmdocuments/Ptyxiakes-old/PDFs/Delis_Kiriakoulis_%CE%A4%CE%B5%CF%87%CE%BD%CE%BF%CE%BB%CE%BF%CE%B3%CE%AF%CE%B1%20Web%20Services.pdf)

Η δομή ενός WSDL εγγράφου ξεκινά με ένα στοιχείο ορισμών, το οποίο καλείται definitions. Το στοιχείο αυτό περιέχει όλα τα υπόλοιπα στοιχεία που μπορεί να συναντηθούν μέσα στο έγγραφο, δηλαδή μπορεί να περιέχει:

- Στοιχεία τεκμηρίωσης. Τα στοιχεία αυτά καλούνται documentation elements και χρησιμοποιούνται για να παρέχουν χρήσιμες πληροφορίες κατανοητές από τον άνθρωπο γύρω από την υπηρεσία διαδικτύου.
- Στοιχεία εισαγωγών. Τα στοιχεία αυτά καλούνται import elements. Τα στοιχεία δίνουν τη δυνατότητα επαναχρησιμοποίησης ήδη υπαρχόντων WSDL εγγράφων επιτρέποντας σε διάφορα στοιχεία να έχουν αναφορές σε άλλα ήδη υπάρχοντα που βρίσκονται σε διαφορετικό έγγραφο σε διαφορετικό αρχείο.
- Προαιρετικά ένα στοιχείο types.
- Στοιχεία message.
- Στοιχεία portType. Συνήθως μόνο ένα συναντάται.
- Στοιχεία binding. Συνήθως μόνο ένα συναντάται, το οποίο έχει να κάνει με το στοιχείο portType.
- Στοιχεία service. Και πάλι συνήθως μόνο ένα είναι αρκετό.

2.7. UDDI (Universal Description, Discovery and Integration)

Το Universal Description, Definition and Integration (UDDI) αποτελεί ένα πρωτόκολλο καταχώρησης για Διαδικτυακές υπηρεσίες. Σκοπός του είναι να διευκολύνει την ανακάλυψη υπηρεσιών και κατά το χρόνο σχεδίασης, και δυναμικά κατά το χρόνο εκτέλεσης. Συνεπώς το UDDI λειτουργεί ως ένα κοινό άμεσα συνδεδεμένο κατάλογο (και των αντιστοίχων υπηρεσιών). Ο κατάλογος αυτός, συνήθως αναφέρεται ως κατάλογος επιχειρήσεων UDDI (UDDI Business Registry).

Για να γίνει κάποιος διαχειριστής ενός τέτοιου καταλόγου, πρέπει να ακολουθήσει αυστηρές συμφωνίες για την αντιγραφή των δεδομένων, το απόρρητο των δεδομένων και τις διάφορες πολιτικές. Από την οπτική μιας καταχωρημένης επιχείρησης αλλά και από την οπτική του απλού χρήστη, η απαίτηση είναι να μην υπάρχει διαφορά ανάμεσα στα διαφορετικά αντίγραφα του καταλόγου. Οι επιχειρήσεις μπορούν να κάνουν εγγραφή σε οποιονδήποτε διαχειριστή UDDI, και τα δεδομένα τους θα αντιγραφούν σε όλα τα άλλα αντίγραφα του καταλόγου στους άλλους διαχειριστές.

Επομένως, οι χρήστες μπορούν να αναζητήσουν στον κατάλογο οποιουδήποτε διαχειριστή για να βρουν τις επιχειρήσεις που αναζητούν, άσχετα με το διαχειριστή που έχει επιλέξει η συγκεκριμένη επιχείρηση για να εγγραφεί.

Το UDDI είναι κάτι παραπάνω από ένα κατάλογο επιχειρήσεων και υπηρεσιών. Ορίζει ένα σύνολο από δομές δεδομένων και προδιαγραφές API, για την προγραμματική αναζήτηση και καταχώρηση εταιριών, υπηρεσιών, δεσμών, και ειδών υπηρεσιών. Στα τυπικά σενάρια υπηρεσιών διαδικτύου, οι παροχείς υπηρεσιών διαδικτύου θα θελήσουν να δημοσιεύσουν τις περιγραφές των εταιριών και των υπηρεσιών τους σε ένα κατάλογο, και όσοι αναζητούν υπηρεσίες, είτε κατά τη σχεδίαση, είτε κατά την εκτέλεση, θα θελήσουν να αναζητήσουν στον κατάλογο περιγραφές υπηρεσιών. Οι προδιαγραφές API του UDDI παρέχουν γι' αυτό το λόγο ένα σύνολο από προγραμματιστικές διεπαφές (API) δημοσίευσης για την καταχώρηση υπηρεσιών, και αντίστοιχες διεπαφές ερωτήσεων API για την εύρεση υπηρεσιών.

Επιπλέον από την παροχή μιας προγραμματιστικής διεπαφής, οι διαχειριστές του καταλόγου UDDI, παρέχουν ένα σύστημα αλληλεπίδρασης με το χρήστη, βασισμένο στο διαδίκτυο, για την καταχώρηση, διαχείριση, ανεύρεση εταιριών και υπηρεσιών μέσα στον κατάλογο.

Ένας απλός τρόπος κατηγοριοποίησης της πληροφορίας που περιέχεται σε μια υπηρεσία καταγραφής (registry) UDDI παρουσιάζει ομοιότητες με τον τηλεφωνικό κατάλογο:

- Λευκές σελίδες, στις οποίες οι πελάτες UDDI μπορούν να βρουν τις Διαδικτυακές υπηρεσίες που προσφέρονται από μια συγκεκριμένη επιχείρηση.
- Κίτρινες σελίδες, όπου είναι δυνατή η αναζήτηση υπηρεσιών ανάλογα με την κατηγορία στην οποία ανήκουν, σύμφωνα με κάποιο σχήμα ταξινόμησης.
- Πράσινες σελίδες, που περιγράφουν πώς μπορεί να καλέσει ένα πρόγραμμα πελάτη μια συγκεκριμένη διαδικτυακή υπηρεσία.

Κεφάλαιο 3. Ερευνητική Μεθοδολογία

Η Ερευνητική Μεθοδολογία ορίζεται ως μια δομημένη συλλογή από οδηγίες ή/και δραστηριότητες η οποία αποβλέπει στην εποικοδομητική παραγωγή αξιόπιστων αποτελεσμάτων (Mingers, 2001). Αποτελεί γεγονός ότι κατά τη διάρκεια εκπόνησης των μελετητικών τμημάτων έργων που αφορούν σε ανάπτυξη Ολοκληρωμένων Συστημάτων διευκολύνει η επιλογή μιας μοναδικής μεθοδολογικής προσέγγισης καταγραφής και ανάλυσης των απαιτήσεων των χρηστών με στόχο τη μεγιστοποίηση της ακρίβειας των αποτελεσμάτων (J. McGrath, 1982).

Κατόπιν σφαιρικής εξετάσεως και προσεγγίσεως του κοινού-στόχου αυτής της εργασίας, κρίθηκε σκόπιμο να υιοθετηθεί συγκεκριμένη μέθοδος έρευνας (research method), δηλαδή η αναζήτηση της πληροφορίας έγινε βάσει στρατηγικής η οποία ξεκίνησε από θεμελιώδεις φιλοσοφικές αρχές και συνδύασε σχεδιασμό έρευνας και συλλογή δεδομένων.

Ως γνωστό, οι μέθοδοι έρευνας κατηγοριοποιούνται με ποικίλους τρόπους, ωστόσο η πιο κοινή διάκριση είναι εκείνη μεταξύ των δυο επικρατέστερων: της ποιοτικής (qualitative) και της ποσοτικής (quantitative) μεθόδου έρευνας. Βέβαια, παρόλο που οι περισσότεροι ερευνητές υιοθετούν είτε την ποσοτική είτε την ποιοτική, μερικοί προτείνουν συνδυασμό μιας ή περισσοτέρων μεθόδων στην μελέτη τους (ο αποκαλούμενος τριγωνισμός - triangulation), όμως στην παρούσα εργασία υιοθετήθηκε η μέθοδος της ποιοτικής έρευνας.

3.1. Ποιοτική Έρευνα

Η ποιοτική έρευνα έχει τις ρίζες της στην κοινωνιολογία και ασχολείται περισσότερο με την ανάπτυξη εννοιών και ιδεών παρά με την εφαρμογή ήδη προϋπαρχουσών εννοιών (Halfpenny, 1979).

Στην παρούσα εργασία η ποιοτική έρευνα περιέλαβε τη χρήση ποιοτικών δεδομένων, όπως συνεδριάσεις, τηλεφωνικές επικοινωνίες, ντοκουμέντα και δεδομένα εξέτασης άλλων παρόμοιων συστημάτων. Σημειωτέο πως τα ποιοτικά δεδομένα δεν είναι αριθμητικά ή μετρήσιμα και η πλειοψηφία των ερευνητών που ασπάζονται την ποιοτική μέθοδο προτιμούν τον όρο “εμπειρικό υλικό- empirical material” από τη λέξη “δεδομένα-data”.

Οι ερευνητές αυτής της μεθόδου ενδιαφέρονται περισσότερο για την κατανόηση και γνώση των ερωτηθέντων από το να κατηγοριοποιούν ανθρώπους ή γεγονότα σε όρους ακαδημαϊκής θεωρίας (Rubin & Rubin, 1995) και μεταχειρίζονται τους συμμετέχοντες ως συνδιαλεγόμενους συνεταίρους παρά ως πληροφοριακά αντικείμενα. Η προσέγγιση αυτή δίνει έμφαση στην “ακρόαση” δεδομένων και στην κατανόηση του νοήματος μέσα από τα συμφραζόμενα, όπου αυτό το νόημα προέρχεται από αλληλεπίδραση με τους συμμετέχοντες. Επίσης, οι πολλαπλές εμπειρίες αφηγούμενες από διαφορετικούς συμμετέχοντες είναι χαρακτηριστικό της ποιοτικής προσέγγισης.

Στη συγκεκριμένη εργασία χρησιμοποιούμε την ποιοτική προσέγγιση και όχι την ποσοτική διότι (Hancock et al, 2007):

- η ποιοτική προσέγγιση εστιάζει στο πώς οι άνθρωποι μπορούν και έχουν διαφορετικούς τρόπους αντίληψης μιας πραγματικότητας

- εστιάζει και ερευνά τις δηλώσεις της εμπειρίας των ατόμων και τα δεδομένα που δεν θα μπορούσαν να εκφραστούν επαρκώς με αριθμούς
- μελετά τις συμπεριφορές ατόμων στο φυσικό τους πλαίσιο-περιβάλλον και χρησιμοποιεί τις περιγραφές τους ως δεδομένα
- εστιάζει στην περιγραφή και ερμηνεία των ατόμων με την πιθανότητα ανάπτυξης μιας νέας θεωρίας και στην αξιολόγηση μιας ήδη υπάρχουσας διαδικασίας
- η ποιοτική προσέγγιση εφαρμόζει μια ευέλικτη αλλά συστηματική διαδικασία έρευνας

3.1.1. Πλεονεκτήματα και μειονεκτήματα ποιοτικής έρευνας

Παρόλη την αφθονία και σχετικότητα των δεδομένων, η ποιοτική προσέγγιση δεδομένων έχει τις αδυναμίες της και τους περιορισμούς της (Miles, 1979). Αρχικά, η συλλογή και ανάλυση ποιοτικών δεδομένων είναι απαιτητική και παίρνει υπερβολικό χρόνο, διότι μαζεύονται δεδομένα πολλών μορφών και ο ερευνητής κατακλύζεται από τον όγκο τους. Δεύτερον, δεν έχουν καθιερωθεί καλά τα όρια των διαφόρων μεθοδολογιών ανάλυσης ποσοτικών δεδομένων (Cavaye, 1996), και τέλος, μία συνηθισμένη κριτική που δέχονται οι ερευνητές που χρησιμοποιούν την ποιοτική μεθοδολογία είναι το ότι τα αποτελέσματα μιας ποιοτικής ανάλυσης δεν μπορούν να γενικευθούν σε ένα ευρύτερο πληθυσμό διότι το δείγμα ανθρώπων είναι συνήθως μικρό και οι συμμετέχοντες δεν διαλέγονται τυχαία (Hancock et al, 2007).

Πλεονεκτήματα	Μειονεκτήματα
<ul style="list-style-type: none"> • Διερεύνηση που χαρακτηρίζεται από λεπτομέρεια και βάθος • Είναι δυνατόν να οδηγήσει στη διερεύνηση φαινομένων, διαδικασιών και συμπεριφορών που δεν είχαν προβλεφθεί από πριν • Διερεύνηση της εμπειρίας των κοινωνικών υποκειμένων. Ο ερευνητής «βλέπει» και κατανοεί τον κόσμο μέσα από τα μάτια και την αντίληψη των κοινωνικών υποκειμένων • Γίνεται προσπάθεια για την αποφυγή a priori κρίσεων 	<ul style="list-style-type: none"> • Συνήθως αφορά σε μικρά δείγματα • Χαρακτηρίζεται από σχετικά περιορισμένες δυνατότητες γενίκευσης και σύγκρισης • Εξαρτάται αρκετά από τις προσωπικές αντιλήψεις του ερευνητή και από τα επικοινωνιακά του προσόντα • Η συμμετοχή ή η εμπλοκή του ερευνητή μπορεί να μεταβάλλει τα χαρακτηριστικά του υπό μελέτη κοινωνικού φαινομένου ή διαδικασίας

Πίνακας 4.1.2 Πλεονεκτήματα και Μειονεκτήματα της ποιοτικής έρευνας

(Πηγή: Ιωσηφίδης, 2008)

3.1.2. Τύποι ποιοτικής έρευνας

Υπάρχουν πολλές προσεγγίσεις στην ποιοτική έρευνα. Αυτό συμβαίνει επειδή δεν είναι όλες οι ερωτήσεις ποιοτικής έρευνας του ίδιου τύπου και επειδή οι ερευνητές προέρχονται από ένα ευρύ φάσμα παράδοσης ερευνών. Η επιλογή της κάθε προσέγγισης έρευνας θα επηρεάσει τον τρόπο συλλογής δεδομένων από τον ερευνητή (Hancock et al, 2007).

Η στρατηγική έρευνας ορίζεται ως «ένα προσεγμένο σχέδιο ή μέθοδος ειδικά για την πραγμάτωση ενός στόχου» (Webster, 1985).

Οι πιο σημαντικοί τύποι ποιοτικής έρευνας είναι οι εξής:

- Εθνογραφία (Ethnography)
- Θεμελιωμένη θεωρία (Grounded theory)
- Μελέτη Περίπτωσης (Case study research)
- Έρευνα Δράσης (Action research)
- Μελέτη πεδίου (Field study)
- Περιγραφή Εφαρμογής (Application Description)

3.2. Έρευνα Δράσης (Action Research)

Η έρευνα δράσης επιλέχθηκε διότι αναφέρεται ομόφωνα στη διεθνή βιβλιογραφία πως, σε συνδυασμό με ομάδες ενδιαφέροντος (focus groups) και συνεντεύξεις-διάλογους, είναι η καταλληλότερη καθώς επιτρέπει προσαρμοστικότητα στη μελέτη ώστε να βρίσκεται η καλύτερη δυνατή λύση στο πρόβλημα.

Μερικοί από τους ευρύτερα αποδεκτούς ορισμούς της Έρευνας Δράσης περιλαμβάνουν τα ακόλουθα:

«Η ενεργή έρευνα στοχεύει να συμβάλει αμφότερα στις πρακτικές ανησυχίες των ανθρώπων σε μια άμεση προβληματική κατάσταση και στους σκοπούς της κοινωνικής επιστήμης με κοινές συνεργασίες μέσα σε ένα αμοιβαία αποδεκτό εθνικό πλαίσιο.» (Rapoport 1970: 499)

«Η Έρευνα Δράσης είναι μια μορφή συλλογικής αυτο-αναστοχαστικής (selfreflective) έρευνας που αναλαμβάνεται από τους συμμετέχοντες στις κοινωνικές καταστάσεις, προκειμένου να βελτιώσουν τις κοινωνικές ή εκπαιδευτικές πρακτικές τους, καθώς επίσης και να κατανοήσουν τις πρακτικές τους και τις καταστάσεις στις οποίες πραγματοποιούνται». (Kemmis and McTaggart 1988: 56)

«Η έρευνα δράση είναι μια παρέμβαση μικρής κλίμακας στη λειτουργία του πραγματικού κόσμου και μια εξέταση από κοντά των επιδράσεων αυτής της παρέμβασης». (L. Cohen, L. Manion 2000:258)

Ο πιο επιγραμματικός ορισμός της Έρευνας Δράσης δίνεται από τον John Elliot (1991: 69): «Η Έρευνα Δράσης είναι η μελέτη μιας κοινωνικής κατάστασης, με στόχο τη βελτίωση της ποιότητας της δράσης στα πλαίσια αυτής της κατάστασης».

3.2.1. Στόχοι της Έρευνας Δράσης

Οι βασικοί στόχοι της Έρευνας Δράσης είναι:

1. η υποστήριξη των μελών μιας ομάδας στην αντιμετώπιση των προκλήσεων και των προβλημάτων ενός συστήματος και στην υλοποίηση καινοτομιών μετά από προβληματισμό και συζήτηση.
2. να προσφέρει μια στρατηγική στην ανάπτυξη του συστήματος.
3. να προσφέρει μια στρατηγική εκπαίδευσης για τους χρήστες και θεωρεί ότι το σημαντικότερο πεδίο εκπαίδευσης είναι η πρακτική.

Η Έρευνα Δράσης έχει ως βάση τις παρακάτω παραδοχές της θεωρίας του *αναστοχαστικού ορθολογισμού*:

1. Τα πολύπλοκα πρακτικά προβλήματα απαιτούν εξειδικευμένες λύσεις.
2. Οι λύσεις αυτές μπορούν να αναπτυχθούν μόνο μέσα στο περιβάλλον στο οποίο αναδύεται το πρόβλημα και όπου κρίσιμο και καθοριστικό στοιχείο είναι ο χρήστης.
3. Οι λύσεις δεν μπορούν να εφαρμοστούν με επιτυχία σε άλλες καταστάσεις, αλλά μπορούν να τεθούν στη διάθεση άλλων χρηστών ως υποθέσεις προς έλεγχο.

Οι παραδοχές αυτές αποτελούν τη βάση της Έρευνας Δράσης, (Altrichter, Posch, & Somekh 2001).

Εν κατακλείδι, η Έρευνα – Δράσης στοχεύει να υπηρετήσει τόσο τις πρακτικές ανησυχίες των ανθρώπων που βιώνουν μια προβληματική κατάσταση, όσο και τους σκοπούς των κοινωνικών επιστημών, μέσα σε ένα αμοιβαία αποδεκτό ηθικό – δεοντολογικό πλαίσιο.

3.2.2. Τα βασικά χαρακτηριστικά της Έρευνας Δράσης

Τα βασικά χαρακτηριστικά της έρευνας δράσης είναι τα εξής:

1. Η Έρευνα Δράσης διεξάγεται από ανθρώπους που σχετίζονται άμεσα με την κοινωνική κατάσταση που ερευνάται.
2. Η Έρευνα Δράσης ξεκινά από πρακτικά ζητήματα που προκύπτουν από την καθημερινή χρήση ενός συστήματος και στοχεύει, τόσο στην βελτίωση της πρακτικής κατάστασης, όσο και στην ανάπτυξη της γνώσης σχετικά με τις πρακτικές των συμμετεχόντων. Δηλαδή, τα ερευνητικά ερωτήματα αναδύονται από αλληλεπίδραση των χρηστών με το σύστημα και όχι από γενικούς προβληματισμούς.
3. Η Έρευνα Δράσης πρέπει να συμβατή με τις εκπαιδευτικές αξίες του σχολείου και με τις συνθήκες εργασίας των εκπαιδευτικών και συμβάλλει στην περαιτέρω ανάπτυξη των αξιών και στη βελτίωση των συνθηκών.
4. Η Έρευνα Δράσης οδηγεί σε βελτίωση της κατανόησης από την πλευρά του ερευνητή κοινωνικών καταστάσεων, δομών και συμπεριφορών και προσανατολίζει τη δράση τόσο του ίδιου του όσο και των συμμετεχόντων. (Altrichter, Posch, & Somekh 2001)

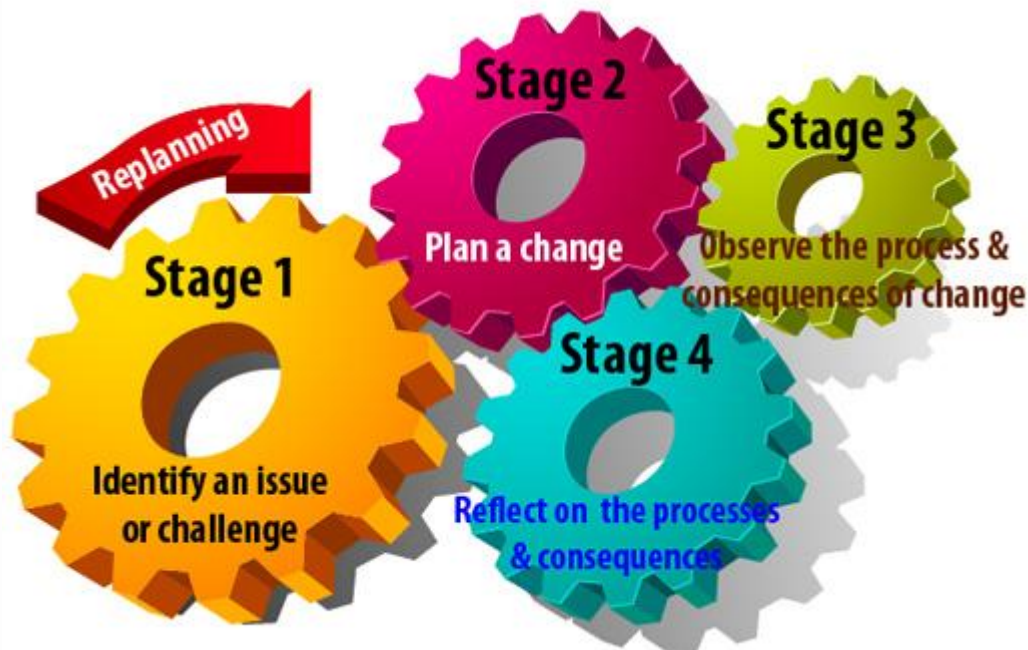
5. Η Έρευνα Δράσης είναι δυνατόν να οδηγήσει σε αποτελέσματα που θα έχουν ευρύτερη επίδραση. (Ιωσηφίδης 2003).

3.2.3. Μεθοδολογία

Η Έρευνα Δράσης ανήκει στις συμμετοχικές μεθοδολογίες έρευνας, οι οποίες χαρακτηρίζονται από κριτική διάθεση, πραγματιστικό πνεύμα, αποσκοπούν στην άμεση λύση προβλημάτων και στοχεύουν στην αλλαγή, περιορισμένη ή ευρύτερη. Το κοινό τους χαρακτηριστικό είναι η σχετικοποίηση της διάκρισης ερευνητή και ερευνούμενων.

Δεν υπάρχει καμιά αυστηρή διαδοχική διαδικασία στην Έρευνα Δράσης. Όλοι οι θεωρητικοί της Έρευνας Δράσης προτείνουν μοντέλα σπειροειδή ή κυκλικά, στα οποία ο σχεδιασμός, η δράση και ο αναστοχασμός επαναλαμβάνονται σε όλη την ερευνητική δραστηριότητα. (Hart E. and Bond M. 1995).

Η έρευνα δράσης δεν είναι μια γραμμική ερευνητική μεθοδολογία που ξεκινά με τη διατύπωση υποθέσεων, που ελέγχονται μέσα από τη διερευνητική διαδικασία, για να καταλήξει στην επιβεβαίωση ή τη διάψυσή τους. Δεν έχει δηλαδή μια δεδομένη αρχή και ένα δεδομένο τέλος. Η έρευνα δράσης αναπτύσσεται μέσα από τη σπείρα *αυτο-στοχασμού*: έναν έλικα από κύκλους *σχεδιασμού, δράσης* (υλοποίησης του σχεδιασμού), *συστηματικής παρατήρησης, στοχασμού...* και μετά επανα-σχεδιασμού, περαιτέρω υλοποίησης, παρατήρησης και στοχασμού... Αποτελεί ουσιαστικά μια αλυσίδα επάλληλων κύκλων χωρίς τέλος. Η έρευνα δράσης δηλαδή είναι μια ανοικτή κυκλική διαδικασία, κατά την οποία κάθε κύκλος οδηγεί στον επόμενο, που θα σημάνει επαναπροσδιορισμό του σχεδίου και νέα στρατηγική δράσης, επιπρόσθετη κριτική και αυτοκριτική παρέμβαση και αναστοχασμό σε ένα πλαίσιο διαρκούς σύνθεσης και ανασύνθεσης. Ενδεικτικό της ανοικτής αυτής κυκλικής διαδικασίας είναι η σπείρα κύκλων σχεδιασμού, δράσης, παρατήρησης και αναστοχασμού, όπως φαίνεται στο σχήμα:



Σχήμα 2.6.1 Κυκλική διαδικασία έρευνας δράσης

(http://www3.hku.hk/literacy/index.php?option=com_content&view=section&layout=blog&id=6&Itemid=6)

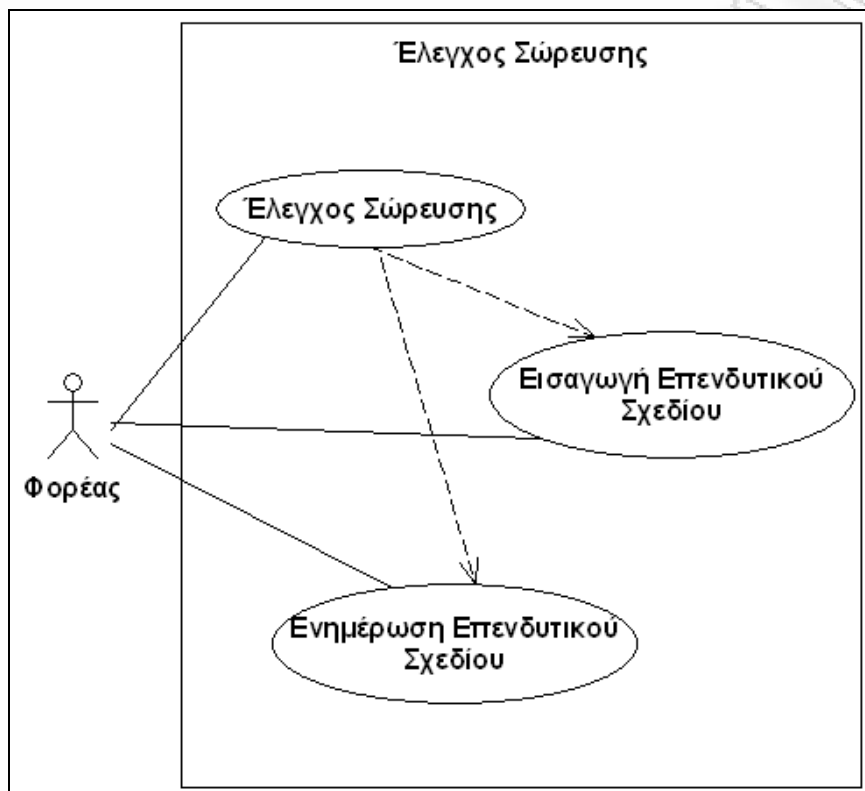
Η κυκλική αυτή διαδικασία χρησιμοποιήθηκε και στην παρούσα εργασία. Αρχικά έγιναν συναντήσεις και επικοινωνίες μεταξύ των εμπλεκόμενων φορέων οι οποίοι προσπάθησαν να εστιάσουν και να προσδιορίσουν ακριβώς το πρόβλημα. Έπειτα σε συνδυασμό με τους προγραμματιστές ολοκληρώθηκε ο σχεδιασμός και η υλοποίηση του συστήματος το οποίο πέρασε από μια σειρά δοκιμών και ελέγχων μέσα από τις οποίες προέκυψαν διορθώσεις και βελτιώσεις του συστήματος. Έτσι το σύστημα απέκτησε την τελική του μορφή, η οποία ωστόσο μπορεί να αλλάξει και πάλι, εφόσον ανακαλυφθούν νέες βελτιώσεις από τους χρήστες.

3.2.4. Πότε χρησιμοποιείται η Έρευνα Δράσης;

Η Έρευνα Δράσης χρησιμοποιείται σε πραγματικές καταστάσεις, παρά στις σχεδιασμένες, πειραματικές μελέτες, δεδομένου ότι η αρχική εστίασή της είναι στην επίλυση των πραγματικών προβλημάτων. Μπορεί, εντούτοις, να χρησιμοποιηθεί από τους κοινωνικούς επιστήμονες για την προκαταρκτική ή πειραματική έρευνα, ειδικά όταν η κατάσταση είναι πάρα πολύ διαφορετική για να πλαισιώσει μια ακριβή ερευνητική ερώτηση. Συνήθως, επιλέγεται όταν οι περιστάσεις απαιτούν ευελιξία, συμμετοχή των ανθρώπων στην έρευνα, ή όταν μια αλλαγή πρέπει να πραγματοποιηθεί γρήγορα (O'Brien, R. 2001).

Κεφάλαιο 4. Ανάλυση και Σχεδίαση

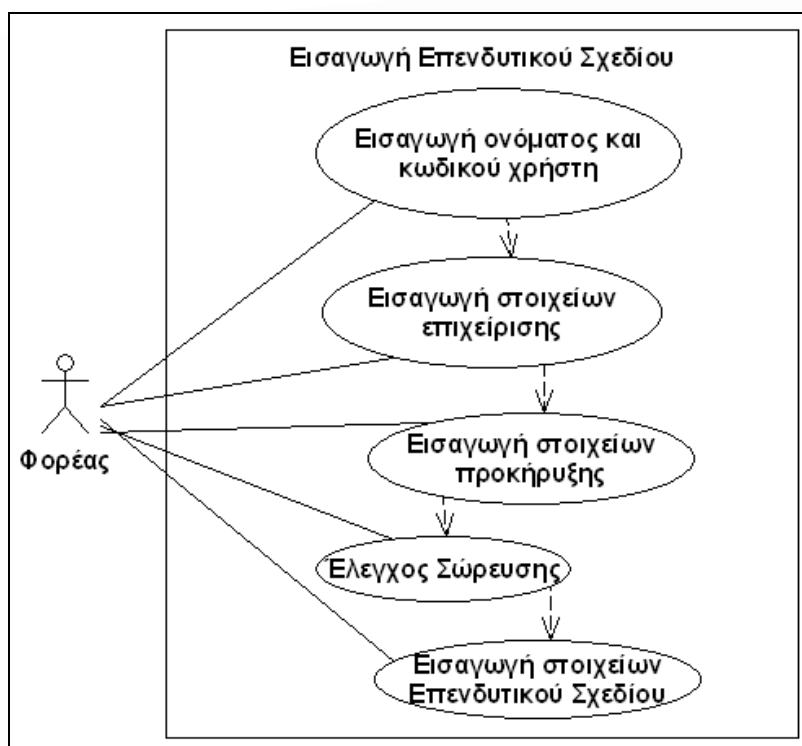
Το σύστημα ελέγχου σώρευσης παρέχει τη δυνατότητα εισαγωγής και ενημέρωσης των επενδυτικών σχεδίων καθώς και τον έλεγχο σώρευσης μιας επιχείρησης όπως φαίνεται και στο γενικό use case diagram που ακολουθεί.



Σχήμα 4.1. Γενικό Use Case Diagram

4.1. Εισαγωγή επενδυτικού σχεδίου

Η διαδικασία εισαγωγής ενός επενδυτικού σχεδίου αποτελείται από συγκεκριμένα βήματα. Όπως φαίνεται και στο case diagram που ακολουθεί αρχικά ο φορέας που παρέχει την κρατική ενίσχυση, αφού συνδεθεί στο σύστημα, θα πρέπει να καταχωρήσει τα στοιχεία της επιχείρησης που δέχεται την ενίσχυση αλλά και τα στοιχεία του επενδυτικού σχεδίου.



Σχήμα 4.2. Εισαγωγή Επενδυτικού Σχεδίου - Use Case Diagram

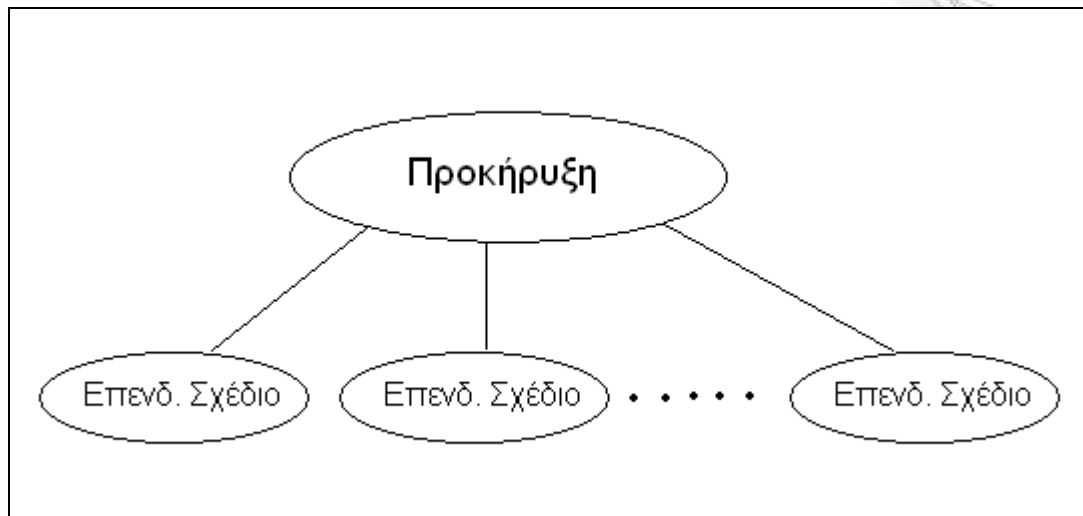
Για την εισαγωγή της επιχείρησης θα πρέπει να συμπληρωθεί στην κατάλληλη οθόνη μια σειρά από πεδία όπως είναι το ΑΦΜ και η επωνυμία της επιχείρησης. Αναλυτικά τα στοιχεία που πρέπει να συμπληρωθούν φαίνονται στην παρακάτω εικόνα.

Προσθήκη Νέας Επιχείρησης			
Επωνυμία Επιχείρησης	<input type="text"/>	Διακριτικός Τίτλος	<input type="text"/>
Κωδικός Κύριας Δραστηριότητας	<input type="text"/>	Τομέας	ΜΕΤΑΠΟΙΗΤΙΚΩΝ - ΜΕΤΑΛΛΕΥΤ <input type="button" value="v"/>
ΑΦΜ	<input type="text"/>	ΔΟΥ	<input type="text"/>
Χώρα Εδρας	EL <input type="button" value="m"/>		
Περιφέρεια Εδρας	<input type="text"/>	Νομός Εδρας	<input type="text"/>
Δήμος Εδρας	<input type="text"/>	Δημοτικό Διαμέρισμα Εδρας	<input type="text"/>
Αριθμός Απασχολουμένων στην Επιχείρηση	<input type="text"/>	Υφιστάμενη	ΝΑΙ <input type="button" value="v"/>
Τηλέφωνο	<input type="text"/>	Fax	<input type="text"/>
Οδός	<input type="text"/>	Αριθμός	<input type="text"/>
Ταχυδρομικός Κώδικας	<input type="text"/>	Ιστοτόπος Εταιρείας	<input type="text"/>
Επώνυμο Εκπροσώπου Εταιρείας	<input type="text"/>	Όνομα Εκπροσώπου Εταιρείας	<input type="text"/>
Θέση Εκπροσώπου στην Εταιρεία	<input type="text"/>	ΑΦΜ Εκπροσώπου	<input type="text"/>
Αριθμός Δελτίου Ταυτότητας	<input type="text"/>	Κινητό Εκπροσώπου	<input type="text"/>
Νομική Μορφή	A.E <input type="button" value="v"/>	Είδος Βιβλίων	A' Κατηγορίας <input type="button" value="v"/>
Email	<input type="text"/>		

Προσθήκη Επιχείρησης Ακύρωση

Εικόνα 4.3. Προσθήκη νέας επιχείρησης

Ένα επενδυτικό σχέδιο θα πρέπει να συνδέεται με μία Πρόσκληση – Προκήρυξη. Η σχέση τους είναι ένα προς πολλά. Δηλαδή κάτω από μια προκήρυξη μπορεί να «κρέμονται» περισσότερα από ένα επενδυτικά σχέδια, όπως φαίνεται και στο παρακάτω σχήμα.



Σχήμα 4.4. Σχέση Προκήρυξης – Επενδυτικού σχεδίου

Οπότε, για να γίνει εισαγωγή ενός επενδυτικού σχεδίου θα πρέπει να έχει αρχικά καταχωρηθεί η προκήρυξη με την οποία συνδέεται. Αν έχει ήδη γίνει η εισαγωγή της προκήρυξης ο φορέας μπορεί να προχωρήσει με την εισαγωγή του επενδυτικού σχεδίου.

Κατά την εισαγωγή μιας προκήρυξης θα πρέπει να εισαχθούν στο σύστημα τα πλήρη στοιχεία της τα οποία φαίνονται στην παρακάτω εικόνα.

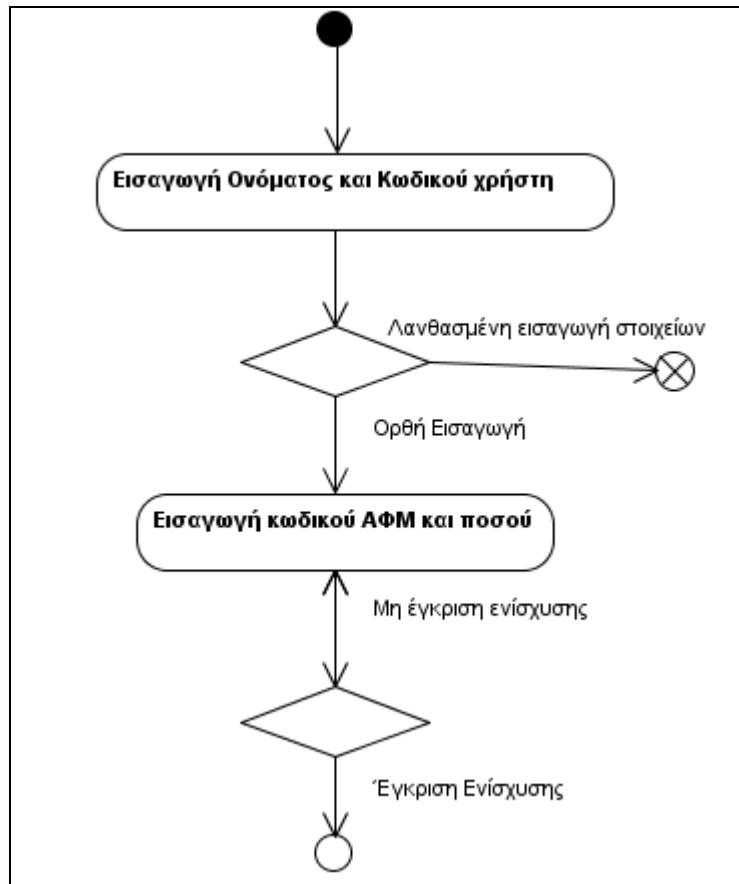
Προκήρυξη			
Επιχειρησιακή Περίοδος	<input type="text"/>	Επιχειρησιακό Πρόγραμμα	<input type="text"/>
Κωδικός Αξονα	<input type="text"/>	Κωδικός Φορέα Προκήρυξης	1010400
Κωδικός Προκήρυξης	<input type="text"/>	Είδος Αξιολόγησης	ΑΜΕΣΗ
Εναρξη Υποβολής	<input type="text"/>	Λήξη Υποβολής	<input type="text"/>
Περίοδος Υποβολής	<input type="text"/>	Περίοδος Αξιολόγησης	<input type="text"/>
Εναρξη Επιλεξιμότητας	<input type="text"/>	Λήξη Επιλεξιμότητας	<input type="text"/>
Ποσό	<input type="text"/>	Τίτλος	<input type="text"/>
Αριθμός Πρωτοκόλλου	<input type="text"/>	Ημερομηνία Πρωτοκόλλου	<input type="text"/>
Ημερομηνία Ολοκλήρωσης	<input type="text"/>	Καθεστώς	Καν.(ΕΚ) 800/2008 - Ζ
Αριθμός Κοινοποίησης	<input type="text"/>	Αριθμός Απόφασης	<input type="text"/>

+ Προσθήκη Προκήρυξης
 🏠 Ακύρωση

Εικόνα 4.5. Προσθήκη Προκήρυξης

Αφού ολοκληρωθεί η εισαγωγή της προκήρυξης μπορεί να γίνει εισαγωγή του επενδυτικού σχεδίου (αναλυτικά τα βήματα αναφέρονται στο κεφάλαιο 8).

Πριν την καταχώρηση του επενδυτικού σχεδίου θα πρέπει γίνει έλεγχος σώρευσης της επιχείρησης για την οποία θέλουμε να καταχωρηθεί το επενδυτικό σχέδιο.



Εικόνα 4.6. Activity Diagram - Έλεγχος Σώρευσης

Ο έλεγχος σώρευσης γίνεται από την αρχική σελίδα όπως φαίνεται και στην παρακάτω οθόνη. Ο φορέας θα πρέπει να καταχωρήσει το ΑΦΜ της επιχείρησης που θέλει να χρηματοδοτήσει και το ποσό του επενδυτικού σχεδίου.

Επενδυτικά Σχέδια				
Τίτλος	Απ. Προυπ/σμός	Ημερομηνία Έγκρισης	Εισχ. Προυπ/σμός	Επιχορήγηση


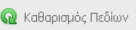
Κορυφή σελίδας

Copyright © 2011 Υπουργείο Οικονομίας, Ανταγωνιστικότητας & Ναυτιλίας
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης - Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης

Εικόνα 4.7. Έλεγχος Σώρευσης

Ανάλογα με τις χρηματοδοτήσεις που έχει λάβει η επιχείρηση στο διάστημα των τριών ετών το σύστημα θα επιστρέψει αναφορά ανάλογα με τον αν η επιχείρηση σωρεύει ή όχι.

Ελεγχος Σώρευσης	
ΑΦΜ Επιχείρησης	109953771
Αιτούμενο Ποσό	150,000.00






Αναφορά Ελέγχου Σώρευσης: Για το Α.Φ.Μ (109953771) που δηλώσατε και για το ποσό (150000) μπορείτε να εγκρίνετε το επενδυτικό σχέδιο

Εικόνα 4.8. Αποτέλεσμα Ελέγχου Σώρευσης

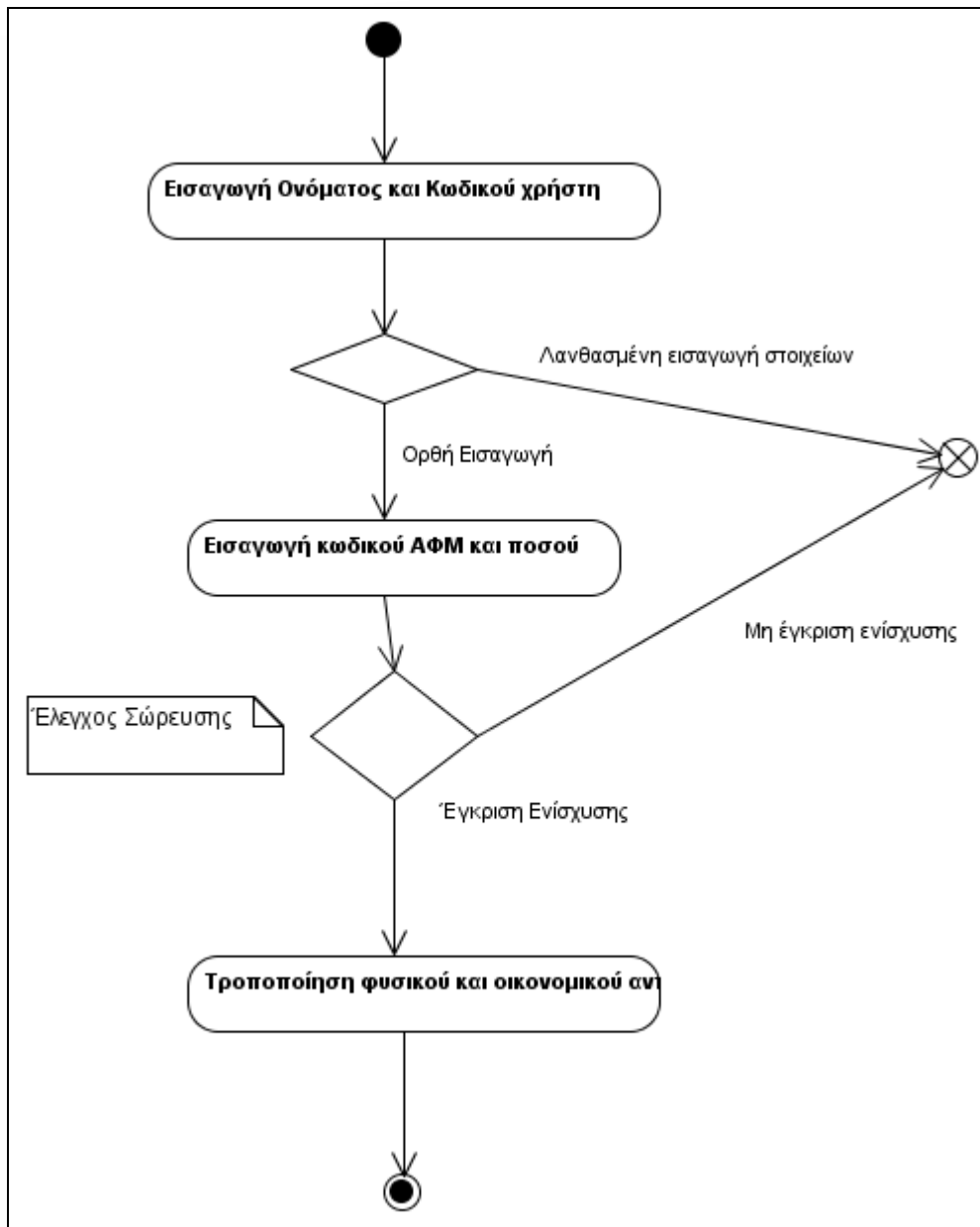
Εφόσον η αναφορά σώρευσης είναι θετική μπορούμε να προχωρήσουμε στην καταχώρηση του επενδυτικού σχεδίου όπου και εδώ αντίστοιχα θα πρέπει να γίνει συμπλήρωση μιας σειράς από πεδία τα οποία φαίνονται στην επόμενη εικόνα.

Επενδυτικό Σχέδιο			
Τίτλος Επενδυτικού Σχεδίου	<input type="text"/>	MIS	<input type="text"/>
Αιτούμενος Προυπολογισμός	<input type="text"/>	Επιχορηγούμενος Προυπολογισμός	<input type="text"/>
Ενισχυόμενος Προυπολογισμός	<input type="text"/>	Δημόσια Επιχορήγηση	<input type="text"/>
Ημερομηνία Εγκρισης	<input type="text" value="31"/>	Ημερομηνία Ολοκλήρωσης	<input type="text" value="31"/>
Οδός	<input type="text"/>	Αριθμός	<input type="text"/>
Ταχυδρομικός κώδικας	<input type="text"/>		
Α.Φ.Μ Επιχείρησης	<input type="text"/>	Κωδικός Επένδυσης	<input type="text"/>
Τεχνική Περιγραφή			
<input style="height: 40px;" type="text"/>			

 Προσθήκη Επενδυτικού Σχεδίου
  Ακύρωση

Εικόνα 4.9. Προσθήκη Επενδυτικού Σχεδίου

Η ενημέρωση ενός Επενδυτικού σχεδίου μπορεί να γίνει μόνο εφόσον η επιχείρηση περνάει τον έλεγχο σώρευσης με επιτυχία.



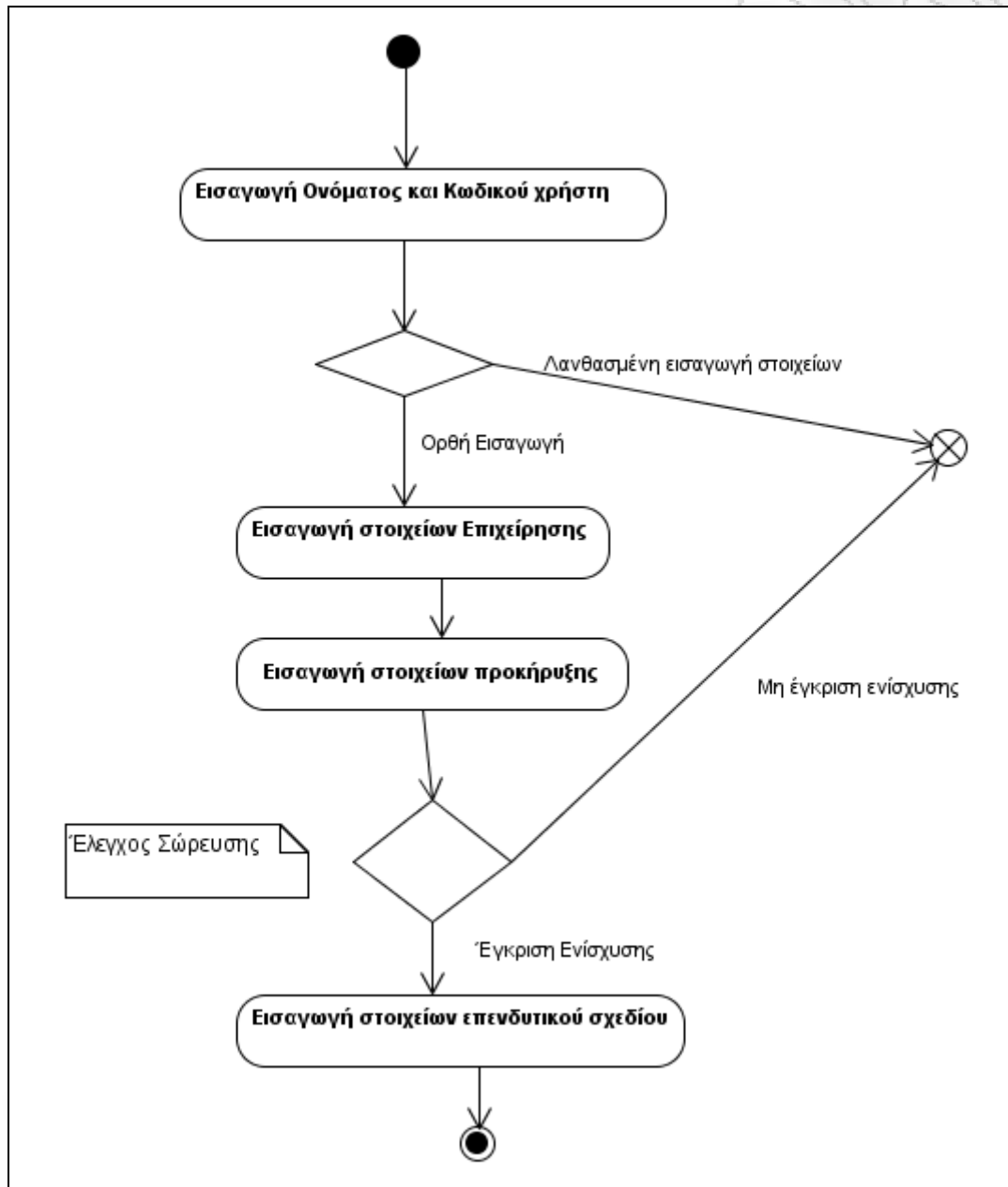
Εικόνα 4.10. Activity Diagram - Τροποποίηση Επενδυτικού Σχεδίου

Αντίστοιχα από τη πλευρά των διεπαφών υπάρχει η δυνατότητα εισαγωγής επενδυτικού σχεδίου και όλης της πληροφορίας που απαιτείται πριν την εισαγωγή του επενδυτικού σχεδίου και η δυνατότητα ελέγχου σώρευσης μιας επιχείρησης (βλέπε αντίστοιχα xml στο Παράρτημα 10.3 Request CheckAccumulation.xml και 10.4 Request InsertIPlan.xml).

Πιο αναλυτικά κατά την εισαγωγή του επενδυτικού σχεδίου:

1. Γίνεται έλεγχος για το αν έχει ήδη καταχωρηθεί η επιχείρηση και αν όχι γίνεται η εισαγωγή της
2. Γίνεται έλεγχος για το αν υπάρχει ήδη στο σύστημα η αντίστοιχη προκήρυξη και αν όχι γίνεται η εισαγωγή της

3. Εκτελείται έλεγχος σώρευσης για την επιχείρηση με το ποσό του επενδυτικού σχεδίου
4. Αν ο έλεγχος σώρευσης είναι θετικός καταχωρείται το επενδυτικό σχέδιο.
5. Η διεπαφή επιστρέφει το συνολικό ποσό με το οποίο έχει ενισχυθεί η επιχείρηση μέσα στη τελευταία τριετία, το κωδικό της σώρευσης που έτρεξε, το κωδικό σφάλματος αν υπάρχει και ένα επεξηγηματικό μήνυμα.



Εικόνα 4.11. Activity Diagram - Εισαγωγή Επενδυτικού Σχεδίου

Από τα παραπάνω συμπεραίνουμε ότι τα xml αρχεία που στέλνονται για την καταχώρηση επενδυτικού σχεδίου θα πρέπει να περιέχουν πλήρη στοιχεία επιχείρησης, προκήρυξης και επενδυτικού σχεδίου. Αν ο έλεγχος σώρευσης στο 4^ο βήμα είναι αρνητικός δεν καταχωρείται κανένα στοιχείο.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <insertIPlanResponse xmlns="http://parameters.services.accws.mou/">
      <iplanresponse>
        <poso>22030.75</poso>
        <accumulationId>61</accumulationId>
      </iplanresponse>
      <errorCode>0</errorCode>
      <errorDescription>Το επενδυτικό σχέδιο εισήχθει επιτυχώς.</errorDescription>
    </insertIPlanResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Εικόνα 4.12. Response Διεπαφής Εισαγωγής Επενδυτικού Σχεδίου

Όσον αφορά την διεπαφή ελέγχου σώρευσης δέχεται το ΑΦΜ της επιχείρησης και το ποσό της επένδυσης και επιστρέφει το συνολικό ποσό με το οποίο έχει ενισχυθεί η επιχείρηση μέσα στη τελευταία τριετία, το κωδικό σφάλματος αν υπάρχει και ένα επεξηγηματικό μήνυμα για το αν μπορεί να ενισχυθεί περαιτέρω η επιχείρηση με το ποσό που αιτηθήκαμε.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <checkAccumulationResponse xmlns="http://parameters.services.accws.mou/">
      <balance>22030.75</balance>
      <errorCode>0</errorCode>
      <errorDescription>Για το Α.Φ.Μ (999467638) που δηλώσατε και για το ποσό (10000) μπορείτε να εγκρίνετε το επενδυτικό σχέδιο</errorDescription>
    </checkAccumulationResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

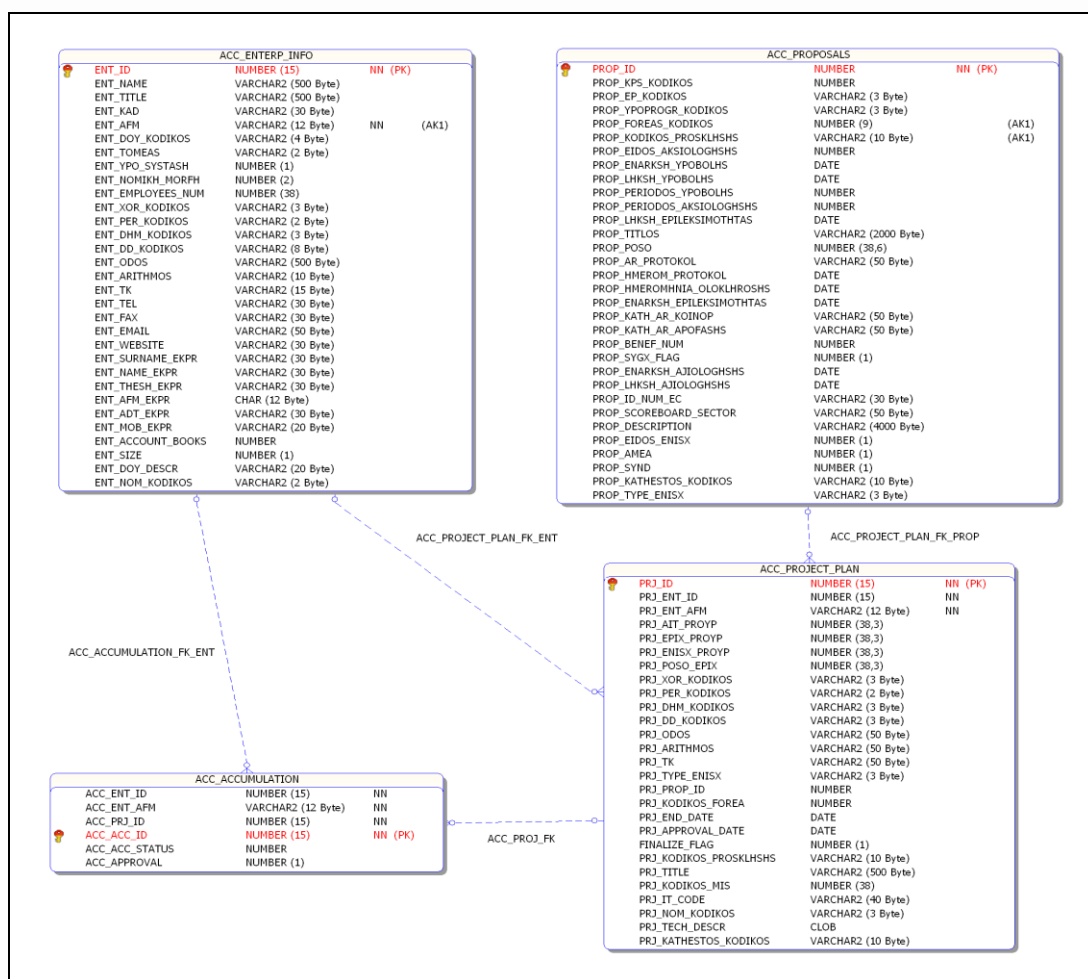
Εικόνα 4.13. Response Διεπαφής Ελέγχου Σώρευσης

4.2. Δομή βάσης δεδομένων

Το σχήμα της βάσης αποτελείται από πέντε βασικούς πίνακες. Οι πίνακες αυτοί είναι οι εξής:

- ACC_ENTERP_INFO: Σε αυτό τον πίνακα αποθηκεύονται τα στοιχεία των επιχειρήσεων.
- ACC_PROPOSALS: Σε αυτό τον πίνακα αποθηκεύονται τα στοιχεία των προσκλήσεων.
- ACC_PROJECT_PLAN: Σε αυτό τον πίνακα αποθηκεύονται τα στοιχεία των επενδυτικών σχεδίων.
- ACC_ACCUMULATION: Σε αυτό τον πίνακα αποθηκεύονται τα στοιχεία των ελέγχων σώρευσης που εκτελούνται.

Η σχέση αυτών των πινάκων φαίνεται στο παρακάτω σχήμα.



Σύστημα Σώρευσης	Βασικό Διάγραμμα Οντοτήτων - Συσχετίσεων	04/2011	V 1.0
Λύση ΜΟΣ		© Χαϊντίνη Ασημίνα	

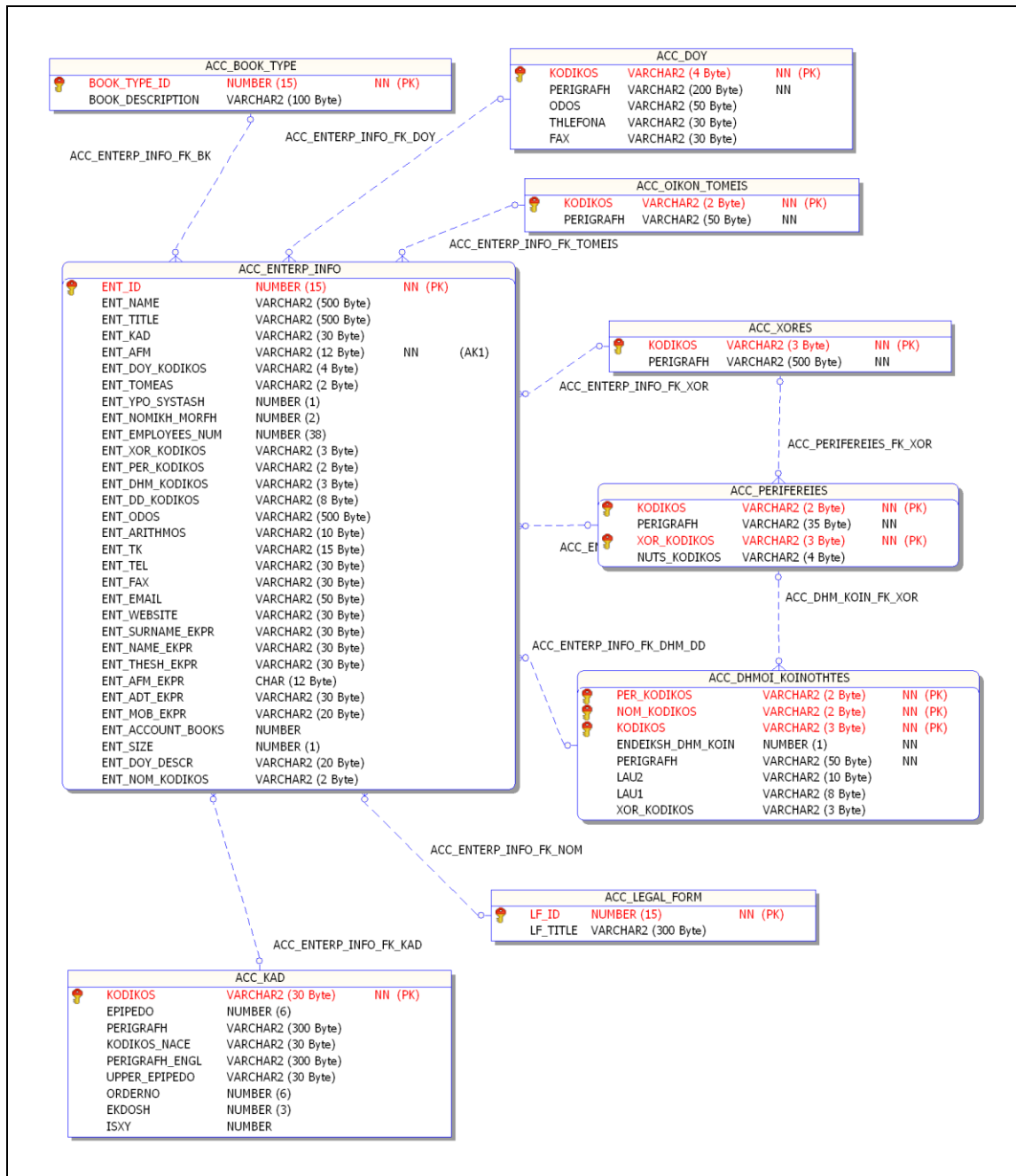
Αναλυτικά τα στοιχεία του πίνακα ACC_ENTERP_INFO φαίνονται στον πίνακα που ακολουθεί.

Τίτλος Πεδίου	Table Column	Τύπος Πεδίου	Μέγεθος Πεδίου	Παρατηρήσεις
AA	ENT_ID	Αριθμός	15	Κλειδί πίνακα (PK)
Επωνυμία Επιχείρησης	ENT_NAME	Κείμενο	200	
Διακριτικός Τίτλος	ENT_TITLE	Κείμενο	200	
Κωδικός κύριας Δραστηριότητας	ENT_KAD	Κείμενο	25	Παραμετρικός πίνακας ACC_KAD

Α.Φ.Μ.	ENT_AFM	Κείμενο	12	
Δ.Ο.Υ.	ENT_DOY_K ODIKOS	Κείμενο	4	Παραμετρικός πίνακας ACC_DOY
Τομέας Δραστηριότητας	ENT_TOMEA S	Κείμενο	2	Παραμετρικός πίνακας ACC_TOMEA S
Υπό σύσταση / Υφιστάμενη	ENT_YPO_SY STASH	Αριθμός	1	(ΝΑΙ/ΟΧΙ)
Νομική Μορφή	ENT_NOMIK H_MORFH	Αριθμός	2	Παραμετρικός πίνακας ACC_LEGAL_ FORM
Αριθμός απασχολ/νων στην Επιχείρηση	ENT_EMPLO YYES_NUM	Αριθμός	38	Σε EME
Χώρα	ENT_XOR_K ODIKOS	Κείμενο	3	Παραμετρικός πίνακας acc_xores
Περιφέρεια Έδρας	ENT_PER_KO DIKOS	Κείμενο	2	Παραμετρικός πίνακας ACC_PERIFER EIES
Δήμος Έδρας	ENT_DHM_K ODIKOS	Κείμενο	3	Παραμετρικός πίνακας ACC_DHMOI_ KOINOTHTES
Δημοτικό Διαμέρισμα Έδρας	ENT_DD_KO DIKOS	Κείμενο	3	Παραμετρικός πίνακας ACC_DHMOI_ KOINOTHTES
Οδός	ENT_ODOS	Κείμενο	50	
Αριθμός	ENT_ARITHM OS	Κείμενο	10	
TK	ENT_TK	Κείμενο	10	
Τηλέφωνο	ENT_TEL	Κείμενο	30	
FAX	ENT_FAX	Κείμενο	30	
email	ENT_EMAIL	Κείμενο	30	
website	ENT_WEBSIT E	Κείμενο	30	
Επωνυμία	ENT_SURNA	Κείμενο	30	

Εκπροσώπου	ME_EKPR			
Όνομα Εκπροσώπου	ENT_NAME_EKPR	Κείμενο	30	
Θέση στην επιχείρηση	ENT_THESH_EKPR	Κείμενο	30	
ΑΦΜ Εκπροσώπου	ENT_AFM_EKPR	Κείμενο	12	
Έγγραφο ταυτοποίησης Εκπροσώπου	ENT_ADT_EKPR	Κείμενο	30	
Τηλέφωνο Εκπροσώπου	ENT_MOB_EKPR	Κείμενο	20	
Είδος Βιβλίων	ENT_ACCOUNT_BOOKS	Αριθμός		Α,Β,Γ, Μη τήρηση
Μέγεθος Επιχείρησης	ENT_SIZE	Αριθμός	1	
Περιγραφή ΔΥΟ	ENT_DOY_DESCR	Κείμενο	20	Για τους χρήστες που δεν ξέρουν τον κωδικό
Νομός	ENT_NOM_KODIKOS	Κείμενο	2	Παραμετρικός πίνακας ACC_NOMOI

Η σχέση του πίνακα ACC_ENTERP_INFO με τους παραμετρικούς πίνακες φαίνεται στο σχήμα που ακολουθεί.



Σύστημα Σάρρευσης	Σύνδεση πίνακα ACC_ENTERP_INFO με τους παραμετρικούς πίνακες	04/2011	V 1.0
Λύση ΜΟΣ		© Χαϊντίνη Ασημίνα	

Τα στοιχεία του πίνακα ACC_PROPOSALS φαίνονται στον πίνακα που ακολουθεί.

Τίτλος Πεδίου	Table column	Τύπος Πεδίου	Μέγεθος Πεδίου	Παρατηρήσεις
ΑΑ	PRJ_ID	Αριθμός		Κλειδί πίνακα (PK)
Προγραμματιστική Περίοδος	PROP_KPS_KODIKOS	Αριθμός		Παραμετρικός πίνακας ACC_YPOPROGRAMMATA
Επιχειρησιακό Πρόγραμμα	PROP_EP_KODIKOS	Κείμενο	3	Παραμετρικός πίνακας ACC_YPOPROGRAMMATA
Άξονας Προτεραιότητας	PROP_YPOPROGRAM_R_KODIKOS	Κείμενο	3	Παραμετρικός πίνακας ACC_YPOPROGRAMMATA
Κωδικός Φορέα Προκήρυξης	PROP_FOREAS_KODIKOS	Αριθμός	9	Παραμετρικός πίνακας ACC_UM_FOREIS
Κωδικός Προκήρυξης	PROP_KODIKOS_PROSKLHSHS	Κείμενο	10	
Είδος Αξιολόγησης	PROP_EIDOS_AKSIOLOGHSHS	Αριθμός		
Έναρξη Υποβολής	PROP_ENARKSH_YPOBOLHS	Ημερομηνία		
Λήξη Υποβολής	PROP_LHKSH_YPOBOLHS	Ημερομηνία		
Περίοδος Υποβολής	PROP_PERIODOS_YPOBOLHS	Αριθμός		
Περίοδος Αξιολόγησης	PROP_PERIODOS_AKSIOLOGHSHS	Αριθμός		
Λήξη Επιλεξιμότητας	PROP_LHKSH_EPILEKSIMOTHAS	Ημερομηνία		
Τίτλος Προκήρυξης	PROP_TITLOS	Κείμενο	2000	
Ποσό Προκήρυξης	PROP_POSO	Αριθμός	38,6	
Αριθ.	PROP_AR_PROT	Κείμενο	50	

Πρωτοκόλλου	OKOL			
Ημερομ. Πρωτοκόλλου	PROP_HMEROM_PROTOKOL	Ημερομηνία		
Ημερομ. Ολοκλήρωσης	PROP_HMEROM_HNIA_OLOKLHROS	Ημερομηνία		
Έναρξη Επιλεξιμότητας	PROP_ENARKSH_EPILEKSIMOTHAS	Ημερομηνία		
Αριθμός Κοινοποίησης	PROP_KATH_AR_KOINOP	Κείμενο	50	
Αριθμός Απόφασης Έγκρισης ΕΕ	PROP_KATH_AR_APOFASHS	Κείμενο	50	
Έναρξη Αξιολόγησης	PROP_ENARKSH_AJIOLOGHSHS	Ημερομηνία		
Λήξη Αξιολόγησης	PROP_LHKSH_AJIOLOGHSHS	Ημερομηνία		
Περιγραφή	PROP_DESCRIPTOR	Κείμενο	4000	
Είδος Ενίσχυσης	PROP_EIDOS_ENISX	Αριθμός	1	
Καθεστώς	PROP_KATHESTOS_KODIKOS	Κείμενο	10	Παραμετρικός πίνακας ACC_KATHESTOTA
Τύπος Ενίσχυσης	PROP_TYPE_ENISX	Κείμενο	3	Παραμετρικός πίνακας ACC_TYPER_ENISX

Η σχέση του πίνακα ACC_PROPOSALS με τους παραμετρικούς πίνακες φαίνεται στο σχήμα που ακολουθεί.



Σύστημα Σώρευσης

Λύση ΜΟΣ

Σύνδεση πίνακα ACC_PROPOSALS
με τους παραμετρικούς πίνακες

04/2011

V 1.0

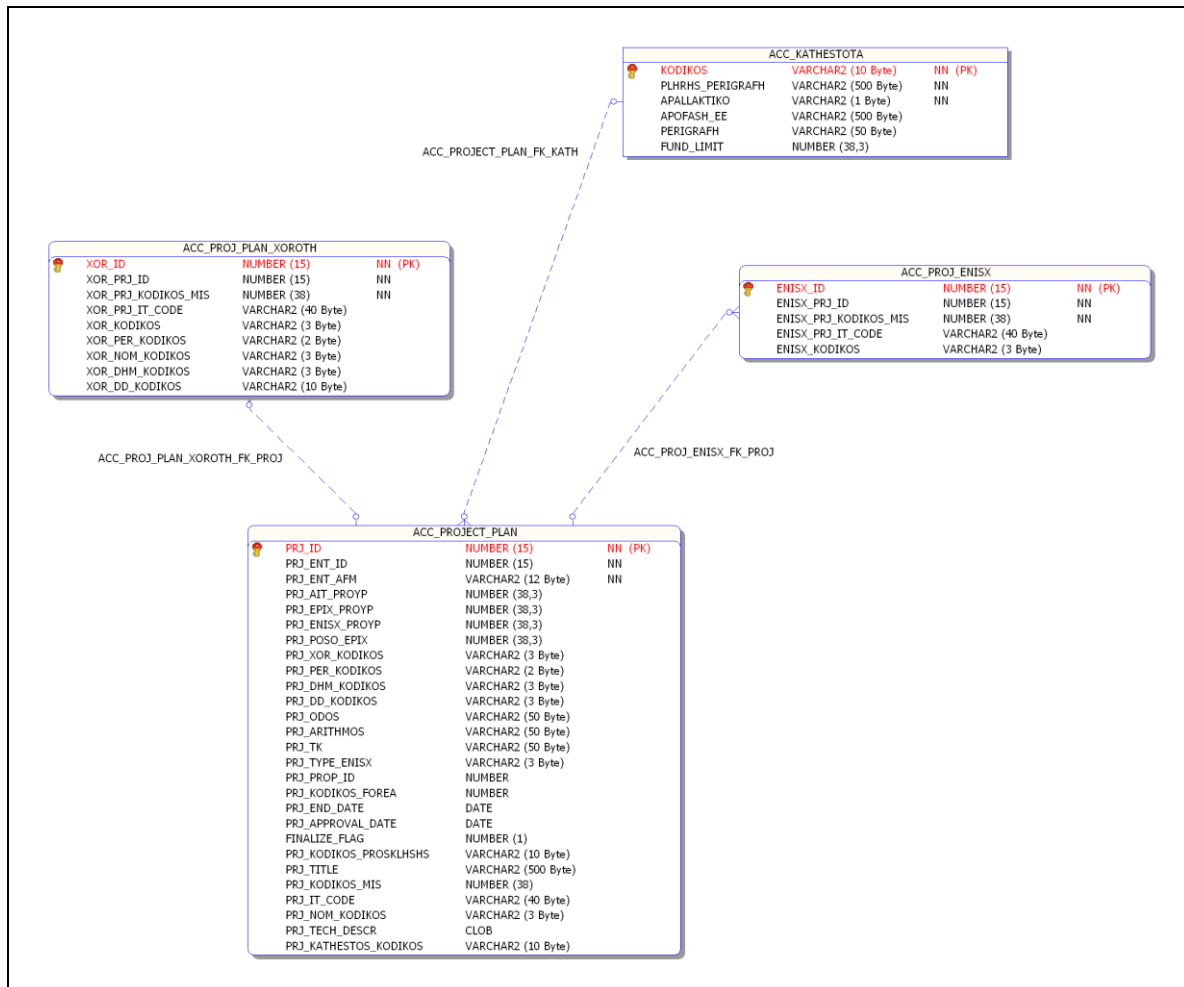
© Χαϊντίνη Ασημίνα

Τα στοιχεία του πίνακα ACC_PROJECT_PLAN φαίνονται στον πίνακα που ακολουθεί.

Τίτλος Πεδίου	Table column	Τύπος Πεδίου	Μέγεθος Πεδίου	Παρατηρήσεις
ΑΑ	PRJ_ID	Αριθμός	15	
ΑΑ Επιχείρησης	PRJ_ENT_ID	Αριθμός	15	Εξωτερικό κλειδί με τον πίνακα ACC_ENTERP_INFO (FK)
ΑΦΜ Επιχείρησης	PRJ_ENT_AFM	Κείμενο	20	Εξωτερικό κλειδί με τον πίνακα ACC_ENTERP_INFO (FK)
Αιτούμενος Προϋπολογισμός	PRJ_AIT_PROYYP	Αριθμός	38,3	
Επιχορηγούμενος Προϋπολογισμός	PRJ_EPIX_PROYYP	Αριθμός	38,3	
Ενισχυόμενος Προϋπολογισμός	PRJ_ENISX_PROYYP	Αριθμός	38,3	
Δημόσια Επιχορήγηση	PRJ_POSO_EPIX	Αριθμός	38,3	
Χώρα Τόπου εγκατάστασης	PRJ_XOR_KODIKOS	Κείμενο	3	Παραμετρικός πίνακας ACC_XORES
Περιφέρεια Τόπου εγκατάστασης	PRJ_PER_KODIKOS	Κείμενο	2	Παραμετρικός πίνακας ACC_PERIFERIES
Νομός Τόπου εγκατάστασης	PRJ_NOM_KODIKOS			Παραμετρικός πίνακας ACC_NOMOI
Δήμος Τόπου εγκατάστασης	PRJ_DHM_KODIKOS	Κείμενο	3	Παραμετρικός πίνακας ACC_DHMOI_KOINOTHTES
Δημ. Διαμέρισμα Τόπου εγκατάστασης	PRJ_DD_KODIKOS	Κείμενο	3	Παραμετρικός πίνακας ACC_DHMOI_KOINOTHTES
Οδός	PRJ_ODOS	Κείμενο	50	
Αριθμός	PRJ_ARITHMOS	Κείμενο	50	

TK	PRJ_TK	Κείμενο	50	
Σύντομη Τεχνική Περιγραφή	PRJ_TECH_DESCR	Κείμενο	4000	
ΑΑ Προκήρυξης	PRJ_PROP_ID	Αριθμός		Εξωτερικό κλειδί με τον πίνακα ACC_PROPOSALS (FK)
Κωδικός Φορέα Προκήρυξης	PRJ_KODIKOS_FOR EA	Αριθμός		Εξωτερικό κλειδί με τον πίνακα ACC_ENTERP_INFO (FK)
Ημερομηνία Λήξης	PRJ_END_DATE	Ημερομηνία		
Ημερομηνία Έγκρισης	PRJ_APPROVAL_DATE	Ημερομηνία		
Ένδειξη Ολοκλήρωσης	FINALIZE_FLAG	Αριθμός	1	
Καθεστώς	PRJ_KATHESTOS_KODIKOS	Αριθμός		Παραμετρικός πίνακας ACC_KATHES TOTA
Κωδικός Προκήρυξης	PRJ_KODIKOS_PRO SKLHSHS	Κείμενο	10	
Τίτλος Επενδυτικού Σχεδίου	PRJ_TITLE	Κείμενο	500	
Κωδικός Πράξης ΜΙΣ	PRJ_KODIKOS_MIS	Αριθμός	38	
Κωδικός Έργου	PRJ_IT_CODE	Κείμενο	40	

Η σχέση του πίνακα ACC_PROJECT_PLAN με τους παραμετρικούς πίνακες φαίνεται στο σχήμα που ακολουθεί.

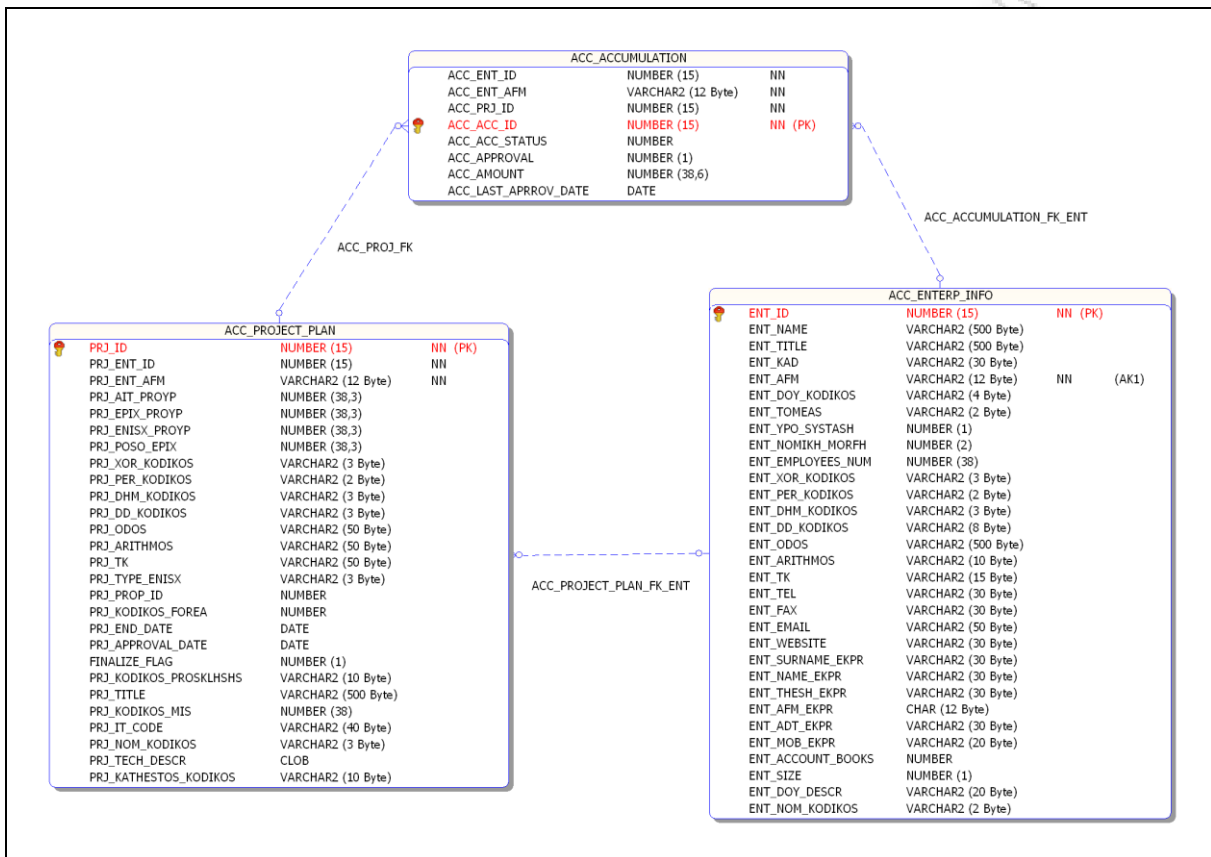


Σύστημα Σώρευσης	Σύνδεση πίνακα ACC_PROJECT_PLAN με τους παραμετρικούς πίνακες	04/2011	V 1.0
Λύση ΜΟΣ		© Χαϊντίνη Ασημίνα	

Τα στοιχεία του πίνακα ACC_ACCUMULATION φαίνονται στον πίνακα που ακολουθεί.

Τίτλος Πεδίου	Table column	Τύπος Πεδίου	Μέγεθος Πεδίου	Παρατηρήσεις
ΑΑ Επιχείρησης	ACC_ENT_ID	Αριθμός	15	Εξωτερικό κλειδί με τον πίνακα ACC_ENTERP_INFO (FK)
ΑΦΜ Επιχείρησης	ACC_ENT_AFM	Κείμενο	20	Εξωτερικό κλειδί με τον πίνακα ACC_ENTERP_INFO (FK)
ΑΑ Επενδυτικού Σχεδίου	ACC_PRJ_ID	Αριθμός	15	Εξωτερικό κλειδί με τον πίνακα ACC_PROJECT_PLANS (FK)
Κωδικός Σώρευσης	ACC_ACC_ID	Αριθμός	15	
Κωδικός Φορέα	ACC_ENT_AFM	Αριθμός	20	Παραμετρικός πίνακας ACC_UM_FOREIS
Έγκριση	ACC_APPROVAL	Αριθμός	1	ΝΑΙ/ΟΧΙ
Σωρευμένο Ποσό	ACC_AMOUNT	Αριθμός	38,6	
Είδος Ενίσχυσης	ACC_ACC_STAT US	Κείμενο	3	Ad hoc ενίσχυση ή μέτρο ενίσχυσης ή και τα δύο
Ημερομηνία λήξης επιλεξιμότητας των επενδυτικών δαπανών	ACC_LAST_APPROV_DATE	Ημερομηνία		

Η σχέση του πίνακα ACC_ACCUMULATION με τους παραμετρικούς πίνακες φαίνεται στο σχήμα που ακολουθεί.



Σύστημα Σόρευσης	Σύνδεση πίνακα ACC_ACCUMULATION με τους παραμετρικούς πίνακες	04/2011	V 1.0
Λύση ΜΟΣ		© Χαϊντίνη Ασημίνα	

Κεφάλαιο 5. Εργαλεία Υλοποίησης

Στις επόμενες ενότητες θα γίνει μια εκτενής αναφορά των εργαλείων που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος. Στις πιο σύνθετες εφαρμογές, είναι πολύ πιθανό να προκύψουν προβλήματα στη σχεδίαση και οργάνωση της λειτουργικότητας, καθώς και δυσκολίες συντήρησης και περιορισμοί στη δυνατότητα αναβάθμισης. Για το λόγο αυτό, δόθηκε ιδιαίτερο βάρος στη δημιουργία και χρήση προτύπων σχεδίασης. Έχοντας λοιπόν σαν βάση το Spring Framework χρησιμοποιήθηκε το πρότυπο σχεδίασης MVC Model και η δυνατότητα που μας δίνει για Dependency Injection.

5.1. Spring

Η Spring είναι ένα ελεύθερο (open source) περιβάλλον εργασίας για εφαρμογές Java που δημιουργήθηκε από τον Rob Johnson και περιγράφηκε στο βιβλίο του “Expert One-on-One: J2EE Design and Development”. Αναπτύχθηκε κατά κύριο λόγο, για να αντιμετωπίσει την πολυπλοκότητα ανάπτυξης enterprise εφαρμογών. Η Spring κάνει εφικτή την χρήση απλών JavaBeans για την επίτευξη πραγμάτων που προηγουμένως μπορούσαν να γίνουν μόνο μέσω EJBs. Παρόλα αυτά, η Spring δεν είναι μόνο χρήσιμη για την ανάπτυξη server-side εφαρμογών. Η χρήση της μπορεί να βοηθήσει στην απλοποίηση του κώδικα, τον ευκολότερο και πιο αποτελεσματικό έλεγχο και τη χαλαρή διασύνδεση (loose coupling) κάθε Java εφαρμογής.

Η Spring κάνει πολλά πράγματα, αλλά, αν την αναλύσει κάποιος στα βασικά της κομμάτια, θα δει ότι είναι ένας ελαφρύς (lightweight) aspect-oriented container ο οποίος εφαρμόζει dependency injection, ενώ παράλληλα αποτελεί και περιβάλλον εργασίας (framework). Για να γίνουν τα πράγματα πιο κατανοητά θα γίνεται ανάλυση της περιγραφής αυτής :

- **Lightweight (Ελαφρύς):** Με τον όρο αυτό αναφερόμαστε τόσο στο μέγεθος όσο και στον επιπλέον φόρτο (overhead). Η διανομή του Spring Framework μπορεί να συμπεριληφθεί σε ένα απλό jar αρχείο μεγέθους 2,5 MB, ενώ ο επιπρόσθετος φόρτος εργασίας που απαιτείται είναι αμελητέος. Ο προγραμματιστής χρειάζεται να κάνει ελάχιστες, αν όχι καθόλου, αλλαγές στον κώδικα της εφαρμογής που αναπτύσσει έτσι, ώστε να επωφεληθεί από τον πυρήνα της, ενώ μπορεί οποιαδήποτε στιγμή να σταματήσει να τη χρησιμοποιεί. Βέβαια αυτού του είδους η ευχέρεια παρέχεται μόνο στον πυρήνα της Spring. Πολλά επιπλέον μέρη της, όπως είναι η πρόσβαση δεδομένων, απαιτούν πολύ στενότερη «διασύνδεση» (coupling) από ότι το Spring framework. Παρόλα αυτά, το κέρδος στις περισσότερες από αυτές τις περιπτώσεις είναι πολύ σημαντικό.
- **Dependency Injection:** Η Spring προάγει τη χαλαρή διασύνδεση χρησιμοποιώντας μία τεχνική που είναι γνωστή ως dependency injection (DI). Όταν εφαρμόζεται η DI, τα αντικείμενα λαμβάνουν παθητικά τις εξαρτήσεις τους (dependencies) αντί να τις δημιουργούν ή να τις αναζητούν μόνα τους. Είναι κάτι σαν ένα αντίστροφο Java Naming and Directory Interface (JNDI) αντί ένα αντικείμενο να ψάχνει μόνο του για τις εξαρτήσεις του σε έναν container, ο container δίνει τις εξαρτήσεις στο αντικείμενο χωρίς να περιμένει πρώτα να ερωτηθεί.
- **Aspect – Oriented:** Η Spring παρέχει πλούσια υποστήριξη σε Aspect - Oriented Programming (AOP). Αυτό έχει σαν αποτέλεσμα τη δημιουργία

εφαρμογών με περισσότερη συνοχή ακόμη και αν είναι απαραίτητη η συνύπαρξη διαφορετικών λογικών λειτουργίας. Έτσι, κάθε αντικείμενο της εφαρμογής κάνει μόνο ότι το αφορά και δεν είναι υπεύθυνο (ενδεχομένως δε γνωρίζει καν) για πράγματα που έχουν να κάνουν με άλλα συστήματα. Τέτοιου είδους πράγματα μπορεί να είναι η υποστήριξη συνδιαλλαγών ή η καταγραφή των διαφορών κινήσεων (logging).

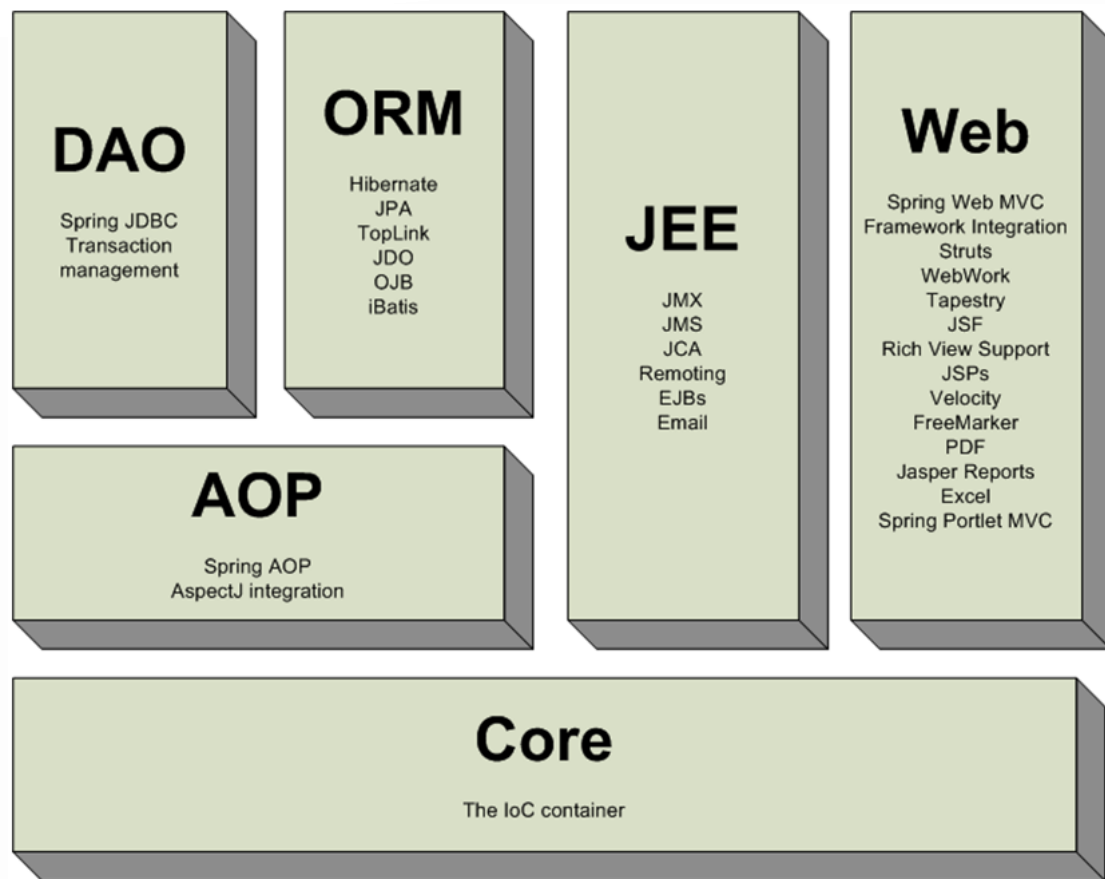
- **Container:** Η Spring αποτελεί έναν container με την έννοια ότι κρατάει και διαχειρίζεται τον κύκλο ζωής και τη διαμόρφωση των αντικειμένων της εφαρμογής. Ο προγραμματιστής μπορεί να ορίσει πώς θέλει να διαμορφώνονται και να δημιουργούνται τα αντικείμενα, καθώς και ποιες θέλει να είναι οι σχέσεις μεταξύ τους.
- **Framework (Περιβάλλον Εργασίας):** Η Spring δίνει τη δυνατότητα να δημιουργηθούν πολύπλοκες εφαρμογές με το συνδυασμό άλλων, λιγότερο πολύπλοκων, κομματιών. Στη Spring τα αντικείμενα δημιουργούνται μέσω δηλώσεων σε απλά XML αρχεία. Επίσης παρέχει πολλές δομικές λειτουργίες, όπως η διαχείριση συναλλαγών κ.ά., επιτρέποντας στον προγραμματιστή να ασχοληθεί περισσότερο με τη λογική της εφαρμογής του.

Έτσι, μπορούμε να πούμε ότι η Spring είναι ένα περιβάλλον εργασίας το οποίο βοηθάει τον προγραμματιστή να δημιουργήσει εφαρμογές με χαλαρά διασυνδεδεμένο κώδικα. Ακόμη και αν αυτό ήταν το μόνο που έκανε, τα οφέλη θα ήταν πάρα πολλά από άποψη συντηρησιμότητας και ελεγχιμότητας και θα άξιζε τον κόπο να γίνει χρήση της. Όμως η Spring είναι πολλά περισσότερα. Περιέχει κομμάτια που χρησιμοποιούν DI και AOP και έτσι συνθέτουν μία αρκετά αξιόλογη πλατφόρμα στην οποία μπορούν να αναπτυχθούν εφαρμογές.

5.1.1. Τα module της Spring

Το περιβάλλον εργασίας της Spring αποτελείται από αρκετά, καλά ορισμένα, κομμάτια (modules). Τα κομμάτια αυτά σαν σύνολο δίνουν στον προγραμματιστή ότι χρειάζεται, για να αναπτύξει enterprise εφαρμογές. Δεν απαιτείται από τον προγραμματιστή να βασίσει όλη την εφαρμογή του πάνω στη Spring. Αντίθετα, του παρέχεται η δυνατότητα να επιλέξει όποια από τα modules πιστεύει ότι καλύπτουν τις ανάγκες του. Επιπλέον παρέχονται τρόποι σύνδεσης με άλλα περιβάλλοντα εργασίας και βιβλιοθήκες έτσι, ώστε ο προγραμματιστής να μη χρειαστεί να τα αναπτύξει από την αρχή.

Όπως φαίνεται στο σχήμα 5.1.1 όλα τα modules της Spring είναι χτισμένα πάνω από το βασικό της container - πυρήνα. Ο πυρήνας καθορίζει πώς δημιουργούνται τα beans, πώς αρχικοποιούνται και πώς γίνεται ο χειρισμός τους, κάτι που αποτελεί τις κυριότερες λειτουργίες της Spring. Όμως ο προγραμματιστής ενδιαφέρεται περισσότερο για άλλα modules, αυτά που χρησιμοποιούν τον πυρήνα και τις υπηρεσίες που αυτός παρέχει.



Σχήμα 5.1.1 Τα modules της Spring

(Πηγή: <http://static.springsource.org/spring/docs/2.0.x/reference/introduction.html>)

- Ο πυρήνας - core container

Ο πυρήνας της Spring παρέχει όλη τη βασική λειτουργικότητα του Spring Framework. Το module αυτό περιέχει το BeanFactory, το οποίο είναι ο θεμελιώδης container και η βάση με την οποία η Spring υλοποιεί το DI.

- Το DAO module

Η δουλειά με JDBC συχνά περιλαμβάνει τη δημιουργία αρκετού κώδικα, ο οποίος κάνει βασικές και τετριμμένες ενέργειες όπως σύνδεση με τη βάση, δημιουργία κάποιας εντολής (statement), επεξεργασία του αποτελέσματος και κλείσιμο της σύνδεσης. Μέσω του Data Access Object (DAO) module η Spring δίνει τη δυνατότητα απλοποίησης όλης αυτής της διαδικασίας περιορίζοντας τα προβλήματα που προκύπτουν. Επιπλέον, χτίζει ένα layer από exceptions πάνω από τα μηνύματα σφάλματος που παράγουν αρκετοί εξυπηρετητές βάσεων δεδομένων.

Επιπρόσθετα, αυτό το module (χρησιμοποιώντας το AOP module) παρέχει υπηρεσίες διαχείρισης συναλλαγών για τα αντικείμενα των εφαρμογών που χρησιμοποιούν τη Spring.

- Το ORM module

Η Spring παρέχει το ORM module object - relational mapping (ORM) (αντιστοίχιση σχέσης - αντικείμενου) αντί για την απευθείας χρήση του JDBC. Αυτό χτίζει πάνω στο DAO παρέχοντας έναν αρκετά ολικό τρόπο για ανάπτυξη DAO πάνω από αρκετές λύσεις που προσφέρουν object - relational mapping. Παρόλο που η Spring δεν παρέχει το δικό της ORM, υποστηρίζει αρκετά δημοφιλή περιβάλλοντα

εργασίας, συμπεριλαμβανομένων των Hibernate, Java Persistence API, Java Data Objects και iBatis SQL Maps. Το σύστημα διαχείρισης συναλλαγών της Spring υποστηρίζει εκτός από αυτά τα περιβάλλοντα εργασίας και το JDBC.

➤ Το AOP module

Η Spring προσφέρει υποστήριξη για aspect - oriented programming στο AOP module της. Όπως και το DI, το AOP υποστηρίζει τη χαλαρή διασύνδεση μεταξύ των αντικειμένων κάθε εφαρμογής, όμως με το AOP σημαντικά θέματα που αφορούν το σύνολο των εφαρμογών (όπως οι συναλλαγές και η ασφάλεια) χωρίζονται από τα αντικείμενα στα οποία αυτά εφαρμόζονται.

➤ Το Web module

Η Spring παρέχει μία πλούσια συλλογή κλάσεων για τη δημιουργία εφαρμογών που στηρίζονται στο διαδίκτυο (web - based applications) μέσω του Web module της. Υποστηρίζεται η χρήση του προτύπου σχεδίασης εφαρμογών Model/View/Controller (MVC), ο χειρισμός ηλεκτρονικής αλληλογραφίας (mail support) καθώς και η χρήση απομακρυσμένων υπηρεσιών (remoting support).

Όσον αφορά τη χρήση του προτύπου σχεδίασης εφαρμογών Model/View/Controller (MVC), η Spring υποστηρίζει πλήρως το πιο γνωστό MVC περιβάλλον εργασίας, το Apache Struts. Με αυτόν τον τρόπο δίνεται η δυνατότητα στον προγραμματιστή να χρησιμοποιήσει στις ήδη σχεδιασμένες κλάσεις του τις αρχές του dependency injection που προσφέρει η Spring. Βέβαια, πέραν της υποστήριξης του Apache Struts, η Spring προσφέρει και τη δική της υλοποίηση του MVC. Μέσω αυτού μπορεί να γίνει χρήση μίας ευρείας γκάμας τεχνολογιών παρουσίασης, από σελίδες JSP και Apache Jakarta Velocity μέχρι Microsoft Excel και Adobe PDF.

Σε σχέση με την αποστολή μηνυμάτων ηλεκτρονικής αλληλογραφίας, η Spring παρέχει ένα αρκετά απλοποιημένο API, το οποίο ταιριάζει με την όλη φιλοσοφία του DI που αυτή προσφέρει. Για το σκοπό αυτό παρέχονται δύο υλοποιήσεις, μία του JavaMail και μία της MailMessage κλάσης από το πακέτο com.oreilly.servlet του Jason Hunter. Μέσω αυτών δίνεται η δυνατότητα δημιουργίας ενός πρότυπου μηνύματος το οποίο στη συνέχεια μπορεί να χρησιμοποιηθεί σαν βάση για την αποστολή αλληλογραφίας μέσω της εφαρμογής.

Τέλος, σε σχέση με χρήση απομακρυσμένων υπηρεσιών, παρέχεται εκτεταμένη υποστήριξη μίας μεγάλης συλλογής τεχνικών απομακρυσμένης πρόσβασης για τη γρήγορη δημιουργία και πρόσβαση σε απομακρυσμένες υπηρεσίες. Τέτοιες τεχνικές είναι το Java RMI, το JAXRPC, το Caucho Hessian και το Caucho Burlap. Εκτός αυτών, η Spring παρέχει και το δικό της πρωτόκολλο, που χρησιμοποιεί το HTTP πρωτόκολλο επικοινωνίας και το οποίο βασίζεται στο απλό Java serialization.

5.1.2. Java EE Connector API (JCA)

Αν και τα τελευταία χρόνια όλο και περισσότεροι προγραμματιστές τείνουν να χρησιμοποιούν τις λεγόμενες ελαφριές πλατφόρμες με τεχνολογίες όπως η Spring και ο Tomcat, υπάρχουν πολλές περιπτώσεις όπου η χρήση αρκετών παραδοσιακών J2EE APIs κρίνεται απαραίτητη. Η σύνδεση και επικοινωνία μεταξύ τέτοιου είδους εφαρμογών και εφαρμογών που χρησιμοποιούν άλλες τεχνολογίες μπορεί να είναι δύσκολη. Το JCA της Spring παρέχει ένα σταθερό και αξιόπιστο τρόπο για την επικοινωνία με μία σειρά από τέτοιου είδους enterprise συστήματα. Τα κυριότερα

APIs που υποστηρίζονται είναι το Java Naming and Directory Interface (JNDI), τα EJB καθώς και η Java Message Service (JMS).

5.1.3. Dependency Injection (DI)

Αν και η Spring έχει πολλές δυνατότητες, το DI αποτελεί την καρδιά του περιβάλλοντος εργασίας. Παρά το γεγονός ότι ο όρος αυτός ακούγεται σαν μια περίπλοκη προγραμματιστική τεχνική ή τρόπο σχεδιασμού (design pattern) στην πραγματικότητα δεν είναι τόσο πολύπλοκο όσο ακούγεται. Αντίθετα, η χρήση του κάνει τον κώδικα πολύ πιο απλό, ευκολονόητο και εύκολα συντηρήσιμο.

Κάθε μη τετριμμένη εφαρμογή αποτελείται από δύο ή περισσότερες κλάσεις οι οποίες συνεργάζονται, για να εκτελέσουν τις απαραίτητες ενέργειες. Παραδοσιακά κάθε αντικείμενο είναι υπεύθυνο να διατηρεί τις αναφορές του (reference) στα αντικείμενα με τα οποία 'συνεργάζεται' (δηλαδή τις εξαρτήσεις του — dependencies). Κάτι τέτοιο οδηγεί σε έναν ισχυρά δεμένο και δύσκολο στον έλεγχο κώδικα.

Όταν γίνεται χρήση του DI, τα αντικείμενα λαμβάνουν τις εξαρτήσεις τους κατά την ώρα της δημιουργίας τους από μία εξωτερική οντότητα η οποία συντονίζει κάθε αντικείμενο στο όλο σύστημα. Με άλλα λόγια, οι εξαρτήσεις (dependencies) δίνονται (injected) στα αντικείμενα. Έτσι, το DI αποτελεί ενός είδους αντιστροφής στην ευθύνη που αφορά τη διατήρηση των αναφορών των αντικειμένων. Η λειτουργικότητα αυτή παρέχεται από τον πυρήνα της Spring, ο οποίος είναι υπεύθυνος να δίνει στα αντικείμενα τις εξαρτήσεις τους. Ο πυρήνας γνωρίζει πώς πρέπει να δώσει αυτές τις εξαρτήσεις στα κατάλληλα αντικείμενα διαβάζοντας το XML αρχείο ρυθμίσεων της Spring.

Το κλειδί της παραπάνω τεχνικής είναι η χαλαρή διασύνδεση. Αν ένα αντικείμενο γνωρίζει για τις εξαρτήσεις του μόνο μέσω μίας διεπαφής (όχι μέσω της υλοποίησής τους ή μέσω του τρόπου που δημιουργούνται), η εξάρτηση μπορεί να αλλαχτεί πολύ εύκολα από μια διαφορετική υλοποίηση χωρίς το αντικείμενο να γνωρίζει για τη διαφορά.

5.1.4. Aspect – Oriented Programming

Όπως περιγράφηκε στην προηγούμενη παράγραφο, το DI καθιστά εφικτή τη χαλαρή διασύνδεση τμημάτων λογισμικού σε εφαρμογές. Αντίθετα, το aspect-oriented programming επιτρέπει τη συγκέντρωση λειτουργιών οι οποίες χρησιμοποιούνται σε όλη την εφαρμογή σε επαναχρησιμοποιήσιμα κομμάτια.

Το aspect-oriented programming συχνά ορίζεται ως μια τεχνική η οποία προάγει το διαχωρισμό ευθυνών μέσα στο σύστημα λογισμικού. Τα συστήματα αποτελούνται από πολλά ξεχωριστά κομμάτια το καθένα από τα οποία είναι υπεύθυνο για συγκεκριμένο κομμάτι λειτουργικότητας. Συχνά όμως αυτά τα κομμάτια έχουν επιπλέον ευθύνες πέρα από τις βασικές λειτουργίες για τις οποίες προορίζονται. Υπηρεσίες όπως η διατήρηση αρχείου, η διαχείριση συναλλαγών και η ασφάλεια συχνά υλοποιούνται μέσα σε κομμάτια λογισμικού των οποίων η βασική ευθύνη είναι τελείως διαφορετική.

Μοιράζοντας τέτοιου είδους ευθύνες κατά μήκος πολλαπλών κομματιών κώδικα δημιουργούνται δύο επίπεδα πολυπλοκότητας στον κώδικα που αναπτύσσεται. Ο

κώδικας ο οποίος υλοποιεί λειτουργίες που αφορούν όλη την εφαρμογή είναι διασκορπισμένος σε πολλά διαφορετικά τμήματα κώδικα αντί να είναι συγκεντρωμένος. Αυτό σημαίνει ότι, αν ο προγραμματιστής αποφασίσει να αλλάξει τον τρόπο που εκτελούνται τέτοιου είδους λειτουργίες, πρέπει να ανατρέξει σε πολλά διαφορετικά κομμάτια. Ακόμα και αν η διαφοροποίηση έγκειται μόνο στην αλλαγή του τρόπου κλήσης μίας και μόνο μεθόδου, η διόρθωση θα πρέπει να γίνει σε πολλά διαφορετικά κομμάτια κώδικα.

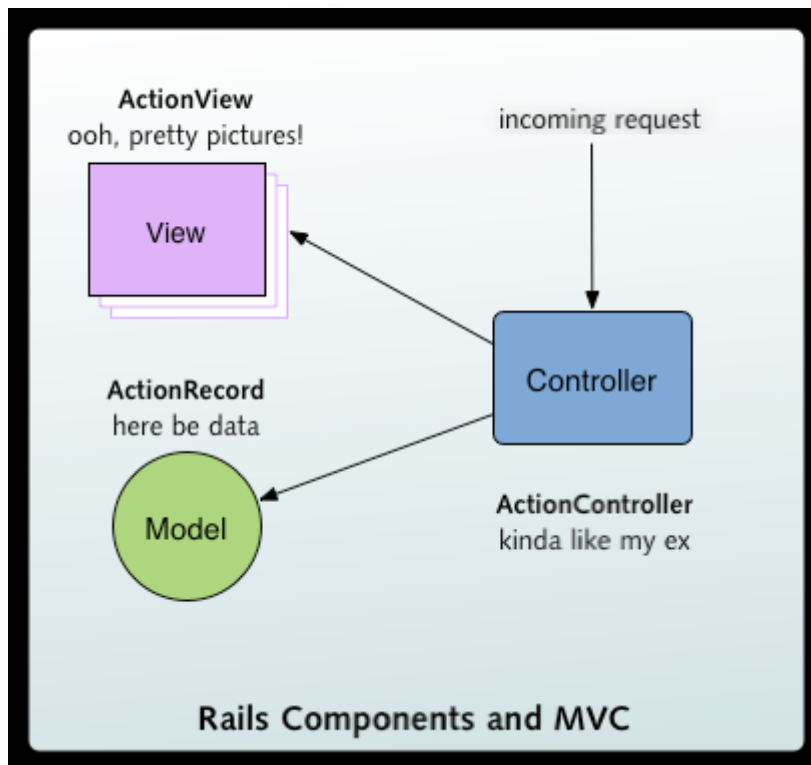
Τα διάφορα κομμάτια κώδικα είναι επιφορτισμένα με λειτουργίες οι οποίες δεν τα αφορούν προσθέτοντάς τους επιπλέον κώδικα. Για παράδειγμα, αν η λειτουργία μίας μεθόδου είναι να προσθέσει μία εγγραφή σε έναν τηλεφωνικό κατάλογο, δεν πρέπει να την αφορά το αν αυτή η διαδικασία γίνει με ασφάλεια.

Το AOP κάνει εφικτό το διαχωρισμό των διαφόρων υπηρεσιών επιτρέποντας να γίνεται χρήση τους, όποτε κάτι τέτοιο κρίνεται επιθυμητό, από τα κομμάτια κώδικα (components) που τις χρειάζονται. Αυτό οδηγεί στην ανάπτυξη πιο συνεκτικών κομματιών τα οποία είναι περισσότερο συγκεντρωμένα στο βασικό τους στόχο, αγνοώντας τελείως τις διάφορες άλλες υπηρεσίες του συστήματος. Με λίγα λόγια η χρήση των aspects διασφαλίζει ότι τα POJOs θα παραμείνουν απλά (plain).

5.1.5. Model – View - Controller

Μία από τις πιο συχνά εμφανιζόμενες μορφές προγραμμάτων είναι αυτά που ανάλογα με τις εντολές του χρήστη δημιουργούν, ενημερώνουν ή διαγράφουν δεδομένα που υπάρχουν σε κάποιο αρχείο ή βάση δεδομένων. Οι εφαρμογές αυτές εμφανίζονται σε πολλά διαφορετικά περιβάλλοντα, από εφαρμογές ιστού μέχρι εφαρμογές στη γραμμή εντολών. Το μοντέλο που χρησιμοποιείτε συνήθως για την ανάπτυξη τέτοιων εφαρμογών είναι το Model-View-Controller (MVC). Η λογική αυτού του προτύπου σχεδίασης όπως προκύπτει και από το όνομά του είναι ότι το πρόβλημα χωρίζεται σε τρία κύρια συστατικά:

- Το Μοντέλο του προβλήματος (model) που διαχειρίζεται τη συμπεριφορά και τα δεδομένα στο πεδίο της εφαρμογής. Το μοντέλο επιστρέφει πληροφορίες σχετικά με την κατάστασή του στις προβολές και αντιδρά σε αιτήσεις για αλλαγή στην κατάστασή του που προέρχονται από τον ελεγκτή.
- Οι Προβολές (views), οι οποίες είναι ο τρόπος με τον οποίο αναπαριστώνται τα δεδομένα. Αξίζει να σημειώσουμε ότι οι προβολές δεν είναι αναγκαίο να είναι μια διεπαφή προς τον χρήστη, αλλά μπορεί να είναι μια διεπαφή για άλλα εξωτερικά συστατικά. Αν και ο χρήστης δεν το αντιλαμβάνεται αυτό, η προβολή δεν γνωρίζει καμία λεπτομέρεια σχετικά με το επιχειρησιακό μοντέλο του συστήματος και ούτε με τον τρόπο υλοποίησης των υπηρεσιών που του προσφέρονται. Στην ιδανική περίπτωση το συνθέτημα της προβολής αρκείται στο να λαμβάνει πληροφορίες από το μοντέλο και να τις παρουσιάζει στον χρήστη.
- Ο Ελεγκτής (controller) είναι το συστατικό το οποίο ενθυλακώνει τη λογική της εφαρμογής και αναλαμβάνει το συντονισμό της. Επεξεργάζεται τις αιτήσεις που προέρχονται από τις προβολές και αφού εκτελέσει τις απαραίτητες ενέργειες στέλνει αιτήσεις προς το μοντέλο για να πραγματοποιήσει αλλαγές στην κατάστασή του.



5.1.2. Model – View – Controller

(Πηγή: http://artemis.cslab.ntua.gr/el_thesis/artemis.ntua.ece/DT2009-0195/DT2009-0195.pdf)

Όπως είναι προφανές ένα σύστημα σχεδιασμένο σύμφωνα με το μοντέλο MVC έχει συνθετήματα με ξεκάθαρες αρμοδιότητες. Η συντήρηση και η περαιτέρω ανάπτυξη της εφαρμογής είναι ευκολότερη. Δεδομένων μάλιστα της επεκτασιμότητας και ευελιξίας που είναι αναγκαίο να χαρακτηρίζουν την εφαρμογή, η επιλογή του μοντέλου MVC υπήρξε μονόδρομος. Όταν ένα σύστημα καλείται να αλλάξει τον τρόπο λειτουργίας του, ή την δομή δεδομένων του για να προσαρμοστεί σε άλλες εφαρμογές είναι εύλογη η ανάγκη σαφούς κατανομής των αρμοδιοτήτων των συνθετημάτων του. Ο προγραμματιστής μπορεί ανά πάσα στιγμή να γνωρίζει τι ακριβώς πρέπει να αλλάξει και σε ποιο σημείο θα βρει αυτό που αναζητά.

Οι βασικότερες λειτουργίες που χρειάζονται για το μεγαλύτερο μέρος των εφαρμογών που αναπτύσσονται με αυτό το σχεδιαστικό πρότυπο είναι η δημιουργία, ανάκτηση, ενημέρωση και διαγραφή των δεδομένων από κάποια πηγή δεδομένων (data source) η οποία μπορεί να είναι ένα σύστημα αρχείων, μια τοπική ή απομακρυσμένη βάση δεδομένων κλπ. Οι λειτουργίες αυτές είναι γνωστές και σαν CRUD από τα αρχικά των λειτουργιών Create-Retrieve-Update-Delete. Έχουν εμφανιστεί αρκετές διαφορετικές προσεγγίσεις που επιτρέπουν την εύκολη δημιουργία των αντικειμένων του μοντέλου μιας εφαρμογής, με πιο εύχρηστη αυτή του Ruby On Rails. Το RoR επιτρέπει στον προγραμματιστή να δημιουργήσει όλες τις κλάσεις του μοντέλου του και κάποιες προβολές που επιτρέπουν τις λειτουργίες CRUD με λίγες εντολές στη γραμμή εντολών και χρησιμοποιώντας σαν είσοδο τη βάση δεδομένων. Στην περίπτωση μας οι λειτουργίες αυτές χρησιμοποιούνται για δικτυακές εφαρμογές που αναπτύσσονται με τη γλώσσα προγραμματισμού Java.

Πιο αναλυτικά, οι απαιτήσεις που υπάρχουν για τις λειτουργίες μας είναι οι εξής:

- Δημιουργία κώδικα κλάσεων των αντικειμένων

- Δημιουργία αντικειμένου σε μια βάση δεδομένων
- Ανάκτηση του αντικειμένου βάσει διαφορετικών κριτηρίων
- Ενημέρωση ήδη υπάρχοντος αντικειμένου
- Διαγραφή αντικειμένου
- Συσχετίσεις ανάμεσα στα αντικείμενα
- Δημιουργία απλών zk σελίδων που θα επιτρέπουν σε ένα χρήστη τις παραπάνω λειτουργίες

Το μοντέλο μπορεί εύκολα να περιγραφεί χρησιμοποιώντας αντικείμενα java POJOs (POJOs Plain Old Java Objects). Κατά τη διάρκεια του σχεδιασμού της εφαρμογής αρκεί να εφαρμόσουμε στα αντικείμενα του μοντέλου μας το stereotype <<POJO>>. Για τα αντικείμενα του μοντέλου μας πρέπει να υπάρχουν και οι αντίστοιχες λειτουργίες CRUD.

Κατά τη διάρκεια του μετασχηματισμού δημιουργείτε μία κλάση για κάθε αντικείμενο του μοντέλου που επιτρέπει να πραγματοποιηθούν οι λειτουργίες CRUD καθώς και τα αντίστοιχα αρχεία ρυθμίσεων. Οι προβολές αποτελούνται από σελίδες zk. Μια πρότυπη μορφή τέτοιων σελίδων δημιουργείτε αυτόματα κατά το μετασχηματισμό. Σαν controller χρησιμοποιείτε ένα Java Servlet το οποίο αναλαμβάνει να δέχεται αιτήσεις από τις σελίδες zk και χρησιμοποιώντας τις κλάσεις που προσφέρουν τις υπηρεσίες να πραγματοποιεί τις αντίστοιχες αλλαγές στο μοντέλο.

5.2. Apache Axis

Ο Apache Axis είναι ένα πλαίσιο ανοιχτού λογισμικού, για διαδικτυακές υπηρεσίες, το οποίο βασίζεται στην γλώσσα xml. Αποτελείται από δύο υλοποιήσεις εξυπηρετητών SOAP (μία σε γλώσσα Java και μία σε C++), από διάφορα εργαλεία και από APIs που εξυπηρετούν την δημιουργία και την τοποθέτηση των διαδικτυακών υπηρεσιών. Χρησιμοποιώντας τον Apache Axis, οι προγραμματιστές είναι σε θέση να δημιουργήσουν δομημένες υπολογιστικές εφαρμογές, οι οποίες λειτουργούν σε οποιοδήποτε υπολογιστικό σύστημα.

Ο Apache Axis2 είναι ένας πυρήνας για διαδικτυακές υπηρεσίες. Πρόκειται για τον ευρύτερα χρησιμοποιούμενο Apache Axis1, ο οποίος σχεδιάστηκε και κατασκευάστηκε από την αρχή, με καινούργια χαρακτηριστικά. Ο Apache Axis2 παρέχει την δυνατότητα στον χρήστη του, να προσθέσει διεπαφές διαδικτυακών υπηρεσιών σε διαδικτυακές εφαρμογές αλλά μπορεί να λειτουργήσει και ως ένας ξεχωριστός εξυπηρετητής εφαρμογών (server application). Υλοποιήσεις του Axis2 είναι διαθέσιμες σε γλώσσα Java και C++. Αναπτύχθηκε υπό την αιγίδα του οργανισμού Apache Software Foundation.

Στην περίπτωση χρήσης της Java έκδοσης του Axis, υπάρχουν δύο τρόποι να εκτεθεί ο κώδικας Java ως διαδικτυακή υπηρεσία. Ο πρώτος τρόπος, που είναι και πιο εύκολος, είναι η χρήση των Axis JWS (Java Web Service) αρχείων. Ο δεύτερος τρόπος, ο οποίος θα χρησιμοποιηθεί στην παρούσα εργασία, περιλαμβάνει τοποθέτηση των διαδικτυακών υπηρεσιών με τον τυπικό πλέον, τρόπο. Η τυπική τοποθέτηση δίνει την πρόσθετη δυνατότητα στον χρήστη να καθορίζει τα τμήματα του κώδικα που επιθυμεί να θέσει ως διαδικτυακές υπηρεσίες.

Στην συνέχεια παρουσιάζονται συνοπτικά οι δύο τρόποι δημιουργίας, τοποθέτησης και προσπέλασης των διαδικτυακών υπηρεσιών.

➤ 1ος τρόπος

1. Δημιουργία διαδικτυακών υπηρεσιών με χρήση των JWS αρχείων

Τα JWS αρχεία είναι αρχεία πηγαίου κώδικα Java και περιέχουν τις κλάσεις Java που θα τεθούν ως Διαδικτυακές Υπηρεσίες. Η διαφορά τους με τα τυπικά αρχεία java είναι η επέκταση του αρχείου(.jws αντί για .java) καθώς και ότι τοποθετούνται ως πηγαίος κώδικας (.java) κι όχι ως μεταφρασμένα αρχεία κλάσεων (.class). Η δημιουργία λοιπόν περιλαμβάνει την δημιουργία του κατάλληλου JWS αρχείου.

2. Τοποθέτηση διαδικτυακών υπηρεσιών με χρήση των JWS αρχείων

Εφόσον τεθεί το Axis2 servlet στον εξυπηρετητή, η τοποθέτηση πλέον γίνεται με απλή αντιγραφή των αρχείων jws στον φάκελο του Axis στον εξυπηρετητή. Αυτό ισχύει σε περίπτωση που χρησιμοποιείται ο Apache Tomcat ως υποδοχέας. Σε περίπτωση που χρησιμοποιείται κάποιος άλλος, η τυπική δημιουργία αρχείου WAR είναι απαραίτητη.

3. Πρόσβαση στις διαδικτυακές υπηρεσίες που έχουν δημιουργηθεί με JWS αρχεία

Οι διαδικτυακές υπηρεσίες υλοποιημένες με JWS αρχεία, είναι προσβάσιμες μέσω της διαδικτυακής διεύθυνσης τους, π.χ. "<http://localhost:8080/axis/.../file.jws>". Αν χρησιμοποιείται διαφορετικός υποδοχέας από τον Apache Tomcat, η διαδικτυακή διεύθυνση μπορεί να είναι διαφορετική.

➤ 2ος τρόπος

1. Τυπική δημιουργία διαδικτυακών υπηρεσιών με χρήση Axis Archive αρχείων (.aar)

Αυτός ο τρόπος δημιουργίας απαιτεί την ύπαρξη ενός συγκεκριμένου περιγραφέα τοποθέτησης που ονομάζεται WSDD (Web Service Deployment Descriptor). Μπορεί να χρησιμοποιηθεί για να προσδιορίσει ποιές πηγές μπορούν να τεθούν ως διαδικτυακές υπηρεσίες.

Η δημιουργία της διαδικτυακής υπηρεσίας περιλαμβάνει την δημιουργία ενός Axis Archive αρχείου (.aar) το οποίο περιέχει το μεταφρασμένο αρχείο (.class) του αρχείου Java της Διαδικτυακής Υπηρεσίας, καθώς και ένα αρχείο XML.

2. Τυπική τοποθέτηση διαδικτυακών υπηρεσιών

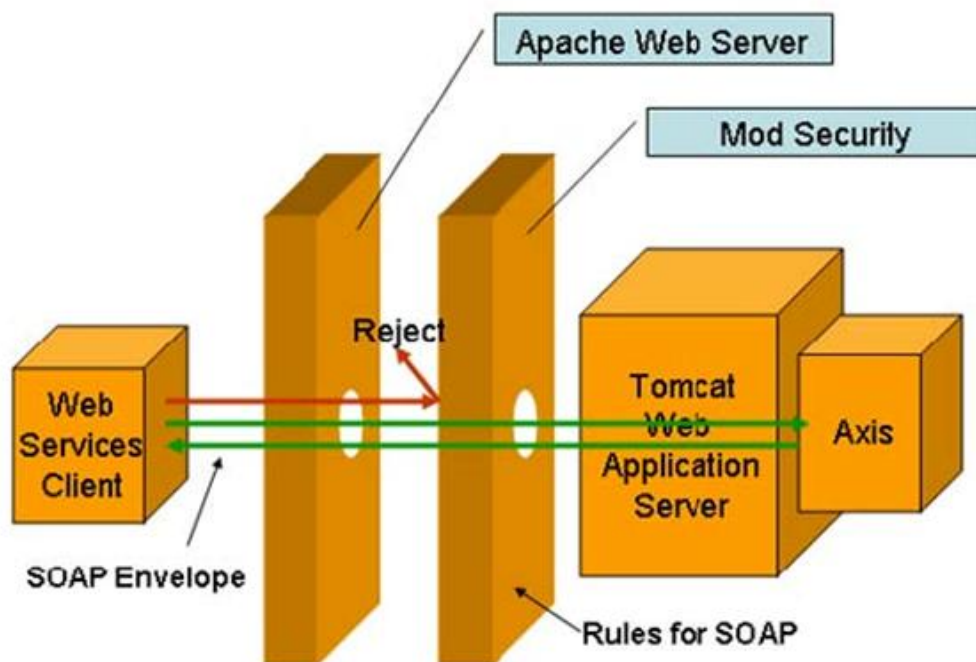
Η τυπική τοποθέτηση περιλαμβάνει την αντιγραφή του αρχείου .aar που αντιπροσωπεύει την διαδικτυακή εφαρμογή στο φάκελο υπηρεσιών του υποδοχέα.

3. Πρόσβαση στις τυπικές διαδικτυακές υπηρεσίες

Οι διαδικτυακές υπηρεσίες είναι προσβάσιμες μέσω της διαδικτυακής διεύθυνσης τους, π.χ. "http://localhost:8080/axis/.../name_of_aar". Αν χρησιμοποιείται διαφορετικός υποδοχέας από τον Apache Tomcat, η διαδικτυακή διεύθυνση μπορεί να είναι διαφορετική.

4. Αυτόματη παραγωγή WSDL κώδικα

Όταν τεθεί μία διαδικτυακή υπηρεσία, κατά την προσπέλαση της δημιουργείται αυτόματα ένα αρχείο WSDL με την κατάληξη .WSDL.



Εικόνα 5.2.1: Axis 2

(Πηγή: http://artemis.cslab.ntua.gr/el_thesis/artemis.ntua.ece/DT2008-0021/DT2008-0021.pdf)

5.2.1. Τα γενικά χαρακτηριστικά του AXIS2

Ο Axis2 ως πυρήνας Διαδικτυακών Υπηρεσιών τηρεί τις βασικές προϋποθέσεις για την λειτουργία τους, όπως η υποστήριξη βασικών πρωτοκόλλων επικοινωνίας, η υποστήριξη του βασικού προφίλ WS 1.1 κτλ.

Παρακάτω παρατίθεται ο πίνακας γενικών χαρακτηριστικών. Πέρα από τον πυρήνα Axis2 αποτυπώνεται επίσης η υποστήριξη ή μη, και άλλων πυρήνων διαδικτυακών υπηρεσιών.

Πίνακας γενικών χαρακτηριστικών AXIS2

(Πηγή: http://artemis.cslab.ntua.gr/el_thesis/artemis.ntua.ece/DT2008-0021/DT2008-0021.pdf)

Χαρακτηριστικό	Axis 1.x	Axis 2	Celtix	Glue	JBoss WS	XFire (1.2)	GlassFish	OracleAS 10g
Συνέπεια ως προς το Basic Profile 1.1 (WS-I)	X	X	X	X	X	X	X	X
Εύκολη Δημιουργία Υπηρεσιών από POJOs	X	X	X	X	X	X	X	X

Ανοιχτός Κώδικας	X	X	X		X	X	X	
RPC-Κωδικοποίηση	X			X	X		X	X
Υποστήριξη Spring	X	X	X			X	X	
Υποστήριξη REST		X					X	X
IDEA πρόσθετα		X			X	X	X	Ant
Eclipse πρόσθετα		X			X	X	X	Ant
NetBeans πρόσθετα							X	Ant
JDeveloper								X
Hot Deployment		X			X	?	X	X
Soap 1.1	X	X	X	X	X	X	X	X
Soap 1.2	X	X		X	X	X	X	X
Streaming XML (StAX based)		X				X	X	
WSDL 1.1 >Code (Client)	X	X	X	X	X	X	X	X
WSDL 1.1 >Code (Server)	X	X	X	X	X	X	X	X
WSDL 2.0 >Code (Client)		X	?	?	?	?		?
WSDL2.0 >Code (Server)		X	?	?	?	?		?
Ασύγχρονη Επικοινωνία από πλευρά Πελάτη	X	X					X	With BPEL
Ασύγχρονη Επικοινωνία από πλευρά Εξυπηρετητή	X	X					X	With BPEL
Δημιουργία Κώδικα με συγκεκριμένη Πολιτική		X						

Ο Apache Axis2 είναι πρότυπος, πιο αποτελεσματικός, και χρησιμοποιεί περισσότερο την γλώσσα XML απ' όσο η προηγούμενη έκδοση. Είναι προσεκτικά σχεδιασμένος για να υποστηρίζει την εύκολη ενσωμάτωση πρόσθετων μονάδων, οι οποίες συνεισφέρουν στην ασφάλεια και στην αξιοπιστία του συστήματος. Τα πρόσθετα αυτά είναι διαθέσιμα ή υπό κατασκευή και περιλαμβάνουν:

- WS-Αξιόπιστη Ανταλλαγή Μηνυμάτων -Υπό τη υποστήριξη του Apache Sandesha2
- WS-Συντονισμός και WS-Ατομική Δοσοληψία -Υπό τη υποστήριξη του Apache Kandula2
- WS-Ασφάλεια -Υπό τη υποστήριξη του Apache Rampart
- WS-Διευθυνσιοδότηση -Μονάδα που περιλαμβάνεται στον πυρήνα του Axis2

Ο Apache Axis2 έχει κατασκευαστεί πάνω στον Apache AXIOM, ένα μοντέλο υψηλής απόδοσης βασισμένο στην XML. Ο Axis2 συνοδεύεται από πολλά νέα χαρακτηριστικά, από ενισχυμένες λειτουργίες και από ειδικές επιχειρηματικές υλοποιήσεις. Πρόκειται για χαρακτηριστικά που τον καθιστούν καλύτερο σε σχέση με τους υπόλοιπους πυρήνες όπως φαίνεται και στον πίνακα 2:

- Ταχύτητα:** Ο Axis2 χρησιμοποιεί το δικό του μοντέλο αντικειμένου και το δικό του StAX (Streaming API for XML) αναλυτή κειμένου με αποτέλεσμα να παρέχει στους χρήστες του, μεγαλύτερη ταχύτητα από προηγούμενες εκδόσεις του Apache Axis.
- Μικρό μέγεθος απαιτούμενης μνήμης:** Ο Axis2 σχεδιάστηκε από κάτω προς τα πάνω προσπαθώντας διαρκώς να κρατήσει τις απαιτήσεις σε μνήμη σε χαμηλά επίπεδα.
- AXIOM:** Ο Axis2 συνοδεύεται από το δικό του ελαφρύ μοντέλο αντικειμένου, AXIOM, για επεξεργασία μηνυμάτων, το οποίο είναι επεκτάσιμο, έχει υψηλή απόδοση και είναι εύχρηστο στον προγραμματιστή.
- Εύκολη και Άμεσα Αναγνωρίσιμη Τοποθέτηση:** Ο Axis2 είναι εφοδιασμένος με την δυνατότητα τοποθέτησης διαδικτυακών υπηρεσιών και χειριστών ενώ το σύστημα είναι σε λειτουργία. Με άλλα λόγια, νέες διαδικτυακές υπηρεσίες μπορούν να προστεθούν στο σύστημα, χωρίς να διακόψει την λειτουργία του ο εξυπηρετητής. Αρκεί απλά, η τοποθέτηση του αρχείου της επιθυμητής διαδικτυακής υπηρεσίας στον φάκελο υπηρεσιών και το μοντέλο εγκατάστασης θα εγκαταστήσει αυτόματα την υπηρεσία και την κάνει διαθέσιμη προς χρήση.
- Ασύγχρονες Διαδικτυακές Υπηρεσίες:** Ο Axis2 πλέον υποστηρίζει ασύγχρονες διαδικτυακές υπηρεσίες και ασύγχρονη κλήση διαδικτυακών υπηρεσιών χρησιμοποιώντας ασύγχρονες εφαρμογές-πελάτες και ασύγχρονες μεταφορές.
- Υποστήριξη MEP:** Ο Axis2 πλέον παρέχει την ευελιξία για υποστήριξη των Προτύπων Ανταλλαγής Μηνυμάτων (Message Exchange Patterns (MEPs)) με εσωτερική υποστήριξη για τα βασικά MEPs όπως καθορίζονται στο WSDL 2.0.
- Ευελιξία:** Η αρχιτεκτονική του Axis2 δίνει στον προγραμματιστή τη πλήρη ελευθερία να εισάγει επεκτάσεις στο σύστημα για τυπική επεξεργασία των επικεφαλίδων, για τη διαχείριση του συστήματος, και για οτιδήποτε άλλο μπορεί να φανταστεί κανείς.
- Σταθερότητα:** Ο Axis2 προσδιορίζει ένα σύνολο από δημοσιευμένες διαπροσωπίες οι οποίες αλλάζουν σχετικά αργά, συγκριτικά με τα υπόλοιπα συστατικά του Axis.

ix. Ανάπτυξη προσανατολισμένη στα στοιχεία: Μπορεί κανείς εύκολα να ορίσει επαναχρησιμοποιούμενα δίκτυα των χειριστών προκειμένου να υλοποιήσει γνωστές τεχνικές επεξεργασίας των εφαρμογών του ή να τις μοιραστεί με τους συνεργάτες του. Πλαίσιο Μεταφορών: Υπάρχει πλέον μια ξεκάθαρη ιδέα για την ενοποίηση και την χρήση των μεταφορών μεταξύ πρωτοκόλλων (π.χ. αποστολές και ακροατές για SOAP πάνω σε διάφορα πρωτόκολλα όπως SMTP, FTP, μηνυματοστραφές ενδιάμεσο λογισμικό κτλ), και ο πυρήνας είναι πλήρως ανεξάρτητος των μεταφορών.

x. Υποστήριξη WSDL: Το Axis2 υποστηρίζει τις εκδόσεις 1.1 και 2.0, της Web Service Description Language, η οποία επιτρέπει στο χρήστη να έχει πρόσβαση σε απομακρυσμένες υπηρεσίες και επίσης να εξάγει αυτόματα αναγνώσιμες από μηχανές, περιγραφές της διαδικτυακής υπηρεσίας

xi. Πρόσθετα: Αρκετές προδιαγραφές των Διαδικτυακών Υπηρεσιών ενσωματώθηκαν συμπεριλαμβανομένου του WSS4J για ασφάλεια (Apache Rampart), του Sandesha για αξιόπιστη ανταλλαγή μηνυμάτων, του Kandula το οποίο είναι ενθυλάκωση των WS Συντονισμός, WS Ατομική Δοσοληψία and WS Επιχειρηματική Δραστηριότητα.

xii. Δυνατότητα Σύνθεσης και Επεκτασιμότητα: Υπάρχουν πολλές μονάδες που αυξάνουν την δυνατότητα για σύνθεση και επέκταση. Οι μονάδες αυτές, υποστηρίζουν την δυνατότητα σύνθεσης καθώς και τις καινούργιες προδιαγραφές WS -*, με έναν απλό και ξεκάθαρο τρόπο. Η τοποθέτηση τους ωστόσο δεν είναι άμεσα αναγνωρίσιμη, εφόσον αλλάζουν την όλη συμπεριφορά του συστήματος.

5.2.2. Tomcat

Ο Apache Tomcat είναι ένας δικτυακός υποδοχέας, ή εξυπηρετητής εφαρμογών που αναπτύχθηκε από την Apache Software Foundation (ASF). Ο Tomcat υλοποιεί τις προδιαγραφές του Java Servlet και των Java Server Pages (JSP) όπως αυτές προσδιορίζονται από τη Sun Microsystems, παρέχοντας ένα περιβάλλον για εκτέλεση Java κώδικα σε συνεργασία με έναν δικτυακό εξυπηρετητή (web server). Διαθέτει εργαλεία για τη ρύθμιση των παραμέτρων του και για τη διαχείριση του, αλλά μπορεί επίσης να ρυθμιστεί με επεξεργασία των αρχείων ρυθμίσεων τα οποία συνήθως έχουν XML μορφή. Ο Tomcat περιλαμβάνει τον δικό του εσωτερικό HTTP εξυπηρετητή.

Θεωρείται από τους πλέον αξιόπιστους Web Servers και χρησιμοποιείται ευρέως στην ανάπτυξη δικτυακών υπηρεσιών. Στην παρούσα εφαρμογή χρησιμοποιήθηκε η έκδοση 6.0.16 η οποία παρέχεται ελεύθερα στο διαδίκτυο μέσω του συνδέσμου <http://tomcat.apache.org>.

5.2.3. Apache HTTP Server

Ο Apache HTTP Server, ο οποίος συχνά αποκαλείται απλά Apache, είναι ένας web server γνωστός για το ρόλο κλειδί που έπαιξε στην αρχική ανάπτυξη του παγκόσμιου ιστού. Ο Apache ήταν η πρώτη βιώσιμη εναλλακτική στον Web Server Netscape Communications Corporation (γνωστός σήμερα ως Sun Java System Web Server), και έχει εξελιχθεί ώστε να ανταγωνιστεί άλλους Web Servers βασισμένους σε Unix όσον αφορά τη λειτουργικότητα και τις επιδόσεις.

Ο Apache αναπτύχθηκε και συντηρήθηκε από μια ανοιχτή κοινωνία αναλυτών υπό το πρίσμα της Apache Software Foundation. Η εφαρμογή είναι διαθέσιμη για μια ποικιλία λειτουργικών συστημάτων, όπως Unix, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X και Microsoft Windows. Ο Apache χαρακτηρίζεται ως ελεύθερο λογισμικό και ως λογισμικό ανοιχτού κώδικα.

5.3. ZK Framework

Το πλαίσιο ανάπτυξης πλούσιων διαδικτυακών εφαρμογών (Rich Internet Applications) ZK ανήκει στην κατηγορία των Java server-side ή server-centric Frameworks με πλήρη υποστήριξη Ajax, που αποκρύπτει απόλυτα από τον προγραμματιστή την ύπαρξη του XHR αντικειμένου και τη χρήση της JavaScript, ενώ υποστηρίζει το μοντέλο προγραμματισμού βασισμένο σε γεγονότα (event-driven programming). Το ZK εξαρτάται από την τεχνολογία Java, τόσο για την ανάπτυξη εφαρμογών όσο για την εκτέλεση τους. Σε αντίθεση δηλαδή με τα άλλα Frameworks, το ZK είναι στενά συνδεδεμένο με τις τεχνολογίες του Servlet API και εκτελείται σε ένα διακομιστή με περιβάλλον Servlet Container.

Το ZK επιτυγχάνει το στόχο του παρέχοντας ένα πλούσιο σύνολο αντικειμένων που ονομάζονται Components, και αντιστοιχούν σε στοιχεία διεπαφής χρήστη με Ajax δυνατότητες, μια ειδική γλώσσα σήμανσης (ZUML) για σχεδίαση σελίδων UI με ειδική μορφή που αποκαλούνται σελίδες ZK, κι έναν μηχανισμό απάντησης σε ενέργειες που προκαλεί ο χρήστης στη σελίδα. Το ZK είναι προϊόν ανοιχτού κώδικα, και η τρέχουσα έκδοση του είναι η 5.0.

5.3.1. Η αρχιτεκτονική του ZK

Ο πυρήνας του περιβάλλοντος εκτέλεσης και του μηχανισμού εξυπηρέτησης Ajax αιτημάτων αποτελείται από τρία τμήματα:

1. Το φορτωτή σελίδων ZK Loader, ο οποίος αναλαμβάνει να απαντήσει σε ένα αίτημα, ανακτώντας την κατάλληλη ZK σελίδα και μετατρέποντας την σε HTML μορφή κατάλληλη να εμφανιστεί στον περιηγητή.
2. Το μηχανισμό ZK Client Engine, μια συλλογή αντικειμένων γραμμένων σε JavaScript, που δημιουργείται κατά το πρώτο αίτημα του χρήστη και αποστέλλεται στον περιηγητή, όπου και παραμένει για όλη τη διάρκεια της συνεδρίας του χρήστη. Περιέχει τις απαραίτητες μεθόδους, αφενός για να επικοινωνεί ασύγχρονα με τον διακομιστή, μέσω ειδικών Ajax αιτημάτων που ονομάζονται αιτήματα ZK, και αφετέρου να διαχειρίζεται τα DOM (Dynamic Object Model) αντικείμενα της σελίδας αναλόγως.
3. Τον μηχανισμό Ασύγχρονης Εξυπηρέτησης ZK AU (Asynchronous Update) Engine, σύμφωνα με τον οποίο μια συλλογή αντικειμένων στην πλευρά του διακομιστή διαχειρίζεται ασύγχρονα αιτήματα που λαμβάνει από το Client Engine και αναλαμβάνει να προωθήσει την εξυπηρέτησή τους.

Συνεπώς, οι μηχανισμοί ZK Client και ZK AU συνεργάζονται για να υλοποιήσουν την επικοινωνία Ajax, με την συνεχή ενημέρωση του διακομιστή για τα γεγονότα που συμβαίνουν στην πλευρά του πελάτη και την ανάλογη ενημέρωση κομματιών της σελίδας.

Ακολουθεί ένα τυπικό σενάριο εξυπηρέτησης μιας ZK σελίδας από το Framework:

1. Το Servlet Container λαμβάνει ένα αίτημα, το οποίο αφορά σε μια ZK σελίδα, και το προωθεί στον ZK Loader.
2. ZK Loader εντοπίζει τη σελίδα και την ερμηνεύει για να δημιουργήσει τα αντικείμενά της. Αυτό γίνεται σε 4 φάσεις:
 - Φάση αρχικοποίησης: εκτελούνται ειδικές εντολές επεξεργασίας, αν υπάρχουν.
 - Φάση δημιουργίας: τα Components που απαρτίζουν τη σελίδα δημιουργούνται ως Java αντικείμενα.
 - Φάση εκτέλεσης γεγονότων: για κάθε ένα από τα Components εκτελείται μια μέθοδος αρχικοποίησης onCreate().
 - Φάση Μετατροπής σε HTML: όλα τα Components μετατρέπονται σε αντίστοιχο HTML και JavaScript κώδικα.
3. Το αποτέλεσμα προκύπτει ως HTML σελίδα, η οποία αποστέλλεται ως απάντηση στον πελάτη, μαζί με το ZK client engine.
4. Το client engine παραμένει στον περιηγητή και αντιδρά σε γεγονότα της σελίδας, με το να αποστέλλει αιτήματα ZK στο AU engine.
5. Το ZK AU επεξεργάζεται το αίτημα και προωθεί την εκτέλεσή του. Αν το αποτέλεσμα του αιτήματος αφορά σε αλλαγές στα αντικείμενα DOM της σελίδας, το ZK AU αποστέλλει στο ZK client τις αλλαγές με τη μορφή ειδικής ZK απάντησης.
6. Το ZK client ερμηνεύει την ZK απάντηση ως οδηγίες για τροποποίηση των στοιχείων της σελίδας και, στη συνέχεια, τις πραγματοποιεί.
7. Επαναλαμβάνονται τα βήματα 4-6 μέχρι να ληφθεί καινούριο αίτημα για νέα σελίδα.

Η παραπάνω αρχιτεκτονική δίνει την ελευθερία στον σχεδιαστή να επιλέξει το επίπεδο διαδραστικότητας κάθε component. Οι εφαρμογές που προκύπτουν μπορεί να καλύπτουν όλο το φάσμα, από απλές εφαρμογές τύπου MVC με συνήθεις λειτουργίες μέχρι εξαιρετικά διαδραστικές εφαρμογές που μιμούνται την πλούσια λειτουργικότητα που συναντάται σε αντίστοιχες Desktop εφαρμογές (Desktop-like Web applications).

5.3.2. Η γλώσσα σήμανσης ZUMML

Το ZK Framework παρέχει τη δική του γλώσσα σήμανσης για σχεδίαση σελίδων, που

βασίζεται στην XML, η οποία δημιουργήθηκε στα πλαίσια του Mozilla Project, προορισμένη να χρησιμοποιηθεί στον δημοφιλή περιηγητή Firefox.

Μια ZK σελίδα αποτελείται από μια ιεραρχία ετικετών XML, που η κάθε μία αντιστοιχεί σε μια Java κλάση για ένα component. Για παράδειγμα, η ετικέτα <window>, που αποτελεί υποχρεωτικά το πρώτο στοιχείο της σελίδας και περικλείει όλα τα υπόλοιπα, αναφέρεται στην κλάση-Component org.zkoss.zul.Window.

Όπως και στην HTML, οι ετικέτες μπορούν να περιέχουν δήλωση ιδιοτήτων που έχουν να κάνουν με την εμφάνιση των αντίστοιχων components. Επίσης, μπορεί να

περιέχουν αναφορές σε μεθόδους – χειριστές γεγονότων – του component. Για παράδειγμα, η δήλωση:

```
<button id="one_button" width="100px" label="Click me"
onClick="one_button.setValue("Clicked!")"/>
```

δημιουργεί τελικά ένα HTML button στη σελίδα με τιμή «Click me» και πλάτος 100 pixels, που όταν το πατήσει ο χρήστης, αλλάζει την τιμή του σε «Clicked!».

Από το παράδειγμα γίνεται προφανές ότι επιτρέπεται η χρήση της Java μέσα σε μια σελίδα ZUML. Η εντολή «one_button.setValue("Clicked!")» είναι μια Java εντολή απόδοσης τιμής στην ιδιότητα «value» του Java αντικείμενου «one_button». Χάρη στη συνεργασία των ZK AU και ZK Client engine που περιγράφηκε, η εντολή αυτή ενεργεί με τον ίδιο τρόπο ταυτόχρονα και στο DOM αντικείμενο «one_button», αλλάζοντας την τιμή του.

Εκτός από τις δηλώσεις χειριστών γεγονότων, ένα προγραμματιστής μπορεί επίσης να περιλάβει κώδικα Java σε μια ZUML σελίδα και μέσω της ειδικής ετικέτας <zscript>. Τα αντικείμενα που δημιουργούνται με αυτό τον τρόπο είναι προσβάσιμα στην υπόλοιπη σελίδα, π.χ. μέσα σε χειριστές γεγονότων, χρησιμοποιώντας εκφράσεις γραμμένες σε γλώσσα παρόμοια με την JSP Expression Language.

Πέρα από τα αντικείμενα Java που δημιουργεί ο χρήστης, η Expression Language (και κατ' επέκταση τα components της σελίδας) του ZK, έχει πρόσβαση σε ένα σύνολο υπονοουμένων (implicit) αντικειμένων. Αυτά αφορούν:

- Αντικείμενα για αναφορά σε components και στην ίδια τη σελίδα: τα αντικείμενα «self» και «event» μπορούν να χρησιμοποιηθούν μέσα σε μία μέθοδο χειριστή γεγονότων ενός component για να υπονοήσουν το ίδιο το component και το γεγονός που μόλις εκτελέστηκε, αντίστοιχα. Επίσης, είναι διαθέσιμο το αντικείμενο «page» που αναφέρεται στη σελίδα και το αντικείμενο «desktop» που αφορά σε όλο το περιβάλλον εκτέλεσης.
- Αντικείμενα που αντιστοιχούν στις οντότητες του Servlet API: όπως είναι γνωστό, το περιβάλλον εκτέλεσης ZK χρησιμοποιεί την υποδομή ενός Servlet Container, την οποία αποκρύπτει εν μέρει από τον προγραμματιστή. Όλα τα αντικείμενα που είναι διαθέσιμα σε μια τυπική εφαρμογή Servlet ή JSP, όπως τα request, session, applicationContext, γίνονται προσβάσιμα και στην ZUML σελίδα, μέσα από ειδικές μεταβλητές του ZK.

Η δυνατότητα ενσωμάτωσης καθαρού Java κώδικα σε μια ZUML σελίδα είναι σίγουρα χρήσιμη, ωστόσο, όπως και στην περίπτωση της JSP, δε θεωρείται καλή πρακτική και συνιστάται η ελαχιστοποίηση της χρήσης της.

5.3.3. Τα Components του ZK και ο χειρισμός γεγονότων

Στο ZK υπάρχει ένα εκτενές σύνολο από έτοιμα αντικείμενα διεπαφής με Ajax δυνατότητες. Αυτά αντιστοιχούν στις ετικέτες της ZUML σελίδας και ονομάζονται ZK Components. Τα ZK Components αντιστοιχούν σε ενισχυμένες διαδραστικές HTML δομές και έχουν διπλή «υπόσταση»: αφενός είναι Java αντικείμενα, αφετέρου αντιστοιχούν σε αντικείμενα του DOM δέντρου της σελίδας. Ο μηχανισμός του ZK εξασφαλίζει ότι οι δύο αυτές «υποστάσεις» έχουν πάντα

συγχρονισμένες καταστάσεις, δηλαδή ότι κάποιο γεγονός στην πλευρά του DOM εντοπίζεται από το αντικείμενο Java, και κάθε αλλαγή του αντικειμένου Java αντικατοπτρίζεται στο αντίστοιχο αντικείμενο DOM.

Τα components που προσφέρει το ZK, και που ξεπερνούν τα 120, διακρίνονται στις παρακάτω κατηγορίες:

- Δομικά Στοιχεία της σελίδας: αντικείμενα που συνήθως αντιστοιχούν σε περιοχές της σελίδας, περικλείουν άλλα components, και καθορίζουν τον τρόπο τοποθέτησής τους. Πιο συγκεκριμένα:
 - Window: το εξωτερικό αντικείμενο κάθε σελίδας, που αντιστοιχεί στο ομώνυμο αντικείμενο DOM. Αντίθετα με το μοντέλο του DOM, ένα ZK Window μπορεί να περικλείει άλλα αντικείμενα Window.
 - Box, Hbox, VBox: Περικλείουν αντικείμενα και καθορίζουν τον τρόπο τοποθέτησής τους, π.χ. σε οριζόντια διάταξη.
 - Layouts και Panels: τύποι αντικειμένων που καθορίζουν διακριτές περιοχές σε μια σελίδα, π.χ. το αντικείμενο ColumnLayout «χώριζε» τη σελίδα σε στήλες.
- Στοιχεία φόρμας και κειμένου: αντικείμενα που αντιστοιχούν στα συνήθη DOM αντικείμενα μιας HTML φόρμας όπως Button, Label, Checkbox, κ.α.
- Στοιχεία σύνθετων δομών: πιο σύνθετες δομές διάταξης περιεχομένου, όπως πίνακες (Grid), δένδρα (Tree), και λίστες επιλογής τιμών (Listbox). Οι δομές αυτές δεν αφορούν μόνο στην εμφάνιση, αλλά έχουν και στοιχεία λειτουργικότητας, π.χ. ένα αντικείμενο Listbox διαθέτει μια μέθοδο αυτόματης ταξινόμησης των τιμών που περιέχει.
- Στοιχεία Εργαλειοθήκης (Toolbar) και Μενού: αντικείμενα (Toolbar, Toolbarbutton, Menu, κ.α.) που δίνουν τη δυνατότητα δημιουργίας σύνθετων, πολλών επιπέδων λειτουργιών μενού, παρόμοιων με αυτές που προσφέρουν οι εφαρμογές Desktop.
- Στοιχεία Πολυμέσων: δίνουν την δυνατότητα προσθήκης πολυμεσικού περιεχομένου, εικόνων (Image), και αρχείων ήχου (Audio).

Πέρα από τα παραπάνω σύνθετα components, το ZK διαθέτει Java αντικείμενα σχεδόν για όλα τα αντικείμενα του DOM μοντέλου, ακόμα και τα πιο απλά. Στην περίπτωση που ο προγραμματιστής επιθυμεί να συμπεριλάβει αυτούσιο HTML κώδικα σε μια σελίδα, μπορεί να χρησιμοποιήσει το αντικείμενο Java «HTML», που παρέχει το ZK.

Επίσης, το ZK παρέχει ένα σύνολο αντικειμένων που αποτελούν «περιτύλιγμα» για αντικείμενα, οντότητες, ακόμα και JavaScript Widgets που προέρχονται από άλλες τεχνολογίες. Ενδεικτικά:

- Gmaps: αντικείμενο που προσθέτει στη σελίδα το –διάσημο– JavaScript Widget GoogleMaps, δηλαδή ένας διαδραστικό χάρτη, και το διαχειρίζεται με χρήση καθαρού Java κώδικα.
- FCKeditor: αντικείμενο-περιτύλιγμα για τον ομώνυμο δημοφιλή JavaScript Επεξεργαστή Εμπλουτισμένου Κειμένου (Rich Content Editor).

- Jasper Report: αντικείμενο εμφάνισης εγγράφων-αναφορών, που έχουν υλοποιηθεί με το γνωστό εργαλείο Jasper Reports, το οποίο καθιστά δυνατή τη δημιουργία αναφορών από συστήματα βάσεων δεδομένων.

Τέλος το ZK, παρά την πλούσια συλλογή του σε components, παρέχει τη δυνατότητα σε προγραμματιστές να δημιουργήσουν με εύκολο τρόπο νέα components, με συνδυασμό των υπαρχόντων ή ακόμα εισάγοντας μια τελείως νέα λειτουργικότητα, τα οποία στη συνέχεια μπορεί να διαμοιραστούν και να επαναχρησιμοποιηθούν από τα μέλη της κοινότητας Ανοιχτού Κώδικα.

Κάθε ένα από τα components που παρέχονται στο ZK αναγνωρίζει και ανταποκρίνεται σε ένα πλούσιο σύνολο πιθανών γεγονότων και ενεργειών του χρήστη, κάτι που τα καθιστά εξαιρετικά διαδραστικά. Τα γεγονότα αυτά, όπως αναφέρθηκε, εντοπίζονται στην πλευρά του πελάτη από το αντικείμενο ZK client Engine και στη συνέχεια αποστέλλονται στο διακομιστή, με ασύγχρονο πάντα τρόπο. Τα είδη των γεγονότων που υποστηρίζονται είναι διαφορετικά για κάθε component και ποικίλουν από τα πιο απλά, όπως π.χ. το γεγονός «ο χρήστης πάτησε ένα πλήκτρο», έως πολύ εξεζητημένα, όπως π.χ. «ο χρήστης εκτέλεσε τη λειτουργία ταξινόμησης σε ένα component τύπου Listbox».

Δύο πολύ ενδιαφέροντες τύποι γεγονότων στο ZK είναι:

- Γεγονότα Drag and Drop: σε όλα σχεδόν τα components του ZK μπορεί να ανατεθεί η συμπεριφορά Draggable και Droppable, που σημαίνει ότι ο χρήστης μπορεί να αλλάξει την τοποθέτηση τους στην σελίδα απλά, επιλέγοντας τα και σύροντας το δείκτη του ποντικιού στην επιθυμητή νέα θέση – μια λειτουργία αρκετά συνηθισμένη σε Desktop εφαρμογές, αλλά όχι και στις αντίστοιχες Web. Αυτά τα γεγονότα ενημερώνουν το διακομιστή για την αντίστοιχη ενέργεια του χρήστη.
- Γεγονός αλλαγής Σελιδοδείκτη: ειδοποιεί το διακομιστή ότι ο χρήστης πάτησε το κουμπί «πίσω» στον περιηγητή.

Το ZK παρέχει δύο εναλλακτικούς τρόπους δήλωσης των Java εντολών που θα εκτελεστούν στην πλευρά του διακομιστή ως αποτέλεσμα του εντοπισμού κάποιου γεγονότος. Ο στατικός τρόπος αναφέρθηκε ήδη στο παράδειγμα της ενότητας 6.3.2. και αφορά στη δήλωσή τους απευθείας μέσα στην ετικέτα του αντίστοιχου component, το οποίο αφορά στο γεγονός, στην ZUML σελίδα.

Ο δυναμικός τρόπος, ώστε να επιτευχθεί ακριβώς το ίδιο αποτέλεσμα με αυτό του παραδείγματος, είναι η εκτέλεση των παρακάτω Java εντολών κατά την αρχειοποίηση της σελίδας:

```
Button button=new Button("one_button");
button.setLabel("Click me!");
button.setWidth("100px");
button.addEventListener(Events.onClick, , new EventListener() {
    public void onEvent(Event event) throws Exception {
        button.setLabel("Clicked!");
    }
});
```

Προφανώς, ο δεύτερος τρόπος δίνει τη δυνατότητα για μεγαλύτερη ευελιξία, όταν αυτή χρειάζεται, και αποτελεί εξαιρετικά ισχυρό εργαλείο κατά το σχεδιασμό της εφαρμογής. Θα μπορούσε να χρησιμοποιηθεί, για παράδειγμα, στη τροποποίηση της συμπεριφοράς ενός γεγονότος δυναμικά, ή ως αποτέλεσμα της εκτέλεσης κάποιου άλλου γεγονότος πρώτα, ή για να ενεργοποιεί ή να απενεργοποιεί, ανάλογα με την κατάσταση της σελίδας, την ανταπόκριση ενός component σε συγκεκριμένα γεγονότα, κ.α.

Τέλος, το ZK παρέχει έναν τρόπο αυτόνομης δημιουργίας και χειρισμού γεγονότων στην πλευρά του πελάτη, χωρίς την παρέμβαση του χρήστη, μια διαδικασία που αναφέρεται κάποιες φορές με τον όρο "Ajax Push". Δηλαδή, είναι δυνατό να ενεργοποιηθούν γεγονότα απευθείας πάνω σε αντικείμενα DOM της σελίδας, μέσα από κώδικα Java που εκτελείται

στην πλευρά του διακομιστή, χρησιμοποιώντας τις μεθόδους `sendEvent` και `postEvent` της κλάσης `Events`. Αυτός ο ιδιαίτερα ισχυρός μηχανισμός πρέπει να χρησιμοποιείται προσεχτικά, ώστε να μην οδηγήσει σε απρόβλεπτες, για τον τελικό χρήστη, καταστάσεις.

5.4. Ant

Για λόγους ευκολίας στην εργασία χρησιμοποιήθηκε το πρόγραμμα `ant` για ενέργειες όπως `compile`, `build` κλπ. της εφαρμογής. Το πρόγραμμα `ant` χρησιμοποιείται για τη διαδικασία ελέγχου παραγωγής κώδικα στη γλώσσα Java. Στο `ant` δίνουμε ένα αρχείο XML (`buildfile`) το οποίο περιέχει τις εργασίες που απαιτούνται για την παραγωγή του προγράμματος και τις εξαρτήσεις μεταξύ των εργασιών αυτών. Το `ant` μας δίνει τη δυνατότητα να ορίσουμε δικές μας εργασίες και παρέχει και ένα πλήθος έτοιμων εργασιών, εκτός της μεταγλώττισης (δημιουργία αρχείων `jar`, `ftp`, χειρισμός αρχείων, αλληλεπίδραση με το `CVS`, αλληλεπίδραση με τον `Tomcat`, κ.λπ.)

Κεφάλαιο 6. Λεπτομέρειες Υλοποίησης

Στο κεφάλαιο αυτό παρουσιάζεται ο τρόπος υλοποίησης της επιχειρησιακής λογικής του συστήματος, η οποία αναπτύχθηκε σε προηγούμενη ενότητα καθώς και η υλοποίηση επιμέρους λειτουργιών που υποστηρίζει η εφαρμογή. Η υλοποίηση της εφαρμογής βασίστηκε στις πλατφόρμες (frameworks) J2EE και Spring κάνοντας χρήση επιλεγμένων συστατικών λογισμικού για την υλοποίηση κάθε λειτουργίας.

Ακολουθεί περιγραφή της υλοποίησης του τμήματος των web forms και του τμήματος των web services.

6.1. Υλοποίηση Web Forms

Το κομμάτι των web forms του συστήματος βασίστηκε στο αρχιτεκτονικό μοντέλο MVC, κομμάτια από το οποίο χρησιμοποιήθηκαν και από τα web services, όπως είναι τα DAO.

6.1.1. Υλοποίηση του στρώματος Μοντέλου (View Layer)

Εδώ «κρύβεται» σχεδόν ολόκληρη η επιχειρησιακή λογική της εφαρμογής. Το μοντέλο είναι ένα σύνολο τριών άλλων συνθετημάτων:

- Τα αντικείμενα μεταφοράς δεδομένων (DTO)
- Τα αντικείμενα πρόσβασης δεδομένων (DAO)
- Οι διαχειριστές των αντικειμένων (MANAGER)

6.1.1.1. Αντικείμενα μεταφοράς δεδομένων (DTO)

Τα αντικείμενα μεταφοράς δεδομένων (DTO-Data Transfer Object), τα οποία στην εφαρμογή ονομάζονται Value Objects (VO), περιέχουν τύπους δεδομένων και αντίστοιχες μεθόδους που αποθηκεύουν ή ανακτούν αυτά τα δεδομένα. Χρησιμοποιούνται για τη μεταφορά των δεδομένων ανάμεσα στα υποσυστήματα ή συνθετήματα της εφαρμογής. Για παράδειγμα αρχικοποιούνται από τους διαχειριστές των αντικειμένων και στη συνέχεια μεταφέρουν τις πληροφορίες, από τα αντικείμενα πρόσβασης δεδομένων DAO στην προβολή της εφαρμογής ή αντίστροφα δέχονται πληροφορίες από την προβολή της εφαρμογής και τις μεταφέρουν στα DAO ώστε να αποθηκευτούν στη Βάση Δεδομένων. Τα αντικείμενα μεταφοράς δεδομένων περιέχουν τις κατάλληλες δομές δεδομένων (για παράδειγμα Long,int,String,ArrayList κ.α.), ολόκληρα στιγμιότυπα άλλων αντικειμένων VO καθώς επίσης και κατάλληλες μεθόδους (Getters - Setters) για την ανάκτησή τους και την αποθήκευσή τους από και προς τη κλάση. Ένα παράδειγμα αντικειμένου μεταφοράς δεδομένων παρουσιάζεται στο σχήμα 6.1.1.

```

//-----
package mou.core.persistence.model;
//-----
import java.io.Serializable;
import java.util.LinkedList;
import java.util.List;
//-----
public class CompanyVO implements Serializable {
    //-----
    private static final long serialVersionUID =3335237854285610763L;
    //-----
    private long ent_id;
    private String ent_name;
    private String ent_title;

    .....
    //-----
    public CompanyVO() {
    }
    //-----
    public CompanyVO(long ent_id) {
        this.ent_id = ent_id;
    }
    //-----
    public long getEnt_id() {
        return ent_id;
    }
    //-----
    public void setEnt_id(long entId) {
        ent_id = entId;
    }
    //-----
    public String getEnt_name() {
        return ent_name;
    }
    //-----
    public void setEnt_name(String entName) {
        ent_name = entName;
    }
    //-----
    public String getEnt_title() {
        return ent_title;
    }
    //-----
    public void setEnt_title(String entTitle) {
        ent_title = entTitle;
    }
    //-----

    .....
}
//-----

```

Εικόνα 6.1.1 CompanyVO.java

6.1.1.2. Αντικείμενα πρόσβασης δεδομένων (DAO)

Τα αντικείμενα Πρόσβασης Δεδομένων (DAO: Data Access Object) είναι κλάσεις που παρέχουν συγκεκριμένες μεθόδους πρόσβασης σε δεδομένα χωρίς να προδίδουν λεπτομέρειες για αυτά. Με τη χρήση τους γίνεται ο διαχωρισμός σχετικά με το τι είδους δεδομένα χρειάζεται η εφαρμογή και ποιος είναι ο τρόπος απόκτησής τους από τη Βάση Δεδομένων. Ο διαχωρισμός αυτός είναι το βασικό πλεονέκτημα χρήσης των DAO. Η αλλαγή της επιχειρησιακής λογικής της εφαρμογής δεν έχει καμία επίδραση στον τρόπο ανάκτησης ή αποθήκευσης δεδομένων και αντίστοιχα η πιθανή μελλοντική αλλαγή του σχήματος της βάσης ή της τεχνολογίας της δεν αφορά καθόλου την λογική του συστήματος. Για παράδειγμα η κλάση, τύπου DAO, Company υλοποιεί το interface ICompanyDAO (σχήμα 6.1.2). Όσον αφορά τις επιχειρήσεις υπάρχουν πέντε βασικές μέθοδοι με τις οποίες η εφαρμογή επικοινωνεί με τη βάση διαμέσου των DAOs. Αυτές είναι οι:

- fetchCompany
- insertCompany
- updateCompany
- removeCompany
- companyExists

Όπως είναι αυτονόητο η μέθοδος fetchCompany χρησιμοποιείται κατά την κλήση της λίστας των επιχειρήσεων ώστε να μας φέρει τις επιχειρήσεις βάση των κριτηρίων που έχουμε ορίσει. Η μέθοδος insertCompany χρησιμοποιείται κατά την εισαγωγή μιας επιχείρησης. Αντίστοιχα οι μέθοδοι updateCompany και removeCompany χρησιμοποιούνται κατά την τροποποίηση και κατά την αντιγραφή μιας επιχείρησης αντίστοιχα. Τέλος η μέθοδος companyExists χρησιμοποιείται πριν από την εισαγωγή μιας επιχείρησης όπου γίνεται έλεγχος βάση του ΑΦΜ για το αν η συγκεκριμένη επιχείρηση υπάρχει ήδη στο σύστημα.

```
//-----  
package mou.core.persistence.dao.interfaces;  
//-----  
import mou.common.exceptions.result.AccResult;  
import mou.common.paging.Page;  
import mou.core.persistence.model.CompanyVO;  
//-----  
public interface ICompanyDao {  
    //-----  
    public CompanyVO fetchCompany(final CompanyVO filter);  
    //-----  
    public Page<CompanyVO> fetchCompanies(final CompanyVO filter, int  
pageNo, int pageSize);  
    //-----  
    public AccResult insertCompany(final CompanyVO company);  
    //-----  
    public AccResult updateCompany(final CompanyVO company);  
    //-----  
    public void removeCompany(final CompanyVO company);  
    //-----  
    public long companyExists(final CompanyVO company) throws  
Exception;  
    //-----  
}  
//-----
```

Εικόνα 6.1.2 ICompanyDAO.java

6.1.1.3. Διαχειριστής (Manager)

Οποιαδήποτε επικοινωνία ανάμεσα στον ελεγκτή (αναλύεται παρακάτω) και στο μοντέλο της εφαρμογής γίνεται μέσω του διαχειριστή (Manager). Ο διαχειριστής παρέχει στον ελεγκτή μέσω των μεθόδων που του προσφέρει πρόσβαση στα αντικείμενα διαχείρισης δεδομένων και από εκεί στη Βάση Δεδομένων του συστήματος. Βέβαια η αξιοποίησή του στην εφαρμογή δε περιορίζεται μόνο στη χρήση του ως γέφυρα ανάμεσα στα αναφερόμενα συνθετήματα της εφαρμογής. Βασικός του ρόλος είναι να μεταφέρει δεδομένα από και προς τις δυο κατευθύνσεις αλλά εφαρμόζοντας πρώτα την επιχειρησιακή λογική «Business Logic» που του δίνει ο προγραμματιστής. Δηλαδή ανάλογα με τα δεδομένα που δέχεται ως είσοδο και το σύνολο των παραμέτρων που καθορίζονται, ίσως από την περειαίρω επικοινωνία του με άλλα υποσυστήματα, παράγει νέα δεδομένα ως έξοδο μεταδίδοντάς τα στα κατάλληλα συνθετήματα-υποσυστήματα.

```
//-----  
package mou.core.persistence.dao;  
//-----  
import java.sql.Connection;  
.....  
//-----  
public class CompanyDaoImpl implements ICompanyDao {  
    //-----  
    private static final Logger log =  
Logger.getLogger(CompanyDaoImpl.class);  
    private DataSource dataSource;  
    private TransactionTemplate txTemplate;  
    private String insertCompany;  
    private String retrieveCompany;  
    private String updateCompany;  
    private String deleteCompany;  
    private String fetchCompanies;  
    //-----  
    public void setDataSource(DataSource dataSource) {  
        this.dataSource = dataSource;  
    }  
    //-----  
    public void setTransactionManager(PlatformTransactionManager  
txManager) {  
        this.txTemplate = new TransactionTemplate(txManager);  
  
this.txTemplate.setPropagationBehavior(DefaultTransactionDefinition.PRO  
PAGATION_REQUIRED);  
    }  
    //-----  
    @SuppressWarnings("unchecked")  
    public Page<CompanyVO> fetchCompanies(final CompanyVO filter, int  
pageNo, int pageSize) {  
        JdbcTemplate select = new JdbcTemplate(dataSource);  
        PaginationHelper<CompanyVO> ph = new  
PaginationHelper<CompanyVO>();  
        String query = fetchCompanies;  
        boolean applyCriteria = false;  
        if (filter.getEnt_name() != null &&  
filter.getEnt_name().length() > 0) {  
            query += (" where acc_enterp_info.ent_name like '" +  
filter.getEnt_name() + "'");  
            applyCriteria = true;  
        }  
    }  
}
```

```

    }
    if (filter.getEnt_afm() != null && filter.getEnt_afm().length()
> 0) {
        if(applyCriteria){
            query += (" and acc_enterp_info.ent_afm like '" +
filter.getEnt_afm() + "'");
        }else{
            query += (" where acc_enterp_info.ent_afm like '" +
filter.getEnt_afm() + "'");
            applyCriteria = true;
        }
    }
}
return ph.fetchPage(
    select,
    "SELECT count(acc_enterp_info.ent_id) " +
fetchCompanies.substring(query.indexOf("from")),
    query,
    new Object[] {},
    Paging.YES,
    pageNo,
    pageSize,
    new ParameterizedRowMapper<CompanyVO>() {
        public CompanyVO mapRow(ResultSet rs, int i) throws
SQLException {
            CompanyVO company = new CompanyVO();
            company.setEnt_id(rs.getLong("ent_id"));
            company.setEnt_name(rs.getString("ent_name"));

            company.setEnt_title(rs.getString("ent_title"));
            .....
            return company;
        }
    }
);
}
//-----
@SuppressWarnings("unchecked")
public CompanyVO fetchCompany(final CompanyVO company) {
    JdbcTemplate select = new JdbcTemplate(dataSource);
    return (CompanyVO) select.queryForObject(retrieveCompany,
        new Object[] {company.getEnt_id()},
        new RowMapper() {
            public Object mapRow(ResultSet rs, int rowNum)
throws SQLException {
                CompanyVO _company = new CompanyVO();
                _company.setEnt_id(rs.getLong("ENT_ID"));
                _company.setEnt_name(rs.getString("ENT_NAME"));

                _company.setEnt_title(rs.getString("ENT_TITLE"));
                .....
                return _company;
            }
        }
    ));
}
//-----

```

```

@SuppressWarnings("unchecked")
public AccResult insertCompany(final CompanyVO company) {
    final AccResult result = new AccResult();
    final JdbcTemplate jdbcTemplate = new JdbcTemplate(dataSource);
    final KeyHolder keyHolder = new GeneratedKeyHolder();
    try {
        txTemplate.execute(new TransactionCallbackWithoutResult() {
            public void
doInTransactionWithoutResult(TransactionStatus status) {
                jdbcTemplate.update(
                    new PreparedStatementCreator() {
                        public PreparedStatement
createPreparedStatement(Connection connection) throws SQLException {
                            PreparedStatement ps =
connection.prepareStatement(insertCompany, new String[]{"ENT_ID"});
                            if(company.getEnt_name() != null){
                                ps.setString(1,
company.getEnt_name());
                                }else{
                                    ps.setNull(1, Types.NULL);
                                }
                                if(company.getEnt_title() != null){
                                    ps.setString(2,
company.getEnt_title());
                                }else{
                                    ps.setNull(2, Types.NULL);
                                }
                                .....
                                return ps;
                            }
                        },
                    keyHolder);
                }
            });
        result.setErrorCode(keyHolder.getKey().longValue());
        return result;
    } catch (DataAccessException e) {
        e.printStackTrace();
        result.setErrorCode(-1);
        result.setErrorDescription(e.getMessage());
        return result;
    }
}
//-----

```

```

@SuppressWarnings("unchecked")
public AccResult updateCompany(final CompanyVO company) {
    final AccResult result = new AccResult();
    try {
        final JdbcTemplate jdbcTemplate = new
JdbcTemplate(dataSource);
        txTemplate.execute(new TransactionCallbackWithoutResult() {
            public void
doInTransactionWithoutResult(TransactionStatus status) {
                jdbcTemplate.update(updateCompany,
                    new Object[]{
                        company.getEnt_name(),
                        company.getEnt_title(),
                        .....
                    });
            }
        });
        result.setErrorCode(0);
        return result;
    } catch (DataAccessException e) {
        e.printStackTrace();
        result.setErrorCode(-1);
        result.setErrorDescription(e.getMessage());
        return result;
    }
}
//-----
public long companyExists(final CompanyVO company) throws Exception
{
    CompanyExists companyExists = new CompanyExists(dataSource);
    return companyExists.execute(company.getEnt_afm());
}
//-----
public void removeCompany(final CompanyVO company) {
}
//-----
public String getInsertCompany() {
    return insertCompany;
}
//-----
public void setInsertCompany(String insertCompany) {
    this.insertCompany = insertCompany;
}
//-----
public String getRetrieveCompany() {
    return retrieveCompany;
}
//-----
public void setRetrieveCompany(String retrieveCompany) {
    this.retrieveCompany = retrieveCompany;
}
//-----
public String getUpdateCompany() {
    return updateCompany;
}
//-----

```

```

public void setUpdateCompany(String updateCompany) {
    this.updateCompany = updateCompany;
}
//-----
public String getDeleteCompany() {
    return deleteCompany;
}
//-----
public void setDeleteCompany(String deleteCompany) {
    this.deleteCompany = deleteCompany;
}
//-----
public String getFetchCompanies() {
    return fetchCompanies;
}
//-----
public void setFetchCompanies(String fetchCompanies) {
    this.fetchCompanies = fetchCompanies;
}
//-----
}
//-----

```

Εικόνα 6.1.3 CompanyDAOImpl.java

6.1.2 Υλοποίηση του στρώματος της Προβολής (View Layer)

Η προβολή (View) παράγει μια παρουσίαση των δεδομένων στον χρήστη της εφαρμογής. Αναλόγως τη τεχνολογία που έχει επιλεγεί για την ανάπτυξη της διεπαφής χρήστη και εφαρμογής, η προβολή μπορεί να αποτελείται από αρχεία τύπου JSP, HTML, XHTML κ.ά. Στη δική μας περίπτωση όπως αναφέρθηκε και νωρίτερα θα είναι σελίδες zui. Στα αρχεία αυτά περιγράφεται ο τρόπος απεικόνισης των δεδομένων του μοντέλου και επιπλέον δίνεται η δυνατότητα στον χρήστη να διαχειριστεί τα δεδομένα αυτά και να αλληλεπιδράσει με την εφαρμογή, μέσα στα στενά πάντα περιθώρια που η εφαρμογή καθορίζει, αναλόγως με τον τρόπο που έχει αναπτυχθεί από τον προγραμματιστή.

Για κάθε λειτουργία του συστήματος αναπτύσσεται διαφορετική προβολή που αλληλεπιδρά με το αντίστοιχο μοντέλο μέσω των αντικειμένων μεταφοράς δεδομένων (DTOs). Στο σχήμα 6.1.4 φαίνεται ο κώδικας δημιουργίας της προβολής για τη λίστα των επιχειρήσεων.

```

<?xml version="1.0" encoding="UTF-8"?>
<?init class="org.zkoss.zk.ui.util.Composition" arg0="/layouts/standard.zul"?>
<?init class="org.zkoss.zkplus.databind.AnnotatedDataBinderInit"
arg0="./companylist"?>
<?variable-resolver class="org.zkoss.spring.DelegatingVariableResolver"?>
<?component name="toolbarbutton" extends="toolbarbutton"
class="mou.common.zkcomponents.MouToolbarbutton" ?>
<?component name="mypaging" extends="paging"
class="mou.common.zkcomponents.MouPaging" ?>

<window id="companylist"
use="mou.acc.windows.acc.CompanyListWindow"
xmlns="http://www.zkoss.org/2005/zul"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.zkoss.org/2005/zul
http://www.zkoss.org/2005/zul/zul.xsd"
self="{define(content)}" xmlns:h="http://www.w3.org/1999/xhtml">

    <!-- MENU -->
    <div width="100%" sclass="menudiv" align="right">
        <hbox id="menuBox" valign="middle">
            <menubar autodrop="false">
                <menu label="Διαχείριση" tooltipText="Διαχείριση Σώρευσης"
image="/img/create_e.gif">
                    <menupopup>
                        <menuitem label="Αλλαγή Συνθηματικού"
image="/img/gray_square.jpg"
                                sclass="menu"
onClick="companylist.onChangePasswordClick()"/>
                    </menupopup>
                </menu>
            </menubar>
            <toolbarbutton id="startpageButton" label="Αρχική Σελίδα"
tooltipText="Επιστροφή στην Αρχική Σελίδα" sclass="cancelbutton"
onClick="companylist.onCancelButtonClick()"/>
            <toolbarbutton id="cancelButton" label="Επιστροφή"
tooltipText="Επιστροφή" sclass="cancelbutton"
onClick="companylist.onCancelButtonClick()"/>
        </hbox>
    </div>

    <vbox id="container" width="100%" style="padding:0 5px 5px 5px; min-
height: 336px;">
        <!-- CRITERIA -->
        <vbox align="left" sclass="criteriaDiv">
            <hbox align="center" widths="50%, 50%">
                <div align="left">
                    <label value="Κριτήρια αναζήτησης Επιχειρήσεων"
style="color:#153881; font-size:11px; font-weight:bold;"/>
                </div>
                <div align="right">
                    <hbox align="center">
                        <listbox id="pageSizeList" mold="select" rows="1"
sclass="lstbox" width="45px"
onSelect="companylist.onFilterButtonClick();">
                            <listitem value="10" label="10"/>

```

```

<listitem value="20" label="20" selected="true"/>
    <listitem value="50" label="50"/>
</listbox>
<label value="αποτελέσματα ανά σελ." style="color:
#091A59;"/>
    </hbox>
</div>
</hbox>
<hbox align="center" style="margin-top:5px;">
    <vbox style="margin-right:3px;">
        <div sclass="criterion">
            <label value="Επωνυμία Επιχείρησης (Εισαγωγή % για
αναζήτηση)" width="100px"/>
        </div>
        <textbox id="ent_name" sclass="atextbox"/>
    </vbox>
    <vbox style="margin-right:3px;">
        <div sclass="criterion">
            <label value="Α.Φ.Μ" width="80px"/>
        </div>
        <textbox id="afm" sclass="atextbox" maxlength="12"/>
    </vbox>
    <vbox style="margin-right:3px;">
        <div sclass="criterion">
            <label value="Νομική Μορφή"/>
        </div>
        <listbox id="lfList" mold="select" rows="1"
            sclass="lstbox" width="170px">
            <listitem label="Όλες οι νομικές μορφές..."
selected="true"/>
        </listbox>
    </vbox>
    <div style="padding:15px 0px 0px 5px;">
        <toolbarbutton id="filterButton"
onClick="companylist.onFilterButtonClick();" label="."
width="16px" height="16px"
tooltiptext="Εκτέλεση Αναζήτησης" sclass="filterbutton"/>
    </div>
    <div style="padding:15px 0px 0px 5px;">
        <toolbarbutton id="resetButton"
onClick="companylist.onResetButtonClick();" label="."
width="16px" height="16px"
tooltiptext="Καθαρισμός" sclass="resetbutton"/>
    </div>
</hbox>
</vbox>
<div id="companyListDiv">
    <listbox id="companyList" fixedLayout="true"
onSelect="companylist.onCompanyListSelect()"
rows="0" multiple="false" context="optionsPopup"
width="994px" tooltiptext="Επιλέξτε Επιχείρηση και δεξί κλικ για ενέργειες">
        <listhead sizable="true">
            <listheader id="companynameHeader" align="center"
label="Επωνυμία Επιχείρησης" sort="auto" width="490px"/>
            <listheader id="afmHeader" align="center" label="Α.Φ.Μ
Επιχείρησης" sort="auto" width="110px"/>
            <listheader id="kadHeader" align="center" label="ΚΑΔ"
sort="auto" width="100px"/>
            <listheader id="legalformHeader" align="center"
label="Νομική Μορφή" sort="auto" width="100px"/>
        </listhead>

```



```

</listbox>
    <mypaging id="myPaging" autohide="false" detailed="true"
listbox="companyList"/>
    <menupopup id="optionsPopup">
        <menuitem id="viewButton" label="Επισκόπηση Επιχείρησης"
image="/img/view_e.gif"
            onClick="companylist.onViewButtonClick()"/>
        <menuitem id="editButton" label="Επεξεργασία Επιχείρησης"
image="/img/edit_e.gif"
            onClick="companylist.onEditButtonClick()"/>
        <menuitem id="addShareholderButton" label="Προσθήκη Μειόχου"
image="/img/add.gif"
            onClick="companylist.onAddShareholderButtonClick()"/>
        <menuitem id="checkAccumulationButton" label="Έλεγχος
Σώρευσης" image="/img/accumulation.png"
            onClick="companylist.onCheckAccumulationButtonClick()"/>
    </menupopup>
</div>
<div id="errorsdiv" sclass="deltaerrorsdiv" visible="false">
    <label id="noResultsLabel" value="Δεν βρέθηκαν Επιχειρήσεις!"/>
</div>
</vbox>
</window>

```

Εικόνα 6.1.4 companyList.zul

6.1.3. Υλοποίηση του στρώματος Ελεγκτή (Controller Layer)

Όπως προαναφέρθηκε, ο ελεγκτής αποτελεί τον συνδετικό κρίκο των παραπάνω στρωμάτων της προβολής και του μοντέλου. Συνεπώς για κάθε ζευγάρι μοντέλου – προβολής αναπτύσσεται και ο αντίστοιχος ελεγκτής που πραγματοποιεί τη διασύνδεση με το μοντέλο. Μέσα σε κάθε αρχείο ελεγκτή καθορίζονται όλες οι λειτουργίες που θα πρέπει να εκτελούνται από τον ελεγκτή. Στην αρχή της κλάσης του ελεγκτή γίνεται η σύνδεσή του με το αντίστοιχο μοντέλο (σχήμα 6.1.5).

Μέσα στη κλάση του ελεγκτή για τις επιχειρήσεις είναι υλοποιημένες οι παρακάτω μέθοδοι που είχαν δημιουργηθεί και στο αντίστοιχο DAO:

- fetchCompany
- fetchCompanies
- insertCompany
- updateCompany
- removeCompany
- companyExists

```

//-----
package mou.core.controller;
//-----
import mou.common.exceptions.result.AccResult;
import mou.common.paging.Page;
import mou.core.persistence.dao.CompanyDaoImpl;
import mou.core.persistence.model.CompanyVO;
//-----
public class CompanyController {
    //-----
    private CompanyDaoImpl dao;
    //-----
    public CompanyVO fetchCompany(CompanyVO company) {
        return dao.fetchCompany(company);
    }
    //-----
    public Page<CompanyVO> fetchCompanies(final CompanyVO filter, int pageNo,
int pageSize) {
        return dao.fetchCompanies(filter, pageNo, pageSize);
    }
    //-----
    public AccResult insertCompany(final CompanyVO company) {
        return dao.insertCompany(company);
    }
    //-----
    public AccResult updateCompany(final CompanyVO company) {
        return dao.updateCompany(company);
    }
    //-----
    public void removeCompany(final CompanyVO company) {
        dao.removeCompany(company);
    }
    //-----
    public long companyExists(final CompanyVO company) throws Exception {
        return dao.companyExists(company);
    }
    //-----
    public void setDao(CompanyDaoImpl dao) {
        this.dao = dao;
    }
    //-----
}
//-----

```

Εικόνα 6.1.5 CompanyController.java

6.2. Υλοποίηση Web Service

Το κομμάτι των web services αποτελείται από ένα βασικό web service, το οποίο περιλαμβάνει δύο κύριες μεθόδους. Η μια μέθοδος κάνει τον έλεγχο της σώρευσης και μας ενημερώνει για τον αν μπορεί η επιχείρηση να χρηματοδοτηθεί ή έχει εξαντλήσει το όριο των 200.000€ και η δεύτερη κάνει την εισαγωγή της επιχείρησης καθώς και του επενδυτικού σχεδίου στο οποίο συμμετέχει εφόσον έχει περάσει επιτυχώς τον έλεγχο σώρευσης.

Όπως φαίνεται και στο wsdl και το αντίστοιχο xsd που δημιουργήθηκαν για την υλοποίηση του web service (βλέπε παράρτημα, ενότητες 11.1 AccumulationService.wsdl και 11.2 InsertIPlan.xsd) τα web services που δημιουργήθηκαν είναι το checkAccumulation και το insertIPlan.

6.2.1. Μέθοδος checkAccumulation

Η πρώτη μέθοδος και λιγότερο σύνθετη είναι αυτή του ελέγχου σώρευσης. Ουσιαστικά αυτή η μέθοδος παίρνει σαν παραμέτρους το ΑΦΜ της επιχείρησης και το ποσό της επένδυσης με το οποίο θέλουμε να τη χρηματοδοτήσουμε, και μας ενημερώνει για το αν η επιχείρηση έχει εξαντλήσει το επιτρεπόμενο όριο των 200.000€ που έχει ορίσει η Ευρωπαϊκή Ένωση.

Παρακάτω φαίνεται το request που στέλνουμε για έλεγχο σώρευσης της επιχείρησης με ΑΦΜ 999467638 και ποσό επένδυσης 10.000€.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:par="http://parameters.services.accws.mou">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" soapenv:mustUnderstand="1">
      <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="USRToken">
        <wsse:Username>accforeas</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">accforeas</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <par:checkAccumulationRequest>
      <par:afm>999467638</par:afm>
      <par:poso>10000</par:poso>
    </par:checkAccumulationRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Εικόνα 6.2.1 checkAccumulation request

Η απάντηση που λαμβάνουμε από το web service

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <checkAccumulationResponse xmlns="http://parameters.services.accws.mou/">
      <balance>32030.75</balance>
      <errorCode>0</errorCode>
      <errorDescription>Για την επιχείρηση (ΑΦΟΙ ΙΩΑΝ. ΛΕΤΣΙΟΥ Ο.Ε.) για το Α.Φ.Μ
(999467638) που δηλώσατε και για το ποσό (10000) μπορείτε να εγκρίνετε το επενδυτικό
σχέδιο</errorDescription>
    </checkAccumulationResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Εικόνα 6.2.2 checkAccumulation response

Αρα η συγκεκριμένη επιχείρηση καλύπτει τις προϋποθέσεις που όρισε η Ευρωπαϊκή Ένωση και μπορεί να ενισχυθεί με το ποσό των 10.000€.

6.2.2. Μέθοδος insertIPlan

Η δεύτερη μέθοδος περιλαμβάνει όλη τη πληροφορία που χρειάζεται για να καταχωρηθεί ένα επενδυτικό σχέδιο. Όπως φαίνεται και στο request στο Παράρτημα 10.3 Request InsertIPlan.xml αρχικά στέλνονται τα πλήρη στοιχεία της επιχείρησης. Αν η επιχείρηση είναι ήδη καταχωρημένη τα αγνοεί διαφορετικά την καταχωρεί. Έπειτα στέλνονται τα στοιχεία της προκήρυξης τα οποία καταχωρούνται μόνο αν η προκήρυξη δεν είναι καταχωρημένη στο σύστημα. Μετά από κάθε καταχώρηση είτε επιχείρησης είτε προκήρυξης γίνεται commit ανεξάρτητα με το αν θα καταχωρηθεί επενδυτικό σχέδιο. Αυτό γίνεται γιατί θέλουμε το πληροφοριακό μας σύστημα να είναι όσο το δυνατόν περισσότερο ενημερωμένο.

Αφού έχει γίνει καταχώρηση της επιχείρησης και της προκήρυξης από την οποία προκύπτει η επένδυση, καλείται η μέθοδος checkAccumulation (ενότητα 7.1.1) και γίνεται έλεγχος σώρευσης της επιχείρησης με το ποσό του επενδυτικού σχεδίου που στάλθηκε με το web service. Τα στοιχεία του επενδυτικού σχεδίου καταχωρούνται μόνο εφόσον ο έλεγχος σώρευσης είναι επιτυχημένος.

Η απάντηση που λαμβάνουμε από το web service

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <checkAccumulationResponse xmlns="http://parameters.services.accws.mou/">
      <balance>22030.75</balance>
      <errorCode>0</errorCode>
      <errorDescription>Για το Α.Φ.Μ (999467638) που δηλώσατε και για το ποσό (10000)
μπορείτε να εγκρίνετε το επενδυτικό σχέδιο</errorDescription>
    </checkAccumulationResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

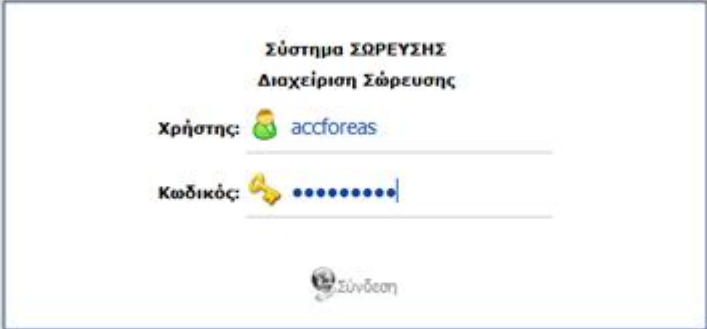
Εικόνα 6.2.3 insertIPlan response

6.3. Ασφάλεια Συστήματος

Η ασφάλεια ενός πληροφοριακού συστήματος είναι άμεσα συνυφασμένη με την αξιοπιστία του και την αποδοχή του από τους χρήστες-επισκέπτες τους. Ένα πληροφοριακό σύστημα πρέπει να παρέχει επαρκές επίπεδο ασφάλειας και αξιοπιστίας διασφαλίζοντας τις εξής παραμέτρους:

- i. Ακεραιότητα (integrity): Η πληροφορία που δημοσιεύεται, διακινείται, επεξεργάζεται και αποθηκεύεται παραμένει αναλλοίωτη.
- ii. Εμπιστευτικότητα (confidentiality): Πρόσβαση στην πληροφορία έχουν μόνο όσοι διαθέτουν κατάλληλη εξουσιοδότηση.
- iii. Αναγνώριση (identification): Ο προσδιορισμός της ταυτότητας του χρήστη.
- iv. Πιστοποίηση ταυτότητας (authentication): Η ενέργεια που διασφαλίζει ότι η ταυτότητα που δηλώνει ο χρήστης είναι η πραγματική.
- v. Εξουσιοδότηση (authorization): Η εξασφάλιση ότι κάθε οντότητα έχει πρόσβαση στους επιτρεπόμενους σε αυτή πόρους του συστήματος, συμπεριλαμβανομένης της ίδιας της πληροφορίας.
- vi. Διαθεσιμότητα (availability): Η πληροφορία είναι διαθέσιμη κάθε στιγμή που ένας εξουσιοδοτημένος χρήστης επιχειρεί να αποκτήσει πρόσβαση σε αυτή.
- vii. Μη άρνηση συμμετοχής (non-repudiation): Ένας χρήστης δεν μπορεί να αρνηθεί ότι εκτέλεσε μία ενέργεια σχετική με πρόσβαση, καταχώρηση και επεξεργασία πληροφορίας.


Για την πρόσβαση στο σύστημα είτε αυτό είναι το τμήμα των web forms είτε το τμήμα των web services απαιτείται η εισαγωγή αναγνωριστικού (username) και κωδικού πρόσβασης (password) με τα οποία γίνεται εξακρίβωση της ταυτότητας των χρηστών.



Σύστημα ΣΩΡΕΥΣΗΣ
Διαχείριση Σώρευσης

Χρήστης:

Κωδικός:

Υλοποίηση / Υποστήριξη Ιστοχώρου 

Εικόνα 6.3.1 Οθόνη εισαγωγής ονόματος και κωδικού χρήστη

Αναφορικά με την ασφάλεια των web services:

- Η αυθεντικοποίηση (authentication) γίνεται σε επίπεδο μηνύματος (message layer) με μηχανισμούς που ακολουθούν τις προδιαγραφές του προτύπου OASIS WS-Security. Για την υλοποίηση του μηχανισμού αυθεντικοποίησης χρησιμοποιήθηκε το Apache Rampart module (έκδοση 1.5).
- Η ακεραιότητα (integrity) και εμπιστευτικότητα (confidentiality) εξασφαλίζεται με κρυπτογράφηση των μηνυμάτων σε επίπεδο μεταφοράς (transport layer) με χρήση SSL.

Για την αυθεντικοποίηση έχουν υλοποιηθεί μηχανισμοί που ακολουθούν τις προδιαγραφές του προτύπου OASIS WS-Security. Συγκεκριμένα, κατά τη διαδικασία αυθεντικοποίησης η ταυτότητα του καλούντος (αποστολέας) που αιτείται αυθεντικοποίηση, ελέγχεται βάσει ενός ονόματος χρήστη (username) και κωδικού χρήστη (password) το οποίο αποστέλλεται μαζί με το υπόλοιπο SOAP μήνυμα. Η αυθεντικοποίηση με χρήση αναγνωριστικού/ συνθηματικού απαιτεί την υλοποίηση του στοιχείου (element) <UsernameToken> στην επικεφαλίδα (header) του μηνύματος. Έτσι η ταυτότητα αυτού που καλεί το webservice θα ελέγχεται σε κάθε μήνυμά του. Αυτό σημαίνει ότι όλα τα μηνύματα θα πρέπει να φέρουν στην επικεφαλίδα τους ένα έγκυρο ζεύγος αναγνωριστικού/συνθηματικού όπως φαίνεται

Το ακόλουθο απόσπασμα μηνύματος (message fragment) αποτελεί χαρακτηριστικό παράδειγμα ενός έγκυρου SOAP μηνύματος στο οποίο μαζί με τα υπόλοιπα στοιχεία συνυπάρχουν και τα στοιχεία που απαιτούνται για την ταυτοποίηση του αποστολέα. Οι πληροφορίες accforeas και accforeas αποτελούν αντίστοιχα το αναγνωριστικό και το συνθηματικό του αποστολέα. Να σημειωθεί ότι, η τρέχουσα έκδοση των προσφερόμενων υπηρεσιών δεν απαιτεί κρυπτογράφηση του συνθηματικού. Αυτό σημαίνει ότι το συνθηματικό θα πρέπει να αποστέλλεται ως plain text. Για την προστασία του μηνύματος κατά τη μεταφορά μέσω δικτύου χρησιμοποιείται SSL.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:par="http://parameters.services.accws.mou">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" soapenv:mustUnderstand="1">
      <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="USRToken">
        <wsse:Username>accforeas</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">accforeas</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <par:checkAccumulationRequest>
      <par:afm>999467638</par:afm>
      <par:poso>10000</par:poso>
    </par:checkAccumulationRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Εικόνα 6.3.2 Μήνυμα SOAP στο οποίο αποστέλλονται όνομα και κωδικός χρήστη

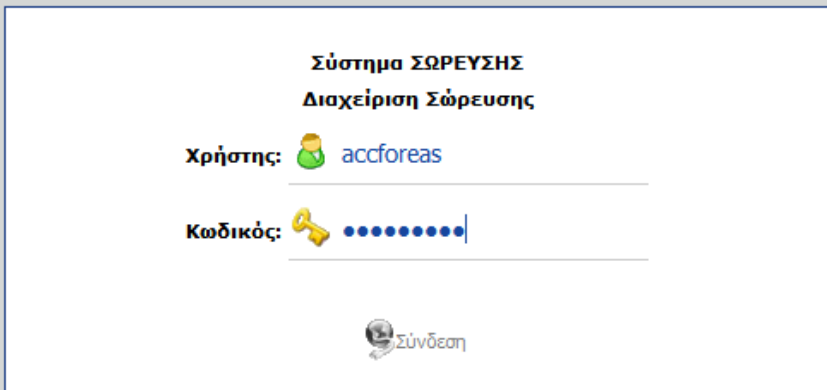
Κεφάλαιο 7. Σενάριο χρήσης εφαρμογής

Σε αυτό το κεφάλαιο θα παρουσιαστεί ένα σενάριο χρήσης του συστήματος από έναν Φορέα Διαχείρισης Κρατικών Ενισχύσεων.


➤ Βήμα 1: Εισαγωγή στο σύστημα


Ο Φορέας διαθέτει έναν κωδικό εισόδου στο σύστημα Κρατικών Ενισχύσεων. Για να τον αποκτήσει θα πρέπει να υποβάλει σχετικό αίτημα.


Ο Φορέας με την χρήση ενός φυλλομετρητή πλοηγείται στην διεύθυνση (<http://195.167.92.174:8080/acc/login.zul>) και χρησιμοποιώντας τον κωδικό του εισάγεται στο σύστημα:




Σύστημα ΣΩΡΕΥΣΗΣ
Διαχείριση Σώρευσης

Χρήστης:  accforeas

Κωδικός:  ●●●●●●●●●●

 Σύνδεση

Υλοποίηση / Υποστήριξη Ιστοχώρου 

➤ Βήμα 2: Έλεγχος σώρευσης

Στην αρχική οθόνη ο φορέας μπορεί να κάνει έναν άμεσο έλεγχο για το αν μία συγκεκριμένη επιχείρηση ξεπερνά το μέχρι τώρα όριο σώρευσης εισάγοντας το ΑΦΜ και το ποσό επιχορήγησης.



Σύστημα Σώρευσης Καλώς ήλθατε, **accforeas** | Αποσύνδεση

[Διαχείριση](#) ▾

Ελεγχος Σώρευσης

ΑΦΜ Επιχείρησης



Αιτούμενο Ποσό

 Ελεγχος Σώρευσης
  Καθαρισμός Πεδίων

↑ Κορυφή σελίδας

Copyright © 2010 Υπουργείο Οικονομίας, Ανταγωνιστικότητας & Ναυτιλίας

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης - Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης

Αν το σύστημα επιτρέπει την εισαγωγή της συγκεκριμένης επένδυσης, ο φορέας θα πρέπει να εισάγει όλη την απαραίτητη πληροφορία μέσα στο σύστημα διαφορετικά δεν δύναται να χρηματοδοτήσει την επιχείρηση με το συγκεκριμένο ποσό.

➤ Βήμα 3: Λίστα προσκλήσεων

Ο φορέας ελέγχει αν έχουν περαστεί τα στοιχεία της προκήρυξης /πρόσκλησης μέσα στο σύστημα.

Σύστημα Σώρευσης Καλώς ήλθατε, **accforeas** | Αποσύνδεση

[Διαχείριση](#) ▾ [Επιστροφή](#) 🏠

Κριτήρια αναζήτησης Προσλήσεων

Κωδ. Πρόσκλησης Τίτλος Αρ. Πρωτοκόλλου Ημερ. Πρωτοκόλλου


Ημερ. Εναρξης Υποβολής Ημερ. Λήξης Υποβολής
 Ημερ. Εναρξης Επιλεξιμότητας Ημερ. Λήξης Επιλεξιμότητας

Κωδ. Πρόσκλησης	Τίτλος	Αρ. Πρωτοκόλλου	Ημ. Πρωτοκόλλου	Ημ. Εναρξης Υποβόλ	Ημ. Λήξης Υποβολής	Ημ. Εναρξης Επιλεξιμ	Ημ. Λήξης Επιλεξιμότητας
1111123	11119	1115	12/10/2010	21/10/2010	22/10/2010	23/10/2010	06/10/2010
1111	111	11	12/10/2010	21/10/2010	22/10/2010	23/10/2010	06/10/2010

↑ Κορυφή σελίδας

Copyright © 2010 Υπουργείο Οικονομίας, Ανταγωνιστικότητας & Ναυτιλίας

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης - Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης

Επιλέξτε Πρόσκληση και δεξί κλικ για ενέργειες



Αν έχουν περαστεί τότε του δύναται η δυνατότητα επίβλεψης/διόρθωσης.

➤ Βήμα 4: Εισαγωγή Πρόσκλησης

Αν δεν έχουν περαστεί τότε θα πρέπει να εισάγει τα στοιχεία της προκήρυξης/πρόσκλησης

Πρόσκληση



Επιχειρησιακή Περίοδος	<input type="text"/>	Επιχειρησιακό Πρόγραμμα	<input type="text"/>
Κωδικός Αξονα	<input type="text"/>	Κωδικός Φορέα Πρόσκλησης	1010400
Κωδικός Πρόσκλησης	<input type="text"/>	Είδος Αξιολόγησης	ΑΜΕΣΗ
Εναρξη Υποβολής	<input type="text"/>	Λήξη Υποβολής	<input type="text"/>
Περίοδος Υποβολής	<input type="text"/>	Περίοδος Αξιολόγησης	<input type="text"/>
Εναρξη Επιλεξιμότητας	<input type="text"/>	Λήξη Επιλεξιμότητας	<input type="text"/>
Ποσό	<input type="text"/>	Τίτλος	<input type="text"/>
Αριθμός Πρωτοκόλλου	<input type="text"/>	Ημερομηνία Πρωτοκόλλου	<input type="text"/>
Ημερομηνία Ολοκλήρωσης	<input type="text"/>	Καθεστώς	Καν.(ΕΚ) 800/2008 - Α
Αριθμός Κοινοποίησης	<input type="text"/>	Αριθμός Απόφασης	<input type="text"/>

 Προσθήκη Πρόσκλησης
  Ακύρωση

➤ **Βήμα 5: Λίστα επενδυτικών σχεδίων για την πρόσκληση**

Ο φορέας δύναται να δει την λίστα των επενδυτικών σχεδίων που έχουν υποβληθεί για μία πρόσκληση

Σύστημα Σώρευσης Καλώς ήλθατε, **accforeas** | Αποσύνδεση

 Διαχείριση  Επιστροφή

Κριτήρια αναζήτησης Προσλήσεων

Κωδ. Πρόσκλησης
 Τίτλος
 Αρ. Πρωτοκόλλου
 Ημερ. Πρωτοκόλλου

Ημερ. Εναρξης Υποβολής
 Ημερ. Λήξης Υποβολής
 Ημερ. Εναρξης Επιλεξιμότητας
 Ημερ. Λήξης Επιλεξιμότητας

Κωδ. Πρόσκλησης	Τίτλος	Αρ. Πρωτοκόλλου	Ημ. Πρωτοκόλλου	Ημ. Εναρξης Υποβ	Ημ. Λήξης Υποβολή	Ημ. Εναρξης Επιλεξιμ	Ημ. Λήξης Επιλεξιμ	Προ
1111123	11119	1115	12/10/2010	21/10/2010	27/10/2010	23/10/2010	06/10/2010	
1111	111	11	12/10/2010	21/10/2010			06/10/2010	

1 Κορυφή σελίδας

Copyright © 2010 Υπουργείο Οικονομίας, Ανταγωνιστικότητας & Ναυτιλίας

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης - Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης

Μπορεί να επενεργήσει σε αυτά τα επενδυτικά σχέδια (Επίβλεψη/Διόρθωση)

Σύστημα Σώρευσης Καλώς ήλθετε, accforeas! Αποσύνδεση

Διαχείριση
 Επιστροφή

Κριτήρια αναζήτησης Επενδυτικών Σχεδίων 20 αποτελέσματα ανά σελ.

Τεχν. Περιγραφή | Αιτ. Προϋπολογισμός | Επιχ. Προϋπολογισμός | Ενισχ. Προϋπολογισμός | Ποσό Επιχορήγησης

Περιγραφή	Αιτ. Προϋ/σμός	Επιχ. Προϋ/σμός	Ενισχ. Προϋ/σμός	Επιχορήγηση
techdescr13	10000.0	30000.0	20002.0	10008.0

- Επεξεργασία Επενδυτικού Σχεδίου
- Επισκόπηση Επενδυτικού Σχεδίου
- Προσθήκη Παραστατικού
- Λίστα Παραστατικών

↑ Κορυφή σελίδας

Copyright © 2010 Υπουργείο Οικονομίας, Ανταγωνιστικότητας & Ναυτίας
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης - Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης

➤ Βήμα 6: Προσθήκη Επενδυτικού Σχεδίου

Αν δεν υπάρχει το επενδυτικό σχέδιο ο Φορέας θα πρέπει να το εισάγει στο σύστημα.

Επενδυτικό Σχέδιο

Απούμενος Προϋπολογισμός	<input type="text"/>	Επιχορηγούμενος Προϋπολογισμός	<input type="text"/>
Ενισχυόμενος Προϋπολογισμός	<input type="text"/>	Δημόσια Επιχορήγηση	<input type="text"/>
Περιφέρεια Τόπου Εγκατάστασης	<input type="text"/>	Νομός Τόπου Εγκατάστασης	<input type="text"/>
Δήμος Τόπου Εγκατάστασης	<input type="text"/>	Δημοτικό Διαμέρισμα Τόπου Εγκατάστασης	<input type="text"/>
Οδός	<input type="text"/>	Αριθμός	<input type="text"/>
Ταχυδρομικός κώδικας	<input type="text"/>	Τύπος Ενίσχυσης	Μή επιστρεπτέα ενίσχυση
A.Φ.Μ Επιχείρησης	<input type="text"/>	Κωδικός Επένδυσης	<input type="text"/>
Σύντομη Τεχνική Περιγραφή			
<input style="width: 100%; height: 100%;" type="text"/>			

Προσθήκη Επενδυτικού Σχεδίου
 Ακύρωση



Κατά την εισαγωγή του επενδυτικού σχεδίου και κατά την εισαγωγή του ΑΦΜ της επιχείρησης που χρηματοδοτείται θα ελεγχθεί με βάση το ΑΦΜ της αν έχει εισαχθεί στο σύστημα. Αν έχει θα επιτραπεί να συνεχίσει με την εισαγωγή διαφορετικά ο φορέας θα πρέπει να εισάγει τα στοιχεία της στην σχετική φόρμα (βλέπε Βήμα 7). Στο τέλος της διαδικασίας ο φορέας θα λάβει έναν κωδικό σώρευσης για την συγκεκριμένη επιχείρηση τον οποίο και θα χρησιμοποιεί.

➤ Βήμα 7: Προσθήκη Επιχείρησης

Ο Φορέας εισάγει τα στοιχεία της επιχείρησης στο σύστημα

Προσθήκη Νέας Επιχείρησης

Επωνυμία Επιχείρησης	<input type="text"/>	Διακριτικός Τίτλος	<input type="text"/>
Κωδικός Κύριας Δραστηριότητας	<input type="text"/>	Τομέας	<input type="text"/>
ΑΦΜ	<input type="text"/>	ΔΟΥ	<input type="text"/>
Περιφέρεια Εδρας	<input type="text"/>	Νομός Εδρας	<input type="text"/>
Δήμος Εδρας	<input type="text"/>	Δημοτικό Διαμέρισμα Εδρας	<input type="text"/>
Αριθμός Απασχολουμένων στην Επιχείρηση	<input type="text"/>	Υφιστάμενη	ΝΑΙ <input type="button" value="v"/>
Τηλέφωνο	<input type="text"/>	Fax	<input type="text"/>
Οδός	<input type="text"/>	Αριθμός	<input type="text"/>
Ταχυδρομικός Κώδικας	<input type="text"/>	Ιστότοπος Εταιρείας	<input type="text"/>
Επώνυμο Εκπροσώπου Εταιρείας	<input type="text"/>	Όνομα Εκπροσώπου Εταιρείας	<input type="text"/>
Θέση Εκπροσώπου στην Εταιρεία	<input type="text"/>	ΑΦΜ Εκπροσώπου	<input type="text"/>
Αριθμός Δελτίου Ταυτότητας	<input type="text"/>	Κινητό Εκπροσώπου	<input type="text"/>
Νομική Μορφή	A.E <input type="button" value="v"/>	Είδος Βιβλίων	A' Κατηγορίας <input type="button" value="v"/>
Email	<input type="text"/>		

 Προσθήκη Επιχείρησης  Ακύρωση

Κεφάλαιο 8. Επίλογος

Στην ενότητα αυτή συνοψίζουμε τα αποτελέσματα τις διπλωματικής εργασίας και παραθέτουμε τις μελλοντικές επεκτάσεις που έχουν νόημα και ενδιαφέρον να υλοποιηθούν.

8.1. Σύνοψη και συμπεράσματα

Είδαμε λοιπόν ότι αναπτύχθηκε μια λογισμική εφαρμογή, η οποία σύμφωνα με τις οδηγίες της Ευρωπαϊκής Ένωσης επιτρέπει στους φορείς που χρηματοδοτούν τις επιχειρήσεις να ελέγξουν γρήγορα και εύκολα μια επιχείρηση πριν εγκρίνουν μια επένδυση. Επίσης αυτό το σύστημα αποτελεί μια βάση δεδομένων που συγκεντρώνει όλη την πληροφορία των κρατικών ενισχύσεων και μπορούν εύκολα να αντληθούν σε επίπεδο επιχείρησης αλλά και προκήρυξης.

Γενικά λοιπόν, όπως παρουσιάστηκε στα προηγούμενα κεφάλαια, το σύστημα μας αποτελεί μια ικανοποιητική λύση για τον σωρευτικό έλεγχο που υποχρεώνει η Ευρωπαϊκή Ένωση να γίνεται για τις κρατικές ενισχύσεις.

8.2. Μελλοντικές Επεκτάσεις

Οι επεκτάσεις και οι βελτιώσεις που μπορούν να γίνουν στο σύστημα, έτσι ώστε αυτό να γίνει ολοκληρωμένο και να μπορεί να ανταγωνιστεί άλλες υλοποιήσεις συστημάτων είναι πολλές και απαιτούν για την υλοποίησή τους αρκετές ώρες εργασίας και προσπάθειας. Εμείς, ωστόσο, αναφέρουμε στη συνέχεια μερικές από αυτές που τις θεωρούμε πιο σημαντικές.

- Σχετικά με το business της εφαρμογής μια μελλοντική επέκταση αφορά των έλεγχο σώρευσης σε επίπεδο μετόχου επιχείρησης. Επίσης θα πρέπει να προστεθούν εκτυπώσεις που θα μπορούν να καλέσουν οι φορείς και έντυπα που θα μπορούσαν να παρουσιάσουν επίσημα.
- Όσον αφορά το security καλό θα ήταν να υπάρχει ένας μηχανισμός encryption των δεδομένων που έρχονται από τη βάση, ώστε να μην μπορεί κάποιος να τα υποκλέψει.
- Μια άλλη μελλοντική επέκταση θα ήταν το WSDL αρχείο που περιγράφει το Web Service να γίνεται publish σε μια UDDI registry. Να ρωτάται δηλαδή ο χρήστης αν μετά το deployment της Web Service θέλει να την κάνει publish και σε μια UDDI registry.

Βιβλιογραφία

- Kent Ka lok Tong (2008) Developing Web Services with Apache Axis2
- Mingers, J. Combining IS Research Methods: Towards a Pluralist Methodology, Information Systems Research
- J. McGrath (1982) Dilemmatics: The Study of Research Choices and Dilemmas. In Judgement Calls in Research, J. McGrath, J. Martin, and R. Kulka (eds.), Sage Publications.
- Hancock B., Windridge K. and Ockleford E. (2007) An introduction to qualitative research, Trend RDSU.
- Miles, M. B. (1979) Qualitative data as an attractive nuisance: the problem of analysis, Administrative Science Quarterly 24, p.590-601
- Cavaye, A.L. (1996) Case study research: a multi-faceted approach for IS, Information Systems Journal 6, p.227-242
- Ιωσηφίδης, Θ. (2008) Ποιοτικές Μέθοδοι Έρευνας στις Κοινωνικές Επιστήμες, Εκδόσεις Κριτική ΑΕ
- Webster, Merrian (1985) The Merrian-Webster Dictionary, New York
- Seth Ladd with Darren Davison, Steven Devijver and Colin Yates (2006) Expert Spring MVC and Web Flow.
- Craig Walls and Ryan Breidenbach (2005) Spring in Action.
- Rod Johnson (2008) The Spring Framework - Reference Documentation.
- Κιλτένη Κ. (2008) Διαχείριση Αδειών Χρήσης Λογισμικού σε Περιβάλλον Πλέγματος
- Ζευγολάτης Ν.Σ. (2009) Μηχανή εκτέλεσης ροών εργασίας για συστήματα χαμηλών προδιαγραφών
- Αντζουλάτος Ν.Ρ. Ανάπτυξη Μεθοδολογίας για τη Σχεδίαση Διαδραστικών Web Εφαρμογών.
- Τσούλος Κ. (2005) Ολοκλήρωση Συστημάτων και Εφαρμογών με χρήση Web Services
- Κυριακούλιας Κ. – Ντελής Ε. (2006) Τεχνολογία Web Services
- Λαμού Κ. (2003) Δημοσιοποίηση πληροφοριών βάσεων δεδομένων με χρήση συνδυασμένων υπηρεσιών web.
- Μπασλής Κ. (2010) Μελέτη της χρήσης ajax τεχνολογιών για την ανάπτυξη web εφαρμογών σε περιβάλλον java.
- Nadir Weibel - Rudi Belotti (2002) Web Services Technologies
- Roy Thomas Fielding (2000) - Architectural Styles and the Design of Network - based Software Architectures
- IBM technical library (2008) RESTFUL web services: The basics
- Σπυράκης Π. (2011) Πρότυπη εφαρμογή για δημοσίευση και συλλογή περιεχομένου σε πραγματικό χρόνο στα κοινωνικά μέσα.

Ιστότοποι

- <http://www.w3.org/TR/xmlschema-0/>
- <http://www.w3.org/TR/xmlschema-1/>
- <http://www.w3.org/TR/ws-arch/>
- <http://www.w3.org/TR/soap/>
- ZK Developer s Guide (http://books.zkoss.org/wiki/ZK_Developer%27s_Guide)
- <http://www.softcov.com/el/database/xalan--achieved-through-a-relational-database-to.html>
- <http://static.springsource.org/spring/docs/2.0.x/reference/introduction.html>
- <http://axis.apache.org/axis2/java/core/docs/pojoguide.html>
- http://en.wikipedia.org/wiki/Main_Page
- <http://www.ceid.upatras.gr/courses/katanemhmena/ds2/index.php/pe2010-2011/62-spring>
- <http://www.docstoc.com/docs/28954952/USING-QUALITATIVE-METHODS-FOR-RESEARCH>
- http://www.scu.edu.au/schools/gcm/ar/arp/guide.html#a_g_intro
- <http://www.dmst.aueb.gr/louridas/lectures/dais/configuration/ar01s14.html>
- http://en.wikipedia.org/wiki/Representational_state_transfer
- <http://www.actionresearch.gr/el>

Κεφάλαιο 9. Ευρετήριο Όρων

Δ

ΔΑ – Διαχειριστική Αρχή

Η διαχείριση κάθε Επιχειρησιακού Προγράμματος του ΕΣΠΑ, ασκείται μέσω **Ειδικής Υπηρεσίας** (Διαχειριστική Αρχή), η οποία λειτουργεί ή συνιστάται για τους σκοπούς αυτούς. Η Διαχειριστική Αρχή είναι υπεύθυνη για τη διαχείριση του Επιχειρησιακού Προγράμματος σύμφωνα με το **Σύστημα Διαχείρισης και Ελέγχου** και την **αρχή της χρηστής δημοσιονομικής διαχείρισης**.

Διαχειριστική Αρχή του ΚΠΣ (ΚΔΑ)

Η διαχείριση του ΚΠΣ και η εκτέλεση της αποστολής και των καθηκόντων του άρθρου 2 του Ν. 2860/00 ασκούνται μέσω της Γενικής Διεύθυνσης Περιφερειακής Πολιτικής και Επενδύσεων του Υπουργείου Εθνικής Οικονομίας (Διαχειριστική Αρχή του ΚΠΣ), η οποία μετονομάζεται σε Γενική Διεύθυνση Αναπτυξιακού Προγραμματισμού, Περιφερειακής Πολιτικής και Δημοσίων Επενδύσεων.

Η ΚΔΑ έχει τη συνολική ευθύνη του προγραμματισμού και της εφαρμογής του ΚΠΣ. Ιδίως έχει ως σκοπό την εξασφάλιση της αποτελεσματικότητας και της νομιμότητας της διαχείρισης και της εφαρμογής του ΚΠΣ, καθώς και το συντονισμό του προγραμματισμού και της εφαρμογής των Επιχειρησιακών Προγραμμάτων και το συντονισμό των Διαχειριστικών Αρχών των Επιχειρησιακών Προγραμμάτων.

Ε

ΕΔΕΛ (Επιτροπή Δημοσιονομικού Ελέγχου)

Στο Υπουργείο Οικονομικών (Γενικό Λογιστήριο του Κράτους, ΓΛΚ) συνιστάται Επιτροπή Δημοσιονομικού Ελέγχου με σκοπό τη διενέργεια ελέγχων των Διαχειριστικών Αρχών, της Αρχής Πληρωμής, των Τελικών Δικαιούχων και πράξεων.

ΕΕ – Ευρωπαϊκή Ένωση

Η Ευρωπαϊκή Ένωση (ΕΕ) είναι μια οικογένεια ευρωπαϊκών δημοκρατικών χωρών που συνεργάζονται για τη βελτίωση του τρόπου ζωής των πολιτών τους και για τη δημιουργία ενός καλύτερου κόσμου. Η ΕΕ καθιέρωσε την ειρήνη και την ευημερία μεταξύ των πολιτών της, δημιούργησε ένα ενιαίο ευρωπαϊκό νόμισμα (το ευρώ) και μία «**ενιαία αγορά**» χωρίς σύνορα, όπου τα αγαθά, οι άνθρωποι, οι υπηρεσίες και τα κεφάλαια κυκλοφορούν ελεύθερα.

Η Ευρωπαϊκή Ένωση (ΕΕ) δημιουργήθηκε με τη **συνθήκη για την Ευρωπαϊκή Ένωση (Μαάστριχτ, 1992)**. Η Ένωση αποτελεί συγχρόνως πολιτικό πρόγραμμα και νομική οργάνωση.

Η επιτυχία της Ευρωπαϊκής Ένωσης οφείλεται σε μεγάλο βαθμό στον ασυνήθιστο τρόπο λειτουργίας της, αφού οι χώρες που αποτελούν την Ένωση (τα «κράτη - μέλη» της) συνεχίζουν να είναι ανεξάρτητα κυρίαρχα έθνη, αλλά συνενώνουν τις εθνικές κυριαρχίες τους ούτως ώστε να αποκτήσουν ισχύ και παγκόσμια επιρροή την οποία καμία δεν θα μπορούσε να έχει μόνη της. Ο **συγκερασμός των εθνικών κυριαρχιών** σημαίνει πρακτικά ότι τα κράτη - μέλη μεταβιβάζουν ορισμένες από τις εξουσίες λήψης αποφάσεων που διαθέτουν σε κοινά θεσμικά όργανα τα οποία έχουν συσταθεί από τα ίδια, ούτως ώστε οι αποφάσεις για συγκεκριμένα θέματα κοινού ενδιαφέροντος να μπορούν να λαμβάνονται δημοκρατικά σε ευρωπαϊκό επίπεδο.

ΕΠΑΕ

Επιχειρησιακό Πρόγραμμα ΕΣΠΑ για την Ανταγωνιστικότητα και Επιχειρηματικότητα (ΕΠΑΕ)

ΕΠΑΝ

Επιχειρησιακό Πρόγραμμα ΚΠΣ Ανταγωνιστικότητα

Επιχειρησιακό πρόγραμμα (ΕΠ)

Έγγραφο το οποίο καταρτίζεται από τα αρμόδια Υπουργεία, περιφέρειες και τους λοιπούς αρμόδιους φορείς, υποβάλλεται από το Υπουργείο Οικονομίας και Οικονομικών και εγκρίνεται από την Επιτροπή και το οποίο καθορίζει μια αναπτυξιακή στρατηγική με τη χρήση ενός συνεκτικού συνόλου προτεραιοτήτων, που θα επιτευχθεί με τη συνδρομή ενός Ταμείου ή, στην περίπτωση του στόχου «Σύγκλιση» και ειδικότερα για τα Επιχειρησιακά Προγράμματα «Ενίσχυση της Προσπελασιμότητας» και «Περιβάλλον - Αειφόρος Ανάπτυξη», με τη συνδρομή του Ταμείου Συνοχής και του ΕΤΠΑ.

ΕΣΠΑ - Εθνικό Στρατηγικό Πλαίσιο Αναφοράς

Το **ΕΣΠΑ 2007 - 2013** αποτελεί το έγγραφο αναφοράς για τον προγραμματισμό των **Ταμείων** της Ευρωπαϊκής Ένωσης σε εθνικό επίπεδο για την περίοδο 2007-2013. Εκπονήθηκε στο πλαίσιο της νέας στρατηγικής προσέγγισης για την **Πολιτική Συνοχής** της Ευρωπαϊκής Ένωσης, σύμφωνα με την οποία το ΕΣΠΑ «εξασφαλίζει ότι η συνδρομή από τα Ταμεία συμβαδίζει με τις κοινοτικές στρατηγικές κατευθυντήριες γραμμές για τη συνοχή και προσδιορίζει το σύνδεσμο μεταξύ των κοινοτικών προτεραιοτήτων αφενός και του εθνικού προγράμματος μεταρρυθμίσεων αφετέρου». Οι αποφάσεις του Ευρωπαϊκού Συμβουλίου του Δεκεμβρίου του 2005, με τις οποίες διασφαλίσθηκαν έως το 2013 οι **πόροι της Ευρωπαϊκής Πολιτικής Συνοχής** για τη χώρα, οι **νέοι Κανονισμοί των Ταμείων** της ΕΕ, καθώς και οι **Στρατηγικές Κατευθυντήριες Γραμμές** για την Πολιτική Συνοχής, αποτέλεσαν το πλαίσιο στο οποίο βασίστηκαν, μεταξύ των άλλων, οι εθνικές αρχές προκειμένου να

προσεγγίσουν τις βασικές παραμέτρους του αναπτυξιακού προγραμματισμού και να καταρτίσουν το ΕΣΠΑ.

ΕΥΔ

Είναι **Ειδικές Υπηρεσίες του πρώην Υπουργείου Δημοσίων Έργων και νυν ΥΠΕΧΩΔΕ**, των οποίων η σύσταση προβλεπόταν από τον Ν. 679/1977 για την εκτέλεση σημαντικών έργων. Οι διαδικασίες σύστασης, η έδρα, η διάρκεια λειτουργίας, η οργάνωση, η στελέχωση και οι αρμοδιότητες των ΕΥΔΕ ορίζονται από ειδικά σχετικά Προεδρικά Διατάγματα, σύμφωνα με τον Ν. 679/1977, όπως τροποποιήθηκε με το άρθρο 23 του Ν. 1418/1984.

Κ

Καθεστώς De Minimis Γ' ΚΠΣ - Κανόνας De Minimis στις κρατικές ενισχύσεις του Γ' ΚΠΣ Ο κανόνας de minimis εισήχθη το 1992 και αφορά την πολιτική της Επιτροπής για τις κρατικές ενισχύσεις των μικρομεσαίων επιχειρήσεων. Συγκεκριμένα, ενισχύσεις μεγαλύτερες των 100.000 ΕΥΡΩ μέσα σε χρονικό διάστημα μιας τριετίας ανά επιχείρηση πρέπει να ανακοινώνονται στην Επιτροπή για λόγους ελέγχου της τήρησης των κανόνων του ανταγωνισμού.

Καθεστώς De Minimis ΕΣΠΑ – Κανόνας De Minimis στις κρατικές ενισχύσεις του ΕΣΠΑ

Ο κανόνας de minimis (ενισχύσεις ήσσονος σημασίας) εισήχθη το 1992, τροποποιήθηκε το 2006 και αφορά την πολιτική της Επιτροπής για τις κρατικές ενισχύσεις των μικρομεσαίων επιχειρήσεων, καθώς και των επιχειρήσεων που δραστηριοποιούνται **στους** τομείς των μεταφορών **και της** μεταποίησης **και** εμπορίας γεωργικών προϊόντων. Συγκεκριμένα, ενισχύσεις μεγαλύτερες των 200.000 Ευρώ και των 100.000 Ευρώ για τον τομέα των οδικών μεταφορών μέσα σε χρονικό διάστημα τριών ετών ανά επιχείρηση, πρέπει να ανακοινώνονται στην Επιτροπή για λόγους ελέγχου της τήρησης των κανόνων του ανταγωνισμού.

Κανονισμός 69/2001 (De minimis)

Κανονισμός (ΕΚ) αριθ. 69/2001 της Επιτροπής της 12ης Ιανουαρίου 2001 για την εφαρμογή των άρθρων 87 και 88 της συνθήκης ΕΚ στις ενισχύσεις ήσσονος σημασίας. Το πλήρες κείμενο είναι διαθέσιμο στην ιστοσελίδα <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32001R0069:EL:HTML>

Ο κανονισμός αυτός αναθεωρείται και συμπληρώνεται με τον Κανονισμό 1998/2006..

Κανονισμός 1998/2006 (De minimis)

Κανονισμός (ΕΚ) αριθ. [1998/2006](#) της Επιτροπής, της 15ης Δεκεμβρίου 2006, για την εφαρμογή των άρθρων 87 και 88 της συνθήκης στις ενισχύσεις ήσσονος σημασίας (de minimis). Το πλήρες κείμενο είναι διαθέσιμο στην ιστοσελίδα

<http://eur->

lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:379:0005:01:EL:HTML

Κρατική Ενίσχυση στα πλαίσια της ΕΕ

Κάθε μέτρο που ενδυναμώνει οικονομικά συγκεκριμένες επιχειρήσεις ή κλάδους παραγωγής στην εσωτερική αγορά της Ευρωπαϊκής Ένωσης (ΕΕ) μέσω κρατικών πόρων. Σε γενικές γραμμές, η Κοινότητα αντιμετωπίζει κατ' αρχήν αρνητικά τις κρατικές ενισχύσεις, η χορήγηση των οποίων εμποδίζει την ομαλή λειτουργία μιας πραγματικά ελεύθερης και ενιαίας αγοράς αγαθών και υπηρεσιών που αποτελεί θεμελιακό λίθο της ευρωπαϊκής ολοκλήρωσης. Κατ' εξαίρεση, όμως, της πιο πάνω αρχής, η ΕΕ επιτρέπει τις κρατικές ενισχύσεις, που είτε έχουν αμελητέες συνέπειες στον ανταγωνισμό, είτε είναι αναγκαίες για την προώθηση γενικότερων πολιτικών και στόχων της ΕΕ οι οποίοι συχνά δεν επιτυγχάνονται χωρίς την κρατική παρέμβαση στην αγορά. Για τον πιο πάνω σκοπό, η Ευρωπαϊκή Επιτροπή έχει θεσπίσει συγκεκριμένους κανόνες, με τους οποίους ρυθμίζονται οι όροι και προϋποθέσεις για την παραχώρηση κρατικών ενισχύσεων.

ΚτΠ: Για την προώθηση της Κοινωνίας της Πληροφορίας (ΚτΠ) στην Ελλάδα με τρόπο συνεκτικό και ολοκληρωμένο, έχει ενταχθεί ένα ξεχωριστό Επιχειρησιακό Πρόγραμμα για την ΚτΠ, το Ε.Π. "ΚτΠ", στο πλαίσιο του 3ου ΚΠΣ, την περίοδο 2000 - 2006. διαρθρωτικά ταμεία

O

ΟΠΣ – Ολοκληρωμένο Πληροφοριακό Σύστημα

Το πληροφοριακό σύστημα του Υπουργείου Οικονομίας και Οικονομικών, όπου καταχωρούνται τα δεδομένα που αφορούν τα προγράμματα, τις επενδύσεις του επενδυτικού νόμου και στο Πρόγραμμα Δημοσίων Επενδύσεων. Αποτελεί ένα **σύστημα πληροφόρησης και διαχείρισης**, η χρήση του οποίου επιβάλλεται από την απαίτηση της Ευρωπαϊκής Επιτροπής **για ομοιόμορφη ηλεκτρονική ανταλλαγή δεδομένων** με τα κράτη - μέλη και από την ανάγκη ύπαρξης ενός ενιαίου Πληροφοριακού Συστήματος

T

TEMPME - Ταμείο Εγγυοδοσίας Μικρών και Πολύ Μικρών Επιχειρήσεων

<http://www.tempme.gr>/Το Ταμείο Εγγυοδοσίας Μικρών και Πολύ Μικρών Επιχειρήσεων (TEMPME Α.Ε), είναι μια ανώνυμη εταιρεία, της οποίας το μετοχικό κεφάλαιο φτάνει το ποσό των €1.712.885.700,00. Η TEMPME ΑΕ σχεδιάζει τις δράσεις της και προσαρμόζει τις λειτουργικές της δομές, έτσι ώστε να προάγει την καινοτομία και την αποτελεσματικότητα στο πεδίο της επιχειρηματικότητας, έχοντας

πάντα ως γνώμονα την επίτευξη των στόχων της πολιτικής που χαράζει το Υπουργείο Ανάπτυξης. Παράλληλα, εκμεταλλεύεται τις ευκαιρίες που παρουσιάζονται στο πλαίσιο των προγραμμάτων της Ευρωπαϊκής Ένωσης δημιουργώντας δράσεις οι οποίες έχουν άμεσα αποτελέσματα για την πραγματική οικονομία της χώρας. Βασικός σκοπός του είναι η διευκόλυνση της πρόσβασης των μικρών και πολύ μικρών επιχειρήσεων στην χρηματοπιστωτική και χρηματοδοτική αγορά, λειτουργώντας ως ένας κρίκος μεταξύ της μικρής επιχείρησης και της τράπεζας.

Κεφάλαιο 10. Παράρτημα

10.1. AccumulationService.wsdl

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://services.accws.mou/"
  xmlns:par="http://parameters.services.accws.mou/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"
  name="AccumulationService"
  targetNamespace="http://services.accws.mou/">
  <wsp:Policy wsu:Id="USNToken"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
    <wsp:ExactlyOne>
      <wsp:All>
        <sp:SupportingTokens xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200702">
          <wsp:Policy>
            <sp:UsernameToken
              sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200702/IncludeToken/AlwaysToRecipient">
              </sp:UsernameToken>
            </wsp:Policy>
          </sp:SupportingTokens>
        </wsp:All>
      </wsp:ExactlyOne>
    </wsp:Policy>
  </wsdl:types>
  <xsd:schema targetNamespace="http://services.accws.mou/"
    <xsd:import namespace="http://parameters.services.accws.mou/"
    schemaLocation="accquery.xsd"/>
    <xsd:import namespace="http://parameters.services.accws.mou/"
    schemaLocation="insertiplan.xsd"/>
  </xsd:schema>
</wsdl:types>
<wsdl:message name="checkAccumulationRequest">
  <wsdl:part element="par:checkAccumulationRequest" name="parameters"/>
</wsdl:message>
<wsdl:message name="checkAccumulationResponse">
  <wsdl:part element="par:checkAccumulationResponse" name="parameters" />
</wsdl:message>
<wsdl:message name="insertIPlanRequest">
  <wsdl:part element="par:insertIPlanRequest" name="parameters"/>
</wsdl:message>
<wsdl:message name="insertIPlanResponse">
  <wsdl:part element="par:insertIPlanResponse" name="parameters" />
</wsdl:message>
<wsdl:portType name="AccumulationService">
  <wsdl:operation name="checkAccumulation">
    <wsdl:input message="tns:checkAccumulationRequest"/>
    <wsdl:output message="tns:checkAccumulationResponse"/>
  </wsdl:operation>
```

```

    <wsdl:operation name="insertIPlan">
      <wsdl:input message="tns:insertIPlanRequest"/>
      <wsdl:output message="tns:insertIPlanResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="AccumulationServiceSOAP" type="tns:AccumulationService">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsp:PolicyReference URI="#USNToken" wsdl:required="true"/>
    <wsdl:operation name="checkAccumulation">
      <soap:operation soapAction="checkAccumulation"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="insertIPlan">
      <soap:operation soapAction="insertIPlan"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="AccumulationService">
    <wsdl:documentation>Accumulation Web Service</wsdl:documentation>
    <wsdl:port binding="tns:AccumulationServiceSOAP" name="AccumulationServiceSOAP">
      <soap:address location="http://localhost:8081/accws/services/AccumulationService"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

10.2. InsertIPlan.xsd

```

<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://parameters.services.accws.mou/"
  targetNamespace="http://parameters.services.accws.mou/"
  elementFormDefault="qualified"
  xml:lang="gr">
  <xsd:complexType name="CompanyInfoType">
    <xsd:annotation>
      <xsd:documentation>Company Data</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="ent_name" type="xsd:string" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>Company
Name</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="ent_title" type="xsd:string" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>Company
Title</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```

```

        <xsd:element name="ent_kad" type="xsd:string" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company
KAD</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="ent_afm" type="xsd:string" minOccurs="1">
            <xsd:annotation>
                <xsd:documentation>Company
AFM</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="ent_doy_kodikos" type="xsd:string"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company
DOY</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="ent_tomeas" type="xsd:string" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company
Tomeas</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="ent_ypo_systash" type="xsd:integer"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company
Systash</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="ent_nomikh_morfh" type="xsd:integer"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company's Nomikh
Morfh</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="ent_employees_num" type="xsd:double"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company's Employees
Num</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="ent_xor_kodikos" type="xsd:string" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company's Kodikos
Xoras</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="ent_per_kodikos" type="xsd:string" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company's Kodikos
Perifereias</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="ent_nom_kodikos" type="xsd:string"
minOccurs="0">

```

```

        <xsd:annotation>
            <xsd:documentation>Nomos
Kodikos</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="ent_dhm_kodikos" type="xsd:string"
minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Company's Kodikos
Dhmos</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="ent_dd_kodikos" type="xsd:string" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Company's Kodikos
Dhm.Diam.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="ent_odos" type="xsd:string" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Company's
Odos</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="ent_arithmos" type="xsd:string" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Company's
Arithmos</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="ent_tk" type="xsd:string" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Company's
TK</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="ent_tel" type="xsd:string" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Company's Phone
Num</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="ent_fax" type="xsd:string" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Company's
Fax</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="ent_email" type="xsd:string" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Company's
Email</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="ent_website" type="xsd:string" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Company's
Website</xsd:documentation>
        </xsd:annotation>
    </xsd:element>

```

```

        <xsd:element name="ent_surname_ekpr" type="xsd:string"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company's Surname
Ekprosopou</xsd:documentation>
            </xsd:annotation>
        </xsd:element>

        <xsd:element name="ent_name_ekpr" type="xsd:string" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company's Name
Ekprosopou</xsd:documentation>
            </xsd:annotation>
        </xsd:element>

        <xsd:element name="ent_thesh_ekpr" type="xsd:string" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company's Thesh
Ekprosopou</xsd:documentation>
            </xsd:annotation>
        </xsd:element>

        <xsd:element name="ent_afm_ekpr" type="xsd:string" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company's AFM
Ekprosopou</xsd:documentation>
            </xsd:annotation>
        </xsd:element>

        <xsd:element name="ent_adt_ekpr" type="xsd:string" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company's ADT
Ekprosopou</xsd:documentation>
            </xsd:annotation>
        </xsd:element>

        <xsd:element name="ent_mob_ekpr" type="xsd:string" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company's Mob
Ekprosopou</xsd:documentation>
            </xsd:annotation>
        </xsd:element>

        <xsd:element name="ent_account_books" type="xsd:integer"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Company's Account
Books</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CompanyType">
    <xsd:annotation>
        <xsd:documentation>A company</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="CompanyInfo" type="CompanyInfoType"/>

```



```

        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="IplanInfoType">
        <xsd:annotation>
            <xsd:documentation>Iplan Data</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="prj_ent_afm" type="xsd:string" minOccurs="1">
                <xsd:annotation>
                    <xsd:documentation>Company's
AFM</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="prj_ait_proyp" type="xsd:double" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Iplan's
ait_proyp</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="prj_epix_proyp" type="xsd:double" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Iplan's
epix_proyp</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="prj_enisx_proyp" type="xsd:double"
minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Iplan's
enisx_proyp</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="prj_poso_epix" type="xsd:double" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>prj_poso_epix</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="prj_odos" type="xsd:string" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Iplan's Odos</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="prj_arithmos" type="xsd:string" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Iplan's
Arithmos</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="prj_tk" type="xsd:string" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Iplan's TK</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="prj_tech_descr" type="xsd:string" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Iplan's Tech
Description</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>

```

```

</xsd:element>
<xsd:element name="prj_prop_id" type="xsd:integer" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>Iplan's
prop_id</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="prj_kodikos_forea" type="xsd:integer"
minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>Iplan's Kodikos
Forea</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="prj_end_date" type="xsd:date" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>Iplan's End
Date</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="prj_approval_date" type="xsd:date"
minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>Iplan's Approval
Date</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="finalize_flag" type="xsd:integer" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>Iplan's
completion</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="prj_kathestos_kodikos" type="xsd:integer"
minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>Iplan's
Kathestos</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="prj_it_code" type="xsd:string" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>Iplan's IT
code</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="company" type="CompanyType" minOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Company
Info</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="IplanXoroth" type="IPlanXorothetiseis"
minOccurs="1">
  <xsd:annotation>
    <xsd:documentation>Iplan Xorothethseis
List</xsd:documentation>
  </xsd:annotation>

```

```

        </xsd:element>
        <xsd:element name="IplanEnisx" type="IplanEnisxyseis"
minOccurs="1">
            <xsd:annotation>
                <xsd:documentation>Iplan Typoi Enisxyshs
List</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="IplanType">
    <xsd:annotation>
        <xsd:documentation>A project plan</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="IplanInfo" type="IplanInfoType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ProposalInfoType">
    <xsd:annotation>
        <xsd:documentation>Proposal Data</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="prop_kps_kodikos" type="xsd:integer"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's KPS
Kodikos</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_ep_kodikos" type="xsd:string"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's EP
Kodikos</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_ypoprogr_kodikos" type="xsd:string"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's Aksonas
Kodikos</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_foreas_kodikos" type="xsd:integer"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's Foreas
Kodikos</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_kodikos_prosklhshs" type="xsd:string"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's
Kodikos</xsd:documentation>
            </xsd:annotation>
        </xsd:element>

```

```

        <xsd:element name="prop_eidos_aksiologshs" type="xsd:integer"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's Eidos
Aksiologshs</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_enarksh_ypobolhs" type="xsd:date"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's Enarksh
Ypobolhs</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_lhksh_ypobolhs" type="xsd:date"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's Lhksh
Ypobolhs</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_periodos_ypobolhs" type="xsd:integer"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's Periodos
Ypobolhs</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_periodos_aksiologshs" type="xsd:integer"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's Periodos
Aksiologshs</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_lhksh_epileksimothtas" type="xsd:date"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's Lhksh
Epileksimothtas</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_titlos" type="xsd:string" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's
Title</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_poso" type="xsd:double" minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's
Poso</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_ar_protokol" type="xsd:string"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's Arithmos
Protokoloy</xsd:documentation>
            </xsd:annotation>

```

```

        </xsd:element>
        <xsd:element name="prop_hmerom_protokol" type="xsd:date"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's Hmerom
Protokol</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_hmeromhnia_oloklhroshs" type="xsd:date"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's Hmerom
Oloklhroshs</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_kathestos_kodikos" type="xsd:integer"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's
Kathestos</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_enarksh_epileksimothtas" type="xsd:date"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's Enarksh
Epileksimothtas</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_kath_ar_koinop" type="xsd:string"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's
kath_ar_koinop</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="prop_kath_ar_apofashs" type="xsd:string"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Proposal's
kath_ar_apofashs</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="Iplan" type="IplanType" minOccurs="1">
            <xsd:annotation>
                <xsd:documentation>Iplan Info</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="IplanXorothetiseis">
    <xsd:sequence>
        <xsd:element name="Xorothethsh" type="IplanXorothInfo"
maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="IplanXorothInfo">
    <xsd:annotation>
        <xsd:documentation>Χωροθέτηση Επενδυτικού
Σχεδίου</xsd:documentation>

```

```

        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="prj_xor_kodikos" type="xsd:string" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Iplan's Kodikos
Xoras</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="prj_per_kodikos" type="xsd:string" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Iplan's Kodikos
Perifereias</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="prj_nom_kodikos" type="xsd:string"
minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Iplan's Kodikos
Nomou</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="prj_dhm_kodikos" type="xsd:string"
minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Iplan's Kodikos
Dhmoy</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="prj_dd_kodikos" type="xsd:string" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Iplan's Kodikos
Dhm.Diam.</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="IPlanEnisxyseis">
        <xsd:sequence>
            <xsd:element name="TypeEnisxyshs" type="IPlanEnisxisi"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="IPlanEnisxisi">
        <xsd:annotation>
            <xsd:documentation>Τύποι Ενίσχυσης Επενδυτικού
Σχεδίου</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="prj_type_enisx" type="xsd:string" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Iplan's Type
Enisxisis</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ProposalType">
        <xsd:annotation>
            <xsd:documentation>A proposal</xsd:documentation>

```

```

        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="ProposalInfo" type="ProposalInfoType"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="insertIPlanRequest">
        <xsd:complexType>
            <xsd:annotation>
                <xsd:documentation>Wrapper WS request</xsd:documentation>
            </xsd:annotation>
            <xsd:sequence>
                <xsd:element name="iplanrequest" type="ProposalType"/>
                <xsd:element name="errorCode" type="xsd:integer"/>
                <xsd:element name="errorDescription" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:complexType name="IPlanResponseType">
        <xsd:annotation>
            <xsd:documentation>A iplan response</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="poso" type="xsd:double"/>
            <xsd:element name="accumulationId" type="xsd:integer"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="insertIPlanResponse">
        <xsd:complexType>
            <xsd:annotation>
                <xsd:documentation>Wrapper WS
response</xsd:documentation>
            </xsd:annotation>
            <xsd:sequence>
                <xsd:element name="iplanresponse"
type="IPlanResponseType"/>
                <xsd:element name="errorCode" type="xsd:integer"/>
                <xsd:element name="errorDescription" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>

```

10.3. Request CheckAccumulation.xml

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:par="http://parameters.services.accws.mou/">
    <soapenv:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" soapenv:mustUnderstand="1">
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wsu:Id="USRToken">
                <wsse:Username>accforeas</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-
token-profile-1.0#PasswordText">accforeas</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </soapenv:Header>
    <soapenv:Body>
        <par:checkAccumulationRequest>

```

```

    <par:afm>999467638</par:afm>
    <par:poso>10000</par:poso>
  </par:checkAccumulationRequest>
</soapenv:Body>
</soapenv:Envelope>

```

10.4. Request InsertIPlan.xml

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:par="http://parameters.services.accws.mou/">
  <soapenv:Header>
    <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
soapenv:mustUnderstand="1">
      <wsse:UsernameToken
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="USRToken">
        <wsse:Username>accforeas</wsse:Username>
        <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">accforeas</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <par:insertIPlanRequest>
      <par:iplanrequest>
        <par:ProposalInfo>
          <!--Optional:-->
          <par:prop_kps_kodikos>4</par:prop_kps_kodikos>
          <!--Optional:-->
          <par:prop_ep_kodikos>09</par:prop_ep_kodikos>
          <!--Optional:-->
          <par:prop_ypopogr_kodikos>04</par:prop_ypopogr_kodikos>
          <!--Optional:-->
          <par:prop_foreas_kodikos>1090216</par:prop_foreas_kodikos>
          <!--Optional:-->
          <par:prop_kodikos_prosklhshs>EY01Δ</par:prop_kodikos_prosklhshs>
          <!--Optional:-->
          <par:prop_eidos_aksiologhshs>1</par:prop_eidos_aksiologhshs>
          <!--Optional:-->
          <par:prop_enarksh_ypobolhs>2009-09-30</par:prop_enarksh_ypobolhs>
          <!--Optional:-->
          <par:prop_lhksh_ypobolhs>2009-12-31</par:prop_lhksh_ypobolhs>
          <!--Optional:-->
          <par:prop_periodos_ypobolhs>3</par:prop_periodos_ypobolhs>
          <!--Optional:-->
          <par:prop_periodos_aksiologhshs>4</par:prop_periodos_aksiologhshs>
          <!--Optional:-->
          <par:prop_lhksh_epileksimothtas>2009-12-31</par:prop_lhksh_epileksimothtas>
          <!--Optional:-->
          <par:prop_titlos>ΑΠΤΙΟΠΙΟΣΚΛΗΡΩΤΙΚΕΣ ΔΟΜΕΣ</par:prop_titlos> <!--Optional:-->
          <par:prop_poso>15000000</par:prop_poso>
          <!--Optional:-->
          <par:prop_ar_protokol>3452/EYΣ2323</par:prop_ar_protokol>
          <!--Optional:-->
          <par:prop_hmerom_protokol>2009-08-25</par:prop_hmerom_protokol>
          <!--Optional:-->
          <par:prop_hmeromhnia_oloklhroshs>2012-12-31</par:prop_hmeromhnia_oloklhroshs>

```



```

<!--Optional:-->
<par:prop_kathestos_kodikos>8</par:prop_kathestos_kodikos>
<!--Optional:-->
<par:prop_enarksh_epileksimothtas>2009-09-01</par:prop_enarksh_epileksimothtas>
<!--Optional:-->
<par:prop_kath_ar_koinop>N306/2008</par:prop_kath_ar_koinop>
<!--Optional:-->
<par:prop_kath_ar_apofashs>EL7954389</par:prop_kath_ar_apofashs>
<par:Iplan>
<par:IplanInfo>
<par:prj_ent_afm>999467638</par:prj_ent_afm>
<!--Optional:-->
<par:prj_ait_proyp>10000</par:prj_ait_proyp>
<!--Optional:-->
<par:prj_epix_proyp>9000</par:prj_epix_proyp>
<!--Optional:-->
<par:prj_enisx_proyp>9000</par:prj_enisx_proyp>
<!--Optional:-->
<par:prj_poso_epix>4500</par:prj_poso_epix>
<!--Optional:-->
<par:prj_odos>ΚΑΛΛΗΜΑΝΕΩΝ 18</par:prj_odos> <!--Optional:-->
<par:prj_arithmos>18</par:prj_arithmos>
<!--Optional:-->
<par:prj_tk>11045</par:prj_tk>
<!--Optional:-->
<par:prj_tech_descr>THIS IS A TEST VALUE</par:prj_tech_descr> <!--Optional:-->
<par:prj_prop_id></par:prj_prop_id>
<!--Optional:-->
<par:prj_kodikos_forea>1090216</par:prj_kodikos_forea>
<!--Optional:-->
<par:prj_end_date>2011-10-12</par:prj_end_date>
<!--Optional:-->
<par:prj_approval_date>2010-12-31</par:prj_approval_date>
<!--Optional:-->
<par:finalize_flag>0</par:finalize_flag>
<!--Optional:-->
<par:prj_kathestos_kodikos>8</par:prj_kathestos_kodikos>
<!--Optional:-->
<par:prj_it_code>EA-00024</par:prj_it_code>
<par:company>
<par:CompanyInfo>
<!--Optional:-->
<par:ent_name>ΑΦΟΙ ΙΩΑΝ. ΛΕΤΣΙΟΥ Ο.Ε.</par:ent_name> <!--Optional:-->
<par:ent_title>JOOLS</par:ent_title>
<!--Optional:-->
<par:ent_kad>46.48.10.08</par:ent_kad>
<par:ent_afm>999467638</par:ent_afm>
<!--Optional:-->
<par:ent_doy_kodikos></par:ent_doy_kodikos>
<!--Optional:-->
<par:ent_tomeas>2</par:ent_tomeas>
<!--Optional:-->
<par:ent_ypo_systash>1</par:ent_ypo_systash>
<!--Optional:-->
<par:ent_nomikh_morph>3</par:ent_nomikh_morph>
<!--Optional:-->
<par:ent_employees_num>100</par:ent_employees_num>
<!--Optional:-->
<par:ent_xor_kodikos>EL</par:ent_xor_kodikos>
<!--Optional:-->

```

<par:ent_per_kodikos>01</par:ent_per_kodikos>
<!--Optional:-->
<par:ent_nom_kodikos>08</par:ent_nom_kodikos>
<!--Optional:-->
<par:ent_dhm_kodikos>001</par:ent_dhm_kodikos>
<!--Optional:-->
<par:ent_dd_kodikos>GR114002</par:ent_dd_kodikos>
<!--Optional:-->
<par:ent_odos>XABPIOY</par:ent_odos>
<!--Optional:-->
<par:ent_arithmos>9</par:ent_arithmos>
<!--Optional:-->
<par:ent_tk>10562</par:ent_tk>
<!--Optional:-->
<par:ent_tel>210 3250215</par:ent_tel>
<!--Optional:-->
<par:ent_fax>210 3250485</par:ent_fax>
<!--Optional:-->
<par:ent_email>INFO@JOOLSILVER.GR</par:ent_email>
<!--Optional:-->
<par:ent_website>WWW.JOOLSILVER.GR</par:ent_website>
<!--Optional:-->
<par:ent_surname_ekpr>ΛΕΤΣΙΟΣ</par:ent_surname_ekpr>
<!--Optional:-->
<par:ent_name_ekpr>ΗΛΙΑΣ</par:ent_name_ekpr>
<!--Optional:-->
<par:ent_thesh_ekpr>Νόμιμος Εκπρόσωπος</par:ent_thesh_ekpr> <!--Optional:-->
<par:ent_afm_ekpr>069000886</par:ent_afm_ekpr>
<!--Optional:-->
<par:ent_adt_ekpr>ΑΕ909090</par:ent_adt_ekpr>
<!--Optional:-->
<par:ent_mob_ekpr>6976997906</par:ent_mob_ekpr>
<!--Optional:-->
<par:ent_account_books>2</par:ent_account_books>
</par:CompanyInfo>
</par:company>
<par:IplanXoroth>
<!--1 or more repetitions:-->
<par:Xorothethsh>
<!--Optional:-->
<par:prj_xor_kodikos>EL</par:prj_xor_kodikos>
<!--Optional:-->
<par:prj_per_kodikos>04</par:prj_per_kodikos>
<!--Optional:-->
<par:prj_nom_kodikos>04</par:prj_nom_kodikos>
<!--Optional:-->
<par:prj_dhm_kodikos>007</par:prj_dhm_kodikos>
<!--Optional:-->
<par:prj_dd_kodikos></par:prj_dd_kodikos>
</par:Xorothethsh>
</par:IplanXoroth>
<par:IplanEnisx>
<!--1 or more repetitions:-->
<par:TypeEnisxyshs>
<!--Optional:-->
<par:prj_type_enisx>01</par:prj_type_enisx>
</par:TypeEnisxyshs>
</par:IplanEnisx>
</par:IplanInfo>
</par:Iplan>

```

</par:ProposalInfo>
</par:iplanrequest>
<par:errorCode></par:errorCode>
<par:errorDescription></par:errorDescription>
</par:insertIPlanRequest>
</soapenv:Body>
</soapenv:Envelope>

```

10.5. CompanyDaoImpl.java

```

//-----
package mou.core.persistence.model;
//-----
import java.io.Serializable;
import java.util.LinkedList;
import java.util.List;
//-----
public class CompanyVO implements Serializable {
    //-----
    private static final long serialVersionUID = 3335237854285610763L;
    //-----
    private long ent_id;
    private String ent_name;
    private String ent_title;
    private String ent_kad;
    private String ent_afm;
    private String ent_doy_descr;
    private String ent_tomeas;
    private int ent_ypo_systash;
    private int ent_nomikh_morph;
    private int ent_account_books;
    private double ent_employees_num;
    private String ent_xor_kodikos;
    private String ent_per_kodikos;
    private String ent_nom_kodikos;
    private String ent_dhm_kodikos;
    private String ent_dd_kodikos;
    private String ent_odos;
    private String ent_arithmos;
    private String ent_tk;
    private String ent_tel;
    private String ent_fax;
    private String ent_email;
    private String ent_website;
    private String ent_surname_ekpr;
    private String ent_name_ekpr;
    private String ent_thesh_ekpr;
    private String ent_afm_ekpr;
    private String ent_adt_ekpr;
    private String ent_mob_ekpr;
    //-----
    public CompanyVO() {
    }
    //-----
    public CompanyVO(long ent_id) {
        this.ent_id = ent_id;
    }
    //-----
    public long getEnt_id() {
        return ent_id;
    }
}

```

```

}
//-----
public void setEnt_id(long entId) {
    ent_id = entId;
}
//-----
public String getEnt_name() {
    return ent_name;
}
//-----
public void setEnt_name(String entName) {
    ent_name = entName;
}
//-----
public String getEnt_title() {
    return ent_title;
}
//-----
public void setEnt_title(String entTitle) {
    ent_title = entTitle;
}
//-----
public String getEnt_kad() {
    return ent_kad;
}
//-----
public void setEnt_kad(String entKad) {
    ent_kad = entKad;
}
//-----
public String getEnt_afm() {
    return ent_afm;
}
//-----
public void setEnt_afm(String entAfm) {
    ent_afm = entAfm;
}
//-----
public String getEnt_doy_descr() {
    return ent_doy_descr;
}
//-----
public void setEnt_doy_descr(String ent_doy_descr) {
    this.ent_doy_descr = ent_doy_descr;
}
//-----
public String getEnt_tomeas() {
    return ent_tomeas;
}
//-----
public void setEnt_tomeas(String entTomeas) {
    ent_tomeas = entTomeas;
}
//-----
public int getEnt_ypo_systash() {
    return ent_ypo_systash;
}
//-----
public void setEnt_ypo_systash(int entYpoSystash) {
    ent_ypo_systash = entYpoSystash;
}

```

```

}
//-----
public int getEnt_nomikh_morph() {
    return ent_nomikh_morph;
}
//-----
public void setEnt_nomikh_morph(int entNomikhMorph) {
    ent_nomikh_morph = entNomikhMorph;
}
//-----
public double getEnt_employees_num() {
    return ent_employees_num;
}
//-----
public void setEnt_employees_num(double entEmployeesNum) {
    ent_employees_num = entEmployeesNum;
}
//-----
public String getEnt_xor_kodikos() {
    return ent_xor_kodikos;
}
//-----
public void setEnt_xor_kodikos(String entXorKodikos) {
    ent_xor_kodikos = entXorKodikos;
}
//-----
public String getEnt_per_kodikos() {
    return ent_per_kodikos;
}
//-----
public void setEnt_per_kodikos(String entPerKodikos) {
    ent_per_kodikos = entPerKodikos;
}
//-----
public String getEnt_dhm_kodikos() {
    return ent_dhm_kodikos;
}
//-----
public void setEnt_dhm_kodikos(String entDhmKodikos) {
    ent_dhm_kodikos = entDhmKodikos;
}
//-----
public String getEnt_dd_kodikos() {
    return ent_dd_kodikos;
}
//-----
public void setEnt_dd_kodikos(String entDdKodikos) {
    ent_dd_kodikos = entDdKodikos;
}
//-----
public String getEnt_odos() {
    return ent_odos;
}
//-----
public void setEnt_odos(String entOdos) {
    ent_odos = entOdos;
}
//-----
public String getEnt_arithmos() {
    return ent_arithmos;
}

```

```

}
//-----
public void setEnt_arithmos(String entArithmos) {
    ent_arithmos = entArithmos;
}
//-----
public String getEnt_tk() {
    return ent_tk;
}
//-----
public void setEnt_tk(String entTk) {
    ent_tk = entTk;
}
//-----
public String getEnt_tel() {
    return ent_tel;
}
//-----
public void setEnt_tel(String entTel) {
    ent_tel = entTel;
}
//-----
public String getEnt_fax() {
    return ent_fax;
}
//-----
public void setEnt_fax(String entFax) {
    ent_fax = entFax;
}
//-----
public String getEnt_email() {
    return ent_email;
}
//-----
public void setEnt_email(String entEmail) {
    ent_email = entEmail;
}
//-----
public String getEnt_website() {
    return ent_website;
}
//-----
public void setEnt_website(String entWebsite) {
    ent_website = entWebsite;
}
//-----
public String getEnt_surname_ekpr() {
    return ent_surname_ekpr;
}
//-----
public void setEnt_surname_ekpr(String entSurnameEkpr) {
    ent_surname_ekpr = entSurnameEkpr;
}
//-----
public String getEnt_name_ekpr() {
    return ent_name_ekpr;
}
//-----
public void setEnt_name_ekpr(String entNameEkpr) {
    ent_name_ekpr = entNameEkpr;
}

```

```

}
//-----
public String getEnt_thesh_ekpr() {
    return ent_thesh_ekpr;
}
//-----
public void setEnt_thesh_ekpr(String entTheshEkpr) {
    ent_thesh_ekpr = entTheshEkpr;
}
//-----
public String getEnt_afm_ekpr() {
    return ent_afm_ekpr;
}
//-----
public void setEnt_afm_ekpr(String entAfmEkpr) {
    ent_afm_ekpr = entAfmEkpr;
}
//-----
public String getEnt_adt_ekpr() {
    return ent_adt_ekpr;
}
//-----
public void setEnt_adt_ekpr(String entAdtEkpr) {
    ent_adt_ekpr = entAdtEkpr;
}
//-----
public String getEnt_mob_ekpr() {
    return ent_mob_ekpr;
}
//-----
public void setEnt_mob_ekpr(String entModEkpr) {
    ent_mob_ekpr = entModEkpr;
}
//-----
public int getEnt_account_books() {
    return ent_account_books;
}
//-----
public void setEnt_account_books(int ent_account_books) {
    this.ent_account_books = ent_account_books;
}
//-----
public String getEnt_nom_kodikos() {
    return ent_nom_kodikos;
}
//-----
public void setEnt_nom_kodikos(String ent_nom_kodikos) {
    this.ent_nom_kodikos = ent_nom_kodikos;
}
//-----
}
//-----

```

10.6. CompanyController.java

```

//-----
package mou.core.controller;
//-----
import mou.common.exceptions.result.AccResult;
import mou.common.paging.Page;

```

```

import mou.core.persistence.dao.CompanyDaoImpl;
import mou.core.persistence.model.CompanyVO;
//-----
public class CompanyController {
//-----
private CompanyDaoImpl dao;
//-----
public CompanyVO fetchCompany(CompanyVO company){
return dao.fetchCompany(company);
}
//-----
public Page<CompanyVO> fetchCompanies(final CompanyVO filter, int pageNo, int pageSize){
return dao.fetchCompanies(filter, pageNo, pageSize);
}
//-----
public AccResult insertCompany(final CompanyVO company){
return dao.insertCompany(company);
}
//-----
public AccResult updateCompany(final CompanyVO company){
return dao.updateCompany(company);
}
//-----
public void removeCompany(final CompanyVO company){
dao.removeCompany(company);
}
//-----
public long companyExists(final CompanyVO company) throws Exception {
return dao.companyExists(company);
}
//-----
public void setDao(CompanyDaoImpl dao) {
this.dao = dao;
}
//-----
}
//-----

```

10.7. CompanyDaoImpl.java

```

//-----
package mou.core.persistence.dao;
//-----
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Types;
import javax.sql.DataSource;
import mou.common.exceptions.result.AccResult;
import mou.common.paging.Page;
import mou.common.paging.PaginationHelper;
import mou.common.paging.common.Paging;
import mou.core.persistence.dao.interfaces.ICompanyDao;
import mou.core.persistence.model.CompanyVO;
import mou.core.persistence.storedprocedures.CompanyExists;
import org.apache.log4j.Logger;
import org.springframework.dao.DataAccessException;
import org.springframework.jdbc.core.JdbcTemplate;

```



```

import org.springframework.jdbc.core.PreparedStatementCreator;
import org.springframework.jdbc.core.RowMapper;
import org.springframework.jdbc.core.simple.ParameterizedRowMapper;
import org.springframework.jdbc.support.GeneratedKeyHolder;
import org.springframework.jdbc.support.KeyHolder;
import org.springframework.transaction.PlatformTransactionManager;
import org.springframework.transaction.TransactionStatus;
import org.springframework.transaction.support.DefaultTransactionDefinition;
import org.springframework.transaction.support.TransactionCallbackWithoutResult;
import org.springframework.transaction.support.TransactionTemplate;
//-----
public class CompanyDaoImpl implements ICompanyDao {
//-----
private static final Logger log = Logger.getLogger(CompanyDaoImpl.class);
private DataSource dataSource;
private TransactionTemplate txTemplate;
private String insertCompany;
private String retrieveCompany;
private String updateCompany;
private String deleteCompany;
private String fetchCompanies;
//-----
public void setDataSource(DataSource dataSource) {
    this.dataSource = dataSource;
}
//-----
public void setTransactionManager(PlatformTransactionManager txManager) {
    this.txTemplate = new TransactionTemplate(txManager);

this.txTemplate.setPropagationBehavior(DefaultTransactionDefinition.PROPROPAGATION_REQUIRE
D);
}
//-----
@SuppressWarnings("unchecked")
public Page<CompanyVO> fetchCompanies(final CompanyVO filter, int pageNo, int pageSize) {
    JdbcTemplate select = new JdbcTemplate(dataSource);
    PaginationHelper<CompanyVO> ph = new PaginationHelper<CompanyVO>();
    String query = fetchCompanies;
    boolean applyCriteria = false;
    if (filter.getEnt_name() != null && filter.getEnt_name().length() > 0) {
        query += (" where acc_enterp_info.ent_name like " + filter.getEnt_name() + "");
        applyCriteria = true;
    }
    if (filter.getEnt_afm() != null && filter.getEnt_afm().length() > 0) {
        if(applyCriteria){
            query += (" and acc_enterp_info.ent_afm like " + filter.getEnt_afm() + "");
        }else{
            query += (" where acc_enterp_info.ent_afm like " + filter.getEnt_afm() + "");
            applyCriteria = true;
        }
    }
    if (filter.getEnt_nomikh_morph() != 0) {
        if(applyCriteria){
            query += (" and acc_enterp_info.ent_nomikh_morph = " + filter.getEnt_nomikh_morph());
        }else{
            query += (" where acc_enterp_info.ent_nomikh_morph = " + filter.getEnt_nomikh_morph());
        }
    }
    return ph.fetchPage(
        select,

```

```

        "SELECT count(acc_enterp_info.ent_id) " +
fetchCompanies.substring(query.indexOf("from"),
    query,
    new Object[]{}),
    Paging.YES,
    pageNo,
    pageSize,
    new ParameterizedRowMapper<CompanyVO>() {
        public CompanyVO mapRow(ResultSet rs, int i) throws SQLException {
            CompanyVO company = new CompanyVO();
            company.setEnt_id(rs.getLong("ent_id"));
            company.setEnt_name(rs.getString("ent_name"));
            company.setEnt_title(rs.getString("ent_title"));
            company.setEnt_kad(rs.getString("ent_kad"));
            company.setEnt_afm(rs.getString("ENT_AFM"));
            company.setEnt_doy_descr(rs.getString("ENT_DOY_DESCR"));
            company.setEnt_tomeas(rs.getString("ENT_TOMEAS"));
            company.setEnt_ypo_systash(rs.getInt("ENT_YPO_SYSTASH"));
            company.setEnt_nomikh_morph(rs.getInt("ENT_NOMIKH_MORFH"));
            company.setEnt_employees_num(rs.getInt("ENT_EMPLOYEES_NUM"));
            company.setEnt_xor_kodikos(rs.getString("ENT_XOR_KODIKOS"));
            company.setEnt_nom_kodikos(rs.getString("ENT_NOM_KODIKOS"));
            company.setEnt_per_kodikos(rs.getString("ENT_PER_KODIKOS"));
            company.setEnt_dhm_kodikos(rs.getString("ENT_DHM_KODIKOS"));
            company.setEnt_dd_kodikos(rs.getString("ENT_DD_KODIKOS"));
            company.setEnt_odos(rs.getString("ENT_ODOS"));
            company.setEnt_arithmos(rs.getString("ENT_ARITHMOS"));
            company.setEnt_tk(rs.getString("ENT_TK"));
            company.setEnt_tel(rs.getString("ENT_TEL"));
            company.setEnt_fax(rs.getString("ENT_FAX"));
            company.setEnt_email(rs.getString("ENT_EMAIL"));
            company.setEnt_website(rs.getString("ENT_WEBSITE"));
            company.setEnt_surname_ekpr(rs.getString("ENT_SURNAME_EKPR"));
            company.setEnt_name_ekpr(rs.getString("ENT_NAME_EKPR"));
            company.setEnt_thesh_ekpr(rs.getString("ENT_THESH_EKPR"));
            company.setEnt_afm_ekpr(rs.getString("ENT_AFM_EKPR"));
            company.setEnt_adt_ekpr(rs.getString("ENT_ADT_EKPR"));
            company.setEnt_mob_ekpr(rs.getString("ENT_MOB_EKPR"));
            company.setEnt_account_books(rs.getInt("ENT_ACCOUNT_BOOKS"));
            return company;
        }
    }
);
}
//-----
@SuppressWarnings("unchecked")
public CompanyVO fetchCompany(final CompanyVO company) {
    JdbcTemplate select = new JdbcTemplate(dataSource);
    return (CompanyVO) select.queryForObject(retrieveCompany,
        new Object[]{company.getEnt_id()},
        new RowMapper() {
            public Object mapRow(ResultSet rs, int rowNum) throws SQLException {
                CompanyVO _company = new CompanyVO();
                _company.setEnt_id(rs.getLong("ENT_ID"));
                _company.setEnt_name(rs.getString("ENT_NAME"));
                _company.setEnt_title(rs.getString("ENT_TITLE"));
                _company.setEnt_kad(rs.getString("ENT_KAD"));
                _company.setEnt_afm(rs.getString("ENT_AFM"));
                _company.setEnt_doy_descr(rs.getString("ENT_DOY_DESCR"));
                _company.setEnt_tomeas(rs.getString("ENT_TOMEAS"));
            }
        }
    );
}

```

```

        _company.setEnt_ypo_systash(rs.getInt("ENT_YPO_SYSTASH"));
        _company.setEnt_nomikh_morph(rs.getInt("ENT_NOMIKH_MORFH"));
        _company.setEnt_employees_num(rs.getInt("ENT_EMPLOYEES_NUM"));
        _company.setEnt_xor_kodikos(rs.getString("ENT_XOR_KODIKOS"));
        _company.setEnt_per_kodikos(rs.getString("ENT_PER_KODIKOS"));
        _company.setEnt_dhm_kodikos(rs.getString("ENT_DHM_KODIKOS"));
        _company.setEnt_dd_kodikos(rs.getString("ENT_DD_KODIKOS"));
        _company.setEnt_odos(rs.getString("ENT_ODOS"));
        _company.setEnt_arithmos(rs.getString("ENT_ARITHMOS"));
        _company.setEnt_tk(rs.getString("ENT_TK"));
        _company.setEnt_tel(rs.getString("ENT_TEL"));
        _company.setEnt_fax(rs.getString("ENT_FAX"));
        _company.setEnt_email(rs.getString("ENT_EMAIL"));
        _company.setEnt_website(rs.getString("ENT_WEBSITE"));
        _company.setEnt_surname_ekpr(rs.getString("ENT_SURNAME_EKPR"));
        _company.setEnt_name_ekpr(rs.getString("ENT_NAME_EKPR"));
        _company.setEnt_thesh_ekpr(rs.getString("ENT_THESH_EKPR"));
        _company.setEnt_afm_ekpr(rs.getString("ENT_AFM_EKPR"));
        _company.setEnt_adt_ekpr(rs.getString("ENT_ADT_EKPR"));
        _company.setEnt_mob_ekpr(rs.getString("ENT_MOB_EKPR"));
        _company.setEnt_account_books(rs.getInt("ENT_ACCOUNT_BOOKS"));
        return _company;
    }
}
});
}
//-----
@SuppressWarnings("unchecked")
public AccResult insertCompany(final CompanyVO company) {
    final AccResult result = new AccResult();
    final JdbcTemplate jdbcTemplate = new JdbcTemplate(dataSource);
    final KeyHolder keyHolder = new GeneratedKeyHolder();
    try {
        txTemplate.execute(new TransactionCallbackWithoutResult() {
            public void doInTransactionWithoutResult(TransactionStatus status) {
                jdbcTemplate.update(
                    new PreparedStatementCreator() {
                        public PreparedStatement createPreparedStatement(Connection connection)
throws SQLException {
                            PreparedStatement ps = connection.prepareStatement(insertCompany, new
String[]{"ENT_ID"});
                            if(company.getEnt_name() != null){
                                ps.setString(1, company.getEnt_name());
                            }else{
                                ps.setNull(1, Types.NULL);
                            }
                            if(company.getEnt_title() != null){
                                ps.setString(2, company.getEnt_title());
                            }else{
                                ps.setNull(2, Types.NULL);
                            }
                            if(company.getEnt_kad() != null){
                                ps.setString(3, company.getEnt_kad());
                            }else{
                                ps.setNull(3, Types.NULL);
                            }
                            if(company.getEnt_afm() != null){
                                ps.setString(4, company.getEnt_afm());
                            }else{
                                ps.setNull(4, Types.NULL);
                            }
                        }
                    }
                );
            }
        });
    }
}

```

```

if(company.getEnt_doy_descr() != null){
    ps.setString(5, company.getEnt_doy_descr());
}else{
    ps.setNull(5, Types.NULL);
}
if(company.getEnt_tomeas() != null){
    ps.setString(6, company.getEnt_tomeas());
}else{
    ps.setNull(6, Types.NULL);
}
ps.setInt(7, company.getEnt_ypo_systash());
ps.setInt(8, company.getEnt_nomikh_morph());
ps.setDouble(9, company.getEnt_employees_num());
if(company.getEnt_xor_kodikos() != null){
    ps.setString(10, company.getEnt_xor_kodikos());
}else{
    ps.setNull(10, Types.NULL);
}
if(company.getEnt_per_kodikos() != null){
    ps.setString(11, company.getEnt_per_kodikos());
}else{
    ps.setNull(11, Types.NULL);
}
if(company.getEnt_nom_kodikos() != null){
    ps.setString(12, company.getEnt_nom_kodikos());
}else{
    ps.setNull(12, Types.NULL);
}
if(company.getEnt_dhm_kodikos() != null){
    ps.setString(13, company.getEnt_dhm_kodikos());
}else{
    ps.setNull(13, Types.NULL);
}
if(company.getEnt_dd_kodikos() != null){
    ps.setString(14, company.getEnt_dd_kodikos());
}else{
    ps.setNull(14, Types.NULL);
}
if(company.getEnt_odos() != null){
    ps.setString(15, company.getEnt_odos());
}else{
    ps.setNull(15, Types.NULL);
}
if(company.getEnt_arithmos() != null){
    ps.setString(16, company.getEnt_arithmos());
}else{
    ps.setNull(16, Types.NULL);
}
if(company.getEnt_tk() != null){
    ps.setString(17, company.getEnt_tk());
}else{
    ps.setNull(17, Types.NULL);
}
if(company.getEnt_tel() != null){
    ps.setString(18, company.getEnt_tel());
}else{
    ps.setNull(18, Types.NULL);
}
if(company.getEnt_fax() != null){
    ps.setString(19, company.getEnt_fax());
}

```

```

    }else{
        ps.setNull(19, Types.NULL);
    }
    if(company.getEnt_email() != null){
        ps.setString(20, company.getEnt_email());
    }else{
        ps.setNull(20, Types.NULL);
    }
    if(company.getEnt_website() != null){
        ps.setString(21, company.getEnt_website());
    }else{
        ps.setNull(21, Types.NULL);
    }
    if(company.getEnt_surname_ekpr() != null){
        ps.setString(22, company.getEnt_surname_ekpr());
    }else{
        ps.setNull(22, Types.NULL);
    }
    if(company.getEnt_name_ekpr() != null){
        ps.setString(23, company.getEnt_name_ekpr());
    }else{
        ps.setNull(23, Types.NULL);
    }
    if(company.getEnt_thesh_ekpr() != null){
        ps.setString(24, company.getEnt_thesh_ekpr());
    }else{
        ps.setNull(24, Types.NULL);
    }
    if(company.getEnt_afm_ekpr() != null){
        ps.setString(25, company.getEnt_afm_ekpr());
    }else{
        ps.setNull(25, Types.NULL);
    }
    if(company.getEnt_adt_ekpr() != null){
        ps.setString(26, company.getEnt_adt_ekpr());
    }else{
        ps.setNull(26, Types.NULL);
    }
    if(company.getEnt_mob_ekpr() != null){
        ps.setString(27, company.getEnt_mob_ekpr());
    }else{
        ps.setNull(27, Types.NULL);
    }
    ps.setInt(28, company.getEnt_account_books());
    ps.setLong(29, company.getEnt_id());
    return ps;
}
},
keyHolder);
}
});
result.setErrorCode(keyHolder.getKey().longValue());
return result;
} catch (DataAccessException e) {
    e.printStackTrace();
    result.setErrorCode(-1);
    result.setErrorDescription(e.getMessage());
    return result;
}
}
}

```

```

//-----
@SuppressWarnings("unchecked")
public AccResult updateCompany(final CompanyVO company) {
    final AccResult result = new AccResult();
    try {
        final JdbcTemplate jdbcTemplate = new JdbcTemplate(dataSource);
        txTemplate.execute(new TransactionCallbackWithoutResult() {
            public void doInTransactionWithoutResult(TransactionStatus status) {
                jdbcTemplate.update(updateCompany,
                    new Object[]{
                        company.getEnt_name(),
                        company.getEnt_title(),
                        company.getEnt_kad(),
                        company.getEnt_afm(),
                        company.getEnt_doy_descr(),
                        compan.getEnt_tomeas(),
                        company.getEnt_ypo_systash(),
                        company.getEnt_nomikh_morph(),
                        company.getEnt_employees_num(),
                        company.getEnt_xor_kodikos(),
                        company.getEnt_per_kodikos(),
                        company.getEnt_nom_kodikos(),
                        company.getEnt_dhm_kodikos(),
                        company.getEnt_dd_kodikos(),
                        company.getEnt_odos(),
                        company.getEnt_arithmos(),
                        company.getEnt_tk(),
                        company.getEnt_tel(),
                        company.getEnt_fax(),
                        company.getEnt_email(),
                        company.getEnt_website(),
                        company.getEnt_surname_ekpr(),
                        company.getEnt_name_ekpr(),
                        company.getEnt_thesh_ekpr(),
                        company.getEnt_afm_ekpr(),
                        company.getEnt_adt_ekpr(),
                        company.getEnt_mob_ekpr(),
                        company.getEnt_account_books(),
                        company.getEnt_id()
                    });
            }
        });
        result.setErrorCode(0);
        return result;
    } catch (DataAccessException e) {
        e.printStackTrace();
        result.setErrorCode(-1);
        result.setErrorDescription(e.getMessage());
        return result;
    }
}
//-----
public long companyExists(final CompanyVO company) throws Exception {
    CompanyExists companyExists = new CompanyExists(dataSource);
    return companyExists.execute(company.getEnt_afm());
}
//-----
public void removeCompany(final CompanyVO company) {
}
//-----

```

```

public String getInsertCompany() {
    return insertCompany;
}
//-----
public void setInsertCompany(String insertCompany) {
    this.insertCompany = insertCompany;
}
//-----
public String getRetrieveCompany() {
    return retrieveCompany;
}
//-----
public void setRetrieveCompany(String retrieveCompany) {
    this.retrieveCompany = retrieveCompany;
}
//-----
public String getUpdateCompany() {
    return updateCompany;
}
//-----
public void setUpdateCompany(String updateCompany) {
    this.updateCompany = updateCompany;
}
//-----
public String getDeleteCompany() {
    return deleteCompany;
}
//-----
public void setDeleteCompany(String deleteCompany) {
    this.deleteCompany = deleteCompany;
}
//-----
public String getFetchCompanies() {
    return fetchCompanies;
}
//-----
public void setFetchCompanies(String fetchCompanies) {
    this.fetchCompanies = fetchCompanies;
}
//-----
}
//-----

```

10.8. panelCompany.zul

```

<?variable-resolver class="org.zkoss.spring.DelegatingVariableResolver"?>
<?page id="Company"?>
<?component name="coreBandbox" extends="bandbox" class="mou.acc.bandboxes.CoreBandbox"
?>
<?component name="toolbarbutton" extends="toolbarbutton"
class="mou.common.zkcomponents.MouToolbarbutton" ?>
<window id="panelCompany"
    xmlns="http://www.zkoss.org/2005/zul"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.zkoss.org/2005/zul http://www.zkoss.org/2005/zul/zul.xsd"
    use="mou.acc.windows.acc.Company">
<hbox width="990px" style="border:1px solid black;background-color:#D1E9F6;padding:3px; ">
    <label value="Στοιχεία Επιχείρησης" style="font-weight:bold;"/>
</hbox>

```

```

<separator/>
<div id="errorsArea" width="95%" style="padding: 5px" visible="false">
  <hbox widths="15%, 85%" valign="top">
    <label value="Λάθη Εγγραφής;" style="color:red; font-weight:bold;"/>
    <vbox id="errorsvbox"/>
  </hbox>
</div>
<separator/>
<grid id="CompanyGrid" width="990px">
  <columns>
    <column width="245px" align="left"/>
    <column width="245px" align="left"/>
    <column width="245px" align="left"/>
    <column width="245px" align="left"/>
  </columns>
  <rows>
    <row sclass="arow" spans="1,3">
      <label value="Επωνυμία Επιχείρησης" sclass="mandLabel"/>
      <textbox id="ent_name" sclass="atextbox" maxlength="500"
width="700px"/>
    </row>
    <row sclass="arow" spans="1,3">
      <label value="Διακριτικός Τίτλος" sclass="mandLabel"/>
      <textbox id="ent_title" sclass="atextbox" maxlength="500"
width="700px"/>
    </row>
    <row sclass="arow">
      <label value="Κωδικός Κύριας Δραστηριότητας (ΚΑΔ 2008)" width="100px"/>
      <coreBandbox id="ent_kad" sclass="atextbox" maxlength="30"/>
      <label value="Τομέας" width="100px"/>
      <listbox id="ent_tomeas" mold="select" rows="1" sclass="lstbox" width="200px"/>
    </row>
    <row sclass="arow">
      <label value="ΑΦΜ" sclass="mandLabel" />
      <textbox id="ent_afm" sclass="atextbox" maxlength="12"/>
      <label value="ΔΟΥ" sclass="mandLabel" />
      <textbox id="ent_doy_descr" sclass="atextbox" maxlength="20"/>
    </row>
    <row sclass="arow">
      <label value="Χώρα Εδρας" sclass="mandLabel" />
      <coreBandbox id="ent_xor_kodikos" sclass="atextbox" maxlength="2"/>
      <label/>
      <label/>
    </row>
    <row sclass="arow">
      <label value="Περιφέρεια Εδρας" sclass="mandLabel" />
      <coreBandbox id="ent_per_kodikos" sclass="atextbox" maxlength="2"/>
      <label value="Νομός Εδρας" sclass="mandLabel" />
      <coreBandbox id="ent_nom_kodikos" sclass="atextbox" maxlength="3"/>
    </row>
    <row sclass="arow">
      <label value="Δήμος Εδρας" sclass="mandLabel" />
      <coreBandbox id="ent_dhm_kodikos" sclass="atextbox" maxlength="3"/>
      <label value="Δημοτικό Διαμέρισμα Εδρας" sclass="mandLabel"/>
      <coreBandbox id="ent_dd_kodikos" sclass="atextbox"/>
    </row>
    <row sclass="arow">
      <label value="Αριθμός Απασχολουμένων στην Επιχείρηση" sclass="mandLabel"/>

```



```

        <decimalbox id="ent_employees_num" format="#,##0.00" sclass="atextbox"
maxlength="35" width="100px"/>
        <label value="Υφιστάμενη" sclass="mandLabel" />
        <listbox id="ent_ypo_systash" mold="select" rows="1" sclass="lstbox"
width="45px">
            <listitem value="1" label="NAI" selected="true"/>
            <listitem value="0" label="OXI"/>
        </listbox>
    </row>
</row>
<row sclass="arow">
    <label value="Τηλέφωνο" sclass="mandLabel"/>
    <textbox id="ent_tel" sclass="atextbox" maxlength="30"/>
    <label value="Fax" sclass="mandLabel"/>
    <textbox id="ent_fax" sclass="atextbox" maxlength="30"/>
</row>
<row sclass="arow">
    <label value="Οδός" sclass="mandLabel"/>
    <textbox id="ent_odos" sclass="atextbox" maxlength="500"/>
    <label value="Αριθμός" sclass="mandLabel"/>
    <textbox id="ent_arithmos" sclass="atextbox" maxlength="10"/>
</row>
<row sclass="arow">
    <label value="Ταχυδρομικός Κώδικας" sclass="mandLabel"/>
    <textbox id="ent_tk" sclass="atextbox" maxlength="15"/>
    <label value="Ιστότοπος Εταιρείας" sclass="mandLabel"/>
    <textbox id="ent_website" sclass="atextbox" maxlength="30"/>
</row>
<row sclass="arow">
    <label value="Επώνυμο Εκπροσώπου Εταιρείας" sclass="mandLabel"/>
    <textbox id="ent_surname_ekpr" sclass="atextbox" maxlength="30"/>
    <label value="Όνομα Εκπροσώπου Εταιρείας" sclass="mandLabel"/>
    <textbox id="ent_name_ekpr" sclass="atextbox" maxlength="30"/>
</row>
<row>
<row sclass="arow">
    <label value="Θέση Εκπροσώπου στην Εταιρεία" sclass="mandLabel"/>
    <textbox id="ent_thesh_ekpr" sclass="atextbox" maxlength="30"/>
    <label value="ΑΦΜ Εκπροσώπου" sclass="mandLabel"/>
    <textbox id="ent_afm_ekpr" sclass="atextbox" maxlength="12"/>
</row>
<row sclass="arow">
    <label value="Αριθμός Δελτίου Ταυτότητας" sclass="mandLabel"/>
    <textbox id="ent_adt_ekpr" sclass="atextbox" maxlength="50"/>
    <label value="Κινητό Εκπροσώπου" sclass="mandLabel"/>
    <textbox id="ent_mob_ekpr" sclass="atextbox" maxlength="30"/>
</row>
<row sclass="arow">
    <label value="Νομική Μορφή" sclass="mandLabel" />
    <listbox id="ent_nomikh_morfh" mold="select" rows="1" sclass="lstbox"/>
    <label value="Είδος Βιβλίων" sclass="mandLabel" />
    <listbox id="ent_account_books" mold="select" rows="1" sclass="lstbox"/>
</row>
<row sclass="arow" spans="1,3">
    <label value="Email" sclass="mandLabel"/>
    <textbox id="ent_email" sclass="atextbox" maxlength="50"/>
</row>
</rows>
</grid>
<separator height="5px"/>
<div sclass="buttonsdiv" align="center">

```

```
<toolbarbutton id="addbtn" sclass="addbuttonprj"  
onClick="panelCompany.onModifyCompanyClick()"  
tooltiptext="Τροποποίηση Επιχείρησης"  
label="Τροποποίηση Επιχείρησης"/>  
</div>  
</window>
```

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΠΑ