



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
ΚΑΤΕΥΘΥΝΣΗ «ΟΡΓΑΝΩΣΗ ΚΑΙ ΔΙΟΙΚΗΣΗ»

**Ανάπτυξη ενός wrapper με την χρήση της τεχνολογίας των
Web Services για τον έλεγχο στατιστικής αποκάλυψης**

από την

Μαρία Κ. Φραντζή

Διατριβή που υπεβλήθη για την μερική ικανοποίηση των απαιτήσεων για την απόκτηση
Μεταπτυχιακού Διπλώματος Ειδίκευσης

Χανιά, 2007

© Copyright υπό Μαρία Φραντζή

2007

Η διατριβή της Μαρίας Φραντζή εγκρίνεται

Ματσατσίνης Νικόλαος	
Γρηγορούδης Ευάγγελος	
Δούμπος Μιχάλης	

Περίληψη

Η αυξανόμενη παραγωγή στατιστικών δεδομένων καθώς και η αυξανόμενη ροή δεδομένων προς εθνικούς/κυβερνητικούς, ερευνητικούς καθώς και ιδιωτικούς στατιστικούς οργανισμούς συνάγει με την όλο και αυξανόμενη απαίτηση για διασφάλιση του υψηλού βαθμού εμπιστευτικότητας (confidentiality) των δεδομένων αυτών.

Οι αρμόδιες στατιστικές υπηρεσίες συλλέγουν μεγάλες ποσότητες δεδομένων προκειμένου να τα επεξεργαστούν στατιστικά. Από την άλλη πλευρά αυτοί που παρέχουν προς στατιστική επεξεργασία τα δεδομένων τους (respondents), είναι πρόθυμοι να παρέχουν, στις αντίστοιχες στατιστικές υπηρεσίες, την απαραίτητη πληροφορία, με την προϋπόθεση ότι τα δεδομένα τους θα τύχουν ασφαλούς διαχείρισης. Αυτό σημαίνει ότι η εμπιστευτικότητα των ανταποκρινόμενων (respondents' confidentiality) πρέπει ανα πάσα στιγμή να εγγυάται. Ο βαθμός δε της απαιτούμενης εμπιστευτικότητας δύναται να ποικίλει σε σχέση με την φύση των υπό στατιστική επεξεργασία δεδομένων, έτσι για παράδειγμα δεδομένα που αφορούν τους ασθενείς ενός νοσοκομείου θα πρέπει να αποστέλλονται και κατόπιν να επεξεργάζονται υιοθετώντας το μέγιστο βαθμό εμπιστευτικότητας. Ωστόσο η δέσμευση αυτή για εμπιστευτικότητα των δεδομένων μοιραία έχει επιβάλλει κάποιους περιορισμούς ως προς το ποσό λεπτομερής μπορεί να είναι οι «δημοσιεύσεις» των δεδομένων.

Σκοπός της παρούσας εργασίας είναι να αναλύσει, σχεδιάσει και υλοποιήσει ένα ολοκληρωμένο σύστημα ελέγχου στατιστικής αποκάλυψης (Statistical Disclosure Control-SDC) διαθέσιμο σε απομακρυσμένους χρήστες, κάνοντας χρήση των τεχνολογιών του διαδικτύου και συγκεκριμένα των web services.

Αν και η ευρεία χρήση του διαδικτύου τα τελευταία χρόνια έχει ωφελήσει σημαντικά στην κατεύθυνση της διάθεσης μια συγκεκριμένης υπηρεσίας επεξεργασίας στατιστικών δεδομένων σε όλο και περισσότερους χρήστες ανεξαρτήτως της τοποθεσίας τους. Ωστόσο, αυτού του είδους η διάθεση των στατιστικών υπηρεσιών κρύβουν αυξημένους κινδύνους αποκάλυψης συγκριτικά με αυτούς που μπορεί να εντοπιστούν σε ένα τοπικό δίκτυο υπολογιστών.

Προκειμένου λοιπόν να περιοριστεί ο κίνδυνος της στατιστικής αποκάλυψης ακόμα και κατά την μετάδοση δεδομένων μέσω του διαδικτύου, η παρούσα εργασία προτείνει μια αρχιτεκτονική που αποτελείται από έναν client με μια γραφική διεπαφή για την αλληλεπίδραση με τον τελικό χρήστη και από ένα δεύτερο μέρος εγκατεστημένο στην πλευρά του server το οποίο ονομάστηκε Δομικό Τμήμα ή Building Block. Το εν λόγω Δομικό Τμήμα που προτείνεται είναι ουσιαστικά ένας wrapper, ένα κομμάτι δηλαδή λογισμικού το οποίο περικλείει μια εσωτερική (αυτόνομη) εφαρμογή, το τ-Argus και εν συνεχεία μέσω της υιοθέτησης της τεχνολογίας των Web Services την κάνει διαθέσιμη μέσω του διαδικτύου.

Το δε τ-Argus είναι λογισμικό που έχει αναπτυχθεί από το Statistics Netherlands, σκοπός του οποίου είναι να προστατεύει πίνακες δεδομένων υλοποιώντας τεχνικές ελέγχου στατιστικής αποκάλυψης και έχει προταθεί και χρησιμοποιείται από την ευρωπαϊκή στατιστική υπηρεσία - Eurostat.

Σημαντικά πλεονέκτημα της προτεινόμενης αρχιτεκτονικής πέραν της ασφαλούς μετάδοσης στατιστικών δεδομένων είναι το γεγονός ότι πρόκειται για ένα κομμάτι λογισμικού αφενός επαναχρησιμοποιήσιμο (reusable) και αφετέρου δυσλειτουργικό (interoperable). Επομένως, μπορεί να αποτελέσει τμήμα άλλων συστημάτων αλλά και να συνδεθεί και να αλληλεπιδράσει με τρίτες εφαρμογές, ανεξαρτήτως των τυχόν διαφορετικές γλωσσών προγραμματισμού υλοποίησης, πλατφόρμων, καθώς και των λοιπών τεχνολογικών τους χαρακτηριστικών.

Ευχαριστίες

Με την ολοκλήρωση της μεταπτυχιακής ερευνητικής μου εργασίας θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Νικόλαο Ματσατσίνη, Αναπληρωτή Καθηγητή του τμήματος Μηχανικών Παραγωγής και Διοίκησης του Πολυτεχνείου Κρήτης, για τις υποδείξεις, κατευθύνσεις και διορθώσεις που μου προσέφερε, για το χρόνο που μου αφιέρωσε και κυρίως για την κατανόηση που επέδειξε όταν αυτό ήταν αναγκαίο.

Συνάμα, οφείλω να ευχαριστήσω τα υπόλοιπα δύο μέλη της εξεταστικής επιτροπής, και συγκεκριμένα τον κ. Κωνσταντίνο Ζοπουνίδη, Καθηγητή του τμήματος Μηχανικών Παραγωγής και Διοίκησης του Πολυτεχνείου Κρήτης και τον κ. Ευάγγελο Γρηγορούδη, Λέκτορα του τμήματος Μηχανικών Παραγωγής και Διοίκησης του Πολυτεχνείου Κρήτης, για το χρόνο που διέθεσαν για την ανάγνωση της διατριβής και τις κριτικές παρατηρήσεις τους.

Θα ήταν παράληψη να μην ευχαριστήσω τη συμφοιτήτριά, συνάδελφο και φίλη Νατάσα Καραναστάση, για την ηθική αλλά και ουσιαστική συμβολή της για την ολοκλήρωση της παρούσας μεταπτυχιακής διατριβής.

Αφιέρωση

Στους δικούς μου, που με συντροφεύουν στην πορεία μου ακόμα κι αν στέκουν μακριά στο ακριτικό νησί μου. Είναι αυτοί που με στηρίζουν στα δύσκολα, με ενθαρρύνουν στα διλήμματα και με κάνουν κάθε στιγμή υπερήφανη.

Περιεχόμενα

ΕΥΧΑΡΙΣΤΙΕΣ.....	V
ΑΦΙΕΡΩΣΗ	VI
ΠΕΡΙΕΧΟΜΕΝΑ	VII
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ.....	- 10 -
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	- 12 -
ΚΕΦΑΛΑΙΟ 1^ο ΕΙΣΑΓΩΓΗ	- 13 -
1.1 ΑΝΑΓΚΑΙΟΤΗΤΑ.....	- 13 -
1.2 ΔΟΜΗ ΤΗΣ ΕΡΓΑΣΙΑΣ	- 14 -
ΚΕΦΑΛΑΙΟ 2^ο	- 17 -
ΈΛΕΓΧΟΣ ΣΤΑΤΙΣΤΙΚΗΣ ΑΠΟΚΑΛΥΨΗΣ	- 17 -
2.1 ΈΛΕΓΧΟΣ ΣΤΑΤΙΣΤΙΚΗΣ ΑΠΟΚΑΛΥΨΗΣ - STATISTICAL DISCLOSURE CONTROL (SDC).....	- 17 -
2.2 ΣΤΑΤΙΣΤΙΚΗ ΕΜΠΙΣΤΕΥΤΙΚΟΤΗΤΑ (STATISTICAL CONFIDENTIALITY)	- 19 -
2.2.1 Υπόσχεση προς τον Ανταποκρινόμενο – <i>Promise Respondent</i>	- 19 -
2.2.2 Ενέργειες για διατήρηση της εμπιστευτικότητας.....	- 19 -
2.2.3 Παρουσίαση (<i>Release</i>) Στατιστικής Πληροφορίας.....	- 20 -
2.2.4 Έλεγχος αποκάλυψης.....	- 20 -
2.3 ΑΣΦΑΛΕΙΑ.....	- 20 -
2.4 ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ	- 21 -
ΚΕΦΑΛΑΙΟ 3^ο	- 24 -
ΕΦΑΡΜΟΓΗ T-ARGUS	- 24 -
3.1 ΕΙΣΑΓΩΓΗ	- 24 -
3.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	- 26 -
ΚΕΦΑΛΑΙΟ 4^ο	- 29 -
ΑΡΧΙΤΕΚΤΟΝΙΚΗ	- 29 -
4.1 ΕΙΣΑΓΩΓΗ.....	- 29 -

4.2 WRAPPER	- 29 -
4.3 ΓΕΝΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ	- 30 -
4.4 ΕΠΙΚΟΙΝΩΝΙΑ ΜΕΤΑΞΥ ΠΕΛΑΤΗ-ΕΞΥΠΗΡΕΤΗΤΗ (CLIENT-SERVER) ΤΟΥ ΔΟΜΙΚΟΥ ΤΜΗΜΑΤΟΣ T-ARGUS - 34	
-	
4.5 ΜΕΤΑΦΟΡΑ ΑΡΧΕΙΩΝ ΜΕΤΑΞΥ ΠΕΛΑΤΗ-ΕΞΥΠΗΡΕΤΗΤΗ (CLIENT-SERVER)	- 34 -
ΚΕΦΑΛΑΙΟ 5^ο	- 36 -
ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ.....	- 36 -
5.1 ΕΙΣΑΓΩΓΗ	- 36 -
5.2 ΜΟΝΤΕΛΟ ΤΗΣ ΒΒ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	- 37 -
5.2.1 Διάγραμμα Σχέσεων Οντοτήτων (E-R).....	- 37 -
5.2.2 Πίνακες	- 38 -
5.3 ORACLE 10G DATABASE.....	- 40 -
ΚΕΦΑΛΑΙΟ 6^ο	- 41 -
ΠΡΟΔΙΑΓΡΑΦΕΣ ΤΟΥ WEB SERVICE ΔΟΜΙΚΟΥ ΤΜΗΜΑΤΟΣ.....	- 41 -
6.1 ΕΙΣΑΓΩΓΗ	- 41 -
6.2 ΜΕΘΟΔΟΙ.....	- 41 -
6.3 UML SEQUENCE DIAGRAMS	- 43 -
6.4 ΑΡΧΕΙΟ WSDL.....	- 45 -
ΚΕΦΑΛΑΙΟ 7^ο	- 48 -
ΣΧΕΔΙΑΣΜΟΣ ΤΟΥ CLIENT ΤΟΥ WEB SERVICE ΔΟΜΙΚΟΥ ΤΜΗΜΑΤΟΣ (BB).....	- 48 -
7.1 ΕΙΣΑΓΩΓΗ	- 48 -
7.2 ΤΕΧΝΟΛΟΓΙΕΣ ΕΠΙΚΟΙΝΩΝΙΑΣ.....	- 48 -
7.3 WEB SERVICE CLIENTS.....	- 49 -
7.4 ΣΧΕΔΙΑΣΤΙΚΕΣ ΘΕΩΡΗΣΕΙΣ ΓΙΑ J2SE CLIENT	- 50 -
7.5 ΛΟΙΠΕΣ ΘΕΩΡΗΣΕΙΣ	- 52 -
7.5.1 Διαχείριση Διαλογικής Κατάστασης – Managing Conversational State.....	- 52 -
7.5.2 Συγχρονισμός από κοινού καταστάσεων μεταξύ clients – Synchronizing Shared State among Clients	- 52 -
7.5.3 Συναλλαγές Μακράς Διάρκειας – Long Running Transactions.....	- 53 -
7.6 ΡΟΗ (FLOW) ΚΛΗΣΗΣ WEB SERVICE	- 53 -
7.7 ΈΚΔΟΣΗ (PUBLICATION) ΤΟΥ CLIENT BUILDING BLOCK WEB SERVICE INTERFACE.....	- 56 -

7.8 ΕΠΙΠΕΔΟ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ (INTERACTION) BUILDING BLOCK SERVICE	- 56 -
7.9 INTERFACE ENDPOINT	- 57 -
ΚΕΦΑΛΑΙΟ 8^ο	- 59 -
ΟΔΗΓΟΣ ΧΡΗΣΗΣ ΕΦΑΡΜΟΓΗΣ (APPLICATION'S USER GUIDE)	- 59 -
8.1 ΕΙΣΑΓΩΓΗ	- 59 -
8.2 BUILDING BLOCK (ΔΤ) T-ARGUS SERVICE	- 59 -
8.3 TARGUS ΑΡΧΕΙΑ ΕΙΣΟΔΟΥ (INPUT FILES)	- 60 -
8.4 ΚΛΗΣΗ (INVOCATION) TARGUS SERVICE	- 62 -
8.5 TARGUS ΑΡΧΕΙΑ ΕΞΟΔΟΥ (OUTPUT FILES)	- 67 -
ΚΕΦΑΛΑΙΟ 9^ο	- 70 -
ΟΔΗΓΟΣ ΕΓΚΑΤΑΣΤΑΣΗΣ ΚΑΙ ΔΙΑΜΟΡΦΩΣΗΣ ΤΟΥ APPLICATION SERVER.....	- 70 -
9.1 ΕΙΣΑΓΩΓΗ	- 70 -
9.2 ΟΡΙΣΜΟΣ ΠΕΔΙΟΥ ΟΡΙΣΜΟΥ (DOMAIN) TARGUS BUILDING BLOCK.....	- 70 -
9.3 ΕΚΚΙΝΗΣΗ APPLICATION SERVER.....	- 77 -
9.4 DEPLOY TARGUS BUILDING BLOCK.....	- 78 -
9.5 ΔΙΑΔΙΚΑΣΙΑ ΠΡΩΤΟΥ DEPLOYMENT	- 79 -
9.5.1 Ορισμός startup class	- 81 -
9.5.2 Αρχείο BBTArgus.properties	- 83 -
9.6 REDEPLOYMENT	- 84 -
ΚΕΦΑΛΑΙΟ 10^ο	- 86 -
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	- 86 -
10.1 ΣΥΜΠΕΡΑΣΜΑΤΑ	- 86 -
10.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	- 87 -
ΠΑΡΑΡΤΗΜΑ.....	- 90 -
1. WSDL ΑΡΧΕΙΟ ΤΟΥ T-ARGUS SERVICE	- 90 -
ΣΥΝΤΟΜΕΥΣΕΙΣ.....	- 104 -
ΒΙΒΛΙΟΓΡΑΦΙΑ	- 106 -

Κατάλογος Σχημάτων

ΣΧΗΜΑ 1: ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ T-ARGUS.....	- 25 -
ΣΧΗΜΑ 2: INTERACTIVE ΜΟΡΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ T-ARGUS	- 28 -
ΣΧΗΜΑ 3: ΑΠΕΙΚΟΝΙΣΗ ΤΟΥ WRAPPER ΣΤΑ ΠΛΑΙΣΙΑ ΤΗΣ ΥΛΟΠΟΙΗΣΗΣ ΤΟΥ ΔΟΜΙΚΟΥ ΤΜΗΜΑΤΟΣ (BB).....	- 30 -
ΣΧΗΜΑ 4: ΓΕΝΙΚΗ (GENERIC) ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ ΠΡΟΣΕΓΓΙΣΗΣ ΤΟΥ ΔΟΜΙΚΟΥ ΤΜΗΜΑΤΟΣ (BB).....	- 33 -
ΣΧΗΜΑ 5: ΔΙΑΓΡΑΜΜΑ E-R BB ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	- 37 -
ΣΧΗΜΑ 6: SEQUENCE DIAGRAM ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ CLIENT ΜΕ WEB SERVICE.....	- 44 -
ΣΧΗΜΑ 7: ΔΙΑΧΕΙΡΙΣΗ ΤΗΣ ΟΥΡΑΣ ΤΩΝ T-ARGUS ΕΡΓΑΣΙΩΝ (JOB QUEUE)	- 45 -
ΣΧΗΜΑ 8: WSDL ΚΑΙ WEB SERVICES.....	- 46 -
ΣΧΗΜΑ 9: ΕΝΑΛΛΑΚΤΙΚΟΙ WEB SERVICE CLIENTS.....	- 50 -
ΣΧΗΜΑ 10: ΤΥΠΙΚΗ ΡΟΗ ΚΛΗΣΗΣ (FLOW) WEB SERVICE	- 55 -
ΣΧΗΜΑ 11: ΑΡΧΕΙΑ ΕΙΣΟΔΟΥ ΤΟΥ T-ARGUS SERVICE.....	- 60 -
ΣΧΗΜΑ 12: BATCH FILE ΑΡΧΕΙΟ ΕΙΣΟΔΟΥ	- 61 -
ΣΧΗΜΑ 13: ΠΙΝΑΚΟΕΙΔΗΣ ΑΡΧΕΙΟ ΕΙΣΟΔΟΥ	- 61 -
ΣΧΗΜΑ 14: ΑΡΧΕΙΟ (ΕΙΣΟΔΟΥ) ΜΕΤΑΔΕΔΟΜΕΝΩΝ	- 62 -
ΣΧΗΜΑ 15: ΑΡΧΕΙΟ (ΕΙΣΟΔΟΥ) ΙΕΡΑΡΧΙΑΣ.....	- 62 -
ΣΧΗΜΑ 16: ΚΥΡΙΩΣ ΠΑΡΑΘΥΡΟ ΔΙΑΛΟΓΟΥ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	- 63 -
ΣΧΗΜΑ 17: ΚΥΡΙΩΣ ΠΑΡΑΘΥΡΟ ΔΙΑΛΟΓΟΥ ΤΗΣ ΚΛΗΣΗΣ ΤΟΥ T-ARGUS SERVICE (MAIN T-ARGUS SERVICE INVOCATION DIALOG).....	- 63 -
ΣΧΗΜΑ 18: ΣΧΗΜΑ 19: ΚΥΡΙΩΣ ΠΑΡΑΘΥΡΟ ΔΙΑΛΟΓΟΥ ΤΗΣ ΚΛΗΣΗΣ ΤΟΥ T-ARGUS SERVICE ΜΕ ΔΕΔΟΜΕΝΑ ΕΙΣΟΔΟΥ	- 64 -
ΣΧΗΜΑ 20: ΠΑΡΑΘΥΡΟ ΑΠΟΤΕΛΕΣΜΑΤΩΝ	- 66 -
ΣΧΗΜΑ 21: ΑΡΧΕΙΟ ΕΞΟΔΟΥ ΤΗΣ ΜΟΡΦΗΣ CSV	- 68 -
ΣΧΗΜΑ 22: ΑΡΧΕΙΟ (ΕΞΟΔΟΥ) - REPORT FILE- ΣΕ ΜΟΡΦΗ HTML	- 68 -
ΣΧΗΜΑ 23: LOG ΑΡΧΕΙΟ ΕΞΟΔΟΥ	- 69 -
ΣΧΗΜΑ 24: ΜΕΝΟΥ BEA WEBLOGIC SERVER	- 70 -
ΣΧΗΜΑ 25: BEA WEBLOGIC CONFIGURATION WIZARD – ΔΗΜΙΟΥΡΓΙΑ Η ΕΠΕΚΤΑΣΗ ΔΙΑΜΟΡΦΩΣΗΣ	- 71 -
ΣΧΗΜΑ 26: BEA WEBLOGIC CONFIGURATION WIZARD – ΕΠΙΛΟΓΗ ΠΕΡΙΓΡΑΜΜΑΤΟΣ ΔΙΑΜΟΡΦΩΣΗΣ	- 72 -
ΣΧΗΜΑ 27: BEA WEBLOGIC CONFIGURATION WIZARD – ΕΠΙΛΟΓΗ ΤΑΧΕΙΑΣ Η ΚΑΤ’ ΑΠΑΙΤΗΣΗ ΔΙΑΜΟΡΦΩΣΗΣ	- 73 -
ΣΧΗΜΑ 28: BEA WEBLOGIC CONFIGURATION WIZARD – ΔΙΑΜΟΡΦΩΣΗ ΤΟΥ ΔΙΑΧΕΙΡΙΣΤΙΚΟΥ USERNAME ΚΑΙ PASSWORD.....	- 74 -
ΣΧΗΜΑ 29: BEA WEBLOGIC CONFIGURATION WIZARD – ΔΙΑΜΟΡΦΩΣΗ ΤΟΥ SERVER ΕΚΚΙΝΗΣΗΣ ΚΑΙ ΤΟΥ JAVA SDK.....	- 75 -
ΣΧΗΜΑ 30: BEA WEBLOGIC CONFIGURATION WIZARD – ΔΗΜΙΟΥΡΓΙΑ ΔΙΑΜΟΡΦΩΣΗΣ	- 77 -
ΣΧΗΜΑ 31: ΜΕΝΟΥ BEA WEBLOGIC PLATFORM – ΕΚΚΙΝΗΣΗ APPLICATION SERVER	- 78 -
ΣΧΗΜΑ 32: ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΔΙΑΧΕΙΡΙΣΤΙΚΗ ΚΟΝΣΟΛΑ ΤΟΥ WEBLOGIC APPLICATION SERVER.....	- 78 -

ΣΧΗΜΑ 33: DEPLOYMENT TOY WEB SERVICE (1)	- 79 -
ΣΧΗΜΑ 34: DEPLOYMENT TOY WEB SERVICE (2)	- 79 -
ΣΧΗΜΑ 35: DEPLOYMENT TOY WEB SERVICE (3)	- 80 -
ΣΧΗΜΑ 36: DEPLOYMENT TOY WEB SERVICE (4)	- 81 -
ΣΧΗΜΑ 37: ΔΗΜΙΟΥΡΓΙΑ STARTUP ΚΛΑΣΗΣ (1).....	- 81 -
ΣΧΗΜΑ 38: ΔΗΜΙΟΥΡΓΙΑ STARTUP ΚΛΑΣΗΣ (2).....	- 82 -
ΣΧΗΜΑ 39: ΣΤΙΓΜΙΟΤΥΠΟ ΑΠΟ ΤΟ COMMAND PROMPT TOY APPLICATION SERVER	- 84 -

Κατάλογος Πινάκων

ΠΙΝΑΚΑΣ 1: ΜΕΘΟΔΟΙ ΤΟΥ BB WEB SERVICE	- 43 -
ΠΙΝΑΚΑΣ 2: ΑΝΤΙΣΤΟΙΧΗΣΗ ΜΕΤΑΞΥ ΤΩΝ ΜΕΤΑΔΕΔΟΜΕΝΩΝ ΤΩΝ BLOCK'S ΚΑΙ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ WSDL ΕΓΓΡΑΦΟΥ	- 47 -

Κεφάλαιο 1^ο

Εισαγωγή

Οι αυξανόμενες απαιτήσεις ερευνητών, πολιτικών και άλλων παραγόντων για όλο και περισσότερο λεπτομερή στατιστική πληροφορία έχει αναπόφευκτα οδηγήσει σε σημεία τριβής και σύγκρουσης. Οι αρμόδιες στατιστικές υπηρεσίες συλλέγουν μεγάλες ποσότητες δεδομένων προκείμενου να τα επεξεργαστούν στατιστικά. Από την άλλη πλευρά αυτοί που παρέχουν προς στατιστική επεξεργασία τα δεδομένων τους (respondents), είναι πρόθυμοι να παρέχουν στις αντίστοιχες στατιστικές υπηρεσίες την απαραίτητη πληροφορία με την προϋπόθεση ότι τα δεδομένα τους θα τύχουν ασφαλούς διαχείρισης. Αυτό σημαίνει ότι η εμπιστευτικότητα των ανταποκρινόμενων (respondents' confidentiality) πρέπει ανα πάσα στιγμή να εγγυάται. Ωστόσο η δέσμευση αυτή είναι μοιραίο να έχει επιβάλλει κάποιους περιορισμούς ως προς το ποσό λεπτομερής μπορεί να είναι οι «δημοσιεύσεις» των δεδομένων. Τόσο η πράξη όσο και η έρευνα έχουν προσφέρει στον κόσμο της στατιστικής επεξεργασίας δεδομένων βαθιά γνώση σχετικά με το πως μπορούν να προστατευτούν τα προς επεξεργασία δεδομένα, ωστόσο το πρόβλημα δεν έχει οριστικά επιλυθεί και επιδέχεται περαιτέρω έρευνα και ανάλυση.

1.1 Αναγκαιότητα

Η αυξανόμενη παραγωγή στατιστικών δεδομένων καθώς και η αυξανόμενη ροή δεδομένων προς εθνικούς/κυβερνητικούς, ερευνητικούς καθώς και ιδιωτικούς στατιστικούς οργανισμούς αντικατοπτρίζει ταυτόχρονα το βαθμό της σπουδαιότητας της εμπιστευτικότητας (confidentiality) των δεδομένων. Πολλές από τις ήδη υπάρχουσες μεθοδολογίες στον χώρο αυτό είναι προσανατολισμένες στη φύση (τύπο, κατηγορία, μορφή, κ.α.) των δεδομένων που επεξεργάζονται και παράλληλα απαιτούν αρκετή επεξεργαστική ισχύ για να πραγματοποιήσουν τους ελέγχους τους. Οπότε πρόκειται για λύσεις που δεν μπορούν να χαρακτηριστούν αρκετά γενικές (generic) ώστε να καλύπτουν μια μεγάλη γκάμα δεδομένων και επιπλέον δεν είναι βέλτιστες ως προς τον τρόπο και το χρόνο επεξεργασίας αυτών. Έτσι, η περαιτέρω έρευνα για αναζήτηση μεθοδολογιών σχετικές με τον αποτελεσματικό έλεγχο αποκαλυψιμότητας (disclosure control) κρίνονται απαραίτητες.

Επιπρόσθετα, σε σημαντικές κρατικές και μη υπηρεσίες «τρέχει» ένας μεγάλος αριθμός από εφαρμογές με σκοπό την παραγωγή στατιστικών δεδομένων. Αυτές οι εφαρμογές ενώ αρχικά εξυπηρετούν παρεμφερείς σκοπούς παρουσιάζουν προβλήματα ασυμβατότητας μεταξύ των. Αναγκαία λοιπόν είναι η δημιουργία επαναχρησιμοποιήσιμων τμημάτων λογισμικού (reusable software components) τα οποία εκτελούν επί μέρους λειτουργίες (sub-functions) της όλης επεξεργασίας (validation, estimation, disclosure, seasonal adjustment). Αυτά τα τμήματα λογισμικού μπορούν να χρησιμοποιηθούν προκειμένου να συνδεθούν διαφορετικές εφαρμογές μεταξύ τους καθώς και να αναλυθούν επί μέρους (factorised) στην περίπτωση δημιουργίας νέων εφαρμογών.

Μέχρι σήμερα έχουν υλοποιηθεί κάποια εργαλεία σε διαφορετικές περιοχές στατιστικού ενδιαφέροντος, τα οποία όμως δεν ήταν αρκετά γενικά ή δεν επέτρεπαν την εύκολη διασύνδεση τους με άλλες εφαρμογές (Demetra, τ-ARGUS software).

Σκοπός της παρούσας εργασίας είναι να κάνει διαθέσιμο σε άλλες εφαρμογές (διαλειτουργικότητα) ένα ήδη υπάρχων λογισμικό για SDC - το τ-ARGUS το οποίο και χρησιμοποιείται ήδη από την ευρωπαϊκή στατιστική υπηρεσία (Eurostat) – άσχετα με τις διαφορετικές πλατφόρμες στις οποίες ενδεχομένως να τρέχουν οι εφαρμογές, τις διαφορετικές γλώσσες προγραμματισμού στις οποίες είναι γραμμένες καθώς και τις λοιπές τεχνολογίες που υιοθετούν. Αυτός ο σκοπός μπορεί να επιτευχθεί με την υλοποίηση ενός wrapper με την χρήση της τεχνολογίας των web services.

Ο wrapper που θα υλοποιηθεί θα αποτελείται από επαναχρησιμοποιήσιμο λογισμικό το οποίο με υπό-διαδικασίες της επεξεργασίας των στατιστικών δεδομένων, όπως για παράδειγμα επαλήθευση, αποτίμηση, αποκάλυψη, παροδική ρύθμιση, κ.ά. Αυτά τα επιμέρους κομμάτια λογισμικού μπορούν κάλλιστα να διατεθούν μέσω ενός interface και να συνδεθούν με εξωτερικές εφαρμογές ή να χρησιμοποιηθούν από νέες.

1.2 Δομή της εργασίας

Μετά το παρόν κεφάλαιο, το οποίο αποτελεί και την εισαγωγή της εργασίας, ακολουθούν εννέα ακόμη κεφάλαια. Συνοπτικά, αναφέρεται το αντικείμενο του καθενός από αυτά:

- **Κεφάλαιο 2 – Έλεγχος Στατιστικής Αποκάλυψης (SDC):** παρουσιάζεται η έννοια του ελέγχου στατιστικής αποκάλυψης καθώς και της στατιστικής εμπιστευτικότητας (statistical confidentiality) καθώς και τα χαρακτηριστικά αυτής. Εν συνεχεία επιχειρείται η προεπισκόπηση της ισχύουσας κατάστασης σε γνωστικά πεδία συναφή προς την εν λόγω εργασία. Επίσης γίνεται αναφορά σε μια παρεμφερή προσέγγιση του ελέγχου στατιστικής αποκάλυψης με τη χρήση της τεχνολογίας των web services, παρουσιάζοντας παράλληλα τα μειονεκτήματα της εφαρμογής αυτής έναντι εκείνης που αναπτύχθηκε στα πλαίσια της παρούσας μεταπτυχιακής διατριβής.
- **Κεφάλαιο 3 – Εφαρμογή τ-Argus:** Παρουσιάζεται η εφαρμογή τ-Argus, τα χαρακτηριστικά της καθώς και ο τρόπος λειτουργίας της. Η batch μορφή της εφαρμογής αυτής αποτελεί μέρος της αρχιτεκτονικής που προτείνεται στο επόμενο κεφάλαιο.
- **Κεφάλαιο 4 – Αρχιτεκτονική:** Παρουσιάζεται η γενική αρχιτεκτονική του προτεινόμενου συστήματος, εξηγείται η έννοια του wrapper και αναλύονται το θέμα του τρόπου επικοινωνίας μεταξύ client και server καθώς και η προσέγγιση που ακολουθήθηκε για την μεταφορά αρχείων μεταξύ client και server.
- **Κεφάλαιο 5 – Βάση Δεδομένων:** παρουσιάζεται το σχήμα της σχεσιακής βάσης δεδομένων που σχεδιάστηκε και υλοποιήθηκε για τις ανάγκες της εφαρμογής, ενώ παράλληλα αναλύονται οι επί μέρους πίνακες και τα αντίστοιχα στοιχεία τους.
- **Κεφάλαιο 6 – Προδιαγραφές του Web Service του Δομικού Τμήματος (BB):** Περιγράφεται πλήρως το web service Δομικού Τμήματος που σχεδιάστηκε και υλοποιήθηκε στα πλαίσια της παρούσας διατριβής. Περιγράφονται οι μέθοδοι του web service, παρουσιάζονται τα UML διαγράμματα διαδοχής που δείχνουν την διαδοχή των ενεργειών και παρατίθεται το WSDL έγγραφο που αποσκοπεί στον ολοκληρωμένο προσδιορισμό όλων των παραμέτρων του web service.
- **Κεφάλαιο 7 – Σχεδιασμός του client του Web Service Δομικού Τμήματος (BB):** Αναλύεται ο σχεδιασμός που ακολουθήθηκε για την προσέγγιση του client του BB web service. Επιχειρείται η συστηματική μελέτη των υφιστάμενων τεχνικών εναλλακτικών πάνω σε θέματα ειδικού ενδιαφέροντος που απασχόλησαν της συγκεκριμένη σχεδίαση καθώς και οι τελικές επιλογές που υιοθετήθηκαν.
- **Κεφάλαιο 8 – Οδηγός Χρήσης Εφαρμογής (User's Application Guide):** Παρουσιάζονται βήμα προς βήμα οι ενέργειες που πρέπει να ακολουθηθούν προκειμένου να χρησιμοποιηθεί οι εν λόγω εφαρμογή. Ιδιαίτεροι αναφορά γίνεται στα απαραίτητα αρχεία εισόδου που δέχεται η εφαρμογή καθώς και στα αναμενόμενα αρχεία εξόδου. Ενώ οι περιγραφές πλαισιώνονται από τα αντίστοιχα screenshots της εφαρμογής. Το κεφάλαιο αυτό ουσιαστικά αναφέρεται στον τρόπο λειτουργίας του client μέρους της εφαρμογής.
- **Κεφάλαιο 9 – Οδηγός Εγκατάστασης και Διαμόρφωσης του Application Server:** Περιγράφονται αναλυτικά όλες οι απαιτούμενες ενέργειες για την επιτυχή εγκατάσταση, την διαμόρφωση καθώς και την εκκίνηση του application server της εφαρμογής.
- **Κεφάλαιο 10 – Συμπεράσματα:** συνοψίζει το περιεχόμενο αυτής της εργασίας και προτείνει ενδεχόμενες μελλοντικές επεκτάσεις

Σημείωση

Σε όποια σημεία της εργασίας αυτής κρίθηκε άστοχη η απόδοση των αγγλικών τεχνικών όρων στην ελληνική γλώσσα, αυτό αποφεύχθηκε προς αποφυγή σύγχυσης του αναγνώστη. Όπου η απόδοση στα ελληνικά κρίνεται εύστοχη, τότε ο αντίστοιχος αγγλικός όρος παρουσιάζεται εντός των παρενθέσεων. Ενώ στις περιπτώσεις όπου γίνεται χρήση καθιερωμένων τεχνικών όρων στην ελληνική τότε παραλείπεται η παράθεση του όρου αυτού και στην αγγλική.

Κεφάλαιο 2^ο

Έλεγχος Στατιστικής Αποκάλυψης

2.1 Έλεγχος Στατιστικής Αποκάλυψης - Statistical Disclosure Control (SDC)

Ο ρόλος των στατιστικών γραφείων είναι να παράγουν και να δημοσιεύουν στατιστική πληροφορία σχετική με την κοινωνία. Τα δεδομένα που συλλέγονται τελικά παρουσιάζονται σε μια κατάλληλη μορφή σε πολιτικούς, ερευνητές καθώς και στο ευρύτερο κοινό, ικανοποιώντας στατιστικούς σκοπούς. Η παρουσίαση μιας τέτοιας πληροφορίας ενδεχομένως να έχει ανεπιθύμητο αποτελέσματα αποκάλυψης όσων αφορά τα άτομα (individuals) σε αντίθεση με τις μεγάλες ομάδες ατόμων. Σε αυτό το σημείο εύλογα τίθεται το ερώτημα πως είναι δυνατόν αυτή η διαθέσιμη πληροφορία να τροποποιείται με τέτοιο τρόπο ώστε τα δεδομένα που παρουσιάζονται να μπορούν να είναι στατιστικά χρήσιμα χωρίς ταυτόχρονα να επηρεάζεται η ιδιωτικότητα (privacy) των ατόμων. Η θεωρία του ελέγχου στατιστικής αποκάλυψης είναι αυτή που υιοθετείται για την επίλυση αυτού του προβλήματος, πως δηλαδή μπορεί να επιτευχθεί η παρουσίαση όσων το δυνατόν περισσότερων δεδομένων δίχως το κίνδυνο της αποκάλυψης πληροφορίας που αφορά άτομα. (Willenborg, L.C.R.J. and De Waal, A.G.,1996).

Η ακριβής αποτίμηση της επικινδυνότητας αποκάλυψης είναι απαραίτητος προάγγελος του αποτελεσματικού ελέγχου στατιστικής αποκάλυψης (statistical disclosure control - SDC). Αυτή η αποτίμηση συχνά βασίζεται στην ιδέα της μοναδικότητας, όπου μια εγγραφή (record) είναι μοναδική μέσα σε ένα σύνολο δεδομένων (dataset) όσων αφορά ένα προκαθορισμένο σύνολο από χαρακτηριστικά. Ο προσδιορισμός επικινδυνών ή “ειδικού ενδιαφέροντος” μοναδικά σημεία μεταξύ των δεδομένων, αποτελεί το βασικό στοιχείο της αποτίμησης του κινδύνου αποκάλυψης σε σχέση με παρουσίαση (release) αυτών των δεδομένων (Elliot et al., 2002).

Το Statistical Disclosure Control (SDC) – στατιστικός έλεγχος αποκάλυψης- αφορά την ασφάλεια της εμπιστευτικότητας (confidentiality) των πληροφοριών που συλλέγει ένας οργανισμός σχετικά με ανθρώπους, επιχειρήσεις, κ.α.

Το SDC περιλαμβάνει την τροποποίηση των δεδομένων ούτως ώστε ο κίνδυνος αναγνώρισης προσώπων (risk of identifying individuals) να μειώνεται σε ένα επιθυμητό επίπεδο. Σε γενικές γραμμές αυτή η διαδικασία περιλαμβάνει την αφαίρεση πληροφορίας (broad-banding variables, κτλ.) ή την καταστροφή πληροφορίας (perturbing values). Οι μέθοδοι SDC αποσκοπούν στην εύρεση μιας βέλτιστης ισορροπίας μεταξύ της βελτίωσης στην προστασία της εμπιστευτικότητας των δεδομένων και την αντίστοιχη μείωση της ποιότητάς τους.

Σε γενικές γραμμές, σπάνια συνδυασμοί χαρακτηριστικών οδηγούν σε ταυτοποίηση προσώπων (individuals), όπως για παράδειγμα μια δεκαεξάχρονη χήρα, μια γυναίκα μεταλλωρύχος, μια μοναδική βιομηχανία σε συγκεκριμένη περιοχή, κτλ. Συνήθως οι μέθοδοι DC εφαρμόζονται εφόσον ηθικές, πρακτικές και νομικές προϋποθέσεις επιτρέπουν στα δεδομένα να είναι κατά το μέγιστο προστατευμένα.

Αυτή τη στιγμή τεχνικές για SDC χρησιμοποιούνται σε ένα μεγάλο εύρος περιοχών των Εθνικών Στατιστικών Υπηρεσιών (National Statistics), για παράδειγμα το Census, το Neighbourhood Statistics Service, κ.α. Διαφορετικοί τύποι δεδομένων αντιπροσωπεύουν διαφορετικούς τύπους προβλημάτων και αναπόφευκτα απαιτούν διαφορετικές τεχνικές επίλυσης.

Μεταξύ του μεγάλου όγκου της πληροφορίας που είναι σήμερα στη διάθεση του ερευνητή-επιστήμονα και εξυπηρετεί στατιστικούς σκοπούς, βρίσκονται και μια σειρά από πρότυπες (standard) μελέτες ασφάλειας. Εξάλλου αποτελεί κοινό τόπο ότι οι στατιστικοί οργανισμοί έχουν ανάγκη να βεβαιώνονται κάθε φορά ότι η πληροφορία που τους προωθείτε, είτε από μεμονωμένα άτομα είτε από οργανισμούς, δεν είναι δυνατό να αποκαλυφθεί σε μη εξουσιοδοτημένη θέαση. Αυτός ακριβώς είναι και ο λόγος που προσδοκούν να χρησιμοποιήσουν πρότυπους μηχανισμούς ασφαλείας. Ωστόσο, ο ίδιος ο σκοπός της συστηματικής στατιστικής έρευνας είναι η αποκάλυψη της πληροφορίας – με την έννοια της παροχής πρόσβασης και όχι απαγόρευσης αυτής. Το αντικείμενο του ελέγχου στατιστικής αποκάλυψης είναι η ρύθμιση του κινδύνου να εξορυχθεί - από συγκεκριμένα άτομα - σημαντική πληροφορία μεταξύ των αθροισμάτων των στατιστικών αποτελεσμάτων.

Η στατιστική διασπορά (statistical dissemination) έχει να κάνει με την έκδοση/παρουσίαση της πληροφορίας. Ωστόσο, η πληροφορία που τελικά παρουσιάζεται υποτίθεται ότι θα ρέπει να διαφωτίζει τον αναγνώστη σχετικά με τη μέση συμπεριφορά των ομάδων, και όχι για τις ιδιαίτερες λεπτομέρειες συγκεκριμένων ατόμων.

Το πρόβλημα της διατήρησης της εμπιστευτικότητας (maintaining confidentiality) των αποκρίσεων των ατόμων (individual responses) έχει μελετηθεί εδώ και κάποιες δεκαετίες και αυτό είχε σαν αποτέλεσμα να υπάρχει μια εκτεταμένη σχετική βιβλιογραφία. Στο παρόν κεφάλαιο πρόκειται να προσδιοριστούν τα κύρια θέματα και προσεγγίσεις που έχουν χρησιμοποιηθεί. Πρόκειται για μια σύντομη αναφορά στην εν λόγω περιοχή και ερμηνεία της παρούσας κατάστασης. Προκειμένου να επιτευχθεί πληρέστερη και ολοκληρωμένη κατανόηση σκόπιμη είναι χρήση της αντίστοιχης Βιβλιογραφίας που παρατίθεται στο τέλος της εργασίας.

2.2 Στατιστική Εμπιστευτικότητα (Statistical Confidentiality)

2.2.1 Υπόσχεση προς τον Ανταποκρινόμενο – Promise Respondent

Είτε ο ανταποκρινόμενος είναι ένα πρόσωπο, μια επιχείρηση είτε ένας οργανισμός, ένας τίμιος συλλογέας δεδομένων πρέπει να δίνει την υπόσχεση ότι οι ατομικές πληροφορίες που παρέχονται θα πρέπει να παραμένουν εμπιστευτικές. Κάποιες φορές δε αυτή η υπόσχεση προφυλάσσεται και από την ισχύουσα νομοθεσία. Έτσι λοιπόν είναι απαραίτητες οι διαδικασίες εκείνες που να εκπληρώνουν τούτη την υπόσχεση, όπως επίσης και να μπορούν να αποδεικνύουν ότι αυτή έχει εκπληρωθεί.

Σε μεγάλο βαθμό, αξίζει τον κόπο να σημειώσει κανείς ότι αυτή η υπόσχεση μπορεί να επηρεάσει την ποιότητα των απαντήσεων, αφού ο φόβος της αποκάλυψης ενδέχεται να εμποδίσει τον respondent από το να στείλει όλα τα αληθινά αποτελέσματα αλλά ένα μέρος τους, στις περιπτώσεις των ευαίσθητων ερωτημάτων.

2.2.2 Ενέργειες για διατήρηση της εμπιστευτικότητας

Διαφορετικοί τύποι κινδύνου αποκάλυψης ανέρχονται σε διαφορετικά του στατιστικού κύκλου ζωής. Προφανώς το μεγαλύτερο ρίσκο (greatest risk of compromise) παρουσιάζεται κατά την μετάδοση και επεξεργασία των πρωτότυπων δεδομένων. Ωστόσο, σε αυτό το στάδιο η απαίτηση δεν είναι μια

και συγκεκριμένη για την εκάστοτε στατιστική πληροφορία, επομένως τα περισσότερα πρότυπα συστήματα ασφαλείας μπορούν κάλλιστα να χρησιμοποιηθούν για να προστατέψουν τα δεδομένα πηγής.

2.2.3 Παρουσίαση (Release) Στατιστικής Πληροφορίας

Ο σκοπός της συστηματικής έρευνας, ή της στατιστικής αναφοράς από ένα διαχειριστικό σύστημα (administrative system) είναι να παρουσιάσει (release) πληροφορία σχετικά με τα χαρακτηριστικά του υπό μελέτη συστήματος. Αυτό συνήθως επιτυγχάνεται με την προσαρμογή μοντέλων και την αποτίμηση των παραμέτρων αυτών των μοντέλων, είτε με την αναφορά της συνολικής πληροφορίας που εξάγεται για όλους ή για επιμέρους ομάδες των ανταποκρινόμενων (respondents).

2.2.4 Έλεγχος αποκάλυψης

Σε περιπτώσεις που το στατιστικό αποτέλεσμα είναι βασισμένο σε πολύ μικρό αριθμό από respondents τότε είναι πιθανό οδηγηθεί κανείς σε συμπεράσματα ή να εξάγει πληροφορία σχετικά με τα άτομα ξεχωριστά από τα συνολικά αθροίσματα. Αυτό όμως είναι ένα ειδικό πρόβλημα όταν ο «εισβολέας» (intruder) έχει στην κατοχή του μόνο βοηθητική πληροφορία σχετικά με έναν ανταποκρινόμενο-στόχο (targeted respondent) σχετικά με respondents της ίδιας υποομάδας.

2.3 Ασφάλεια

Όλα τα συστήματα τα οποία αποθηκεύουν και επεξεργάζονται ευαίσθητα δεδομένα έχουν ανάγκη υποστήριξης τους από μηχανισμούς ασφαλείας.

Παραδείγματα τέτοιων μηχανισμών είναι τα ακόλουθα:

- Απόκρυψη (encryption) εγγραφών, ιδιαίτερα όταν η μεταφορά τους γίνεται από μη-ασφαλή κανάλια. Ένα τέτοιο παραδείγματα μπορούσε να είναι η μετάδοση απαντήσεων μέσω του διαδικτύου από τον respondent browser προς μια κεντρική μονάδα αποθήκευσης.
- Προστασία της διαδικασίας αποθήκευσης των δεδομένων, συμπεριλαμβανομένου του ελέγχου προσβασιμότητας προκειμένου να διαβεβαιώνεται πάντα ότι μόνο όσοι είναι εξουσιοδοτημένοι μπορούν να βλέπουν/τροποποιούν την αποθηκευμένη πληροφορία.

- Πρακτικές εργασίας που να εξασφαλίζουν ότι το μη-εξουσιοδοτημένο προσωπικό δεν μπορεί να βλέπει ευαίσθητη πληροφορία μέσω εξουσιοδοτημένων χρηστών, ή αποκτώντας πρόσβαση σε σταθμούς εργασίας που έχουν αφεθεί δίχως επιτήρηση.
- Απόκρυψη χαρακτηριστικών πεδίων όταν αυτά δεν είναι απαραίτητα για την τρέχουσα επεξεργασία.

Όλες οι παραπάνω είναι καθιερωμένες πρακτικές, όχι μόνο για στατιστική πληροφορία, οι οποίες θα πρέπει να υιοθετούνται κατ'ελάχιστον από τα στατιστικά συστήματα. Ο απαιτούμενος βαθμός ασφάλειας μπορεί να ποικίλει από σύστημα σε σύστημα και έχει να κάνει με την ευαισθησία της πληροφορίας που συλλέγεται και αποθηκεύεται.

Έτσι, ένα παράδειγμα είναι αυτό του νοσοκομείου που προσδιορίζει με μοναδικά νούμερα τους ασθενείς του προκειμένου να εκδώσει στατιστικά δεδομένα σε αντίστοιχους οργανισμούς χωρίς όμως να επιτρέψει ταυτόχρονη πρόσβαση στα πληροφοριακά συστήματα του νοσοκομείου. Στην περίπτωση λοιπόν που η πληροφορία αφορά HIV και AIDS tests δεν είναι επιθυμητό να παρουσιαστούν τα πλήρη ονόματα των ασθενών. Μια λύση λοιπόν είναι να χρησιμοποιηθούν κωδικοί βασισμένοι στο επίθετο και στην ημερομηνία γέννησης του ασθενούς. Εφόσον τόσο το επίθετο όσο η ημερομηνία γέννησης δεν είναι μοναδικά για κάθε άνθρωπο, και συχνά μάλιστα αναφέρονται με σφάλματα, μπορεί να μην παρέχει την άριστη αντιστοίχιση αλλά δεν παύει να σημαίνει ότι υπάρχει μεγάλη πιθανότητα η αντιστοίχιση να είναι τελικά ορθή.

2.4 Σχετικές εργασίες

Αναζητώντας κανείς βιβλιογραφία σχετική με το αντικείμενο του ελέγχου στατιστικής αποκάλυψης διαπιστώνει ότι οι περισσότερες ομάδες στο χώρο χρηματοδοτούνται κυρίως από εθνικούς και κυβερνητικούς οργανισμούς. Οι σημαντικότεροι φορείς που δραστηριοποιούνται σ' αυτή την γνωστική περιοχή είναι:

- Ο οργανισμός Statistics Netherlands, ο οποίος έχει αναπτύξει και το πρωτότυπο πακέτο λογισμικού τ-Argus το οποίο και χρησιμοποιείται στην παρούσα εργασία.
- Η ερευνητική ομάδα της Dr. Sweeney assistant professor στο τμήμα Computer Science, Technology and Policy στο πανεπιστήμιο Carnegie-Melon. Η Dr. Sweeney διατηρεί εταιρία που ασχολείται με την έκδοση ιατρικών δεδομένων.

- Το αμερικανικό Εθνικό Ινστιτούτο Στατιστικών Επιστημών (National Institute of Statistical Sciences-NISS) εμφανίζεται αρκετά καινοτόμο και πρωτοπόρο σε θέματα που άπτονται του ζητήματος ελέγχου στατιστικής αποκάλυψης.
- Τέλος η ευρωπαϊκή ένωση και ειδικότερα η στατιστική της υπηρεσία, γνωστή ως Eurostat, χρηματοδοτεί μια σειρά από έργα που σκοπό έχουν την ανάπτυξη μεθόδων για την μείωση του κινδύνου της στατιστικής αποκάλυψης δεδομένων.

Λαμβάνοντας κανείς την πρόσφατη έρευνα που έχει πραγματοποιηθεί στο χώρο αυτό, εντοπίζει την ανάπτυξη διαφόρων ενδιαφερόντων και χρήσιμων τεχνικών για την αποφυγή αποκάλυψης των ελάχιστων δεδομένων. Οι τεχνικές αυτές έχουν να κάνουν με καταστολή (suppression), προσθήκη θορύβου, χρήση καθολικών (global) / τοπικών (local) εγγράφων, κ.α. (Waal, 1996 και Takemura, 1999). Οι δε τεχνικές αυτές είναι αποτελεσματικές όσον αφορά τον έλεγχο του κινδύνου της αποκάλυψης των δεδομένων αφού οι τροποποιήσεις στα δεδομένα υφίστανται πριν αυτά αποκαλυφθούν στο κοινό.

Ωστόσο ανοιχτό παραμένει το ερώτημα του κατά πόσο είναι ασφαλή η αποστολή ευαίσθητων δεδομένων μέσω του διαδικτύου ώστε να επιχειρηθεί κεντρικά η διαδικασία κωδικοποίησης τους και άρα να μειωθεί ο περαιτέρω κίνδυνος αποκάλυψης τους.

Αρκετά είναι εκείνα τα services που έχουν αναπτυχθεί στην κατεύθυνση αυτή και κάνουν χρήση των τεχνολογιών της Java ή CGI συστημάτων μέσω του WWW browser (Inoue et al 2001). Σε αυτές τις εφαρμογές online στατιστικά λογισμικά έχουν εγκατασταθεί οι οποίες με τη σειρά τους επιχειρούν απευθείας πρόσβαση στα γνήσια σύνολα στατιστικών δεδομένων και εν συνεχεία οι εξουσιοδοτημένοι χρήστες μπορούν να έχουν πρόσβαση στην ειδικά επεξεργασμένη πληροφορία – όπου κι αν αυτοί βρίσκονται- μέσω του διαδικτύου.

Παρ'όλα αυτά μόνο μια αναφορά σε αντίστοιχο σύστημα εντοπίστηκε στην σχετική ερευνητική περιοχή που να υιοθετεί την τεχνολογία των web services. Συγκεκριμένα πρόκειται για την υλοποίηση ενός web service για data swapping από το NISS.

Data swapping είναι η πρακτική του περιορισμού στατιστικής αποκάλυψης (statistical disclosure limitation) η οποία εναλλάσσει της τιμές των μεταβλητών κατά ζεύγη, περιορίζοντας κατ' αυτό τον τρόπο την αποκάλυψη των αρχικών δεδομένων.

Έτσι το NISS έχει αναπτύξει ένα web service που το ονομάζει WebSwap, το οποίο εναλλάσσει τις τιμές ενός ή περισσότερων πεδίων σε προκαθορισμένες από τον χρήστη εγγραφές ενός αρχείου με μικροδεδομένα (microdata). Το δε αρχείο με τα αρχικά δεδομένα έχει φορτωθεί από τον υπολογιστή του χρήστη στον κεντρικό server και αντιστοίχως από το server, μετά την ολοκλήρωση της επεξεργασίας, μπορεί να ανακτηθεί το τελικό αρχείο με τα swapped records.

Μειονέκτημα της παραπάνω προσέγγισης είναι κυρίως το γεγονός ότι μπορεί να εφαρμοστεί μόνο σε περιπτώσεις αρχείων με μικροδεδομένα και όχι στις περιπτώσεις των δεδομένων σε πινακοειδή (tabular) μορφή. Επιπλέον υλοποιεί μια μονοδιάστατη τεχνική του data swapping, ενώ στην εν λόγω εργασία επιχειρείται η παραγωγή ασφαλών πινάκων δεδομένων έναντι του κινδύνου της στατιστικής αποκάλυψης. Έτσι οι ελάχιστες πίνακες επανασχεδιάζονται, στήλες και γραμμές συνενώνονται, ευαίσθητα δεδομένα αποκρύπτονται και όλα αυτά με τη χρήση καθιερωμένων αλγορίθμων και μεθόδων (hypercube, modular, optimal, GHMITER). Επιπρόσθετα, στο παρόν σύστημα έχει παράλληλα αναπτυχθεί και ένα σύστημα διαχείρισης ουράς εργασιών ώστε να ικανοποιείται η ταυτόχρονη ανάγκη για παραγωγή ασφαλών πινάκων με στατιστικά δεδομένα.

Κεφάλαιο 3^ο

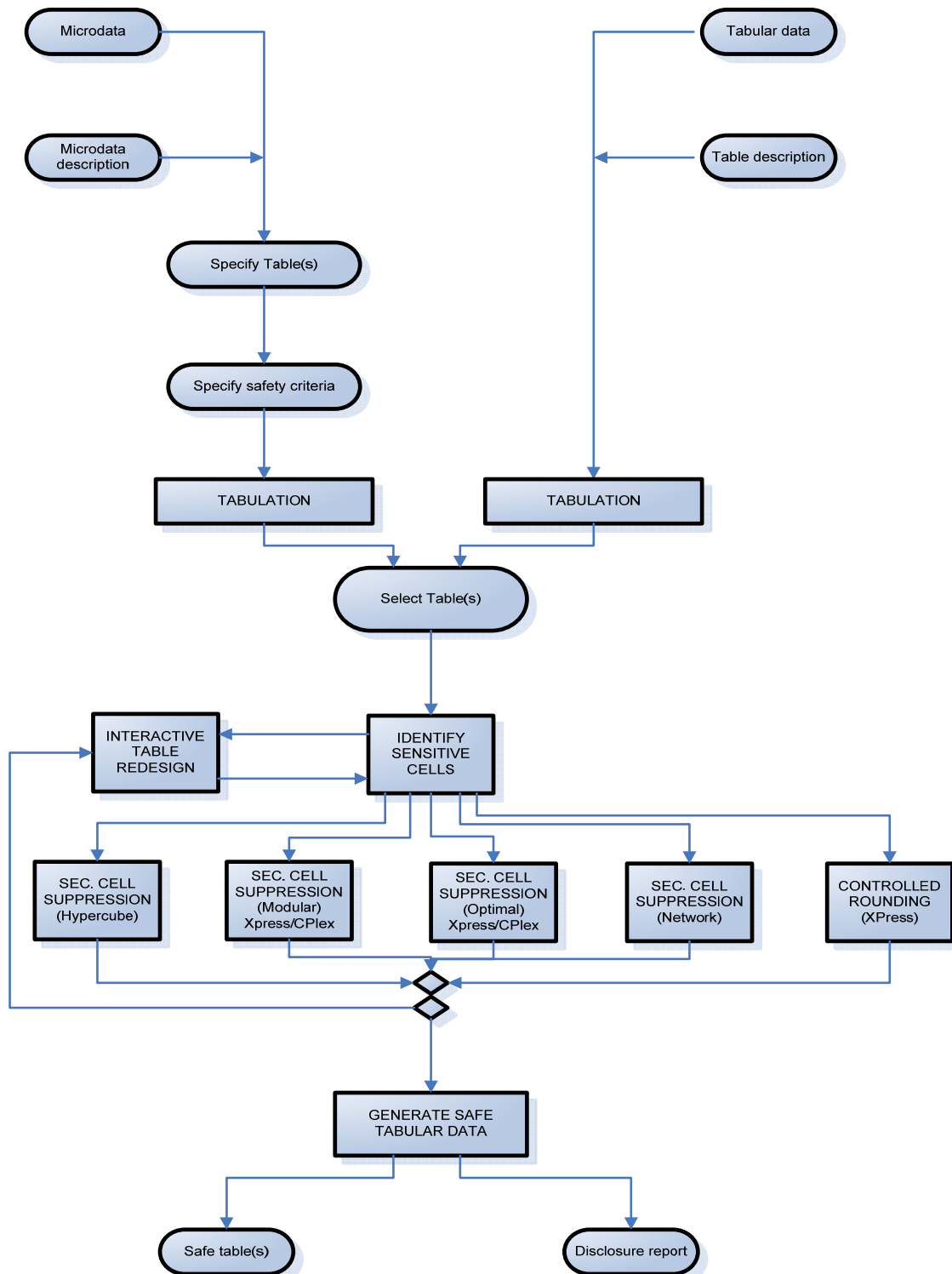
Εφαρμογή τ-Argus

3.1 Εισαγωγή

Σκοπός της εφαρμογής τ-Argus είναι η προστασία των πινάκων δεδομένων έναντι του κινδύνου της αποκάλυψής τους, για παράδειγμα η συμπτωματική ή η εσκεμμένη αποκάλυψη πληροφορίας σχετική με τους στατιστικούς πίνακες από άτομα (individuals) ή οργανισμούς. Αυτό επιτυγχάνεται με τέτοια διαμόρφωση του πίνακα ώστε να περιέχει πληροφορία που να είναι κατά το δυνατό λιγότερο λεπτομερής. Το τ-Argus επιτρέπει έναν αριθμό από αλλαγές στον εκάστοτε πίνακα δεδομένων: ένας πίνακας μπορεί να επανασχεδιαστεί, με την έννοια του συνδυασμού των στηλών και των γραμμών μεταξύ τους; τα ευαίσθητα κελιά μπορούν αν αποσιωπηθούν (suppressed) και τα υπόλοιπα κελιά μπορούν να παρουσιαστούν με βέλτιστο τρόπο (αποσιώπηση δευτέρου βαθμού - secondary cell suppression). Η εφαρμογή τ-Argus αποτελεί το ένα μέλος ενός διδύμου που έχει να κάνει με τον έλεγχο αποκάλυψης. Το δε δεύτερο μέλος είναι το μ-Argus (αντίστοιχο της εφαρμογής τ-Argus για την επεξεργασία microdata).

Η εφαρμογή τ-Argus αποτελεί ουσιαστικά προϊόν του 5^{ου} framework CASC (Computational Aspects of Statistical Confidentiality) project. Πρόκειται για ένα έργο συνέχεια του 4^{ου} Framework SDC-project που υλοποιείται από το Computational Aspects of Statistical Confidentiality (CASC) - στο οποίο μετέχουν εθνικά στατιστικά ινστιτούτα, οργανισμοί και εταιρίες από όλη την ευρωπαϊκή ένωση. Κύρια έμφαση στο έργο αυτό δίνεται στην ανάπτυξη πρακτικών εργαλείων και κυρίως της σειράς του Argus-λογισμικού.

Οι «βασικές λειτουργίες» που παρέχει το λογισμικό τ-ARGUS προκειμένου να γίνει η προετοιμασία των δεδομένων και να γίνουν αυτά «ασφαλή» φαίνονται στο ακόλουθο σχήμα:



Σχήμα 1: Βασικές λειτουργίες της εφαρμογής τ-Argus

3.2 Χαρακτηριστικά

Το τ-Argus είναι μια MS-Windows εφαρμογή η υλοποίηση της οποίας έχει γίνει σε MS C/C++, συνάμα συνοδεύεται με μια γραφική διεπαφή (user interface). Τα αρχεία εισόδου (input files) της εφαρμογής είναι:

1. Τ-Argus αρχείο εντολών (command file) το οποίο προσδιορίζει τις ακριβείς εργασίες που πρέπει να εκτελεστούν στα αρχεία εισόδου.
2. Αρχείο δεδομένων (data file), το οποίο περιλαμβάνει τα μη επεξεργασμένα δεδομένα (όπως για παράδειγμα δεδομένα με αυξημένο κίνδυνο αποκάλυψης). Συγκεκριμένα υπάρχουν δύο τύποι δεδομένων: τα πινακοειδή (tabular) και τα μικροδεδομένα (microdata). Το αρχείο δεδομένων είναι τύπου Comma Separated Value – CSV, δηλαδή οι τιμές σ' αυτό διαχωρίζονται με την χρήση του χαρακτήρα «;».
3. Αρχείο μεταδεδομένων (metadata file), το οποίο περιγράφει τα περιεχόμενα του αρχείου δεδομένων.
4. Προαιρετικά, χρησιμοποιείται ως αρχείο εισόδου και το αρχείο ιεραρχίας (hierarchy file) το οποίο με τη σειρά του καθορίζει την ιεραρχική οργάνωση των δεδομένων.

Τα αρχεία εξόδου της εφαρμογής είναι:

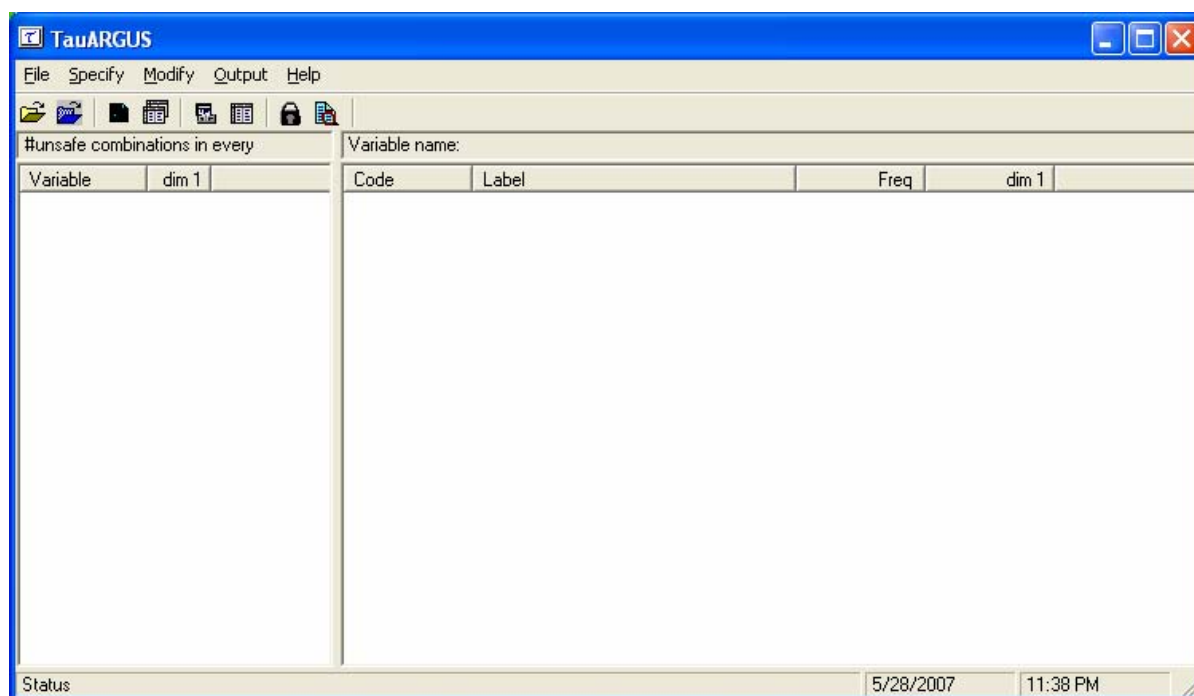
1. Αρχείο αποτελεσμάτων (results file), το οποίο είναι σε CSV μορφή.
2. Αρχείο ενημέρωσης (log file), το οποίο περιέχει συγκεκριμένη πληροφορία εξόδου που ενημερώνει τον χρήστη σχετικά με την κατάσταση των υπολογισμών (calculations) καθώς και για το αν και κατά πόσον η διαδικασία επεξεργασίας έληξε ομαλά.
3. Αρχείο αναφοράς (report file), το οποίο είναι της μορφής HTML και περιγράφει τόσο τα αρχεία εισόδου όσο και τα αρχεία με τα αποτελέσματα της επεξεργασίας καθώς επίσης περιλαμβάνει πληροφορίες σχετικά με το βαθμό σπουδαιότητας των αποτελεσμάτων (όπως για παράδειγμα η αναφορά των συγκεκριμένων κανόνων ασφάλειας που εφαρμόζονται στο τ-Argus εάν μια στατιστική παράμετρος δεν παρουσιάζει κίνδυνο αποκάλυψης).

4. Μια άλλη σημαντική θεώρηση είναι ότι δεν είναι γνωστό εάν η εφαρμογή τArgus είναι thread-safe, το οποίο σημαίνει ότι έχει τη δυνατότητα να εκτελούνται παράλληλα 2 ή περισσότερα instances του τArgus στον ίδιο υπολογιστή και μάλιστα χωρίς το ένα instance να επηρεάζει την εκτέλεση του άλλου/άλλων. Όσον αφορά την thread-safe εκτέλεση εφαρμογών σε περιβάλλον MS-Windows, ένα σημαντικό θέμα είναι πως οι εφαρμογές χρησιμοποιούν τις Dynamic Link Libraries – DLL, καθώς και το πώς τα DLLs έχουν σχεδιαστεί και υλοποιηθεί ώστε να εξασφαλίζουν την ιδιότητα του thread-safe.

Επειδή μάλιστα τα DLLs περιέχουν κώδικα και δεδομένα, μπορούν να διαμοιραστούν μεταξύ διαφορετικών εφαρμογών καθώς επίσης και στα διαφορετικά instances της ίδιας εφαρμογής. Ταυτόχρονα τα DLLs απαιτούν ειδικό σχεδιασμό και τεχνικές ανάπτυξης προκειμένου να χαρακτηριστούν ως thread-safe. Ειδικά, εάν δύο instances της ίδιας εφαρμογής κάνει χρήση του ίδιου non-thread-safe DLL, τότε η ενδεχόμενη απώλεια ή φθορά των δεδομένων είναι αναμενόμενη, με αποτέλεσμα την εξόρυξη εσφαλμένων αποτελεσμάτων.

Για τους σκοπούς της εν λόγω εργασίας, η υπόθεση ασφαλείας είναι ότι η εφαρμογή τ-Argus δεν είναι thread-safe. Αυτό σημαίνει ότι μόνο ένα instance της εφαρμογής θα μπορεί να εκτελείται κάθε φορά από τον BB server.

Τέλος, η εφαρμογή τ-Argus είναι διαθέσιμη τόσο στην interactive όσο και στην batch μορφή της.



Σχήμα 2: Interactive μορφή της εφαρμογής τ-Argus

Κεφάλαιο 4°

Αρχιτεκτονική

4.1 Εισαγωγή

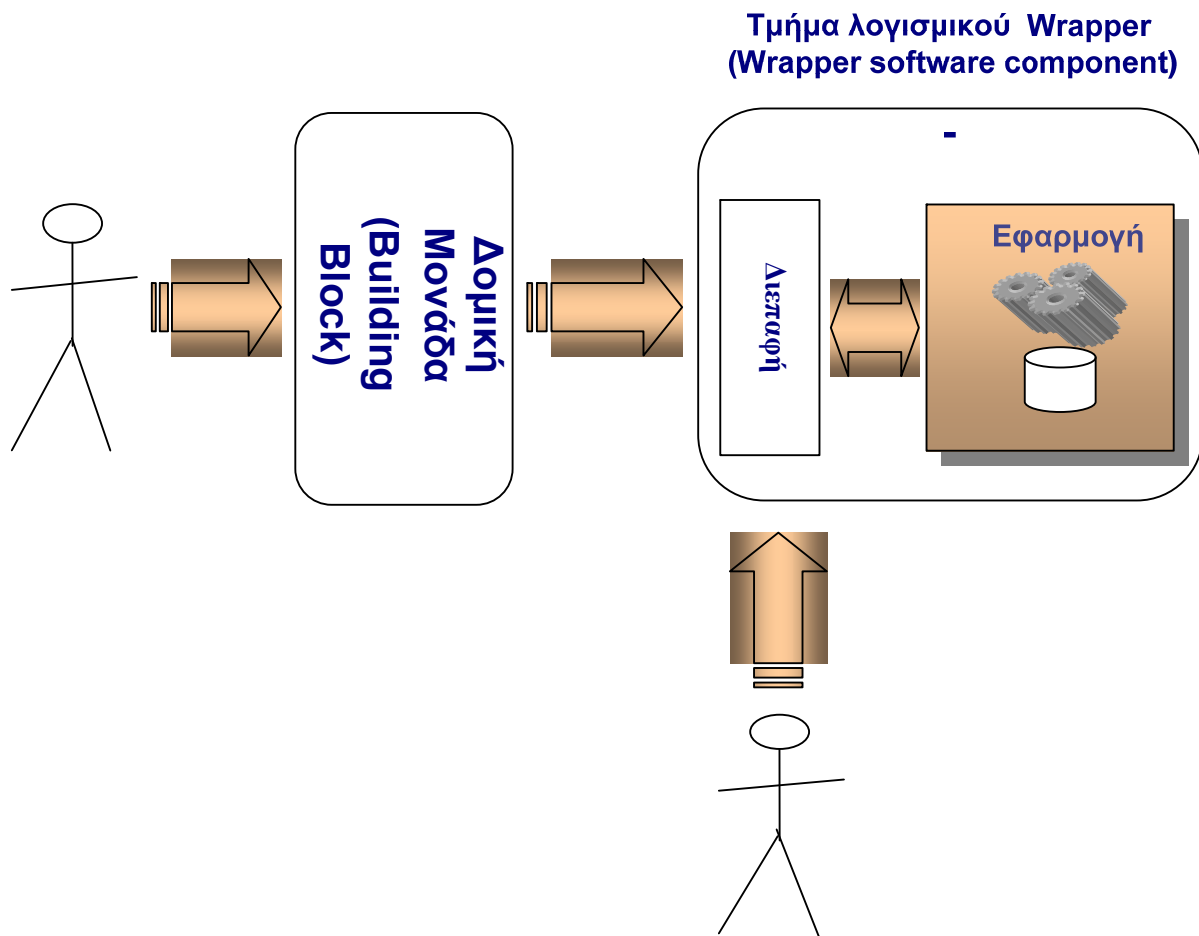
Το **Δομικό Τμήμα -ΔΤ-** ή **Building Block-BB-**, όπως επιλέχτηκε να ονομαστεί, αποτελεί ουσιαστικά το αντικείμενο υλοποίησης της παρούσας εργασίας, είναι ουσιαστικά μια εφαρμογή δέσμης λογισμικού (batch software) η οποία έχει σκοπό να βοηθήσει το χρήστη να προστατέψει την εμπιστευτικότητα των διεσπαρμένων στατιστικών δεδομένων (confidentiality of disseminated statistical data). Το Δομικό Τμήμα λοιπόν, είναι λογισμικό (software) που περιβάλλει την αρχική εφαρμογή τ-Argus. Έτσι μπορεί να παρέχει και να εξασφαλίζει επικοινωνία με άλλες εφαρμογές χωρίς να λαμβάνονται υπόψιν τα ιδιαίτερα χαρακτηριστικά της αρχικής εφαρμογής τ-Argus -όπως η γλώσσα προγραμματισμού στην οποία είναι ανεπτυγμένη, εισαγωγή των αρχείων δεδομένων με τρόπο που να υπακούει στις συνθήκες της αρχικής εφαρμογής, κτλ. . Ο διεθνής τεχνικός όρος στην περιοχή της πληροφορικής, κατάλληλος για να περιγράψει τις παραπάνω ιδιότητες είναι ο λεγόμενος wrapper. Η χρήση του wrapper εξασφαλίζει και ενισχύει την επαναχρησιμοποίηση του εκάστοτε προγράμματος λογισμικού.

4.2 Wrapper

Wrapper ονομάζεται μια δομή δεδομένων ή λογισμικού το οποίο περιέχει ("wraps around") άλλα δεδομένα ή προγράμματα λογισμικού, έτσι ώστε τα υπάρχοντα στοιχεία (elements) των αρχικών δομών-εφαρμογών να εξακολουθούν να υφίστανται και στο νέο σύστημα. Ο όρος συχνά χρησιμοποιείται για τμήματα λογισμικού (component software), όπου ο wrapper τοποθετείται γύρω από την αρχική-κύρια ρουτίνα (legacy routine) κάνοντας την να συμπεριφέρεται σαν αντικείμενο (object). Αυτή η λειτουργία του wrapper είναι επίσης γνωστή και σαν "encapsulation" ή "wrapper encapsulation".

Στην εν λόγω εργασία, ο wrapper είναι το κομμάτι εκείνο του λογισμικού το οποίο περικλείει την αρχική εφαρμογή τ-Argus και εν συνεχεία μέσω της υιοθέτησης της τεχνολογίας των Web Services την κάνει διαθέσιμη μέσω του διαδικτύου.

Όπως γίνεται αντιληπτό και από το ακόλουθο σχήμα, ο όρος wrapper αναφέρεται στο ότι η εφαρμογή μπορεί να γίνει διαθέσιμη (interfaced) ορίζοντας απλά ένα σύνολο ιδιοτήτων, μεθόδων και γεγονότων προκειμένου να συνδεθεί με άλλες εφαρμογές, ανεξαρτήτως τεχνολογίας, πλατφόρμας, κτλ.



Σχήμα 3: Απεικόνιση του wrapper στα πλαίσια της υλοποίησης του Δομικού Τμήματος (BB)

4.3 Γενική Αρχιτεκτονική

Δοθέντος ότι η εφαρμογή τArgus, εξ'ορισμού παρέχει μια διεπαφή βασισμένη στη χρήση αρχείων (file-based interface), αυτός είναι και ο τρόπος που θα πρέπει να λειτουργήσει η εφαρμογή δέσμης

λογισμικού (batch application). Η φυσική και επόμενη επιλογή για την διεπαφή του ΔΤ του τ-Argus είναι αυτή ακριβώς που βασίζεται στην χρήση αρχείων. Με άλλα λόγια το Δομικό Τμήμα τ-Argus θα λαμβάνει από τον πελάτη ΔΤ (BB client) τα απαιτούμενα αρχεία εισόδου, κατόπιν θα καλεί την εφαρμογή τ-Argus να εκτελέσει την υπό εξέλιξη εργασία και τέλος να συλλέξει τα (τρία) αρχεία εξόδου του τ-Argus output (3 files), τα οποία και θα προωθηθούν πίσω στον BB client.

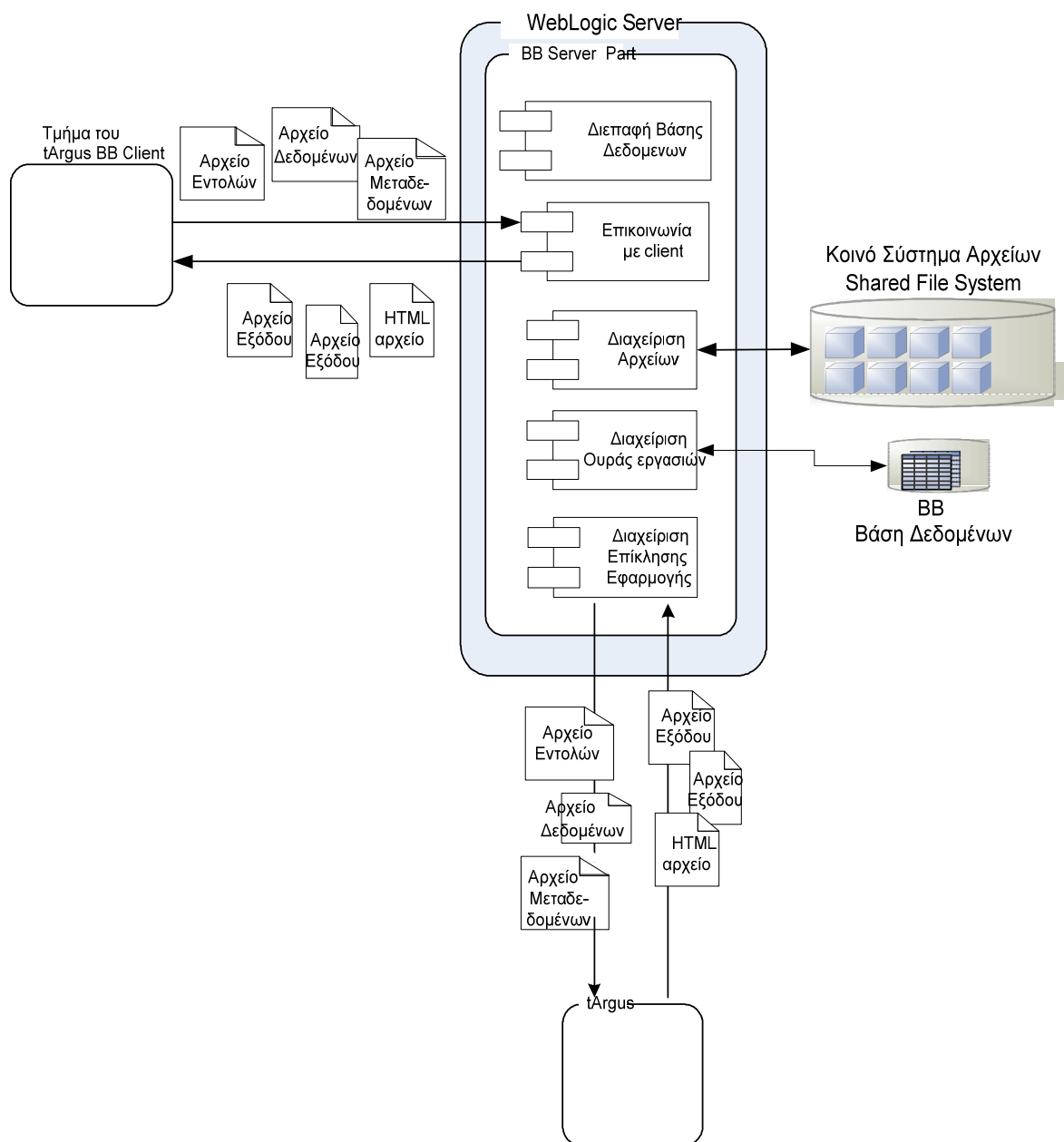
Η γενική αρχιτεκτονική του ΔΤ παρουσιάζεται στο Σχήμα 4: Γενική (generic) αρχιτεκτονική της προσέγγισης του Δομικού Τμήματος (BB). Το τμήμα του πελάτη ΔΤ (BB client) στέλνει στον BB server τα 3 (ή και περισσότερα) τ-Argus αρχεία εισόδου. Κατόπιν ο BB server δημιουργεί ένα χρονοδιάγραμμα για την εκτέλεση της εκάστοτε τ-Argus εργασίας -job- (παράλληλα συντηρείται μια ουρά εργασιών - job queue), καλεί το τ-Argus και εν συνεχεία συλλέγει τα αποτελέσματα της επεξεργασίας, δηλαδή τα 3 αρχεία εξόδου, τα οποία και επιστρέφονται στον πελάτη (client).

Όπως φαίνεται και στο σχήμα που ακολουθεί, το τμήμα του τ-Argus εξυπηρετητή του Δομικού Τμήματος αποτελείται από τα ακόλουθα μέρη:

- **Διεπαφή βάσης δεδομένων (Database interface).** Είναι το τμήμα εκείνο του ΔΤ που είναι υπεύθυνο για την εισαγωγή και εξαγωγή δεδομένων, η εξόρυξη των οποίων γίνεται αντίστοιχα από τα αντίστοιχα αρχεία εισόδου-εξόδου).
- **Επικοινωνία πελάτη (Client communication).** Η διαχείριση όλων των διαδικασιών επικοινωνίας του πελάτη του ΔΤ με τον αντίστοιχο εξυπηρετητή γίνεται μέσα από αυτή την υπομονάδα.
- **Διαχείριση αρχείου (File management).** Το Δομικό Τμήμα τ-Argus χρειάζεται να επεξεργάζεται αρχεία και αυτή η εργασία αναλαμβάνεται από την υπομονάδα διαχείρισης αρχείου.
- **Διαχείριση ουράς εργασιών (Job queue management).** Στα πλαίσια του ΔΤ θα πρέπει να υλοποιηθεί μεταξύ των άλλων και ένας μηχανισμός για την διαχείριση της εκτέλεσης των εργασιών που πρόκειται να προωθηθούν και να εκτελεστεί από το τ-Argus. Για λόγους συνέπειας (persistency) και σταθερότητας (robustness), το τ-Argus

Δομικό Τμήμα θα κάνει χρήσης ενός RDBMS. Στην εν λόγω βάση δεδομένων πρόκειται να αποθηκευτούν δεδομένα τα οποία είναι σχετικά με το χρονοδιάγραμμα και την εκτέλεση της εκάστοτε τArgus εργασίας (job).

- **Διαχείριση επίκλησης εφαρμογής (Application invocation management).** Αυτή η υπομονάδα είναι υπεύθυνη για την διαχείριση της εκάστοτε ενεργής επίκλησης (actual invocation) της τ-Argus MS-Windows εφαρμογής.



Σχήμα 4: Γενική (generic) αρχιτεκτονική της προσέγγισης του Δομικού Τμήματος (BB)

4.4 Επικοινωνία μεταξύ πελάτη-εξυπηρετητή (client-server) του Δομικού Τμήματος τ-Argus

Οι διαθέσιμες επιλογές προκειμένου να επιτευχθεί η μεταφορά αρχείων μεταξύ των client-server, περιγράφονται παρακάτω:

- **Web service.** Βασίζεται στην χρήση του HTTP ως πρωτόκολλο μεταφοράς. Το WSDL αρχείο που αντιστοιχεί στο εκάστοτε service επιτρέπει τόσο τους clients όσο και τα services να χρησιμοποιούν τεχνολογίες κατά πολύ διαφορετικές προκειμένου να αντιστοιχηθούν και να μετατραπούν κατάλληλα στα αντίστοιχα αντικείμενα δεδομένων (data objects).
- **RMI/IIOP.** Οι J2EE application μπορούν επίσης να κάνουν χρήση του RMI/IIOP για να επιτύχουν απομακρυσμένες κλήσεις (remote calls) μέσω του διαδικτύου. Το RMI/IIOP λειτουργεί πάνω σε ένα Java-based περιβάλλον, το οποίο πρακτικά δεν είναι εφικτό για non-Java clients.
- **Java Message Service (JMS).** Το JMS είναι ένας άλλος τρόπος που μπορούν να υιοθετήσουν οι J2EE clients για να επικοινωνήσουν με τον αντίστοιχο server της εφαρμογής. Μειονέκτημα του JMS είναι ότι η λειτουργία του περιορίζεται στα J2EE περιβάλλοντα.

Λαμβάνοντας υπόψιν τα παραπάνω στοιχεία, η μέθοδος που θεωρήθηκε προτιμότερη για την υλοποίηση της επικοινωνίας BB client/server μέσω Web services, είναι η χρήση ουσιαστικά του SOAP πάνω σε HTTP πρωτόκολλο.

4.5 Μεταφορά αρχείων μεταξύ πελάτη-εξυπηρετητή (client-server)

Η γενική αρχιτεκτονική του λογισμικού του τ-Argus ΔΤ βασίζεται σε μεγάλο βαθμό στην μεταφορά αρχείων μεταξύ πελάτη και εξυπηρετητή. Τα δε διαθέσιμα μέσα για μεταφορά αρχείων είναι τα ακόλουθα:

- 1 **Αντιγραφή αρχείου μέσω διαδικτύου (Network file copy).** Δεδομένου ότι υπάρχει ένα σύστημα αρχείων (file system) στο οποίο μπορούν να έχουν πρόσβαση τόσο ο client όσο και ο server για να διαβάσουν/γράψουν, τότε η μεταφορά αρχείων μεταξύ αυτών μπορεί να υλοποιηθεί σαν μια απλή εντολή αντιγραφής αρχείου.

- 2 **FTP.** Ένας FTP server μπορεί να χρησιμοποιηθεί έτσι ώστε ο τ-Argus BB client να μπορεί να «ανεβάσει» (upload) τα αρχεία εισόδου αλλά και να «κατεβάσει» τα αντίστοιχα αρχεία εξόδου.
- 3 **HTTP.** Η περίπτωση του HTTP παρουσιάζει πολλές ομοιότητες με τη μέθοδο του FTP server, όπου ομοίως ο client έχει τη δυνατότητα να «ανεβάζει» αρχεία εισόδου μέσω του HTTP καθώς επίσης και να «κατεβάζει» τα αρχεία εξόδου του τ-Argus.
- 4 **SOAP attachments.** Τα SOAP attachments επιτρέπουν την μεταφορά αρχείων μεταξύ των client και server.

Μεταξύ των τεσσάρων διαφορετικών επιλογών προαναφέρθηκαν παραπάνω, οι πρώτες τρεις έχουν το μειονέκτημα ότι χρησιμοποιούν διαφορετικά πρωτοκόλλα επικοινωνίας (π.χ. FTP, HTTP) από αυτά που παραδοσιακά χρησιμοποιούν τα Web services (SOAP and SAX RPC). Επιπλέον, δεν αποκλείονται οι ασυνέπειες, δεδομένου ότι μαζικές αλληλεπιδράσεις μεταξύ client και server γίνονται μέσω του πρωτοκόλλου HTTP/SOAP, έτσι είναι προτιμότερο η μεταφορά αρχείων να γίνεται μέσω άλλων πρωτοκόλλων ή εναλλακτικού server component (όπως για παράδειγμα FTP server ή HTTP server) τρόπος ο οποίος διαφέρει σαφώς από τα Web services. Γι'αυτούς ακριβώς του λόγους, καθώς επίσης και λόγω του ότι δεν υπάρχουν περιορισμοί (π.χ. περιορισμοί που υποβάλλονται λόγω των περιορισμών ασφαλείας), η μεταφορά των αρχείων δεν επιλέχτηκε τελικά να γίνεται με τη χρήση SOAP attachments (web services).

Κεφάλαιο 5°

Βάση Δεδομένων

5.1 Εισαγωγή

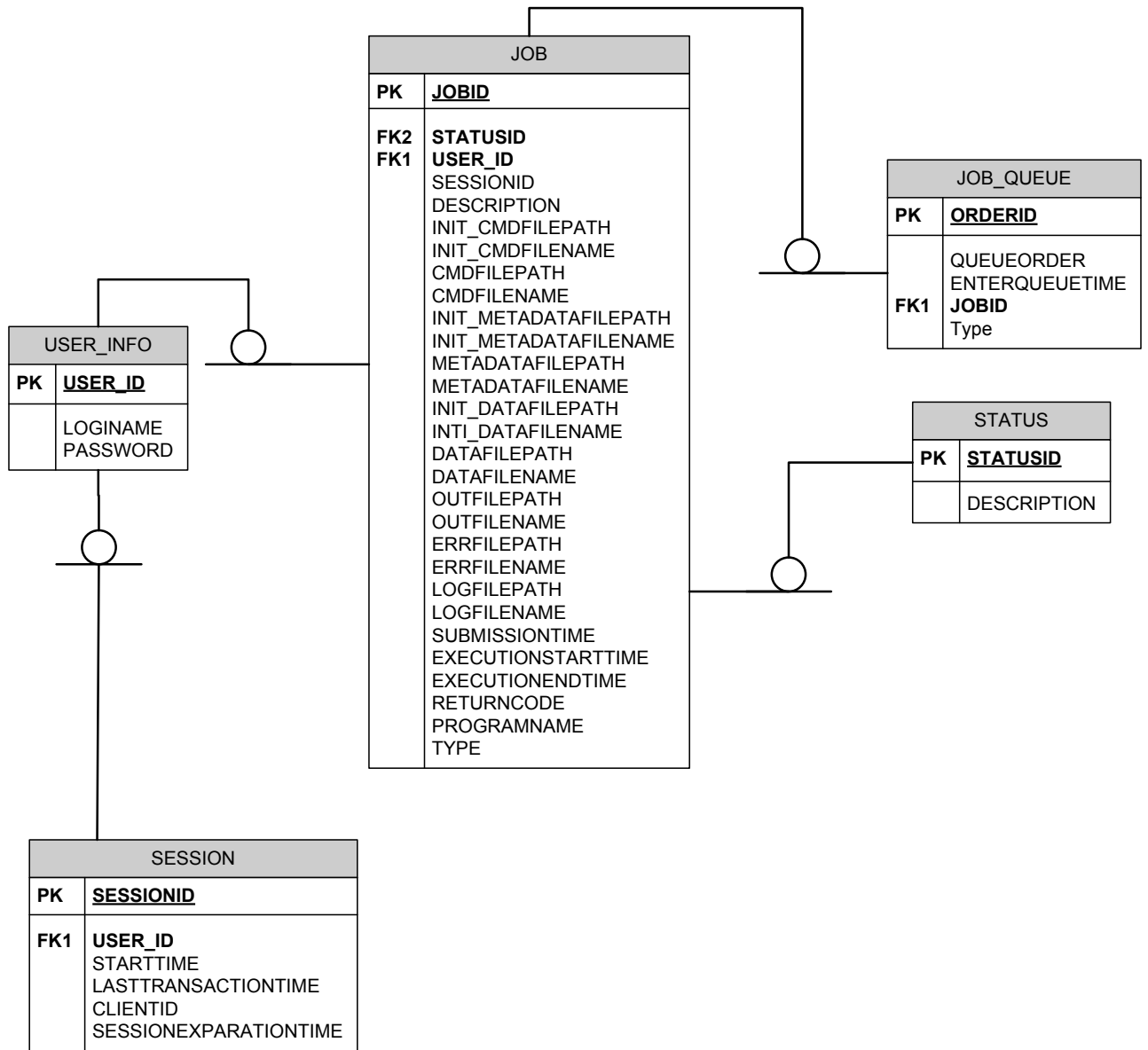
Το Building Block Web Service κάνει χρήση μιας εσωτερικής (internal) βάσης δεδομένων, η οποία έχει ως σκοπό την αποθήκευση των δεδομένων που καλύπτουν τα ακόλουθα πεδία:

1. Εξουσιοδοτημένοι (authorized) χρήστες
2. Εργασίες (jobs) που υποβάλλονται για εκτέλεση (execution)
3. Κατάσταση (status) της διαδικασίας εκτέλεσης της εργασίας
4. Διαχείριση του session του εκάστοτε χρήστη

Η δομή αυτής της βάσης δεδομένων (tables, elements, attributes, etc.) πρόκειται να περιγραφεί στις επόμενες ενότητες.

5.2 Μοντέλο της BB Βάσης Δεδομένων

5.2.1 Διάγραμμα Σχέσεων Οντοτήτων (E-R)



Σχήμα 5: Διάγραμμα E-R BB Βάσης Δεδομένων

Το παραπάνω σχήμα αναπαριστά το διάγραμμα Σχέσεων-Οντοτήτων της βάσης δεδομένων των Building Blocks.

5.2.2 Πίνακες

Ακολουθεί η σύντομη παρουσίαση των πινάκων του ΔΤ Βάσης Δεδομένων.

5.2.2.1 USER

Πίνακας USER:	Αποθηκεύει τα δεδομένα για τους εξουσιοδοτημένους χρήστες που έχουν πρόσβαση (access) στο σύστημα.
- userId: Μοναδικός ταυτοποιητής (unique identifier) για τον εισηγμένο στο σύστημα χρήστη (logged in user)..	
- loginName: Το όνομα του εισηγμένου στο σύστημα χρήστη.	
- password: Ο κώδικας του εισηγμένου στο σύστημα χρήστη.	

5.2.2.2 SESSION

Table SESSION:	Αποθηκεύει τα δεδομένα σχετικά με τα ενεργά (active) sessions (π.χ. conversation state) που ξεκινά κάθε φορά που ο client καλεί το BB Web service
<ul style="list-style-type: none"> - sessionId: Μοναδικός identifier για το session. - userId: Identifier του χρήστη. - startTime: Χρόνος όπου το session δημιουργήθηκε. - lastTransactionTime: Τχρόνος τελευταίας ενέργειας εντός του session. Αρχικά, lastActionTime = startTime. - clientId: Client id. - sessionExpirationTime; Χρόνος λήξης του session. sessionExpirationTime = lastActionTime + προκαθορισμένη χρονική διάρκεια (predefined time duration) 	

5.2.2.3 JOB

Table JOB:	Αποθηκεύει τα δεδομένα των jobs που έχουν υποβάλλονται από τον BB πελάτη (client) προκειμένου να εκτελεστούν από την εφαρμογή.
<ul style="list-style-type: none"> - jobId: Μοναδικός identifier ο οποίος περιγράφει μια εργασία (job). - description: Περιγραφή εργασίας (job description). - init_cmdFilePath: Το αρχικό μονοπάτι (path) του αρχείου (εισόδου) εντολών (command file). - init_cmdFileName: Το αρχικό όνομα του αρχείου (εισόδου) εντολών (command file). - cmdFilePath: Το πλήρες μονοπάτι (path) του αρχείου (εισόδου) εντολών (command file). - cmdFileName: Το όνομα του αρχείου (εισόδου) εντολών (command file). - init_dataFilePath: Το αρχικό μονοπάτι (path) του αρχείου (εισόδου) δεδομένων (data file). - init_dataFileName: Το αρχικό όνομα του αρχείου (εισόδου) δεδομένων (data file). - dataFilePath: Το πλήρες μονοπάτι (path) του αρχείου (εισόδου) δεδομένων (data file). - dataFileName: Το όνομα του αρχείου (εισόδου) δεδομένων (data file). - init_metadataFilePath: Το αρχικό μονοπάτι (path) του αρχείου (εισόδου) μεταδεδομένων (metadata file). - init_metadataFileName: Το αρχικό όνομα του αρχείου (εισόδου) μεταδεδομένων (metadata file). - metadataFilePath: Το πλήρες μονοπάτι (path) του αρχείου (εισόδου) μεταδεδομένων (metadata file). - metadataFileName: Το όνομα του αρχείου (εισόδου) μεταδεδομένων (metadata file). - outFilePath: Το μονοπάτι (path) του αρχείου εξόδου. - outFileName: Το όνομα του αρχείου εξόδου. - errorFilePath: Το μονοπάτι (path) του αρχείου σφάλματος. - errorFileName: Το όνομα του αρχείου σφάλματος. - logFilePath: Το μονοπάτι (path) του log αρχείου (log file) εξόδου. - logFileName: Το όνομα του log αρχείου (log file) εξόδου. - submissionTime: Χρόνος υποβολής της εργασίας (job). - executionStartTime: Χρόνος εκκίνησης της εκτέλεσης του job. - executionEndTime: Ο χρόνος όπου λαμβάνει τέλος η εκτέλεση του job. - returnCode: Ο κωδικός αριθμός (code) που επιστρέφεται από την εφαρμογή μετά την εκτέλεση. - statusId: Η τρέχουσα κατάσταση της εργασίας (job), κατά τη διάρκεια της διεργασίας. - user_Id: Το id του χρήστη. 	

5.3 Oracle 10g Database

Η Building Block βάση δεδομένων υλοποιείται σε Oracle 10g, αφού η oracle είναι ευρέως αποδεκτό ότι αποτελεί μια από της πλέον σημαντικές τεχνολογίες στην περιοχή των βάσεων δεδομένων.

Η Oracle 10g παρέχει κλιμάκωση (proven scalability) στη διαμόρφωση όλων των hardware και είναι διαθέσιμη σε άλλα τα υποστηριζόμενα από την Oracle λειτουργικά συστήματα (συμπεριλαμβανομένου των Windows, Linux και Unix) και μπορεί να χρησιμοποιηθεί για τη διαχείριση πολύ μεγάλων μεγεθών πληροφορίας, παρέχοντας παράλληλα υψηλό βαθμό εγγύησης της παρεχόμενης ασφάλειας.

Εξάλλου η χρήση της Oracle παρέχει μοναδικά πλεονεκτήματα που έχουν να κάνουν με την προστασία των δεδομένων από επιζήμια ανθρώπινα λάθη καθώς επίσης χαρακτηρίζεται από ικανότητες αυτοδιαχείρισης που βοηθούν να μειωθούν τα λειτουργικά κόστη και επιπλέον παρέχει ανοχή στα σφάλματα που προέρχονται είτε από αστοχίες του υλικού είτε από προγραμματισμένες διακοπές ρεύματος. Οι αυτοματοποιημένες δυνατότητες αποθηκευτικής διαχείρισης (storage management) που αναπτύσσονται εντός της Oracle 10g αντικατοπτρίζουν τα δεδομένα απέναντι στις διαθέσιμες αποθηκευτικές συσκευές προκειμένου να επιτευχθεί η προστασία των δεδομένων έναντι της ενδεχόμενης αποτυχίας αποθήκευσης. Επιπλέον, υποστηρίζει την ανάπτυξη (deployment) ενός μεγάλου αριθμού από χρήστες χρησιμοποιώντας το κλειδίωμα κάθε ξεχωριστής γραμμής-επιπέδου (row level) καθώς επίσης και η συνέπεια των πολλαπλών εκδόσεων (multi-version consistency).

Επιπρόσθετα, η Oracle 10g υποστηρίζει όλους τους standard τύπους σχεσιακών δεδομένων και επίσης την πρόσβαση στα αποθηκευμένα δεδομένα μέσω standard interfaces όπως είναι το JDBC.

Κεφάλαιο 6^ο

Προδιαγραφές του Web Service Δομικού Τμήματος

6.1 Εισαγωγή

Στο παρόν κεφάλαιο πρόκειται να παρουσιαστεί μια πλήρης τεχνική περιγραφή του Web Service που υλοποιείται στα πλαίσια τούτης της εργασίας. Προκειμένου να περιγραφεί πλήρως τεχνικά ένα Web Service απαραίτητη είναι η απαρίθμηση των μεθόδων του, η παρουσίαση του WSDL (Web Service Description Language) αρχείου και προαιρετικά η παρουσίαση των UML διαγραμμάτων συνέχειας (sequence diagrams).

6.2 Μέθοδοι

Ο πίνακας που ακολουθεί παραθέτει τις μεθόδους που απαρτίζουν το BB Web service.

Μέθοδοι	Ορίσματα	Μεταβλητή Επιστροφής	Περιγραφή
TArgus	Χωρίς ορίσματα	Void	Αρχικοποιεί το web service (παραμέτρους, κτλ.).
getUserID	String loginName, String password	String userId	Εξουσιοδοτεί (authenticates) τον χρήστη.
generateSessionID	String userId	String sessionId	Εκκινεί τον διάλογο μεταξύ χρήστη-web service και κατόπιν παράγει ένα μοναδικό session identifier.
runTArgus	String sessionId, String userId, String batchPath, String dataPath, String metadataPath	String jobId	Θέτει τα input file (batch, data, metadata) paths, δημιουργεί ένα νέο job instance και εισάγει το job σε μια λίστα από jobs (queue).

insertJob	String job_desc	Boolean	Εισάγει μια εργασία (job) μέσα στην βάση δεδομένων καθώς και στην ουρά εργασιών.
checkTArgus	String sessionId	TArgusConfig ¹	Επιστρέφει μια λίστα με την κατάσταση των εργασιών που έχουν εκκινηθεί από το χρήστη με συγκεκριμένο session ID.
generateJobId	Χωρίς ορίσματα	String	Παράγει τυχαία (random) id για τα jobs.
getFileConfiguration	String sessionId,String userId,String description,String batchPath,String dataPath,String metadataPath,String logFilePath,boolean isCopy	TArgusConfig	Επιστρέφει πληροφορία (σχετική με τα αρχεία εισόδου/εξόδου) που αφορά τα διαφορα jobs που έχει εκκινήσει ο χρήστης, σε ένα συγκεκριμένο session.
copyInputFiles	TArgusConfig conf,String sessionId,String userId	TArgusConfig	Αντιγράφει τα αρχεία εισόδου σε προκαθορισμένη τοποθεσία, επιστρέφει την σχετική πληροφορία.
copyFiles	File source, File dest	Void	Αντιγράφει ένα αρχείο από μια δοσμένη τοποθεσία σε μια άλλη επίσης προκαθορισμένη.
setOutputDir	String sessionId,File file	Void	Δημιουργεί το directory εξόδου στο οποίο θα τοποθετηθούν τα αρχεία

¹ TArgusConfig είναι μια δομή (κλάση) που περιγράφει πλήρως ένα τ-Argus job (ονόματα και τοποθεσίες αρχείων, χρονική στιγμή υποβολής εργασίας, χρονική στιγμή εκκίνησης επεξεργασίας, χρονική στιγμή τέλους επεξεργασίας, αποτέλεσμα επεξεργασίας, κτλ.).

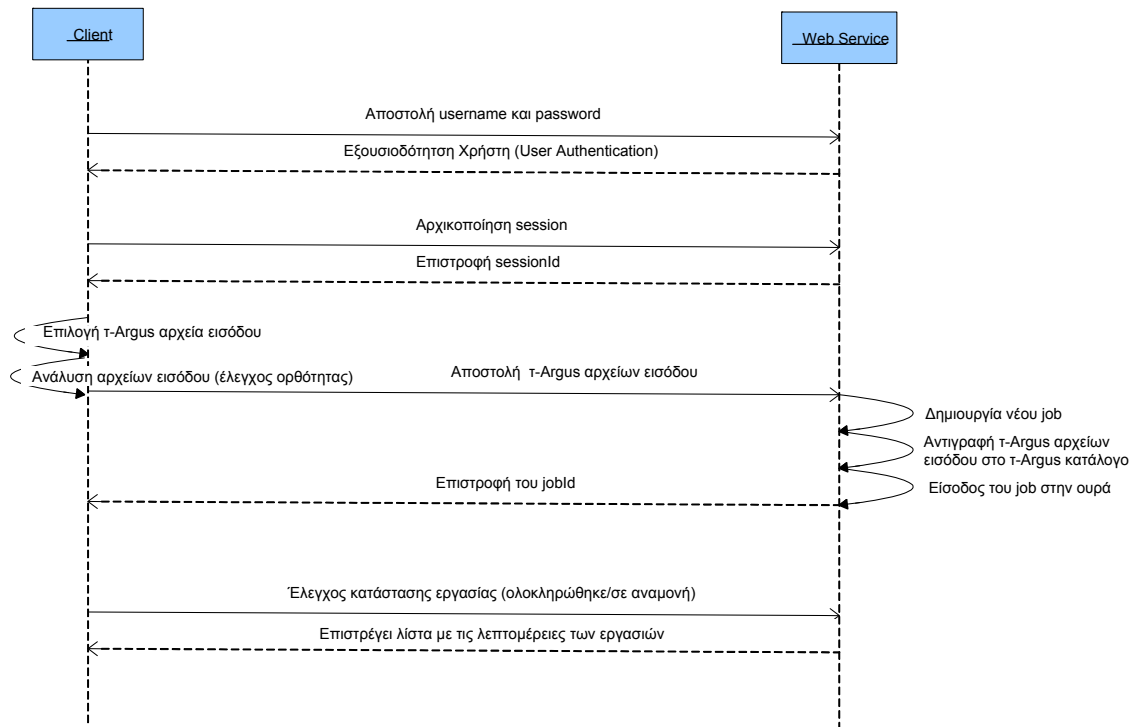
			εξόδου.
<i>Log</i>	String msg	Void	Τυπώνει μηνύματα εξόδου.
<i>Created</i>	Χωρίς ορίσματα	Void	Ανοίγει τη σύνδεση με τη βάση.
<i>writeFileTArgusResults</i>	String sessionId,String user,String loginName	Void	Δημιουργεί ένα αρχείο (report) με λεπτομέρειες για τις εργασίες (jobs) του δεδομένου χρήστη για το συγκεκριμένο session.

Πίνακας 1: Μέθοδοι του BB Web ServiceΣημείωση

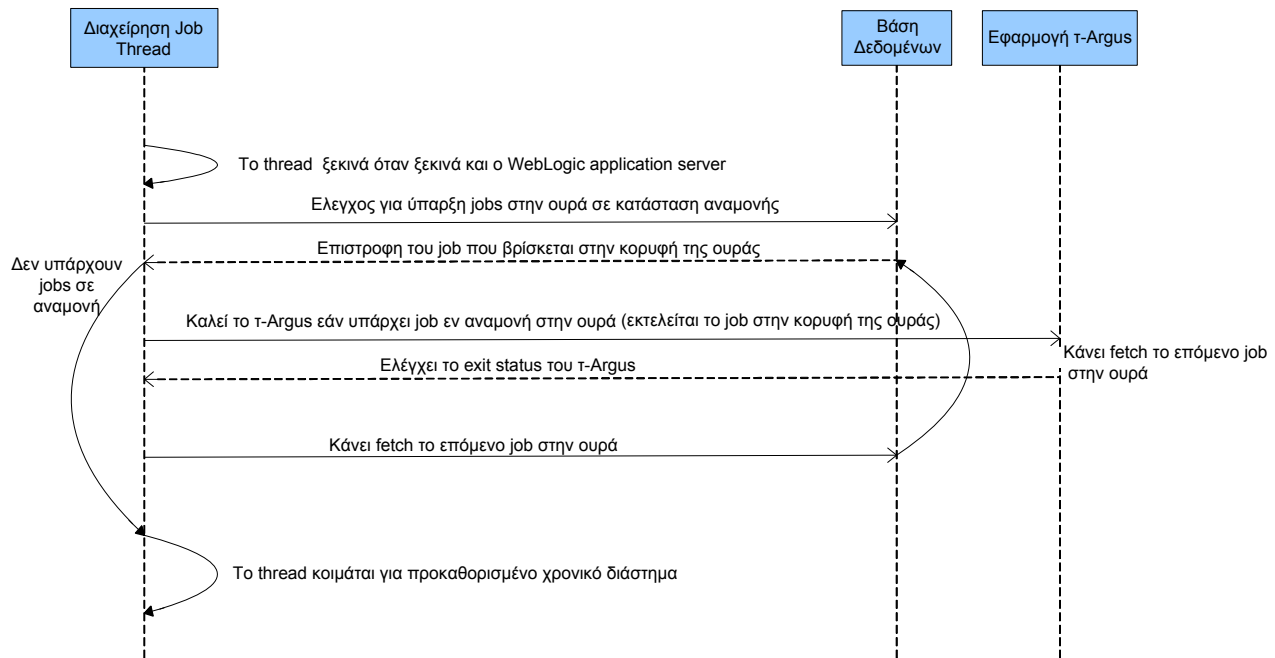
Με κανονική γραμματοσειρά σημειώνονται οι public μέθοδοι, ενώ με italics οι αντίστοιχες private μέθοδοι του web service

6.3 UML Sequence Diagrams

Τα ακόλουθα σχήματα παρουσιάζουν τα UML sequence διαγράμματα της αλληλεπίδρασης του client με το Web Service καθώς επίσης και τη διαχείριση ουράς των τ-Argus εργασιών αντίστοιχα.



Σχήμα 6: Sequence diagram αλληλεπίδρασης client με Web Service



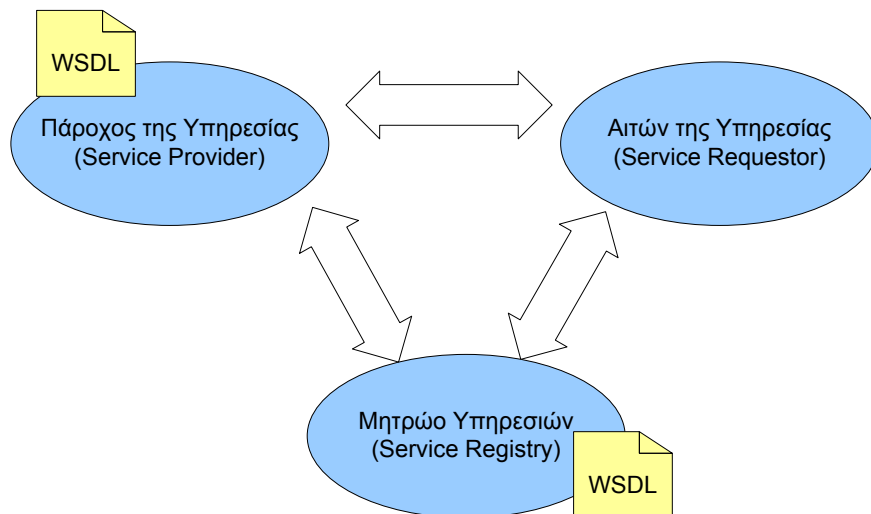
Σχήμα 7: Διαχείριση της ουράς των t-Argus εργασιών (job queue)

6.4 Αρχείο WSDL

Το WSDL ή Web Services Description Language είναι ένα αρχείο σε XML μορφή (format) για την περιγραφή διαδικτυακών υπηρεσιών σαν ένα σύνολο από endpoints που βασίζουν τη λειτουργιά τους σε μηνύματα που περιέχουν πληροφορία προσανατολισμένη είτε στα έγγραφα είτε στις διαδικασίες (document-oriented or procedure-oriented information). Οι λειτουργίες (operations) και τα μηνύματα (messages) περιγράφονται περιληπτικά, και υπακούουν σε ένα αυστηρώς συγκεκριμένο διαδικτυακού πρωτόκολλο όπως επίσης και σε συγκεκριμένη μορφή μηνυμάτων που ορίζεται στο εκάστοτε endpoint. Τα δε endpoints που σχετίζονται μεταξύ τους συνδυάζονται σε ενιαίο αφηρημένα endpoints (services). Το WSDL μπορεί επίσης να επεκταθεί ώστε να επιτρέψει την περιγραφή των εν λόγω endpoints καθώς και των μηνυμάτων τους σε σχέση με το ποιό είναι το format του μηνύματος ή το πρωτόκολλο δικτύου που χρησιμοποιείται για την επικοινωνία. Τα WSDL αρχεία συγκεντρώνουν παλαιότερες προτάσεις που είχαν αναπτυχθεί σε αυτή την περιοχή μεταξύ των οποίων τα NASSL, SCL and SDL.

Το διάγραμμα που ακολουθεί δείχνει πώς τα WSDL έγγραφα ενσωματώνονται με την αρχιτεκτονική των web services. Τα WSDL έγγραφα, τα οποία βρίσκονται στον πάροχο (provider) και στο μητρώο

(registry), είναι προσπελάσιμα από τον αιτών του service (requestor) ο οποίος επιθυμεί να χρησιμοποιήσει αυτό το service.



Σχήμα 8: WSDL και Web Services

Στον πίνακα που ακολουθεί επιχειρείται η αντιστοιχία των μεταδεδομένων των Building Blocks με τα αντίστοιχα στοιχεία (elements) του WSDL εγγράφου.

Μεταδεδομένα των Δομικών Μερών (BB)	Στοιχεία του εγγράφου WSDL	Παράδειγμα
Όνομα του service	στοιχείο <service>, χαρακτηριστικό (attribute): name	<service name="TArgusBB">
Προβλεπόμενες μέθοδοι (Provided Functions)	<operation> υποστοιχείο (subelement) του στοιχείου <portType>	<portType name="TArgusService"> <operation name="runTArgus"> <input message="runTArgusRequest"> <output message="runTArgusResponse"> <fault message="TArgusException"> </operation> </portType>
Παράμετροι εισόδου (απαιτούνται)	<message> (το στοιχείο <types> ορίζει τους τύπους δεδομένων που χρησιμοποιούνται στο στοιχείο <message>)	<types> <complexType name="TArgusConfiguration"> <sequence> <element name="TArgusHomeDir" type="xsd:string"> <element name="logFileName" type="xsd:string"> </sequence> </complexType> </types> <message name="getConfigurationResponse">

		<pre> <part name="fault" type="impl:TArgusConfiguration" /> </message> </pre>
Παράμετροι Εξόδου (παράμετροι επιστροφής)	<pre> <message>, (το στοιχείο <types> ορίζει τους τύπους δεδομένων που χρησιμοποιούνται στο στοιχείο <message>) </pre>	<pre> <types> <complexType name="TArgusException"> <sequence> <element name="detailMessage" type="xsd:string"> <element name="errorMessage" type="xsd:string"> </sequence> </complexType> </types> <message name="TArgusException"> <part name="fault" type="impl:TArgusException" /> </message> </pre>
Ακριβής Φυσική Τοποθεσία (Actual Physical Location)	στοιχείο <service>, υποστοιχείο <port>	<pre> <port name="TArgusServicePort" binding="impl:TArgusServicePortSoapBinding"> <soap:address location="http://jsword- technologies:7001/bblocks/services.TArgusService /"> </port> </pre>

Πίνακας 2: Αντιστοίχιση μεταξύ των μεταδεδομένων των Block's και των στοιχείων του WSDL εγγράφου

Στην ενότητα Παράρτημα (στο τέλος της εργασίας) παρατίθεται το WSDL αρχείο το οποίο ορίζει πλήρως το Web Service που υλοποιείται στα πλαίσια αυτής της εργασίας .

Κεφάλαιο 7^ο

Σχεδιασμός του Client του Web Service Δομικού Τμήματος (BB)

7.1 Εισαγωγή

7.2 Τεχνολογίες Επικοινωνίας

Η διαλειτουργικότητα είναι το κυριότερο πλεονέκτημα χρήσης των web services όσον αφορά την επικοινωνία. Η χρήση των web services δίνουν στους πελάτες (clients) την δυνατότητα να λειτουργούν αρμονικά με σχεδόν κάθε τύπο συστήματος και εφαρμογής, ανεξάρτητα με το ποια είναι η πλατφόρμα πάνω στην οποία τρέχει το σύστημα ή η εφαρμογή. Επιπλέον, ο client έχει στην διάθεση του μια πληθώρα από τεχνολογίες για να της χρησιμοποιήσει για επικοινωνία.

Υπάρχουν τέσσερις προσεγγίσεις επικοινωνίας οι οποίες είναι διαθέσιμες σε ένα J2EE client για να αποκτήσει πρόσβαση στο έκτοτε service. Οι τέσσερις αυτοί τρόποι επικοινωνίας παρουσιάζονται παρακάτω.

- **Web service.** Τα web services χρησιμοποιούν το πρωτόκολλο HTTP σαν πρωτόκολλο μεταφοράς, το οποίο επιτρέπει του clients να λειτουργούν ομαλά με συστήματα μέσα από firewalls. Το δε WSDL έγγραφο του service επιτρέπει τους clients και τα services τα οποία χρησιμοποιούν διαφορετικές τεχνολογίες να αντιστοιχήσουν και να μετατρέψουν τα αντίστοιχα data objects. Στην περίπτωση των services και των clients που είναι βασισμένοι στο JAX-RPC, το JAX-RPC runtime αναλαμβάνει να διαχειριστεί την αντιστοίχιση.
- **RMI/IIOP.** Οι J2EE clients μπορούν επίσης να κάνουν χρήση του RMI/IIOP για να πραγματοποιούν απομακρυσμένες κλήσεις στην εφαρμογή μέσω του διαδικτύου. Το RMI/IIOP συχνά χρησιμοποιείται από clients που λειτουργούν σε περιβάλλοντα intranet, όπου υπάρχει μεγαλύτερος βαθμός ελέγχου της ανάπτυξης του client καθώς και του J2EE server. Αυτά τα ελεγχόμενα περιβάλλοντα παρέχουν έναν client container ο οποίος διαχειρίζεται την ασφάλεια επικοινωνίας, καθόσον το πέρασμα μέσα firewalls

- **Java Message Service (JMS).** Είναι ένας άλλος τρόπος για τους J2EE clients να επικοινωνούν με τις server εφαρμογές. Το JMS παρέχει έναν τρόπο για ασύγχρονη επικοινωνία.
- **Κατ' απαίτηση εφαρμογές (custom approaches) μέσω HTTP.** Αν και τα Web services παρέχουν έναν καθιερωμένο τρόπο για την εξαγωγή XML εγγράφων μέσω του HTTP, ο προγραμματιστής είναι ελεύθερος να χρησιμοποιήσει και μη καθιερωμένες (nonstandard) προσεγγίσεις. Η επικοινωνία μέσω του HTTP πρωτοκόλλου απαιτεί μόνο απλή υποδομή για την αποστολή και λήψη μηνυμάτων. Ωστόσο ιδιαίτερη προσοχή απαιτείται εάν χρησιμοποιηθεί αυτή η προσέγγιση καθώς δεν είναι καθιερωμένη.

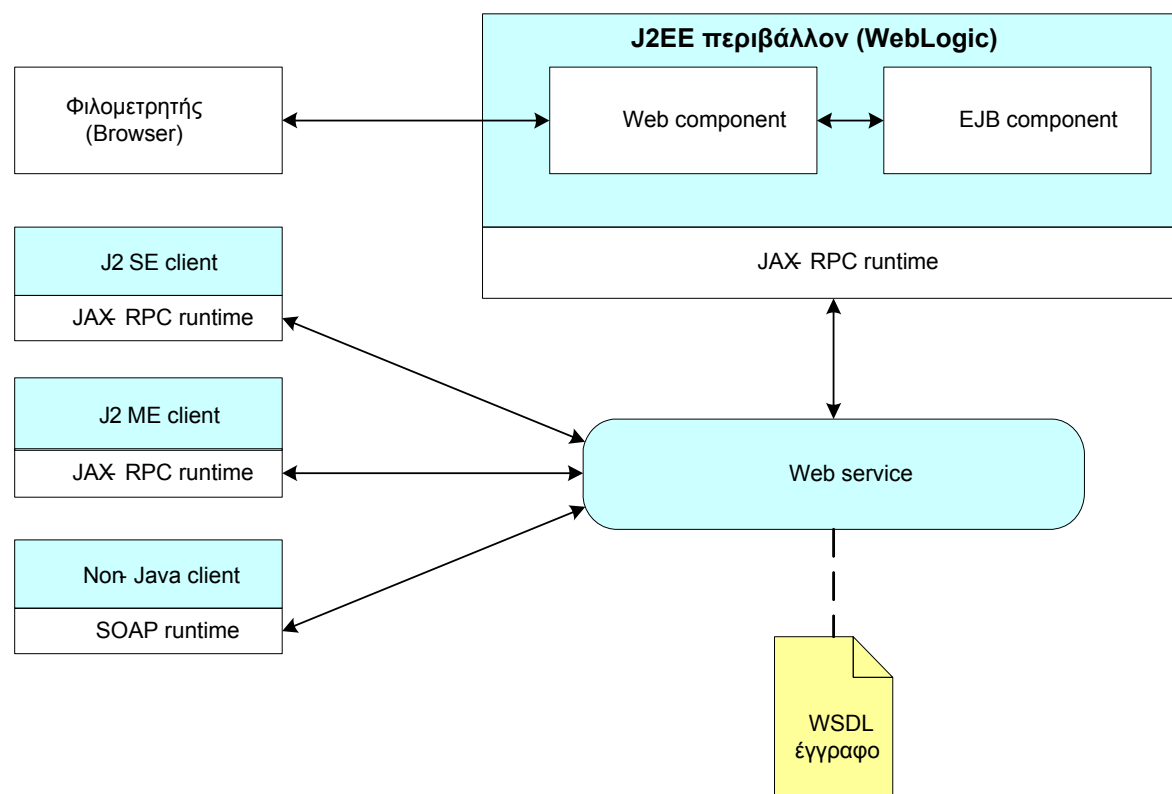
7.3 Web Service Clients

Όπως φαίνεται και στο ακόλουθο σχήμα μπορούν να υπάρξουν τρία εναλλακτικά clients σενάρια, τα οποία και παρουσιάζονται παρακάτω:

1. **J2EE component.** Σε αυτό το σενάριο, ένα J2EE component αποκτά πρόσβαση στο BB web service. Το J2EE component λαμβάνει τα αποτελέσματα από το service και ακολούθως τα μορφοποιεί. Εναλλακτικά, το J2EE μπορεί να χρησιμοποιήσει το BB web service σαν ένα επί μέρους βήμα μια σειράς βημάτων που αφορούν την παραγωγή στατιστικών αποτελεσμάτων.
2. **J2SE client.** Ένας J2SE client μπορεί να αποκτήσει πρόσβαση στο ίδιο το web service όπως ακριβώς και ένα J2EE component. Ο J2SE client, δεδομένου ότι βασίζεται στην πλούσια θεμελιώδη λειτουργικότητα της J2SE πλατφόρμας, έχει στη διάθεσή του μια διόλου ευκαταφρόνητη ευελιξία και υπολογιστική δύναμη, που του επιτρέπει να παρουσιάζει περισσότερο λεπτομερή τα αποτελέσματα που επιστρέφονται από το service.

3. **Non-Java client.** Ένας non-Java client μπορεί να αποκτήσει πρόσβαση στο ίδιο το web service με τη βοήθεια του SOAP μέσω HTTP. Ένα τέτοιο παράδειγμα είναι το σενάριο όπου μια εφαρμογή που έχει αναπτυχθεί με εργαλεία Microsoft έχει πρόσβαση στο BB web service.

Στην παρούσα εργασία, η πρόσβαση στο BB web service επιτυγχάνεται με τη χρήση ενός J2SE client.



Σχήμα 9: Εναλλακτικοί web service clients

7.4 Σχεδιαστικές θεωρήσεις για J2SE client

Σε αντίθεση με την J2EE πλατφόρμα, η J2SE δεν παρέχει το ίδιο εύρος και βάθος λειτουργικότητας. Οπότε οι αναλυτές/προγραμματιστές μιας non-J2EE πλατφόρμας όπως είναι η

J2SE είναι υπεύθυνοι για θέματα όπως η αναζήτηση στοιχείων για το service, η δημιουργία και η συντήρηση κλάσεων που σκοπό έχουν την πρόσβαση σε αυτό, κ.α.

Έτσι εντοπίζονται κάποια σημεία τα οποία χρήζουν ιδιαίτερης προσοχής σε σχέση με τον J2SE Building Block client. Τα σημεία αυτά είναι τα ακόλουθα:

- **Εφαρμογές μακράς διάρκειας (long-running applications).** Η χρήση J2SE αποδεικνύεται εξαιρετικά ευεργετική για την περίπτωση των web service clients οι οποίοι τρέχουν για εκτεταμένους χρονικούς περιόδους.
- **Πλούσιες γραφικές διεπαφές για τα σύνθετα δεδομένα (Rich graphical user interface (GUI) for complex data).** Οι J2SE clients μπορούν να παρέχουν στους χρήστες μια πλούσιο όψη των δεδομένων. Αυτό το πλούσιο interface είναι ικανό να επιτρέψει στον χρήστη να πλοηγηθεί και να τροποποιήσει (offline) ένα μεγάλο σύνολο δεδομένων που επιστρέφονται από το service.
- **Ασυνεχής πρόσβαση στο δίκτυο (intermittent network access).** Ένας J2SE client μπορεί να χρησιμοποιήσει ένα Web services δίχως να είναι απαραίτητο να έχει συνεχής πρόσβαση στο δίκτυο, απαλλάσσοντας έτσι το δίκτυο από κάποια επιβάρυνση. Ιδιαίτερη προσοχή απαιτείται για να διασφαλιστεί η συνοχή των δεδομένων μεταξύ του service και του client.
- **Σύνθετη υπολογισμοί (complex computations) στον client.** Οι J2SE clients είναι πολύ καλά ορισμένοι όσον αφορά την επεξεργασία σύνθετων μαθηματικών υπολογισμών, καθώς και λειτουργιών ενημέρωσης των δεδομένων, και υποβολής των αποτελεσμάτων αυτών στο service. Για παράδειγμα, τέτοιου είδους clients διαθέτουν καλύτερους πόρους για την επεξεργασία εικόνων κι έτσι μπορούν να αλαλάξουν τον server από περαιτέρω επιβάρυνση. Έτσι, οι J2SE clients μπορούν να παρέχουν ένα σαφώς καλύτερο περιβάλλον –σε σχέση με αυτό του service- για τις αλληλεπιδράσεις με τα σύνολα των δεδομένων.

7.5 Λοιπές θεωρήσεις

Υπάρχει μια σειρά από κάποιες επιπρόσθετες γενικές θεωρήσεις (consideration) όπου πρέπει να λαμβάνονται σοβαρά υπόψιν κατά την υλοποίηση των web service clients. Αυτές οι λοιπές σημαντικές θεωρήσεις θα αναλυθούν σε αυτή την ενότητα.

7.5.1 Διαχείριση Διαλογικής Κατάστασης – Managing Conversational State

Ένας γενικός κανόνας είναι ότι οι web services clients πρέπει να αντιμετωπίζουν τα web services σαν να μην έχουν ένα σταθερό σημείο πρόσβασης (stateless) και να θεωρούν ότι το service endpoint δεν συγκρατεί καμιά πληροφορία για τις προηγούμενες αλληλεπιδράσεις του client. Όσων αφορά την παρούσα εργασία, ωστόσο, είναι απαραίτητο μερικές φορές να πραγματοποιεί ο client μια αλληλουχία από κλήσεις προς το service προκειμένου να φέρει σε πέρας τη δοθείσα εργασία.

Ένας προτεινόμενος τρόπος για την διαχείριση αυτής της κατάστασης μεταξύ των κλήσεων, είναι να πραγματοποιηθεί αυτή η εργασία εξ' ολοκλήρου στον client. Εξάλλου υπάρχουν τρόπο για τον client να διευθετήσει ο ίδιος πολλαπλές κλήσεις προς το service και να διαχειριστεί με τέτοιο τρόπο την κατάστασή του ώστε να βρίσκεται σε συγχρονισμό και με την κατάσταση του server.

Μια δεύτερη προσέγγιση είναι να αναλάβει το ίδιο το web service τη διαχείριση της διαλογικής κατάστασης (conversational state) με τον client.

Στην παρούσα εργασία το BB web service είναι αυτό που αναλαμβάνει την διαχείριση των πολλαπλών κλήσεων με τη βοήθεια μιας σχεσιακής βάσης δεδομένων.

7.5.2 Συγχρονισμός από κοινού καταστάσεων μεταξύ clients – Synchronizing Shared State among Clients

Ο τελικός χρήστης του BB Web services είναι δυνατό να «φορτώσει» πάντα από ένα instance του BB client. Αυτό μπορεί να πραγματοποιηθεί είτε τρέχοντας παραπάνω από ένα instances του BB client στον ίδιο υπολογιστή είτε τρέχοντας ένα ή παραπάνω instances σε παραπάνω από ένα υπολογιστές.

Δεδομένου ότι το BB Web service αποθηκεύει τα δεδομένα του service σε μια βάση δεδομένων (RDBMS), η φάση ανάγνωσης δεδομένων (read-only state) δεν απαιτεί κανενός είδους συγχρονισμού μεταξύ των διαφορετικών clients, αφού οι clients σε αυτή τη φάση πρόκειται να κάνουν μόνο ανάκτηση δεδομένων. Προκειμένου λοιπόν να αποφευχθούν προβλήματα ασυνεπειών,

απαραίτητη είναι η υιοθέτηση τεχνικών όπως ο έλεγχος time stamp ιδιαίτερα σε περιπτώσεις όπου ο client δύναται να επιχειρήσει ανανέωση των δεδομένων (π.χ. υποβολή ενός νέου job).

Σε κάθε περίπτωση πάντως ο σχεδιασμός και η υλοποίηση του BB εξασφαλίζει ότι δεν πρόκειται να υπάρξουν προβλήματα ασυνεπειών στα δεδομένα, σε περιπτώσεις όπου ο ίδιος χρήστης τρέχει παράλληλα πολλά instances του BB client.

7.5.3 Συναλλαγές Μακράς Διάρκειας – Long Running Transactions

Ο client του Building Block κανονικά πρόκειται να εκτελεί εργασίες μακράς διάρκειας απομακρυσμένα στην server πλευρά τους web service. Ο βασικός λόγος είναι το γεγονός ότι απαιτούμενοι υπολογισμοί ενδέχεται να είναι εξαιρετικά αιτητικοί σε υπολογιστική ισχύ άρα και σε απαιτούμενο χρόνο για την διεκπεραίωση τους. Έτσι, δεν αποκλείεται η επεξεργασία των δεδομένων από το τ-Argus να διαρκεί από μερικά δευτερόλεπτα ως και μερικά λεπτά. Αυτό λοιπόν σημαίνει ότι τα Building Blocks πρέπει να είναι σε θέση να διαχειρίζονται συναλλαγές μακράς διάρκειας.

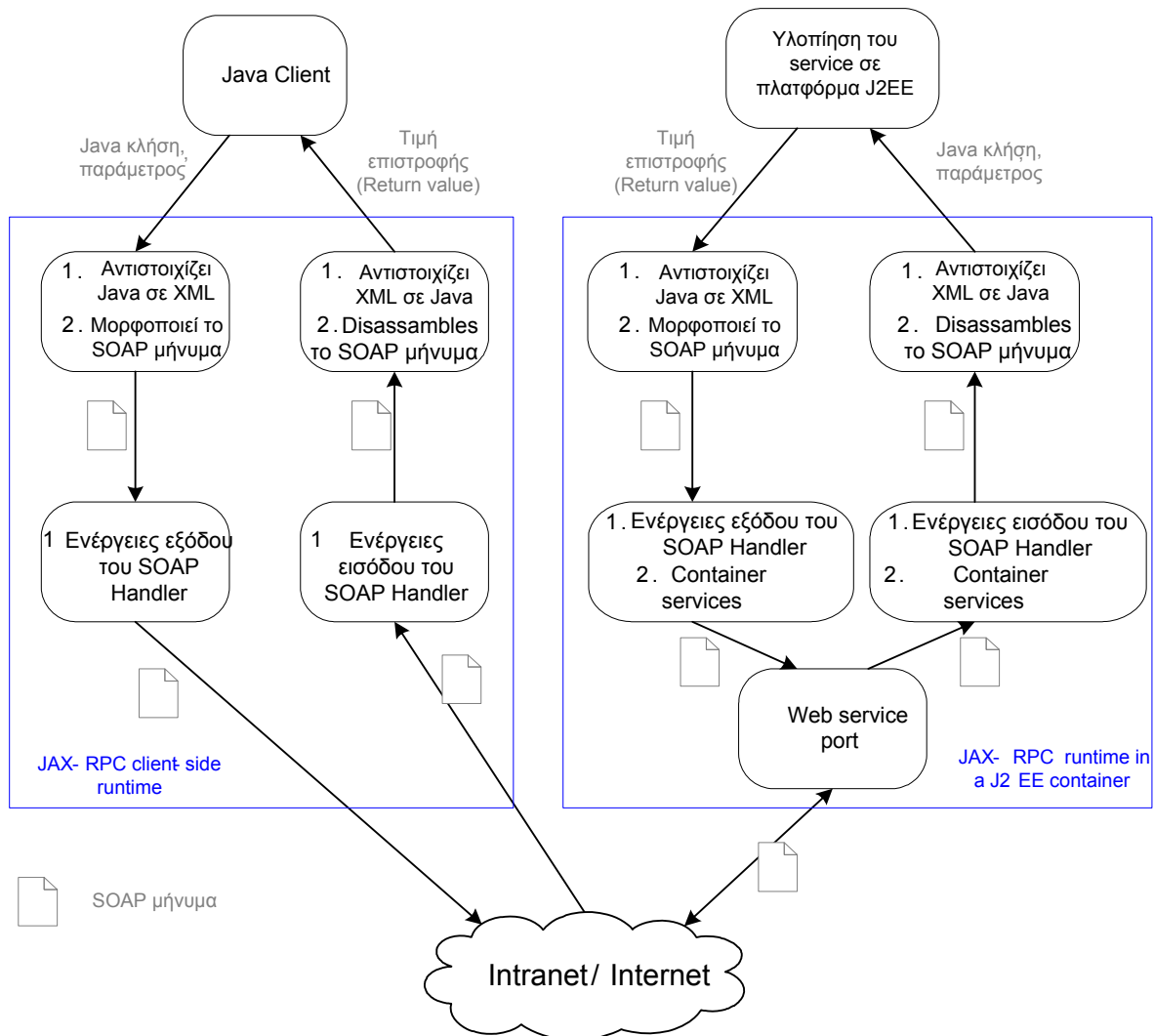
7.6 Ροή (Flow) κλήσης Web Service

Στο σενάριο του BB web service, ένας BB client στέλνει ένα αίτημα (request) στο BB Web service, κάτι αντίστοιχο σαν την ερώτηση «ποια είναι η κατάσταση των εργασιών που έχουν υποβληθεί (submitted jobs);», ενώ το service από την πλευρά του, αφού επεξεργαστεί το αίτημα, στέλνει μια απάντηση στο BB client που καλύπτει το αίτημα που προηγουμένως του είχε αποσταλεί. Εφόσον τόσο ο BB client όσο και το BB service έχουν υλοποιηθεί σε περιβάλλον Java, έτσι ο client πραγματοποιεί μια κλήση στο service με μια απλή επίκληση (invocation) της αντίστοιχης Java μεθόδου, έχοντας βέβαια θέσει και περάσει τις απαραίτητες παραμέτρους, στο τέλος αυτού του διαλόγου ο client λαμβάνει το αποτέλεσμα της επίκλησης της μεθόδου.

Προκειμένου να γίνει αντιληπτό το περιβάλλον στο οποίο δρα το BB Service, παρουσιάζεται στο Σχήμα 10 ένα τυπικό διάγραμμα ροής κλήσης ενός web service, υλοποιημένο σε περιβάλλον Java. Στο Σχήμα 10 φαίνεται πως ακριβώς ο BB Java client επικοινωνεί με το Java web service πάνω στην J2EE 1.4 πλατφόρμα. Πρέπει να επισημανθεί πως το Σχήμα 10 δύναται να τροποποιηθεί όταν ένας non-Java client αλληλεπιδρά με το Java Web service. Σ'αυτή την περίπτωση, η δεξιά πλευρά του σχήματος, η οποία αναπαριστά τις ενέργειες web service, παραμένει ακριβώς η ίδια,

ενώ αλλάζει η αριστερή πλευρά που παρουσιάζει τις ενέργειες της client πλατφόρμας. Στην περίπτωση που ο Java client κάνει μια κλήση σε web service το οποίο είναι σε μια non-Java πλατφόρμα, τότε η δεξιά πλευρά του σχήματος αλλάζει ώστε να αναπαραστήσει την web service πλατφόρμα και όσον αφορά το αριστερό μέρος, το οποίο αναπαριστά τις ενέργειες του client, τότε αυτό παραμένει ακριβώς όπως φαίνεται στο εν λόγω σχήμα.

Αφού ο client γνωρίζει πως να αποκτήσει πρόσβαση στο service, έτσι δημιουργεί ένα αίτημα προς το service κάνοντας μια απλή κλήση στην αντίστοιχη Java μέθοδο, η οποία με τη σειρά της περνά τις απαραίτητες παραμέτρους στην πλευρά του client-side JAX-RPC runtime. Με την κλήση της μεθόδου, ο client ουσιαστικά καλεί μια συγκεκριμένη λειτουργία (operation) του service. Το δε JAX-RPC runtime αντιστοιχίζει όλους τους Java types σε standard XML types και εν συνεχεία μορφοποιεί ένα SOAP μήνυμα το οποίο συμπυκνώνει την ίδια την κλήσης της μεθόδου και των απαραίτητων παραμέτρων. Το runtime τότε περνά το SOAP μήνυμα μέσω των SOAP handlers, εάν αυτοί υπάρχουν, και κατόπιν στο port του service στην πλευρά του server.



Σχήμα 10: Τυπική ροή κλήσης (flow) Web Service

Πριν να στείλει το port το εν λόγω αίτημα προς το endpoint του web service, πρέπει να βεβαιώσει ότι ο J2EE container έχει κάνει όλους του απαραίτητους ελέγχους (ασφαλείας) στο SOAP αίτημα που εστάλη. Κατόπιν ο J2EE container εξάγει την απαραίτητη πληροφορία που πηγάζει από την κλήση του client και εκτελεί οποιαδήποτε αντιστοίχιση XML-to-Java αντικειμένων κρίνεται απαραίτητη. Παρόμοια βήματα ακολουθούνται και την περίπτωση που το service στέλνει πίσω στον client την απάντηση στο αίτημα που δέχτηκε.

Φυσικά, όλες οι λεπτομέρειες που λαμβάνουν χώρα στη διαδικασία κλήσης της αντίστοιχης μεθόδου καθώς και στην αναμενόμενη απάντηση (response), πραγματοποιούνται με τη φροντίδα

της J2EE πλατφόρμας στην πλευρά του server side (στην παρούσα εργασία πρόκειται για τον WebLogic application server) καθώς επίσης με την φροντίδα του Java run-time στην πλευρά του client. Η ίδια η J2EE πλατφόρμα προσυλλάσσει τον σχεδιαστή/προγραμματιστή από τις διάφορες λεπτομέρειες. Αντιθέτως, ο σχεδιαστής/προγραμματιστής έχει να αντιμετωπίσει στην πλειονότητα τους μόνο τυπικά ζητήματα που σχετίζονται με τον προγραμματισμό με τη χρήση της ίδιας της Java, όπως η κλήση Java μεθόδων, τύπων, κ.ά.

7.7 Έκδοση (Publication) του client Building Block Web Service Interface

Η έκδοση του client BB Web service interface έχει να κάνει με το πως οι clients ανακαλύπτουν και χρησιμοποιούν το εκάστοτε service, με άλλα λόγια το πώς το service θα γίνει διαθέσιμο στους clients. Τα δύο άκρα του φάσματος του service το κάνουν ευρέως γνωστό (public) και το εγγράφουν σε μια public registry ή ακόμη περιορίζουν τη διαθεσιμότητα αυτού μόνο σε clients που είναι ρητά πιστοποιημένοι γι'αυτό. Ένα τυπικό παράδειγμα μιας τέτοιας προσέγγισης είναι το γνωστό Google SOAP Search API (<http://www.google.com/apis/>) μέσω του οποίου επιτρέπεται η επίκληση (invocation) των λειτουργιών αναζήτησης του Google κάνοντας χρήση του πρωτοκόλλου SOAP και του WSDL.

Στα πλαίσια της εν λόγω εργασίας το BB Web service δεν πρόκειται να είναι ευρέως διαθέσιμο, αντιθέτως ένα configuration file περιγράφει τα χαρακτηριστικά το service με τη βοήθεια του οποίου είναι εφικτή η επίκληση του από εξουσιοδοτημένους clients.

7.8 Επίπεδο αλληλεπίδρασης (interaction) Building Block service

Το επίπεδο αλληλεπίδρασης που μπορεί να διακρίνει κανείς σε ένα web service έχει ορισμένες σημαντικές αρμοδιότητες, η πλέον σημαντική είναι ο σχεδιασμός του interface του service που παρουσιάζεται στον client. Εφόσον ο clients αποκτά πρόσβαση στο service μέσω αυτού, το interface είναι το σημείο αφετηρία της αλληλεπίδρασης μεταξύ client και service. Επίσης, το επίπεδο αλληλεπίδρασης διαχειρίζεται και άλλες αρμοδιότητες, όπως αυτή της λήψης των αιτημάτων του client, μεταβίβαση των αιτημάτων στην κατάλληλη επιχειρησιακή λογική (business logic) καθώς και η δημιουργία και αποστολή απαντήσεων (responses).

Δύο είναι οι προσεγγίσεις για την ανάπτυξη του εκάστοτε service interface, οι ακόλουθοι:

1. **Java σε WSDL:** Ο προγραμματιστής ξεκινά με ένα σύνολο από Java interfaces για το Web service και με βάση αυτά αυτός δημιουργεί το αντίστοιχη WSDL περιγραφή του service, η οποία με τη σειρά της είναι διαθέσιμη σε τρίτους για να την χρησιμοποιήσουν.
2. **WSDL σε Java:** Ο προγραμματιστής ξεκινά από ένα WSDL έγγραφο περιγράφοντας τις λεπτομέρειες του Web service interface και έπειτα χρησιμοποιεί αυτή την πληροφορία για να χτίσει τα αντίστοιχα Java interfaces.

Ο δεύτερος τρόπος όπου ξεκινώντας από τα Java interfaces δημιουργείται τελικά το WSDL έγγραφο είναι ο πλέον εύκολος τρόπος μεταξύ των δύο. Με αυτή την προσέγγιση ο προγραμματιστής δεν χρειάζεται να γνωρίζει καμιά λεπτομέρεια σχετικά με το WSDL επειδή χρησιμοποιεί αυτόματα εργαλεία για να δημιουργήσει την WSDL περιγραφή.

Η προσέγγιση WSDL σε Java διευκολύνει τη διατήρηση ενός σταθερού web services interface, αλλά δεν είναι καλός ορισμένο όσων αφορά την εξελικτική προσέγγιση ενός web service.

Εφόσον λοιπόν το BB Web service πρέπει να αναπτυχθεί κατά τέτοιο τρόπο ώστε να υποστηρίζεται η εξελικτική του πορεία και οι ενδεχόμενες αλλαγές, το σχήμα Java σε WSDL είναι αυτό που θα υιοθετήθηκε στην παρούσα εργασία. Το αντίστοιχο εργαλείο που χρησιμοποιείται για την δημιουργία του WSDL από τις Java classes, είναι το WebLogic JWSC.

7.9 Interface Endpoint

Στην J2EE πλατφόρμα υπάρχουν οι ακόλουθες δύο επιλογές για την υλοποίηση του web service interface:

1. **JAX-RPC service endpoint**, επίσης αναφέρεται και σαν Web tier endpoint
2. **EJB service endpoint**, επίσης αναφέρεται και σαν EJB tier endpoint

Η επιλογή του τύπου του endpoint εξαρτάται με τη φύση του web service. Όταν ολόκληρη η επιχειρησιακή λογική υλοποιείται στο web επίπεδο, τότε είναι λογικό ότι και η υλοποίηση του endpoint θα γίνει σαν JAX-RPC service endpoint. Από την άλλη πλευρά, όταν επιχειρησιακή λογική είναι υλοποιημένη σε EJBs, τότε είναι λογικό να υλοποιηθεί ομοίως ένα EJB service endpoint.

Δεδομένου ότι το BB web service είναι κάτι εξ' ολοκλήρου καινούριο, στο οποίο δεν έχει εφαρμοστεί κανενός είδους λογική και δεν έχει υλοποιηθεί κώδικας γι' αυτό, η επιλογή του endpoint πρέπει να βασιστεί στις συγκεκριμένες απαιτήσεις των Building Blocks.

Συγκεκριμένα υπάρχουν πολλά οφέλη από την υλοποίηση EJB service endpoint τα σημαντικότερα από αυτά είναι τα ακόλουθα:

- Πρόσβαση (access) multi-threaded.
- Παράλληλη (concurrent) πρόσβαση του client.
- Διαχείριση συναλλαγών (transaction handling).
- Άδειες πρόσβασης με βάση του επίπεδο της μεθόδου (method-level access permissions).

Είναι ξεκάθαρο από τις παραπάνω θεωρήσεις ότι η επιλογή του EJB service endpoint παρέχει σημαντικά πλεονεκτήματα. Γι' αυτό τον λόγο και επιλέχτηκε για την υλοποίηση του BB Web service endpoint.

Κεφάλαιο 8^ο

Οδηγός Χρήσης Εφαρμογής (Application's User Guide)

8.1 Εισαγωγή

Σκοπός του BB τ-Argus interface είναι να παρέχει στον χρήστη την δυνατότητα μετά από κατάλληλα βήματα να καλεί το τ-Argus service το οποίο βρίσκεται σε έναν απομακρυσμένο application server. Σύμφωνα με τις επιλογές του χρήστη, αυτόματα δημιουργούνται τα κατάλληλα αρχεία εισόδου, ενώ μετά την επεξεργασία τους και την αποστολή τους στην τ-Argus εφαρμογή ο χρήστης ενημερώνεται για τα αποτελέσματα της επεξεργασίας των δεδομένων που προηγουμένως εισήγαγε.

8.2 Building Block (ΔΤ) τ-Argus Service

Το Building Block τ- Argus service έχει σαν αντικειμενικό σκοπό να παρέχει τη δυνατότητα τόσο σε χρήστες όσο και εφαρμογές συνδεδεμένες στο διαδίκτυο να καλούν και να εκτελούν επεξεργασία των δεδομένων τους μέσω της εφαρμογής τ-Argus. Αυτή η υπηρεσία (service) καθιστά το τ-Argus λογισμικό διαθέσιμο ακόμα και σε απομακρυσμένες περιοχές απλά χρησιμοποιώντας τα web services standards.

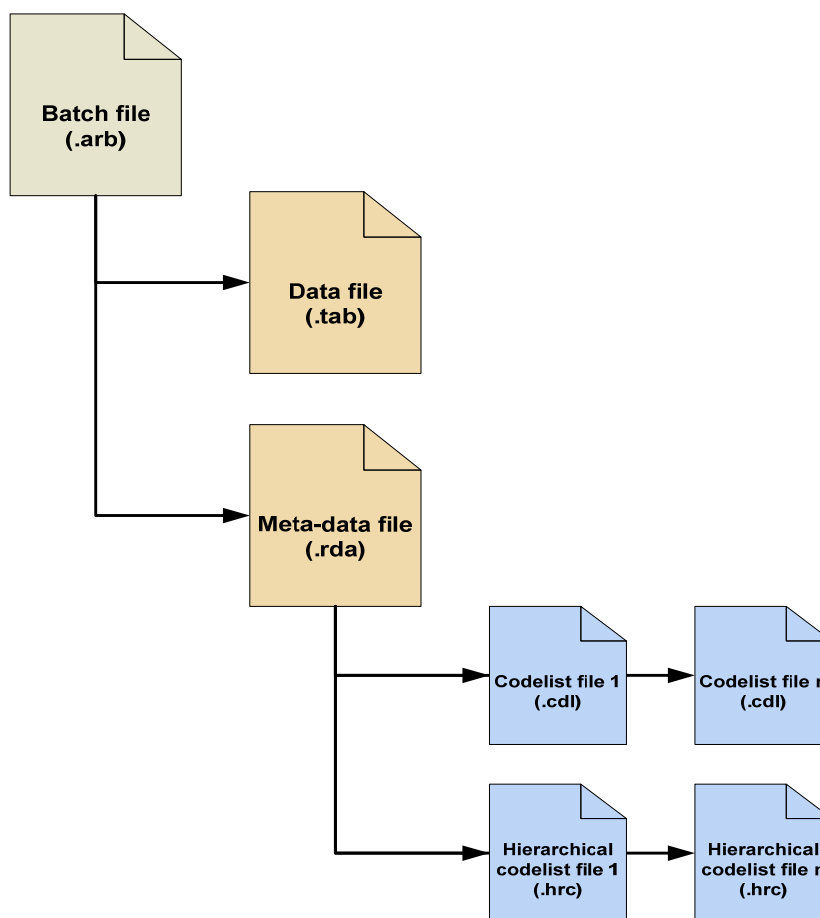
Όπως έχει ήδη αναφερθεί σε προηγούμενα κεφάλαια το ΔΤ που υλοποιείται σ'αυτή την εργασία δεν είναι τίποτε άλλο, παρά ένας wrapper γύρω από την εφαρμογή τ-Argus, κάνοντας μάλιστα χρήση της batch mode εκδοχή της και όχι την αλληλεπιδραστική της (interactive).

Στην παρούσα φάση της υλοποίησης του λογισμικού εργαλείου της τ-Argus Δομικού Τμήματος απαραίτητη είναι η «χειροκίνητη» (manually) εισαγωγή των αρχείων εισόδου και η κλήση για επεξεργασία τους προκειμένου να ληφθούν τα αρχεία εξόδου.

Σε μελλοντικές εναλλακτικές της παρούσας έκδοσης του λογισμικού είναι δυνατόν να αυτοματοποιηθεί η διαδικασία δημιουργίας των επί μέρους αρχείων εισόδου.

8.3 τArgus Αρχεία Εισόδου (input files)

Το πρώτο βήμα για το χρήστη της εφαρμογής είναι να προετοιμάσει τα αρχεία εισόδου για να χρησιμοποιηθούν από το τ-Argus service.



Σχήμα 11: Αρχεία εισόδου του τ-Argus service

Το κυρίως αρχείο εισόδου του τ-Argus είναι το **batch file**. Το batch file έχει συνήθως κατάληξη “.arb”.

Ένα batch file είναι ουσιαστικά ένα αρχείο κειμένου (text file) που περιέχει όλες εκείνες τις εντολές που πρέπει να εκτελεστούν έτσι ώστε από το τ-Argus. Το αρχείο αυτό μπορεί να δημιουργηθεί είτε χρησιμοποιώντας το τ-Argus user interface είτε «χειροκίνητα» από τον ίδιο τον χρήστη. Η σύνταξη του αρχείου αυτού περιγράφεται εκτενώς στο εγχειρίδιο της εφαρμογής τ-Argus [20]. Ένα παράδειγμα batch file είναι το ακόλουθο.

```

<OPENTABLEDATA> "C:\WorkDir\eurodata.tab"
<OPENMETADATA> "C:\WorkDir\eurodata.rda"
<SPECIFYTABLE> "NACE" | "CostResp" | "Weight" | "CostResp"
<SAFETYRULE>FREQ(2,25) |
<READTABLE>
<SUPPRESS> GH(1,100)
<WRITETABLE>
(1,1,1,"C:/TArgusOut/175642110518886400/batch_output.csv")

```

Σχήμα 12: Batch file αρχείο εισόδου

Εν συνεχεία, είναι το αρχείο δεδομένων (data file) το οποίο μπορεί να έχει μορφή πινακοειδής - tabular data (.tab)- είτε να είναι σε μορφή μικροδεδομένων - micro-data file (.asc). Το αρχείο αυτό περιέχει τα δεδομένα που πρέπει να προστατευτούν από το τ-Argus. Η σύνταξη του αρχείου αυτού περιγράφεται εκτενώς στο εγχειρίδιο της εφαρμογής τ-Argus [20]. Ένα παράδειγμα data file είναι το ακόλουθο.

```

"C";44.00;1;276.00;S
"CA";4.00;1;55.00;S
"CA10";0.00;1;10.00;Q
"CA11";4.00;1;45.00;Q
"CA12";0.00;1;0.00;S
"CB";40.00;1;221.00;S
"CB13";2.00;1;11.00;S
"CB14";38.00;1;210.00;S

```

Σχήμα 13: Πινακοειδής αρχείο εισόδου

Το δε αρχείο μεταδεδομένων -meta-data file (.rda)- περιέχει την περιγραφή του περιεχομένου του αρχείου δεδομένων. Και αυτό το αρχείο έχει τη μορφή κειμένου και περιέχει περιγραφές μεταβλητών και εντολές. Η σύνταξη του αρχείου αυτού περιγράφεται εκτενώς στο εγχειρίδιο της εφαρμογής τ-Argus [20]. Ένα παράδειγμα metadata file είναι το ακόλουθο.

```

<SEPARATOR> ";"
<SAFE> "S"
<UNSAFE> "Q"
<PROTECT> "P"
NACE
  <RECODEABLE>
  <TOTCODE> C
  <HIERARCHICAL>
  <HIERCODELIST> "hierarchy.hrc"

```

```

<HIERLEADSTRING> @
CostResp
    <NUMERIC>
Weight
    <NUMERIC> <SHADOW>
FreqVar
    <NUMERIC>
    <FREQUENCY>
StatusVar
    <STATUS>

```

Σχήμα 14: Αρχείο (εισόδου) μεταδεδομένων

Το αρχείο μεταδεδομένων μπορεί επίσης να περιέχει αναφορές και παραπομπές σε αρχεία λίστας κωδικών -code list files (.cdl)- και αρχεία ιεραρχικής λίστας κωδικών -hierarchical code list files (.hrc). Η σύνταξη των αρχείου αυτών περιγράφεται εκτενώς στο εγχειρίδιο της εφαρμογής τ-Argus [20]. Ένα παράδειγμα hierarchical file είναι το ακόλουθο.

```

CA
@CA10
@CA11
@CA12
CB
@CB13
@CB14

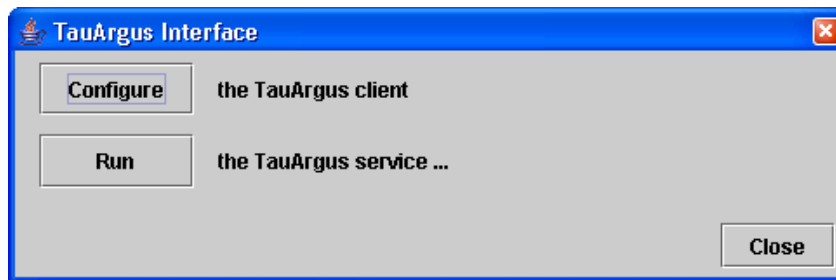
```

Σχήμα 15: Αρχείο (εισόδου) ιεραρχίας

Είναι στην ευθύνη του χρήστη να δημιουργήσει τα απαιτούμενα αρχεία εισόδου (batch, data, meta-data, code lists και hierarchical code lists).

8.4 Κλήση (invocation) τArgus Service

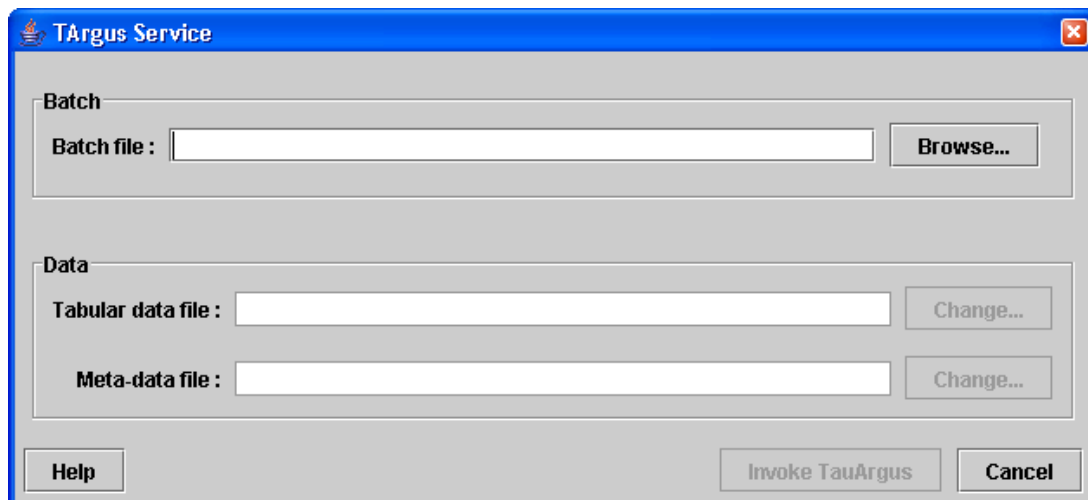
Εφόσον τα αρχεία εισόδου είναι διαθέσιμα, ο χρήστης μπορεί να ανοίξει το κυρίως παράθυρο διαλόγου της εφαρμογής (→ menu Program → τ-Argus).



Σχήμα 16: Κυρίως παράθυρο διαλόγου της εφαρμογής

Το κυρίως παράθυρο διαλόγου έχει δύο κουμπιά που αντιστοιχούν σε δύο επιλογές, τα εξής :

- **Configure:** αυτό το κουμπί ανοίγει ένα νέο παράθυρο στον client της εφαρμογής, στο οποίο ορίζονται οι εκάστοτε ρυθμίσεις για την ομαλή λειτουργία της εφαρμογής.
- **Run:** το κουμπί αυτό ανοίγει το κυρίως παράθυρο διαλόγου της κλήσης του τ-Argus service (main τ-Argus Service invocation dialog), μέσω αυτού του παραθύρου ο χρήστης μπορεί να καλέσει το τ-Argus service.



Σχήμα 17: κυρίως παράθυρο διαλόγου της κλήσης του τ-Argus service (main τ-Argus Service invocation dialog)

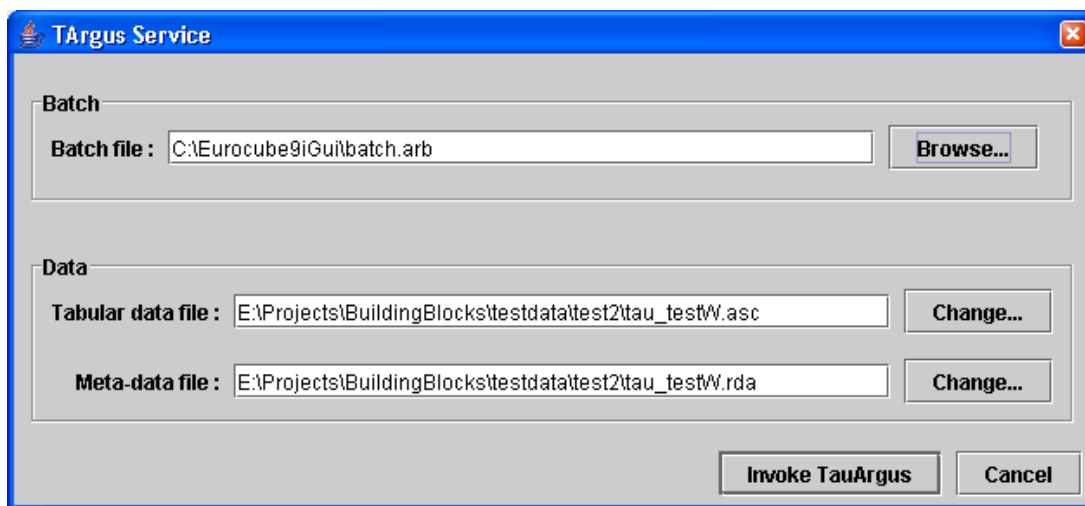
Κατόπιν, το πρώτο βήμα είναι ο χρήστης να επιλέξει ένα υπάρχων αρχείο εντολών (batch file), αυτή η επιλογή διευκολύνεται με τη χρήση του κουμπιού *Browse* που βρίσκεται στο batch group του παραθυρικού διαλόγου.

Όταν επιλεχτεί το batch file (.arb), τότε το σύστημα αναλύει (parse) αυτόματα το επιλεγμένο αρχείο εντολών και πραγματοποιεί ελέγχους που σχετίζονται με αν και κατά πόσο τα δεδομένα του αρχείου είναι ορθά, όπως επίσης και οι αναφορές στα αντίστοιχα αρχεία δεδομένων και μεταδεδομένων.

Επιπλέον, το σύστημα αυτόματα αναλύει (parse) το αρχείο μεταδεδομένων, κάνοντας ελέγχους ορθότητας όσων αφορά τη σύνταξη αλλά και την ενδεχόμενη ύπαρξη αναφοράς (reference) σε αρχεία ιεραρχιών (code list / hierarchical code list files).

Στο σημείο αυτό πρέπει να σημειωθεί ότι μόνο βασικοί έλεγχοι πραγματοποιούνται αυτόματα από το σύστημα, κυρίως όσων αφορά τη συντακτική ορθότητα των περιεχομένων των αρχείων, την προσβασιμότητα τους και την ύπαρξη εκείνων των αρχείων που περιγράφονται ως παραπομπές.

Αφού ολοκληρωθούν οι έλεγχοι ύπαρξης και προσβασιμότητας των επί μέρους αρχείων και εφόσον είναι απαραίτητο εμφανίζεται ένας παραθυρικός διάλογος που υποδεικνύει και το αντίστοιχο σφάλμα. Εάν όμως όλα τα αρχεία εισόδου είναι ορθά δομημένα και είναι διαθέσιμα, τότε το κουμπί *Invoke τ-Argus* μετατρέπεται αυτομάτως σε ενεργό:



Σχήμα 18: Σχήμα 19: κυρίως παράθυρο διαλόγου της κλήσης του τ-Argus service με δεδομένα εισόδου

Από αυτό το σημείο πλέον, ο χρήστης μπορεί:

- Να διαλέξει κάποιο άλλο batch file οπότε και να ξεκινήσει εκ νέου η διαδικασία ελέγχου ορθότητας των δεδομένων εισόδου,
- να αλλάξει το αρχείο δεδομένων ή και μεταδεδομένων και να διαλέξει κάποιο(α) διαφορετικά από αυτά που αναφέρονται στο batch file,
- να καλέσει το τ-Argus service με δεδομένα τα αρχεία εισόδου.

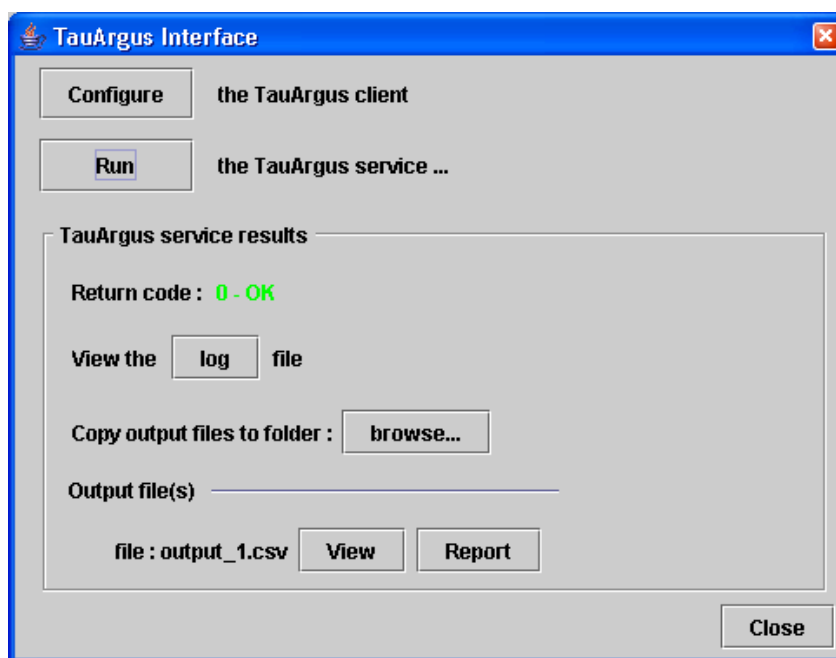
Όταν ο χρήστης εκκινεί τη διαδικασία κλήσης (invocation) του service με το πάτημα του αντίστοιχου κουμπιού, τότε το σύστημα:

- Συνδέεται με το service και λαμβάνει την απαραίτητη πληροφορία διαμόρφωσης (configuration information) του service,
- Λαμβάνει ένα session ID και ακολούθως δημιουργεί ένα session folder σε ένα shared folder, έτσι ώστε να επιτρέπεται η πρόσβαση τόσο στους πελάτες (clients) όσο και στον ίδιο το server. Η παραγωγή του session ID γίνεται με τέτοιο τρόπο ώστε να εξασφαλίζεται η τυχαιότητα και η μοναδικότητα αυτού. Πρόκειται άλλωστε για έναν αριθμό 20 ψηφίων.
- Αντιγράφει τα προκαθορισμένα αρχεία εισόδου (batch, data, meta-data, code lists) στο session folder (κατόπιν το τα αρχεία batch και meta-data τροποποιούνται ώστε να περιέχουν αναφορές στα σωστά αρχεία στην περίπτωση της εκτέλεσης της απομακρυσμένης διεργασία τ-Argus- remote τ-Argus process).
- Εκτελείτε το τ-Argus service δοθέντος του αντίστοιχου session ID.

Όσο διαρκεί αυτή η επεξεργασία, ο χρήστης βλέπει έναν δείκτη αναμονής που του δείχνει ότι το σύστημα είναι απασχολημένο με την εκτέλεση των εργασιών.

Οποιοδήποτε στιγμή, και εφόσον υπάρξει κάποιο σφάλμα, εμφανίζεται ένα pop-up παράθυρο με σύντομη περιγραφή του σφάλματος και αντιστοίχως σταματά η διαδικασία επεξεργασίας δεδομένων.

Στην περίπτωση που η κλήση (invocation) του service επιτυγχάνεται επιτυχώς, τότε το αντίστοιχο παράθυρο της κλείνει ενώ στο κυρίως παράθυρο της εφαρμογής παρουσιάζονται τα αποτελέσματα της επεξεργασίας τους από το τ-Argus:



Σχήμα 20: Παράθυρο αποτελεσμάτων

Ο κώδικας επιστροφής (return code) που φαίνεται στο παραπάνω screenshot είναι ένα νούμερο που αναπαριστά των κωδικό που επιστρέφεται από την εφαρμογή τ-Argus.

Η μηδενική τιμή (0) σημαίνει ότι το αρχείο εντολών (batch file) έχει επεξεργαστεί επιτυχώς. Οποιοδήποτε άλλη τιμή σημαίνει ότι το τ-Argus δεν μπόρεσε να φέρει σε πέρας όλη τη διαδικασία επεξεργασίας που περιγράφεται στο αρχείο εντολών.

Με το πάτημα του κουμπιού *log*, εμφανίζεται το log αρχείο που δημιουργήθηκε από το τ-Argus, το οποίο είναι εξαιρετικά χρήσιμο σε περίπτωση σφάλματος.

Το κουμπί *browse* επιτρέπει στον χρήστη να αντιγράψει όλα τα τ-Argus αρχεία εξόδου από εκεί που αρχικά έχουν τοποθετηθεί από την εφαρμογή σε έναν επιλεγμένο φάκελο στο τοπικό σύστημα αρχείων του χρήστη (local folder on the user file system).

Ο χρήστης μπορεί να δει το CSV αρχείο εξόδου με την επιλογή του κουμπιού View, όπως επίσης μπορεί να δει το HTML report με την περιγραφή των χαρακτηριστικών της τ-Argus επεξεργασίας πατώντας το αντίστοιχο κουμπί *Report*.

Τέλος είναι στην ευθύνη του χρήστη να ερμηνεύσει τα αποτελέσματα της επεξεργασίας από το τ-Argus.

8.5 τArgus αρχεία εξόδου (output files)

Κάθε κλήση του τ-Argus service έχει ως αποτέλεσμα τη δημιουργία τουλάχιστον ενός αρχείου μηνυμάτων ενημέρωσης (log file) και ενός ή περισσότερων αρχείων εξόδου, πλήθος το οποίο μπορεί να ποικίλει σε σχέση με τις εντολές που δίνονται στο αρχείο εντολών (batch file). Αυτά ακριβώς τα αρχεία αποτελούν ουσιαστικά και τα αρχεία εξόδου του Δομικού Τμήματος, του τ-Argus web service.

Πιο συγκεκριμένα, τα αρχεία εξόδου που προκύπτουν κατά την επεξεργασία των δεδομένων από την εφαρμογή τ-Argus manual, είναι τα ακόλουθα:

- ένα αρχείο σε μορφή CSV (.csv), με τα αποτελέσματα της επεξεργασίας από το τ-Argus,
- ένα αρχείο αναφοράς σε μορφή (html), το οποίο περιγράφει όλα τα απαραίτητα δεδομένα για την επεξεργασία των δεδομένων από το τ-Argus (αρχεία εισόδου/εξόδου, κανόνες ασφάλειας, αλγόριθμος που επιλεχθεί, κατάσταση κελιών ασφαλή/μη ασφαλή/προστατευμένα, απαιτούμενος χρόνος για προστασία του εκάστοτε πίνακα)
- ένα log αρχείο, το οποίο περιέχει τα μηνύματα που εξάγονται από την τ-Argus διαδικασία.

, "Total", "CA", "CA10", "CA11", "CA12", "CB", "CB13", "CB14"
 , 44, x, -, x, -, x, x, 38,

Σχήμα 21: Αρχείο εξόδου της μορφής csv

t-ARGUS Report

Table created date: 23-Feb-2007, time 12:30:39

Original file:	C:\WorkDir\neurodata.tab
Meta file:	C:\WorkDir\neurodata.rda
Table file:	C:/TArgusOut/175642110518886400/batch_output.csv

Table read as table

Table structure

Var	Function	# codes
Response var:	CostResp	
Explanatory var 1:	NACE	8
Shadow variable :	Weight	
Cost variable :	CostResp	

Safety Rule:

Minimum individual cell frequency: 2; safety margin: 25 %
 Manual safety margin: 20%
 Missing codes have been considered unsafe

GHMITER solution

GHMITER range ratio used: 0.500

Σχήμα 22: Αρχείο (εξόδου) - report file- σε μορφή html

```
04-Dec-2006 16:44:30 : Start Batch run:
C:\TArgusIn\1866433862758891520\eurodata.arb

04-Dec-2006 16:44:30 : Command: <OPENTABLEDATA> "C:\WorkDir\eurodata.tab"

04-Dec-2006 16:44:30 : Command: <OPENMETADATA> "C:\WorkDir\eurodata.rda"

04-Dec-2006 16:44:30 : Command: <SPECIFYTABLE>
"NACE" | "CostResp" | "CostVar" | "CostResp"

04-Dec-2006 16:44:30 : Command: <SAFETYRULE>FREQ(2,25) |

04-Dec-2006 16:44:30 : Command: <READTABLE>
```

Σχήμα 23: Log αρχείο εξόδου

Κεφάλαιο 9^ο

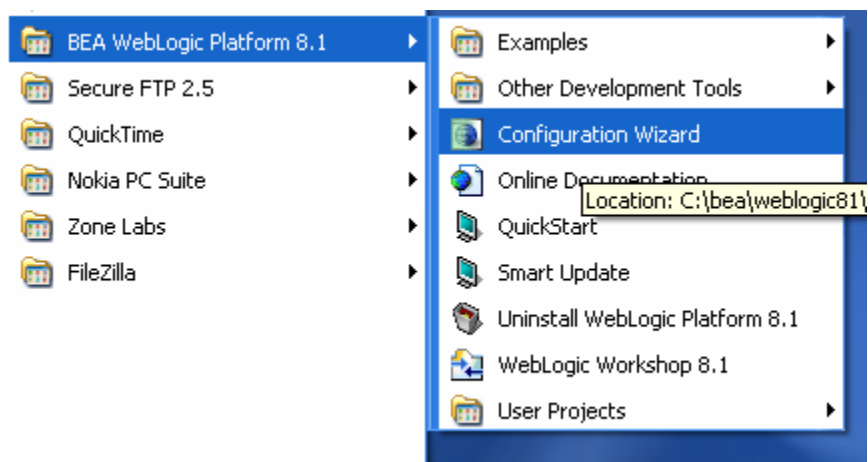
Οδηγός Εγκατάστασης και Διαμόρφωσης του Application Server

9.1 Εισαγωγή

Στην ενότητα αυτή περιγράφεται βήμα προς βήμα η διαδικασία εγκατάστασης και διαμόρφωσης του τArgus Web Service με τη βοήθεια του BEA Weblogic Server. Η διαδικασία αυτή απαιτεί την προεγκατάσταση του BEA Weblogic Server στο μηχάνημα (server) που θα χρησιμοποιηθεί για τη φιλοξενία του εν λόγω web service.

9.2 Ορισμός πεδίου ορισμού (domain) τArgus Building Block

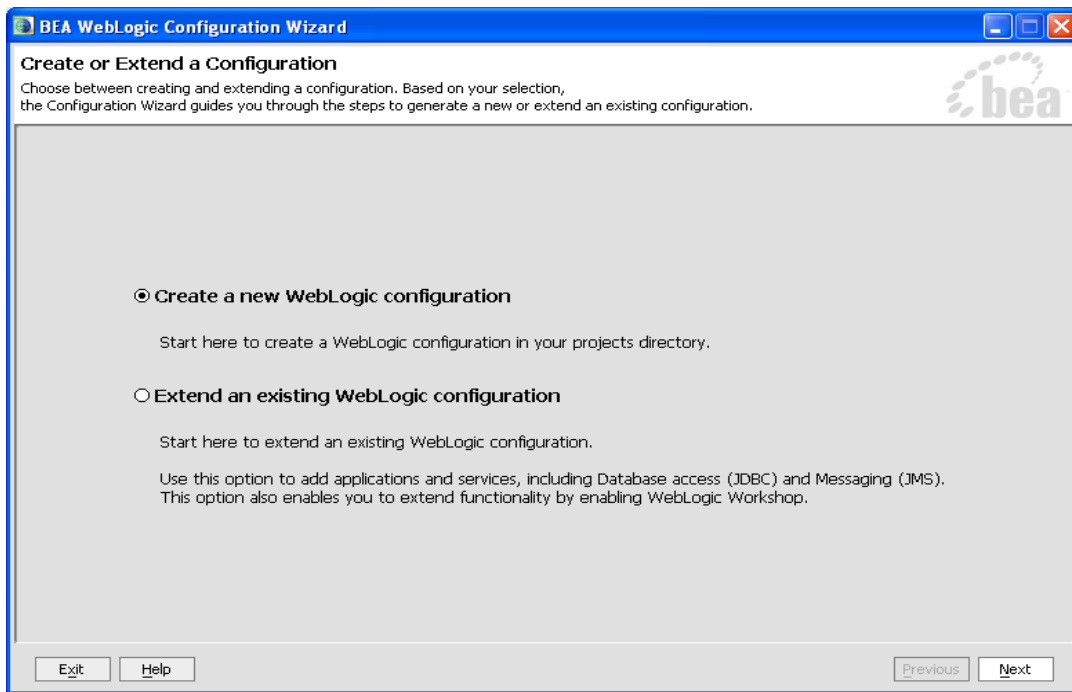
Αρχικά, επιλέξτε από το μενού εκκίνησης BEA Weblogic 8.1 platform → Configuration Wizard.



Σχήμα 24: Μενού BEA Weblogic Server

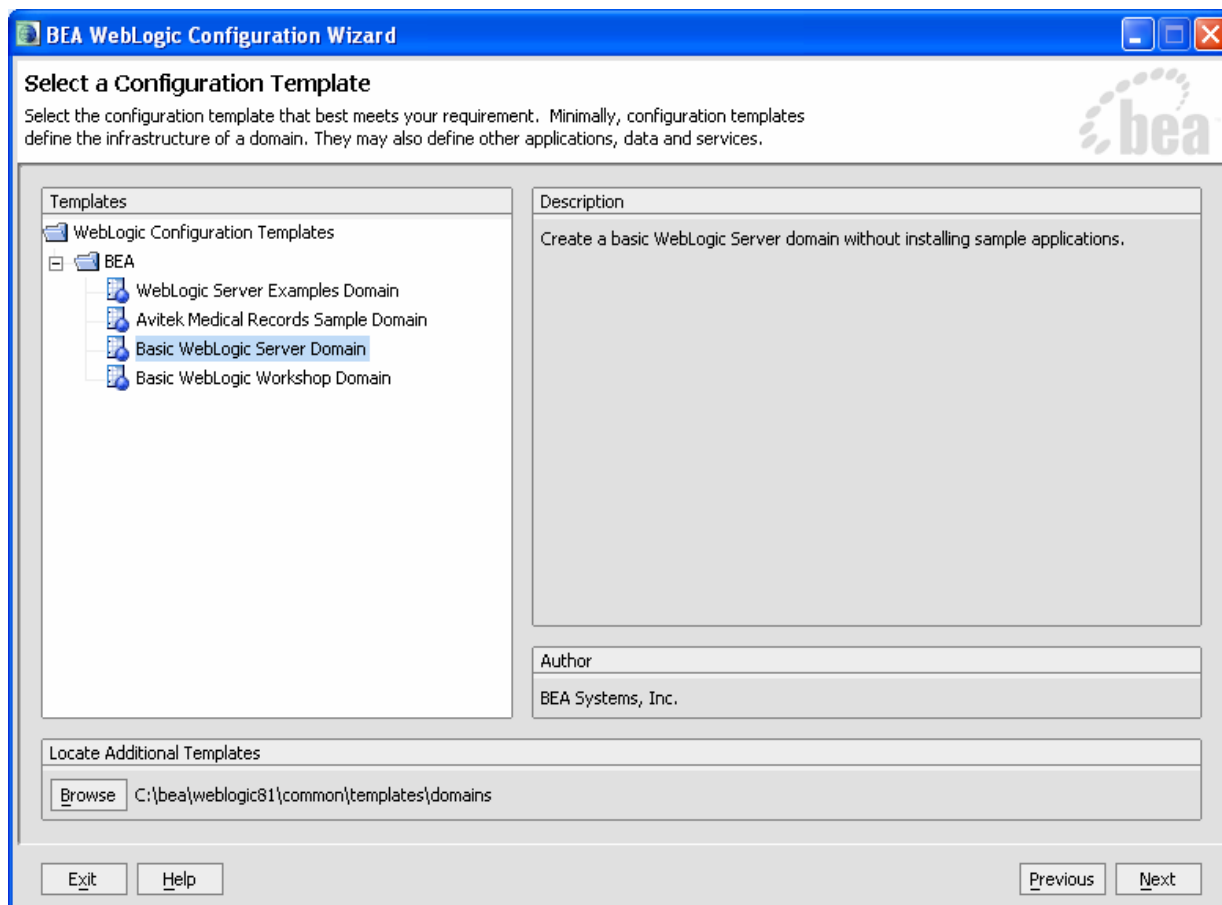
Ένα παράθυρο με τον BEA Weblogic Configuration Wizard ανοίγει. Ο wizard αυτός έχει τα ακόλουθα βήματα:

Create or Extend a Configuration: Επιλέξτε Create a new Weblogic configuration και πατήστε το κουμπί Next.



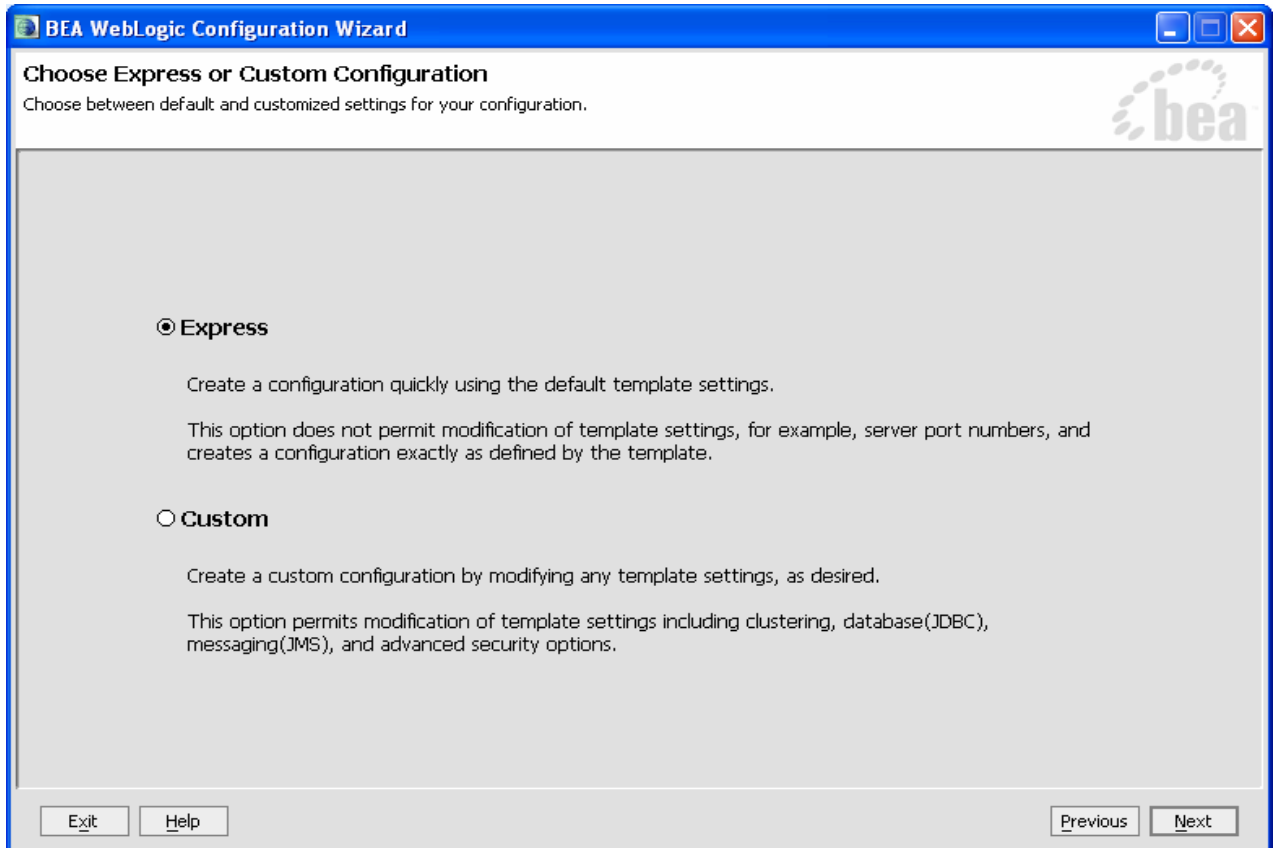
Σχήμα 25:BEA Weblogic Configuration Wizard – Δημιουργία ή επέκταση Διαμόρφωσης

Select a Configuration Template: Επιλέξτε BEA → Basic Weblogic Server Domain από τα Templates. Θέστε ως domain location στο Locate Additional Templates, για παράδειγμα: C:\bea\weblogic81\common\templates\domains).



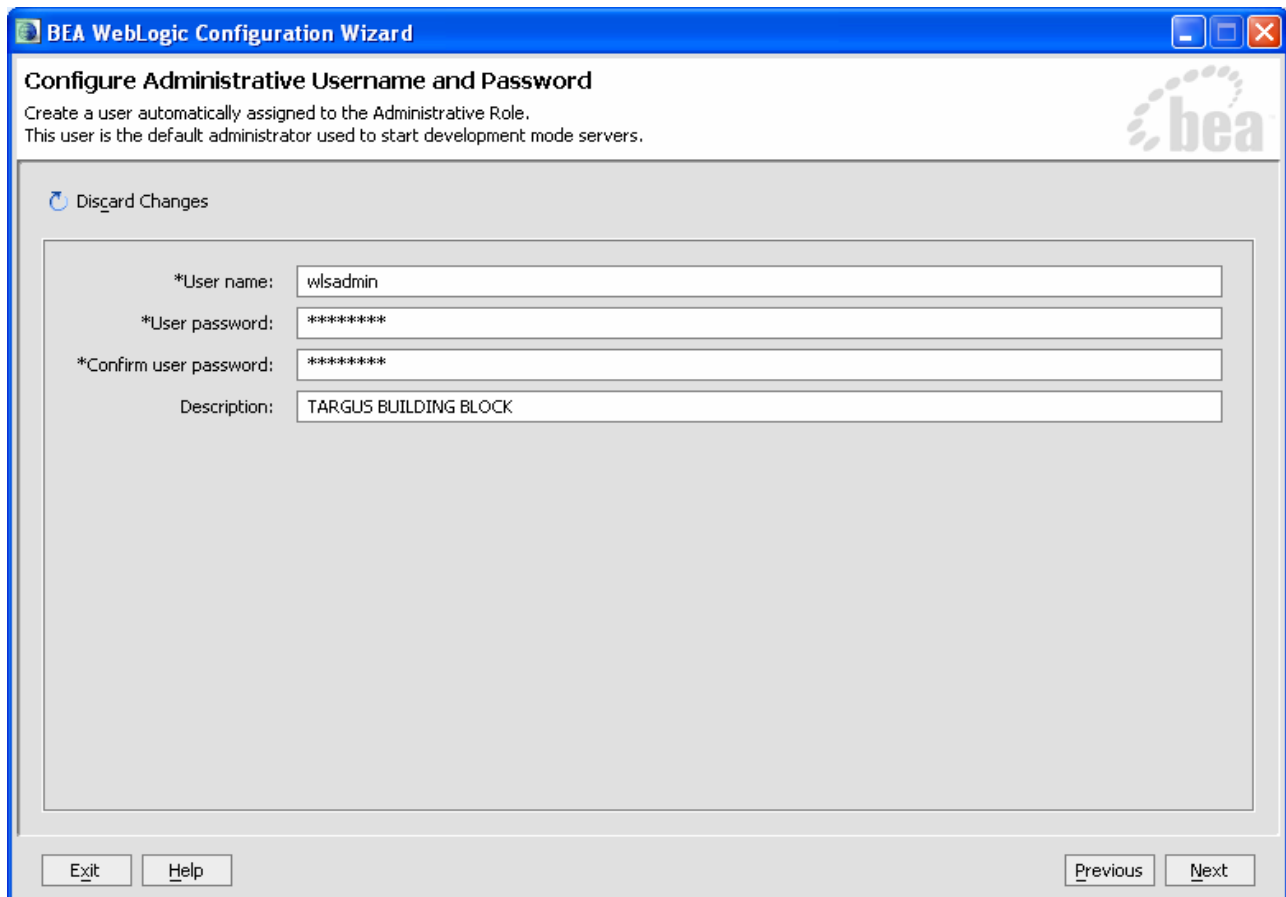
Σχήμα 26: BEA Weblogic Configuration Wizard – Επιλογή περιγράμματος διαμόρφωσης

Choose Express or Custom Configuration: Επιλέξτε Express.



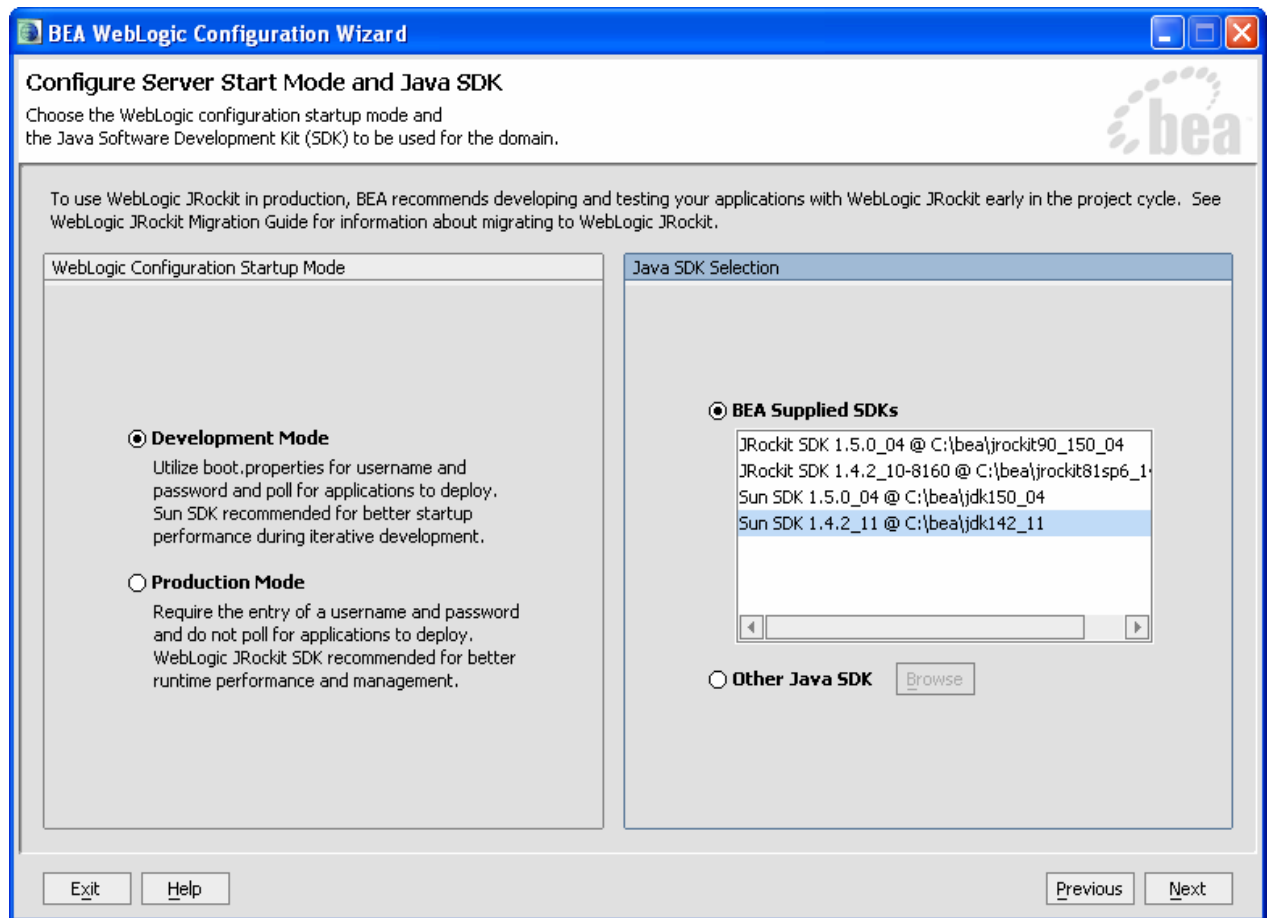
Σχήμα 27: BEA Weblogic Configuration Wizard – Επιλογή ταχείας ή κατ'απαίτηση διαμόρφωσης

Configure Administrative Username and Password: Προσδιορίστε το username και το password που πρόκειται να χρησιμοποιηθούν για την εκκίνηση του Administration Server.



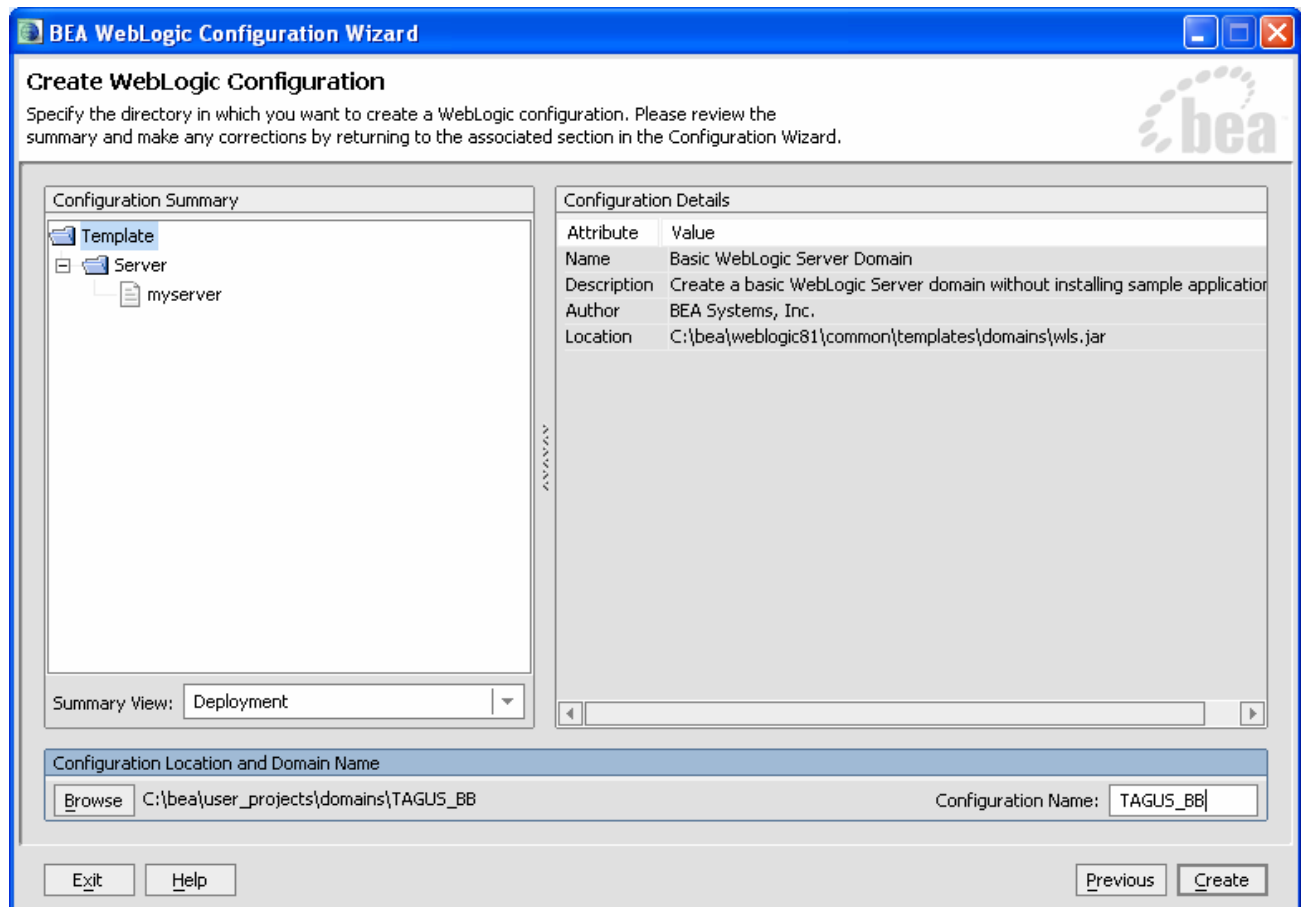
Σχήμα 28: BEA Weblogic Configuration Wizard – Διαμόρφωση του διαχειριστικού username και password

Configure Server Start Mode and Java SDK: Επιλέξτε *Development Mode* από το Weblogic Configuration Startup Mode. Εν συνεχεία, επιλέξτε Java SDK 1.4 στο πεδίο BEA Supplied SDKs.

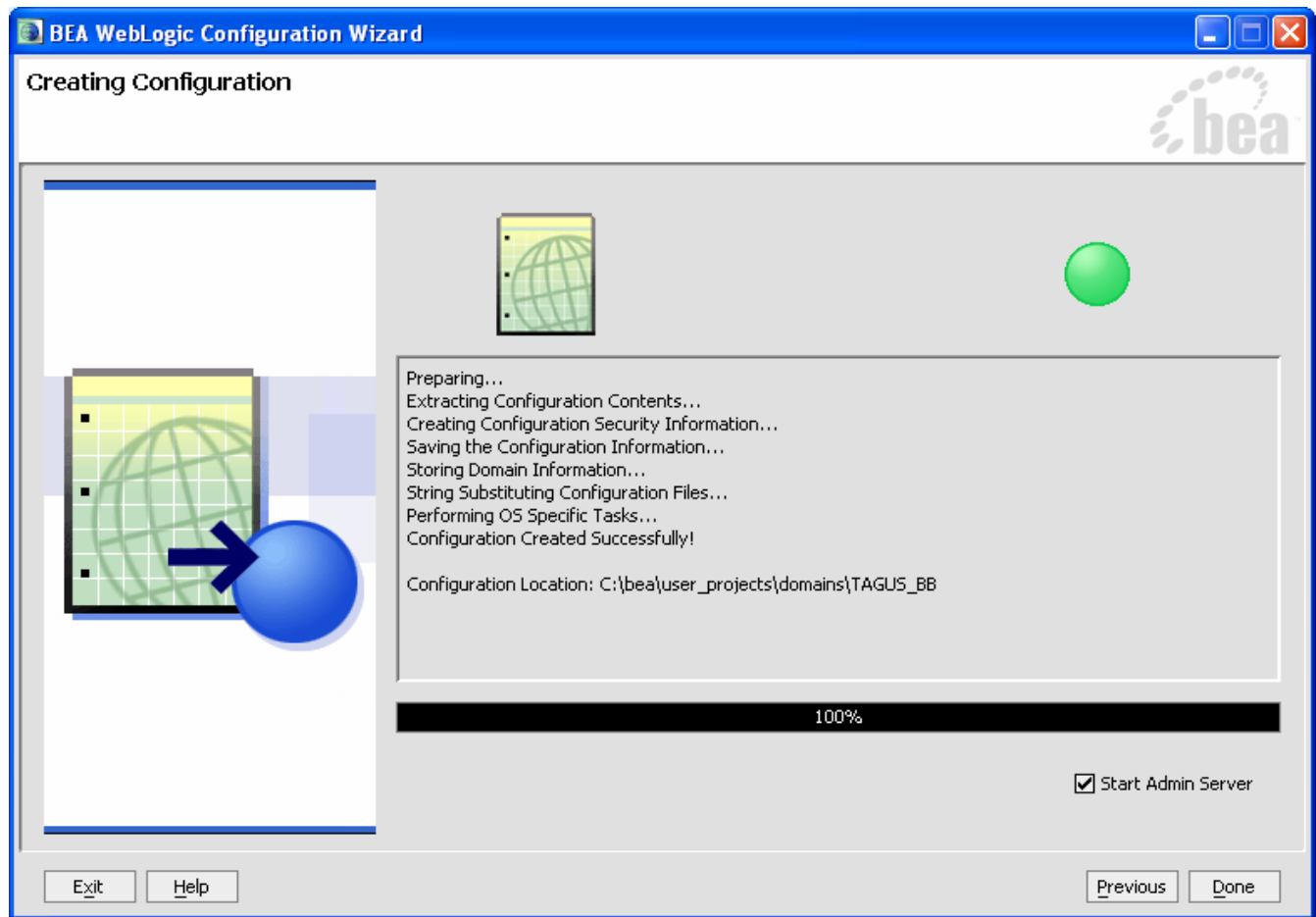


Σχήμα 29: BEA Weblogic Configuration Wizard – Διαμόρφωση του server εκκίνησης και του Java SDK

Create Weblogic Configuration: θέστε το Summary View στο *Deployment*. Επιλέξτε ως Configuration location για παράδειγμα C:\bea\user_projects\domains\ και ως Configuration Name «TARGUS_BB» και πατήστε OK.



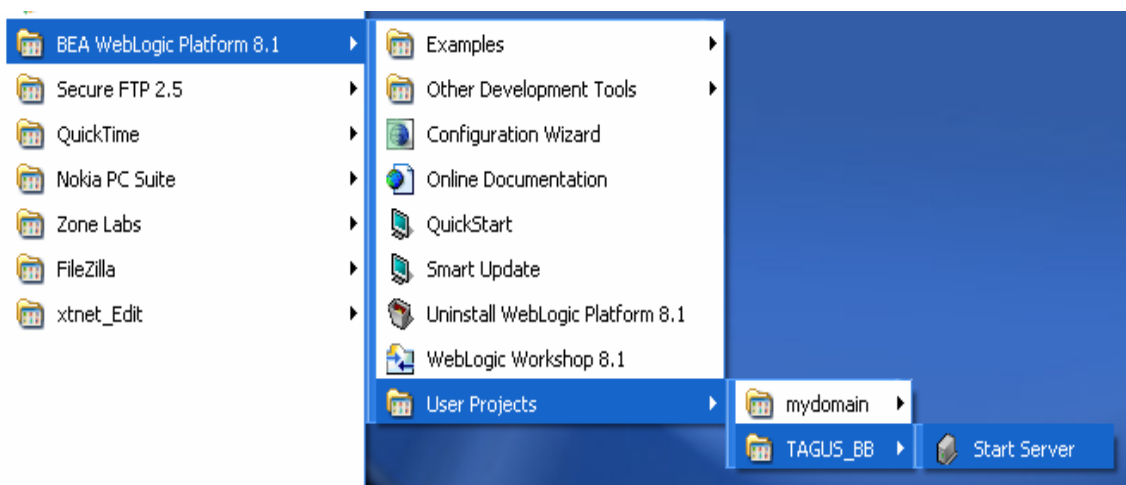
Creating Configuration: Επιλέξτε Start Admin Server προκειμένου να ξεκινήσει application server και τέλος πιέστε το *Done* έτσι ώστε να κλείσει ο wizard.



Σχήμα 30: BEA Weblogic Configuration Wizard – Δημιουργία Διαμόρφωσης

9.3 Εκκίνηση Application Server

Εναλλακτικά, μπορείτε να ξεκινήσετε τον the application server από το μενού εκκίνησης: *BEA Weblogic Platform 8.1* → *User Projects* → *TARGUS_BB* → *Start Server*



Σχήμα 31: Μενού BEA Weblogic Platform – Εκκίνηση Application Server

9.4 Deploy τArgus Building Block

Αρχικά ξεκινήστε τον application server (δείτε την ενότητα: 9.3 Εκκίνηση Application Server).

Η κονσόλα του application server μπορεί να ξεκινήσει επιλέγοντας το URL:

server_name:7001/console/ σε έναν web browser.

Ας υποθέσουμε ότι ο application server είναι φιλοξενείται (hosted) στον **server_name** windows server.



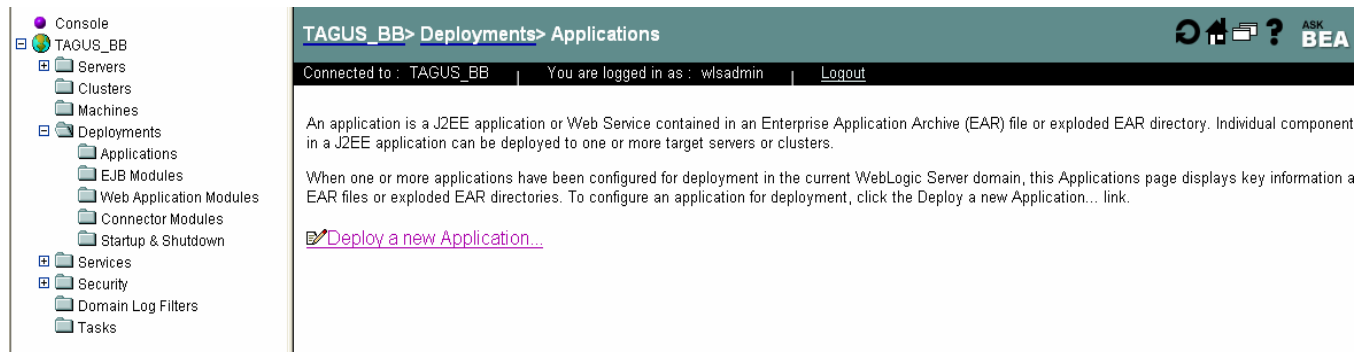
Σχήμα 32: Εισαγωγή στην Διαχειριστική Κονσόλα του Weblogic Application Server

Εισάγεται το username και το password, που προηγουμένως είχατε επιλέξει, προκειμένου να κάνετε log in.

9.5 Διαδικασία πρώτου deployment

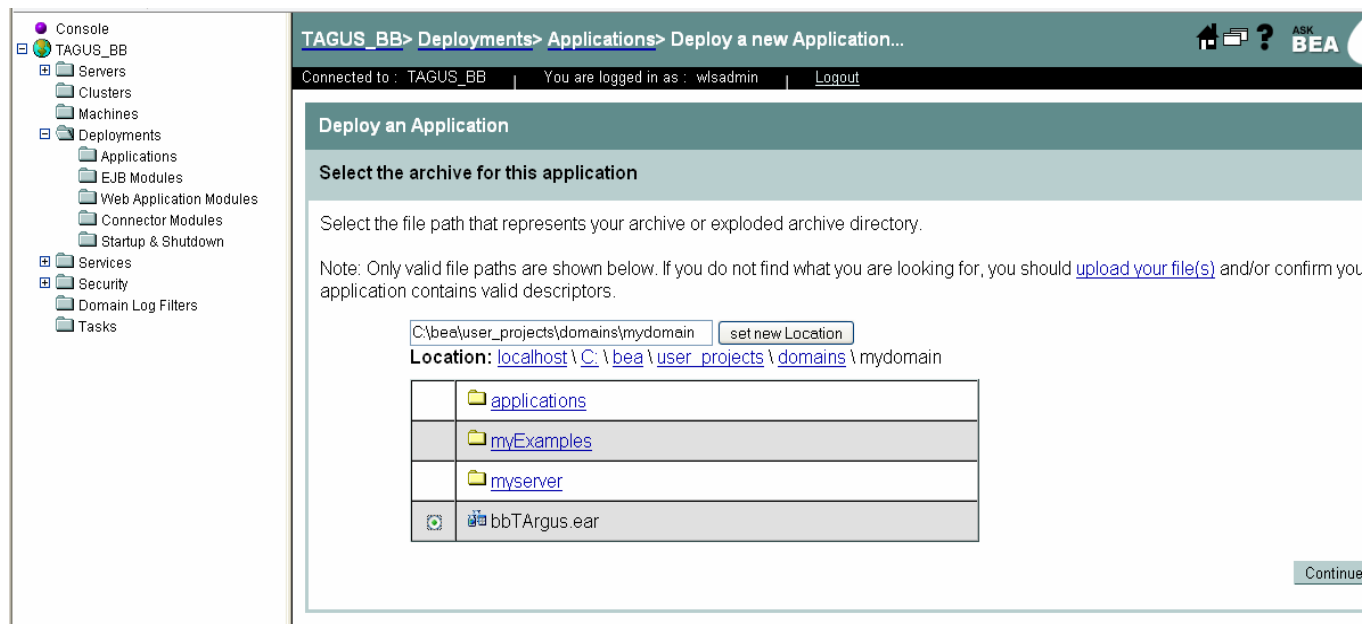
Επιλέξτε *TARGUS_BB* → *Deployments* → *Applications* από το αριστερό μέρος του παραθύρου.

Κατόπιν κάντε Click πάνω στο *Deploy a new Application* link.



Σχήμα 33: Deployment του web service (1)

Επιλέξτε το file path στο οποίο βρίσκεται το αρχείο **bbTArgus.ear** και εν συνεχεία πιάστε το κουμπι *Continue*.



Σχήμα 34: Deployment του web service (2)

Εισάγεται το όνομα που πρόκειται να χρησιμοποιηθεί για την εφαρμογή (π.χ. BBTArgus) και πιέστε το *Deploy* προκειμένου να γίνει deploy η εφαρμογή.

TAGUS_BB> Deployments> Applications> Deploy a new Application...

Connected to : TAGUS_BB | You are logged in as : wlsadmin | [Logout](#)

Deploy an Application

Review your choices and deploy

Deployment Targets

Your application will be deployed to the following locations:

bbTArgus will be deployed to
Servers - myserver

Source Accessibility

Since this is a single server environment, no further stage configuration is required. The server will access this application's files from the location specified.

Identity

Enter a name to be used to identify this application.

Name:

The name of this application deployment.

Deploy

Σχήμα 35: Deployment του web service (3)

Το deployment ολοκληρώνεται, όταν το **Success** status εμφανίζεται τόσο για το *Status of Last Action* of EJB Modules Deployment όσο και για το Web Modules Deployment.

The screenshot shows the JBoss Administration Console interface. The breadcrumb navigation is TAGUS_BB > Deployments > Applications > bbTargus. The user is logged in as wlsadmin. The 'Deploy' tab is selected. The page contains two tables showing deployment status for EJB and Web Application Modules.

Deployment status for EJB Modules

Module	Target	Target Type	Deployment Status	Status of Last Action
bbtargus.jar	myserver	Server	Available	Success

Deployment Status for Web Application Modules

Module	Target	Target Type	Deployment Status	Status of Last Action
webservice	myserver	Server	Available	Success

Buttons at the bottom: Stop Application, Redeploy Application.

Σχήμα 36: Deployment του web service (4)

9.5.1 Ορισμός startup class

Επιλέξτε: *TARGUS_BB* → *Deployments* → *Startup & Shutdown* από την αριστερή πλευρά του παραθύρου και έπειτα κάνουμε click στο *Configure a new Startup Class* link.

The screenshot shows the JBoss Administration Console interface for the 'Startup & Shutdown' configuration. The breadcrumb navigation is TAGUS_BB > Deployments > Startup & Shutdown. The user is logged in as wlsadmin. The left sidebar shows the navigation tree with 'Startup & Shutdown' selected. The main content area contains instructions and links to configure startup and shutdown classes.

Startup & Shutdown

A startup class is a Java program that is automatically loaded and executed when a server is started or restarted. By default, startup classes are loaded and executed after all other server subsystems have initialized and after the server deploys modules. A shutdown class is a Java program that is automatically loaded and executed when you gracefully shut down a server.

To configure a startup class, click the [Configure a new Startup Class...](#) link. To configure a shutdown class, click the [Configure a new Shutdown Class...](#) link.

Σχήμα 37: Δημιουργία startup κλάσης (1)

Θέστε όνομα: **BBThread** και ClassName: **BBTargus.BBThread** και κατόπιν πιέστε το κουμπί *Create*.

TAGUS_BB > Deployments > Startup Classes > Create a new StartupClass...

Connected to : TAGUS_BB | You are logged in as : wlsadmin | [Logout](#)

Configuration | Target and Deploy | Notes

This page allows you to define the general configuration of this startup class.

Name:
 A name to represent this startup class in the Administration Console.

ClassName:
 The fully qualified name of this startup class.

Deployment Order:
 A numerical value that indicates when WebLogic Server executes this startup class, relative to other startup classes. Classes with lower Deployment Order values are executed before those with higher values.

Arguments:

The arguments that the targeted server(s) use to initialize this startup class.

☐ **Failure is Fatal**
 Specifies whether a failure in this startup class prevents the targeted server(s) from starting.

☐ **Run Before Application Deployments**
 Causes the targeted server(s) to load and run this startup class before activating JMS and JDBC services and before starting deployment for applications and EJBs.

☐ **Run Before Application Activations**
 Causes the targeted server(s) to load and run this startup class after activating JMS and JDBC services and before activating applications and EJBs.

[Create](#)

Σχήμα 38: Δημιουργία startup κλάσης (2)

Εν συνεχεία, προσθέστε το **BBTArgus.jar** στο **startWebLogic.cmd** classpath:

- Πλοηγηθείτε στην τοποθεσία όπου βρίσκεται το **startWebLogic.cmd** (π.χ.. C:\bea\user_projects\domains\TAGUS_BB\)
- Κάντε edit to **startWebLogic.cmd**
- Αντικαταστήστε τη γραμμή: *set CLASSPATH=%WEBLOGIC_CLASSPATH%;%POINTBASE_CLASSPATH%;%JAVA_HOME%\jre\lib\rt.jar;%WL_HOME%\server\lib\webservices.jar;%CLASSPATH%* with *CLASSPATH=%WEBLOGIC_CLASSPATH%;%POINTBASE_CLASSPATH*

```
%,%JAVA_HOME%\jre\lib\rt.jar;%WL_HOME%\server\lib\webservices.jar;%BBT  
ArgusJAR_LOCATION%;%CLASSPATH%
```

Π.χ.:

```
set
```

```
CLASSPATH=%WEBLOGIC_CLASSPATH%;%POINTBASE_CLASSPATH%;  
%JAVA_HOME%\jre\lib\rt.jar;%WL_HOME%\server\lib\webservices.jar;C:\be  
a\user_projects\domains\TAGUS_BB\bbtargus.jar;%CLASSPATH%
```

9.5.2 Αρχείο **BBTArgus.properties**

Αντιγράψτε το αρχείο **bbTArgus.properties** στον κατάλογο **C:\bea\user_projects\domains**.

Αλλάξτε τις παραμέτρους.

Π.χ.:

```
tauArgusOutFiles_location = "C:\TArgusOut"  
tauArgusInFiles_location = "C:\TArgusIn"  
mytauArgusOutFiles_location = "C:\MyTArgus"  
thread_sleep_time = "10000"  
oraconn = "jdbc:oracle:thin:@hostname:port:SID"
```

Κλείστε το Start Server window και επανακινήσετε τον application server.

Τα αναμενόμενα log messages είναι τα ακόλουθα:

```

C:\WINDOWS\system32\cmd.exe
CLASSPATH=C:\bea\jdk142-1\lib\tools.jar;C:\bea\WEBLOG-3\server\lib\weblogic_sp.jar;C:\bea\WEBLOG-3\server\lib\weblogic.jar;C:\bea\WEBLOG-3\common\eval\pointbase\lib\pbserver44
PATH=C:\bea\WEBLOG-3\server\bin;C:\bea\jdk142-1\jre\bin;C:\bea\jdk142-1\bin;D:\oracle\product\10.2.0\db_1\bin;C:\j2sdk1.4.2\bin;C:\oracle\product\10.2.0\db_1\bin;C:\WINDOWS\sys
*****
* To start WebLogic Server, use a username and *
* password assigned to an admin-level user. For *
* server administration, use the WebLogic Server *
* console at http://[hostname]:[port]/console *
*****
<Oct 31, 2006 1:09:43 PM EET> <Info> <WebLogicServer> <BEA-000377> <Starting WebLogic Server with Java HotSpot(TM) Client VM Version 1.4.2_11-b06 from Sun Microsystems Inc.>
<Oct 31, 2006 1:09:43 PM EET> <Info> <Configuration Management> <BEA-150016> <This server is being started as the administration server.>
<Oct 31, 2006 1:09:43 PM EET> <Info> <Management> <BEA-141107> <Version: WebLogic Server 8.1 SP6 Wed Jun 21 08:18:55 PDT 2006 781680>
WebLogic XML Module 8.1 SP6 Wed Jun 21 08:18:55 PDT 2006 781680
<Oct 31, 2006 1:09:43 PM EET> <Notice> <Management> <BEA-140005> <Loading domain configuration from configuration repository at C:\bea\user_projects\domains\TAGUS_BB\config.>
<Oct 31, 2006 1:09:44 PM EET> <Notice> <Log Management> <BEA-170019> <The server log file C:\bea\user_projects\domains\TAGUS_BB\myserver\myserver.log is opened. All server side
<Oct 31, 2006 1:09:45 PM EET> <Notice> <Security> <BEA-090082> <Security initializing using security realm myrealm.>
<Oct 31, 2006 1:09:45 PM EET> <Notice> <WebLogicServer> <BEA-000327> <Starting WebLogic Admin Server "myserver" for domain "TAGUS_BB">
<Oct 31, 2006 1:09:50 PM EET> <Warning> <EJB> <BEA-010001> <While deploying EJB 'Targus', class BBTargus.TargusBean was loaded from the system classpath. As a result, this clas
<Oct 31, 2006 1:09:50 PM EET> <Warning> <EJB> <BEA-010001> <While deploying EJB 'Targus', class BBTargus.TargusHome was loaded from the system classpath. As a result, this clas
<Oct 31, 2006 1:09:50 PM EET> <Warning> <EJB> <BEA-010001> <While deploying EJB 'Targus', class BBTargus.Targus was loaded from the system classpath. As a result, this class ca
Stratup class starting...
>>Into getConfiguration...
database_location = "file:c:\databases\BBlocksDB071307"
tauArgusOutFiles_location = "C:\TargusOut"
tauArgusInFiles_location = "C:\TargusIn"
mytauArgusOutFiles_location = "C:\MyTargus"
thread_sleep_time = "10000"
oraconn = "jdbc:oracle:thin:@192.168.2.6:1521:BBlockDB"
>>Into getDatabaseLocation, content:database_location = "file:c:\databases\BBlocksDB071307"
tauArgusOutFiles_location = "C:\TargusOut"
tauArgusInFiles_location = "C:\TargusIn"
mytauArgusOutFiles_location = "C:\MyTargus"
thread_sleep_time = "10000"
oraconn = "jdbc:oracle:thin:@192.168.2.6:1521:BBlockDB"
>>dbLocation: file:c:\databases\BBlocksDB071307
>>Into getTargusInLocation, content:database_location = "file:c:\databases\BBlocksDB071307"
tauArgusOutFiles_location = "C:\TargusOut"
tauArgusInFiles_location = "C:\TargusIn"
mytauArgusOutFiles_location = "C:\MyTargus"
thread_sleep_time = "10000"
oraconn = "jdbc:oracle:thin:@192.168.2.6:1521:BBlockDB"
>>tauArgusInFiles_location: C:\TargusIn
>>Into getTargusOutLocation, content:database_location = "file:c:\databases\BBlocksDB071307"
tauArgusOutFiles_location = "C:\TargusOut"
tauArgusInFiles_location = "C:\TargusIn"
mytauArgusOutFiles_location = "C:\MyTargus"
thread_sleep_time = "10000"
oraconn = "jdbc:oracle:thin:@192.168.2.6:1521:BBlockDB"
>>tauArgusOutFiles_location: C:\TargusOut
>>Into getMyTargusOutLocation, content:database_location = "file:c:\databases\BBlocksDB071307"
tauArgusOutFiles_location = "C:\TargusOut"
tauArgusInFiles_location = "C:\TargusIn"
mytauArgusOutFiles_location = "C:\MyTargus"
thread_sleep_time = "10000"
oraconn = "jdbc:oracle:thin:@192.168.2.6:1521:BBlockDB"
>>mytauArgusOutFiles_location: C:\MyTargus
>>Into getThreadSleepTime, content:database_location = "file:c:\databases\BBlocksDB071307"
tauArgusOutFiles_location = "C:\TargusOut"
tauArgusInFiles_location = "C:\TargusIn"
mytauArgusOutFiles_location = "C:\MyTargus"
thread_sleep_time = "10000"
oraconn = "jdbc:oracle:thin:@192.168.2.6:1521:BBlockDB"
>>thread_sleep_time: 10000
>>Into getOraConnString, content:database_location = "file:c:\databases\BBlocksDB071307"
tauArgusOutFiles_location = "C:\TargusOut"
tauArgusInFiles_location = "C:\TargusIn"
mytauArgusOutFiles_location = "C:\MyTargus"
thread_sleep_time = "10000"
oraconn = "jdbc:oracle:thin:@192.168.2.6:1521:BBlockDB"
>>oraconn: jdbc:oracle:thin:@192.168.2.6:1521:BBlockDB
>>Into createDB..
>>ConnectDB
>>Into getConfiguration...

```

Σχήμα 39: Στιγμιότυπο από το command prompt του application server

9.6 Redeployment

Προκειμένου να εκτελεστεί στη συνέχεια εκτέλεση (execution) του TArgus Building Block redeployment, τα ακόλουθα βήματα είναι απαραίτητο να ακολουθηθούν:

1. Ενημερώστε (update) το **bbTArgus.ear**, το οποίο βρίσκεται στον προκαθορισμένο κατάλογο αρχείων(directory).
2. Ενημερώστε (update) το **bbtargus.jar**, το οποίο βρίσκεται στον προκαθορισμένο κατάλογο αρχείων(directory).

3. Κλείστε το παράθυρο του **application server** και κάντε του **restart**.

Κεφάλαιο 10°

Συμπεράσματα

10.1 Συμπεράσματα

Στην παρούσα εργασία προτάθηκε, παρουσιάστηκε, αναλύθηκε, σχεδιάστηκε και υλοποιήθηκε ένα ολοκληρωμένο σύστημα ελέγχου στατιστικής αποκάλυψης δεδομένων με τη χρήση της τεχνολογίας των web services.

Κατά την μετάδοση στατιστικής πληροφορίας από έναν απομακρυσμένο οργανισμό/στατιστικό γραφείο προς μια κεντρική μονάδα αποθήκευσης στατιστικών δεδομένων, ελλοχεύει ο κίνδυνος της αποκάλυψης ευαίσθητων προσωπικών και μη δεδομένων. Για το σκοπό αυτό έχουν αναπτυχθεί μια σειρά από μεθόδους ελέγχου της στατιστικής αποκάλυψης (statistical disclosure control). Ιδιαίτερο ενδιαφέρον για την ανάπτυξη αυτών των μεθόδων παρουσιάζουν κυρίως εθνικοί/κυβερνητικοί οργανισμοί και ιδρύματα καθώς και φορείς από τον ιατρικό χώρο, όπου τα δεδομένα που διαπραγματεύονται παρουσιάζουν ιδιαίτερο βαθμό ευαισθησίας.

Αν και η ευρεία χρήση του διαδικτύου τα τελευταία χρόνια έχει ωφελήσει σημαντικά στην κατεύθυνση της διάθεσης μια συγκεκριμένης υπηρεσίας επεξεργασίας στατιστικών δεδομένων σε όλο και περισσότερους χρήστες ανεξαρτήτως της τοποθεσίας τους. Ωστόσο, αυτού του είδους η διάθεση των στατιστικών υπηρεσιών κρύβουν αυξημένους κινδύνους αποκάλυψης συγκριτικά με αυτούς που μπορεί να εντοπιστούν σε ένα τοπικό δίκτυο υπολογιστών.

Προκειμένου λοιπόν να περιοριστεί ο κίνδυνος της στατιστικής αποκάλυψης ακόμα και κατά την μετάδοση δεδομένων μέσω του διαδικτύου, η παρούσα μεταπτυχιακή διατριβή παρουσίασε μια ολοκληρωμένη αρχιτεκτονική. Αυτή αποτελείται από έναν client με μια γραφική διεπαφή για την αλληλεπίδραση με τον τελικό χρήστη και από ένα δεύτερο μέρος εγκατεστημένο στην πλευρά του server -το οποίο ονομάστηκε Δομικό Τμήμα ή Building Block. Το εν λόγω Δομικό Τμήμα που προτάθηκε είναι ουσιαστικά ένας wrapper, ένα κομμάτι δηλαδή λογισμικού το οποίο περιλαμβάνει μια εσωτερική (αυτόνομη) εφαρμογή, το τ-Argus και εν συνεχεία μέσω της υιοθέτησης της τεχνολογίας των Web Services την κάνει διαθέσιμη μέσω του διαδικτύου.

Το τ-Argus είναι λογισμικό το οποίο αναπτύχθηκε από το Statistics Netherlands, σκοπός του οποίου είναι η προστασία των πινάκων δεδομένων έναντι του κινδύνου της αποκάλυψής τους. Ο στόχος αυτός επιτυγχάνεται με κατάλληλη διαμόρφωση του εκάστοτε πίνακα δεδομένων ώστε να περιέχει πληροφορία που να είναι κατά το δυνατό λιγότερο λεπτομερής. Έτσι επιτρέπει έναν αριθμό από αλλαγές στον πίνακα δεδομένων: ένας πίνακας μπορεί να επανασχεδιαστεί, με την έννοια του συνδυασμού των στηλών και των γραμμών μεταξύ τους; τα ευαίσθητα κελιά μπορούν αν αποσιωπηθούν (suppressed) και τα υπόλοιπα κελιά μπορούν να παρουσιαστούν με βέλτιστο τρόπο (secondary cell suppression).

Ο χρήστης δημιουργεί και εισάγει τα απαραίτητα αρχεία εισόδου στην client εφαρμογή που είναι εγκατεστημένη στον προσωπικό του υπολογιστή. Τα δεδομένα αυτά αποστέλλονται στον κεντρικό server και τα αποτελέσματα αυτής της επεξεργασίας αποστέλλονται πίσω στον χρήστη (αρχεία εξόδου). Οι απαραίτητοι διάλογοι επικοινωνίας μεταξύ client και web service είναι βασισμένη στο πρωτόκολλο JAX-RPC. Το θέμα της ασφάλειας καθώς και οι λοιπές λεπτομέρειες της επικοινωνίας ρυθμίζονται και εξασφαλίζονται μέσω της χρήσης της πλατφόρμας J2EE.

Ο application server που επιλέχθηκε για την φιλοξενία του υλοποιημένου web service είναι ο BEA Weblogic Server. Το WSDL αρχείο παράγεται με τη βοήθεια του JWSC εργαλείου του BEA Weblogic Server. Χάρης το WSDL αρχείο, το web service του Δομικού Τμήματος γίνεται εύκολα εντοπισμό από τους εκάστοτε clients και επί πλέον συμβάλει στην διευκόλυνση της περαιτέρω χρήσης και επεκτασιμότητας του Building Block.

10.2 Μελλοντικές Επεκτάσεις

Η εφαρμογή που σχεδιάστηκε και υλοποιήθηκε για τις ανάγκες της παρούσας μεταπτυχιακής διατριβής αποτελεί από μόνη της ένα ανεξάρτητο αυτοτελές σύστημα που αποτελείται από το μέρος client με γραφικό interface (GUI) για την αλληλεπίδραση με τον τελικό χρήστη και το μέρος του server όπου βρίσκεται εγκατεστημένο το web service, η εφαρμογή τ-Argus καθώς και η σχεσιακή βάση δεδομένων της εφαρμογής

Παρόλο λοιπόν που το προτεινόμενο σύστημα θεωρείται ως ολοκληρωμένο, μπορεί κάλλιστα να δεχτεί επεκτάσεις ώστε να καλύψει μεγαλύτερο εύρος απαιτήσεων για έλεγχο στατιστικής αποκάλυψης και παραγωγή ασφαλών δεδομένων. Η επιλογή άλλωστε της υιοθέτησης της τεχνολογικής

πλατφόρμας J2EE και ειδικότερα των web services εγγυάται από μόνη της, την διαλειτουργικότητα και την επαναχρησιμοποιησιμότητα της παρούσας εφαρμογής, χαρακτηριστικά που την καθιστούν εύκολα επεκτάσιμη. Οι προτάσεις που ακολουθούν είναι αυτές που θα μπορούσαν ουσιαστικά να αποτελέσουν και μελλοντικές επεκτάσεις του παρόντος συστήματος.

- Επέκταση του υπάρχοντος user interface. Το νέο επεκταμένο user interface θα πρέπει να παρέχει στο χρήστη περισσότερες παραμέτρους (συσχετιζόμενες με την τ-Argus εφαρμογή, π.χ. καθορισμός αλγορίθμου, καθορισμός κανόνων ασφάλειας, εύρος ασφάλειας, εύρος συχνότητας, συντελεστής λάμδα, κ.α.) προς διαμόρφωση. Αυτό θα έχει σαν αποτέλεσμα την αυτόματη δημιουργία μερικών ή όλων των αρχείων εισόδου
- Ο client που αναπτύχθηκε σε αυτή την εργασία υλοποιήθηκε για να επιτευχθεί η επίδειξη των λειτουργιών του όλου συστήματος Building Block. Ωστόσο αυτός ο client μπορεί να αντικατασταθεί ανά πάσα στιγμή από κάποια άλλη πιο σύνθετη –ενδεχόμενος ήδη υπάρχουσα ή ακόμη και νέα – client εφαρμογή. Έτσι για παράδειγμα ένας νέος client που θα μπορούσε να επικοινωνήσει άριστα με το παρών σύστημα του Building Block, θα μπορούσε να είναι ένα γραφικό εργαλείο –εγκατεστημένο τοπικά στο pc του χρήστη- το οποίο ανακτά στατιστικά δεδομένα από κάποια βάση και επιτρέπει την όποια τροποποίηση και μορφοποίηση τους. Το μόνο που απαιτείται σε αυτή τη περίπτωση για την άριστη επικοινωνία του εργαλείου αυτού με το Building Block σύστημα είναι η γνώση του WSDL αρχείου του αντίστοιχου web service και φυσικά η δυνατότητα πρόσβασης στο διαδίκτυο. Το WSDL αρχείο περιγράφει όλες τις απαραίτητες πληροφορίες για τον εντοπισμό του web service καθώς και τον τρόπο επικοινωνίας με αυτό.
- Το παρών σύστημα μπορεί να αποτελέσει υποσύνολο ενός μεγαλύτερου συστήματος για την στατιστική επεξεργασία δεδομένων. Το BB web service που έχει υλοποιηθεί μπορεί να επικοινωνήσει με άλλα web services στα πλαίσια ενός μεγαλύτερου ενιαίου συστήματος. Η επικοινωνία και σε αυτή την περίπτωση μπορεί να επιτευχθεί με την έκδοση του WSDL όπως ακριβώς περιγράφηκε παραπάνω. Έτσι λοιπόν η προσέγγιση και η υλοποίηση που προτάθηκε και υλοποιήθηκε στην παρούσα εργασία μπορεί να αποτελέσει υποσύστημα οποιουδήποτε μεγαλύτερου συστήματος –όποια κι αν είναι η πλατφόρμα λειτουργίας του-

- που θέλει να πραγματοποιήσει έλεγχο αποκάλυψης των στατιστικών του δεδομένων και ταυτόχρονα παραγωγή ασφαλών πινάκων.
- Με μικρές τροποποιήσεις όσων αφορά την πλευρά του server είναι δυνατό να τροποποιηθεί το υφιστάμενο web service προκειμένου να εκτελεί στατιστικό έλεγχο αποκάλυψης και παραγωγή ασφαλών δεδομένων, κάνοντας χρήση της εφαρμογής μ-Argus. Η εφαρμογή μ-Argus έχει αναπτυχθεί και αυτή από το Statistics Netherlands και εξειδικεύεται σε περιπτώσεις μικροδεδομένων (microdata).

Παράρτημα

1. WSDL αρχείο του τ-Argus service

```
<?xml version='1.0' encoding='UTF-8'?>
<definitions name="TArgusBBServi ceDefi ni ti ons"
targetNamespace="http://exampl e.org"
xml ns="http://schemas.xml soap.org/wsdl/" xml ns:s0="http://exampl e.org"
xml ns:s1="http://schemas.xml soap.org/wsdl/soap/">
  <types>
    <xs:schema attributeFormDefaul t="unqual i fi ed"
elementFormDefaul t="qual i fi ed" targetNamespace="http://exampl e.org"
xml ns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:el ement name="checkAdmi nAI l Jobs">
        <xs:compl exType>
          <xs:sequence/>
        </xs:compl exType>
      </xs:el ement>
      <xs:el ement name="checkAdmi nAI l JobsResponse">
        <xs:compl exType>
          <xs:sequence>
            <xs:el ement name="return" type="j ava:TArgusConfi g"
xml ns:j ava="j ava:i b"/>
          </xs:sequence>
        </xs:compl exType>
      </xs:el ement>
      <xs:el ement name="setBBI nputFi l esAndQueueJob">
        <xs:compl exType>
          <xs:sequence>
            <xs:el ement name="sessi onId" type="xs:stri ng"/>
            <xs:el ement name="userId" type="xs:stri ng"/>
            <xs:el ement name="batchPath" type="xs:stri ng"/>
            <xs:el ement name="dataPath" type="xs:stri ng"/>
            <xs:el ement name="metadataPath" type="xs:stri ng"/>
          </xs:sequence>
        </xs:compl exType>
      </xs:el ement>
      <xs:el ement name="setBBI nputFi l esAndQueueJobResponse">
        <xs:compl exType>
```

```
<xs: sequence>
  <xs: element name="return" type="xs: int"/>
</xs: sequence>
</xs: complexType>
</xs: element>
<xs: element name="checkAdminJobsPerUser">
  <xs: complexType>
    <xs: sequence>
      <xs: element name="userId" type="xs: string"/>
    </xs: sequence>
  </xs: complexType>
</xs: element>
<xs: element name="checkAdminJobsPerUserResponse">
  <xs: complexType>
    <xs: sequence>
      <xs: element name="return" type="java: TArgusConfig"
xml ns: java="java: ib"/>
    </xs: sequence>
  </xs: complexType>
</xs: element>
<xs: element name="initiateBBSession">
  <xs: complexType>
    <xs: sequence>
      <xs: element name="userId" type="xs: string"/>
    </xs: sequence>
  </xs: complexType>
</xs: element>
<xs: element name="initiateBBSessionResponse">
  <xs: complexType>
    <xs: sequence>
      <xs: element name="return" type="xs: string"/>
    </xs: sequence>
  </xs: complexType>
</xs: element>
<xs: element name="checkUserAllJobStatus">
  <xs: complexType>
    <xs: sequence>
      <xs: element name="sessionId" type="xs: string"/>
      <xs: element name="jobId" type="xs: string"/>
    </xs: sequence>
  </xs: complexType>
```

```
</xs:element>
<xs:element name="checkUserAllJobStatusResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="return" type="java:TArgusConfig"
xml ns:java="java:ib"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="closeBBClientSession">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="closeBBClientSessionResponse">
  <xs:complexType>
    <xs:sequence/>
      <xs:element name="return" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="getTArgusBBUserId">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="loginName" type="xs:string"/>
      <xs:element name="password" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="getTArgusBBUserIdResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="return" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="send2BBInputFiles">
  <xs:complexType>
    <xs:sequence>
```

```
        <xs:element name="sessionId" type="xs:string"/>
        <xs:element name="f" type="java:File"
xml ns:java="java:java.io"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="send2BBInputFilesResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="return" type="xs:boolean"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="checkTArgusJobStatus">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="sessionId" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="checkTArgusJobStatusResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="return" type="java:TArgusConfig"
xml ns:java="java:ib"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="receiveFromBBOutputFiles">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="sessionId" type="xs:string"/>
            <xs:element name="f" type="java:File"
xml ns:java="java:java.io"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="receiveFromBBOutputFilesResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="return" type="xs:boolean"/>
```

```
</xs: sequence>
</xs: complexType>
</xs: element>
</xs: schema>
<xs: schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="java: examples. webservices. complex"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="java: java.lang"/>
  <xs:complexType name="BasicStruct">
    <xs:complexContent>
      <xs:extension base="java: Thread" xmlns:java="java: java.lang"/>
    </xs:complexContent>
  </xs:complexType>
</xs: schema>
<xs: schema attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="java: java.lang"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="Thread">
    <xs:sequence>
      <xs:element minOccurs="1" name="Priority" nillable="false"
type="xs: int"/>
      <xs:element minOccurs="1" name="Name" nillable="true"
type="xs: string"/>
      <xs:element minOccurs="1" name="Daemon" nillable="false"
type="xs: boolean"/>
      <xs:element minOccurs="1" name="ContextClassLoader"
nillable="true" type="java: ClassLoader" xmlns:java="java: java.lang"/>
      <xs:element minOccurs="1" name="DefaultUncaughtExceptionHandler"
nillable="true" type="java: Thread_UncaughtExceptionHandler"
xmlns:java="java: java.lang"/>
      <xs:element minOccurs="1" name="UncaughtExceptionHandler"
nillable="true" type="java: Thread_UncaughtExceptionHandler"
xmlns:java="java: java.lang"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ClassLoader"/>
  <xs:complexType name="Thread_UncaughtExceptionHandler"/>
</xs: schema>
```



```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="java:java.io"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="File"/>
</xs:schema>

<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="java:ib"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="TArgusConfig">
    <xs:sequence>
      <xs:element minOccurs="1" name="TArgusHomeDir" nillable="true"
type="xs:string">
        <xs:annotation>
          <xs:documentation>Gets the TArgusHomeDir value for this
TArgusConfiguration. </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="BatchFileDir" nillable="true"
type="xs:string">
        <xs:annotation>
          <xs:documentation>Gets the batchFileDir value for this
TArgusConfiguration. </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="BatchFileName" nillable="true"
type="xs:string">
        <xs:annotation>
          <xs:documentation>Gets the batchFileName value for this
TArgusConfiguration. </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="DataFileDir" nillable="true"
type="xs:string">
        <xs:annotation>
          <xs:documentation>Gets the dataFileDir value for this
TArgusConfiguration. </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" name="DataFileName" nillable="true"
type="xs:string">
        <xs:annotation>
```

```
<xs:documentation>Gets the dataFileName value for this
TArgusConfiguration. </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="ExeFileDir" nillable="true"
type="xs:string">
<xs:annotation>
<xs:documentation>Gets the exeFileDir value for this
TArgusConfiguration. </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="ExeFileName" nillable="true"
type="xs:string">
<xs:annotation>
<xs:documentation>Gets the exeFileName value for this
TArgusConfiguration. </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="LogFileDir" nillable="true"
type="xs:string">
<xs:annotation>
<xs:documentation>Gets the logFileDir value for this
TArgusConfiguration. </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="LogFileName" nillable="true"
type="xs:string">
<xs:annotation>
<xs:documentation>Gets the logFileName value for this
TArgusConfiguration. </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="MetaDataFileDir" nillable="true"
type="xs:string">
<xs:annotation>
<xs:documentation>Gets the metaDataFileDir value for this
TArgusConfiguration. </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="MetaDataFileName"
nillable="true" type="xs:string">
```

```
<xs: annotation>
  <xs: documentation>Gets the metaDataFileName value for this
TArgusConfiguration. </xs: documentation>
</xs: annotation>
</xs: element>
<xs: element minOccurs="1" name="JobId" nillable="true"
type="xs:string">
  <xs: annotation>
    <xs: documentation>Gets the jobId value for this
TArgusConfiguration. </xs: documentation>
  </xs: annotation>
</xs: element>
<xs: element minOccurs="1" name="SessionId" nillable="true"
type="xs:string">
  <xs: annotation>
    <xs: documentation>Gets the sessionId value for this
TArgusConfiguration. </xs: documentation>
  </xs: annotation>
</xs: element>
<xs: element minOccurs="1" name="UserId" nillable="true"
type="xs:string">
  <xs: annotation>
    <xs: documentation>Gets the userId value for this
TArgusConfiguration. </xs: documentation>
  </xs: annotation>
</xs: element>
<xs: element minOccurs="1" name="Status" nillable="true"
type="xs:string">
  <xs: annotation>
    <xs: documentation>Gets the status value for this
TArgusConfiguration. </xs: documentation>
  </xs: annotation>
</xs: element>
<xs: element minOccurs="1" name="TArgusReturnCode"
nillable="true" type="xs:string">
  <xs: annotation>
    <xs: documentation>Gets the TArgusReturnCode value for this
TArgusConfiguration. </xs: documentation>
  </xs: annotation>
</xs: element>
</xs: sequence>
```

```
</xs:complexType>
</xs:schema>
</types>
<message name="checkAdminAllJobs">
  <part element="s0:checkAdminAllJobs" name="parameters"/>
</message>
<message name="checkAdminAllJobsResponse">
  <part element="s0:checkAdminAllJobsResponse" name="parameters"/>
</message>
<message name="setBBInputFilesAndQueueJob">
  <part element="s0:setBBInputFilesAndQueueJob" name="parameters"/>
</message>
<message name="setBBInputFilesAndQueueJobResponse">
  <part element="s0:setBBInputFilesAndQueueJobResponse"
name="parameters"/>
</message>
<message name="checkAdminJobsPerUser">
  <part element="s0:checkAdminJobsPerUser" name="parameters"/>
</message>
<message name="checkAdminJobsPerUserResponse">
  <part element="s0:checkAdminJobsPerUserResponse" name="parameters"/>
</message>
<message name="initiateBBSession">
  <part element="s0:initiateBBSession" name="parameters"/>
</message>
<message name="initiateBBSessionResponse">
  <part element="s0:initiateBBSessionResponse" name="parameters"/>
</message>
<message name="checkUserAllJobStatus">
  <part element="s0:checkUserAllJobStatus" name="parameters"/>
</message>
<message name="checkUserAllJobStatusResponse">
  <part element="s0:checkUserAllJobStatusResponse" name="parameters"/>
</message>
<message name="closeBBClientSession">
  <part element="s0:closeBBClientSession" name="parameters"/>
</message>
<message name="closeBBClientSessionResponse">
  <part element="s0:closeBBClientSessionResponse" name="parameters"/>
</message>
<message name="getTArgusBBUserID">
```

```

    <part element="s0: getTArgusBBUserI d" name="parameters"/>
  </message>
  <message name="getTArgusBBUserI dResponse">
    <part element="s0: getTArgusBBUserI dResponse" name="parameters"/>
  </message>
  <message name="send2BBI nputFi l es">
    <part element="s0: send2BBI nputFi l es" name="parameters"/>
  </message>
  <message name="send2BBI nputFi l esResponse">
    <part element="s0: send2BBI nputFi l esResponse" name="parameters"/>
  </message>
  <message name="checkTArgusJobStatus">
    <part element="s0: checkTArgusJobStatus" name="parameters"/>
  </message>
  <message name="checkTArgusJobStatusResponse">
    <part element="s0: checkTArgusJobStatusResponse" name="parameters"/>
  </message>
  <message name="recei veFromBBOutputFi l es">
    <part element="s0: recei veFromBBOutputFi l es" name="parameters"/>
  </message>
  <message name="recei veFromBBOutputFi l esResponse">
    <part element="s0: recei veFromBBOutputFi l esResponse"
name="parameters"/>
  </message>
  <portType name="Compl exPortType">
    <operation name="checkAdmi nAI l Jobs" parameterOrder="parameters">
      <i nput message="s0: checkAdmi nAI l Jobs"/>
      <output message="s0: checkAdmi nAI l JobsResponse"/>
    </operation>
    <operation name="setBBI nputFi l esAndQueueJob"
parameterOrder="parameters">
      <i nput message="s0: setBBI nputFi l esAndQueueJob"/>
      <output message="s0: setBBI nputFi l esAndQueueJobResponse"/>
    </operation>
    <operation name="checkAdmi nJobsPerUser" parameterOrder="parameters">
      <i nput message="s0: checkAdmi nJobsPerUser"/>
      <output message="s0: checkAdmi nJobsPerUserResponse"/>
    </operation>
    <operation name="i ni ti ateBBSessi on" parameterOrder="parameters">
      <i nput message="s0: i ni ti ateBBSessi on"/>
      <output message="s0: i ni ti ateBBSessi onResponse"/>

```

```

</operati on>
<operati on name="checkUserAl l JobStatus" parameterOrder="parameters">
  <i nput message="s0: checkUserAl l JobStatus" />
  <output message="s0: checkUserAl l JobStatusResponse" />
</operati on>
<operati on name="cl oseBBCl i entSessi on" parameterOrder="parameters">
  <i nput message="s0: cl oseBBCl i entSessi on" />
  <output message="s0: cl oseBBCl i entSessi onResponse" />
</operati on>
<operati on name="getTArgusBBUserl d" parameterOrder="parameters">
  <i nput message="s0: getTArgusBBUserl d" />
  <output message="s0: getTArgusBBUserl dResponse" />
</operati on>
<operati on name="send2BBI nputFi l es" parameterOrder="parameters">
  <i nput message="s0: send2BBI nputFi l es" />
  <output message="s0: send2BBI nputFi l esResponse" />
</operati on>
<operati on name="checkTArgusJobStatus" parameterOrder="parameters">
  <i nput message="s0: checkTArgusJobStatus" />
  <output message="s0: checkTArgusJobStatusResponse" />
</operati on>
<operati on name="recei veFromBBOutputFi l es"
parameterOrder="parameters">
  <i nput message="s0: recei veFromBBOutputFi l es" />
  <output message="s0: recei veFromBBOutputFi l esResponse" />
</operati on>
</portType>
<bi ndi ng name="TArgusBBServi ceSoapBi ndi nghttp"
type="s0: Compl exPortType">
  <s1: bi ndi ng styl e="document"
transport="http: //schemas. xml soap. org/soap/http" />
  <operati on name="checkAdmi nAl l Jobs">
    <s1: operati on soapActi on="" styl e="document" />
    <i nput>
      <s1: body parts="parameters" use="l i teral " />
    </i nput>
    <output>
      <s1: body parts="parameters" use="l i teral " />
    </output>
  </operati on>
  <operati on name="setBBI nputFi l esAndQueueJob">

```

```
<s1:operation soapAction="" style="document"/>
<input>
  <s1:body parts="parameters" use="literal"/>
</input>
<output>
  <s1:body parts="parameters" use="literal"/>
</output>
</operation>
<operation name="checkAdminJobsPerUser">
  <s1:operation soapAction="" style="document"/>
  <input>
    <s1:body parts="parameters" use="literal"/>
  </input>
  <output>
    <s1:body parts="parameters" use="literal"/>
  </output>
</operation>
<operation name="initiateBBSession">
  <s1:operation soapAction="" style="document"/>
  <input>
    <s1:body parts="parameters" use="literal"/>
  </input>
  <output>
    <s1:body parts="parameters" use="literal"/>
  </output>
</operation>
<operation name="checkUserAllJobStatus">
  <s1:operation soapAction="" style="document"/>
  <input>
    <s1:body parts="parameters" use="literal"/>
  </input>
  <output>
    <s1:body parts="parameters" use="literal"/>
  </output>
</operation>
<operation name="closeBBClientSession">
  <s1:operation soapAction="" style="document"/>
  <input>
    <s1:body parts="parameters" use="literal"/>
  </input>
  <output>
```

```
<s1:body use="literal"/>
</output>
</operation>
<operation name="getTArgusBBUserId">
  <s1:operation soapAction="" style="document"/>
  <input>
    <s1:body parts="parameters" use="literal"/>
  </input>
  <output>
    <s1:body parts="parameters" use="literal"/>
  </output>
</operation>
<operation name="send2BBInputFiles">
  <s1:operation soapAction="" style="document"/>
  <input>
    <s1:body parts="parameters" use="literal"/>
  </input>
  <output>
    <s1:body parts="parameters" use="literal"/>
  </output>
</operation>
<operation name="checkTArgusJobStatus">
  <s1:operation soapAction="" style="document"/>
  <input>
    <s1:body parts="parameters" use="literal"/>
  </input>
  <output>
    <s1:body parts="parameters" use="literal"/>
  </output>
</operation>
<operation name="receiveFromBBOutputFiles">
  <s1:operation soapAction="" style="document"/>
  <input>
    <s1:body parts="parameters" use="literal"/>
  </input>
  <output>
    <s1:body parts="parameters" use="literal"/>
  </output>
</operation>
</binding>
<service name="TArgusBBSERVICE">
```



```
<port binding="s0: TArgusBBSoapBindinghttp" name="TArgusBBServicPort">
  <s1: address
Location="http://localhost: 7001/complex/TArgusBBServic"/>
  </port>
</servic>
</definitions>
```

Συντομεύσεις

Abbreviation	Meaning
API	Application Programming Interface
BB	Building Block
BNF	Backus-Naur Form
CGI	Common Gateway Interface
COMEXT	Statistics of the European Union's internal and external trade
DLL	Dynamic Link Library
EJB	Enterprises Java Bean
FTP	File Transfer Protocol
GUI	Graphical User Interface
JAX-RPC	Java API for XML-Based Remote Procedure Call
JDBC	(Java DataBase Connectivity). A programming interface that lets Java applications access a database via the SQL language. Since Java interpreters (Java Virtual Machines) are available for all major client platforms, this allows a platform-independent database application to be written. JDBC is the Java counterpart of Microsoft's ODBC.
JMS	Java Message Service
JTA	Java Transaction API (Sun)
J2EE	Java 2 Platform, Enterprise Edition (Sun)
J2ME	Java 2 Micro Edition (Sun)
J2SE	Java 2.0 Standard Edition (Sun)
OS	Operating System
RDBMS	Relational Database Management System
RMI/IIOP	Remote Method Invocation over Internet Inter-ORB (Object Request Border-CORBA) Protocol. An extension to Sun's RMI procedure call that

	enables Java programs to execute CORBA objects. It allows Java access to non-Java processes via CORBA
RPC	Remote Procedure Call
SAX-RPC	Simple API for XML – Bases Remote Procedure Call
SDC	Statistical Disclosure Control
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
UML	Unified Modeling Language
WS	Web Services
XML	Extensible Markup Language
WSDL	Web Services Description Language
XT-NET	External Trade Information System

Βιβλιογραφία

- [1] L. C. R. J. Willenborg and T. de Waal. *Statistical Disclosure Control in Practice*. Springer–Verlag, New York, 1996.
- [2] L. C. R. J. Willenborg and T. de Waal. *Elements of Statistical Disclosure Control*. Springer–Verlag, New York, 2001.
- [3] National Institute of Statistical Sciences Research Triangle Park, NC 27709-4006, USA
National: <http://www.niss.org/dg/index.html>
- [4] Shanti Gomatam, Alan F. Karr, Chunhua “Charlie” Liu and Ashish P. Sanil¹. Data Swapping: A Risk–Utility Framework and Web Service Implementation. National Institute of Statistical Sciences Research Triangle Park, NC 27709-4006, USA.
- [5] <http://www.niss.org/dgii/presentations.html>
- [6] Ashish Sanil, Shanti Gomatam, Alan F. Karr National Institute of Statistical Sciences. NISS WebSwap: A Web Service for Data Swapping. National Institute of Statistical Sciences Research Triangle Park, NC 27709-4006, USA, 2002.
- [7] Latanya Sweeney, Associate Professor of Computer Science, Technology and Policy, Institute for Software Research, School of Computer Science, Carnegie Mellon University:
<http://lab.privacy.cs.cmu.edu/people/sweeney/index.html>
- [8] Kooiman, P., Nobel, J.R. and Willenborg, L.C.R.J. (1999), ‘Statistical data protection at Statistics Netherlands’ in Netherlands Official Statistics, Volume 14, spring 1999, pp. 21-25.
- [9] Schulte Nordholt, E. (1998), ‘Imputation: methods, simulation experiments and practical examples’ in International Statistical Review, Volume 66, Nr. 2, pp. 157-180.
- [10] Schulte Nordholt, E. (2001), ‘Progress in the implementation of SDC methods and techniques in Central and Eastern Europe, List of key issues for discussion’, paper presented at the second Joint Eurostat / UN-ECE Work Session on Statistical Data Confidentiality held in Skopje in March 2001.

- [11] Willenborg, L.C.R.J. and De Waal, A.G. (1996), 'Statistical disclosure control in practice', Lecture Notes in Statistics 111, Springer-Verlag, New York.
- [12] Willenborg, L.C.R.J. and De Waal, A.G. (2001), 'Elements of statistical disclosure control', Lecture Notes in Statistics 155, Springer-Verlag, New York.
- [13] Statistical Disclosure Control of the Statistics Netherlands Employment and Earnings Data Submitted by Statistics Netherlands, STATISTICAL COMMISSION and COMMISSION OF THE ECONOMIC COMMISSION FOR EUROPE Joint ECE/Eurostat Work Session on Statistical Data Confidentiality, Thessaloniki, Greece, 8-10 March 1999.
- [14] Statistical Disclosure Control (SDC) in Practice: Some Examples In Official Statistics Of Statistics Netherlands, Skopje, 14-16 March 2001
- [15] FASTER project, ftp://ftp.cordis.europa.eu/pub/ist/docs/ka3/iep_faster.pdf
- [16] Work Session on Statistical Data Confidentiality Geneva 9-11 November 2005, Eurostat.
- [17] Lucia Buzzigoli, Luigi Biggeri. Statistical Disclosure Control and Data Access for Research Purposes: Critical Issues and Possible Solutions, Department of Statistics "G. Parenti", University of Florence, Italy.
- [18] CASC Project: <http://neon.vb.cbs.nl/casc/>
- [19] http://en.wikipedia.org/wiki/Main_Page
- [20] τ-Argus v3.1, User's Manual, Statistics Netherlands (Anco Hundepool, Aad van de Wetering and Ramya Ramaswamy, Peter-Paul de Wolf (Modular/Hitas), Sarah Giessing (GHMiter), Mateo Fishetti, and Juan-Jose Salazar (Optimisation), Jordi Castro (Network Solutions), Philip Lowthian (manual).
- [21] Monographs of official statistics Work session on statistical data confidentiality Luxembourg, 7 to 9 April 2003.
- [22] http://statistics.gov.uk/about/methodology/general_methodology/sdc.asp
- [23] <http://www.sasc.co.uk/Guides/sdc.htm>
- [24] <http://stats.oecd.org/glossary/>

- [25] <http://www.geog.leeds.ac.uk/projects/sdc.html>
- [26] <http://neon.vb.cbs.nl/casc/NTTSEric.pdf>
- [27] <http://www.unece.org/stats/documents/1999/03/confidentiality/8.e.pdf>
- [28] www.bea.com
- [29] www.sun.com
- [30] Designing Web Services with the J2EE 1-4 Platform -- JAX-RPC, SOAP, and XML Technologies, Mark Hapner, Distinguished EngineerLead Architect, J2EE platform Sun Microsystems, January 2004
- [31] The J2EE™ 1.4 Tutorial, Eric Armstrong, Jennifer Ball, Stephanie Bodoff, Debbie Bode, Carson, Ian Evans, Dale Green, Kim Haase, Eric Jendrock, Sun Microsystems, December 2005
- [32] Developer's Guide to Building XML-based Web Services with the Java 2 Platform, Enterprise Edition (J2EE), By James Kao (Prepared for Sun Microsystems, Inc.), June 2001
- [33] The Java™ Web Services Tutorial For Java Web Services Developer's Pack, v1.6, Sun Microsystems, June 2005
- [34] Pushing the SOAP EnvelopeWithWeb Services for Scientific Computing, Robert A. van Engelen, Department of Computer Science and School of Computational Science and Information Technology Florida State University
- [35] Service Component Architecture Building Systems using a Service Oriented Architecture A Joint Whitepaper by BEA, IBM, Interface21, IONA, Oracle, SAP, Siebel, Sybase. Version 0.9 November 2005
- [36] Programming Web Services for WebLogic Server Version 9.0, BEA, July 22, 2005
- [37] Programming WebLogic Security Version 9.0, BEA, August 2005
- [38] Deploying Applications to WebLogic Server Version 9.0 , BEA, December 19, 2005