



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

**Δυναμικές Ροές Εργασίας στις Επιχειρηματικές Εφαρμογές
Ενορχήστρωση των Διαδικτυακών Υπηρεσιών για την
Ανάπτυξη Επιχειρηματικών Διαδικασιών στη Βάση
Επιχειρηματικών Κανόνων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σπυρίδων Λαγάρας

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Δημήτριος Ασκούνης

Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2008



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

**Δυναμικές Ροές Εργασίας στις Επιχειρηματικές Εφαρμογές
Ενορχήστρωση των Δικτυακών Υπηρεσιών για την
Ανάπτυξη Επιχειρηματικών Διαδικασιών**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σπυρίδων Λαγάρας

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Δημήτριος Ασκούνης
Επίκουρος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 2008.

.....

Δημήτριος Ασκούνης

Επίκουρος Καθηγητής Ε.Μ.Π.

.....

Ιωάννης Ψαρράς

Καθηγητής Ε.Μ.Π.

.....

Γρηγόριος Μέντζας

Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2008

.....
Σπυρίδων Χ. Λαγάρας

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Σπυρίδων Χ. Λαγάρας, 2008

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

ΕΥΧΑΡΙΣΤΙΕΣ

Η διπλωματική αυτή εργασία εντάσσεται στα πλαίσια της ερευνητικής δραστηριότητας του εργαστηρίου Συστημάτων Αποφάσεων και Διοίκησης. Προσωπικά, θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα καθηγητή μου, κύριο Δημήτριο Ασκούνη και τον υποψήφιο διδάκτορα Γιώργο Γκιώνη, οι οποίοι μου έδωσαν τη ευκαιρία να μυηθώ σε ένα πραγματικά ενδιαφέρον κόσμο που άπτεται των σύγχρονων επιχειρησιακών εφαρμογών πληροφορικής, συνέβαλαν στο να τον κατακτήσω, με υποστήριξαν, με καθοδήγησαν με συνέπεια και υπευθυνότητα καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας, μου δίδαξαν ένα διαφορετικό τρόπο σκέψης και θεώρησης και αποτέλεσαν πηγή έμπνευσης και τεχνογνωσίας. Επίσης, θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για την κατανόηση και την υπομονή τους κατά το διάστημα υλοποίησης της διπλωματικής εργασίας.

Οκτώβριος 2008
Σπυρίδων Χ. Λαγάρας

Περιεχόμενα

1	ΕΙΣΑΓΩΓΗ	- 2 -
1.1	ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ	- 3 -
1.2	Ο ΡΟΛΟΣ ΤΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΣΤΗ ΣΥΓΧΡΟΝΗ ΕΠΙΧΕΙΡΗΣΗ	- 3 -
1.3	Ο ΡΟΛΟΣ ΤΩΝ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ ΣΤΗ ΣΥΓΧΡΟΝΗ ΕΠΙΧΕΙΡΗΣΗ	- 6 -
1.3.1	<i>Ο Κύκλος ζωής των Επιχειρηματικών Διαδικασιών</i>	- 7 -
1.3.1.1	Σχεδιασμός και Ανάλυση (Design and Analysis)	- 8 -
1.3.1.2	Διαμόρφωση (Configuration)	- 8 -
1.3.1.3	Εκτέλεση (Enactment)	- 9 -
1.3.1.4	Αποτίμηση (Evaluation)	- 9 -
1.3.2	<i>Κατηγοριοποίηση των Επιχειρηματικών Διαδικασιών</i>	- 9 -
1.3.2.1	Οργανωσιακές και Λειτουργικές (Organizational Versus Operational)	- 9 -
1.3.2.2	Ενδοεπιχειρησιακές Διαδικασίες και Χορογραφίες Διαδικασιών (Intraorganizational Processes and Process Choreographies)	- 10 -
1.4	Ο ΡΟΛΟΣ ΤΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΝΑΛΛΑΓΩΝ ΣΤΗ ΣΥΓΧΡΟΝΗ ΕΠΙΧΕΙΡΗΣΗ	- 11 -
1.4.1	<i>Χαλαρή και Ισχυρή Σύζευξη</i>	- 12 -
2	ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΕΡΓΑΛΕΙΑ	- 15 -
2.1	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΣΑΝΑΤΟΛΙΣΜΕΝΗ ΣΤΙΣ ΥΠΗΡΕΣΙΕΣ (SERVICE-ORIENTED ARCHITECTURE)	- 16 -
2.1.1	<i>Γενική Περιγραφή</i>	- 16 -
2.1.2	<i>Βασικά Χαρακτηριστικά</i>	- 18 -
2.1.2.1	Υπηρεσίες (Services)	- 18 -
2.1.2.2	Επιχειρησιακός Δίαυλος Υπηρεσιών (Enterprise Service Bus - ESB)	- 23 -
2.1.2.2.1	Περιγραφή και Χαρακτηριστικά	- 23 -
2.1.2.2.2	Συσχέτιση Δεδομένων (Data Mapping)	- 24 -
2.1.2.2.3	Ευφυής Δρομολόγηση (Intelligent Routing)	- 25 -
2.1.2.2.4	Ενασχόληση με Θέματα Ασφαλείας	- 25 -
2.1.2.2.5	Ενασχόληση με Θέματα Αξιοπιστίας	- 25 -
2.1.2.2.6	Διαχείριση Υπηρεσιών (Service Management)	- 25 -
2.1.2.2.7	Επίβλεψη και Αρχειακή Καταγραφή (Monitoring and Logging)	- 25 -
2.1.2.2.8	Επίβλεψη Επιχειρηματικής Δραστηριότητας (Business Activity Monitoring - BAM)	- 26 -
2.1.2.2.9	Υποστήριξη κατά την Υλοποίηση Υπηρεσιών (Service Implementation Support)	- 26 -
2.1.2.2.10	Σχεδιαστικές Αρχιτεκτονικές Προσεγγίσεις	- 26 -
2.1.2.2.11	Οδηγούμενο από πρωτόκολλο και Οδηγούμενο από API ESB (Protocol-Driven versus API-Driven ESB)	- 29 -
2.1.2.3	Αποθήκη και Κατάλογος Υπηρεσιών (Service Repository – Service Registry)	- 29 -
2.1.2.4	Χαλαρή Σύζευξη (Loose-Coupling)	- 30 -
2.1.2.5	Αρχιτεκτονικά Πρότυπα με Βάση την SOA Αρχιτεκτονική (SOA-based Architectural Models)	- 30 -
2.1.2.5.1	Λογικό – Επιχειρηματικό Μοντέλο (Logical - Business Architectural Model)	- 30 -
2.1.2.5.2	Τεχνικό Αρχιτεκτονικό Μοντέλο (Technical Architectural Model)	- 30 -
2.1.2.5.3	Συνδυασμός Αρχιτεκτονικών Μοντέλων (Mixed Architectural Model)	- 31 -
2.1.2.6	Ασφάλεια (Security)	- 31 -
2.1.2.6.1	Βασικά Χαρακτηριστικά	- 31 -
2.1.2.6.2	Ετερογένεια και Ασφάλεια	- 32 -
2.1.2.6.3	Κατανεμημένες Διαδικασίες και Επίπεδα Αφαιρετικότητας	- 32 -
2.2	SOA ΤΕΧΝΟΛΟΓΙΕΣ	- 33 -
2.2.1	<i>Διαδικτυακές Υπηρεσίες (Web Services)</i>	- 33 -
2.2.2	<i>Διαχείριση Επιχειρηματικών Διαδικασιών BPM (Business Process Management)</i>	- 36 -

2.2.3	<i>BPEL (Business Process Execution Language)</i>	- 37 -
2.2.3.1	ΕΙΣΑΓΩΓΗ	- 37 -
2.2.3.2	Χαρακτηριστικά	- 37 -
2.2.3.3	Σύνθεση Υπηρεσιών (Service Composition)	- 38 -
2.2.3.3.1	Ενορχήστρωση και Χορογραφία	- 40 -
2.2.3.4	Εκτελέσιμες και Αφαιρετικές Διαδικασίες (Executable and Abstract Processes)	- 42 -
3	ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΙ ΚΑΝΟΝΕΣ (BUSINESS RULES)	- 44 -
3.1	ΕΙΣΑΓΩΓΗ	- 45 -
3.2	ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΙ ΚΑΝΟΝΕΣ (BUSINESS RULES)	- 46 -
3.3	ΜΗΧΑΝΕΣ ΠΑΡΑΓΩΓΗΣ ΚΑΝΟΝΩΝ (RULES ENGINES)	- 50 -
3.3.1	<i>Προτυποποίηση των Συστημάτων Κανόνων Παραγωγής</i>	- 53 -
3.4	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΓΙΑ ΤΗΝ ΧΡΗΣΗ ΚΑΝΟΝΩΝ ΣΤΙΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ ΣΤΟ ΠΛΑΙΣΙΟ ΤΗΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΠΡΟΣΑΝΑΤΟΛΙΣΜΕΝΗΣ ΣΕ ΥΠΗΡΕΣΙΕΣ	- 54 -
4	ΥΛΟΠΟΙΗΣΗ ΠΡΩΤΟΤΥΠΗΣ ΕΦΑΡΜΟΓΗΣ	- 58 -
4.1	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	- 59 -
4.2	ΑΝΤΙΣΤΟΙΧΙΣΗ ΑΡΧΙΤΕΚΤΟΝΙΚΩΝ ΣΤΟΙΧΕΙΩΝ ΜΕ ΕΡΓΑΛΕΙΑ	- 61 -
4.3	ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΤΕΧΝΟΛΟΓΙΚΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΕΡΓΑΛΕΙΩΝ	- 61 -
4.3.1	<i>JavaEE 5</i>	- 61 -
4.3.2	<i>GlassFish ESB</i>	- 65 -
4.3.2.1	Χαρακτηριστικά - Αρχιτεκτονική	- 65 -
4.3.2.1.1	GlassFish Application Server	- 65 -
4.3.2.1.2	Java Business Integration (JBI)	- 67 -
4.3.2.1.3	NetBeans 6.1	- 68 -
4.3.2.2	Εγκατάσταση	- 73 -
4.3.3	<i>Ο εξυπηρετητής εφαρμογών JBoss AS</i>	- 73 -
4.3.4	<i>ILOG JRules</i>	- 75 -
4.3.4.1	Χαρακτηριστικά - Αρχιτεκτονική	- 75 -
4.3.4.2	Εγκατάσταση	- 83 -
5	ΠΕΡΙΓΡΑΦΗ ΠΕΡΙΠΤΩΣΕΩΝ	- 85 -
5.1.1	<i>Προσέγγιση με βάση τον προγραμματιστικό κώδικα (code-based)</i>	- 87 -
5.1.2	<i>Προσέγγιση που οδηγείται από τη μοντελοποίηση της επιχειρηματικής διαδικασίας</i>	- 94 -
5.1.3	<i>Προσέγγιση προσανατολισμένη στις υπηρεσίες (Service-oriented Approach)</i>	- 101 -
5.1.3.1	Καθορισμός μηνυμάτων επικοινωνίας	- 101 -
5.1.3.2	Δημιουργία των επιχειρηματικών κανόνων και της Υπηρεσίας που ενσωματώνει τους επιχειρηματικούς κανόνες	- 102 -
5.1.3.3	Δημιουργία της επιχειρηματικής διαδικασίας BPEL	- 112 -
6	ΕΠΙΛΟΓΟΣ – ΣΥΜΠΕΡΑΣΜΑΤΑ	- 119 -
7	ΒΙΒΛΙΟΓΡΑΦΙΑ	- 122 -

Σχήματα

ΣΧΗΜΑ 5-1: ΕΙΔΗ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ	5 -
ΣΧΗΜΑ 5-2: ΚΥΚΛΟΣ ΖΩΗΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ	8 -
ΣΧΗΜΑ 5-3: ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ	10 -
ΣΧΗΜΑ 5-4: ΑΣΥΓΧΡΟΝΗ ΕΠΙΚΟΙΝΩΝΙΑ.....	14 -
ΣΧΗΜΑ 6-1: ΕΠΙΠΕΔΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΠΡΟΣΑΝΑΤΟΛΙΖΟΜΕΝΗΣ ΣΤΙΣ ΥΠΗΡΕΣΙΕΣ (SOA).....	17 -
ΣΧΗΜΑ 6-2: ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΥΠΗΡΕΣΙΩΝ	20 -
ΣΧΗΜΑ 6-3: ΕΙΔΗ ΥΠΗΡΕΣΙΩΝ.....	20 -
ΣΧΗΜΑ 6-4: FUNDAMENTAL SOA (ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΒΑΣΙΚΩΝ ΥΠΗΡΕΣΙΩΝ)	21 -
ΣΧΗΜΑ 6-5: FEDERATED SOA (ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΝΘΕΤΩΝ ΥΠΗΡΕΣΙΩΝ)	21 -
ΣΧΗΜΑ 6-6: PROCESS-ENABLED SOA (ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΗΡΕΣΙΩΝ ΔΙΑΔΙΚΑΣΙΩΝ).....	22 -
ΣΧΗΜΑ 6-7: ΚΥΚΛΟΣ ΖΩΗΣ ΥΠΗΡΕΣΙΩΝ.....	23 -
ΣΧΗΜΑ 6-8: ΣΗΜΕΙΟ ΠΡΟΣ ΣΗΜΕΙΟ ΣΥΝΔΕΣΗ (POINT-TO-POINT CONNECTION)	27 -
ΣΧΗΜΑ 6-9: ΔΙΑΜΕΣΟΛΑΒΗΣΗ (MEDIATION).....	28 -
ΣΧΗΜΑ 6-10: ΑΝΑΧΑΙΤΙΣΤΗΣ (INTERCEPTOR).....	28 -
ΣΧΗΜΑ 6-11: ΣΥΝΘΕΣΗ ΥΠΗΡΕΣΙΩΝ.....	39 -
ΣΧΗΜΑ 6-12: ΕΝΟΡΧΗΣΤΡΩΣΗ.....	41 -
ΣΧΗΜΑ 6-13: ΧΟΡΟΓΡΑΦΙΑ.....	41 -
ΣΧΗΜΑ 7-1: ΠΑΡΑΔΕΙΓΜΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΜΕ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥΣ ΚΑΝΟΝΕΣ.....	48 -
ΣΧΗΜΑ 7-2: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΓΙΑ ΤΗΝ ΕΝΣΩΜΑΤΩΣΗ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΝΟΝΩΝ ΣΤΗΝ SOA ΑΡΧΙΤΕΚΤΟΝΙΚΗ.....	55 -
ΣΧΗΜΑ 7-3: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΥΠΗΡΕΣΙΑΣ RULE INTERCEPTOR	56 -
ΣΧΗΜΑ 8-1: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΠΡΩΤΟΤΥΠΟΥ ΥΛΟΠΟΙΗΣΗΣ.....	59 -
ΣΧΗΜΑ 8-2: 4-ΤΙΕΡ ΑΡΧΙΤΕΚΤΟΝΙΚΗ	61 -
ΣΧΗΜΑ 8-3: 3-ΤΙΕΡ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΛΑΤΦΟΡΜΑΣ JAVA ΕΕ.....	62 -
ΣΧΗΜΑ 8-4: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ GLASSFISH ESB	67 -
ΣΧΗΜΑ 8-5: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ GLASSFISH ESB	67 -
ΣΧΗΜΑ 8-6: FILE BC ΩΣ PROVIDER	70 -
ΣΧΗΜΑ 8-7: FILE BC ΩΣ CONSUMER	70 -
ΣΧΗΜΑ 8-8: ΜΟΡΦΗ ΡΟΗΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ BPEL	71 -
ΣΧΗΜΑ 8-9: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΥΠΟΣΥΣΤΗΜΑΤΑ ΤΟΥ ILOG JRULES.....	76 -
ΣΧΗΜΑ 8-10: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΣΧΕΔΙΑΣΜΟΥ ΜΙΑΣ ΕΦΑΡΜΟΓΗΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΝΟΝΩΝ ΣΤΟ ILOG JRULES	78 -
ΣΧΗΜΑ 8-11: ΓΛΩΣΣΕΣ ΣΥΓΓΡΑΦΗΣ, ΕΠΕΞΕΡΓΑΣΙΑΣ ΚΑΙ ΕΚΤΕΛΕΣΗΣ ΚΑΝΟΝΩΝ ΣΤΟ ILOG JRULES.....	79 -
ΣΧΗΜΑ 8-12: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΕΚΤΕΛΕΣΗΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΝΟΝΩΝ ΜΕ ΒΑΣΗ ΤΟΝ ΑΛΓΟΡΙΘΜΟ RETEPLUS	80 -
ΣΧΗΜΑ 8-13: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΕΚΤΕΛΕΣΗΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΝΟΝΩΝ ΜΕ ΒΑΣΗ ΤΟΝ ΑΚΟΛΟΥΘΙΑΚΟ ΑΛΓΟΡΙΘΜΟ	82 -
ΣΧΗΜΑ 8-14: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΕΚΤΕΛΕΣΗΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΝΟΝΩΝ ΜΕ ΒΑΣΗ ΤΟΝ ΑΛΓΟΡΙΘΜΟ FASTPATH	82 -
ΣΧΗΜΑ 8-15: ΟΡΙΣΜΟΣ ΠΟΛΥΠΛΟΚΩΝ ΤΥΠΩΝ ΤΟΥ XML ΣΧΗΜΑΤΟΣ LOANREQUESTOR . XSD	88 -
ΣΧΗΜΑ 8-16: ΟΡΙΣΜΟΣ ΣΤΟΙΧΕΙΩΝ ΤΟΥ XML ΣΧΗΜΑΤΟΣ LOANREQUESTOR . XSD.....	88 -
ΣΧΗΜΑ 8-17: ΟΡΙΣΜΟΣ ΑΦΑΙΡΕΤΙΚΗΣ ΔΟΜΗΣ ΤΟΥ WSDL ΑΡΧΕΙΟΥ LOANREQUESTOR	89 -
ΣΧΗΜΑ 8-18: ASSIGN ΔΡΑΣΤΗΡΙΟΤΗΤΑ COPYCUSTOMERINFO	90 -
ΣΧΗΜΑ 8-19: ASSIGN ΔΡΑΣΤΗΡΙΟΤΗΤΑ COPYLOANSTATUS.....	90 -
ΣΧΗΜΑ 8-20: ΕΠΙΧΕΙΡΗΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ LOANREQUESTOR	91 -

ΣΧΗΜΑ 8-21: ΟΡΙΣΜΟΣ ΚΑΙ ΠΑΡΑΜΕΤΡΟΙ ΤΗΣ ΔΙΑΔΙΚΤΥΑΚΗΣ ΥΠΗΡΕΣΙΑΣ LOANPROCESSOR	- 92 -
ΣΧΗΜΑ 8-22: ΣΥΝΘΕΤΗ ΕΦΑΡΜΟΓΗ LOANREQUESTOR	- 94 -
ΣΧΗΜΑ 8-23: ΕΠΙΧΕΙΡΗΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ CREDITCHECKSERVICE	- 95 -
ΣΧΗΜΑ 8-24: WSDL ΔΟΜΗ ΠΕΛΑΤΗ CREDITCHECKDEFN	- 96 -
ΣΧΗΜΑ 8-25: ΔΟΜΗ ΣΗΜΕΙΟΥ ΑΠΟΦΑΣΕΩΝ.....	- 96 -
ΣΧΗΜΑ 8-26: ASSIGN ΔΡΑΣΤΗΡΙΟΤΗΤΑ ΣΕ ΠΕΡΙΠΤΩΣΗ ΘΕΤΙΚΟΥ ΑΠΟΤΕΛΕΣΜΑΤΟΣ	- 97 -
ΣΧΗΜΑ 8-27: ASSIGN ΔΡΑΣΤΗΡΙΟΤΗΤΑ ΣΕ ΠΕΡΙΠΤΩΣΗ ΑΡΝΗΤΙΚΟΥ ΑΠΟΤΕΛΕΣΜΑΤΟΣ.....	- 97 -
ΣΧΗΜΑ 8-28: ΕΠΙΧΕΙΡΗΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ HOMEAPPRAISALSERVICE	- 98 -
ΣΧΗΜΑ 8-29: ΕΠΙΧΕΙΡΗΜΑΤΙΚΗ ΛΟΓΙΚΗ ΣΗΜΕΙΟΥ ΑΠΟΦΑΣΕΩΝ.....	- 98 -
ΣΧΗΜΑ 8-30: ΕΠΙΧΕΙΡΗΜΑΤΙΚΗ ΛΟΓΙΚΗ ΣΗΜΕΙΟΥ ΑΠΟΦΑΣΕΩΝ ΚΥΡΙΑΣ ΔΙΑΔΙΚΑΣΙΑΣ.....	- 99 -
ΣΧΗΜΑ 8-31: ΚΥΡΙΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ LOANAPPROVALSERVICE.....	- 100 -
ΣΧΗΜΑ 8-32: ΚΥΡΙΑ ΣΥΝΘΕΤΗ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗ ΕΦΑΡΜΟΓΗ LOANAPPROVALSERVICE.....	- 100 -
ΣΧΗΜΑ 8-33: ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΝΟΝΩΝ.....	- 103 -
ΣΧΗΜΑ 8-34: ΔΗΜΙΟΥΡΓΙΑ ΠΑΡΑΜΕΤΡΩΝ ΣΥΝΟΛΟΥ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΝΟΝΩΝ	- 103 -
ΣΧΗΜΑ 8-35: ΔΗΜΙΟΥΡΓΙΑ ΜΕΤΑΒΛΗΤΩΝ ΣΥΝΟΛΟΥ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΝΟΝΩΝ.....	- 104 -
ΣΧΗΜΑ 8-36: ΔΗΜΙΟΥΡΓΙΑ ΡΟΗΣ ΑΠΟΤΙΜΗΣΗΣ ΚΑΙ ΕΚΤΕΛΕΣΗΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΝΟΝΩΝ.....	- 105 -
ΣΧΗΜΑ 8-37: ΔΗΜΙΟΥΡΓΙΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΚΑΝΟΝΑ CHECKAGE	- 106 -
ΣΧΗΜΑ 8-38: ΔΗΜΙΟΥΡΓΙΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΚΑΝΟΝΑ CREDITSCORE.....	- 107 -
ΣΧΗΜΑ 8-39: ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΑ ΑΠΟΦΑΣΕΩΝ SETASSETVALUE	- 107 -
ΣΧΗΜΑ 8-40: ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΑ ΑΠΟΦΑΣΕΩΝ SETINCOMEVALUE	- 108 -
ΣΧΗΜΑ 8-41: ΣΥΣΧΕΤΙΣΗ ΚΑΙ ΑΝΤΙΣΤΟΙΧΙΣΗ ΡΟΗΣ ΕΚΤΕΛΕΣΗΣ ΚΑΙ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΝΟΝΩΝ	- 109 -
ΣΧΗΜΑ 8-42: ΔΗΜΙΟΥΡΓΙΑ ΕΚΤΕΛΕΣΙΜΗΣ ΕΦΑΡΜΟΓΗΣ	- 110 -
ΣΧΗΜΑ 8-43: ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΣΤΟΝ ΕΞΥΠΗΡΕΤΗΤΗ.....	- 110 -
ΣΧΗΜΑ 8-44: ΕΙΣΟΔΟΣ ΣΤΟΝ ΕΞΥΠΗΡΕΤΗΤΗ RULE EXECUTION SERVER	- 111 -
ΣΧΗΜΑ 8-45: ΈΛΕΓΧΟΣ ΟΡΘΟΤΗΤΑΣ ΣΥΝΟΛΟΥ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΝΟΝΩΝ ΣΤΟΝ ΕΞΥΠΗΡΕΤΗΤΗ RULE EXECUTION SERVER.....	- 112 -
ΣΧΗΜΑ 8-46: ΔΗΜΙΟΥΡΓΙΑ WSDL ΑΡΧΕΙΟΥ CREDITREQUEST ΤΟΥ ΠΕΛΑΤΗ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ.....	- 113 -
ΣΧΗΜΑ 8-47: ΔΗΜΙΟΥΡΓΙΑ WSDL ΑΡΧΕΙΟΥ CREDITREQUEST ΤΟΥ ΠΕΛΑΤΗ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ.....	- 114 -
ΣΧΗΜΑ 8-48: ΔΗΜΙΟΥΡΓΙΑ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ ΚΛΗΣΗΣ ΤΟΥ WSDL ΑΡΧΕΙΟΥ ΕΝΣΩΜΑΤΩΣΗΣ ΤΩΝ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΝΟΝΩΝ	- 115 -
ΣΧΗΜΑ 8-49: ΔΗΜΙΟΥΡΓΙΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ CREDITREQUESTPROCESS	- 115 -
ΣΧΗΜΑ 8-50: ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΘΕΤΗΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗΣ ΕΦΑΡΜΟΓΗΣ CREDITREQUESTBP	- 116 -
ΣΧΗΜΑ 8-51: ΚΑΤΑΓΡΑΦΗ ΚΑΙ ΈΛΕΓΧΟΣ ΟΡΘΗΣ ΕΚΤΕΛΕΣΗΣ ΤΗΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ	- 117 -
ΣΧΗΜΑ 8-52: ΓΡΑΦΙΚΗ ΔΙΕΠΑΦΗ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ.....	- 118 -

Πίνακες

ΠΙΝΑΚΑΣ 5-1: ΣΥΓΚΡΙΣΗ ΙΣΧΥΡΗΣ – ΧΑΛΑΡΗΣ ΣΥΖΕΥΞΗΣ.....	- 13 -
ΠΙΝΑΚΑΣ 7-1: ΣΥΓΚΡΙΣΗ ΕΜΠΟΡΙΚΩΝ ΜΗΧΑΝΩΝ ΠΑΡΑΓΩΓΗΣ.....	- 52 -
ΠΙΝΑΚΑΣ 9-1: ΣΥΓΚΡΙΣΗ ΠΡΟΣΕΓΓΙΣΕΩΝ ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΚΑΝΟΝΩΝ	- 121 -

Αρκτηκόλεξο Ξενόγλωσσων Όρων

API:	Application Programming Interface
BAL:	Business Action Language
BAM:	Business Activity Monitoring
BC:	Binding Components
BOM:	Business Object Model
BPEL:	Business Process Execution Language
BPEL4WS:	Business Process Execution Language for Web Services
BPM:	Business Process Management
B2B:	Business to Business
B2C:	Business to Costumer
B2G:	Business to Government
CORBA:	Common Object Request Broker Architecture
DCOM:	Distributed Component Object Model
DOM:	Document Object Model
DTD:	Document Type Definitions
EAI:	Enterprise Application Integration
cXML:	Commerce XML
ebXML:	Electronic Business using eXtensible Markup Language
ERP:	Enterprise Resource Planning
ESB:	Enterprise Service Bus
FTP:	File Transfer Protocol
HTML:	HyperText Markup Language
HTTP:	Hypertext Transfer Protocol
IRL:	ILOG Rule Language
Java EE:	Java Platform, Enterprise Edition
JB1:	Java Business Integration
MIME:	Multipurpose Internet Mail Extensions
MOM:	Message Oriented Architecture
ORB:	Object Request Brokers
OMG:	Object Management Group
REST:	REpresentational State Transfer
RPC:	Remote Procedure Calls
SAX:	Simple API for XML
SE:	Service Engine
SMTP:	Simple Mail Transfer Protocol
SOA:	Service Oriented Architecture
SOAP:	Simple Object Access Protocol
UML:	Unified Modeling Language
WS-BPEL:	Web Service Business Process Execution Language
WS-CDL:	Web Service Choreography Description Language
WSCI:	Web Service Choreography
WSCL:	Web Services Conversation Language
WSDL:	Web Services Description Language
WSFL:	Web Services Flow Language
XLANG:	XML-based extension of Web Services Description Language
XML:	Extensible Markup Language

XOM:	Execution Object Model
XPATH:	XML Path Language
XSD:	XML Schema Definitions
XSLT:	Extensible Stylesheet Language Transformations

Πρόλογος

Οι τεχνολογικές εξελίξεις και ο πολυδιάστατος και ευμετάβλητος χαρακτήρας των επιχειρηματικών φαινομένων σήμερα τοποθετούν τις επιχειρηματικές σχέσεις και εφαρμογές σε ένα διαφορετικό πλαίσιο που επικεντρώνει στην ανάγκη για αυτοματοποίηση και ευελιξία των επιχειρηματικών διαδικασιών ως στρατηγικό πλεονέκτημα και μέσο βιωσιμότητας. Η σύγχρονη επιχείρηση, στα πλαίσια της πολύπλοκης και ανταγωνιστικής δομής της παγκόσμιας αγοράς, καλείται να ακολουθήσει τον ραγδαίο ρυθμό επιχειρηματικών μεταβολών και να προσαρμόζει συνεχώς τα δεδομένα της. Ο ρόλος των Πληροφοριακών Συστημάτων ως αναπόσπαστο τμήμα της σύγχρονης επιχείρησης θεωρείται καίριος προς τη συγκεκριμένη κατεύθυνση, καθώς το σύνολο των επιχειρηματικών δραστηριοτήτων και εφαρμογών υποκινούνται και καθορίζονται με βάση τις προδιαγραφές της υλικής προγραμματιστικής υποδομής της επιχείρησης. Τα σύγχρονα ζητούμενα του παγκοσμιοποιημένου επιχειρηματικού περιβάλλοντος υπαγορεύουν την ανάγκη για ευελιξία, προσαρμοστικότητα στις μεταβολές των επιχειρηματικών μοντέλων, έγκαιρη αντίδραση σε αυτές, επαναχρησιμοποίηση των μονάδων λειτουργικότητας και αποδέσμευση της επιχειρηματικής λογικής από την τεχνική υποδομή. Συγχρόνως, εμφανίζεται σε όλο και μεγαλύτερο βαθμό η ανάγκη για πλήρη διαλειτουργικότητα και ενοποίηση των καταναμημένων συστημάτων ετερογενούς χαρακτήρα και υλοποίησης προς την κατεύθυνση της επαναχρησιμοποίησης και επέκτασης των διαδικασιών.

Επομένως, υφίσταται ανάγκη για ενσωμάτωση των επιχειρηματικών κανόνων της επιχειρησιακής λογικής αλλά πλέον και της επιχειρηματικής πολιτικής στην αρχιτεκτονική υπολογιστική υποδομή της σύγχρονης επιχείρησης ώστε να λειτουργεί ως ανεξάρτητο τμήμα της επιχειρηματικής λογικής παρέχοντας ευελιξία και προσαρμοστικότητα. Στην κατεύθυνση αυτή, η αφομοίωση και αποδοχή των βασικών χαρακτηριστικών της Προσανατολισμένης στις Υπηρεσίες Αρχιτεκτονικής (Service Oriented Architecture) θεωρείται απαραίτητη για την εκμετάλλευση των πλεονεκτημάτων που παρέχει. Στα πλαίσια, συνεπώς, της επέκτασης και προσαρμογής των σύγχρονων αρχιτεκτονικών προσεγγίσεων στην ανάγκη για ενσωμάτωση της επιχειρησιακής πολιτικής και την επίτευξη ευέλικτων ροών εργασίας, γίνεται προσπάθεια στην παρούσα διπλωματική να προσεγγιστεί η αρχιτεκτονική μορφή τεσσάρων βαθμίδων (4-tier) και να υλοποιηθούν πρότυπες ροές εργασίες δυναμικού χαρακτήρα που αναπροσαρμόζουν τη συμπεριφορά τους και οδηγούν στην ενορχήστρωση διαδικτυακών Υπηρεσιών και σύνθετων επιχειρηματικών Διαδικασιών.

Περίληψη

Η συγκεκριμένη διπλωματική κινείται στα πλαίσια της εφαρμογής των Πληροφοριακών Συστημάτων στη σύγχρονη επιχείρηση και ειδικότερα στην οργάνωση και σύνθεση δυναμικών ροών εργασίας κατά την εκτέλεση των επιχειρηματικών συναλλαγών και διαδικασιών. Αποσκοπεί στη ανάπτυξη μιας πρωτότυπης εφαρμογής που να εμφανίζει τη δυνατότητα δημιουργίας, διαχείρισης και σύνθεσης επιχειρηματικών διαδικασιών με δυναμική συμπεριφορά που μπορούν συνδυαστούν για τη δόμηση μιας επιχειρηματικής εφαρμογής υψηλού επιπέδου. Συγχρόνως, επιχειρείται μια προσπάθεια ενσωμάτωσης της επιχειρησιακής πολιτικής με τη μορφή επιχειρηματικών κανόνων στο επίπεδο της επιχειρηματικής λογικής, με αποτέλεσμα την προδιαγραφή μιας αρχιτεκτονικής προσέγγισης τεσσάρων βαθμίδων.

Αρχικά σκιαγραφούνται οι έννοιες των επιχειρηματικών διαδικασιών και των ηλεκτρονικών συναλλαγών στο σύγχρονο επιχειρηματικό περιβάλλον και παρουσιάζονται τα βασικά χαρακτηριστικά τους. Γίνεται ιδιαίτερη αναφορά στα στοιχεία που αποτελούν τα σύγχρονα ζητούμενα στην αρχιτεκτονική και σχεδιαστική προσέγγιση των Πληροφοριακών Συστημάτων και ιδιαίτερα στη χρήση ενδιάμεσου λογισμικού που διευκολύνει/υποστηρίζει δυναμικές ροές εργασίας και τα οποία υπαγορεύει η σύγχρονη παγκόσμια αγορά. Προς τη συγκεκριμένη κατεύθυνση γίνεται λεπτομερής αναφορά στο προσανατολισμένο στις Υπηρεσίες αρχιτεκτονικό πλαίσιο (Service-Oriented Architecture) με βάση το οποίο δομείται υπολογιστικά το οικοδόμημα των σύγχρονων καταναμημένων και ετερογενών συστημάτων. Παράλληλα εξετάζονται οι επιχειρηματικοί κανόνες ως μέσο ορισμού της επιχειρησιακής πολιτικής και η χρήση των μηχανών παραγωγής για τη βέλτιστη παραγωγή κανόνων. Τέλος προδιαγράφεται και υλοποιείται ένα πρωτότυπο μίας αρχιτεκτονικής προσέγγισης που στηρίζεται στην εξωτερίκευση των επιχειρηματικών κανόνων με τη μορφή Διαδικτυακών Υπηρεσιών και την ενσωμάτωση τους σε καταναμημένα περιβάλλοντα διαφορετικών πληροφοριακών συστημάτων.

Λέξεις - Κλειδιά

Επιχειρηματικές Διαδικασίες, Ηλεκτρονικές Συναλλαγές, Ετερογένεια, Καταναμημένα Συστήματα, Διαδικτυακές Υπηρεσίες, Αρχιτεκτονική Προσανατολισμένη στις Υπηρεσίες, Επιχειρηματικός Δίαυλος Υπηρεσιών, Ενορχήστρωση, Διαλειτουργικότητα, Ενοποίηση, Επιχειρηματικοί Κανόνες, Μηχανή Παραγωγής Κανόνων, Ευελιξία, Επαναχρησιμοποίηση, Αρχιτεκτονική Τεσσάρων Επιπέδων

Abstract

The present dissertation is based on the application of Information Systems within modern enterprises and in particular it focuses on the organization and composition of dynamic workflows at the execution of business transactions and processes. The dissertation aims at the development of a prototype application that enables development, management and deployment of composite business processes of dynamic nature that have the ability to be combined in order to structure high-level enterprise application. In addition, an effort has been made so as to incorporate the business policy in the business rules and as a result to develop an 4-tier architectural model.

At first the dissertation represents the concepts of business processes and electronic transactions in the current business environment and analyzes their basic characteristics. Specific attention is given to the elements that are necessary for the architectural and design approach of the infrastructure and particularly to the use of intermediate software that facilitates and supports dynamic workflows and is necessary as a result of the current international market. Moreover, the dissertation examines the widely used Service-Oriented architectural paradigm that is the basis on which the computational structure and landscape of distributed and heterogeneous systems are designed. Furthermore, the current dissertation presents the contribution of business rules to the representation and definition of the business policy and the use of rule engines for the optimum definition and deployment of business rules. At last, a software prototype with an architectural approach based on business rules, is presented and implemented with the contribution of major technologies, such as Web Services. The idea of this paradigm involves the exposition of business rules as web services that are applicable to any major enterprise service bus implementation and distributed landscape.

Λέξεις - Κλειδιά

Business Processes, e-Transactions, Heterogeneity, Distributed Systems, Web Services, Service-Oriented Architecture (SOA), Enterprise Service Bus (ESB), Orchestration, Interoperability, Integration, Business Rules, Rule Engine, Flexibility, Reusability, 4-Tier Architecture

1 Εισαγωγή

1.1 Ορισμός του Προβλήματος

Οι τεχνολογικές εξελίξεις και ο πολυδιάστατος και ευμετάβλητος χαρακτήρας των επιχειρηματικών φαινομένων έχουν οδηγήσει στην ανάγκη εκσυγχρονισμού και βελτίωσης των επιχειρηματικών εφαρμογών και σχέσεων. Οι ηλεκτρονικές συναλλαγές διαδραματίζουν ολοένα και σημαντικότερο ρόλο για την επιβίωση και την απόκτηση στρατηγικού πλεονεκτήματος μιας επιχείρησης έναντι των ανταγωνιστών της, καθώς η ορθή δόμηση και αρχιτεκτονική τους προσέγγιση παρέχει το σύγχρονο και απόλυτο ζητούμενο της ευελιξίας και της επέκτασης των επιχειρηματικών διαδικασιών και εφαρμογών. Το πεδίο της τεχνολογίας της Πληροφορίας και των Υπολογιστικών Συστημάτων φαντάζει λόγω της εγγενούς δυναμικότητας του ως ο βασικός φορέας πειραματισμού και ανάπτυξης. Στον αυριανό επιχειρηματικό κόσμο, μόνο οι επιχειρήσεις που θα συμπεριλάβουν στην τεχνογνωσία και το ερευνητικό τους πεδίο την έννοια της δυναμικότητας και της ευελιξίας των επιχειρηματικών διαδικασιών, θα κυριαρχήσουν και θα υιοθετήσουν την έννοια της μεταβολής ως αυτοσκοπό ανάπτυξης και στρατηγικού σχεδιασμού.

Γίνεται λοιπόν φανερό ότι η νέα πρόκληση που έχει να αντιμετωπίσει μια σύγχρονη επιχείρηση είναι η εφαρμογή επιχειρηματικών κανόνων και διαδικασιών που να έχουν τη δυνατότητα να μεταβάλλονται δυναμικά και να προσδιορίζουν το περιεχόμενο τους με βάση τις μεταβολές στην πολιτική της επιχείρησης και τις ευμετάβλητες απαιτήσεις του σύγχρονου επιχειρηματικού κόσμου. Στα πλαίσια λοιπόν της συγκεκριμένης διπλωματικής γίνεται προσπάθεια να προσεγγιστούν όσο το δυνατόν πληρέστερα οι διάφορες μέθοδοι εφαρμογής δυναμικών ροών εργασίας βασιζόμενοι πάνω σε σύγχρονες τεχνολογίες που αναπτύχθηκαν για να διευκολύνουν το δυναμικό χαρακτήρα της παγκόσμιας αγοράς.

Παράλληλα, μέσα από την υιοθέτηση ενός συγκεκριμένου παραδείγματος επιχειρηματικής διαδικασίας, θα γίνει προσπάθεια σύγκρισης των διαφορετικών μεθόδων και τεχνικών προσέγγισης της επιχειρηματικής λογικής και θα αναδειχθεί η αξία της υιοθέτησης επιχειρηματικών κανόνων, ως οδηγό κατά την εκτέλεση μιας επιχειρηματικής διαδικασίας, σε φυσική γλώσσα και προσιτούς στους επιχειρηματικούς αναλυτές και στους σχεδιαστές της εταιρικής στρατηγικής και πολιτικής. Προτού όμως, καταλήξουμε στην ιδανικότερη επιλογή της υπολογιστικής αρχιτεκτονικής και υλοποίησης της, γίνεται μια περιγραφή των βασικών τεχνολογικών επιτευγμάτων και αρχιτεκτονικών προσεγγίσεων που έχουν συμβάλει και υιοθετηθεί προς τη συγκεκριμένη κατεύθυνση.

1.2 Ο Ρόλος των Πληροφοριακών Συστημάτων στη Σύγχρονη Επιχείρηση

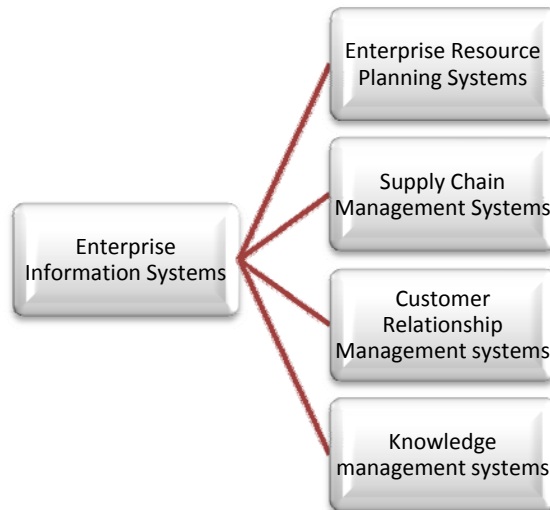
Τα πληροφοριακά συστήματα αποτελούν αναπόσπαστο παράγοντα της σύγχρονης επιχείρησης σε τέτοιο βαθμό ώστε η βιωσιμότητα και η στρατηγική ύπαρξη στην αγορά να εξαρτάται άμεσα από το επίπεδο ανάλυσης και υλοποίησης των επιχειρηματικών συναλλαγών που υποστηρίζονται από τα πληροφοριακά συστήματα της επιχείρησης. Συγκεκριμένα, θεωρείται απαραίτητο για τη σύγχρονη επιχείρηση -

προκειμένου να μπορεί να αποκτήσει στρατηγικό πλεονέκτημα έναντι των ανταγωνιστών της - να επενδύσει στην οργάνωση των πληροφοριακών της συστημάτων και στο σχεδιασμό των βέλτιστων δυνατών επιχειρηματικών διαδικασιών, γεγονός που θα συμβάλει στη διευκόλυνση και την ευελιξία των ροών εργασίας και τη δυναμική μεταβολή του στρατηγικού σχεδιασμού και της πολιτικής της.

Αφετηρία των διαδικασιών εκσυγχρονισμού των επιχειρηματικών εφαρμογών και της μεταβολής στην αρχιτεκτονική αντίληψη αποτέλεσαν οι τεχνολογικές εξελίξεις των τελευταίων δεκαετιών στον τομέα της Πληροφορικής και των Επικοινωνιών. Ιδιαίτερα κάτω από την εκρηκτική εξάπλωση του Διαδικτύου, την αποδοχή της ψηφιακής τεχνολογίας ως αναπόσπαστο τμήμα της επιχειρηματικής πραγματικότητας και την εμφάνιση του φαινομένου της παγκόσμιας αγοράς και του εμπορίου οι επιχειρήσεις προέβησαν σε κινήσεις αναδόμησης και ανανέωσης των τεχνικών υποδομών τους. Παράλληλα επικεντρώθηκαν σε μεγάλο βαθμό στην καλύτερη εκμετάλλευση των πόρων τους και ειδικότερα των δεδομένων και της τεχνογνωσίας τους με απώτερο στόχο να επιτύχουν τη μεγαλύτερη επαναχρησιμοποίηση τους, να διευκολύνουν τις επιχειρηματικές τους διαδικασίες και να μειώσουν το λειτουργικό κόστος τους.

Ορίζοντας σε γενικές γραμμές ένα Πληροφοριακό Σύστημα μπορούμε να ισχυριστούμε ότι αποτελεί ένα σύστημα από συλλογές δεδομένων και δραστηριοτήτων που επεξεργάζονται την παρεχόμενη πληροφορία και προωθούν την αυτοματοποίηση των διαδικασιών [21]. Στόχος τους αποτελεί η συλλογή, η επεξεργασία, η αποθήκευση και τη διανομή πληροφοριών οι οποίες είναι απαραίτητες για την λήψη αποφάσεων και τον έλεγχο μέσα στα πλαίσια ενός οργανισμού. Από επιχειρηματική σκοπιά, τα πληροφοριακά συστήματα αποτελούν σημαντικό χαρτοφυλάκιο πόρων και επενδύσεων των οργανισμών που έχουν ως απώτερο στόχο τη δημιουργία αξίας για την επιχείρηση και την ανάπτυξη στρατηγικού πλεονεκτήματος.

Στη σύγχρονη επιχειρηματική πραγματικότητα, το ενδιαφέρον των επιχειρήσεων έχει μεταβληθεί και έχει απομακρυνθεί από τον προσανατολισμό στο παραγωγικό αποτέλεσμα, με συνέπεια να επικεντρώνεται και να στρέφεται σε έναν προσανατολισμό με βάση τη γνώση (knowledge-based) με την έννοια ότι οι σύγχρονες επιχειρήσεις ανταγωνίζονται όχι ως προς την παραγωγή και το προϊόν αλλά ως προς την ευελιξία των επιχειρηματικών διαδικασιών και την καινοτομία. Η έμφαση έχει μεταφερθεί από την ποσότητα και την ποιότητα της παραγωγής στην ίδια την παραγωγική διαδικασία και τις υπηρεσίες και επιχειρηματικές λειτουργίες που τη συνοδεύουν. Γίνεται, λοιπόν φανερό ότι οι σύγχρονοι οργανισμοί δίνουν ιδιαίτερη βαρύτητα στο σχεδιασμό και την ανάπτυξη των επιχειρηματικών δραστηριοτήτων μέσα από ένα πλαίσιο ηλεκτρονικής διαχείρισης. Προς τη συγκεκριμένη κατεύθυνση έχουν αναπτυχθεί για την οργάνωση και διαχείριση των διεργασιών η τεχνική της διαχείρισης των επιχειρηματικών διεργασιών (business process management) και τεχνολογίες ανάπτυξης επιχειρηματικών κανόνων. Ως αποτέλεσμα μέσα από τον ορθό σχεδιασμό και συντονισμό επιχειρηματικών κανόνων και διαδικασιών δημιουργείται στρατηγικό πλεονέκτημα και μειώνεται το λειτουργικό κόστος και το κόστος που επιφέρουν οι συνεχείς μεταβολές στην εταιρική πολιτική λόγω της επαναχρησιμοποίησης των διαδικασιών.



Σχήμα 5-1: Είδη Πληροφοριακών Συστημάτων

Τα πληροφοριακά συστήματα των επιχειρήσεων είναι επιφορτισμένα με την διεκπεραίωση των επιχειρηματικών διεργασιών και την εκτέλεση των ηλεκτρονικών συναλλαγών, ενώ άπτονται θεμάτων και διαδικασιών τόσο εσωτερικών όσο και σε επίπεδο συνεργασίας με άλλους οργανισμούς. Μέσα στην επιχείρηση διακρίνονται τέσσερα βασικά είδη Πληροφοριακών Συστημάτων καθένα από τα οποία αναλαμβάνουν ένα ξεχωριστό σύνολο επιχειρηματικών διεργασιών και συμβάλλουν στην ολοκλήρωση αυτών σε κάθε διακριτό τομέα. Συγκεκριμένα αναγνωρίζονται τα παρακάτω είδη πληροφοριακών συστημάτων:

- ✓ Enterprise Resource Planning systems (ERP), (Συστήματα Διαχείρισης Επιχειρησιακών Πόρων)
- ✓ Supply Chain Management systems (SCM), (Συστήματα Διαχείρισης της Εφοδιαστικής Αλυσίδας)
- ✓ Customer Relationship Management systems (CRM), (Συστήματα Διαχείρισης Σχέσεων με τους Πελάτες)
- ✓ Knowledge management systems (KMS), (Συστήματα Διαχείρισης Επιχειρησιακής Γνώσης)

Τα συστήματα διαχείρισης πόρων (Enterprise Resource Planning) αποτελούν ένα ενιαίο πληροφοριακό σύστημα που συντονίζει και ολοκληρώνει τις επιχειρηματικές διεργασίες του οργανισμού σχεδιάζοντας τον τρόπο διαχείρισης των διαθέσιμων πόρων. Τα πληροφοριακά συστήματα διαχείρισης της εφοδιαστικής αλυσίδας και σχέσεων με τους πελάτες συντελούν στον συντονισμό των διεργασιών της επιχείρησης που έχουν να κάνουν με τις σχέσεις της επιχείρησης με τους προμηθευτές και τους πελάτες αντίστοιχα, ενώ τα συστήματα επιχειρησιακής γνώσης υποστηρίζουν διαδικασίες που έχουν σχέση με τη συγκέντρωση και εφαρμογή επιχειρησιακής γνώσης και εμπειρικής δραστηριότητας.

Συγχρόνως, ως αποτέλεσμα της εξάπλωσης των διαδικτυακών εφαρμογών, κατέστη απαραίτητη η μετατροπή της επιχείρησης σε ένα ψηφιακό χώρο προσφοράς υπηρεσιών και αλληλεπίδρασης με τους συνεργαζόμενους φορείς, ώστε να εκμεταλλευτεί τις δυνατότητες αποκεντροποίησης και μεταφοράς επιχειρηματικών

διαδικασιών από την ίδια την επιχείρηση και το δυναμικό της στον ψηφιακή πραγματικότητα. Οι σύγχρονες ψηφιακές δυνατότητες και δραστηριότητες που χρησιμοποιούνται άπτονται της προσέγγισης του *ηλεκτρονικού επιχειρείν* (*electronic business, e-business*). Το ηλεκτρονικό επιχειρείν περιλαμβάνει δραστηριότητες του ηλεκτρονικού εμπορίου που αντιστοιχούν στις διαδικασίες ηλεκτρονικών συναλλαγών και επικοινωνίας, αλλά και στις δραστηριότητες που λαμβάνουν χώρα μέσα στα πλαίσια της επιχείρησης και αφορούν στην εσωτερική διοίκηση και τον συντονισμό των σχέσεών της με τους επιχειρηματικούς συνεργάτες της.

1.3 Ο Ρόλος των Επιχειρηματικών Διαδικασιών στη Σύγχρονη Επιχείρηση

Τα πληροφοριακά συστήματα και οι επιχειρησιακές εφαρμογές λογισμικού έχουν εξελιχθεί σε καθοριστικό παράγοντα με στρατηγική σημασία για τις επιχειρήσεις. Οι επιχειρήσεις στηρίζονται σε εφαρμογές λογισμικού που στόχο έχουν να προσομοιώσουν τον τρόπο με τον οποίο λειτουργούν προκειμένου να υλοποιήσουν και να φέρουν σε πέρας τις επιχειρηματικές απαιτήσεις και λειτουργίες. Οι συγκεκριμένοι στόχοι επιτυγχάνονται μέσα από την αυτοματοποίηση των επιχειρηματικών διαδικασιών με ζητούμενο τη βελτίωση και την αποδοτικότητα των λειτουργιών της επιχείρησης και την παροχή επαρκούς και περιεκτικής υποστήριξης των επιχειρηματικών διαδικασιών. Με άλλα λόγια, οι εφαρμογές που υλοποιούν τα πληροφοριακά συστήματα βρίσκονται σε άμεση συνάρτηση με τις επιχειρηματικές διαδικασίες. Στην πραγματικότητα, κατευθύνονται από αυτές.

Οι επιχειρηματικές διαδικασίες αναφέρονται *στον τρόπο με τον οποίο οργανώνεται, συντονίζεται και εστιάζεται η εργασία με στόχο τη παραγωγή ενός πολύτιμου προϊόντος ή υπηρεσίας* [28]. Όσο και αν η απαίτηση για εξάρτηση των επιχειρηματικών εφαρμογών λογισμικού από τις επιχειρηματικές διαδικασίες φαίνεται απλή στην υλοποίηση της, η σύγχρονη πολύπλοκη και ευμετάβλητη πραγματικότητα των καταστάσεων που προκύπτουν σε μια επιχείρηση και η ανάγκη για συνεχή μεταβολή της πολιτικής των επιχειρήσεων ώστε να ακολουθούν το επιχειρηματικό σχέδιο και τις συνεχώς μεταβαλλόμενες τάσεις της αγοράς και της οικονομίας, καθιστούν το σχεδιασμό των επιχειρηματικών διαδικασιών βαρύνουσα σημασίας και ιδιαίτερης προσοχής. Οι επιχειρηματικές διαδικασίες εμφανίζουν στις περισσότερες περιπτώσεις δυναμικό χαρακτήρα. Οι επιχειρήσεις είναι υποχρεωμένες να βελτιώνονται και να διαφοροποιούν συνεχώς το ρόλο τους, να δρουν με ευέλικτο τρόπο, να προσπαθούν να βελτιστοποιούν και να προσαρμόζουν τις επιχειρηματικές διαδικασίες στις ανάγκες της αγοράς και με αυτόν τον τρόπο να βελτιώνουν την απόκριση τους σε σημαντικά γεγονότα [11]. Κάθε μεταβολή ή βελτίωση των επιχειρηματικών διαδικασιών πρέπει να αντικατοπτρίζεται στις εφαρμογές και να τους παρέχει υποστήριξη. Μόνο οι επιχειρήσεις στις οποίες οι εφαρμογές μπορούν να προσαρμοστούν γρήγορα και αποτελεσματικά στις μεταβαλλόμενες επιχειρηματικές ανάγκες, είναι δυνατό να παραμείνουν ανταγωνιστικές στην παγκόσμια αγορά.

Είναι γνωστό ότι η μεταβολή και η διαφοροποίηση εφαρμογών λογισμικού είναι απαιτητική εργασία με μεγάλο χρονικό ορίζοντα. Αυτό σημαίνει ότι τα πληροφοριακά συστήματα δε μπορούν να αντιδράσουν άμεσα στη μεταβολή των επιχειρηματικών διαδικασιών – αντί αυτού υπάρχει η απαίτηση για χρόνο υλοποίησης, ελέγχου και εκτέλεσης των μεταβολών. Ο απαιτούμενος χρόνος

αναφέρεται συνήθως ως *information systems gap* και πρέπει συνήθως να είναι όσο μικρότερος γίνεται. Παρόλα αυτά στην πραγματικότητα υπάρχουν και άλλοι παράγοντες που πρέπει να ληφθούν υπόψη.

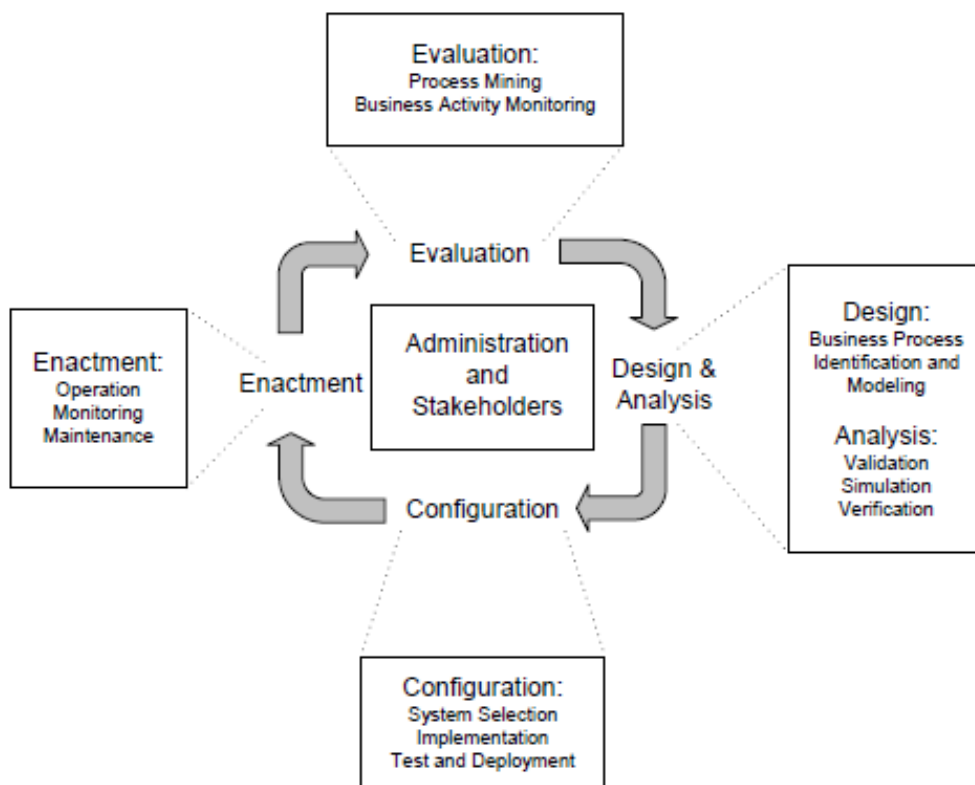
Ο χρόνος που απαιτείται για τη μεταβολή των επιχειρηματικών εφαρμογών εξαρτάται από μια σειρά παραγόντων με κυριότερο από τους οποίους, εκτός από την πολυπλοκότητα και το μέγεθος της μεταβολής, να είναι η κατάσταση της μεταβαλλόμενης εφαρμογής. Στην περίπτωση που η εφαρμογή έχει καλά καθορισμένη αρχιτεκτονική που έχει προνοήσει και παρέχει τη δυνατότητα μελλοντικών μεταβολών, τότε το πρόβλημα περιορίζεται στο ελάχιστο. Και πάλι, κάθε μεταβολή στην εφαρμογή δημιουργεί προβλήματα ευελιξίας στην ήδη υπάρχουσα αρχιτεκτονική, με αποτέλεσμα εφαρμογές που έχουν υποστεί μακροχρόνιες μεταβολές και συντηρούνται για χρόνια να αντιμετωπίζουν μη αναμενόμενες ασυνέπειες. Η κατάσταση γίνεται χειρότερη λαμβάνοντας υπόψη ότι οι περισσότερες επιχειρηματικές διαδικασίες έχουν αναπτυχθεί σε αναχρονιστικά σχεδιαστικά μοντέλα που δεν παρείχαν ούτε λάμβαναν υπόψη το ζητούμενο της υποστήριξης σε ένα διαλειτουργικό και ενοποιημένο περιβάλλον. Συνεπώς, γίνεται εμφανής η ανάγκη για ευελιξία, ολοκλήρωση και συντονισμός των επιχειρηματικών διαδικασιών.

Επομένως, για την επίτευξη αποτελεσματικών ροών αυτοματοποιημένων επιχειρηματικών διαδικασιών υπάρχει ανάγκη να :

- ✓ Παρέχεται ένας πρότυπος τρόπος για την εξωτερίκευση και την πρόσβαση της επιχειρηματικής λειτουργικότητας των εφαρμογών ως Υπηρεσίες.
- ✓ Παρέχεται ένα είδος υποδομής Διαύλου Υπηρεσιών για την επικοινωνία και τη διαχείριση των Υπηρεσιών που να περιλαμβάνει την ανταλλαγή, τη δρομολόγηση μηνυμάτων.
- ✓ Παρέχεται αρχιτεκτονική ολοκλήρωσης και ενοποίησης μεταξύ νέων και παλαιότερων συστημάτων.
- ✓ Παρέχεται μια εξειδικευμένη προγραμματιστική γλώσσα για τη σύνθεση πολυπλοκότερων επιχειρηματικών διαδικασιών.

1.3.1 Ο Κύκλος ζωής των Επιχειρηματικών Διαδικασιών

Ο κύκλος ζωής των επιχειρηματικών διαδικασιών συντίθεται από τις φάσεις του σχεδιασμού και της ανάλυσης (*design and analysis*), της διαμόρφωσης (*configuration*), της εκτέλεσης και παρακολούθησης (*enactment*) και τέλος της αποτίμησης (*evaluation*) [38].



Σχήμα 5-2: Κύκλος ζωής Επιχειρηματικών Διαδικασιών

1.3.1.1 Σχεδιασμός και Ανάλυση (Design and Analysis)

Ο κύκλος ζωής των επιχειρηματικών διαδικασιών αρχίζει από τη φάση του σχεδιασμού και της ανάλυσης όπου διεξάγονται έρευνες και μελέτες σχετικά με το επιχειρηματικό – οργανωσιακό και τεχνικό υπόβαθρο και περιβάλλον των επιχειρηματικών διαδικασιών. Με βάση τα σχετικά αποτελέσματα των ερευνών και ένα σύνολο επιχειρησιακών πολιτικών και λειτουργιών που αναπτύσσονται και καθοδηγούνται από τις αποφάσεις της διοίκησης, οι επιχειρηματικές διαδικασίες αναγνωρίζονται, αναθεωρούνται, ελέγχονται ως προς την αξιοπιστία τους και αναπαρίστανται από μοντέλα επιχειρηματικών διαδικασιών.

Τα μοντέλα επιχειρηματικών διαδικασιών εκφράζονται με σαφή τρόπο μέσω γραφικής σημειογραφίας που διευκολύνει την επικοινωνία και την αλληλοσυσχέτιση μεταξύ των διαφορετικών δραστηριοτήτων. Τεχνικές μοντελοποίησης των επιχειρηματικών διαδικασιών σε συνδυασμό με μεθόδους επικύρωσης, προσομοίωσης και επαλήθευσης χρησιμοποιούνται σε αυτό το στάδιο.

1.3.1.2 Διαμόρφωση (Configuration)

Από τη στιγμή που οι επιχειρηματικές διαδικασίες σχεδιάζονται και επαληθεύονται, δημιουργείται η ανάγκη για την υλοποίηση και πραγματοποίηση τους με βάση ένα σύνολο επιχειρησιακών εφαρμογών λογισμικού και υπολογιστικών συστημάτων. Επιλέγονται συνεπώς τα πληροφοριακά συστήματα που πρόκειται να συμμετάσχουν,

οι λειτουργίες που θα επιτελεί το καθένα και η αρχιτεκτονική ανάπτυξης και επικοινωνίας μεταξύ τους.

1.3.1.3 Εκτέλεση (Enactment)

Η τρίτη φάση του κύκλου ζωής των επιχειρηματικών διαδικασιών περιγράφει και περικλείει τον πραγματικό χρόνο και διαδικασία εκτέλεσης των επιχειρηματικών διαδικασιών. Τα διαφορετικά στιγμιότυπα (instances) μιας επιχειρηματικής διαδικασίας αρχικοποιούνται λόγω της παρέλευσης και της πυροδότησης κάποιου προκαθορισμένου γεγονότος, και στη συνέχεια εκτελούνται προς εκπλήρωση ενός επιχειρηματικού σκοπού. Ένα σύστημα λογισμικού διαχείρισης των επιχειρηματικών διαδικασιών είναι υπεύθυνο για τον έλεγχο της σειράς εκτέλεσης των επιχειρηματικών διαδικασιών με βάση ένα καθορισμένο πρότυπο – μοντέλο επιχειρηματικών διαδικασιών που έχει ήδη σχεδιαστεί. Παράλληλα ένα πρόγραμμα παρακολούθησης (monitoring component) του συστήματος λογισμικού έχει την ευθύνη για την παροχή ορθών πληροφοριών σχετικών με την κατάσταση των διαφορετικών στιγμιότυπων. Τέλος, κατά τη διάρκεια της φάσης εκτέλεσης συγκεντρώνονται σημαντικές πληροφορίες και δεδομένα εκτέλεσης με τη μορφή ενός αρχείου καταγραφής (log file).

1.3.1.4 Αποτίμηση (Evaluation)

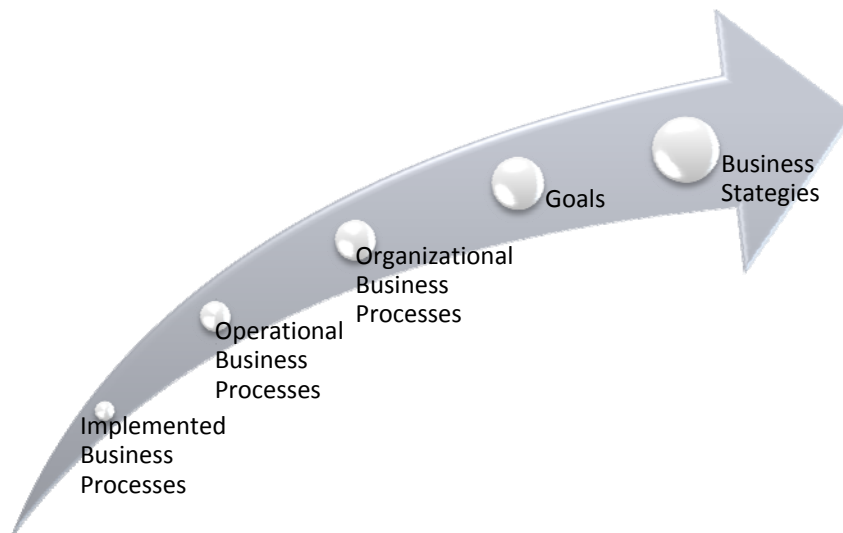
Η φάση της αποτίμησης χρησιμοποιεί τις πληροφορίες της προηγούμενης φάσης προκειμένου να εκτιμήσει και να βελτιώσει τα μοντέλα και τις υλοποιήσεις των επιχειρηματικών διαδικασιών με τη χρήση τεχνικών εξόρυξης δεδομένων (data mining techniques).

1.3.2 Κατηγοριοποίηση των Επιχειρηματικών Διαδικασιών

Στην συνέχεια, εξετάζονται οι διαστάσεις με βάση τις οποίες μπορούν να κατηγοριοποιηθούν οι επιχειρηματικές διαδικασίες.

1.3.2.1 Οργανωσιακές και Λειτουργικές (Organizational Versus Operational)

Κατά τον σχεδιασμό και την αποτύπωση των επιχειρηματικών διαδικασιών μπορούμε να διακρίνουμε και να αναγνωρίσουμε διάφορα επίπεδα που εκτείνονται από τις υψηλού επιπέδου επιχειρηματικές στρατηγικές μέχρι τις χαμηλού επιπέδου υλοποιήσιμες επιχειρηματικές διαδικασίες.



Σχήμα 5-3: Κατηγοριοποίηση Επιχειρηματικών Διαδικασιών

Στο υψηλότερο επίπεδο, καθορίζεται ο στρατηγικός σχεδιασμός της επιχείρησης που περιγράφει τους μακροπρόθεσμους στόχους ανάπτυξης ενός διαχρονικού ανταγωνιστικού πλεονεκτήματος στην αγορά. Στο δεύτερο επίπεδο η επιχειρηματική στρατηγική διασπάται σε επιμέρους λειτουργικούς στόχους. Στη συνέχεια, σε ένα τρίτο επίπεδο, εμφανίζονται οι οργανωσιακές επιχειρηματικές διαδικασίες (organizational business processes) που αποτελούν υψηλού επιπέδου επιχειρηματικές διαδικασίες και τυπικά καθορίζονται από δεδομένα εισόδου και εξόδου, τα επιδιωκόμενα αποτελέσματα και τις εξαρτήσεις με άλλες οργανωσιακές επιχειρηματικές διαδικασίες. Ενώ οι οργανωσιακές επιχειρηματικές διαδικασίες εμφανίζουν ένα είδος επιχειρηματικής λειτουργικότητας σχετικά αυτόνομης και ανεξάρτητης, τυπικά απαιτούνται πολλαπλές επιμέρους λειτουργικές επιχειρηματικές διαδικασίες να συμβάλλουν για την υλοποίησή τους. Στις λειτουργικές επιχειρηματικές διαδικασίες καθορίζονται οι δραστηριότητες τους και οι εξαρτήσεις μεταξύ τους, όμως αποκρύπτεται τελείως ο τρόπος υλοποίησής τους. Οι λειτουργικές επιχειρηματικές διαδικασίες καθορίζονται με βάση ειδικές τεχνικές και μοντέλα διαχείρισης επιχειρηματικών διαδικασιών (BPM) που έχουν αναπτυχθεί τα τελευταία χρόνια ως ένα μέσο γεφύρωσης του χάσματος μεταξύ της επιχειρηματικής και της υπολογιστικής πλευράς. Τέλος, στο χαμηλότερο επίπεδο βρίσκονται οι υλοποιήσιμες επιχειρηματικές διαδικασίες που περιέχουν πληροφορίες για την εκτέλεση των δραστηριοτήτων και για το τεχνικό και οργανωσιακό περιβάλλον εκτέλεσής τους.

1.3.2.2 Ενδοεπιχειρησιακές Διαδικασίες και Χορογραφίες Διαδικασιών (Intraorganizational Processes and Process Choreographies)

Ο συγκεκριμένος διαχωρισμός των επιχειρηματικών διαδικασιών αναφέρεται στο πεδίο χρησιμοποίησης και εξάρτησης των διαδικασιών. Στην περίπτωση που δεν υπάρχει αλληλεπίδραση και επικοινωνία με επιχειρηματικές διαδικασίες που εκτελούνται από διαφορετικούς οργανισμούς, τότε η επιχειρηματική διαδικασία ονομάζεται ενδοεπιχειρησιακή (intraorganizational). Οι ενδοεπιχειρησιακές επιχειρηματικές διαδικασίες εστιάζουν πρωταρχικά στον σχεδιασμό εσωτερικών διαδικασιών με την παράλληλη απομάκρυνση δραστηριοτήτων που δεν προσφέρουν

αξία και προσθέτουν πολυπλοκότητα κατά την εκτέλεση της διαδικασίας. Στις περισσότερες περιπτώσεις όμως, οι επιχειρηματικές διαδικασίες αλληλεπιδρούν και εξαρτούν την εκτέλεση τους από άλλες επιχειρηματικές διαδικασίες στα πλαίσια διαφορετικών οργανισμών σχηματίζοντας με αυτόν τον τρόπο χορογραφίες ή ενορχηστρώσεις διαδικασιών όπως θα αναλυθεί σε επόμενο κεφάλαιο. Κατά το σχεδιασμό και την εκτέλεση εξωτερικών επιχειρηματικών διαδικασιών πρέπει να λαμβάνονται υπόψη θέματα σχετικά με τα πρωτόκολλα επικοινωνίας, την ασφάλεια και την ενοποίηση των ανταλλασσόμενων μηνυμάτων και δεδομένων, τη διαλειτουργικότητα των εφαρμογών και τις μεθόδους αλληλεπίδρασης.

1.4 Ο Ρόλος των Ηλεκτρονικών Συναλλαγών στη Σύγχρονη Επιχείρηση

Οικονομικοί, κοινωνικοί, τεχνολογικοί και παράγοντες της παγκόσμιας αγοράς δημιουργούν ένα υψηλά ανταγωνιστικό επιχειρησιακό περιβάλλον, στο οποίο η σχέση ανάμεσα στους πελάτες και στις επιχειρήσεις είναι το σημείο εστίασης. Οι παράγοντες αυτοί έχουν τη δυνατότητα να αλλάξουν ταχύτατα - μερικές φορές με απρόβλεπτο τρόπο - με αποτέλεσμα οι επιχειρήσεις να λειτουργούν υπό αυξανόμενη πίεση και με λιγότερους πόρους. Ο ρυθμός των μεταβολών και ο βαθμός της αβεβαιότητας στο σύγχρονο ανταγωνιστικό περιβάλλον έχουν αναγάγει την αυτοματοποίηση των επιχειρηματικών διαδικασιών σε απαραίτητο στοιχείο επιβίωσης. Παράλληλα, τα διευρυμένα πλαίσια δραστηριότητας των επιχειρήσεων στηρίζουν την ύπαρξη και υποστήριξη τους σε μεγάλο βαθμό στην ευρύτατη εξάπλωση της τεχνολογίας των ενδοδικτύων (intranets) και του διαδικτύου (internet). Υποκινούμενες από τον σκληρό ανταγωνισμό που χαρακτηρίζει το παγκόσμιο εμπόριο αλλά και από τις αυξανόμενες απαιτήσεις των συνεργατών και πελατών τους, οι επιχειρήσεις υιοθετούν τεχνολογίες που διευκολύνουν τις διαδικτυακές εφαρμογές.

Γίνεται, λοιπόν, φανερό ότι προκύπτει η ανάγκη για την υιοθέτηση των ηλεκτρονικών συναλλαγών σε κάθε βήμα και έκφραση της επιχειρηματικής δράσης, τόσο σε ενδοεπιχειρησιακό επίπεδο όσο και σε διεπιχειρησιακό επίπεδο. Επομένως, το πλαίσιο των ηλεκτρονικών συναλλαγών είναι απαραίτητο να περιλαμβάνει τις καθιερωμένες εφαρμογές αλληλεπίδρασης μεταξύ επιχειρήσεων (B2B), επιχειρήσεων και πελατών (B2C) αλλά και επιχειρήσεων με κρατικούς οργανισμούς (B2G) προκειμένου να επιτύχει την ταχύτατη και αυτοματοποιημένη εκτέλεση των επιχειρηματικών διαδικασιών. Οι ηλεκτρονικές συναλλαγές αποτελούν πλέον σημαντικό πεδίο έρευνας και έχουν να αντιμετωπίσουν, πέραν των θεμάτων αυτοματοποίησης και βελτιστοποίησης της ανταλλαγής μηνυμάτων, και θέματα ασφάλειας, αντιμετώπισης της ετερογένειας των συναλλασσόμενων πλευρών και αλληλεξάρτησης μεταξύ τους.

Γενικότερα, στην προσπάθεια αυτοματοποίησης των διαδικασιών που λαμβάνουν χώρα στις ηλεκτρονικές συναλλαγές χρειάζονται να ληφθούν υπόψη τα εξής βασικά κριτήρια [10]:

- ✘ **Σύζευξη μεταξύ των εταιρών (Coupling):** Εκφράζει τον βαθμό (χαλαρή ή στενή σύζευξη) και την διάρκεια (δυναμική ή μακράς διάρκειας) της σύζευξης των εταιρών που συμμετέχουν σε μια ηλεκτρονική συναλλαγή. Με άλλα λόγια, πρόκειται για μια περιγραφή του τρόπου και του βαθμού με τον οποίο η

μεταβολή σε μια υπηρεσία – λειτουργία ενός συμμετέχοντα παράγοντα επηρεάζει τους υπόλοιπους.

- ✘ **Ετερογένεια (Heterogeneity)**: Αναφέρεται στον βαθμό διαφοροποίησης των συναλλασσόμενων ως προς τα τεχνικά, οργανωσιακά και σημασιολογικά χαρακτηριστικά. Εκφράζεται σε επίπεδο τύπων δεδομένων, δομικών χαρακτηριστικών στο επίπεδο των επιχειρηματικών διαδικασιών και σημασιολογίας για την σύνθεση των εργασιών.
- ✘ **Αυτονομία (Autonomy)**: Σχετίζεται με τον βαθμό συμμόρφωσης και προσκόλλησης των εταίρων ως προς τα παγκόσμια πρότυπα. Η αυτονομία αναγνωρίζεται σε επίπεδο σχεδιασμού, επικοινωνίας και εκτέλεσης. Σε ένα πλήρες αυτόνομο περιβάλλον καθένας θεωρείται ως ένα μαύρο κουτί που επικοινωνεί με τους εταίρους του μέσω διαπροσωπειών ενώ έχει τη δυνατότητα να ελέγχει και να μεταβάλλει τις εφαρμογές του χωρίς να επηρεάζει τους άλλους.
- ✘ **Εξωτερική διαχείριση των διαδικασιών (External Manageability)**: Αφορά στην εξωτερική επίβλεψη και διαχείριση των εφαρμογών των συνεργατών. Δίνεται έτσι η ευκαιρία επίβλεψης και ελέγχου μιας εφαρμογής ως προς την εκτέλεσή της και την επίδοσή της αλλά και πρόβλεψης της κατάστασής της και της διαθεσιμότητάς της.
- ✘ **Προσαρμοστικότητα (Adaptability)**: Σχετίζεται με τη δυνατότητα των εφαρμογών να προσαρμόζεται στις διάφορες μεταβολές. Πράγματι στο σύγχρονο ανταγωνιστικό επιχειρηματικό περιβάλλον, η γρήγορη προσαρμογή στις συνεχείς μεταβολές της αγοράς αποτελεί ζητούμενο αν όχι ανταγωνιστικό πλεονέκτημα.
- ✘ **Ασφάλεια (Security)**: Αναφέρεται στην ανάγκη ανάπτυξης των απαραίτητων συνθηκών ασφάλειας των εφαρμογών των διαφόρων συμμετεχόντων μέσα από την εξασφάλιση αμοιβαίας ταυτοποίησης, εμπιστευτικότητας, ακεραιότητας της επικοινωνίας, αυθεντικότητας, επικύρωσης.
- ✘ **Κλιμάκωση (Scalability)**: Εκφράζει τη δυνατότητα ενός συστήματος να επεκταθεί ως προς τον όγκο των προσπελάσιμων δεδομένων, τον αριθμό των συναλλαγών καθώς και το πλήθος των εξαρτήσεων που μπορεί να υποστηρίξει.

1.4.1 Χαλαρή και Ισχυρή Σύζευξη

Σημαντικό ρόλο στο σύγχρονο επιχειρησιακό περιβάλλον με τα μεγάλης κλίμακας καταναμημένα συστήματα διαδραματίζει η έννοια της σύζευξης μεταξύ των συνεργαζόμενων μονάδων λογισμικού λόγω της ανάγκης για την επίτευξη κλιμάκωσης των πληροφοριακών συστημάτων και ανοχής των προγραμματιστικών σφαλμάτων. Η σύζευξη υποδεικνύει το βαθμό της εξάρτησης [16] μεταξύ των συμμετεχόντων στην επιχειρηματική διαδικασία συστημάτων, με αποτέλεσμα να επηρεάζει την υπολογιστική αρχιτεκτονική και να δρα καθοριστικά στην επιτυχία ή όχι ενός συστήματος λογισμικού. Διακρίνεται σε χαλαρή σύζευξη (loose coupling) και ισχυρή – στενή σύζευξη (tight coupling).

Στην περίπτωση της στενής σύζευξης (tight coupling) εμφανίζεται άμεση αλληλεξάρτηση ανάμεσα στα αλληλεπιδρώντα στοιχεία, σε τέτοιο βαθμό ώστε η απόδοση και η ακεραιότητα του ενός συστήματος να επηρεάζει άμεσα και τα υπόλοιπα. Υπό συνθήκες στενής αλληλεξάρτησης υπάρχει ευκολότερος και

πληρέστερος κεντρικός έλεγχος των διαδικασιών που λαμβάνουν χώρα ωστόσο εμφανίζονται προβλήματα ευελιξίας και επέκτασης των διαδικασιών ή των ορίων των συναλλαγών.

Όμως, έχει αναφερθεί ότι η σύγχρονη ευμετάβλητη παγκόσμια αγορά ευνοεί και οδηγεί στην επιβίωση εκείνες τις επιχειρήσεις που μπορούν να ανταποκριθούν στον ταχύ ρυθμό των μεταβολών και να αποδεχτούν την ετερογένεια ως αναγκαίο χαρακτηριστικό της αρχιτεκτονικής τους υποδομής. Προς τη συγκεκριμένη κατεύθυνση γίνεται εμφανές ότι η σύγχρονη τάση σχεδιασμού και υλοποίησης του αρχιτεκτονικού τοπίου των πληροφοριακών συστημάτων έχει ως στόχο την ελαχιστοποίηση της επίδρασης που μπορεί να επιφέρει μια μεταβολή ή ένα σφάλμα. Δημιουργείται, επομένως η ανάγκη για χαλαρή σύζευξη, όπου η υπολογιστική πολυπλοκότητα μειώνεται στο ελάχιστο και ο αυξημένος βαθμός επέκτασης δίνει τη δυνατότητα μεταβολών και υιοθέτησης νέων σχέσεων και εξαρτήσεων. Η χαλαρή σύζευξη αποτελεί το σύγχρονο αξίωμα επιτυχίας σε ένα καταναμημένο υπολογιστικό σύστημα σε περίπτωση που αντιμετωπίζεται με καλά μελετημένη διαχείριση.

Η επίτευξη της ζητούμενης σύζευξης εμφανίζει διάφορες μορφές εφαρμογής και εκτείνεται από φυσικό επίπεδο μέχρι το επίπεδο της επιχειρηματικής λογικής, όπως φαίνεται στον παρακάτω πίνακα.

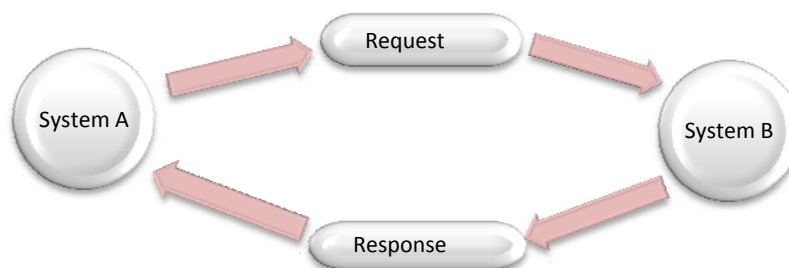
Στο επίπεδο της φυσικής σύνδεσης, η στενή σύζευξη υπαγορεύει την απευθείας σύνδεση σημείο προς σημείο των συνεργαζόμενων μελών, ενώ στην αντίθετη περίπτωση η χαλαρή σύζευξη καθιστά απαραίτητη την ύπαρξη ενός ενδιάμεσου μέσου διασύνδεσης (π.χ. μεσίτης (mediator)) υπεύθυνου για την ορθή επιλογή και τη δρομολόγηση της σύνδεσης.

	Tight Coupling	Loose Coupling
Physical Connections	Point-to-Point	Mediator
Communication Style	Synchronous	Asynchronous
Data Model	Common Complex Types	Simple Common Types only
Type System	Strong	Weak
Interaction Pattern	Navigate through Complex Object Trees	Data-Centric, Self-Contained Message
Control of Process Logic	Central Control	Distributed Control
Binding	Statically	Dynamically
Platform	Strong Dependencies	Platform Independent
Transactionality	2PC (two-face commit)	Compensation
Deployment	Simultaneous	At different times
Versioning	Explicit Upgrades	Implicit Upgrades

Πίνακας 5-1: Σύγκριση Ισχυρής – Χαλαρής Σύζευξης

Στο επίπεδο της επικοινωνίας, η στενή σύζευξη στηρίζεται σε σύγχρονες κλήσεις με τον καλούντα να βρίσκεται σε αναμονή μέχρι την ύπαρξη απάντησης, ενώ η χαλαρή σύζευξη υπαγορεύει τη χρήση ασύγχρονης επικοινωνίας με τον καλούντα να μην βρίσκεται απαραίτητα σε συγχρονισμό με τον καλούμενο και με την παροχή της δυνατότητας να συνεχίζεται η ροή της επιχειρηματικής διαδικασίας χωρίς να

διακόπτεται λόγω αναμονής για απόκριση ή λόγω εύρεσης εκτός λειτουργίας ενός από τους συναλλασσόμενους.



Σχήμα 5-4: Ασύγχρονη Επικοινωνία

Σημαντικό επίπεδο σύζευξης με πολύπλευρες προεκτάσεις είναι η επιλογή των τύπων δεδομένων [22]. Στη στενή σύζευξη είναι δυνατή η ανάπτυξη κοινών προτύπων επιχειρηματικών δεδομένων με πολύπλοκη δομή και υψηλή απόκρυψη (overriding) απλών τύπων δεδομένων. Αντίθετα, η χαλαρή σύζευξη αποσκοπεί στη χρήση μόνο απλών τύπων δεδομένων χωρίς πολύπλοκες δομές και αυστηρά καθορισμένα πρότυπα, ώστε να αποφευχθεί ο μεγάλος βαθμός εξάρτησης μεταξύ συστημάτων που διαχειρίζονται παρόμοιους τύπους δεδομένων και πληροφορίας.

Ένα πρόσθετο επίπεδο όπου διακρίνεται χαλαρή και στενή σύζευξη αφορά στον τρόπο που υλοποιείται η λογική διάσταση (logic) των διαδικασιών, δηλαδή αν αυτό γίνεται κεντροποιημένα ή κατανεμημένα. Στην περίπτωση που οι διαδικασίες είναι κεντρικά διαχειριζόμενες εφαρμόζεται το χαρακτηριστικό της στενής σύζευξης όπως για παράδειγμα στις βάσεις δεδομένων όπου γίνεται κεντρικός έλεγχος της συνέπειας των δεδομένων. Από την άλλη πλευρά, χαλαρή σύζευξη παρατηρείται στην περίπτωση των κατανεμημένων B2B διαδικασιών όπου εκεί υπάρχει ανεξαρτησία αλλά και μεγαλύτερος κίνδυνος για ασυνέπεια των δεδομένων των διαφόρων διαδικασιών ή ασυνέπεια της κατάστασης των συναλλαγών.

Συγχρόνως, ο τρόπος που γίνεται η εύρεση των συνεργατών και των συνεργαζόμενων υπηρεσιών χαρακτηρίζεται από χαλαρή ή στενή σύζευξη. Οι στατικά συνδεδεμένες υπηρεσίες αποτελούν ένα παράδειγμα στενής σύζευξης ενώ η δυναμική σύνθεση των υπηρεσιών είναι αντιπροσωπευτική της χαλαρής σύζευξης (π.χ. UDDI [40] στο περιβάλλον των Διαδικτυακών Υπηρεσιών).

Τέλος, η αντιμετώπιση των σφαλμάτων μέσω αυστηρών πρωτοκόλλων ή μέσω αποζημίωσης των ενεργειών (compensation) που απέτυχαν, η επιλογή του χρονικού σημείου (compile time or runtime) που πραγματοποιείται η διαδικασία της συσχέτισης μεταξύ των διαφορετικών μονάδων λογισμικού (early or late binding), και η ετερογένεια ή όχι των εργαλείων – πλατφόρμων υλοποίησης αποτελούν μορφές εφαρμογής σύζευξης.

2 Τεχνολογίες και Εργαλεία

2.1 Αρχιτεκτονική Προσανατολισμένη στις Υπηρεσίες (Service-Oriented Architecture)

2.1.1 Γενική Περιγραφή

Έχει γίνει προφανές ότι το κλειδί για τη σύγχρονη διαχείριση των καταναμημένων πληροφοριακών συστημάτων και παράλληλα για το σχεδιασμό των επιχειρηματικών διαδικασιών είναι η ευελιξία (flexibility). Η ευελιξία αποτελεί κυρίαρχο παράγοντα στον σύγχρονο αρχιτεκτονικό σχεδιασμό της υπολογιστικής υλικής υποδομής και του επιχειρησιακού λογισμικού για να αντιμετωπιστεί η απαίτηση που δημιουργούν οι συνεχώς μεταβαλλόμενες συνθήκες της παγκόσμιας αγοράς, με αποτέλεσμα να αναδεικνύεται σε σημαντική επιχειρηματική αξία (business value).

Συγχρόνως, οι διαδικασίες και τα συστήματα γίνονται ολοένα και περισσότερο πολύπλοκα, με αποτέλεσμα η προηγούμενη ανάγκη για αυτοματοποίηση των διαδικασιών ενός μόνο συστήματος να δίνει τη θέση της στην ανάγκη για διαλειτουργικότητα και επικοινωνία μεταξύ ετερογενών υποσυστημάτων ενός πληροφοριακού καταναμημένου συστήματος. Οπότε απαραίτητη κρίνεται η αυτονομία των συστημάτων με σκοπό να επιλύσουν προβλήματα ενοποίησης (integration) εφαρμογών, διαχείρισης συναλλαγών, θέματα ασφάλειας ενώ παράλληλα επιδιώκουν την συνεργασία διαφορετικών legacy συστημάτων και την επικοινωνία ανεξαρτήτως πρωτοκόλλων και τεχνολογικών υποδομών. Η σύγχρονη πρόκληση, όμως, της επιστήμης των Πληροφοριακών Συστημάτων περιλαμβάνει και τη συντηρησιμότητα (maintainability) και την επαναχρησιμοποίηση (reusability) πόρων και διαδικασιών προκειμένου να βελτιώνεται η ανταπόκριση του πληροφοριακού συστήματος στις μεταβολές και να αποφεύγεται η επιπλέον σπατάλη οικονομικών πόρων.

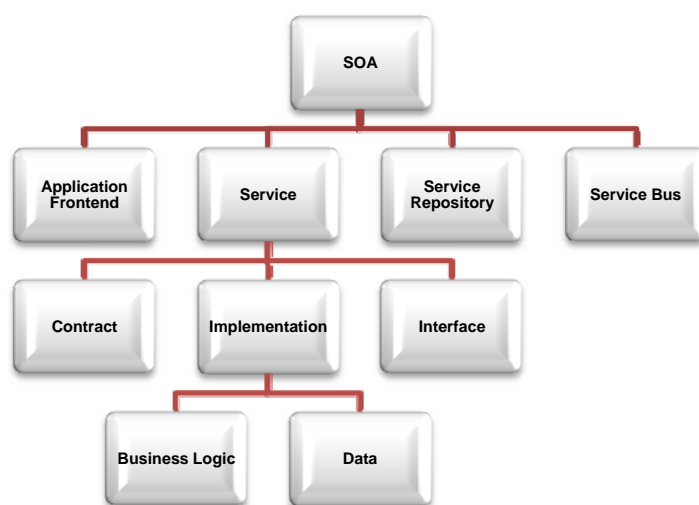
Επομένως, είναι εμφανές ότι οι προϋπάρχοντες τρόποι αντιμετώπισης προβλημάτων κλιμάκωσης (scalability) και κατανομής - διασποράς (distribution) των επιχειρηματικών διαδικασιών με τη χρήση κεντροποιημένων λειτουργιών εναρμόνισης και ελέγχου, δεν μπορούν να δώσουν λύση ούτε να ανταπεξέλθουν στις σύγχρονες απαιτήσεις. Στα πλαίσια της σύγχρονης επιχειρησιακής ανάγκης για ευελιξία, αυτονομία και ικανότητα συντήρησης των επιχειρηματικών διαδικασιών και συναλλαγών προκειμένου να αποκτήσει η σύγχρονη επιχείρηση στρατηγικό πλεονέκτημα και να ανταπεξέλθει στις ανταγωνιστικές συνθήκες της παγκόσμιας αγοράς, κρίθηκε απαραίτητο να αναζητηθεί μια νέα αρχιτεκτονική προσέγγιση για την οργάνωση και επικοινωνία των πληροφοριακών συστημάτων – μια προσέγγιση που να αποδέχεται την ετερογένεια και να οδηγεί στην αποκεντροποίηση των διαδικασιών.

Η Αρχιτεκτονική Προσανατολισμένη σε Υπηρεσίες, Service Oriented Architecture (SOA), είναι η προσέγγιση που καλείται να υλοποιήσει τις επιδιώξεις των επιχειρήσεων και να αντιμετωπίσει τις σύγχρονες προκλήσεις, θέτοντας στο κέντρο των επιχειρηματικών διαδικασιών και συναλλαγών τις υπηρεσίες (services). Στην ουσία, πρόκειται για μια αρχιτεκτονική φιλοσοφία. Μια φιλοσοφία που στοχεύει στην πραγματοποίηση και συντήρηση επιχειρηματικών διαδικασιών, ώστε να είναι δυνατή η αποδοχή και χρησιμοποίησή τους από πολύπλοκα καταναμημένα υπολογιστικά

συστήματα. Άλλωστε, είναι εμφανές ότι η Αρχιτεκτονική Προσανατολισμένη σε Υπηρεσίες αποδέχεται την ετερογένεια και την αποκεντροποίηση και τα καθιστά απαραίτητα και κυρίαρχα στοιχεία γύρω από τα οποία αναπτύσσει τις υπόλοιπες κατευθυντήριες γραμμές της.

Επομένως μπορούμε να καταλήξουμε ότι η Υπηρεσιοστραφής Αρχιτεκτονική αποτελείται από τρία βασικά στοιχεία [30]:

- ✘ **Τις Υπηρεσίες (Services)** που είναι αυτόνομες (self-contained) και καλά ορισμένες μονάδες επιχειρηματικής λειτουργικότητας.
- ✘ Την υλική υποδομή πάνω στην οποία θεμελιώνεται η Υπηρεσιοστραφής Αρχιτεκτονική. Πρόκειται για τον **Ταμειευτήρα Υπηρεσιών (Service Repository)** που αποθηκεύει πληροφορίες σχετικές με τις λειτουργίες και τα δεδομένα των Υπηρεσιών και τον **Επιχειρησιακό Δίαυλο Υπηρεσιών (Enterprise Service Bus)** που καθιστά δυνατή την υλοποίηση της έννοιας της διαλειτουργικότητας και συμβάλλει στην ευφυή κατανομή και επικοινωνία των επιχειρηματικών δεδομένων και διαδικασιών μεταξύ πολλαπλών συστημάτων που χρησιμοποιούν διαφορετικές και ετερογενείς τεχνολογίες.
- ✘ **Τη γαλαρή σύζευξη (loose-coupling)** που αντιπροσωπεύει την έννοια της μικρής εξάρτησης μεταξύ των διαφορετικών συστημάτων και ασχολείται με τη μείωση του κινδύνου και των συνεπειών λόγω μεταβολών και ασυνεπειών των πολλαπλών υποστηρικτικών συστημάτων (backends). Με αυτόν τον τρόπο είναι δυνατό να επιτευχθεί ο στόχος για ευελιξία, κλιμάκωση και ανοχή ασυνεπειών και λαθών.



Σχήμα 6-1: Επίπεδα αρχιτεκτονικής προσανατολιζόμενης στις υπηρεσίες (SOA)

Συνοψίζοντας, θα μπορούσαμε να δούμε τους διάφορους γενικούς ορισμούς της SOA αρχιτεκτονικής που έχουν προταθεί κατά καιρούς, ώστε να γίνουν εμφανή τα βασικά χαρακτηριστικά της προτού αναλυθούν με λεπτομέρεια. Αρχίζουμε με τον ορισμό που δίνεται από τον οργανισμό OASIS [33]: «είναι ένα παράδειγμα οργάνωσης και χρήσης των καταναμημένων δυνατοτήτων που μπορεί να βρισκονται υπό τον έλεγχο διαφορετικών τομέων δικαιοδοσίας. Παρέχει ένα ενιαίο μέσο προσφοράς, ανεύρεσης, αλληλεπίδρασης και χρήσης των δυνατοτήτων με στόχο την επίτευξη των επιθυμητών αποτελεσμάτων που συμφωνούν με μετρήσιμες προϋποθέσεις και επιδιώξεις» (OASIS

definition for SOA: “A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.”). Γίνεται φανερό ότι στον ορισμό τονίζονται η έννοια της ετερογένειας και της διασποράς των καταναμημένων συστημάτων ως ανάγκη για την υιοθέτηση της SOA αρχιτεκτονικής.

Ένας εναλλακτικός ορισμός της SOA αρχιτεκτονικής δίνεται στο δικτυακό τόπο της Wikipedia, όπου τονίζεται η χρήση των Υπηρεσιών ως το μέσο επικοινωνίας και ανταλλαγής πόρων χωρίς να είναι γνωστή η υλοποίηση τους από τα συστήματα από τα οποία διέρχονται. Συγκεκριμένα : «ο όρος Αρχιτεκτονική Προσανατολισμένη σε Υπηρεσίες εκφράζει μια προοπτική αρχιτεκτονικής λογισμικού που ορίζει τη χρήση των Υπηρεσιών για να υποστηρίξει τις απαιτήσεις των χρηστών. Σε ένα περιβάλλον SOA, οι πόροι σε ένα δίκτυο είναι διαθέσιμοι με τη μορφή αυτόνομων Υπηρεσιών που μπορούν να προσπελαστούν χωρίς γνώση της τεχνολογίας υλοποίησης τους» (Wikipedia definition for SOA: “In computing, the term of Service-Oriented Architecture (SOA) expresses a perspective of software architecture that defines the use of services to support the requirements of software users. In an SOA environment, resources on a network are made available as independent services that can be accessed without knowledge of their underlying platform implementation”).

2.1.2 Βασικά Χαρακτηριστικά

2.1.2.1 Υπηρεσίες (Services)

Οι Υπηρεσίες είναι καλά ορισμένες αυτόνομες μονάδες που περιέχουν επιχειρηματική λειτουργικότητα και αποτελούν το κυρίως μέσο επικοινωνίας και αναπαράστασης διαδικασιών και λειτουργιών μεταξύ των διαφορετικών καταναμημένων πληροφοριακών συστημάτων [31]. Η υπηρεσία ως λειτουργική μονάδα αποτελείται από [47]:

- ✘ **το συμβόλαιο (contract)** που είναι μια ολοκληρωμένη περιγραφή - προδιαγραφή (specification) της Υπηρεσίας και περιγράφει τη λειτουργία, τη συμπεριφορά, τους περιορισμούς και τους όρους που διέπουν τη χρήση της. Αποτελεί ένα είδος παρουσίασης και συμφωνίας μεταξύ του προμηθευτή – παροχέα της Υπηρεσίας (Service Provider) με τον καταναλωτή της Υπηρεσίας (Service Consumer) και πρέπει να στηρίζεται σε μια πλήρως καθορισμένη διαπροσωπεία (interface).
- ✘ **τη διαπροσωπεία (interface)** που είναι η περιγραφή της υλοποίησης των βασικών λειτουργιών της Υπηρεσίας
- ✘ **την υλοποίηση (implementation)** που είναι η προγραμματιστική πραγμάτωση (δηλαδή σε μορφή προγράμματος λογισμικού) της επιχειρηματικής λειτουργικότητας της Υπηρεσίας, όπου συμπεριλαμβάνονται τα δεδομένα και οι μέθοδοι που χρησιμοποιούνται.
- ✘ **την επιχειρηματική λειτουργικότητα (business functionality)** που περιγράφει την επιχειρηματική λογική που υλοποιεί (business logic) και περιέχεται στην υλοποίηση του προγραμματιστικού κώδικα

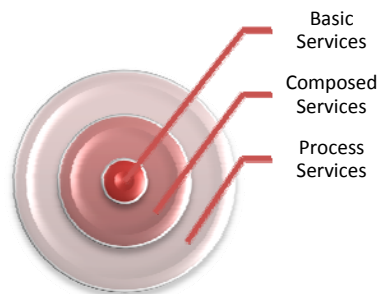
Παράλληλα αξίζει να αναφερθούμε στις επιπλέον ιδιότητες που διέπουν τις Υπηρεσίες ώστε να γίνει κατανοητό το πλαίσιο για το οποίο δημιουργήθηκε η ανάγκη χρησιμοποίησης τους από την SOA αρχιτεκτονική.

- ✘ **Αυτονομία:** Οι Υπηρεσίες είναι αυτόνομες (self-contained) επιχειρηματικές λειτουργικές μονάδες που σημαίνει ότι αποφεύγεται η μεταξύ τους εξάρτηση, ώστε να προωθείται το ζητούμενο της χαλαρής σύζευξης και της ανεξαρτησίας από την τεχνολογία υλοποίησης.
- ✘ **Επαναχρησιμοποίηση (Reusability):** Οι Υπηρεσίες πρέπει να είναι επαναχρησιμοποιήσιμες ώστε να αποφεύγονται οι περιττές πολλαπλές υλοποιήσεις σχετικά με την ίδια επιχειρηματική λειτουργία και να μπορούν χρησιμοποιηθούν για να συνθέσουν άλλες Υπηρεσίες.
- ✘ **Οι Υπηρεσίες είναι Stateless** που σημαίνει ότι δεν πρέπει να εξαρτώνται από την κατάσταση που βρίσκονται χωρίς να διατηρούν δεδομένα για αυτήν.
- ✘ **Οι Υπηρεσίες είναι Composable** που σημαίνει ότι είναι δυνατό να συνθέσουν άλλες Υπηρεσίες με τη μεταξύ τους ενορχήστρωση.
- ✘ **Οι Υπηρεσίες είναι Idempotent** που σημαίνει ότι έχουν την ικανότητα να επαναλάβουν την ίδια λειτουργία σε περίπτωση που είναι άγνωστο αν έχει ολοκληρωθεί η προηγούμενη κλήση χωρίς να υπάρξει πρόβλημα στα δεδομένα.
- ✘ **Οι Υπηρεσίες αποκρύπτουν την υλοποίηση τους και το λογικό υπόβαθρο που τις διέπει**
- ✘ **Οι Υπηρεσίες ακολουθούν και προωθούν το ζητούμενο της διαλειτουργικότητας λόγω της χαλαρής σύζευξης μεταξύ τους (Interoperable)**
- ✘ **Οι Υπηρεσίες είναι ανεξάρτητες της τεχνολογίας που της χρησιμοποιεί (Vendor-Diverse),** δηλαδή μπορούν να υλοποιηθούν από διαφορετικά συστήματα και προϊόντα λογισμικού.
- ✘ Σύμφωνα με την ιδέα του «Σχεδιασμού με βάση το Συμβόλαιο» (**“Design by Contract”**) οι Υπηρεσίες χαρακτηρίζονται από την κατάσταση πριν και μετά τη σύναψη συμβολαίου (Pre- και Post-Conditions) που συμβάλλει στην προδιαγραφή της σημασιολογικής (semantic) συμπεριφοράς των Υπηρεσιών [7]. Οι pre-conditions καθορίζουν και σχετίζονται με τις υποχρεώσεις που ο καταναλωτής πρέπει να ικανοποιεί κατά την κλήση της Υπηρεσίας, ενώ οι post-conditions καθορίζουν τις προδιαγεγραμμένες ιδιότητες που πρέπει να ικανοποιεί το σύστημα μετά την επιτυχή εκτέλεση και σχετίζεται ως ένα βαθμό με την Ποιότητα της Υπηρεσίας (Quality of Service).
- ✘ **Οι Υπηρεσίες είναι Coarse-Grained** που σημαίνει ότι πρέπει να είναι δομημένες κατά τέτοιο τρόπο ώστε η λειτουργία τους να περιορίζεται στο χαμηλότερο απλό επίπεδο (δηλαδή να αποφεύγεται η πολυπλοκότητα και να αναλύεται μέχρι το κατώτερο επίπεδο αυτονομίας).



Σχήμα 6-2: Χαρακτηριστικά Υπηρεσιών

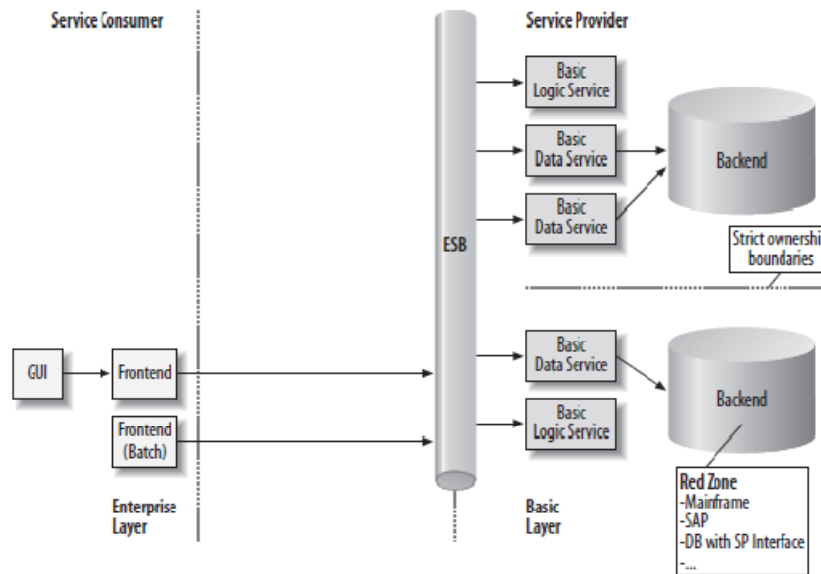
Όσον αφορά τα είδη των Υπηρεσιών που αναπτύσσουμε μπορούμε να κατηγοριοποιήσουμε τις Υπηρεσίες με βάση τα διάφορα επίπεδα χρησιμοποίησής τους στην SOA αρχιτεκτονική [2 16 38].



Σχήμα 6-3: Είδη Υπηρεσιών

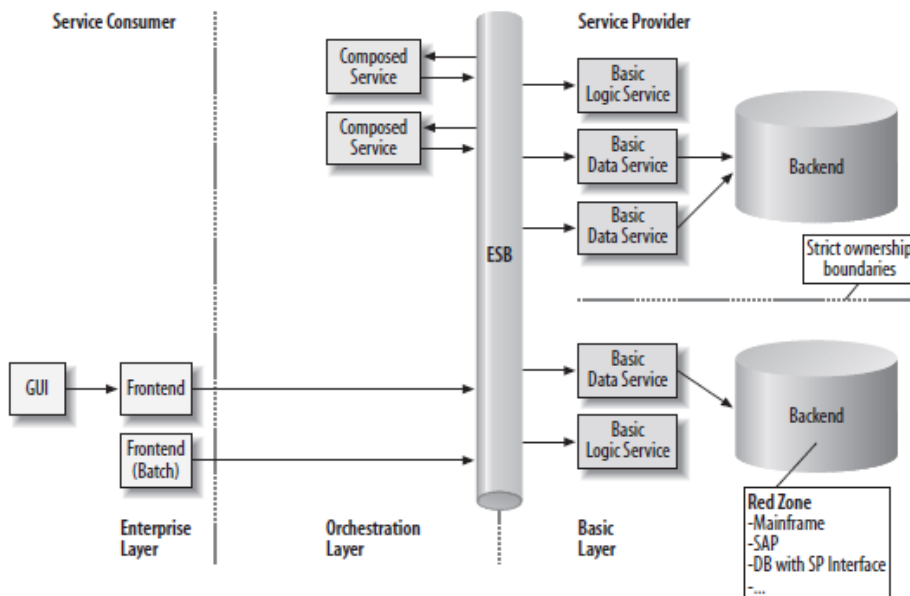
Επομένως, καταλήγουμε στις εξής κατηγορίες :

- ✘ **Βασικές Υπηρεσίες** που είναι εκείνες που παρέχουν βασικές επιχειρηματικές λειτουργίες και δεν μπορούν να διασπαστούν σε μικρότερες πολλαπλές λειτουργικές οντότητες. Οι υπηρεσίες αυτές παρέχουν το πρώτο θεμελιώδες επιχειρηματικό επίπεδο (business layer) που στη περίπτωση της SOA αρχιτεκτονικής αντιστοιχεί στο λεγόμενο **“Fundamental SOA”**, όπου ο ρόλος των Βασικών Υπηρεσιών συνίσταται στο να δημιουργήσουν πρόσβαση προς ένα μόνο συγκεκριμένο backend. Οι Βασικές Υπηρεσίες διακρίνονται σε **Υπηρεσίες Δεδομένων** και **Υπηρεσίες Λογικής**. Στην πρώτη περίπτωση η χρήση τους περιορίζεται μόνο στην ανάγνωση και στην εγγραφή δεδομένων, ενώ στη δεύτερη περίπτωση αντιπροσωπεύουν βασικούς επιχειρηματικούς κανόνες (business rules).



Σχήμα 6-4: Fundamental SOA (Αρχιτεκτονική Βασικών Υπηρεσιών)

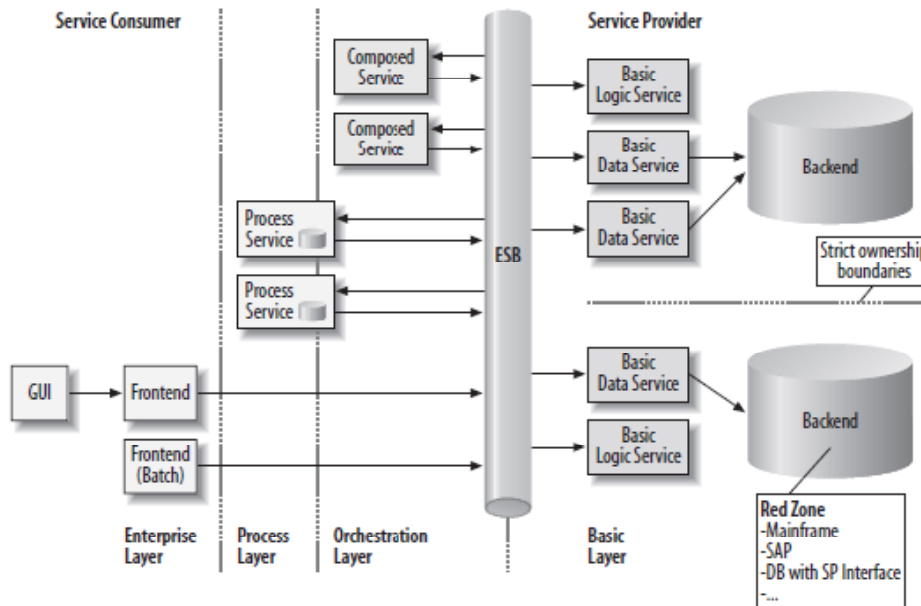
- ✘ Το δεύτερο στάδιο της επέκτασης αποτελούν οι Σύνθετες Υπηρεσίες (Composed Services) που συντίθεται με την ένωση Βασικών Υπηρεσιών. Οι Σύνθετες Υπηρεσίες λειτουργούν σε υψηλότερο επίπεδο από τις Βασικές και μπορούν να προσπελάσουν περισσότερα του ενός backends δημιουργώντας έτσι το δεύτερο επίπεδο της SOA αρχιτεκτονικής που ονομάζεται “**Federated SOA**”.



Σχήμα 6-5: Federated SOA (Αρχιτεκτονική Σύνθετων Υπηρεσιών)

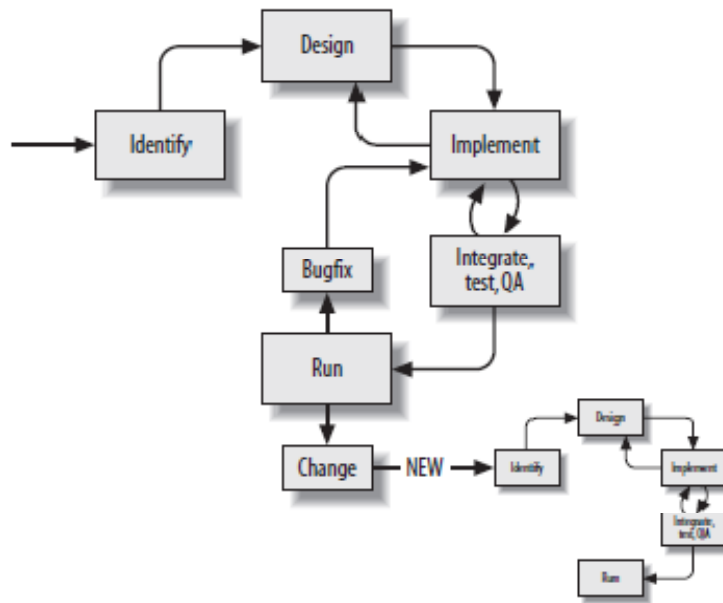
- ✘ Το τρίτο στάδιο της επέκτασης αποτελούν οι Υπηρεσίες Διαδικασιών (Process Services) που συντίθεται με την ένωση Βασικών και Σύνθετων Υπηρεσιών. Από επιχειρηματική σκοπιά οι Process Services αντιπροσωπεύουν μακροχρόνιες δραστηριότητες και ροές δεδομένων (long-

running) που προστίθενται ως τρίτο επιχειρηματικό επίπεδο στην SOA αρχιτεκτονική και ονομάζεται **“Process-Enabled SOA”**.



Σχήμα 6-6: Process-Enabled SOA (Αρχιτεκτονική Υπηρεσιών Διαδικασιών)

Τέλος, λαμβάνοντας υπόψη το γεγονός ότι οι Υπηρεσίες είναι μονάδες λογισμικού, μπορούμε να ισχυριστούμε ότι ακολουθούν κατά τη διάρκεια της ύπαρξής τους τον συνήθη κύκλο ζωής κάθε υπό ανάπτυξη λογισμικού, που κατά κανόνα συνίσταται από ένα βασικό κορμό, όπου κυριαρχούν οι φάσεις του σχεδιασμού (design), της υλοποίησης (implementation), της ενοποίησης (integration) και της εισόδου στην παραγωγική διαδικασία (production). Συγχρόνως μπορούμε να επεκτείνουμε τον κύκλο αυτό προσθέτοντας τα στοιχεία της επανάληψης (iteration) κατά την υλοποίηση, ώστε να επιδιώκεται το καλύτερο δυνατό αποτέλεσμα μέσα από μεταβολές της επιχειρηματικής λογικής και της προγραμματιστικής υλοποίησης, και της ταυτοποίησης – αναγνώρισης (identification) των αναγκαίων Υπηρεσιών που αποτελεί ένα είδος αναγνώρισης και προ-ανάλυσης ώστε να επιτυγχάνεται ο καλύτερος δυνατός σχεδιασμός των επιχειρηματικών διαδικασιών χωρίς περιττές απόπειρες υλοποίησης Υπηρεσιών που τελικά δε θα χρησιμοποιηθούν.



Σχήμα 6-7: Κύκλος ζωής Υπηρεσιών

Προχωρώντας στο επίπεδο της παραγωγής και της εκτέλεσης είναι απαραίτητο να προστεθεί και μια φάση κατά την οποία θα είναι δυνατή η αντιμετώπιση και διόρθωση σφαλμάτων και ασυνεπειών (bug fixing) και τέλος η δυνατότητα για πλήρη απόσυρση – αντικατάσταση της Υπηρεσίας. Προκύπτει, επομένως, ο παραπάνω κύκλος ζωής των Υπηρεσιών.

2.1.2.2 Επιχειρησιακός Δίαυλος Υπηρεσιών (Enterprise Service Bus - ESB)

2.1.2.2.1 Περιγραφή και Χαρακτηριστικά

Για να γίνει δυνατή η υλοποίηση της SOA αρχιτεκτονικής, απαιτείται μια μέθοδος κλήσης και διαχείρισης των Υπηρεσιών, ώστε να παραληφθούν από τον πραγματικό καταναλωτή που πρόκειται να τις χρησιμοποιήσει και να γίνει γενικά δυνατή η αξιόπιστη μεταφορά και μετατροπή των δεδομένων. Μια συγκεκριμένη υποδομή με καθορισμένα χαρακτηριστικά ικανά να αναδείξουν τη λειτουργικότητα και την πρωτοποριακή αντίληψη της SOA αρχιτεκτονικής είναι, λοιπόν, απαραίτητη για την μεταφορά και τη δρομολόγηση των Υπηρεσιών, ώστε να αποτελέσει τη ραχοκοκαλιά του αρχιτεκτονικού τοπίου. Τέτοια υποδομή αποδείχθηκε ο Επιχειρησιακός Δίαυλος Υπηρεσιών (Enterprise Service Bus, για συντομία ESB).

Η μεταφορά Υπηρεσιών από τον Παροχέα στον Καταναλωτή της Υπηρεσίας μπορεί να φαίνεται εύκολη υπόθεση στην πραγματικότητα, όμως, αποτελεί μια πολύπλοκη διαδικασία που απαιτεί και εξαρτάται από τεχνικές και οργανωσιακές προσεγγίσεις με χαρακτηριστικά και ευθύνες που περιλαμβάνουν :

- ✘ την παροχή συνδεσιμότητας
- ✘ την μεταφορά και μετατροπή των δεδομένων
- ✘ την (ευφυή) δρομολόγηση
- ✘ την ενασχόληση με θέματα ασφαλείας

- ✘ την ενασχόληση με θέματα αξιοπιστίας
- ✘ την διαχείριση των Υπηρεσιών
- ✘ την παρακολούθηση (monitoring) και την καταγραφή (logging) των γεγονότων και των μηνυμάτων

Οι συγκεκριμένες λειτουργίες στις περισσότερες περιπτώσεις είναι αναγκαίο να περιλαμβάνουν τη συμμετοχή διαφορετικών συστημάτων λογισμικού και πρωτοκόλλων μεταφοράς.

Επομένως, γίνεται φανερό ότι ο κύριος ρόλος ενός ESB είναι να συμβάλλει στρατηγικά στην παροχή της δυνατότητας για διαλειτουργικότητα των εφαρμογών. Επειδή, λοιπόν, περιλαμβάνει την ενοποίηση διαφορετικών πλατφόρμων λογισμικού και γλωσσών προγραμματισμού, ένα θεμελιώδες μέρος του ρόλου αυτού είναι η μετατροπή δεδομένων. Όπως δηλώνεται στο [14] :

«Η μετατροπή δεδομένων είναι κληρονομικά μέρος του διαύλου στην ανάπτυξη του ESB. Οι υπηρεσίες και οι μέθοδοι μετατροπής που εξειδικεύονται λόγω των αναγκών για αυτόνομες εφαρμογές που συνδέονται στον δίαυλο, μπορούν και πρέπει να βρίσκονται οπουδήποτε και να είναι προσβάσιμες από οπουδήποτε στον δίαυλο. Επειδή η μετατροπή δεδομένων είναι τόσο αναπόσπαστο κομμάτι του ESB, ο Δίαυλος Υπηρεσιών μπορεί να γίνει αντιληπτός »

Η συνήθης προσέγγιση είναι η εισαγωγή και η υιοθέτηση ενός συγκεκριμένου πλαισίου - (format) που να απεικονίζονται (map) όλα τα διαφορετικά APIs και πλατφόρμες λογισμικού. Όπως θα γίνει γνωστό παρακάτω το συγκεκριμένο πλαίσιο για τις Διαδικτυακές Υπηρεσίες είναι συνήθως το πρωτόκολλο SOAP.

Μια άλλη θεμελιώδης λειτουργία του Διαύλου Υπηρεσιών είναι η δρομολόγηση των Υπηρεσιών, ώστε να είναι δυνατή η ορθή μεταφορά των Υπηρεσιών από τον Παροχέα στον Καταναλωτή.

Τα υπόλοιπα χαρακτηριστικά και ευθύνες ενός ESB είναι συνήθως επέκταση της κύριας λειτουργίας που είναι η παροχή διαλειτουργικότητας και συζητούνται παρακάτω λεπτομερώς.

2.1.2.2.2 Συσχέτιση Δεδομένων (Data Mapping)

Λόγω της ανάγκης για χαλαρή σύζευξη, οδηγούμαστε σε μια κατάσταση όπου είναι απαραίτητο να υπάρχουν και να ορίζονται συγκεκριμένοι τύποι δεδομένων σχηματίζοντας ένα κοινά αποδεκτό μοντέλο δεδομένων για όλα τα διαφορετικά APIs. Επομένως, είναι αναγκαίο να εμφανίζεται ένα επίπεδο απεικόνισης (mapping layer) στην πλευρά του καταναλωτή, ώστε να απεικονίζονται τα δεδομένα του παροχέα στους δικούς του τύπους δεδομένων.

Εναλλακτικά, υπάρχουν και άλλες μέθοδοι απεικόνισης μέσω του καθορισμού proxies που απεικονίζουν τα δεδομένα μέσα στο πρωτόκολλο μεταφοράς ή μέσω ενός ενδιάμεσου στρώματος – επιπέδου κατά τη μετατροπή ανάμεσα από την καθορισμένη πλατφόρμα λογισμικού στο συγκεκριμένο πρωτόκολλο μεταφοράς.

2.1.2.2.3 Ευφυής Δρομολόγηση (Intelligent Routing)

Από επιχειρηματική σκοπιά και ανάγκη, μέσω του Δίαυλου Υπηρεσιών μπορεί να δοθεί η δυνατότητα ευφυούς δρομολόγησης των δεδομένων και των Υπηρεσιών με την εφαρμογή της μεθόδου των προτεραιοτήτων, όπου σε κάθε μήνυμα δίνεται και η ανάλογη προτεραιότητα εκτέλεσης. Βέβαια, γίνεται κατανοητό ότι για να γίνει πραγματικότητα η διαχείριση του περιεχομένου των μηνυμάτων και να επιτευχθεί η αναγκαία σήμανση προτεραιότητας, απαιτείται η σημασιολογική γνώση και κατανόηση από τον Δίαυλο Υπηρεσιών τμημάτων της δομής και της λογικής των Υπηρεσιών.

2.1.2.2.4 Ενασχόληση με Θέματα Ασφαλείας

Στα περισσότερα αρχιτεκτονικά τοπία της SOA αρχιτεκτονικής, είναι αναγκαία η διαχείριση και η αντιμετώπιση των θεμάτων ασφαλείας που ανακύπτουν. Θέματα σχετικά με την ασφάλεια στην SOA αρχιτεκτονική θα συζητηθούν σε μεταγενέστερο κεφάλαιο.

2.1.2.2.5 Ενασχόληση με Θέματα Αξιοπιστίας

Διαφορετικά πρωτόκολλα προσφέρουν διαφορετικές μορφές αξιοπιστίας. Δεν είναι όμως αναγκαίο να διασφαλίζουν την αξιόπιστη μεταφορά μηνυμάτων και την παράδοση τους στον σωστό παραλήπτη. Πράγματι, το HTTP πρωτόκολλο που είναι το πρωτόκολλο που υποστηρίζουν οι Διαδικτυακές Υπηρεσίες (που αποτελούν απαραίτητο στοιχείο της SOA αρχιτεκτονικής και θα συζητηθεί παρακάτω) δεν εξασφαλίζει από την φύση του την ορθή παράδοση.

Συνεπώς, ένας Δίαυλος Υπηρεσιών είναι αναγκαίο να καθορίζει τα πλαίσια αξιοπιστίας και διαχείρισης θεμάτων αξιοπιστίας. Ο τρόπος με τον οποίο σφάλματα και τεχνικά λάθη αντιμετωπίζονται μπορεί να επηρεαστεί από το πρωτόκολλο που χρησιμοποιεί το ESB, αλλά είναι το ίδιο το ESB που πρέπει να προσθέσει μηχανισμούς για να μεταβάλλει τη συγκεκριμένη συμπεριφορά.

2.1.2.2.6 Διαχείριση Υπηρεσιών (Service Management)

Καθώς το αρχιτεκτονικό τοπίο της SOA αρχιτεκτονικής μεγαλώνει και αυξάνεται σε μέγεθος, εμφανίζεται το πρόβλημα διαχείρισης των Υπηρεσιών. Αυτό το πρόβλημα μπορεί να είναι επιχειρηματικό ζήτημα ώστε να επιτευχθεί η δυνατότητα για εύκολη αναζήτηση και επαναχρησιμοποίηση των Υπηρεσιών ή μπορεί να υπάρχει η τεχνική απαίτηση για την ανάπτυξη και εκτέλεση μιας υποδομής διαχείρισης των Υπηρεσιών.

2.1.2.2.7 Επίβλεψη και Αρχειακή Καταγραφή (Monitoring and Logging)

Προκειμένου να επιτευχθεί ο στόχος της αξιοπιστίας και της διόρθωσης των σφαλμάτων είναι αναγκαίο να υφίσταται κάποιο είδος επιτήρησης και παρακολούθησης των Υπηρεσιών και της ανταλλαγής μηνυμάτων που πραγματοποιούνται μέσω του ESB. Επομένως, ο Δίαυλος Υπηρεσιών πρέπει να

παρέχει τη δυνατότητα της καταγραφής των ενεργειών που συμβαίνουν και της παρακολούθησης των Υπηρεσιών και την απόδοση του καταναμημένου συστήματος. Προς τη συγκεκριμένη κατεύθυνση πρέπει να υιοθετηθούν έννοιες και αντιλήψεις σχετικές με ταυτοποίηση και αναγνωριστικά συσχετίσεων με τη χρήση υπολογιστικών εργαλείων που υποστηρίζουν το monitoring και το debugging καταναμημένων συστημάτων.

2.1.2.2.8 Επίβλεψη Επιχειρηματικής Δραστηριότητας (Business Activity Monitoring - BAM)

Ιδιαίτερη σημασία από επιχειρηματική σκοπιά, έχει η προσέγγιση ανάπτυξης της δυνατότητας του Διαύλου Υπηρεσιών να επιβλέπει και να παρακολουθεί τις επιχειρηματικές διαδικασίες. Η ιδέα βασίζεται στο γεγονός ότι το ESB είναι δυνατό να παρέχει τη δυνατότητα να παρακολουθείται η ανταλλαγή μηνυμάτων κατά τέτοιο τρόπο ώστε να ενημερώνει για την κατάσταση των επιχειρηματικών δραστηριοτήτων δυναμικά (on the fly) και να επιφέρει άμεση αντίδραση σε περίπτωση μεταβολών. Με άλλα λόγια, η ιδέα του BAM αποτελεί ένα είδος εργαλείου για την συλλογή άμεσων πληροφοριών πάνω στις οποίες μπορεί να στηριχτεί το ESB για την μεταβολή των διαδικασιών. Επιτυγχάνεται στηριζόμενο σε τρεις βασικές τεχνολογικές αρχές [8]:

- ✘ **Συλλέγει Δεδομένα σε πραγματικό χρόνο:** Με την τοποθέτηση εντός της επιχειρηματικής διαδικασίας της BPEL αισθητήρων που να ενεργοποιούνται κάθε φορά που εκτελείται μια συγκεκριμένη δραστηριότητα ή ένα γεγονός και να συλλέγουν τα δεδομένα που εισέρχονται.
- ✘ **Αναλύει τις λειτουργίες και το περιεχόμενο:** Εμφανίζει την ικανότητα να συσχετίζει τα διάφορα γεγονότα και να συνδυάσει τη δράση τους ώστε να αναπτύξει αντικείμενα δεδομένων με επιχειρηματικό νόημα και αξία.
- ✘ **Δημιουργεί Interface για επιχειρηματικούς χρήστες:** Με αυτόν τον τρόπο είναι δυνατή η παρακολούθηση και δυναμική μεταβολή μέσω επιχειρηματικών κανόνων που είναι προσιτοί από επιχειρηματικούς αναλυτές.

2.1.2.2.9 Υποστήριξη κατά την Υλοποίηση Υπηρεσιών (Service Implementation Support)

Τέλος, υπάρχει η δυνατότητα να συνδέεται και να υποστηρίζει το ESB τη την ενορχήστρωση και τη σύνθεση των Υπηρεσιών. Έχει προταθεί να περιλαμβάνεται ενσωματωμένο στο ESB και τεχνολογία παραγωγής διαδικασιών ή ροών δεδομένων (process or workflow engine) ώστε να υπάρχει η δυνατότητα γραφικής αναπαράστασης κατά την παρακολούθηση των Υπηρεσιών και τοπικής δημιουργίας διαδικασιών που να συνυπάρχουν μέσα σε ένα μεγαλύτερο δίκτυο ενοποίησης.

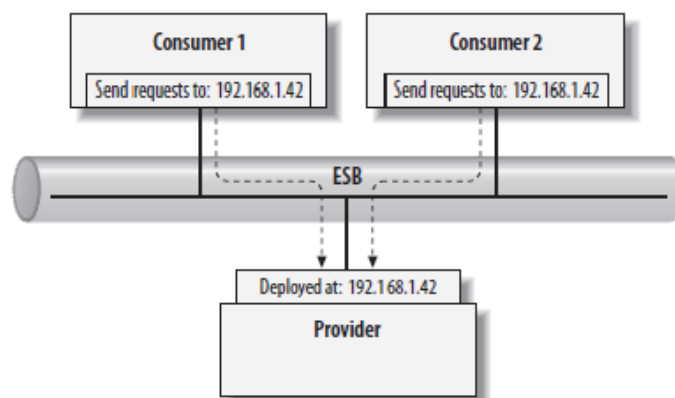
2.1.2.2.10 Σχεδιαστικές Αρχιτεκτονικές Προσεγγίσεις

Η επιλογή της σχεδιαστικής προσέγγισης που θα χρησιμοποιηθεί κατά την υλοποίηση του Διαύλου Υπηρεσιών εξαρτάται από διάφορους παράγοντες και διαφέρει ως προς τις δικαιοδοσίες και τις ευθύνες που μπορεί να μεταφέρει το ESB στους παροχείς και στους καταναλωτές. Από τη μία πλευρά, υπάρχει η δυνατότητα για απλό καθορισμό ενός πρωτοκόλλου επικοινωνίας και μεταφοράς των δεδομένων μεταφέροντας την

ευθύνη της μετατροπής και καθολικής απεικόνισης των δεδομένων και την ανάγκη για δρομολόγηση στους συνεργαζόμενους φορείς. Από την άλλη, υπάρχει η επιλογή ύπαρξης εργαλείων και τεχνολογιών στο ESB που να ενεργούν κεντρικά ή όχι και να είναι υπεύθυνα για το μεγαλύτερο μέρος καθορισμού των παραμέτρων επικοινωνίας και συνδεσιμότητας μεταξύ των εμπλεκόμενων φορέων. Παρακάτω παρέχεται μια επισκόπηση των αρχιτεκτονικών προσεγγίσεων και επιχειρείται σύγκριση των διαφορετικών σχεδιαστικών τεχνικών.

2.1.2.2.10.1 Σύνδεση Σημείο-προς-Σημείο και Διαμεσολάβηση (Point-to-Point Connection versus Mediation)

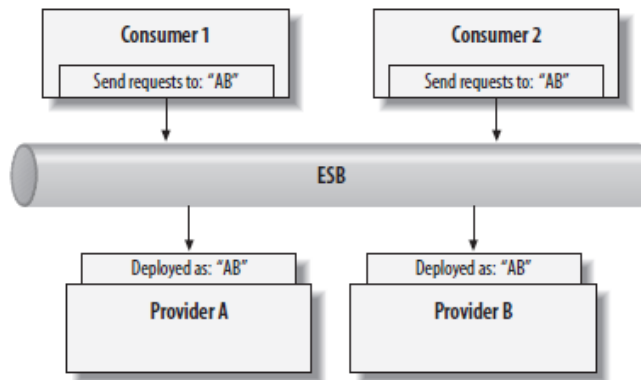
Κατά τον σχεδιασμό του Διάυλου Υπηρεσιών προκύπτει το ερώτημα για το επίπεδο της αφαιρετικότητας (abstraction) όσον αφορά τις διευθύνσεις των φυσικών συνδέσεων.



Σχήμα 6-8: Σημείο προς Σημείο Σύνδεση (Point-to-Point Connection)

Στην περίπτωση που ο καταναλωτής πρέπει να γνωρίζει την ακριβή διεύθυνση (endpoint) για την επικοινωνία του παροχέα Υπηρεσιών, η σύνδεση ονομάζεται σημείο-με-σημείο σύνδεση (point-to-point connection) και εμφανίζει το πρόβλημα πιθανής αποτυχίας της κλήσης στην περίπτωση που ο φυσικός αποδέκτης δεν είναι διαθέσιμος.

Στην περίπτωση που καταναλωτής δεν απαιτείται να έχει γνώση της φυσικής διεύθυνσης του παροχέα, αναγνωρίζει την προσφερόμενη Υπηρεσία από μία ετικέτα ή ένα συμβολικό όνομα που τη χαρακτηρίζει και στη συνέχεια αντιστοιχίζεται και δρομολογείται από το ESB στην κατάλληλη διεύθυνση. Η ετικέτα συνήθως περιέχει το όνομα της Υπηρεσίας και υπάρχει η περίπτωση να περιλαμβάνει και επιπρόσθετες πληροφορίες και ιδιότητες που επιδρούν στη διαδικασία της δρομολόγησης. Για παράδειγμα, ο Δίαυλος Υπηρεσιών μπορεί να διαχειρίζεται προτεραιότητες ή να ακολουθεί διαφορετική πολιτική για κάθε διαφορετικό καταναλωτή. Στην περίπτωση αυτή δρα ως μεσίτης (broker) ή μεσολαβητής (mediator), γεγονός που οδηγεί σε υποδομή χαλαρής σύζευξης.

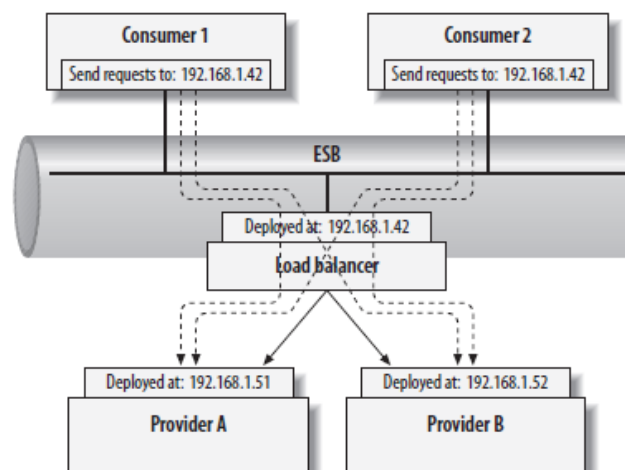


Σχήμα 6-9: Διαμεσολάβηση (Mediation)

Το πλεονέκτημα της έμμεσης προσέγγισης είναι ότι το ESB έχει τη δυνατότητα να αντιμετωπίσει δυναμικές μεταβολές στην αρχιτεκτονική της SOA προσέγγισης. Επομένως ενδεχόμενο φαινόμενο αποτυχίας στη σύνδεση δεν επηρεάζει την ομαλή λειτουργία της δρομολόγησης, καθώς η κλήση διαβιβάζεται σε ένα από τα υπόλοιπα συστήματα παροχής. Για να διαμορφωθεί η άμεση προσέγγιση και να καταστεί αποτελεσματική, το ESB είναι αναγκαίο να διαχειριστεί και να προχωρήσει στην επέκταση της πληροφορίας κατά τη στιγμή της έναρξης (startup time) ή κατά τον χρόνο της εκτέλεσης (runtime) επιτρέποντας στους παροχείς να εγγραφούν (register) όταν είναι διαθέσιμοι. Πάντως, όπως θα φανεί και στην επόμενη ενότητα, είναι δυνατή η έμμεση κλήση Υπηρεσιών και στην περίπτωση του point-to-point πρωτοκόλλου.

2.1.2.2.10.2 Αναχαιτιστές (Interceptors)

Για να καταστεί δυνατή η υποστήριξη έμμεσων κλήσεων Υπηρεσιών στην περίπτωση point-to-point πρωτοκόλλου, γίνεται χρήση ενδιάμεσων εργαλείων που ονομάζονται proxies ή interceptors («αναχαιτιστές»). Στην απλή περίπτωση γίνεται αντικατάσταση της φυσικής διεύθυνσης παροχής της Υπηρεσίας με μια μονάδα λογισμικού που λειτουργεί ως load balancer («εξισορροπιστής φορτίων») για να εκτελέσει την δρομολόγηση της κλήσης σε διαφορετική γνωστή διεύθυνση ενός παροχέα.



Σχήμα 6-10: Αναχαιτιστής (Interceptor)

Σε μια πολυπλοκότερη προσέγγιση, το ESB παρέχει έναν proxy ή έναν interceptor για κάθε παροχέα και για κάθε καταναλωτή. Σε αυτή την περίπτωση, ο καταναλωτής επικοινωνεί με point-to-point τρόπο μόνο με τον interceptor που του αντιστοιχεί.

2.1.2.2.11 Οδηγούμενο από πρωτόκολλο και Οδηγούμενο από API ESB (Protocol-Driven versus API-Driven ESB)

Εν συνεχεία προχωράμε σε ένα είδος κατηγοριοποίησης και σύγκρισης σε ότι αφορά το σημείο από το οποίο αρχίζουν οι ευθύνες του ESB από την οπτική γωνία των καταναλωτών και των παροχέων των Υπηρεσιών. Με την οδηγούμενη από το πρωτόκολλο (protocol-driven) προσέγγιση, το ESB καθορίζει ένα πρωτόκολλο και οι παροχείς και οι καταναλωτές στέλνουν και λαμβάνουν μηνύματα σύμφωνα με το συγκεκριμένο πρωτόκολλο (πρόσεγγιση που ακολουθεί η χρήση των Διαδικτυακών Υπηρεσιών).

Με την οδηγούμενη από το API (Application Programming Interface) προσέγγιση, το ESB καθορίζει APIs σχετικά με κάποια πλατφόρμα λογισμικού και οι παροχείς και οι καταναλωτές χρησιμοποιούν τα συγκεκριμένα APIs για την υλοποίηση και την κλήση των Υπηρεσιών.

Η διαφορά ως προς την προσέγγιση που θα χρησιμοποιηθεί διαδραματίζει ιδιαίτερα σημαντικό ρόλο κατά την διαδικασία της ανάπτυξης του Διαύλου Υπηρεσιών.

Αυτό συμβαίνει διότι στην πρώτη περίπτωση το ESB καθορίζει μόνο το πρωτόκολλο και η απεικόνιση και δρομολόγηση δεδομένων και κλήσεων είναι αποκλειστικά ευθύνη των διαφορετικών καταναλωτών και παροχέων, καθένας από τους οποίους επιλέγει ο ίδιος τα εργαλεία και την τεχνολογία που θα αναπτύξει και θα χρησιμοποιήσει για την προσαρμογή στο δεδομένο πρωτόκολλο.

Στην δεύτερη περίπτωση που το ESB είναι υπεύθυνο για τα APIs, το πρόβλημα της απεικόνισης των κλήσεων και της υλοποίησης των Υπηρεσιών σε καλά ορισμένα μηνύματα για την αποστολή στον Δίαυλο Υπηρεσιών είναι λειτουργία και ευθύνη του ESB.

2.1.2.3 Αποθήκη και Κατάλογος Υπηρεσιών (Service Repository – Service Registry)

Η Αποθήκη Υπηρεσιών παρέχει πληροφορίες και διαχειρίζεται τις Υπηρεσίες από επιχειρηματική σκοπιά. Με άλλα λόγια, διαχειρίζεται τα στοιχεία που αποτελούν τις Υπηρεσίες, δηλαδή τη διαπροσωπεία τους, τα συμβόλαια, τις μεταξύ τους εξαρτήσεις, τους περιορισμούς, με αποτέλεσμα να συνεισφέρει στην αναγνώριση, ταυτοποίηση, ανάπτυξη και σχεδιασμό Υπηρεσιών. Οι πληροφορίες που παρέχει ένας Ταμιευτήρας Υπηρεσιών πρέπει να είναι ανεξάρτητες από τεχνικές λεπτομέρειες για την υλοποίηση και την υλική υποδομή των Υπηρεσιών.

Από τεχνική σκοπιά, η καταγραφή των Υπηρεσιών πραγματοποιείται στον Κατάλογο Υπηρεσιών (Service Registry), όπου προσφέρονται τεχνικές πληροφορίες σχετικές με τις διαπροσωπείες των Υπηρεσιών απαραίτητες για τη χρήση τους και ικανές να

χρησιμοποιηθούν για την ορθή δρομολόγηση των κλήσεων στα διαφορετικά συστήματα που παρέχουν τις αντίστοιχες Υπηρεσίες. Ο Κατάλογος είναι δυνατό να είναι μέρος του ESB.

Η Αποθήκη Υπηρεσιών και ο Κατάλογος Υπηρεσιών είναι απαραίτητα στοιχεία του τοπίου της SOA αρχιτεκτονικής. Υπάρχει, μάλιστα, η δυνατότητα να χρησιμοποιείται ένα σύστημα και για τις δύο λειτουργίες. Μόνο που στη συγκεκριμένη περίπτωση υπάρχει ο κίνδυνος να δημιουργηθεί εξάρτηση της Αποθήκης από την υλική υποδομή που δημιουργεί πρόβλημα σε περίπτωση μελλοντικής μεταβολής της υποδομής.

2.1.2.4 Χαλαρή Σύζευξη (Loose-Coupling)

Έχει αναφερθεί εκτενώς σε προηγούμενο κεφάλαιο η έννοια της χαλαρής σύζευξης. Αυτό που αξίζει να αναφερθεί είναι ότι αποτελεί θεμελιώδες ιδέα της SOA αρχιτεκτονικής και πρέπει να χρησιμοποιηθεί προσεκτικά κατά τον σχεδιασμό ενός πληροφοριακού συστήματος με βάση τη συγκεκριμένη αρχιτεκτονική προκειμένου να αποφευχθεί η μεγάλη πολυπλοκότητα που οδηγεί σε μειωμένους χρόνους απόκρισης και λειτουργίας.

2.1.2.5 Αρχιτεκτονικά Πρότυπα με Βάση την SOA Αρχιτεκτονική (SOA-based Architectural Models)

Ανάλογα με την οπτική γωνία στην οποία εστιάζουμε, το διάγραμμα ενός αρχιτεκτονικού συστήματος βασισμένου στην SOA αντίληψη μπορεί να παρουσιαστεί με βάση επιχειρηματική, τεχνική ή λογική σκοπιά.

2.1.2.5.1 Λογικό – Επιχειρηματικό Μοντέλο (Logical - Business Architectural Model)

Σύμφωνα με το προτεινόμενο σχήμα που φαίνεται παραπάνω, στο λογικό αρχιτεκτονικό μοντέλο υπάρχουν διαφορετικά πεδία (domains) που διαδραματίζουν ένα συγκεκριμένο ρόλο και εμφανίζουν συγκεκριμένες ευθύνες. Ένα πεδίο (domain) είναι οτιδήποτε μπορεί να αναγνωριστεί ως οργανωσιακή (organizational) φυσική οντότητα με συγκεκριμένη επιχειρηματική δομή και ξεκάθαρη οργανωτική ιδιότητα (π.χ. μπορεί να είναι μια επιχειρηματική μονάδα, μια εταιρεία, ένα εταιρικό τμήμα ή μια ομάδα). Το εσωτερικό κομμάτι της εφαρμογής (backend) αντιπροσωπεύει το τεχνικό μέρος που κρύβεται πίσω από κάθε πεδίο (domain) και περιλαμβάνει τα πληροφοριακά υποσυστήματα και τις εφαρμογές λογισμικού που βρίσκονται υπό την ευθύνη του συγκεκριμένου τμήματος. Καθώς ανεβαίνουμε ιεραρχικά το μοντέλο συναντάμε τα διάφορα είδη Υπηρεσιών με την ιεραρχία που έχουμε περιγράψει, για να καταλήξουμε τελικά στο εξωτερικό κομμάτι της εφαρμογής (frontend) που αποτελεί τη μονάδα που βρίσκεται σε διαδραστική σχέση με τον χρήστη και ξεκινά μια επιχειρηματική διαδικασία και ολοκληρώνει την εκτέλεση συλλέγοντας τα αποτελέσματα. Χαρακτηριστικά μπορούμε να αναφέρουμε ότι ως frontend μπορεί να λειτουργήσει ένα portal, μια διαδικτυακή εφαρμογή, μια B2B εφαρμογή.

2.1.2.5.2 Τεχνικό Αρχιτεκτονικό Μοντέλο (Technical Architectural Model)

Στο τεχνικό αρχιτεκτονικό μοντέλο γίνεται εμφανές ότι κυριαρχεί η τεχνική σκοπιά με τον Δίαυλο Υπηρεσιών να βρίσκεται στο κέντρο για να συντονίζει, να ελέγχει και να συνδέει τα διαφορετικά υποσυστήματα. Άξιο παρατηρήσεως είναι το γεγονός ότι οι υψηλού επιπέδου Υπηρεσίες (Process Services και Logic Services) δεν παρέχονται από τα πεδία (domains) αλλά είναι αποτέλεσμα ειδικών τεχνολογικών εργαλείων.

2.1.2.5.3 Συνδυασμός Αρχιτεκτονικών Μοντέλων (Mixed Architectural Model)

Σε ένα ενδιάμεσο αρχιτεκτονικό μοντέλο με επιχειρηματική και τεχνική σκοπιά μπορούμε να παρατηρήσουμε ότι :

- ✘ οι βασικές Υπηρεσίες που περιέχουν επιχειρηματική λογική μπορούν να υλοποιηθούν μέσα από μια μηχανή κανόνων (rules engine) δίνοντας την κατάλληλη ευελιξία,
- ✘ οι Υπηρεσίες Διαδικασιών (Process Services) μπορούν να υλοποιηθούν και να διαχειριστούν από ένα εργαλείο BPM (Business Process Management) και
- ✘ οι κλήσεις των Υπηρεσιών φαίνονται ότι δρομολογούνται μέσα από τον Δίαυλο Υπηρεσιών.

2.1.2.6 Ασφάλεια (Security)

2.1.2.6.1 Βασικά Χαρακτηριστικά

Σε ένα καταναμημένο πληροφοριακό σύστημα, όπως αυτά που εξετάζουμε, η ύπαρξη επαρκών μέτρων ασφάλειας θεωρείται απαραίτητη προκειμένου να εξασφαλιστεί η εύρυθμη λειτουργία. Άλλωστε, η προσβολή του συστήματος από οποιοδήποτε είδος σφάλματος είναι δυνατό να έχει σημαντικές οικονομικές συνέπειες για μια επιχείρηση δεδομένης της στρατηγικής σημασίας των συστημάτων για την εξασφάλιση ανταγωνιστικού πλεονεκτήματος στην αγορά. Το θέμα της ασφάλειας έχει μια σειρά προεκτάσεων που απαιτούν την προσοχή μας και εντάσσονται κυρίως στις παρακάτω κατηγορίες :

- ✘ **Επικύρωση (Authentication):** Με την επικύρωση πραγματοποιείται επαλήθευση της ταυτότητας της φυσική οντότητας που απαιτεί είσοδο στο Πληροφοριακό σύστημα και αναφέρεται στην επαλήθευση του χρήστη ή της φυσικής συσκευής ή του εξωτερικού πελάτη που πρόκειται να καλέσει μια Υπηρεσία.
- ✘ **Έγκριση - Εξακριβωση (Authorization):** Με την έγκριση καθορίζεται το είδος της ταυτότητας του χρήστη (σε όποια μορφή βρίσκεται) που καλεί την Υπηρεσία και εξακριβώνεται αν πράγματι έχει τη δυνατότητα κλήσης της και παρακολούθησης των αποτελεσμάτων.
- ✘ **Εμπιστευτικότητα (Confidentiality):** Σημαντική παράμετρος της ασφάλειας αποτελεί ο καθορισμός της αξιοπιστίας των δεδομένων κατά τη μεταφορά και την αποθήκευση τους. Όσον αφορά τις Υπηρεσίες, αυτό μεταφράζεται στην διαβεβαίωση – διασφάλιση ότι κανείς πέρα από τον χρήστη της Υπηρεσίας δεν έχει πρόσβαση στα δεδομένα κατά τη μεταφορά τους.
- ✘ **Ακεραιότητα (Integrity):** Το κλειδί στο θέμα της ακεραιότητας είναι η διασφάλιση και η εγγύηση ότι τα δεδομένα δεν μπορούν να διαχειριστούν και

να counterfeit κατά τέτοιο τρόπο ώστε να είναι τα δεδομένα εντελώς επισφαλής και λανθασμένα ή ακόμα χειρότερα, τα αναγνωριστικά (credentials) της επικύρωσης και της εξακρίβωσης να είναι εντελώς ψευδή και κατά συνέπεια να μπορούν να αποκτήσουν πρόσβαση στα δεδομένα των Υπηρεσιών.

- ✘ **Διαθεσιμότητα (Availability):** Η ασφάλεια πρέπει να διασφαλίζει από περιπτώσεις όπου τα δεδομένα παραμένουν ακέραια και δεν διαφθείρονται, αλλά το σύστημα μεταβάλλεται κατά τέτοιο τρόπο ώστε να γίνεται αδρανές και να μην λειτουργεί.

Για την αντιμετώπιση των παραπάνω θεμάτων ασφαλείας στην SOA αρχιτεκτονική χρησιμοποιούνται οι ίδιες προσεγγίσεις με προηγούμενες αρχιτεκτονικές κατανεμημένων συστημάτων. Στην περίπτωση της επικύρωσης και της έγκρισης - εξακρίβωσης, υπάρχει η ανάγκη για την υιοθέτηση αναγνωριστικών χρήστη (User IDs) και κωδικών (passwords), ενώ για την περίπτωση της ακεραιότητας και της εμπιστευτικότητας μπορεί να χρησιμοποιηθούν τεχνολογίες όπως κρυπτογράφηση (encryption) και ψηφιακές υπογραφές (digital signatures). Συγκεκριμένα, από πρακτική άποψη, η αντιμετώπιση θεμάτων ακεραιότητας και εμπιστευτικότητας μπορεί να πραγματοποιηθεί στο επίπεδο της μεταφοράς (Transport-layer Security) ή στο επίπεδο των μηνυμάτων (Message-layer security). Στο στρώμα της μεταφοράς χρησιμοποιείται το πρωτόκολλο του συστήματος υποδομής για την εισαγωγή θεμάτων ασφαλείας. Στο στρώμα των μηνυμάτων, από την άλλη πλευρά, η ασφάλεια εισάγεται στο σώμα των μηνυμάτων που ανταλλάσσονται. Τέλος, θέματα ασφαλείας εισάγονται μέσα από τη χρήση των τεχνολογιών της XML και των Διαδικτυακών Υπηρεσιών που ακολουθούν συγκεκριμένες προδιαγραφές ασφαλείας μέσα από καθορισμένα πρωτόκολλα μεταφοράς μηνυμάτων.

2.1.2.6.2 Ετερογένεια και Ασφάλεια

Η SOA αρχιτεκτονική ασχολείται με επιχειρηματικές διαδικασίες που κατανέμονται μεταξύ διαφορετικών ετερογενών συστημάτων. Οι πολιτικές ασφαλείας για κάθε ένα σύστημα διαφέρουν μεταξύ τους, οπότε γίνεται αντιληπτό ότι δημιουργείται η πρόκληση για την εισαγωγή μιας γενικής ιδέας ασφαλείας πάνω από τα διαφορετικά επίπεδα ασφαλείας των διαφόρων συστημάτων.

2.1.2.6.3 Κατανεμημένες Διαδικασίες και Επίπεδα Αφαιρετικότητας

Το σημαντικότερο πρόβλημα στην περίπτωση της SOA αρχιτεκτονικής είναι ότι οδηγεί σε μεγάλο αριθμό διαφορετικών επιπέδων αφαιρετικότητας. Στην περίπτωση, λοιπόν, που κάθε Υπηρεσία αποκρύπτει την επιχειρηματική λειτουργικότητα ενός κατώτερου στρώματος Υπηρεσιών, δημιουργείται και το αναγκαίο αρνητικό στοιχείο της απόκρυψης του περιεχομένου της ταυτότητας του χρήστη από την υφιστάμενη διαδικασία. Σε συνδυασμό με τα θέματα ασφαλείας που εισάγει και η ανάγκη για ετερογένεια των διαφορετικών εσωτερικών τμημάτων των εφαρμογών (backends), γίνεται αντιληπτό ότι δημιουργείται πρόβλημα εξακρίβωσης και επικύρωσης στοιχείων και αναγνωριστικών, ενώ συγχρόνως λόγω της παράλληλης εφαρμογής διαφορετικών επιπέδων ασφαλείας ανάλογα με το ρόλο του συμμετέχοντα στη διαδικασία εμφανίζεται η ανάγκη για διασφάλιση της εμπιστευτικότητας των δεδομένων μεταξύ πολλαπλών κόμβων και συνδέσεων.

Για την αντιμετώπιση της ασαφούς κατάστασης των αναγνωριστικών ανάμεσα σε πολλαπλά frontends και backends, ιδανικά δημιουργείται η ανάγκη για γνώση και διαχείριση των ίδιων ταυτοτήτων και των ρόλων και των χαρακτηριστικών (profiles) με τα οποία σχετίζονται τόσο από τα frontends όσο και από τα backends για κάθε ένα χρήστη.

Όσον αφορά το πρόβλημα της εμπιστευτικότητας και της ακεραιότητας των δεδομένων που κατανέμονται μεταξύ των διαφορετικών συστημάτων, προκειμένου να αποφευχθεί η διαστρέβλωση και η υποκλοπή τους είναι αναγκαία η ύπαρξη ασφαλείας σε επίπεδο μηνυμάτων (message-layer security) ώστε να μην επηρεάζεται από οποιοδήποτε υποσύστημα κι αν προσπελαστεί.

2.2 SOA Τεχνολογίες

Προκειμένου να θεμελιωθεί το οικοδόμημα της Υπηρεσιοστραφούς Αρχιτεκτονικής και να γίνει πραγματικότητα η τεχνική και προγραμματιστική υλοποίηση της ήταν απαραίτητος ο ερχομός και η τελειοποίηση τεχνολογιών και τεχνολογικών εργαλείων βασισμένων σε πρότυπα διαδικτυακής επικοινωνίας (XML-based, HTTP-based) και επιχειρηματικού σχεδιασμού που διευκόλυναν τον σχεδιασμό των επιχειρηματικών διαδικασιών και την επικοινωνία και μεταφορά πληροφοριών μεταξύ ετερογενών συστημάτων. Στη συνέχεια αναλύονται με λεπτομέρεια οι συγκεκριμένες τεχνολογίες.

2.2.1 Διαδικτυακές Υπηρεσίες (Web Services)

Οι Διαδικτυακές Υπηρεσίες αποτελούν την τελευταία τεχνολογία για την επικοινωνία μεταξύ κατανεμημένων συστημάτων (distributed technology) και έχουν επικρατήσει ως η περισσότερη αποδεκτή τεχνολογία για να επιτευχθεί και να γίνει πραγματικότητα η SOA αρχιτεκτονική. Έχουν γίνει η κοινά αποδεκτή και περισσότερο χρησιμοποιήσιμη τεχνολογία για την πραγμάτωση και υλοποίηση της έννοιας της διαλειτουργικότητας και της ενοποίησης εφαρμογών και πληροφοριακών συστημάτων. Οι Διαδικτυακές Υπηρεσίες παρέχουν τα τεχνολογικά θεμέλια για την επίτευξη της διαλειτουργικότητας μεταξύ εφαρμογών που χρησιμοποιούν διαφορετικά ετερογενή λειτουργικά συστήματα, πλατφόρμες λογισμικού και γλώσσες προγραμματισμού. Από τεχνικής σκοπιάς στηρίζονται και υλοποιούνται σε γλώσσα XML. Ενώ, λοιπόν, η XML έχει επικρατήσει ως το de facto πρότυπο ενοποίησης σε επίπεδο δεδομένων, οι Διαδικτυακές Υπηρεσίες έχουν επικρατήσει ως de facto πρότυπο σε επίπεδο Υπηρεσιών μεταξύ των επιχειρήσεων.

Από τεχνολογική σκοπιά, οι Διαδικτυακές Υπηρεσίες είναι ένα είδος κατανεμημένης αρχιτεκτονικής. Το προγραμματιστικό παράδειγμα (paradigm) της κατανεμημένης αρχιτεκτονικής ξεκίνησε με το DCE (Distributed Computing Environment), το RPC (Remote Procedure Call) και τα συστήματα που ήταν προσανατολισμένα στην ανταλλαγή μηνυμάτων (Message-Oriented Middleware όπως τα MQ Series, MSMQ). Στη συνέχεια ακολούθησαν τα κατανεμημένα αντικείμενα και οι ανάλογοι Μεσίτες που ονομάζονταν ORBs (Object Request Brokers), όπως η CORBA αρχιτεκτονική (Common Object Request Broker Architecture), το μοντέλο DCOM (Distributed Component Object Model) και το RMI (Remote Method Invocation) για να

καταλήξουμε στα λεγόμενα component models, όπως τα EJB (Enterprise Java Beans), COM+ (Component Object Model), .NET Enterprise Services και το μοντέλο CCM (CORBA Component Model).

Η διαφορά που οδήγησαν στην υιοθέτηση των Διαδικτυακών Υπηρεσιών έναντι των προηγούμενων μοντέλων που αναφέρθηκαν ήταν η καθολική αποδοχή που είχαν από τις εταιρείες λογισμικού λόγω της εφαρμογής σε κάθε τεχνολογία λογισμικού. Ο [38] αναφέρει συγκεκριμένα ότι «Η σύγχρονη SOA αρχιτεκτονική αντιπροσωπεύει μια ... αρχιτεκτονική που προωθεί τον προσανατολισμό στις Υπηρεσίες και αποτελείται από ... υπηρεσίες που υλοποιούνται ως Διαδικτυακές Υπηρεσίες». Και ο [12] είχε προβλέψει ότι «το 2006 πάνω από το 60% των \$527 δις που δαπανούν οι εταιρείες πληροφορικής θα στηριχτεί στην εκμετάλλευση των δυνατοτήτων του πρότυπου και της τεχνολογίας των Διαδικτυακών Υπηρεσιών». Επομένως, αποτελούν την πρώτη τεχνολογία που ικανοποιεί την υπόσχεση για καθολική διαλειτουργικότητα μεταξύ εφαρμογών που εκτελούνται σε διαφορετικά συστήματα. Οι βασικές προδιαγραφές στις οποίες στηρίζονται οι Διαδικτυακές Υπηρεσίες είναι το πρωτόκολλο SOAP (Simple Object Oriented Protocol), η WSDL (Web Services Description Language) και το UDDI (Universal Description, Discovery and Integration). Οι παραπάνω τεχνολογίες στηρίζονται εξ' ολοκλήρου στην γλώσσα XML, καθιστώντας τα μηνύματα και την περιγραφή των Διαδικτυακών Υπηρεσιών αναγνώσιμα από άνθρωπο (human readable) χωρίς τεχνολογικές γνώσεις λογισμικού.

Από αρχιτεκτονική σκοπιά, οι Διαδικτυακές Υπηρεσίες μπορούν να λειτουργήσουν έχοντας οποιοδήποτε ρόλο και εισάγουν σημαντικές αλλαγές σε σχέση με τις προηγούμενες τεχνολογίες, με αποτέλεσμα να επικρατήσουν ως το σύγχρονο πρότυπο για τους παρακάτω λόγους :

- ✘ Οι Διαδικτυακές Υπηρεσίες υποστηρίζουν τη δυνατότητα χαλαρής σύζευξης μέσω λειτουργιών που ανταλλάσσουν μόνο δεδομένα, γεγονός που διαφέρει από τα μοντέλα Component και κατανεμημένων αντικειμένων, όπου εκτός των δεδομένων ανταλλάσσεται και η συμπεριφορά.
- ✘ Οι λειτουργίες στην περίπτωση των Διαδικτυακών Υπηρεσιών βασίζεται στην ανταλλαγή μηνυμάτων σε XML μορφή που αποτελούνται από ένα δομημένο σύνολο εισόδων, εξόδων και μηνυμάτων λάθους. Ο συνδυασμός των μηνυμάτων καθορίζει τον τύπο της λειτουργίας (μονόδρομη ή one-way, αίτημα/απόκριση ή request/response, μόνο απόκριση ή solicit response ή τέλος ανακοίνωση ή notification) που διαφέρει από προηγούμενες τεχνολογίες.
- ✘ Οι Διαδικτυακές Υπηρεσίες είναι stateless, δηλαδή δεν ακολουθούν το παράδειγμα των αντικειμένων.
- ✘ Οι Διαδικτυακές Υπηρεσίες παρέχουν υποστήριξη και στην περίπτωση ασύγχρονων (όπου δεν είναι αναγκαία η άμεση απόκριση προκειμένου να συνεχιστεί η εκτέλεση της λειτουργίας) και στην περίπτωση σύγχρονων (όπου απαιτείται άμεση απόκριση προκειμένου να συνεχιστεί η εκτέλεση της λειτουργίας) αλληλεπιδράσεων.
- ✘ Οι Διαδικτυακές Υπηρεσίες εισάγουν την έννοια και την ιδέα των endpoints και των intermediaries (μεσάζοντες), με αποτέλεσμα να αναπτύσσονται νέες προσεγγίσεις κατά τη διαχείριση μηνυμάτων.
- ✘ Οι Διαδικτυακές Υπηρεσίες χρησιμοποιούν πρότυπα διαδικτυακά πρωτόκολλα (standard internet protocols) όπως το HTTP (Hyper Text

Transfer Protocol), το SMTP (Simple Mail Transfer Protocol), το FTP (File Transfer Protocol) και το MIME (Multipurpose Internet Mail Extensions). Επομένως, η συνδεσιμότητα μέσα από κλασικές διαδικτυακές συνδέσεις, ακόμα και μέσα από εκείνες που προφυλάσσονται από firewalls, είναι λιγότερο προβληματική σε σχέση με προηγούμενες προσεγγίσεις που είχαν επικρατήσει.

Τέλος, αξίζει να αναφέρουμε τις τεχνικές προδιαγραφές που εισάγουν οι Διαδικτυακές Υπηρεσίες για να αντιμετωπίσουν θέματα ασφαλείας, συναλλαγών και ανταλλαγής μηνυμάτων. Συγκεκριμένα περιγράφουμε τις κυριότερες από αυτές που είναι οι εξής :

- ✘ **WS-Security [46]:** Είναι υπεύθυνο και διευθύνει θέματα (authentication) και ασφάλειας σε επίπεδο ανταλλαγής μηνυμάτων, ενώ συγχρόνως επιτρέπει την ασφαλή επικοινωνία μέσω της τεχνολογίας των Διαδικτυακών Υπηρεσιών.
- ✘ **WS-Coordination [42]:** Καθορίζει ένα πλαίσιο συντονισμού (Coordination Framework) για τις συναλλαγές μέσω Διαδικτυακών Υπηρεσιών και είναι το θεμέλιο πάνω στο οποίο στηρίχθηκε η ανάπτυξη των WS-Atomic Transaction και WS-Business Activity.
- ✘ **Transaction Specifications (WS-Atomic Transaction και WS-Business Activity) [41]:** Καθορίζει και παρέχει υποστήριξη για τις συναλλαγές μεταξύ καταναμημένων συστημάτων μέσω των Διαδικτυακών Υπηρεσιών. Το AtomicTransaction καθορίζει το πλαίσιο συναλλαγών για μικρό χρονικό διάστημα και τις ACID συναλλαγές, ενώ το BusinessActivity καθορίζει καταναμημένες επιχειρηματικές συναλλαγές μεγαλύτερης διάρκειας.
- ✘ **WS-Reliable Messaging:** Παρέχει υποστήριξη για αξιόπιστη και συνεπή επικοινωνία και παράδοση μηνυμάτων μεταξύ των Διαδικτυακών Υπηρεσιών (Web Services) πάνω από διάφορα πρωτόκολλα μεταφοράς και επικοινωνίας.
- ✘ **WS-Addressing:** Καθορίζει το συντονισμό και τη δρομολόγηση των μηνυμάτων.
- ✘ **WS-Inspection:** Παρέχει υποστήριξη για δυναμική ανακάλυψη εγγράφων που περιγράφουν τη λειτουργία των Υπηρεσιών.
- ✘ **WS-Policy:** Καθορίζει τον τρόπο με τον οποίο οι διάφορες πολιτικές ανακηρύσσονται και ανταλλάσσονται μεταξύ των συνεργαζόμενων Διαδικτυακών Υπηρεσιών (Web Services).
- ✘ **WS-Eventing [43]:** Ορίζει ένα μοντέλο στηριζόμενο στην ύπαρξη συμβάντων (event model) για ασύγχρονη και σύγχρονη επικοινωνία.

Οι συγκεκριμένες προδιαγραφές αποτελούν το λεγόμενο «τεχνολογικό υπόβαθρο των Διαδικτυακών Υπηρεσιών» (Web Services Technology Stack) και είναι απαραίτητες για την πλήρη και ασφαλή χρήση των Διαδικτυακών Υπηρεσιών σε επιχειρησιακές εφαρμογές.

Λόγω της ευελιξίας, της εφαρμογής της διαλειτουργικότητας και των υπόλοιπων χαρακτηριστικών που εμφανίζουν οι Διαδικτυακές Υπηρεσίες (Web Services) θεωρούνται ως η καταλληλότερη τεχνολογία για να έκθεση της λειτουργικότητας των επιχειρηματικών διαδικασιών ως Υπηρεσίες και επομένως αποτελούν την καταλληλότερη τεχνολογία για την πραγμάτωση της SOA αρχιτεκτονικής. Εξαιτίας της ευρείας υποστήριξης που έχουν από τους κατασκευαστές λογισμικού, οι Διαδικτυακές Υπηρεσίες παρέχουν τη δυνατότητα να χρησιμοποιηθεί η ίδια

τεχνολογία για να αναπαρασταθούν Υπηρεσίες που η υλοποίησή τους περιλαμβάνει εφαρμογές, οι οποίες εκτείνονται από εφαρμογές legacy συστημάτων μέχρι τις σύγχρονες multi-tier (πολύ-επίπεδες) εφαρμογές.

2.2.2 Διαχείριση Επιχειρηματικών Διαδικασιών BPM (Business Process Management)

Η διαχείριση επιχειρηματικών διαδικασιών ή business process management αποτελεί ένα σημαντικό σύγχρονο πεδίο έρευνας τόσο από πρακτική σκοπιά λόγω της επιχειρηματικής χρησιμότητας που παρουσιάζει στο σχεδιασμό των επιχειρησιακών διαδικασιών και εφαρμογών, όσο και ως εργαλείο ανάπτυξης και σχεδίασης λογισμικού για την ανάπτυξη επιχειρηματικών εφαρμογών.

Η διαχείριση επιχειρηματικών διαδικασιών στηρίζεται στην παρατήρηση ότι κάθε προϊόν ή υπηρεσία που παρέχει κάθε επιχείρηση στην αγορά είναι αποτέλεσμα μιας σειράς εκτέλεσης επιμέρους δραστηριοτήτων [6]. Οι επιχειρηματικές διαδικασίες αποτελούν το εργαλείο-κλειδί για την οργάνωση των παραπάνω επιμέρους δραστηριοτήτων και τη βελτίωση της κατανόησης των σχέσεων μεταξύ τους. Η σύγχρονη επιχείρηση προκειμένου να επιτύχει τους στρατηγικούς της στόχους, είναι αναγκαίο να οργανώσει και να οδηγήσει τους επιχειρηματικούς και υπολογιστικούς της πόρους σε συνεργασία, γεγονός στο οποίο αποφασιστική συμβολή έχουν οι επιχειρηματικές διαδικασίες. Η ευρύτατη εξάπλωση της τεχνολογίας των ενδοδικτύων (intranets) και του διαδικτύου (internet) καθιστά τον ανταγωνισμό ιδιαίτερα απαιτητικό και την ανάγκη για άμεση προσαρμογή στις επιταγές της αγοράς το σύγχρονο ζητούμενο, με αποτέλεσμα οι παραδοσιακοί κύκλοι παραγωγής προϊόντων να μην επαρκούν να περιγράψουν το πλαίσιο της δυναμικότητας των σύγχρονων καταστάσεων της οικονομίας. Γίνεται, λοιπόν, φανερό ότι η διαχείριση επιχειρηματικών διαδικασιών έρχεται για να περιορίσει το χάσμα μεταξύ της οργανωσιακής επιχειρησιακής πλευράς και των πληροφοριακών συστημάτων στα οποία στηρίζεται σε μεγάλο βαθμό η επιτυχία των σύγχρονων επιχειρήσεων.

Επομένως, ως διαχείριση επιχειρηματικών διαδικασιών μπορεί να οριστεί ένα σύνολο από δραστηριότητες που εκτελούνται σε συντονισμό και συνεργασία στα πλαίσια ενός τεχνικού και οργανωσιακού περιβάλλοντος. Οι συγκεκριμένες δραστηριότητες από κοινού πραγματοποιούν έναν επιχειρηματικό στόχο. Η διαχείριση επιχειρηματικών διαδικασιών περιλαμβάνει ιδέες, μεθόδους και τεχνικές που χρησιμοποιούνται για το σχεδιασμό, τη διαχείριση, τη διαμόρφωση και την ανάλυση των επιχειρηματικών διαδικασιών.

Κυρίαρχο ρόλο στη διαχείριση των επιχειρηματικών διαδικασιών διαδραματίζει η χρήση μεθόδων μοντελοποίησης των επιχειρηματικών διαδικασιών (business process modeling) που στηρίζονται στην ανάπτυξη γραφικών εργαλείων και σημειογραφίας των βασικών παραγόντων που συνθέτουν τις επιχειρηματικές διαδικασίες. Παράλληλα, η διαχείριση των επιχειρηματικών διαδικασιών στηρίζεται στην ανάπτυξη και την κατάστρωση ροών δεδομένων και στην εφαρμογή μεθόδων διαχείρισης διαγραμμάτων ροής εργασιών (workflow management).

Προϊόντα που έχουν αναπτυχθεί στα πλαίσια της τεχνικής του BPM είναι: IBM's WebSphere, HP's HP Process Manager, BEA's WebLogic, και Vitria's

BusinessWare, JBoss jBPM, Microsoft BizTalk (περιοριζόμενο μόνο σε εφαρμογές Microsoft Windows and .NET servers).

2.2.3 BPEL (Business Process Execution Language)

2.2.3.1 ΕΙΣΑΓΩΓΗ

Ως επιχειρηματική διαδικασία έχει αναφερθεί ότι ορίζεται μια διαδικασία που σχεδιάζει μια επιχείρηση και αποτελεί μέρος ενός μεγαλύτερου επιχειρηματικού σκοπού προκειμένου να διευκολύνει και να καθορίσει τη ροή δεδομένων και τη λήψη αποφάσεων. Ουσιαστικά, αποτελεί μια σειρά από ατομικές αυτόνομες δραστηριότητες με κάθε δραστηριότητα να εκτελείται με καθορισμένη σειρά. Η αυτοματοποίηση των επιχειρηματικών διαδικασιών έχει υιοθετηθεί ως απαραίτητο στοιχείο της οργάνωσης των σύγχρονων επιχειρήσεων και επομένως, η απαίτηση για μια πρότυπη θεμελίωση με τη μέθοδο μιας εξειδικευμένης προγραμματιστικής γλώσσας για τη σύνθεση των δραστηριοτήτων φαντάζει ως σημαντικό ζητούμενο. Προς τη συγκεκριμένη κατεύθυνση και υπό το πλαίσιο της υιοθέτησης της SOA αρχιτεκτονικής ως τη σύγχρονη σχεδιαστική λύση για την προώθηση της ευελιξίας στις επιχειρηματικές διαδικασίες, αναδύθηκε η BPEL ως η κυρίαρχη πρότυπη γλώσσα σύνθεσης υπηρεσιών μέσα από έναν καλά ορισμένο τρόπο και μεθοδολογία. Ο πρωταρχικός στόχος, λοιπόν, της BPEL είναι η προτυποποίηση της διαδικασίας αυτοματοποίησης μεταξύ των Διαδικτυακών Υπηρεσιών.

Συνεπώς είναι εμφανές ότι με τη συμβολή της BPEL δίνεται η δυνατότητα να οριστούν επιχειρηματικές διαδικασίες που κάνουν χρήση των Υπηρεσιών για τη σύνθεση μεγαλύτερων διαδικασιών και επιχειρηματικές διαδικασίες που εξωτερικεύουν τη λειτουργικότητα τους ως Υπηρεσίες. Εντός των επιχειρήσεων, η BPEL χρησιμοποιείται για την τυποποίηση και την ολοκλήρωση επιχειρησιακών εφαρμογών και για την επέκταση της ενοποίησης σε προηγούμενως απομονωμένα συστήματα. Μεταξύ των επιχειρήσεων, η BPEL επιτρέπει ευκολότερη και περισσότερο αποτελεσματική ολοκλήρωση και ενοποίηση με τους επιχειρηματικούς συνεργάτες. Η BPEL υποκινεί τις επιχειρήσεις να καθορίσουν περαιτέρω τις διαδικασίες τους, με αποτέλεσμα να οδηγηθούν στην βελτιστοποίηση των επιχειρηματικών διαδικασιών και της επιχειρησιακής οργάνωσης, στον επανασχεδιασμό των λειτουργιών τους (reengineering) και στην επιλογή των καταλληλότερων κάθε φορά διαδικασιών. Η BPEL αποτελεί τεχνολογία κλειδί σε επιχειρησιακά περιβάλλοντα όπου η επιχειρηματική λειτουργικότητα εκφράζεται ή εκτίθενται ως Διαδικτυακές Υπηρεσίες.

2.2.3.2 Χαρακτηριστικά

Με την BPEL μπορούμε να ορίσουμε τόσο απλές όσο και πολύπλοκες διαδικασίες. Ως προγραμματιστική γλώσσα εμφανίζει χαρακτηριστικά κοινά με τις παραδοσιακές γλώσσες προγραμματισμού παρέχοντας στοιχεία όπως βρόχους, κόμβους αποφάσεων, διακλαδώσεις, μεταβλητές που επιτρέπουν τον ορισμό επιχειρηματικών διαδικασιών με αλγοριθμικό τρόπο. Η BPEL είναι μια τυποποιημένη γλώσσα που εστιάζει στον ορισμό επιχειρηματικών διαδικασιών με τη χρήση σημαντικών στοιχείων

που σχετίζονται με την επίκληση των Διαδικτυακών Υπηρεσιών. Η BPEL επιτρέπει την κλήση των λειτουργιών των Διαδικτυακών Υπηρεσιών είτε με ασύγχρονο είτε με σύγχρονο τρόπο. Επίσης, επιτρέπει την κλήση λειτουργιών είτε σε σειρά είτε παράλληλα, ενώ δίνεται η δυνατότητα διαχείρισης σφαλμάτων με εύχρηστο και καλά τεκμηριωμένο τρόπο. Παράλληλα, η BPEL στηρίζεται σε XML μορφή κώδικα και παρέχει υποστήριξη για μεγάλης διάρκειας διαδικασίες και αποζημίωση των δραστηριοτήτων σε περίπτωση ασυνεπειών. Ενδεικτικά, αναφέρουμε τα παρακάτω χαρακτηριστικά που προσφέρει η BPEL:

- ✘ Περιγραφή της λογικής των επιχειρηματικών διαδικασιών μέσα από την σύνθεση των Υπηρεσιών.
- ✘ Σύνθεση μεγαλύτερων επιχειρηματικών διαδικασιών από μικρότερες διαδικασίες και Υπηρεσίες.
- ✘ Κλήση και διαχείριση τόσο σύγχρονων όσο και ασύγχρονων λειτουργιών των Υπηρεσιών.
- ✘ Κλήση των λειτουργιών των Υπηρεσιών τόσο σε σειρά όσο και παράλληλα
- ✘ Επιλεκτική αποζημίωση και ακύρωση ολοκληρωμένων δραστηριοτήτων σε περίπτωση προγραμματιστικών σφαλμάτων.
- ✘ Συντήρηση πολλαπλών και χρονοβόρων σε διάρκεια εκτέλεσης δραστηριοτήτων συναλλαγών.
- ✘ Διαχείριση γεγονότων σχετιζόμενων τόσο με την αναμονή ύπαρξης μηνύματος πυροδότησης όσο και με την ύπαρξη χρονομέτρου που υπολογίζει απαιτούμενους χρόνους παρέλευσης.
- ✘ Δόμηση των επιχειρηματικών διαδικασιών σε διαφορετικά επίπεδα αφαιρετικότητας και φάσματος.
- ✘ Παράλληλη εκτέλεση δραστηριοτήτων και καθορισμός του τρόπου με τον οποίο συγχρονίζονται και συγχέουν τη λειτουργία τους.
- ✘ Προγραμματισμός των δραστηριοτήτων με βάση τον χρόνο εκτέλεσης και καθορισμός της σειράς εκτέλεσης τους.
- ✘ Συσχέτιση των αιτημάτων (requests) εντός των επιχειρηματικών διαδικασιών.
- ✘ Δρομολόγηση εισερχόμενων μηνυμάτων στα κατάλληλες διαδικασίες και δραστηριότητες.
- ✘ Διατήρηση διαδικασιών που έχουν διακοπεί λόγω σφάλματος ως το σημείο που ορθής εκτέλεσης, ώστε να περιοριστεί η άσκοπη ενασχόληση με περιττές εργασίες που δεν παρουσιάζουν σφάλματα.

2.2.3.3 Σύνθεση Υπηρεσιών (*Service Composition*)

Στην SOA αρχιτεκτονική, προκειμένου να πετύχουμε το βέλτιστο δυνατό αποτέλεσμα στην οργάνωση και τον σχεδιασμό των επιχειρηματικών διαδικασιών και να οδηγηθούμε στην επαναχρησιμοποίηση των Υπηρεσιών, δημιουργείται η ανάγκη για τη σύνθεση Υπηρεσιών από ήδη υπάρχουσες Υπηρεσίες. Συνεπώς για την περιγραφή και την πραγματοποίηση πολύπλοκων επιχειρηματικών λειτουργιών και ανάλογα με τις απαιτήσεις και το σχεδιαστικό πρότυπο που πρόκειται να ακολουθηθεί, υπάρχει η δυνατότητα σύνθεσης των Υπηρεσιών με βάση δύο βασικές προσεγγίσεις οργάνωσης της ροής των συναλλαγών που θα αναλυθούν εκτενώς σε επόμενη ενότητα και αναφέρονται ως [38]:

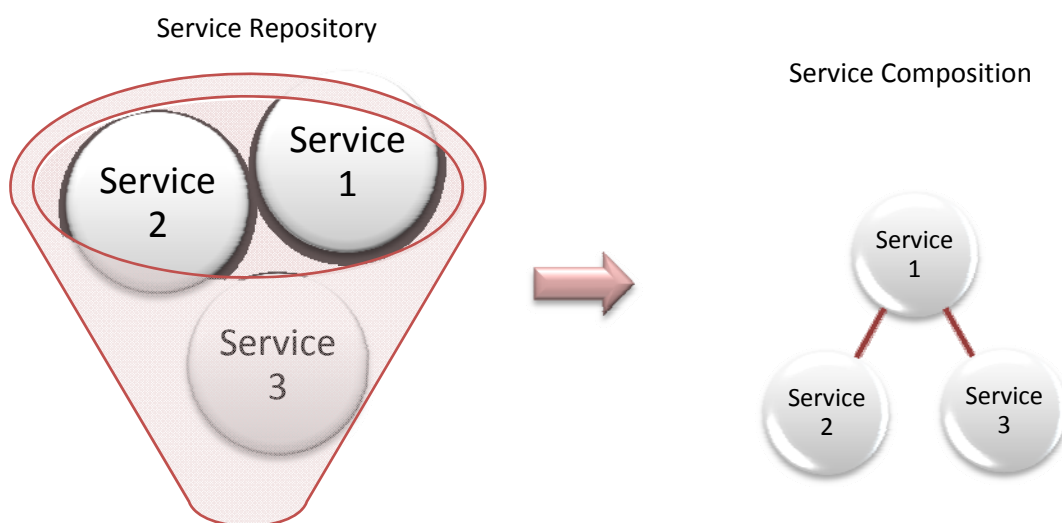
- ✘ **Ενορχήστρωση**

✘ Χορογραφία

Σε μια προσπάθεια ορισμού της έννοιας της σύνθεσης Υπηρεσιών, έχει προταθεί μεγάλος αριθμός διαφορετικών προσεγγίσεων [36 39 20 29 9] χωρίς να υπάρχει ένας ικανοποιητικός ορισμός που να καλύπτει πλήρως το εύρος των επιχειρηματικών διαδικασιών. Η καλύτερη προσέγγιση για το σχεδιασμό μιας επιχειρηματικής διαδικασίας είναι η ανάλυση του ρόλου της και ο περιορισμός της προσοχής στο επίπεδο της επιχείρησης ως σύνολο. Πρέπει να αναλυθεί με βάση διάφορες όψεις της επιχειρηματικής δραστηριότητας, όπως με βάση την ανταλλαγή των δεδομένων, το διαχωρισμό των βασικών λειτουργικών μονάδων, την οργάνωση της ροής και την διαχείριση των αναγκαίων πόρων [1]. Στο [3] γίνεται διαχωρισμός μεταξύ της λειτουργικής, της οργανωσιακής και της πληροφοριακής οπτικής γωνίας των επιχειρηματικών διαδικασιών. Στο [4] υιοθετείται το πρότυπο και πλαίσιο ανάλυσης της επιχειρηματικής διαδικασίας που προτείνεται στο [24] και διακρίνει τα βασικά στοιχεία της επιχειρηματικής διαδικασίας με βάση τις όψεις του *πώς*, *γιατί*, *πότε*, *ποιός*, *τι* και *που*, με αποτέλεσμα να αποδομείται η επιχειρηματική διαδικασία στα βασικά χαρακτηριστικά της βάσει των απαντήσεων στις παραπάνω ερωτήσεις.

Τελικά, μπορούμε να αναγνωρίσουμε τα παρακάτω βασικά χαρακτηριστικά των επιχειρηματικών διαδικασιών :

- ✘ **Δραστηριότητα (Activity):** Μια δραστηριότητα αντιπροσωπεύει μια καλά ορισμένη επιχειρησιακή λειτουργία και αποτελεί μέρος της όψης του *πώς*. Πρόκειται για το στοιχείο της επιχειρηματικής διαδικασίας που υπαγορεύει τον τρόπο υλοποίησης και κλήσης μιας δράσης – ενέργειας.
- ✘ **Συνθήκη (Condition):** Η συμπεριφορά της επιχειρηματικής διαδικασίας ελέγχεται και οδηγείται από την επιχειρηματική λογική και τους κανόνες. Επομένως η συνθήκη είναι το χαρακτηριστικό της επιχειρηματικής διαδικασίας που αποτιμάται και ελέγχεται, ώστε να αποφασιστεί η ροή ελέγχου της διαδικασίας. Αντιστοιχεί στην όψη του *γιατί*.

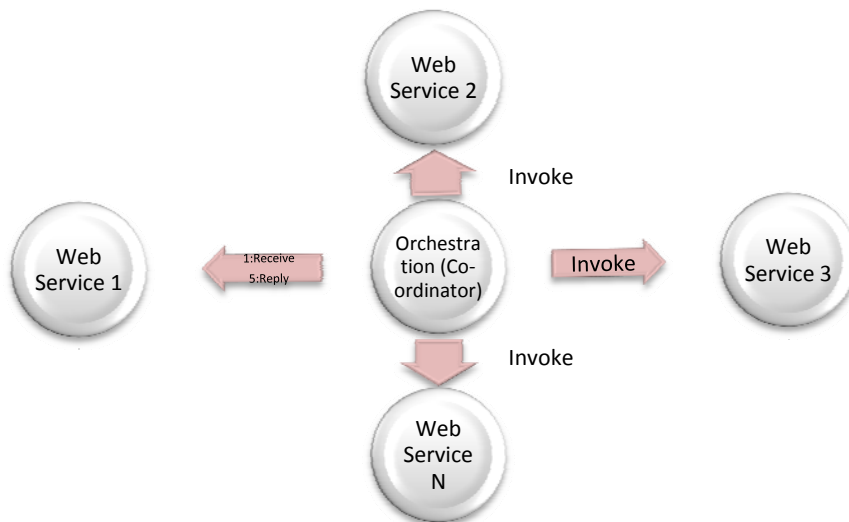


Σχήμα 6-11: Σύνθεση Υπηρεσιών

- ✘ **Γεγονός – Συμβάν (Event):** Τα γεγονότα κατά τη σύνθεση των Υπηρεσιών αντιστοιχούν σε πραγματικά επιχειρησιακά συμβάντα που πραγματοποιούνται και μεταβάλλουν ή καθορίζουν τη ροή της επιχειρηματικής διαδικασίας. Αποτελούν μέρος της όψης του *πότε*, γιατί υπαγορεύουν το χρονικό σημείο πυροδότησης ενός γεγονότος.
- ✘ **Ροή (Flow):** Η ροή εκφράζει και αυτή την όψη του *πώς* και χρησιμοποιείται για να καταδείξει την οργάνωση και την ενορχήστρωση των συμμετεχόντων δραστηριοτήτων. Πρόκειται για τους συνήθεις τύπους - πρότυπα επιλογής ροής, δηλαδή σειριακή - ακολουθιακή ροή, παράλληλη ροή, ροή με βάση συνθήκη, επαναληπτική ροή και συνδυασμός των παραπάνω.
- ✘ **Μήνυμα (Message):** Μηνύματα χρησιμοποιούνται προκειμένου να επιτύχουμε την ανταλλαγή πληροφοριών μεταξύ των διαφορετικών συναλλασσόμενων. Τα μηνύματα αντιπροσωπεύουν την όψη του *τι*, καθώς εκφράζουν το είδος των δεδομένων και τις εξαρτήσεις και αλληλεπιδράσεις που υφίστανται.
- ✘ **Παροχέας (Provider):** Οι συμμετέχοντες – άνθρωποι και υπολογιστικοί πόροι – που αποτελούν μέλη της επιχειρηματικής διαδικασίας απεικονίζονται με τη μορφή του παροχέα. Ο παροχέας ανήκει την όψη του *ποιος* και *που* και πρόκειται για μια συμπαγή Υπηρεσία.
- ✘ **Ρόλος (Role):** Οι ρόλοι αποτελούν τμήμα της όψης του *ποιός* και καθορίζουν την αναμενόμενη συμπεριφορά των συμμετεχόντων στην επιχειρηματική διαδικασία με ένα αφαιρετικό τρόπο.

2.2.3.3.1 Ενορχήστρωση και Χορογραφία

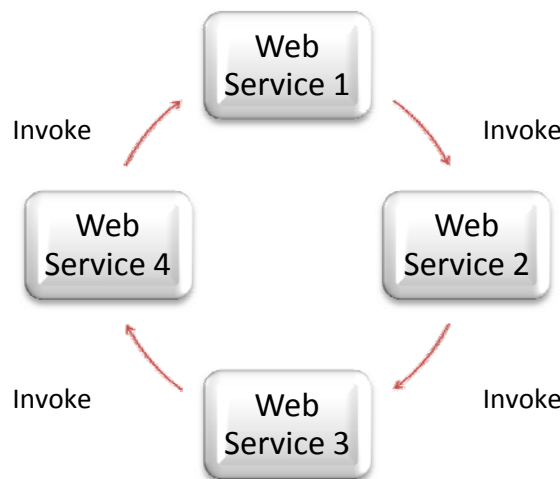
Στην περίπτωση οργάνωσης της ροής των επιχειρηματικών διαδικασιών με βάση το πρότυπο της ενορχήστρωσης, μια κεντρική διαδικασία αναλαμβάνει τον συνολικό έλεγχο και τη διαχείριση των υπηρεσιών που λαμβάνουν μέρος, και συντονίζει την εκτέλεση των διαφορετικών λειτουργιών της κάθε υπηρεσίας. Πρόκειται, με άλλα λόγια, για μια προσέγγιση όπου η διαλειτουργικότητα των εφαρμογών επιτυγχάνεται βάσει μιας κεντρικά ελεγχόμενης ροής εργασιών (workflow). Οι Διαδικτυακές Υπηρεσίες που συμμετέχουν δε γνωρίζουν ότι είναι μέλη μιας επιχειρηματικής διαδικασίας και ότι αποτελούν αναπόσπαστο τμήμα της ροής μιας υψηλότερου επιπέδου υπηρεσίας. Μόνο η υπηρεσία που δρα ως κεντρικός διαχειριστής και συντονιστής γνωρίζει τη συμμετοχή και το ρόλο που διαδραματίζει για την πραγμάτωση της επιχειρηματικής διαδικασίας, κατά τέτοιο τρόπο ώστε η ενορχήστρωση να οργανώνεται και να μορφοποιείται γύρω από αυτήν με σαφή καθορισμό των λειτουργιών και της σειράς με την οποία πρόκειται να κληθούν οι Διαδικτυακές Υπηρεσίες.



Σχήμα 6-12: Ενορχήστρωση

Η προσέγγιση της ενορχήστρωσης προσφέρει ένα περιβάλλον ενοποίησης με διάφορες εφαρμογές ενός οργανισμού έστω και αν αυτές βασίζονται σε διαφορετικές πλατφόρμες. Τέλος, υπάρχει η δυνατότητα μεταβολής ή επέκτασης της λογικής της ροής εργασίας, διευκολύνεται η συγχώνευση των επιχειρηματικών διαδικασιών και εξωτερικεύεται η συμπεριφορά της διαδικασίας ως υπηρεσία με καθορισμένο περιεχόμενο.

Στην περίπτωση οργάνωσης της ροής των επιχειρηματικών διαδικασιών με βάση το πρότυπο της χορογραφίας, δεν υπάρχει κεντρική διαδικασία που να αναλαμβάνει τον έλεγχο της ροής εκτέλεσης των υπηρεσιών, αλλά κάθε υπηρεσία είναι υπεύθυνη από μόνη της να προσδιορίσει και να εκτελέσει το περιεχόμενο της και να συντονίσει τη λειτουργία της με βάση τη ροή της διαδικασίας. Με άλλα λόγια, κάθε συμμετέχων παράγοντας της επιχειρηματικής διαδικασίας γνωρίζει με ακρίβεια τη στιγμή που θα εκτελέσει τη λειτουργία της και τους διαφορετικούς παράγοντες με τους οποίους αλληλεπιδρά. Μια τυπική προσέγγιση που εμφανίζει τον χαρακτήρα της χορογραφίας, αποτελεί η υλοποίηση αλυσίδας διαδικασιών με ακολουθιακή πορεία και καθορισμένη σειρά εκτέλεσης.



Σχήμα 6-13: Χορογραφία

Η προσέγγιση της χορογραφίας αποτελεί μια προσπάθεια συνεργασίας που επικεντρώνεται στην ανταλλαγή μηνυμάτων μεταξύ επιχειρηματικών διαδικασιών, για την εκτέλεση των οποίων έχει συμφωνηθεί η ροή εργασίας που θα ακολουθηθεί και έχει εγκατασταθεί ένα πλαίσιο συνεργασίας μεταξύ των υπηρεσιών που συμμετέχουν. Υπό ιδανικές συνθήκες, η μέθοδος της χορογραφίας μπορεί να εφαρμοστεί σε δημόσιο και διεπιχειρησιακό επίπεδο, όπου οι οργανισμοί θα μπορούσαν να συμφωνήσουν πάνω στη δομή των εσωτερικών διεργασιών τους προκειμένου να διαλειτουργήσουν και να επιτύχουν αυτή την επιδίωξή τους βασιζόμενες σε αυτοματοποιημένες λύσεις. Ειδικότερα στη σημερινή εποχή όπου οι απαιτήσεις διαλειτουργικότητας είναι ιδιαίτερα αυξημένες μια πληθώρα υπηρεσιών διαφορετικών επιχειρήσεων ζητούν τρόπο συνεργασίας. Ένα αντιπροσωπευτικό παράδειγμα προτύπου που επιδιώκει την οργάνωση της ανταλλαγής πληροφοριών μεταξύ πολλαπλών οργανισμών είναι το WS-CDL[44 45].

Από την οπτική της σύνθεσης Διαδικτυακών Υπηρεσιών για την εκτέλεση επιχειρηματικών διαδικασιών, η προσέγγιση της ενορχήστρωσης, ως περισσότερο ευέλικτο σχήμα οργάνωσης της ροής εκτέλεσης, παρουσιάζει πλεονεκτήματα έναντι της χορογραφίας που συνοψίζονται στα εξής :

- ✘ Είναι γνωστός ο παράγοντας που είναι υπεύθυνος για το συντονισμό και την εκτέλεση ολόκληρης της επιχειρηματικής διαδικασίας.
- ✘ Υπάρχει η δυνατότητα εναλλαγής και εναλλακτικής μετατροπής της ροής σε περίπτωση σφάλματος.
- ✘ Υπάρχει η δυνατότητα συγχώνευσης και ενσωμάτωσης υπηρεσιών χωρίς να γνωρίζουν ότι αποτελούν τμήμα επιχειρηματικής διαδικασίας.

2.2.3.4 Εκτελέσιμες και Αφαιρετικές Διαδικασίες (Executable and Abstract Processes)

Με τη βοήθεια της BPEL, μπορούμε να διακρίνουμε τις επιχειρηματικές διαδικασίες σε δύο κατηγορίες :

- ✘ Στις επιχειρηματικές διαδικασίες, στις οποίες μπορούμε να καθορίσουμε με ακρίβεια τις λεπτομέρειες σχεδιασμού και υλοποίησης τους και ονομάζονται **εκτελέσιμες επιχειρηματικές διαδικασίες** (executable business processes). Οι εκτελέσιμες επιχειρηματικές διαδικασίες ακολουθούν το πρότυπο της ενορχήστρωσης.
- ✘ Στις επιχειρηματικές διαδικασίες, στις οποίες μπορούμε να καθορίσουμε και να παρακολουθήσουμε την ανταλλαγή των μηνυμάτων που πραγματοποιείται μεταξύ των συνεργαζόμενων παραγόντων και ονομάζονται **αφαιρετικές επιχειρηματικές διαδικασίες**. Οι αφαιρετικές επιχειρηματικές διαδικασίες ακολουθούν το πρότυπο της χορογραφίας και δεν περιλαμβάνουν εσωτερικές λεπτομέρειες υλοποίησης, ενώ συγχρόνως είναι αδύνατο να εκτελεστούν.

Οι εκτελέσιμες επιχειρηματικές διαδικασίες είναι διαδικασίες που συνθέτουν και συνδυάζουν ένα σύνολο υπαρχουσών υπηρεσιών, ενώ παράλληλα καθορίζουν τον ακριβή αλγόριθμο υλοποίησης και εξάρτησης των δραστηριοτήτων και ανταλλαγής

των μηνυμάτων εισόδου και εξόδου. Για την εκτέλεση τους χρησιμοποιούνται μηχανές BPEL (BPEL engines). Η χρησιμότητα τους είναι προφανής και έγκειται στο γεγονός ότι αποτελούν την άμεση απάντηση στο πρόβλημα της αυτοματοποίησης των επιχειρηματικών διαδικασιών μέσα από προγραμματιστικές μεθόδους λογισμικού με απλό και ευθύ προσανατολισμό. Οι εκτελέσιμες διαδικασίες συμπληρώνουν το χάσμα μεταξύ της προδιαγραφής της διαδικασίας και του προγραμματιστικού κώδικα που είναι υπεύθυνος για την εκτέλεση τους.

Κατά τον καθορισμό των εκτελέσιμων επιχειρηματικών διαδικασιών, ουσιαστικά καθορίζουμε μια καινούργια διαδικτυακή Υπηρεσία που αποτελεί σύνθεση υπαρχουσών Υπηρεσιών. Η διεπαφή (interface) της νέας διαδικτυακής Υπηρεσίας χρησιμοποιεί ένα σύνολο από θύρες (ports) μέσω των οποίων παρέχει τις λειτουργίες της στις υπόλοιπες Διαδικτυακές Υπηρεσίες.

Όσον αφορά τις αφαιρετικές επιχειρηματικές διαδικασίες, έχει αναφερθεί ήδη ότι δεν είναι εκτελέσιμες, οπότε περιορίζονται σε απλή καταγραφή και προδιαγραφή των μηνυμάτων που ανταλλάσσονται μεταξύ των συμμετεχόντων μελών. Τα μηνύματα περιγράφουν τη συμπεριφορά του μέλους που συμμετέχει και είναι τα μόνα παρατηρήσιμα στοιχεία της Υπηρεσίας. Επομένως, γίνεται φανερό ότι οι αφαιρετικές επιχειρηματικές διαδικασίες δεν εμφανίζουν ιδιαίτερη χρησιμότητα. Παρόλα αυτά καθορίζονται για δύο λόγους :

- ✖ Για να περιγράψουν τη συμπεριφορά μιας Υπηρεσίας χωρίς να είναι γνωστό με ακρίβεια το σύνολο των επιχειρηματικών διαδικασιών στις οποίες συμμετάσχει.
- ✖ Για να καθοριστούν τα πρωτόκολλα συνεργασίας μεταξύ των πολλαπλών παραγόντων που συμμετέχουν και να περιγραφεί με ακρίβεια η εξωτερική συμπεριφορά του κάθε συμμετέχοντα.
- ✖ Για να αποτελέσουν τα πρότυπα με βάση τα οποία είναι δυνατόν να σχεδιαστούν οι εκτελέσιμες επιχειρηματικές διαδικασίες.

3 Επιχειρηματικοί Κανόνες (Business Rules)

3.1 Εισαγωγή

Οι σύγχρονες επιχειρήσεις, στην προσπάθειά τους να αντιμετωπίσουν την πολυπλοκότητα των επιχειρησιακών εφαρμογών λογισμικού και των πληροφοριακών συστημάτων και υποδομών, υιοθετούν στον αρχιτεκτονικό σχεδιασμό τη λύση της SOA αρχιτεκτονικής που διευκολύνει την ανάπτυξη επιχειρησιακών εφαρμογών και δραστηριοτήτων με την αποδοχή των επιχειρηματικών υπηρεσιών που είναι λειτουργικές μονάδες λογισμικού με συγκεκριμένη επιχειρηματική λογική. Οι Υπηρεσίες εμφανίζουν τη δυνατότητα της ενοποίησης – ολοκλήρωσης και της επαναχρησιμοποίησης, με αποτέλεσμα να οδηγούν σε ευέλικτες και προσαρμοζόμενες σε μεταβολές αρχιτεκτονικές λύσεις. Παράλληλα, η ενορχήστρωση των Υπηρεσιών και η συνεργασία τους για την δημιουργία σύνθετων επιχειρηματικών λειτουργιών με τη χρήση κοινά αποδεκτών τεχνολογιών διαχείρισης επιχειρηματικών διαδικασιών, όπως η BPEL, έχει συμβάλει στην ανάπτυξη μοντέλων διαδικασιών που μπορούν να επαναχρησιμοποιηθούν και να μεταβάλουν τη συμπεριφορά τους και την επιχειρηματική λογική ως απόκριση νέων επιχειρηματικών απαιτήσεων και προκλήσεων και καθιστούν δυνατή και υλοποιήσιμη την παρατήρηση και διαχείριση της συμπεριφοράς των διαδικασιών σε πραγματικό χρόνο (real-time visibility and management).

Η SOA αρχιτεκτονική υπόσχεται απλοποίηση στην ενοποίηση δεδομένων και εφαρμογών, αυξημένη επαναχρησιμοποίηση, μεγαλύτερη ευελιξία και μειωμένο κίνδυνο και κόστος σε περίπτωση μεταβολών. Η SOA αρχιτεκτονική και η διαχείριση των επιχειρηματικών λειτουργιών (BPM) παρέχουν τη δυνατότητα ευελιξίας μέσα από την υιοθέτηση των παρακάτω κατευθύνσεων :

- ✘ Μειώνουν το χρονικό διάστημα που απαιτείται για την αυτοματοποίηση των επιχειρηματικών διαδικασιών μέσα από την μείωση του κενού - χάσματος μεταξύ της μοντελοποίησης και της υλοποίησης και βελτιώνουν την επαναχρησιμοποίηση των ήδη υφιστάμενων πόρων που μπορούν να αναπτυχθούν ως Υπηρεσίες και στη συνέχεια να αποτελέσουν μέλος διαφορετικών επιχειρηματικών διαδικασιών.
- ✘ Επιτρέπουν στις επιχειρηματικές διαδικασίες που έχουν ήδη υλοποιηθεί ως ενορχήστρωση υφιστάμενων Υπηρεσιών να μεταβάλλονται άμεσα.
- ✘ Ελευθερώνουν την οικονομική δραστηριότητα της επιχείρησης από την ανάγκη για συνεχή επένδυση οικονομικών πόρων σε μεταβολές των πληροφοριακών συστημάτων μέσα από τη συντήρηση και επαναχρησιμοποίηση.

Από την άλλη πλευρά όμως, οι επιχειρηματικές διαδικασίες αποτελούνται από μια σειρά από σημεία αποφάσεων (decision points) στα οποία πραγματοποιείται η αποτίμηση σημαντικών κριτηρίων και τα οποία αποτελούν μέρος του κώδικα, δηλαδή είναι ενσωματωμένα μέσα στην υλοποίηση της διαδικασίας. Ως αποτέλεσμα οι επιχειρηματικοί κανόνες που συνοψίζονται στην ύπαρξη σημείων αποφάσεων οδηγούν στην ουσία τη συμπεριφορά της επιχειρηματικής διαδικασίας και όντας τμήματα προγραμματιστικού κώδικα προκαλούν ορισμένα προβλήματα, όπως τα παρακάτω:

- ✘ Οι επιχειρηματικοί κανόνες (business rules) και η επιχειρηματική λογική που ελέγχουν τη ροή της διαδικασίας μεταβάλλονται συνεχώς και με γρηγορότερο ρυθμό από τις επιχειρηματικές διαδικασίες, με αποτέλεσμα να απαιτείται η συνεχής επέμβαση στον προγραμματιστικό κώδικα για τη μεταβολή της πολιτικής και των κανόνων. Όπως γίνεται φανερό, οι επιχειρηματικοί αναλυτές (business analysts) πρέπει αναγκαστικά να βασίζονται στις αποφάσεις τους σε μεγάλο βαθμό από τα πληροφοριακά συστήματα, καθώς αδυνατούν να επέμβουν στον κώδικα και να τον μεταβάλλουν, ενώ συγχρόνως η συνεχής απασχόληση προγραμματιστών στις μεταβολές των επιχειρηματικών κανόνων δεσμεύει οικονομικούς και χρονικούς πόρους.
- ✘ Οι περισσότερες επιχειρήσεις εμφανίζουν έλλειψη ενός κεντρικού ταμειυτήρα κανόνων (rule repository), με αποτέλεσμα οποιαδήποτε ευρεία μεταβολή στην πολιτική της επιχείρησης να δημιουργεί προβλήματα καθολικής εφαρμογής και να είναι αδύνατο να εφαρμοστεί δυναμικά και να δημιουργήσει δυναμικό χαρακτήρα στις ροές δεδομένων.
- ✘ Οι επιχειρηματικές διαδικασίες δε μπορούν να επαναχρησιμοποιήσουν τους ήδη υπάρχοντες κανόνες, με αποτέλεσμα να απαιτείται ο συνεχής σχεδιασμός κανόνων για κάθε διαδικασία, οδηγώντας σε ασυνέπεια, αστάθεια και περιττή δέσμευση πόρων του πληροφοριακού συστήματος.

Συνεπώς, υφίσταται η ανάγκη για χρησιμοποίηση μιας μηχανής επιχειρηματικών κανόνων (business rules engine) προκειμένου να απελευθερώσει την ενσωμάτωση των σημείων αποφάσεων μέσα στον κώδικα της επιχειρηματικής διαδικασίας και να συμβάλει στην ανάπτυξη ευελιξίας. Παρακάτω στο κεφάλαιο της υλοποίησης του πρωτοτύπου θα εξετάσουμε τις τρεις διαφορετικές προσεγγίσεις στην χρήση επιχειρηματικής λογικής μέσα από μια μελέτη περίπτωσης και θα καταλήξουμε σε αρχιτεκτονικές και προγραμματιστικές προσεγγίσεις για την ανάπτυξη επιχειρηματικών κανόνων μέσα από το πρίσμα και το πλαίσιο της SOA αρχιτεκτονικής. Πρόκειται για προσεγγίσεις που αφορούν :

- ✘ Την ενσωμάτωση των επιχειρηματικών κανόνων στον προγραμματιστικό κώδικα (code-based approach) που παρουσιάζει σημαντικά προβλήματα ευελιξίας και επαναχρησιμοποίησης, όπως ειπώθηκε παραπάνω.
- ✘ Την οδηγούμενη από μοντέλα σχεδιασμού επιχειρηματικών διαδικασιών προσέγγιση (model-driven approach) που στηρίζεται στα εργαλεία διαχείρισης επιχειρηματικών διαδικασιών και συνοψίζεται στη χρήση σημείων αποτίμησης και αποφάσεων (decision points).
- ✘ Την προσανατολισμένη στις υπηρεσίες προσέγγιση που συνδυάζει τα χαρακτηριστικά της SOA αρχιτεκτονικής και μιας μηχανής εκτέλεσης κανόνων (business rules engine) που αποτελεί το ζητούμενο και τον στόχο της συγκεκριμένης διπλωματικής, παρέχοντας ευελιξία, επαναχρησιμοποίηση και συντηρησιμότητα.

3.2 Επιχειρηματικοί Κανόνες (Business rules)

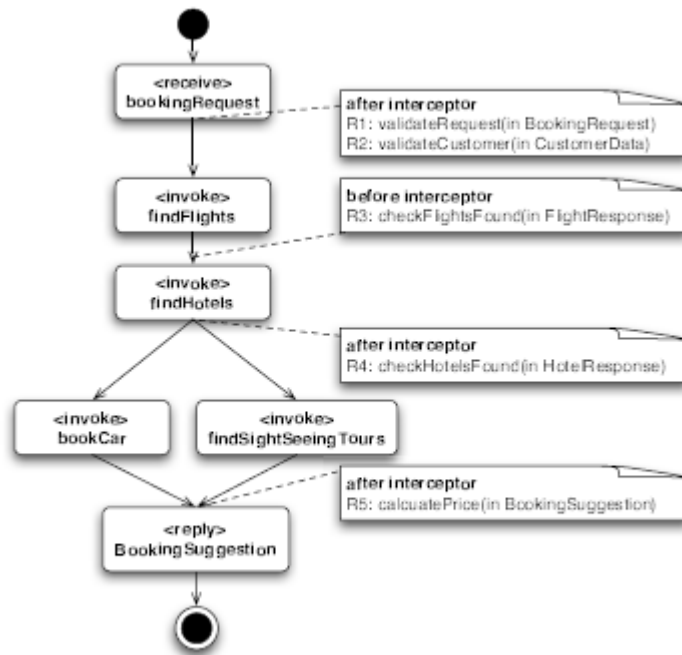
Σύμφωνα με το Business Rules Group [], ως ορισμός των επιχειρηματικών κανόνων θεωρείται ο παρακάτω :

«επιχειρηματικός κανόνας είναι μια δήλωση – ισχυρισμός που καθορίζει ή περιορίζει τη δράση μιας επιχειρηματικής όψης – θέσης. Προορίζεται για να βεβαιώσει - εγκαθιδρύσει επιχειρηματική δομή ή να ελέγξει τη συμπεριφορά της επιχειρηματικής δραστηριότητας».

Οι επιχειρηματικοί κανόνες εκφράζονται επομένως είτε ως *περιορισμοί* είτε με τη μορφή *εάν (υποθέσεις – συνθήκες) τότε δράση (if conditions then action)*. Βασικό χαρακτηριστικό των επιχειρηματικών κανόνων είναι η τάση που παρουσιάζουν να μεταβάλλονται κάθε φορά που μεταβάλλεται η πολιτική στην οποία ενσωματώνονται. Οι επιχειρηματικοί κανόνες είναι εξαιρετικά χρήσιμοι υπό την έννοια ότι αποτελούν ένα μέσο έκφρασης και διαχείρισης της επιχειρηματικής λογικής ανεξάρτητο από τις εφαρμογές όπου εφαρμόζονται.

Οι επιχειρηματικοί κανόνες μπορούν να κατηγοριοποιηθούν με βάση τη λειτουργία τους και έχουν προταθεί στη βιβλιογραφία μια σειρά διαφορετικών προσεγγίσεων. Στο [34] παρουσιάζεται μια κατηγοριοποίηση των επιχειρηματικών κανόνων σε τέσσερις διαφορετικούς τύπους, όπου ο τέταρτος (deontic assignments) αναγνωρίζεται μερικώς, οπότε επικεντρωνόμαστε στους υπόλοιπους τρεις. Για την καλύτερη κατανόηση των διαφόρων κατηγοριών παρουσιάζεται σχηματικά μια απλή επιχειρηματική διαδικασία σε BPEL δομή και γραφή [34], όπου οι επιχειρηματικοί κανόνες παρουσιάζονται ως επισημάνσεις με την ένδειξη “**before interceptor**” στην περίπτωση που πρέπει να εκτελεστούν πριν την δραστηριότητα της BPEL και με την ένδειξη “**after interceptor**” στην αντίθετη περίπτωση. Σε μετέπειτα ενότητα εξηγούνται ως αρχιτεκτονική προσέγγιση οι ιδέες της συγκεκριμένης λογικής προσέγγισης και η χρήση των συγκεκριμένων ενδείξεων. Επιστρέφοντας στο θέμα της κατηγοριοποίησης καταλήγουμε στους εξής τύπους κανόνων :

- ✘ **Integrity Rules or Integrity Constraints (Κανόνες ακεραιότητας ή περιορισμοί ακεραιότητας):** Καθορίζουν δηλώσεις - ισχυρισμούς που πρέπει να ικανοποιούνται σε κάθε στάδιο υλοποίησης και εκτέλεσης του συστήματος. Με βάση τη σχηματική μελέτη περίπτωση, οι κανόνες R1 και R2 αντιπροσωπεύουν τη συγκεκριμένη κατηγορία. Οι κανόνες R1 και R2 ελέγχουν την ορθότητα, τη συνέπεια και την πληρότητα των εισερχόμενων δεδομένων προκειμένου να συνεχιστεί η πορεία της διαδικασίας. Σε διαφορετική περίπτωση αντιμετωπίζεται με τη χρήση εξαιρέσεων (exception).
- ✘ **Derivation Rules or Deduction Rules or Horn clauses:** Αποτελούν δηλώσεις που παράγουν γνώση που πηγάζει από προηγούμενη μορφή γνώσης με τη χρησιμοποίηση ενός μαθηματικού υπολογισμού ή με την εξαγωγή ενός συμπεράσματος. Ο υπολογισμός της τελικής τιμής που πραγματοποιείται μέσω του κανόνα R5 στο σχετικό σχήμα είναι ένας τυπικός τέτοιος κανόνας.
- ✘ **Reaction Rules or Action Rules (Κανόνες Δράσης):** Καθορίζουν την επίκληση και τη δημιουργία δράσης και ενέργειας ως απόκριση σε κάποιο γεγονός. Η ενέργεια πραγματοποιείται μόνο σε περίπτωση που ικανοποιείται και εφαρμόζεται μια συγκεκριμένη κατάσταση. Οι κανόνες R3 και R4 του σχήματος μπορούν να αναγνωριστούν ως κανόνες δράσης με την έννοια ότι επιτρέπουν ή αποτρέπουν την πραγματοποίηση κάποιας συγκεκριμένης ενέργειας με βάση ένα σημασιολογικό περιεχόμενο.



Σχήμα 7-1: Παράδειγμα Επιχειρηματικής Διαδικασίας με Επιχειρηματικούς Κανόνες

Συγχρόνως, το [5] παρέχει μια διαφορετική κατηγοριοποίηση υψηλού επιπέδου, διακρίνοντας τους επιχειρηματικούς κανόνες σε όρους (terms), κανόνες (rules) και γεγονότα (facts). Οι όροι και τα γεγονότα είναι δηλώσεις και αναφορές που περιέχουν ευαίσθητες και σχετικές με επιχειρησιακά θέματα παρατηρήσεις, ενώ οι κανόνες χρησιμοποιούνται προκειμένου να ανακαλύψουν καινούριες πληροφορίες ή να καθοδηγήσουν τη διαδικασία λήψης αποφάσεων. Συγκεκριμένα :

- ✘ **Γεγονότα (Facts):** Είναι αντικείμενα με επιχειρηματική έννοια και αποτελούν οντότητες που πηγάζουν από τις επιχειρηματικές εφαρμογές. Για παράδειγμα, το αντικείμενο «πελάτης», ή το «πιστωτικό υπόβαθρο» σε μια εφαρμογή για τη δανειοληπτική διαδικασία είναι facts.
- ✘ **Κανόνες (Rules):** Είναι σύνολο εκφράσεων που αποτιμώνται και καταλήγουν σε μια απόφαση μεταβάλλοντας την ροή των εργασιών.
- ✘ **Όροι (Terms):** Είναι αντικείμενα και οντότητες που καταγράφουν παρατηρήσεις και εμφανίζουν επιχειρηματική λογική.

Σύμφωνα, πάντως, με το [4] η συγκεκριμένη κατηγοριοποίηση είναι εξαιρετικά γενικευμένη και δε μπορεί να εφαρμοστεί στην περίπτωση της ανάπτυξης σύνθετων Υπηρεσιών. Πράγματι, δεν είναι ξεκάθαρος ο τρόπος με τον οποίο η κατηγοριοποίηση αυτή των επιχειρηματικών κανόνων μπορεί να χρησιμοποιηθεί για να καθορίσει περιορισμούς δρομολόγησης, κριτήρια και υποθέσεις αποφάσεων για την επιλογή δραστηριοτήτων και πόρων, περιορισμούς στην εκτέλεση των Υπηρεσιών και ανησυχίες για την ποιότητα και το κόστος επιλογής των παροχών Υπηρεσιών. Επομένως το [4] προχωρά σε κατηγοριοποίηση των επιχειρηματικών κανόνων με βάση τις παρακάτω διαστάσεις :

- ✘ Η πρώτη διάσταση ασχολείται με το αντικείμενο ενασχόλησης και διακρίνει την περίπτωση που σχετίζεται με ένα σύνθετο στοιχείο (composition element) ή την περίπτωση που έχει σχέση με ένα σύνθετο κανόνα (composition rule).
- ✘ Η δεύτερη διάσταση τοποθετεί τα στοιχεία και τους κανόνες σε σχέση με το λειτουργικό τμήμα της επιχειρηματικής διαδικασίας στο οποίο ανήκουν.

Καταλήγουμε με αυτόν τον τρόπο σε κανόνες σχετικούς με τη δομή (structure), τον επιχειρηματικό ρόλο (role), την ανταλλαγή μηνυμάτων (message), την πυροδότηση γεγονότων (event) και την ύπαρξη περιορισμών (constraint). Συγκεκριμένα :

✘ Κανόνες σχετικοί με τη δομή της Διαδικασίας (Structure related rules)

Είναι κανόνες που σχετίζονται με την αντίληψη του τρόπου και της μεθόδου με την οποία πραγματοποιούνται οι επιχειρηματικές δραστηριότητες σε μια διαδικασία, και διευκολύνουν τη χάραξη κατευθύνσεων ως προς τις προδιαγραφές του τρόπου με τον οποίο η σύνθεση και η ενορχήστρωση Υπηρεσιών πρόκειται να υλοποιηθεί από την πλευρά των δραστηριοτήτων. Για να εκφραστεί η πληροφορία του τρόπου σύνθεσης χρησιμοποιούνται στοιχεία δραστηριότητας (activity) και ροής (flow). Για να συνδυαστούν και να συσχετιστούν τα στοιχεία αυτά αναγνωρίζονται και χρησιμοποιούνται οι λεγόμενοι *flow grouping* και *activity dependency* κανόνες. Οι κανόνες αυτοί αναδεικνύουν αντίστοιχα τον τρόπο με τον οποίο οι δραστηριότητες πρόκειται να ομαδοποιηθούν και τις εξαρτήσεις που υφίστανται μεταξύ τους. Παράδειγμα αποτελεί ο παρακάτω κανόνας εξάρτησης που καθορίζει ένα είδος προτεραιότητας.

```
if (FlightBookingActivity and HotelActivity depended) then
(HotelActivity performedAfter FlightBookingActivity)
```

✘ Κανόνες σχετικοί με τους ρόλους των εμπλεκόμενων στη Διαδικασία (Role related rules)

Είναι κανόνες που ελέγχουν τους συμμετέχοντες που εμπλέκονται στη σύνθεση των Υπηρεσιών. Η αλληλεπίδραση των καταναλωτών και των παροχέων Υπηρεσιών που είναι οι κυρίως εμπλεκόμενοι με άλλα στοιχεία της επιχειρηματικής διαδικασίας περιλαμβάνουν την απόδοση ρόλων σε μια δραστηριότητα, την πυροδότηση ενός γεγονότος και τη συσχέτιση ενός ρόλου σε έναν παροχέα Υπηρεσιών. Για την πραγματοποίηση των αλληλεπιδράσεων αυτών μπορούμε να χρησιμοποιήσουμε *role assignment*, *event raiser* και *role binding* κανόνες αντίστοιχα. Παράδειγμα αποτελεί ο παρακάτω κανόνας που απεικονίζει τις αεροπορικές εταιρείες που μπορούν να πραγματοποιήσουν μια συγκεκριμένη δραστηριότητα :

```
if (FlightBookingActivity is performed) then (Role type is
airline)
```

✘ Κανόνες σχετικοί με την ανταλλαγή Μηνυμάτων (Message related rules)

Είναι κανόνες που ελέγχουν την ανταλλαγή μηνυμάτων και επιτηρούν τη χρήση των πληροφοριών που μεταφέρουν. Σχετίζονται με τα στοιχεία μηνύματος των διαδικασιών (message elements), κατανέμουν τις πληροφορίες τους στις δραστηριότητες που συμμετέχουν και πρόκειται για τους λεγόμενους *message*

distribution κανόνες. Στην περίπτωση που σχετίζονται με εξαρτήσεις μεταξύ μηνυμάτων (*message dependencies*) χρησιμοποιούμε *message dependency* κανόνες. Παράδειγμα αποτελεί ο παρακάτω κανόνας που αποδίδει ένα μήνυμα με τις τιμές εισόδου μιας δραστηριότητας :

```
if (FlightBookingActivity has Input) then (Message contains
departureDate, arrivalDate, from, to)
```

✘ Κανόνες σχετικοί με τα Γεγονότα (Event related rules)

Είναι κανόνες που ελέγχουν τη συμπεριφορά της σύνθετης επιχειρηματικής διαδικασίας ως αντίδραση σε κάποιο γεγονός. Η συσχέτιση της πραγματοποίησης κάποιου γεγονότος με τις δραστηριότητες στις οποίες επιδρά ελέγχεται από τους *activity influence* κανόνες. Η δραστηριότητα που πρέπει να εκτελεστεί για την αντιμετώπιση ενός γεγονότος ελέγχεται από τους *event handler* κανόνες. Παρακάτω δίνεται σχετικό παράδειγμα :

```
if (SeatUnavailableEvent occurs) then (Stop the composition)
```

✘ Κανόνες σχετικοί με Περιορισμούς (Constrained related rules)

Είναι κανόνες που καθοδηγούν τη χρήση περιορισμών σε μια επιχειρηματική διαδικασία και αντιπροσωπεύονται από όρους κατά τη ανάπτυξη σύνθετων Υπηρεσιών. Οι όροι σχετίζονται με δραστηριότητες και καθορίζουν τη μεταφορά της ροής δεδομένων σε διαφορετική δραστηριότητα μέσω προ- και μετα-υποθέσεων (*pre- and post-conditions*). Επομένως, γίνεται χρήση *pre-condition assignment* και *post-condition assignment* κανόνων αντίστοιχα. Επίσης υπάρχει η δυνατότητα ελέγχου της εμφάνισης γεγονότων, ενώ παράλληλα οι κανόνες αυτοί επιτελούν περιορισμούς ενοποίησης με τη χρήση *event control* και *message constraint* κανόνων.

```
if (FlightBookingActivity completed) then (Seat must be
reserved)
```

3.3 Μηχανές Παραγωγής Κανόνων (Rules Engines)

Προκειμένου να αναπτυχθεί μια μηχανή παραγωγής κανόνων υπάρχει η ανάγκη υιοθέτησης των κατάλληλων αλγορίθμων εξαγωγής συμπερασμάτων, αποτίμησης των συνθηκών και προσδιορισμού της σειράς εκτέλεσης. Το εργαλείο που θα επιλεγεί για την ανάπτυξη του μηχανισμού εξαγωγής συμπερασμάτων του έμπειρου συστήματος, το οποίο μπορεί να είναι μία γλώσσα προγραμματισμού, ή πιο συχνά στις μέρες μας ένα κέλυφος, εξαρτάται άμεσα από τη μέθοδο που θα χρησιμοποιηθεί για την αναπαράσταση της γνώσης. Συγκεκριμένες γλώσσες έχουν αναπτυχθεί για τη διαχείριση γνώσης με συγκεκριμένη αναπαράσταση, όπως για παράδειγμα η PROLOG που χρησιμοποιείται για τη διαχείριση του κατηγορηματικής λογικής, και η LISP ως αντιπροσωπευτικό παράδειγμα του συναρτησιακού προγραμματισμού για τη διαχείριση σχημάτων. Συγχρόνως, η επιλογή των αλγορίθμων αποτίμησης και διαχείρισης των κανόνων προϋποθέτει μεθόδους ελέγχου της απόδοσης τους συστήματος και της εφαρμογής σε πραγματικά δεδομένα. Οι μηχανές εξαγωγής συμπερασμάτων που διαχειρίζονται τη γνώση η οποία έχει κωδικοποιηθεί με τη

μορφή κανόνων παραγωγής αποτελούν μέρος των Συστημάτων Διαχείρισης Κανόνων Παραγωγής.

Τα Συστήματα Διαχείρισης Κανόνων Παραγωγής στην πιο εξελιγμένη μορφή τους, πέρα από το μηχανισμό εξαγωγής συμπερασμάτων, περιλαμβάνουν περιβάλλοντα ανάπτυξης και αποσφαλμάτωσης των κανόνων παραγωγής. Πρόκειται, δηλαδή, για ολοκληρωμένα συστήματα αναπαράστασης γνώσης υπό τη μορφή κανόνων που είναι υπεύθυνα για τον πλήρη κύκλο διαχείρισης τους που περιλαμβάνει τη δημιουργία των κανόνων, την επιλογή της σειράς αποτίμησης και του αλγορίθμου εκτέλεσης, τον έλεγχο της ορθής λειτουργίας του συνόλου κανόνων, την εκτέλεση τους με πραγματικά δεδομένα, και τη δυνατότητα αποθήκευσης και μεταβολής τους. Ενδεικτικά μπορούμε να αναφέρουμε τις παρακάτω σύγχρονες μηχανές παραγωγής που έχουν αναπτυχθεί με τα παρακάτω συγκριτικά αποτελέσματα:

- ✘ **Jess (Java Expert System Shell)**
- ✘ **JBoss Rules**
- ✘ **Blaze Advisor**
- ✘ **ILOG JRules**
- ✘ **OpenRules**
- ✘ **VisualRules**
- ✘ **OP SJ**
- ✘ **QuickRules**

	Jess	Drools	Blaze Advisor	ILOG JRules	OpenRules	VisualRules	QuickRules	OP SJ
Υλοποιεί το πρότυπο JSR 94	√	√	√	√	√	√	√	√
Άδεια χρήσης	Free for Academic Use	Open Source, freeware	50.000\$ per developer	50.000 \$ (Starter Pack)	2.895\$/ installation	Pricelist available on request	Pricelist available on request	8.000\$ per developer
BRMS	X	√	Rule Maintenance Applications	Rule Studio	MS Excel, Google Spreadsheet	√	Rule Builder IDE	Windowed Development Environment
Forward Chaining	√	√	√	√	√	√	√	√
Backward Chaining	√	X	√	X	X	X	X	√
Περιβάλλον ανάπτυξης	Eclipse IDE	Eclipse IDE	Integrated Development Environment	Integrated Development Environment	Eclipse IDE	Visual Rools Build Tools	Eclipse IDE, NetBeans IDE	Eclipse IDE
Αλγόριθμος ταυτοποίησης	Enhanced Rete	Rete OO	Rete III	Rete, Sequential	XML-based, non-Rete	Not using inference	Rete	Rete II (the fastest)
Fuzzy Logic	√	X	Managing of missing data	X	X	X	X	X
Δυνατότητα αποσφαλμάτωσης	X	√	√	√	Rule Validator	Visual Rules Modeller	√	X
Υποστήριξη Μηχανικής Μάθησης	X	X	X	X	√	X	X	X

Πίνακας 7-1: Σύγκριση Εμπορικών Μηχανών Παραγωγής

3.3.1 Προτυποποίηση των Συστημάτων Κανόνων Παραγωγής

Μία μηχανή κανόνων παραγωγής είναι ουσιαστικά ένα σύνολο εργαλείων που επιτρέπουν στους αναλυτές των επιχειρήσεων και στους προγραμματιστές να χτίζουν λογική λήψης αποφάσεων. Τα πληροφοριακά συστήματα της επιχείρησης μπορούν να επικοινωνήσουν με τη μηχανή εκτέλεσης των κανόνων, τροφοδοτώντας τη με αρχικά δεδομένα (Γεγονότα), και στη συνέχεια λαμβάνοντας το αποτέλεσμα της εφαρμογής των επιχειρησιακών κανόνων πάνω στα δεδομένα. Η έλλειψη προτύπων, εντούτοις, μπορεί να αποτελεί ισχυρό αποτρεπτικό παράγοντα από το να χρησιμοποιήσει μία επιχείρηση εφαρμογές βασισμένες σε κανόνες. Οι περισσότερες μηχανές κανόνων παραγωγής υλοποιούν εξειδικευμένα API, που καθιστά δύσκολη της ενσωμάτωσή τους με τα υπάρχοντα πληροφοριακά συστήματα της επιχείρησης. Επιπλέον, αν μία μηχανή παύσει να υποστηρίζεται τεχνικά και η επιχείρηση αποφασίσει να υιοθετήσει άλλη μηχανή, το μεγαλύτερο μέρος του κώδικα της εφαρμογής θα πρέπει να υλοποιηθεί ξανά.

Το JSR 94 αποτελεί μία προσπάθεια προτυποποίησης των μηχανών κανόνων παραγωγής για την τεχνολογία της Java. Το JSR 94 παρέχει καθοδήγηση για τη διαχείριση των κανόνων, καθώς επίσης και τα API (Application Programming Interface) που πρέπει να υλοποιεί μία μηχανή για διαδικασίες όπως η εκτέλεση των κανόνων, αλλά δεν προβάλλει κανένα περιορισμό για τη γλώσσα στην οποία θα εκφραστούν οι κανόνες και οι ενέργειες. Ωστόσο, υπάρχουν προσπάθειες προτυποποίησης μίας κοινής γλώσσας κανόνων, συμπεριλαμβανομένης της Rule Markup Language (RuleML).

Το JSR 94 ορίζει ένα απλό API για την πρόσβαση σε μία μηχανή κανόνων παραγωγής από μία Java EE ή SE εφαρμογή. Παρέχει API για :

- ✘ Την εγκατάσταση και την απεγκατάσταση κανόνων
- ✘ Τη διαπέραση των κανόνων
- ✘ Τον έλεγχο των μεταδεδομένων των κανόνων
- ✘ Την εκτέλεση των κανόνων
- ✘ Την ανάκτηση των αποτελεσμάτων
- ✘ Το φιλτράρισμα των αποτελεσμάτων

Το JSR 94 δεν προτυποποιεί:

- ✘ Την ίδια τη μηχανή κανόνων παραγωγής
- ✘ Τη ροή εκτέλεσης των κανόνων
- ✘ Τη γλώσσα που θα χρησιμοποιηθεί για την περιγραφή των κανόνων

Τα API του JSR 94 διακρίνονται σε δύο βασικά πακέτα [27]:

- ✘ **Rules Administration API**: Παρέχει κλάσεις που μπορούν να χρησιμοποιηθούν για την φόρτωση των κανόνων που θα αποτελέσουν μέρος του συνόλου των κανόνων προς εκτέλεση, από εξωτερικές πηγές όπως URI, InputStream, XML, ώστε να δημιουργηθεί ένα σύνολο υποψηφίων προς εκτέλεση κανόνων. Το πακέτο αυτό μπορεί επίσης να χρησιμοποιηθεί για τον

καθορισμό δικαιωμάτων και βαθμίδων πρόσβασης στην εκτέλεση του συνόλου των κανόνων.

- ✘ **Runtime Client API:** Παρέχει κλάσεις που μπορούν να χρησιμοποιηθούν από τις εφαρμογές πελάτη για την εκτέλεση των κανόνων και την λήψη των αποτελεσμάτων ενός υλοποιημένου Rule Engine API. Στην καρδιά του Runtime Client API βρίσκεται το Rule Runtime Interface, που παρέχει μεθόδους οι οποίες επιτρέπουν στους χρήστες να δημιουργούν μία RuleSession. Η RuleSession είναι μία σύνδεση σε χρόνο εκτέλεσης μεταξύ του πελάτη και της μηχανής εκτέλεσης των κανόνων. Συνδέεται με ένα συγκεκριμένο execution set και μπορεί να χρησιμοποιεί πόρους της μηχανής, αλλά πρέπει να καταστρέφεται όταν δεν εξυπηρετεί πλέον το χρήστη. Συνεπώς, οι λειτουργίες της RuleSession περιλαμβάνουν αφενός την παροχή ενός μηχανισμού που επιτρέπει την πρόσβαση στη λίστα των execution sets που έχουν εγκατασταθεί, και αφετέρου να καθορίσει τον τύπο της σύνδεσης που θα εγκατασταθεί με τον πελάτη. Η σύνδεση είναι stateless όταν ενεργοποιείται με την αίτηση του πελάτη και καταστρέφεται με το πέρας της εκτέλεσης της λειτουργίας που αιτήθηκε ο πελάτης, και stateful όταν όλα τα δεδομένα των κανόνων διατηρούνται για όσο διαρκεί η σύνδεση του πελάτη με τη μηχανή.

3.4 Αρχιτεκτονική προσέγγιση για την χρήση κανόνων στις επιχειρηματικές διαδικασίες στο πλαίσιο της αρχιτεκτονικής προσανατολισμένης σε Υπηρεσίες

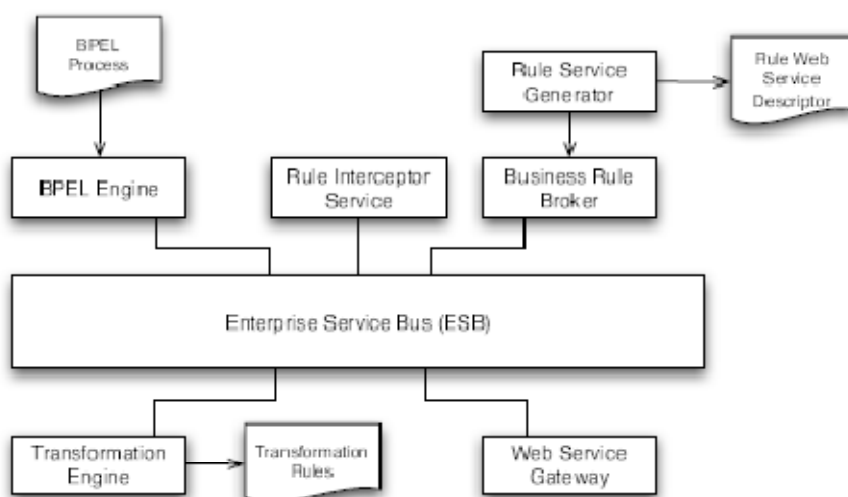
Η ενοποίηση ενός συστήματος βασισμένου σε κανόνες μέσα σε ένα περιβάλλον με αρχιτεκτονική προσέγγιση προσανατολισμένη σε υπηρεσίες είναι ένα πολύπλοκο εγχείρημα λόγω των διαφορετικών αρχιτεκτονικών και εννοιολογικών προσεγγίσεων των προς ενοποίηση πλευρών. Τα συστήματα διαχείρισης κανόνων αποτελούν σημαντικό τμήμα υλοποίησης επιχειρηματικής λογικής και επομένως είναι αναγκαίο η αρχιτεκτονική τους σύνδεση με μια μηχανή ενορχήστρωσης δικτυακών υπηρεσιών [15].

Προς την κατεύθυνση της ενοποίησης και της ολοκλήρωσης της BPEL με ένα σύστημα διαχείρισης κανόνων, υπάρχει η δυνατότητα διαφοροποίησης των προσεγγίσεων ανάλογα με το βαθμό της σύζευξης που είναι επιθυμητός. Σε μια αρχιτεκτονική προσέγγιση με ισχυρή σύζευξη, δίνεται η δυνατότητα άμεσης επικοινωνίας μεταξύ της μηχανής ενορχήστρωσης και του συστήματος παραγωγής κανόνων μέσω του ιδιόκτητου API του συστήματος κανόνων. Λόγω του γεγονότος ότι η προδιαγραφή της BPEL αποφεύγει την προτυποποίηση ενός API πρόσβασης στην μηχανή BPEL, οι περισσότεροι κατασκευαστές συστημάτων κανόνων παραλείπουν τη δημιουργία διεπαφών για την άμεση επικοινωνία με μηχανές BPEL. Επομένως, το παραπάνω σε συνδυασμό με την απώλεια της ταυτότητας του προσανατολισμού σε υπηρεσίες λόγω της ισχυρής σύζευξης, οδηγούν στην υιοθέτηση προσέγγισης χαλαρής σύζευξης.

Στην προσέγγιση που ακολουθεί μειωμένο βαθμό σύζευξης, η ιδέα στηρίζεται στην κατάρτιση των επιχειρηματικών κανόνων με βάση τις δυνατότητες της μηχανής

παραγωγής κανόνων και στη συνέχεια στην εξωτερίκευση τους με τη μορφή Διαδικτυακών Υπηρεσιών, με αποτέλεσμα να μπορούν να προσπελαστούν με απλό τρόπο και να επαναχρησιμοποιηθούν ως υπηρεσίες από άλλες διεπιχειρησιακές εφαρμογές. Με αυτόν τον τρόπο ακολουθείται το παράδειγμα της SOA αρχιτεκτονικής με παράλληλη δυνατότητα ευελιξίας και επαναχρησιμοποίησης.

Όσον αφορά την αρχιτεκτονική προσέγγιση [34], χρησιμοποιείται ως πλατφόρμα ενοποίησης ο Επιχειρησιακός Δίαυλος Υπηρεσιών (Enterprise Service Bus) που αποτελεί το βασικό θεμέλιο στήριξης του οικοδομήματος της αρχιτεκτονικής προσανατολισμένης σε Υπηρεσίες. Η μηχανή BPEL συνδέεται στον δίαυλο ESB με ένα προσαρμογέα (adapter) και τον χρησιμοποιεί ως βαθμίδα ανταλλαγής μηνυμάτων και άμεσης επικοινωνίας με την Πύλη των Διαδικτυακών Υπηρεσιών (Web Service Gateway) προκειμένου να καλέσει εξωτερικά τις Διαδικτυακές Υπηρεσίες μέσω των εντολών <invoke> και <reply> ή να αναμένει για απάντηση μέσω της εντολής <receive>.



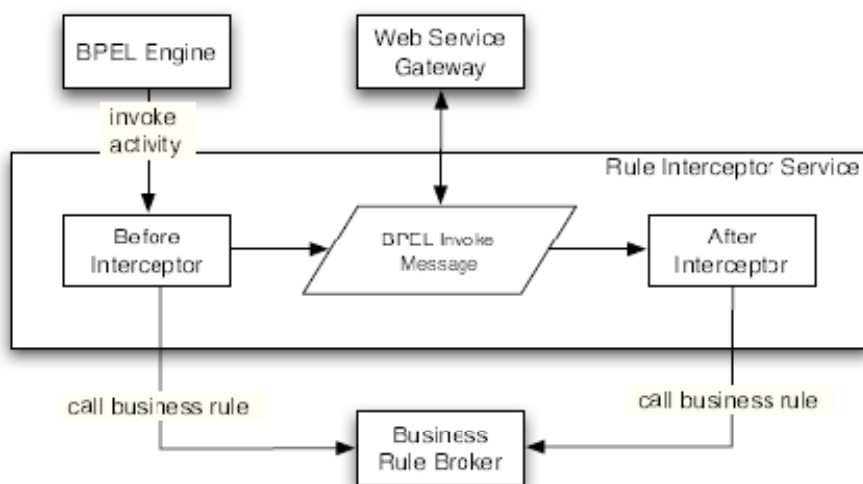
Σχήμα 7-2: Αρχιτεκτονική Προσέγγιση για την Ενσωμάτωση Επιχειρηματικών Κανόνων στην SOA Αρχιτεκτονική

Τα βασικά δομικά στοιχεία της αρχιτεκτονικής είναι τα εξής :

- ✘ **Μεσίτης Επιχειρηματικών Κανόνων (Business Rule Broker):** Εξαιτίας της ετερογένειας των διαφορετικών APIs των μηχανών παραγωγής κανόνων, χρησιμοποιείται ένας Μεσίτης Επιχειρηματικών Κανόνων που παρέχει μια ενοποιημένη πρόσβαση σε διαφορετικά συστήματα διαχείρισης κανόνων μέσω ενός περιβάλλοντος διεπαφής Διαδικτυακών Υπηρεσιών. Η αρχιτεκτονική του μεσίτη εμφανίζει ένα επίπεδο αφαιρετικότητας από τη συγκεκριμένη υλοποίηση κάθε μηχανής κανόνων με την παροχή ενός μηχανισμού plugin για την επίτευξη της ενοποίησης διαφορετικών μηχανών. Το στοιχείο της προσέγγισης αυτής που ξεχωρίζει, είναι η αυτοματοποιημένη παραγωγή των Διαδικτυακών Υπηρεσιών για την εκτέλεση των επιχειρηματικών κανόνων με βάση έναν περιγραφητή *Web Service Rule*

Interface Descriptor, που περιγράφει τις υπηρεσίες που παράγονται και τους κανόνες που πρόκειται να εκτελεστούν από τις συγκεκριμένες υπηρεσίες.

- ✘ **Rule Interceptor Service:** Η Υπηρεσία του Rule Interceptor αποτελεί τη γέφυρα μεταξύ των επιχειρηματικών κανόνων και της εκτελέσιμης επιχειρηματικής διαδικασίας. Η προσέγγιση που χρησιμοποιείται και προτείνεται, είναι να εξαναγκάζεται κάθε εισερχόμενη και εξερχόμενη κλήση διαδικτυακών υπηρεσιών της επιχειρηματικής διαδικασίας BPEL να εφαρμόζει αυτόματα τους επιχειρηματικούς κανόνες που είναι προσπελάσιμοι μέσω της υπηρεσίας του Μεσίτη Επιχειρηματικών Κανόνων. Η αντιστοίχιση και συσχέτιση των BPEL δραστηριοτήτων με τους πραγματικούς επιχειρηματικούς κανόνες πραγματοποιείται μέσω ενός εγγράφου αντιστοίχισης (mapping document) που πρέπει να αναπαραχθεί από τον BPEL Designer. Παράλληλα δίνεται η δυνατότητα επιλογής μεταξύ της εκτέλεσης των κανόνων πριν ή μετά την BPEL δραστηριότητα.



Σχήμα 7-3: Αρχιτεκτονική Προσέγγιση Υπηρεσίας Rule Interceptor

- ✘ **Μηχανή Μετασχηματισμού (Transformation Engine) :** Οι επιχειρηματικοί κανόνες που παράγονται εμφανίζουν μηνύματα διαφορετικών ειδών ως παραμέτρους. Επομένως, υπάρχει η πιθανότητα οι τύποι των μηνυμάτων που αναμένουν οι υπηρεσίες που ενσωματώνουν τη λειτουργία των επιχειρηματικών κανόνων να μην συμβαδίζουν με κάποιο υπάρχον XML σχήμα αλλά να υπάρχει η δυνατότητα μετατροπής τους σε αυτό. Συνεπώς, μια μηχανή μετασχηματισμού είναι απαραίτητη προκειμένου να μετατραπούν τα XML μηνύματα σε διαφορετικές μορφές που να είναι κατανοητές και συμβατές με τους επιχειρηματικούς κανόνες.

Παράλληλα έχουν αναπτυχθεί διαφορετικές αρχιτεκτονικές προσεγγίσεις που επιχειρούν τον ορισμό δυναμικών ροών εργασίας μέσω της χρήσης επιχειρηματικών κανόνων. Στο [13] οι συγγραφείς προτείνουν ένα υβριδικό πρότυπο για την πραγματοποίηση ενοποίησης και ολοκλήρωσης μεταξύ επιχειρηματικών κανόνων (που μοντελοποιούνται ως όψεις - aspects) και μιας μηχανής ενορχήστρωσης που στηρίζεται στη BPEL με τη χρήση προγραμματιστικών αρχιτεκτονικών

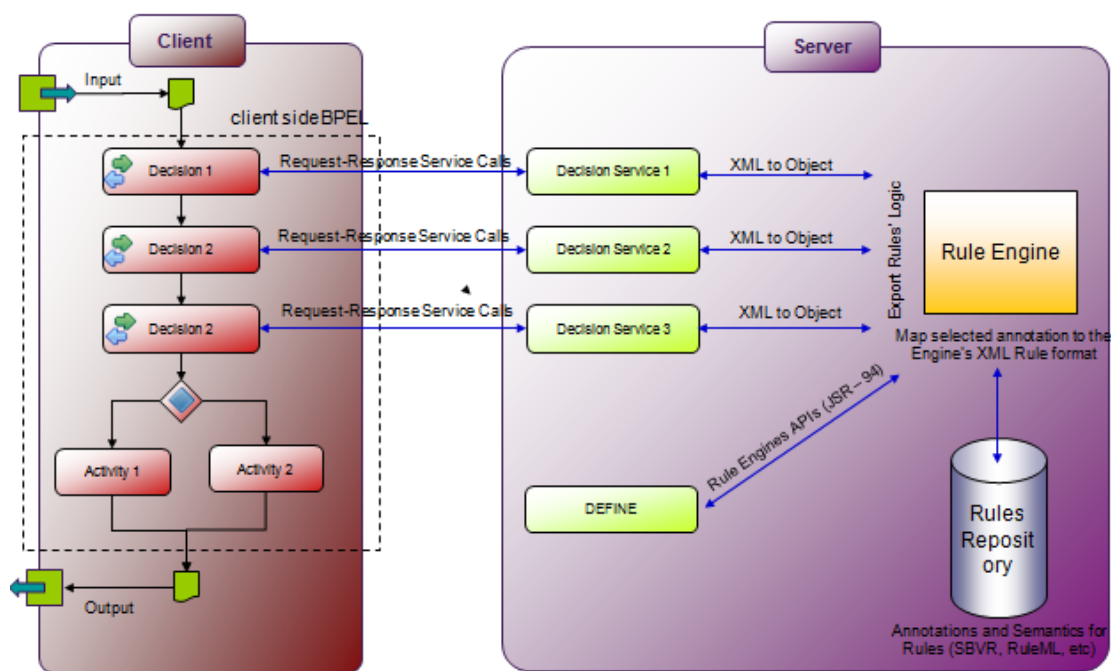
προσανατολισμένων στις όψεις. Στο [4] παρουσιάζεται μια προσέγγιση που οδηγείται από επιχειρηματικούς κανόνες, κατά τη οποία προτείνεται μια τεχνική δυναμικής σύνθεσης επιχειρηματικών διαδικασιών με βάση την ύπαρξη επιχειρηματικών κανόνων.

4 Υλοποίηση Πρωτότυπης Εφαρμογής

4.1 Αρχιτεκτονική του συστήματος

Ένα από τα απαραίτητα στοιχεία που πρέπει να χαρακτηρίζουν τη διαδικασία υλοποίησης ενός επιτυχημένου συστήματος ηλεκτρονικών συναλλαγών είναι η ικανότητα διαχείρισης και πραγματοποίησης ροών εργασίας. Με βάση τη συγκεκριμένη προσέγγιση, για να επιτευχθεί η ανάγκη επικοινωνίας και σύνδεσης μεταξύ των συναλλασσόμενων στα πλαίσια μιας ηλεκτρονικής συναλλαγής, καθένα από τα συμβαλλόμενα πληροφοριακά συστήματα πρέπει να καθορίζει εκ των πρότερων τις εσωτερικές επιχειρηματικές διαδικασίες που θα χρησιμοποιηθούν, ώστε να συμφωνηθεί εκ των υστέρων μια κοινή ροή δραστηριότητας που θα ακολουθηθεί και από τα δύο συστήματα. Στην πράξη, η συγκεκριμένη προσέγγιση αποδεικνύεται ανελαστική και μη λειτουργική, καθώς οποιαδήποτε εκ των υστέρων μεταβολή στην επιχειρηματική διαδικασία προϋποθέτει την σύσταση νέων επιχειρηματικών ροών που πρέπει να συμφωνηθούν εκ νέου.

Η προτεινόμενη λύση, σε πρώτο επίπεδο, για την επίτευξη δυναμικών επιχειρηματικών ροών εργασίας υπαγορεύει την ύπαρξη μιας αρχιτεκτονικής που να περιλαμβάνει τη χρήση εργαλείων και τεχνολογιών που να στηρίζονται και να εκμεταλλεύονται τις βασικές αρχές διαλειτουργικότητας και Διαδικτυακών Υπηρεσιών.



Σχήμα 8-1: Αρχιτεκτονική Προσέγγιση Προτοτύπου Υλοποίησης

Επομένως προτείνεται μια αρχιτεκτονική τριών επιπέδων που στηρίζεται στην πλατφόρμα J2EE και η οποία ορίζει ότι τα δομικά στοιχεία της εφαρμογής πρέπει να διαχωρίζονται στις ακόλουθες τρεις βαθμίδες:

- ✘ **Τη βαθμίδα της παρουσίας :** Είναι η συλλογή των τεχνολογιών J2EE που σχηματίζουν το περιβάλλον διεπαφής του χρήστη με την εφαρμογή.
- ✘ **Τη βαθμίδα της επιχειρησιακής λογικής :** Είναι η βαθμίδα στην οποία υλοποιείται η επιχειρηματική λογική της εφαρμογής και από την οποία εξάγεται η λογική λειτουργίας και δράσης της εφαρμογής.
- ✘ **Τη βαθμίδα της μόνιμης αποθήκευσης των δεδομένων της εφαρμογής :** Στις περισσότερες περιπτώσεις αφορά κάποια σχεσιακή βάση δεδομένων, αλλά οι δυνατότητες μόνιμης αποθήκευσης είναι πολύ περισσότερες και περιλαμβάνουν αρχεία XML, Serialized Java Beans, ή ακόμα και κλασικά ASCII αρχεία.

Σε μία σωστά δομημένη εφαρμογή, τα στοιχεία κάθε βαθμίδας πρέπει να είναι ‘άκρως συνεκτικά’ (highly cohesive) και ‘χαλαρά συνδεδεμένα’ (loosely coupled) υπό την έννοια ότι κάθε στοιχείο πρέπει να εξυπηρετεί μία και μόνο διαδικασία, ενώ τα στοιχεία μίας βαθμίδας δε θα πρέπει να εξαρτώνται από τη δομή των στοιχείων της άλλης βαθμίδας. Επομένως στα πλαίσια της εφαρμογής :

- ✘ Θα οριστούν πρότυπες ροές εργασίας με βάση διεθνή πρότυπα.
- ✘ Θα δημιουργηθεί ένας ταμειυτήρας από μοντέλα επιχειρηματικών διαδικασιών (Process Model Repository).
- ✘ Θα σχεδιαστεί και θα υλοποιηθεί ένας εξυπηρετητής δυναμικών ροών εργασίας (Workflow Composition Server) ο οποίος θα επεξεργάζεται τα μοντέλα και θα δημιουργεί εκτελέσιμες δυναμικές ροές εργασίας (Execution Patterns) στο πρότυπο της BPEL. Οι συγκεκριμένες ροές εργασίας θα μπορούν εν συνεχεία να εκτελεστούν αυτόματα από ήδη υπάρχουσες μηχανές (Workflow Engines) οι οποίες παίρνοντας τις ως είσοδο καλούν αυτόματα τις υφιστάμενες δικτυακές υπηρεσίες και διεκπεραιώνουν τη συναλλαγή.

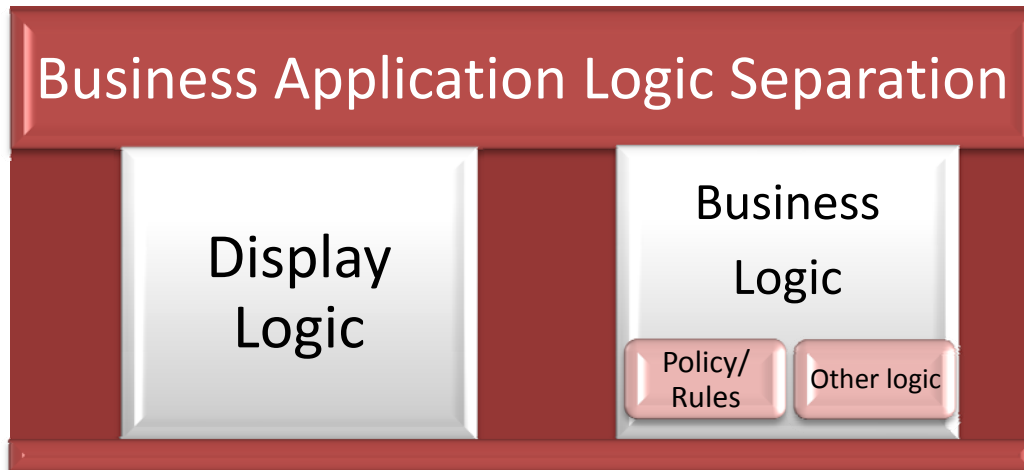
Παράλληλα, στο πλαίσιο της επέκτασης και της ερευνητικής συμβολής της διπλωματικής σε επίπεδο δυναμικών ροών εργασίας, γίνεται προσπάθεια επέκτασης της προτεινόμενης αρχιτεκτονικής και της υιοθέτησης μιας αρχιτεκτονικής τεσσάρων βαθμίδων (4-tier αρχιτεκτονική). Προτείνεται, λοιπόν η περαιτέρω διάσπαση της λογικής του συστήματος σε επιχειρηματική λογική και επιχειρηματική πολιτική.

Η επιχειρηματική πολιτική υλοποιείται με τη μορφή επιχειρηματικών ή νομικών κανόνων (Business and Legal Rules) οι οποίοι διέπουν την λειτουργία της επιχείρησης και όπως είναι φυσιολογικό διαπνέουν και τη λειτουργία των συστημάτων της. Σε τεχνικό επίπεδο οι κανόνες εκτελούνται σε ξεχωριστή μηχανή (Rule Engine) η οποία ανταλλάσει δεδομένα μέσα από δικτυακές υπηρεσίες με τις υπόλοιπες εφαρμογές λογισμικού της επιχείρησης οι οποίες πλέον υλοποιούν μόνο την επιχειρηματική λογική (Business Logic).

Σκοπός της συγκεκριμένης διπλωματικής εργασίας είναι συγχρόνως η δημιουργία και επέκταση ενός πρωτότυπου ταμειυτήρα επιχειρηματικών/νομικών κανόνων. Συγκεκριμένα η παρούσα διπλωματική περιλαμβάνει:

- ✘ Τον ορισμό ενός μοντέλου/οντολογίας για την μοντελοποίηση επιχειρηματικών και νομικών κανόνων.

- ✘ Την εύρεση/δημιουργία μιας κατάλληλης σημειογραφίας, βασισμένη σε XML πρότυπα, για την περιγραφή των κανόνων η οποία θα είναι συμβατή με υπάρχουσες μηχανές εκτέλεσης κανόνων.
- ✘ Τη δημιουργία μιας πρωτότυπης εφαρμογής λογισμικού η οποία με βάση την οντολογία θα προσφέρει ένα περιβάλλον για τον ορισμό κανόνων, θα μεταφράζει τους κανόνες αυτούς στην επιλεγμένη σημειογραφία (XML notation), και τέλος θα εξάγει τη λειτουργικότητα των οριζόμενων κανόνων με τη μορφή διαδικτυακών υπηρεσιών οι οποίες θα μπορούν χρησιμοποιηθούν από οποιοδήποτε πρόγραμμα λογισμικού.



Σχήμα 8-2: Διαχωρισμός Λογικής σε 4-Tier Αρχιτεκτονική

4.2 Αντιστοίχιση αρχιτεκτονικών στοιχείων με εργαλεία

Η υλοποίηση των επιχειρηματικών διαδικασιών πραγματοποιείται εξολοκλήρου σε περιβάλλον ανάπτυξης NetBeans 6.1 το οποίο παρέχει και τον Ταμιευτήρα των Υπηρεσιών. Ως εξυπηρετητής των δυναμικών επιχειρηματικών εφαρμογών και διαδικασιών χρησιμοποιείται ο GlassFish Application Server που συνεργάζεται με βάση την προτυποποίηση της ενοποίησης και διαλειτουργικότητας Java Business Integration, με τη μηχανή παραγωγής κανόνων ILOG JRules προκειμένου να επιτευχθεί το ζητούμενο της αρχιτεκτονικών τεσσάρων επιπέδων.

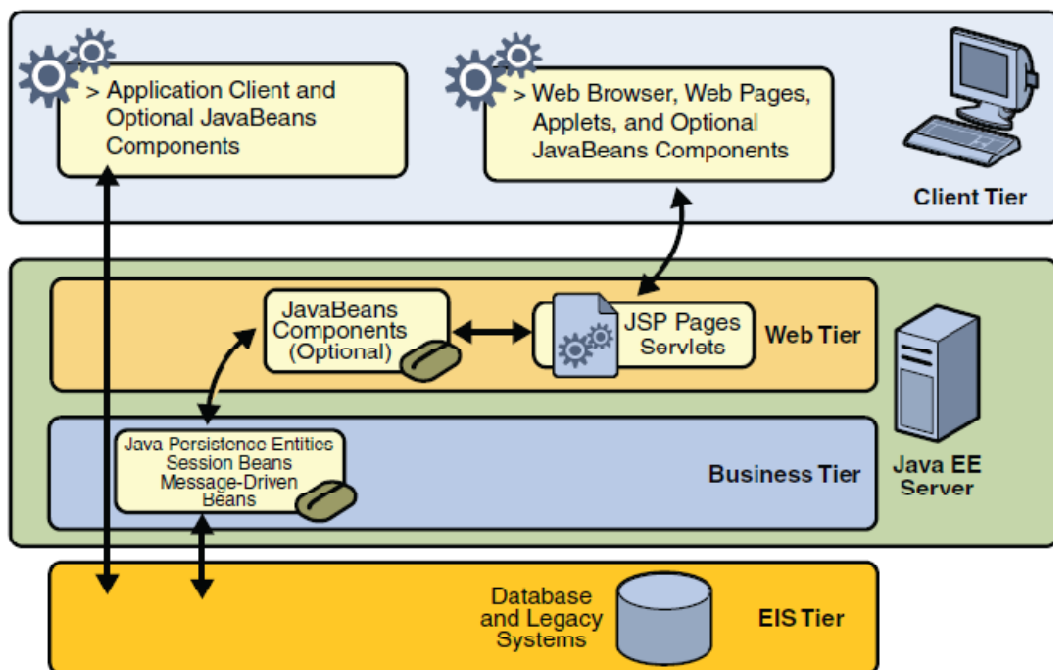
4.3 Περιγραφή των τεχνολογικών προγραμματιστικών εργαλείων

4.3.1 JavaEE 5

Η ανάπτυξη της εφαρμογής έχει στηριχτεί στην πλατφόρμα JavaEE που αποσκοπεί στην υιοθέτηση ενός μεταφέρσιμου, πολύπλευρου και ασφαλούς πρότυπου υλοποίησης επιχειρησιακών εφαρμογών μεγάλης κλίμακας με βάση την προγραμματιστική γλώσσα της Java. Η πλατφόρμα JavaEE κατόρθωσε μέσω του οργανωμένου συνόλου από APIs και της ευελιξίας των εργαλείων ανάπτυξης προγραμματιστικού κώδικα, να εκπληρώσει το σκοπό υλοποίησης της που ήταν ο μειωμένος χρόνος ανάπτυξης εφαρμογών, η μείωση της πολυπλοκότητας των

εφαρμογών και η βελτίωση της απόδοσης. Η πλατφόρμα JavaEE [23] πρόκειται ουσιαστικά για ένα σύνολο προδιαγραφών και στηρίζεται στα παρακάτω δομικά στοιχεία :

- ✘ **Προδιαγραφές:** Κάθε API στο περιβάλλον JavaEE έχει τις δικές του προδιαγραφές οι οποίες είναι διαθέσιμες μέσω του Java Community Process, και οι οποίες πρέπει να ακολουθηθούν κατά την ανάπτυξη JavaEE πλατφορμών. Το γεγονός αυτό αυξάνει τη δυνατότητα μεταφοράς μίας εφαρμογής μεταξύ περιβαλλόντων υλοποιημένων από διαφορετικούς παρόχους, μίας και κάθε πάροχος δεσμεύεται για την έκδοση της προδιαγραφής την οποία έχει ακολουθήσει.
- ✘ **Δοκιμαστικά περιβάλλοντα:** Η Sun παρέχει ένα σύνολο δοκιμαστικών κριτηρίων (Test Compatibility Kit) με τα οποία κάποιος πάροχος μπορεί να επαληθεύσει τη συμβατότητα της πλατφόρμας του με συγκεκριμένη έκδοση των προδιαγραφών. Αν κάποια υλοποίηση περάσει επιτυχώς τις δοκιμές, η Sun παρέχει ένα JavaEE πιστοποιητικό, ώστε να ενημερώσει τους πελάτες ότι η συγκεκριμένη πλατφόρμα εξυπηρετητή είναι JavaEE συμβατή.
- ✘ **Υλοποιήσεις αναφοράς:** Για να διευκολύνει τους προγραμματιστές να αναπτύξουν κώδικα συμβατό με μία πλατφόρμα JavaEE, η Sun παρέχει τις δικές της υλοποιήσεις αναφοράς, χωρίς χρέωση, για κάθε έκδοση του προτύπου JavaEE.



Σχήμα 8-3: 3-Tier Αρχιτεκτονική Πλατφόρμας Java EE

Η JavaEE ορίζει μία ισχυρή σουίτα middleware υπηρεσιών που παρέχουν όλη την απαιτούμενη λειτουργικότητα για την ανάπτυξη multi-tier, server-side εφαρμογών. Στηρίζεται σε αρχιτεκτονική τριών επιπέδων, με βάση τον διαχωρισμό των

εφαρμογών σε επίπεδο παρουσίασης, επιχειρηματικής λογικής και αποθήκευσης των δεδομένων. Η JavaEE επεκτείνει την τεχνολογία J2SE, η οποία καθορίζει τη χρήση των βασικών χαρακτηριστικών της γλώσσας προγραμματισμού Java καθώς και των διαφόρων βιβλιοθηκών που έχουν χτιστεί με βάση τη Java (.net, .awt, .io κ.τ.λ.).

Για την υιοθέτηση ευελιξίας και για την απλοποίηση του προγραμματιστικού κώδικα κατά την ανάπτυξη εφαρμογών, ο Java EE Server έχει σχεδιαστεί με τέτοιο τρόπο ώστε να παρέχει λανθάνουσες υπηρεσίες υπό τη μορφή containers για την υποστήριξη και τη διαχείριση των διαφόρων δομικών στοιχείων της πλατφόρμας. Οι containers είναι μια μορφή διεπαφής (interface) μεταξύ του δομικού στοιχείου και της χαμηλού επιπέδου και συγκεκριμένης πλατφόρμας λειτουργικότητας που υποστηρίζει το συγκεκριμένο στοιχείο. Οι σημαντικότεροι containers είναι οι :

- ✘ **Java EE server:** Πρόκειται για το μέσο στο οποίο εκτελούνται οι εφαρμογές και παρέχει τους EJB και web containers.
- ✘ **Enterprise JavaBeans (EJB) container:** Διαχειρίζεται την εκτέλεση των enterprise beans.
- ✘ **Web container:** Διαχειρίζεται την εκτέλεση των JSP σελίδων και των servlet components.
- ✘ **Application client container:** Διαχειρίζεται την εκτέλεση των application client συστατικών και τρέχει στην πλευρά του πελάτη (client).
- ✘ **Applet container:** Διαχειρίζεται την εκτέλεση των applets. Περιλαμβάνει έναν web browser και ένα Java Plug-in που τρέχουν στην πλευρά του πελάτη.

Κάποιες βασικές τεχνολογίες οι οποίες ορίζονται από το JavaEE και παρέχονται από τις JavaEE -συμβατές υλοποιήσεις, και στις οποίες βασίστηκε η ανάπτυξη του έμπειρου συστήματος, είναι οι ακόλουθες:

- ✘ **Enterprise Java Beans:** Το πρότυπο των EJB ορίζει τη δομή των μονάδων λειτουργικότητας που εγκαθίστανται στους J2EE εξυπηρετητές εφαρμογών και χαρακτηρίζεται από μεθόδους για την υλοποίηση της επιχειρηματικής λογικής. Παρέχει ένα standard πρωτόκολλο επικοινωνίας μεταξύ των μονάδων λειτουργικότητας και των εξυπηρετητών που τις διαχειρίζονται. Τα EJB είναι ο ακρογωνιαίος λίθος των JavaEE server-side εφαρμογών και χρησιμοποιούν μία σειρά άλλων JavaEE τεχνολογιών.
- ✘ **Java Persistence API:** Το Java Persistence API είναι νέο στοιχείο της JavaEE 5 πλατφόρμας και χρησιμοποιεί ένα είδος αντικειμενο-σχεσιακής προσέγγισης προκειμένου να γεφυρώσει το χάσμα μεταξύ ενός αντικειμενοστραφούς προτύπου (όπως είναι η προγραμματιστική προσέγγιση της γλώσσας Java) και μιας σχεσιακής βάσης δεδομένων. Το Java Persistence API αποτελείται από τρία στοιχεία :
 - ✓ Το Java Persistence API
 - ✓ Την σχεσιακή γλώσσα ερωτήσεων σε βάσεις δεδομένων
 - ✓ Τα metadata (δεδομένα σχετικά με τα δεδομένα που χρησιμοποιεί η εφαρμογή) της αντικειμενο-σχεσιακής συσχέτισης.
- ✘ **Java API for XML WS (JAX-WS):** Είναι η κύρια τεχνολογία που παρέχει την υποστήριξη για την ανάπτυξη Διαδικτυακών Υπηρεσιών, σε μία πλατφόρμα JavaEE. Καθορίζει δύο μοντέλα υλοποίησης των Διαδικτυακών

Υπηρεσιών, ένα με βάση την τεχνολογία των Servlets και ένα με βάση την τεχνολογία των EJB, καθώς και μία σειρά απαιτήσεων τόσο στον τρόπο με τον οποίο πρέπει να αναπτύσσονται και να διαμορφώνονται οι Διαδικτυακές Υπηρεσίες, όσο και στον τρόπο με τον οποίο πρέπει αυτά να υποστηρίζονται από την JavaEE πλατφόρμα.

- ✘ **Java Remote Method Invocation (RMI):** Αποτελεί τον βασικό τρόπο με τον οποίο η Java καθορίζει την επικοινωνία μεταξύ κατανεμημένων αντικειμένων, όπως για παράδειγμα μεταξύ διαφορετικών αντικειμένων σε διαφορετικά μηχανήματα.
- ✘ **Java Naming and Directory Interface:** Η τεχνολογία JNDI χρησιμοποιείται για την πρόσβαση σε ονοματικά συστήματα και συστήματα διευθύνσεων. Από τον κώδικα μίας εφαρμογής η τεχνολογία JNDI μπορεί να χρησιμοποιηθεί για πολλαπλούς λόγους, όπως επικοινωνία με μία μονάδα λειτουργικότητας EJB, ή με άλλους πόρους στο δίκτυο, ή για την πρόσβαση σε δεδομένα που έχει αποθηκεύσει ο χρήστης σε μία άλλη υπηρεσία λογισμικού.
- ✘ **Java Database Connectivity:** Πρόκειται για ένα API για την πρόσβαση σε σχεσιακές βάσεις δεδομένων. Η αξία του JDBC έγκειται στο ότι η πρόσβαση σε οποιαδήποτε σχεσιακή βάση μπορεί να επιτευχθεί μέσω ενός και μόνο API.
- ✘ **Java Servlets:** Τα Servlets είναι δομικά στοιχεία που χρησιμοποιούνται για να επεκτείνουν τη λειτουργικότητα ενός Εξυπηρετητή Διαδικτύου. Τα Servlets είναι προσανατολισμένα στη λήψη αιτήσεων από κάποια εφαρμογή πελάτη, όπως ένας Φυλλομετρητής Διαδικτύου, και στην παροχή της απόκρισης στην ίδια εφαρμογή πελάτη. Το γεγονός αυτό καθιστά τα Servlets ιδανικά για την εκτέλεση λειτουργιών που αφορούν το Διαδίκτυο, όπως η κατασκευή μίας HTML διεπαφής.
- ✘ **JavaServer Pages:** Η τεχνολογία των JSP είναι παρόμοια με αυτή των Servlets. Στην πραγματικότητα, ο κώδικας των JSP μεταγλωττίζεται σε Servlets. Η μεγαλύτερη διαφορά μεταξύ των JSP και των Servlets είναι ότι ο κώδικας των JSP δεν αποτελείται από καθαρή Java, αλλά επικεντρώνεται περισσότερο σε θέματα εμφάνισης και αισθητικής.
- ✘ **JavaServer Faces:** Πρόκειται για μια τεχνολογία που παρέχει ένα πλαίσιο εργασίας διεπαφής χρήστη (user interface framework) για την ανάπτυξη διαδικτυακών εφαρμογών με την παροχή γραφικών εργαλείων και τη δυνατότητα σχηματοποίησης των δεδομένων διαφορετικών δηλωτικών γλωσσών (markup languages) και HTML τεχνολογιών.
- ✘ **Java Transaction API:** Το Java Transaction API υπαγορεύει ένα πρότυπο περιβάλλον διεπαφής για την οροθέτηση των συναλλαγών. Η αρχιτεκτονική της πλατφόρμας Java EE επιτρέπει την ύπαρξη ενός αυτοματοποιημένου τρόπου επικύρωσης (commit) και αποζημίωσης των συναλλαγών (rollback compensation).

- ✘ **SOAP with Attachments API for Java:** Πρόκειται για ένα χαμηλού επιπέδου API από το οποίο εξαρτώνται οι τεχνολογίες των Διαδικτυακών Υπηρεσιών και της καταγραφής των XML εφαρμογών (XML registry).

Η πλατφόρμα J2EE εγκαθίσταται μαζί με το περιβάλλον ανάπτυξης NetBeans IDE.

4.3.2 GlassFish ESB

4.3.2.1 Χαρακτηριστικά - Αρχιτεκτονική

Προκειμένου να επιτύχουμε την ανάπτυξη των υπηρεσιών, τη δημιουργία των ροών εργασίας στα πλαίσια της τεχνολογίας που παρέχει η BPEL, την ενορχήστρωση των διαδικτυακών υπηρεσιών και την αποθήκευση των επιχειρηματικών διαδικασιών σε ένα είδος ταμειυτήρα, οδηγηθήκαμε στην επιλογή της τεχνολογίας του GlassFish ESB. Το GlassFish ESB αποτελεί ουσιαστικά μια προσπάθεια ανοιχτού κώδικα (open source) [18] για τη δημιουργία ενός διαύλου υπηρεσιών (ESB) για την ενοποίηση, τη διαλειτουργικότητα και την αποδοχή του φαινομένου της ετερογένειας των συστημάτων μέσα από την υιοθέτηση αρχιτεκτονικών λύσεων που βασίζονται στην προσανατολισμένη στις Υπηρεσίες αρχιτεκτονική.

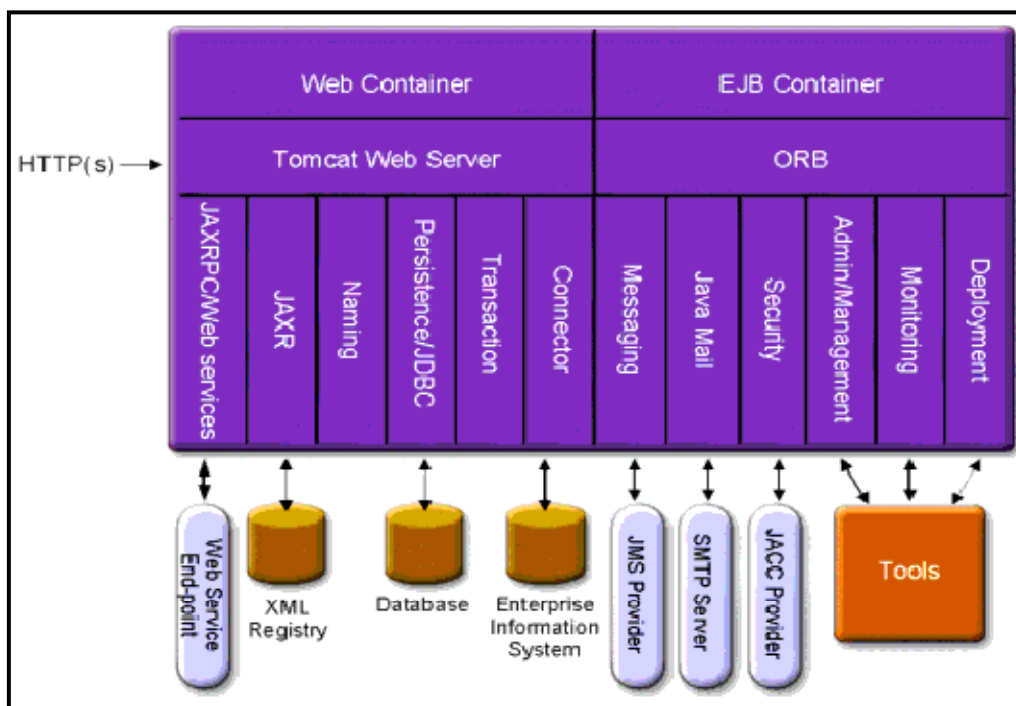
Το GlassFish ESB είναι μια πλατφόρμα δομημένη πάνω σε έναν επεκτάσιμο Java πυρήνα και στηρίζεται στην συμβολή και συνεισφορά κώδικα και προγραμματιστικών λύσεων από προγραμματιστές υπό την εμπορική υποστήριξη της Sun Microsystems. Επομένως παρέχει τη δυνατότητα περαιτέρω επέκτασης μέσα από την ενσωμάτωση εφαρμογών παραγωγής κώδικα για τον εμπλουτισμό της πλατφόρμας με περισσότερα και σταθερότερα εργαλεία. Το GlassFish ESB προσφέρει μια ευέλικτη και ελαφριά πλατφόρμα ESB στο πλαίσιο του open source ερευνητικού έργου *OpenESB* και στηρίζεται στο περιβάλλον ανάπτυξης Java *NetBeans 6.1* και στον *Sun GlassFish Enterprise Server* για την εκτέλεση και τη διαχείριση των διαλειτουργικών επιχειρησιακών εφαρμογών. Παράλληλα, υποστηρίζει πρότυπα διαλειτουργικότητας (WS-*, SOAP και ετερογενή XML πρότυπα) και δίνει βάση στην τεχνολογία του JBI (Java Business Integration) ως μέσου ενοποίησης και ανάπτυξης σύνθετων επιχειρησιακών εφαρμογών. Ας εξετάσουμε ένα-ένα τα χαρακτηριστικά και δομικά στοιχεία του GlassFish ESB.

4.3.2.1.1 GlassFish Application Server

Ο εξυπηρετητής Sun GlassFish Enterprise Server παρέχει τα θεμέλια για την ανάπτυξη και την εκτέλεση Java EE επιχειρηματικών τεχνουργημάτων - εφαρμογών (artifacts). Παρέχει υπηρεσίες προστιθέμενης αξίας (value-added) για τη διαχείριση, την επίβλεψη (monitoring) και την ομαδοποίηση των εφαρμογών (clustering), τη μελέτη συμπτωμάτων (diagnostics) και τη διαχείριση συναλλαγών. Συγχρόνως, επιφέρει σημαντικές μεταβολές στα ήδη υπάρχοντα χαρακτηριστικά και βελτιώνει την αποτελεσματικότητα και επάρκεια ιδιαίτερα κρίσιμων τεχνολογικών στοιχείων, τα σημαντικότερα από τα οποία είναι [17]:

- ✘ **Στοιχεία Προγραμματιστή (Developer Profile):** Το συγκεκριμένο στοιχείο επιτρέπει στον εξυπηρετητή GlassFish να καταναλώνει χαμηλό ποσοστό πόρων, να κατανέμει με ορθολογιστικό τρόπο τους πόρους και να εξασφαλίζει γρήγορη εκκίνηση. Πρόκειται για έναν από τους γρηγορότερους εμπορικούς εξυπηρετητές.
- ✘ **Ευκολία κατά την Ανάπτυξη Προγραμμάτων:** Επιτρέπει την ύπαρξη νέων στοιχείων για τον προγραμματισμό σε Java SE και μειώνει στο ελάχιστο τη συγγραφή προγραμματιστικού κώδικα.
- ✘ **GlassFish Update Center:** Επιτρέπει την εύκολη και άμεση πρόσβαση στις νέες τεχνολογίες που αναπτύσσονται με την ενημέρωση και τον εκσυγχρονισμό των χαρακτηριστικών.
- ✘ **Υποστήριξη και Ενσωμάτωση σε κάθε Περιβάλλον Ανάπτυξης:** Μπορεί να χρησιμοποιηθεί ως εξυπηρετητής των εφαρμογών σε όλα τα ενοποιημένα περιβάλλοντα ανάπτυξης Java εφαρμογών (NetBeans, Eclipse).
- ✘ **Υποστήριξη των JAX-WS 2.0 και JAX-B 2.0:** Επιτρέπει την υλοποίηση και εκτέλεση Διαδικτυακών Υπηρεσιών με δυναμικό χαρακτήρα υποστηρίζοντας τις τελευταίες εκδόσεις των συγκεκριμένων APIs.
- ✘ **Ενοποίηση και Διαλειτουργικότητα:** Επιτρέπει την υλοποίηση εφαρμογών ενοποίησης και διαλειτουργικότητας με βάση το αρχιτεκτονικό πρότυπο SOA.
- ✘ **Αρχιτεκτονική με Διακριτές Υπομονάδες (Modular Architecture):** Επιτρέπει τη συναρμολογησιμότητα και τη σύνδεση με διαφορετικά διακριτά συστήματα, με αποτέλεσμα να επιτυγχάνεται αντικειμενοστραφής συσχέτιση και επικοινωνία.

Τέλος, όσον αφορά την αρχιτεκτονική απλά αξίζει να αναφερθεί ότι στηρίζεται σε ένα σύνολο containers που παρέχουν σημαντικές υπηρεσίες φιλοξενίας, εκτέλεσης και ασφάλειας των εφαρμογών, ενώ συγχρόνως υφίσταται η δυνατότητα επικοινωνίας και σύνδεσης του εξυπηρετητή με τα εξωτερικά χαρακτηριστικά εργαλεία και συστήματα.

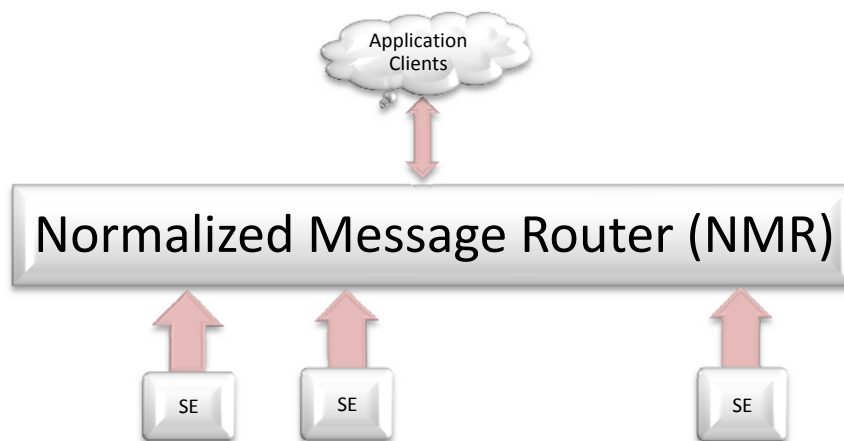


Σχήμα 8-4: Αρχιτεκτονική Προσέγγιση GlassFish ESB

4.3.2.1.2 Java Business Integration (JBI)

Είναι γνωστό - έχει αναλυθεί λεπτομερώς σε προηγούμενο κεφάλαιο - ότι στο σύγχρονο επιχειρηματικό κόσμο τα πληροφοριακά συστήματα που χρησιμοποιεί κάθε εταιρεία διαφέρουν και εμφανίζουν ετερογένεια ως προς τα τεχνολογικά και προγραμματιστικά εργαλεία λογισμικού που χρησιμοποιούν, με αποτέλεσμα να δημιουργούνται προβλήματα κατά την προσπάθεια επικοινωνίας μεταξύ διαφορετικών συστημάτων. Στα πλαίσια υποστήριξης της ετερογένειας και αποδοχής της ως αναγκαίου φαινομένου, προτάθηκε η ιδέα της δημιουργίας μιας προδιαγραφής ενοποίησης και διαλειτουργικότητας που να στηρίζεται σε τεχνολογικά αποδεκτά πρότυπα ανεξάρτητα από την πλατφόρμα υλοποίησης των εφαρμογών. Το αποτέλεσμα της προσπάθειας ήταν η πρόταση της προδιαγραφής *Java Business Integration Specification Request* (JSR 208), που έγινε αποδεκτή και έφερε στην επιφάνεια την υιοθέτηση των JBI στοιχείων. Το JSR 208 [27] πρότεινε ένα πρότυπο πλαίσιο για την επίτευξη ενοποίησης και ολοκλήρωσης παρέχοντας ένα πρότυπο σύνολο από διεπαφές παρόχων υπηρεσιών (service provider interfaces SPIs), ώστε να αντιμετωπιστούν τα προβλήματα της προηγούμενης προσέγγισης του EAI (Enterprise Application Integration) εξυπηρετητή.

Το Java Business Integration αποτελεί συνεπώς ένα Java πρότυπο για τη δόμηση ενοποιημένων επιχειρηματικών συστημάτων με βάση τη γραμμή της SOA αρχιτεκτονικής. Καθορίζει ένα περιβάλλον για την plug-in σύνδεση μονάδων λογισμικού που αλληλεπιδρούν χρησιμοποιώντας ένα υπηρεσιοστραφές πρότυπο με βάση τη γλώσσα WSDL 2.0 μέσω ενός διαύλου επικοινωνίας και μεταφοράς δεδομένων και υπηρεσιών.



Σχήμα 8-5: Αρχιτεκτονική Προσέγγιση GlassFish ESB

Στο επίκεντρο του αρχιτεκτονικού προτύπου που προκύπτει, βρίσκεται ένας διάυλος μεταφοράς μηνυμάτων και υπηρεσιών που εμφανίζει παρόμοια συμπεριφορά με τον ESB διάυλο και ονομάζεται Κανονικοποιημένος Δρομολογητής Μηνυμάτων (Normalized Message Router - NMR). Στην πραγματικότητα ο NMR πρόκειται για

μια υλοποιήσιμη μορφή Διαύλου Υπηρεσιών (ESB) με διαφορετική ονομασία. Οι μονάδες λογισμικού που συνδέονται με τον διάυλο υπηρεσιών ονομάζονται Μηχανές Υπηρεσιών (Service Engines) και περιγράφονται στο πρότυπο JSR 208 ως στοιχεία που παρέχουν επιχειρηματική λογική και υπηρεσίες μετασχηματισμού σε άλλα στοιχεία. Επομένως, ο NMR δρομολογητής είναι υπεύθυνος για την αποδοχή και στη συνέχεια τη μεταφορά – δρομολόγηση των μηνυμάτων σε κανονικοποιημένη μορφή (σε μορφή WSDL) στην κατάλληλη Μηχανή Υπηρεσιών για επεξεργασία. Με αυτόν τον τρόπο πετυχαίνεται η απαιτούμενη ευελιξία και χαλαρή σύζευξη και προστίθεται αξία στις επιχειρηματικές εφαρμογές.

Πέραν των Μηχανών Υπηρεσιών που συνδέονται στον NMR δρομολογητή για την μεταφορά και την εκτέλεση της επιχειρηματικής λογικής, ορίζονται στο πρότυπο JSR 208 και οι binding παράγοντες που συνδέονται στον NMR και προσφέρουν ανεξαρτησία και αυτονομία ως προς τη μεταφορά δεδομένων και μηνυμάτων στον δρομολογητή και τις Service Engines. Ο ρόλος των binding components είναι η απομόνωση των πρωτοκόλλων επικοινωνίας από τους JBI containers, ώστε οι Μηχανές Υπηρεσιών να μην εμφανίζουν ούτε ελάχιστο βαθμό σύζευξης με την υποδομή της επικοινωνίας.

Συνεπώς, κατά τη δημιουργία μιας επιχειρηματικής εφαρμογής που περιέχει μια σειρά από σύνθετες επιχειρηματικές διαδικασίες στο περιβάλλον ανάπτυξης NetBeans, η εφαρμογή βρίσκεται σε μορφή και εκτελείται πάντα ως μονάδα που εμφανίζει τη λειτουργικότητα μιας μονάδας λογισμικού Επιχειρηματικής Ενοποίησης (Java Business Integration) στο περιβάλλον εκτέλεσης JBI Runtime Environment που παρέχεται από τον GlassFish Application Server.

4.3.2.1.3 NetBeans 6.1

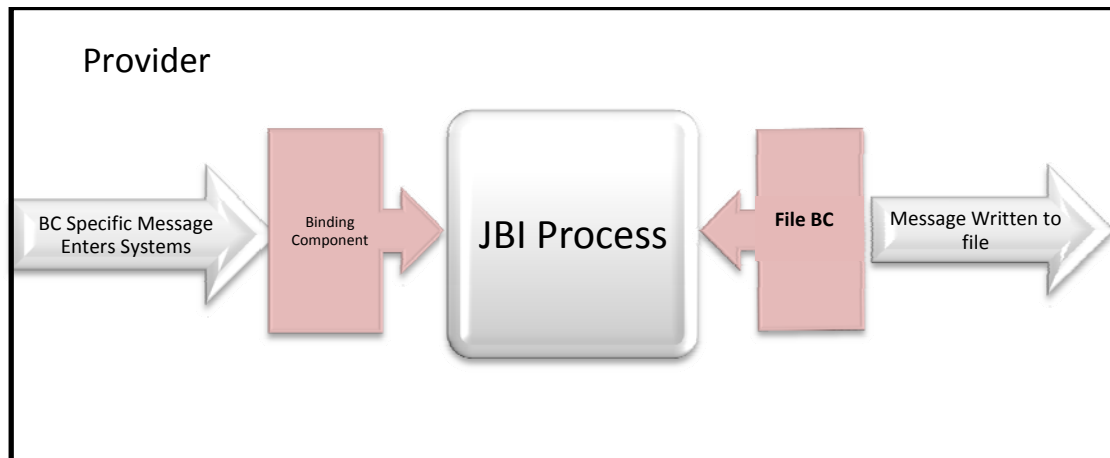
Το NetBeans 6.1 [32] αποτελεί ένα δημοφιλές περιβάλλον ανάπτυξης εφαρμογών λογισμικού και είναι γνωστό ότι στηρίζεται στη γλώσσα προγραμματισμού Java. Αποτελεί το βασικό περιβάλλον ανάπτυξης προγραμματιστικού κώδικα και πολύπλοκων εφαρμογών με βάση την πλατφόρμα JavaEE στο πλαίσιο του GlassFish ESB. Οι εφαρμογές που μπορούν να αναπτυχθούν στο περιβάλλον ανάπτυξης NetBeans περιλαμβάνουν από απλές Java εφαρμογές επιπέδου κλάσεων και κονσόλας μέχρι πολύπλοκες διαδικτυακές και επιχειρηματικές εφαρμογές με γραφικό περιβάλλον και διαδικτυακές υπηρεσίες. Συγχρόνως – στο επίπεδο του ενδιαφέροντος της διπλωματικής - υποστηρίζει την υιοθέτηση της SOA αρχιτεκτονικής και την ανάπτυξη πολύπλοκων επιχειρησιακών εφαρμογών με βάση τις αρχές της συγκεκριμένης αρχιτεκτονικής προσέγγισης και τις προδιαγραφές της BPEL. Επομένως γίνεται κατανοητό ότι αποτελεί το περιβάλλον στο οποίο στηρίχθηκε η ανάπτυξη, η υλοποίηση, η ενορχήστρωση, η εκτέλεση και η αποθήκευση/δημοσίευση των διαδικτυακών υπηρεσιών και των σύνθετων επιχειρηματικών διαδικασιών.

Το περιβάλλον ανάπτυξης NetBeans υποστηρίζει την αρχιτεκτονική προσέγγιση του Διαύλου Υπηρεσιών, με αποτέλεσμα να έχει ενσωματώσει σε μεγάλο βαθμό το σύνολο των Service Engines και των Binding Components που αποτελούν αναπόσπαστο μέρος του GlassFish Application Server, ενώ συγχρόνως υπάρχει η δυνατότητα ανάπτυξης νέων δομικών στοιχείων λόγω του ανοιχτού κώδικα

ερευνητικού έργου GlassFish ESB στο οποίο στηρίζεται. Παρακάτω περιγράφονται οι Μηχανές Υπηρεσιών που υποστηρίζει το NetBeans 6.1 [25]:

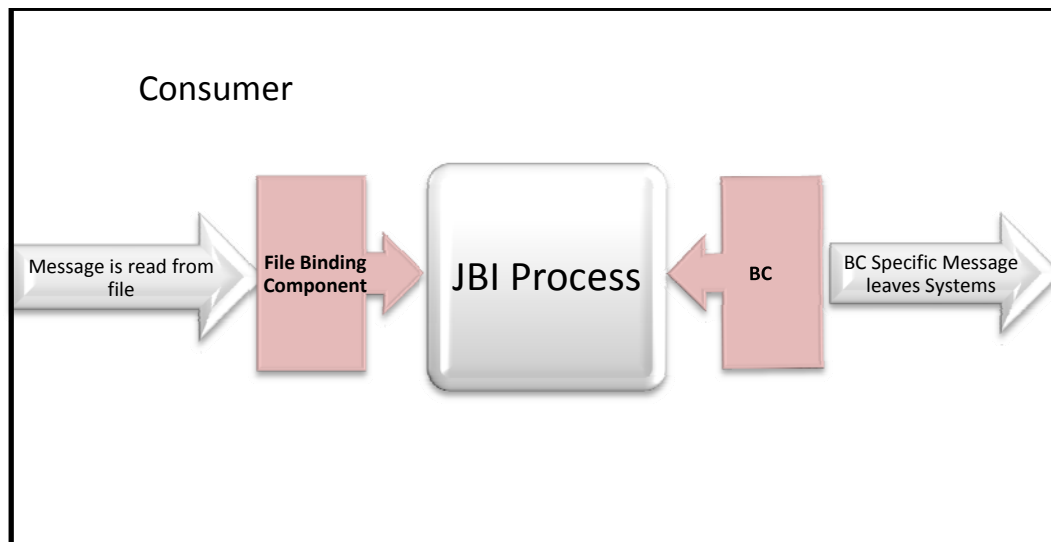
- ✘ **BPEL Service Engine:** Η Μηχανή Υπηρεσιών BPEL αποτελεί τον παράγοντα του NMR που είναι υπεύθυνος για την επίτευξη της σύνθεσης και της ενορχήστρωσης επιχειρηματικών διαδικασιών με βάση την προδιαγραφή WS-BPEL 2.0. Επιτρέπει την ανάπτυξη και εκτέλεση ροών εργασίας που δομούνται από την σύνθεση διαδικτυακών υπηρεσιών.
- ✘ **Java EE Service Engine:** Η Μηχανή Υπηρεσιών Java EE λειτουργεί ως γέφυρα μεταξύ των επιχειρηματικών εφαρμογών σε Java EE και του JBI container επιτρέποντας στις διαδικτυακές υπηρεσίες να καταναλωθούν από τους JBI παράγοντες. Χωρίς την ύπαρξη της Μηχανής Υπηρεσιών Java EE, οι JBI παράγοντες θα έπρεπε να εκτελούν τις Διαδικτυακές Υπηρεσίες μέσω απομακρυσμένων κλήσεων (remote calls) αντί μέσω εσωτερικής (εντός της διαδικασίας) επικοινωνίας που επιτυγχάνεται από την Java EE Service Engine. Η Μηχανή Υπηρεσιών Java EE επιτυγχάνει σημαντικά οφέλη κατά την εκτέλεση επιχειρηματικών εφαρμογών :
 - ✓ Αυξημένη απόδοση
 - ✓ Υποστήριξη των συναλλαγών
 - ✓ Υποστήριξη πρωτοκόλλων και μεθόδων ασφάλειας.
- ✘ **SQL Service Engine:** Η Μηχανή Υπηρεσιών SQL επιτρέπει την εκτέλεση των SQL ερωτημάτων και προτάσεων έναντι σχεσιακών βάσεων δεδομένων και την επιστροφή - μεταφορά των αποτελεσμάτων στην εφαρμογή για περαιτέρω επεξεργασία. Στο περιβάλλον ανάπτυξης NetBeans, χρησιμοποιείται η SQL μονάδα λογισμικού για την αλληλεπίδραση με την Μηχανή Υπηρεσιών SQL. Η SQL μονάδα λογισμικού αποτελείται από τα παρακάτω δομικά στοιχεία :
 - ✓ ένα configuration xml αρχείο που ονομάζεται connectivityInfo.xml και περιέχει πληροφορίες για τη σύνδεση με τη βάση δεδομένων στην οποία αναφέρονται οι SQL προτάσεις,
 - ✓ ένα ή περισσότερα SQL αρχεία που περιλαμβάνουν διακριτές SQL εντολές,
 - ✓ ένα WSDL αρχείο που περιγράφει τις SQL λειτουργίες.
- ✘ **Intelligent Event Processing Service Engine:** Η Μηχανή Υπηρεσιών Επεξεργασίας Ευφυών Γεγονότων (IEP) είναι υπεύθυνη για την ανάγνωση δεδομένων εισόδου και την επεξεργασία τους σε τέτοια μορφή που να μπορεί να χρησιμοποιηθεί για διαφορετικούς σκοπούς όπως για έκθεση στατιστικών αποτελεσμάτων ή για πληροφορίες επιχειρηματικής ευφυΐας.
- ✘ **XSLT Service Engine:** Η Μηχανή Υπηρεσιών XSLT επιτρέπει το μετασχηματισμό των XML εγγράφων από τη μια μορφή σε μια άλλη χρησιμοποιώντας τα πρότυπα μορφοποίησης τεχνολογιών XSL (XSL stylesheets). Η συγκεκριμένη μηχανή υπηρεσιών επιτρέπει στους XSL μετασχηματισμούς να εκτελεστούν ως διαδικτυακές υπηρεσίες που στη συνέχεια μπορούν να χρησιμοποιηθούν από εξωτερικούς πελάτες.

Όσον αφορά τους Binding components που υποστηρίζει η πλατφόρμα ανάπτυξης NetBeans 6.1, οι κυριότεροι αναφέρονται και αναλύονται παρακάτω:



Σχήμα 8-6: File BC ως Provider

- ✘ **File Binding Component:** Ο File Binding Παράγοντας παρέχει ένα μηχανισμό επικοινωνίας για τους JBI παράγοντες προκειμένου να αλληλεπιδρούν με το σύστημα αρχείων. Μπορεί να λειτουργήσει τόσο ως Πάροχος ελέγχοντας για νέα αρχεία προς επεξεργασία, όσο και ως Καταναλωτής παράγοντας στην έξοδο αρχεία που απευθύνονται σε άλλες διαδικασίες.

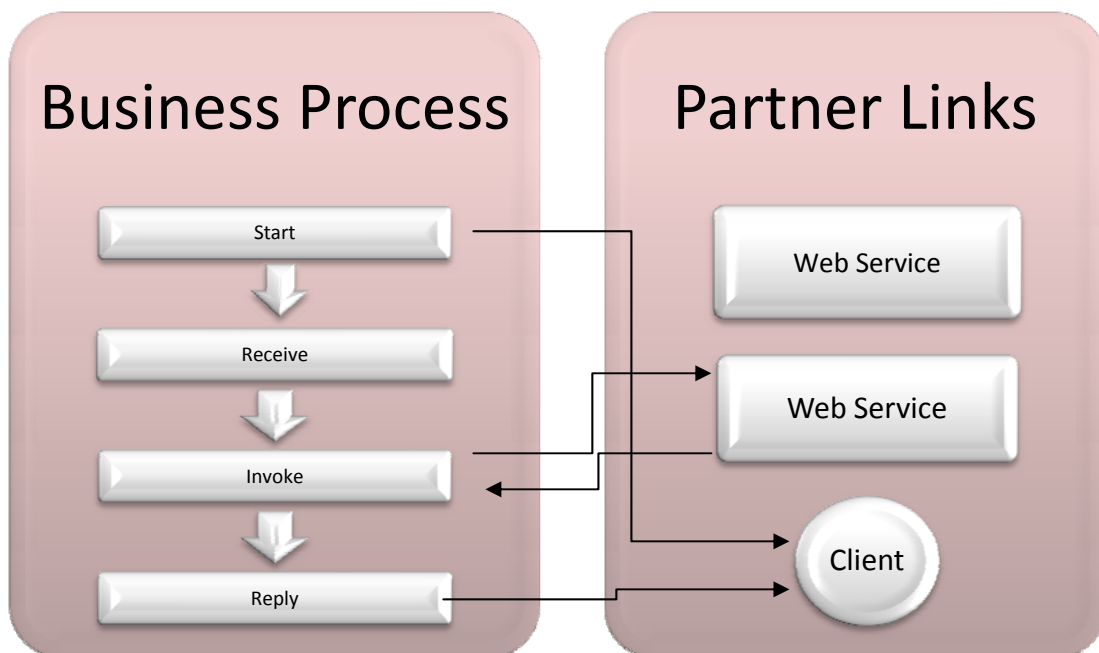


Σχήμα 8-7: File BC ως Consumer

- ✘ **SMTP Binding Component:** Ο SMTP Binding Παράγοντας παρέχει υπηρεσίες email στον JBI εξυπηρετητή και μπορεί να λειτουργήσει με διττό τρόπο τόσο ως Πάροχος που δέχεται εισερχόμενα SMTP μηνύματα όσο και ως Καταναλωτής που στέλνει μηνύματα σε εξωτερικές email διευθύνσεις.

- ✘ **FTP Binding Component:** Ο FTP Binding Παράγοντας παρέχει FTP υπηρεσίες μεταφοράς στους JBI containers και επιτρέπει την αποστολή και αποδοχή μηνυμάτων μέσω του FTP πρωτοκόλλου.
- ✘ **SOAP Binding Component:** Ο SOAP Binding Παράγοντας επιτρέπει στα μηνύματα μεταξύ των JBI στοιχείων να αποστέλλονται και να λαμβάνονται χρησιμοποιώντας SOAP πρωτόκολλο πάνω από HTTP και HTTPS. Ο συγκεκριμένος παράγοντας υποστηρίζει τα σχήματα κωδικοποίησης RPC Literal, RPC Encoded και Document Literal. Σε περιπτώσεις όπου ο JBI container τρέχει μέσα σε firewall, ο SOAP Binding Παράγοντας μπορεί να διαμορφωθεί και να μετασχηματιστεί ώστε να χρησιμοποιεί έναν proxy εξυπηρετητή που να επιτρέπει την πρόσβαση και την κλήση απομακρυσμένων διαδικτυακών υπηρεσιών.
- ✘ **JDBC Binding Component:** Ο JDBC Binding Παράγοντας χρησιμοποιείται σε περίπτωση που απαιτείται πρόσβαση και αλληλεπίδραση με μια Βάση Δεδομένων ως μέλος μιας εφαρμογής. Μπορεί να έχει το ρόλο του Καταναλωτή ή του Παρόχου ανάλογα με τη λειτουργία που επιτελεί.
- ✘ **JMS Binding Component:** Ο JMS Binding Component επιτρέπει στους JBI παράγοντες να επικοινωνούν μέσω JMS μηνυμάτων και ουρών μηνυμάτων.

Τέλος, για την δημιουργία και τη σύνθεση επιχειρηματικών διαδικασιών με βάση τις προδιαγραφές της γλώσσας BPEL, το περιβάλλον ανάπτυξης NetBeans παρέχει ένα ιδιαίτερα εύχρηστο και κατανοητό γραφικό περιβάλλον που απλοποιεί το σχεδιασμό των επιχειρηματικών διαδικασιών και διευκολύνει το χρήστη περιορίζοντας την ανάγκη για τη σε βάθος γνώση της XML σύνταξης και της WSDL γλώσσας. Το γραφικό περιβάλλον του NetBeans 6.1 για το σχεδιασμό BPEL διαδικασιών ονομάζεται BPEL Designer.



Σχήμα 8-8: Μορφή Ροής Επιχειρηματικών Διαδικασιών BPEL

Στο πλαίσιο της προγραμματιστικής σύνταξης μιας επιχειρηματικής διαδικασίας σε περιβάλλον BPEL από μια περισσότερη αφαιρετική και απλοποιημένη σκοπιά, μια επιχειρηματική διαδικασία αρχίζει με την ύπαρξη μιας δραστηριότητας λήψης – αποδοχής μηνύματος (receive), συνεχίζει με την κλήση (invoke) μιας εξωτερικής διαδικτυακής υπηρεσίας που είναι η αφαιρετική υπόσταση της επιχειρηματικής λογικής μιας μονάδας λογισμικού και ολοκληρώνεται με την αποστολή του αποτελέσματος στον πελάτη (client) που σχετίζεται με την διαδικασία. Μια επιχειρηματική διαδικασία τυπικά αλληλεπιδρά με μια ή περισσότερες διαδικτυακές υπηρεσίες, ενώ και η ίδια αποτελεί συγχρόνως υπηρεσία που μπορεί να κληθεί ως μέλος της ροής μιας άλλης διαδικασίας. Οι διαδικτυακές υπηρεσίες και οι πελάτες που συνεργάζονται με την επιχειρηματική διαδικασία και αποτελούν αναπόσπαστο τμήμα της λειτουργίας της ονομάζονται Υπηρεσίες Συνέταιροι (Partner Services).

Το περιβάλλον ανάπτυξης του BPEL Designer παρέχει τη δυνατότητα σχεδιασμού των επιχειρηματικών διαδικασιών και επίβλεψης των δομικών τους στοιχείων με δύο διαφορετικές όψεις εργασίας που συνεχώς συγχρονίζονται :

- ✱ **Όψη Σχεδιασμού (Design View)** : Η Όψη Σχεδιασμού έχει τη μορφή διαγράμματος και ουσιαστικά αποτελεί ένα είδος γραφικού σχεδιασμού της ροής εκτέλεσης των εργασιών. Τα στοιχεία που θα χρησιμοποιηθούν για τη δόμηση της επιχειρηματικής διαδικασίας επιλέγονται από μια σύνολο - παλέτα δραστηριοτήτων (Activity Palette) και εισάγονται στη διαδικασία μέσω της drag-and-drop μεθόδου. Οι δραστηριότητες διακρίνονται σε βασικές (basic), δομημένες (structured) και εκείνες που αφορούν διαδικτυακές υπηρεσίες (web services).
- ✱ **Όψη Κώδικα (Source view)** : Η Όψη Κώδικα υποστηρίζει την προδιαγραφή της γλώσσας BPEL 2.0 στην ολότητα της σε αντίθεση με την Όψη Σχεδιασμού που αδυνατεί να προσαρτήσει και να αναπαραστήσει με γραφικό τρόπο κάποια προγραμματιστικά τμήματα. Στηρίζεται εξολοκλήρου σε σύνταξη XML κώδικα, με αποτέλεσμα να είναι δυσνόητη από τον άπειρο προγραμματιστή και να χρησιμοποιείται σε εξειδικευμένες και λεπτομερείς περιπτώσεις.

Επιπρόσθετο δομικό στοιχείο του BPEL Designer με ιδιαίτερα σημαντικό ρόλο για την δόμηση των επιχειρηματικών διαδικασιών είναι ο BPEL Mapper. Ο BPEL Mapper επιτρέπει την παραμετροποίηση και τη σύνταξη καθορισμένων λειτουργιών συγκεκριμένων στοιχείων της επιχειρηματικής διαδικασίας. Οι λειτουργίες αυτές σχετίζονται με την αντιστοίχιση των μεταβλητών εισόδου και εξόδου και τον καθορισμό των τιμών τους (δραστηριότητα assign), καθώς και την έκφραση και αποτίμηση των συνθηκών σε περίπτωση δομών ελέγχου (δραστηριότητες If, ElseIf, Repeat Until, ForEach). Συγχρόνως, μέσω του γραφικού περιβάλλοντος διεπαφής του BPEL Mapper δίνεται η δυνατότητα εκτέλεσης υπολογισμών με την ανάθεση λειτουργιών και συναρτήσεων μορφής XPath σε μεταβλητές και στοιχεία ή ιδιότητες που ακολουθούν συγκεκριμένο XSD σχήμα.

Τέλος, μπορούμε να προχωρήσουμε σε μια επισκόπηση των κυριότερων δραστηριοτήτων μιας επιχειρηματικής διαδικασίας με βάση τα στοιχεία που παρέχει ο BPEL Designer :

- ✘ **Receive:** Η δραστηριότητα λήψης (Receive) επιτρέπει στις επιχειρηματικές διαδικασίες να επιτελούν μια blocking αναμονή (αναμονή που παρεμποδίζει τη συνέχιση της διαδικασίας μέχρι τη λήψη μηνύματος απάντησης από τον σύνδεσμο συνεργασίας – Partner Link) και αποτελεί το συνηθέστερο τρόπο έναρξης μιας επιχειρηματικής διαδικασίας.
- ✘ **Invoke:** Η δραστηριότητα κλήσης (Invoke) επιτρέπει στις επιχειρηματικές διαδικασίες να καλούν τις λειτουργίες των διαδικτυακών υπηρεσιών με τις οποίες επικοινωνεί, μέσω της one-way ή της request-response μεθόδου.
- ✘ **Reply:** Η δραστηριότητα απάντησης (Reply) χρησιμοποιείται προκειμένου να επιστρέψει ένα μήνυμα από την διαδικασία στον σύνδεσμο που ξεκίνησε την διαδικασία στην περίπτωση της σύγχρονης επικοινωνίας. Σε περίπτωση ασύγχρονης επικοινωνίας χρησιμοποιείται η δραστηριότητα callback.
- ✘ **Assign:** Η δραστηριότητα ανάθεσης (Assign)

4.3.2.2 Εγκατάσταση

Η εγκατάσταση του GlassFish ESB είναι ιδιαίτερα απλή με γραφικό περιβάλλον που καθοδηγεί το χρήστη στα κατάλληλα βήματα που πρέπει να ακολουθήσει. Κατά την εγκατάσταση του λογισμικού, περιλαμβάνονται τα παρακάτω δομικά στοιχεία :

- ✘ **NetBeans 6.1**
- ✘ **GlassFish v2 Update Release 2 (UR2)**
- ✘ **OpenESB Core Components**
- ✘ **Java Business Integration (JBI) service engines**
- ✘ **Java Business Integration (JBI) binding components**
- ✘ **Java Business Integration (JBI) component tooling**

Η διαδικασία εγκατάστασης του GlassFish ESB που ακολουθήθηκε είναι η εξής:

1. Εγκατάσταση του Java Development Kit (JDK) τουλάχιστον της έκδοσης 1.6.0_03.
2. Ορισμός της μεταβλητής JAVA_HOME ώστε να περιέχει το μονοπάτι όπου είναι εγκατεστημένη η έκδοση της Java.
3. Εγκατάσταση των στοιχείων του GlassFish ESB ακολουθώντας το γραφικό περιβάλλον.

4.3.3 Ο εξυπηρετητής εφαρμογών JBoss AS

Ο JBoss Application Server είναι ένας πιστοποιημένος J2EE εξυπηρετητής εφαρμογών υλοποιημένος από την κοινότητα Open Source Software (OSS) και αποτελεί τη σημαντικότερη υλοποίηση από τη σουίτα των OSS εφαρμογών που συνθέτουν το σύστημα JBoss Enterprise Middleware System (JEMS). Ο JBoss AS βασίζεται σε ένα εξελιγμένο Aspect-Oriented πλαίσιο προγραμματισμού (AOP) που σχεδιάστηκε για να απλουστεύσει την ανάπτυξη J2EE-συμβατών εφαρμογών. Η

ισχυρή αλλά ταυτόχρονα ευέλικτη αρχιτεκτονική του JBoss AS, η υψηλή του απόδοση και αξιοπιστία καθώς και το μηδενικό κόστος εγκατάστασης και χρήσης του, τον έχουν καταστήσει μία από τις πιο δημοφιλείς επιλογές των προγραμματιστών, ανεξάρτητων παρόχων λογισμικού (ISV), και επιχειρήσεων. Είναι γεγονός ότι ο JBoss AS έχει ανακηρυχθεί ως ο πιο διαδεδομένος J2EE εξυπηρετητής εφαρμογών, με περισσότερα από 5 εκατομμύρια εγκαταστάσεις, ξεπερνώντας τις αντίστοιχες πλατφόρμες IBM Websphere και BEA Weblogic. Από τα βασικές υπηρεσίες που συντέλεσαν στην επιλογή του JBoss AS ως εξυπηρετητή του συστήματος ελέγχου λειτουργίας της κύριας μηχανής είναι οι ακόλουθες:

- ✘ **Υποστήριξη της τεχνολογίας EJB :** Περιλαμβάνονται απλουστευμένα EJB API για Session Beans, Entity Beans και Message Driven Beans, και υποστηρίζονται πλήρως τα Java Annotations.
- ✘ **Υπηρεσίες Διαδικτυακών Εφαρμογών :** Ο JBoss AS υποστηρίζει τις πιο πρόσφατες προδιαγραφές των JSP και Servlets και υλοποιεί επιπλέον τεχνολογίες για να ικανοποιήσει τις απαιτήσεις των σύγχρονων πολύπλοκων Διαδικτυακών εφαρμογών. Επιπλέον, ενσωματώνει τον Apache Tomcat 5.0, που αποτελεί την υλοποίηση αναφοράς του προτύπου JSP 2.0 & Servlet 2.4 Container. Η ενοποίηση των υπηρεσιών του Tomcat με τις υπηρεσίες του JBoss AS παρέχει ένα περιβάλλον υψηλής αξιοπιστίας, εύκολης κλιμάκωσης και απεριόριστης διαθεσιμότητας για την υποστήριξη των διαδικτυακών εφαρμογών.
- ✘ **Υπηρεσίες μόνιμης αποθήκευσης :** Ο JBoss AS ενσωματώνει την τεχνολογία Hibernate, που αποτελεί την πιο πρωτοπόρα και επιτυχημένη λύση στην αντιστοίχιση σχεσιακών δεδομένων με δομές του αντικειμενοστραφούς προγραμματισμού.
- ✘ **Υποστήριξη των Διαδικτυακών Υπηρεσιών :** Ένα από τα μεγαλύτερα άλματα που έγιναν με το πρότυπο J2EE 1.4 έναντι του προτύπου J2EE 1.3 ήταν η προσθήκη των standards που αφορούν τις Διαδικτυακές Υπηρεσίες, καθώς και η δυνατότητα πιο ολοκληρωμένης επεξεργασίας των XML αρχείων. Ο JBoss AS ενσωματώνει απόλυτα τις προδιαγραφές των Διαδικτυακών Υπηρεσιών και δίνει τη δυνατότητα στις εφαρμογές που χτίζονται πάνω του να αλληλεπιδρούν δυναμικά με τις Διαδικτυακές Υπηρεσίες χρησιμοποιώντας πρωτόκολλα επικοινωνίας όπως το Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), και XML.
- ✘ **Υπηρεσίες Ασφαλείας :** Ο JBoss AS υλοποιεί το μοντέλο ασφαλείας J2EE Authentication and Authorization Security Model (JAAS), αλλά επιπλέον υποστηρίζει εξελιγμένα μοντέλα ασφαλείας για ελεγχόμενη πρόσβαση σε πληροφορίες υψίστης σημασίας, ενώ επιτρέπει την ενσωμάτωση υπαρχόντων πλαισίων εφαρμογής ελεγχόμενης πρόσβασης που μπορεί ήδη να λειτουργούν στην επιχείρηση, μέσω των Pluggable Authentication Modules (PAM).
- ✘ **Διαθέσιμες κονσόλες και εργαλεία :** Ο JBoss AS διαθέτει κονσόλα με την οποία μπορεί να γίνει διαχείριση των ορισμένων από το χρήστη MBeans, να εξαχθούν στατιστικά αποτελέσματα για τη χρήση των Servlets, να οριστούν τα αρχεία διαμόρφωσης των EJB, να οριστεί η αποστολή ηλεκτρονικής ειδοποίησης προς τον διαχειριστή του συστήματος με βάση διαμορφωμένα όρια ασφαλούς λειτουργίας του εξυπηρετητή, να τερματιστεί η λειτουργία του εξυπηρετητή κ.α. Επιπλέον, η JBoss Subscription, δίνει τη δυνατότητα

πρόσβασης στο JBoss Operations Network που παρέχει ισχυρές λειτουργίες διαχείρισης και παρακολούθησης της λειτουργίας του εξυπηρετητή.

Τέλος, το ισχυρότερο κίνητρο επιλογής του JBoss AS είναι ότι αποτελεί μέλος των συστημάτων της JBoss Enterprise Middleware, μίας ολοκληρωμένης σουίτας προϊόντων που στοχεύει στο να παρέχει μία πλατφόρμα ανάπτυξης επιχειρησιακών εφαρμογών ανεξαρτήτως λειτουργικού συστήματος και να διευκολύνει την ανάπτυξη των εφαρμογών αναλαμβάνοντας το χειρισμό διαδικασιών όπως αποθήκευση, ολοκληρωμένη εκτέλεση διεργασιών, επικοινωνία και καταμερισμό του φόρτου κατά μήκος ενός δικτύου. Ο JBoss είναι ο application server στον οποίο εγκαθίστανται και εκτελούνται τα Rule Team Server, Rule Execution Server, and Rule Scenario Manager.

4.3.4 ILOG JRules

4.3.4.1 Χαρακτηριστικά - Αρχιτεκτονική

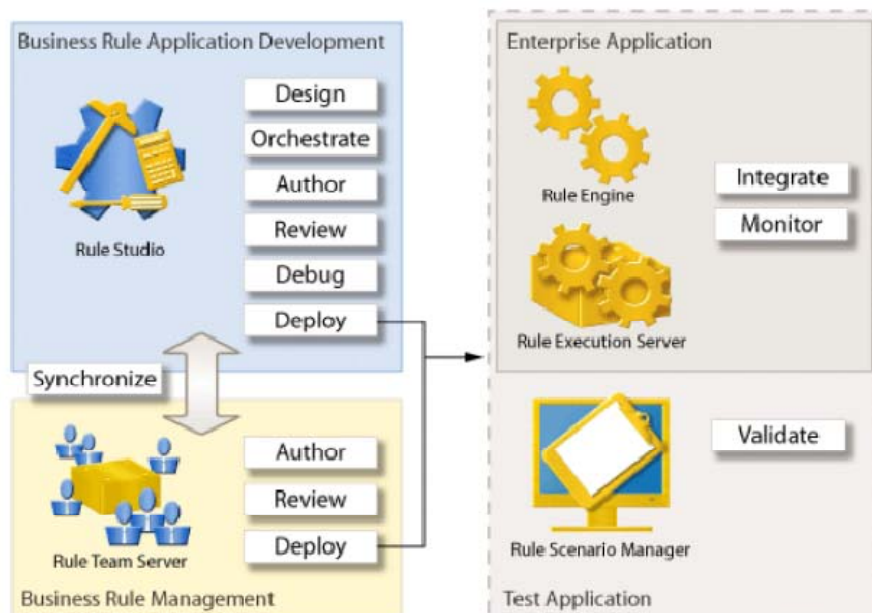
Το ILOG JRules είναι ένα πλήρες σύστημα διαχείρισης και υλοποίησης επιχειρηματικών κανόνων με βάση τη γλώσσα προγραμματισμού Java. Πρόκειται για μια από τις ταχύτερες και περισσότερο αξιόπιστες μηχανές της αγοράς και παρέχει τη δυνατότητα ενσωμάτωσης της μηχανής σε ήδη υπάρχοντα συστήματα με την ανάπτυξη Java εφαρμογών σε οποιοδήποτε περιβάλλον ανάπτυξης έχει επιλεγεί. Η αρχιτεκτονική ενοποιεί και περιλαμβάνει ένα σύνολο από περιεκτικά προγραμματιστικά περιβάλλοντα για τη συγγραφή και τη διαχείριση επιχειρηματικών κανόνων, την ανάπτυξη και την εκτέλεση των επιχειρησιακών εφαρμογών με τους κανόνες που έχουν οριστεί, και την προσομοίωση και έλεγχο της πραγματικής λειτουργίας των επιχειρηματικών κανόνων. Παράλληλα, οι μηχανικοί λογισμικού έχουν τη δυνατότητα να συνδυάσουν αντικειμενοστραφείς τεχνικές προγραμματισμού με την ανάπτυξη επιχειρηματικών κανόνων, με αποτέλεσμα την ενσωμάτωση επιχειρηματικής λογικής σε υπάρχουσες εφαρμογές. Επιπλέον το ILOG JRules παρέχει τη δυνατότητα στους επιχειρηματικούς και στρατηγικούς αναλυτές, που είναι υπεύθυνοι για τη χάραξη της πολιτικής της επιχείρησης, να διαχειρίζονται και να μεταβάλλουν τους κανόνες χωρίς να απαιτείται προγραμματιστική και υπολογιστική γνώση.

Η έκδοση ILOG JRules 6.7 που χρησιμοποιήσαμε παρέχει ένα ολοκληρωμένο BRMS σε τέσσερις δομικές μονάδες. Με βάση την αρχιτεκτονική του JRules επιτυγχάνεται η δημιουργία ενός κύκλου BRM που είναι ανεξάρτητος από τον κύκλο ανάπτυξης της εφαρμογής, αλλά υποστηρίζει το συγχρονισμό μεταξύ της ανάπτυξης της εφαρμογής και της διαχείρισης των κανόνων παραγωγής. Οι εφαρμογές που βασίζονται στο σύστημα ILOG JRules 6.7 μπορούν να αναπτυχθούν από αυτόνομες desktop εφαρμογές, ως mid-tier υπηρεσίες στα πλαίσια μίας SOA αρχιτεκτονικής. Τα υποσυστήματα που συνθέτουν το ILOG JRules είναι:

- ✘ **Rule Team Server:** Αποτελεί έναν εξυπηρετητή διαχείρισης κανόνων και μέσο αποθήκευσης των κανόνων, που υποστηρίζει μέσω ενός διαδικτυακού περιβάλλοντος τη συνεργατική σύνθεση, τη διαχείριση, τον έλεγχο και την εγκατάσταση των κανόνων. Ο Rule Team Server επιτρέπει τη κλιμακούμενη -

διαβαθμισμένη διαχείριση και πρόσβαση στους κανόνες με βάση τους ρόλους των χρηστών. Οι άδειες είναι επιμερισμένες σε τέτοιο βαθμό ώστε να υποστηρίζουν πολλαπλά project για μεγάλες ομάδες προγραμματιστών και ειδικών. Οι βαθμίδες διαχείρισης μπορούν να οριστούν με βάση έντεκα διαφορετικούς ρόλους, με τις άδειες που σχετίζονται με κάθε ρόλο να μπορούν να διαχωριστούν περαιτέρω σε πέντε ή επτά επίπεδα. Από την άλλη, το JRules Repository διαιρείται σε δύο επιμέρους διατάξεις αποθήκευσης, μία για τους ειδικούς της επιχειρησιακής λογικής και μία για τους προγραμματιστές, που διατηρούνται συγχρονισμένες μέσω του Rule Execution Server. Είναι σχεδιασμένο και απευθύνεται σε μεγάλο βαθμό σε ειδικούς επιχειρηματικής πολιτικής χωρίς τεχνική κατάρτιση και επιτρέπει την ύπαρξη ιστορικού επιχειρηματικών κανόνων και προηγούμενων εκδόσεων.

- ✘ **Rule Studio:** Αποτελεί ένα ενοποιημένο περιβάλλον ανάπτυξης εφαρμογών με βάση σύνολα επιχειρηματικών κανόνων και είναι πλήρως ενσωματωμένο στο περιβάλλον Eclipse. Είναι προσανατολισμένο στη χρήση από μηχανικούς της πληροφορικής για την ανάπτυξη, αποσφαλμάτωση και εγκατάσταση εφαρμογών βασισμένων σε κανόνες. Παρέχει εργαλεία ανάπτυξης κανόνων που δίνουν τη δυνατότητα δημιουργίας και διαχείρισης του λεξιλογίου των κανόνων, των επιχειρηματικών και εκτελέσιμων μοντέλων αντικειμένων (Business Object Models and Execution Object Models), που είναι τα βασικά συστατικά προγραμματισμού της μηχανής, και των μεθόδων ταυτοποίησης, ελέγχου και ανάπτυξης τους. Μέσω του Rule Studio είναι δυνατή η εγκατάσταση των κανόνων στον Rule Execution Server και διευκολύνεται η συνεργασία με τους συγγραφείς των επιχειρησιακών κανόνων μέσω της ενοποίησης του συστήματος με τον Rule Team Server.



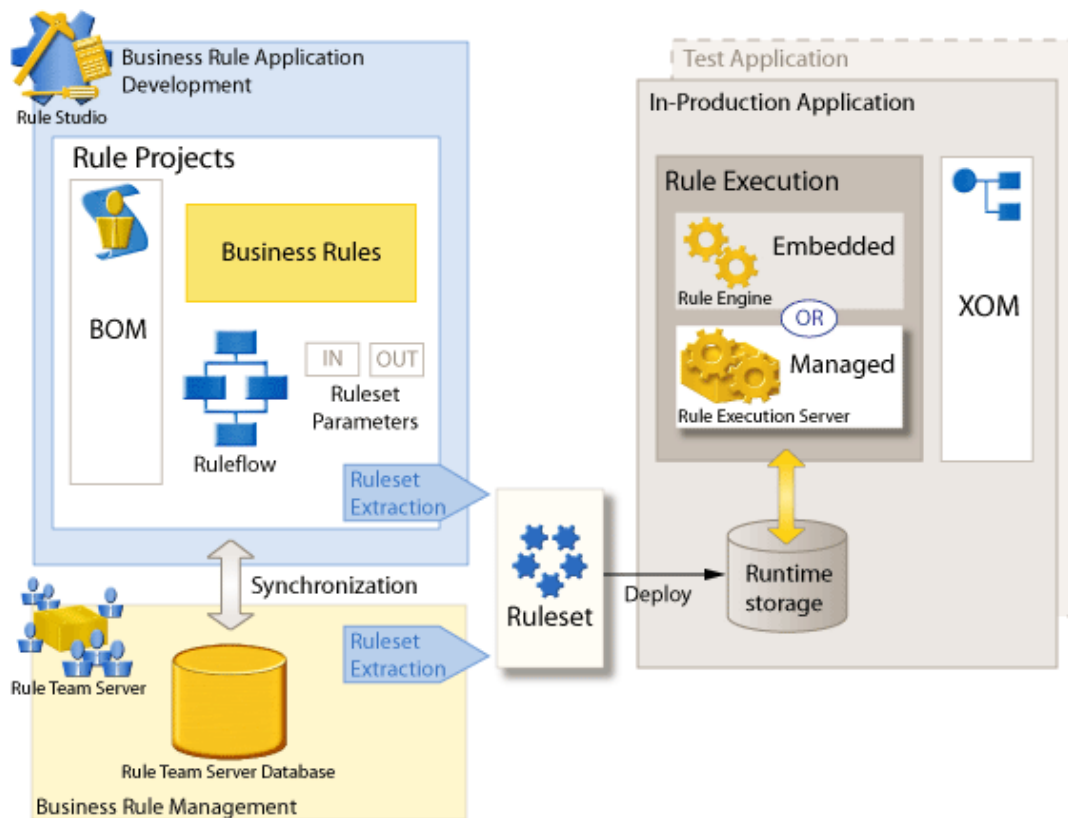
Σχήμα 8-9: Αρχιτεκτονική Προσέγγιση και Υποσυστήματα του ILOG JRules

- ✘ **Rule Execution Server:** Είναι ένα περιβάλλον εκτέλεσης των κανόνων, πλήρως συμβατό με τα πρότυπα J2SE και J2EE, για την εγκατάσταση SOA υπηρεσιών διαχείρισης κανόνων στους βασικότερους εξυπηρετητές

εφαρμογών από την IBM, BEA, JBoss, Oracle και Apache. Ο εξυπηρετητής περιλαμβάνει στοιχεία για τη σύγχρονη, ασύγχρονη και τη σχετική με τις Διαδικτυακές Υπηρεσίες κλήση των επιχειρησιακών κανόνων, και παρέχει διαδικτυακή κονσόλα διαχείρισης. Ο εξυπηρετητής συνεργάζεται με το Rule Studio και τον Rule Team Server για την εγκατάσταση των κανόνων τόσο από μηχανικούς όσο και από ειδικούς της επιχειρησιακής λογικής χωρίς τεχνικό υπόβαθρο.

- ✘ **Rule Scenario Manager:** Ενδυναμώνει το σύστημα ILOG JRules με ισχυρές δυνατότητες ελέγχου των κανόνων και προσομοίωσης των επιχειρησιακών εφαρμογών. Παρέχει ένα ολοκληρωμένο περιβάλλον για την επαλήθευση της ορθότητας των κανόνων και τη διασφάλιση ότι το αποτέλεσμα της εκτέλεσης του συστήματος είναι αυτό που αναμένεται από τους ειδικούς της επιχειρησιακής λογικής. Επιπλέον, οι χρήστες μπορούν να χρησιμοποιήσουν τον Rule Scenario Manager για να προσομοιώσουν αλλαγές στις πολιτικές της επιχείρησης, και να συγκρίνουν είτε τα αποτελέσματα δύο διαφορετικών συνόλων κανόνων πάνω στα ίδια γεγονότα, είτε τα αποτελέσματα εκτέλεσης του ίδιου συνόλου κανόνων πάνω σε διαφορετικά σύνολα γεγονότων.

Όσον αφορά την ανάπτυξη και την εκτέλεση των κανόνων ακολουθείται ένας πλήρης κύκλος διαχείρισης των επιχειρηματικών κανόνων που σχετίζεται με την κατάλληλη ακολουθιακή εκμετάλλευση της υπάρχουσας υποδομής, ώστε να παραχθούν και να εκτελεστούν τα κατάλληλα σύνολα κανόνων (rulesets). Ένα ruleset είναι ένας αυτόνομος και εκτελέσιμος container που αντιστοιχεί και ανταποκρίνεται σε μια απόφαση. Οι κανόνες είναι ατομικές εκφράσεις της εταιρικής πολιτικής και η αυτοματοποίηση των πολιτικών αυτών και η εφαρμογή τους οδηγεί στην παραγωγή αποφάσεων. Οι κανόνες που αναφέρονται σε μια καθορισμένη απόφαση οργανώνονται προς εκτέλεση και αποθηκεύονται σε ένα ruleset.



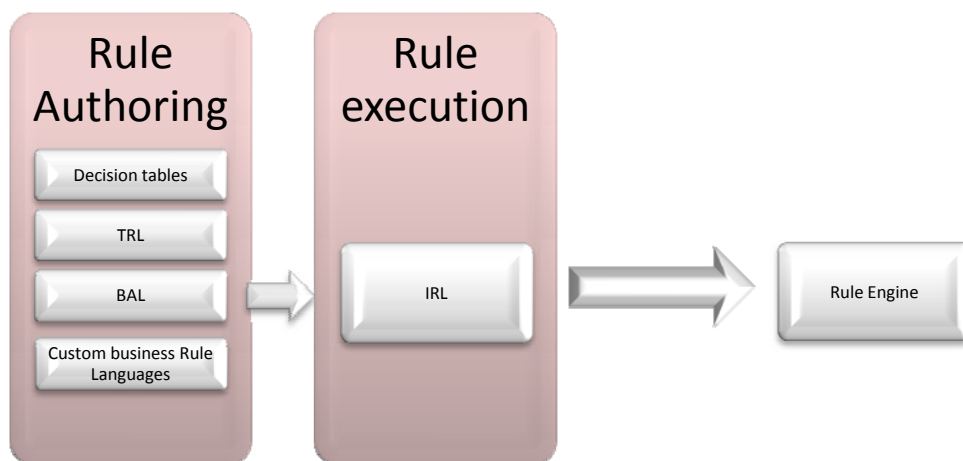
Σχήμα 8-10: Αρχιτεκτονική Προσέγγιση Σχεδιασμού μιας Εφαρμογής Επιχειρηματικών Κανόνων στο ILOG JRules

Το σύστημα διαχείρισης κανόνων της ILOG καθιστά δυνατό τον καθορισμό και τη συγγραφή της επιχειρηματικής λογικής σε φυσική επιχειρηματική γλώσσα με βάση επιχειρηματική ορολογία και όχι σε προγραμματιστικό κώδικα. Στο πλαίσιο της συγκεκριμένης προσέγγισης, βασική δομική μονάδα επεξεργασίας για την ανάπτυξη κανόνων είναι το μοντέλο επιχειρηματικού αντικειμένου (Business Object Model) που περιγράφει τα δεδομένα πάνω στα οποία βασίζεται η κάθε απόφαση προδιαγράφοντας τα σε λεξιλόγιο φυσικής γλώσσας.

Προκειμένου η εφαρμογή να δημιουργήσει τα κατάλληλα επιχειρηματικά αντικείμενα πρέπει να στηριχτεί σε δεδομένα και προγραμματιστικές κλάσεις που να υλοποιούν και να περιγράφουν τα αντικείμενα που επεξεργαζόμαστε σε γλώσσα Java. Επομένως, προκύπτει η ανάγκη για συσχέτιση των BOM αντικειμένων με τα Java αντικείμενα ή τα XML σχήματα που περιγράφουν προγραμματιστικά τις παραμέτρους που θα χρησιμοποιηθούν σε φυσική γλώσσα για την δημιουργία των επιχειρηματικών κανόνων. Τέτοια προγραμματιστικά αντικείμενα ονομάζονται μοντέλα εκτέλεσης αντικειμένου (Execution Object Model). Η συσχέτιση αυτή (mapping) αφορά δεδομένα που η επιχειρηματική εφαρμογή διαχειρίζεται με τη μορφή φυσικών Java Objects ή XML και ο χρήστης διαχειρίζεται με τη μορφή BOM σε φυσική κατανοητή γλώσσα. Παράλληλα, πρέπει να οριστούν και οι παράμετροι εισόδου και εξόδου που περιγράφουν τα δεδομένα που δέχεται το ruleset ως είσοδο και εκείνα που παρέχει ως έξοδο.

Κατά την δημιουργία και επεξεργασία των επιχειρηματικών κανόνων υπάρχει μια σειρά από γλώσσες συγγραφής των κανόνων (rule languages) που έχουν πρόσβαση στα αντικείμενα που έχουμε δημιουργήσει και υποστηρίζονται από τη συγκεκριμένη μηχανή παραγωγής κανόνων.

- ✘ Η **Γλώσσα Επιχειρηματικής Δράσης** (Business Action Language or BAL) είναι γλώσσα επιχειρηματικών κανόνων γενικού σκοπού και εμφανίζει σύνταξη παρόμοια και αρκετά κοντά στη φυσική γλώσσα. Η γλώσσα BAL έχει σχεδιαστεί για να καλύπτει το σύνολο των αναγκών κατά τη συγγραφή επιχειρηματικών κανόνων.
- ✘ Οι **Πίνακες Αποφάσεων** (Decision Tables) είναι κανόνες που συντίθενται από στήλες και γραμμές και χρησιμοποιούνται για να λάβουν υπόψη σε μορφή πίνακα κάθε δυνατή κατάσταση που μπορεί να αντιμετωπιστεί κατά τη αποτίμηση των συνθηκών μιας απόφασης.
- ✘ Τα **Δέντρα Αποφάσεων** (Decision Trees) είναι το γραφικό ανάλογο των Πινάκων Αποφάσεων.
- ✘ Η **Γλώσσα Κανόνων της ILOG** (ILOG Rule Language or IRL) είναι η γλώσσα που μπορεί άμεσα να εκτελεστεί από τη μηχανή παραγωγής κανόνων της ILOG. Η γλώσσα IRL εμφανίζει συντακτικούς κανόνες ανάλογους με την Java και χρησιμοποιείται περισσότερο από τους προγραμματιστές. Οι γλώσσες επιχειρηματικών κανόνων (BAL ή TRL) μπορεί να χρησιμοποιηθούν για τη συγγραφή κανόνων. Εν συνεχεία όμως, για την εκτέλεση των κανόνων, μεταφράζονται σε IRL.
- ✘ Η **Γλώσσα Τεχνικών Κανόνων** (Technical Rule Language) στηρίζεται συντακτικά στη γλώσσα IRL και έχει παραγκωνιστεί στις νεότερες εκδόσεις της ILOG με αντικατάσταση και ενσωμάτωση της στην IRL. Οι περισσότεροι χρήστες δεν γνωρίζουν καν την ύπαρξη της.



Σχήμα 8-11: Γλώσσες Συγγραφής, Επεξεργασίας και Εκτέλεσης Κανόνων στο ILOG JRules

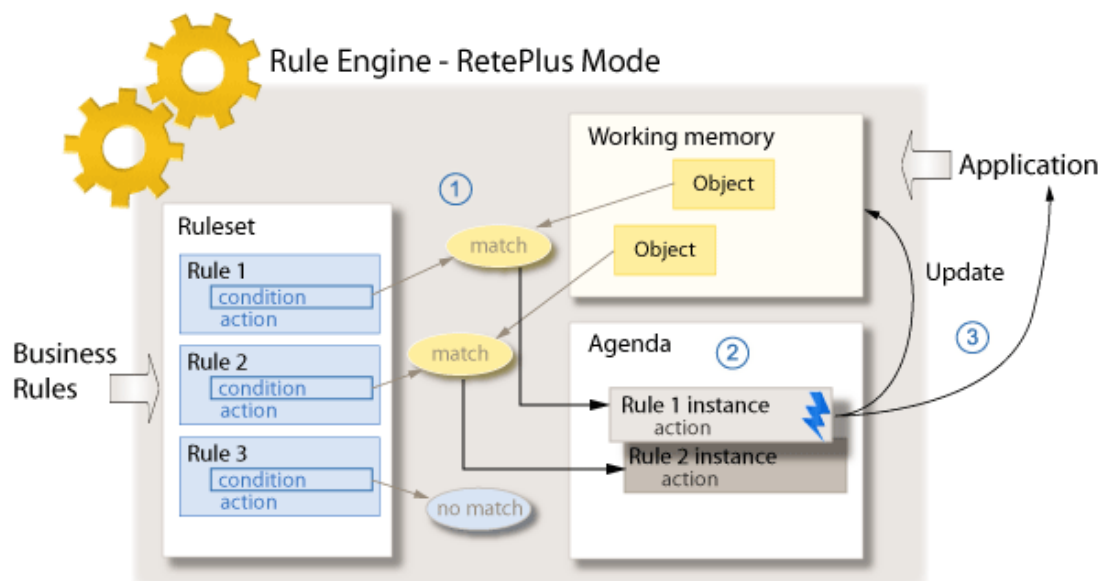
Έχοντας καθορίσει τα δεδομένα στη μορφή επιχειρηματικών αντικειμένων και έχοντας δημιουργήσει τους επιχειρηματικούς κανόνες ή τους πίνακες αποφάσεων με την κατάλληλη ροή εκτέλεσης τους, η φάση της εκτέλεσης που ακολουθεί μπορεί να πραγματοποιηθεί με διάφορους τρόπους :

- ✘ Υπάρχει η δυνατότητα εκτέλεσης της εφαρμογής στο περιβάλλον ελέγχου, ώστε να ελεγχθεί η ορθότητα των κανόνων και να προσομοιωθεί η λειτουργία τους με τη χρήση του Rule Scenario Manager.
- ✘ Υπάρχει η δυνατότητα ενσωμάτωσης κλήσης του επιχειρηματικού μοντέλου μέσα στην επιχειρηματική εφαρμογή και άμεσης αλληλεπίδρασης του με τη μηχανή παραγωγής κανόνων.
- ✘ Υπάρχει η δυνατότητα εξωτερίκευσης της επιχειρηματικής εφαρμογής με τη μορφή Διαδικτυακών Υπηρεσιών και κλήσης της από μια επιχειρηματική διαδικασία που ακολουθεί ανάλογα πρωτόκολλα μεταφοράς και επικοινωνίας και SOA αρχιτεκτονικής.
- ✘ Υπάρχει η δυνατότητα εκτέλεσης στο γραφικό περιβάλλον του Rule Execution Server. Σε αυτήν την περίπτωση δίνεται η δυνατότητα κλήσης του κώδικα της εφαρμογής με χρήση είτε ενός απλού Java Interface είτε ενός τοπικού EJB. Επίσης, μπορεί να κληθεί από μια απομακρυσμένη Java Virtual Machine με τη χρήση απομακρυσμένων (remote) Java Beans ή με ασύγχρονο τρόπο με τη χρήση message driven beans.

Τέλος, αξίζει να αναφερθεί τα είδη των αλγορίθμων που χρησιμοποιούνται στη μηχανή της ILOG για τον τρόπο και τη σειρά εκτέλεσης των κανόνων.

✘ Ο αλγόριθμος RetePlus

Αποτελεί την επέκταση του αλγόριθμου Rete στην μηχανή της ILOG και αποτελεί την προκαθορισμένη μέθοδο εκτέλεσης.



Σχήμα 8-12: Αρχιτεκτονική Προσέγγιση Εκτέλεσης Επιχειρηματικών Κανόνων με Βάση τον Αλγόριθμο RetePlus

Η μέθοδος εκτέλεσης με βάση τον αλγόριθμο RetePlus παρέχει στη μηχανή παραγωγής κανόνων τα απαραίτητα μέσα για την ελαχιστοποίηση του αριθμού των κανόνων και των συνθηκών που πρέπει να αποτιμηθούν, για τον υπολογισμό των κανόνων που τελικά εκτελούνται και για την αναγνώριση της σειράς εκτέλεσης τους. Στα πλαίσια του αλγόριθμου RetePlus, η μηχανή κανόνων χρησιμοποιεί μια

λειτουργική μνήμη (*working memory*) και μια διάταξη - λίστα εργασιών (*agenda*) για να περιέχει και να διαχειρίζεται τα αντικείμενα της εφαρμογής. Η λειτουργική μνήμη περιέχει αναφορές και δείκτες στα αντικείμενα της εφαρμογής και η διάταξη κατατάσσει και διατάσει σε λίστα τα στιγμιότυπα των κανόνων που πρόκειται να πυροδοτηθούν.

Η μηχανή παραγωγής αντιπαρατάσσει και συνταιριάζει τις συνθήκες των κανόνων με τα αντικείμενα της λειτουργικής μνήμης. Κατά τη διαδικασία του ταιριάσματος προτύπων, ο αλγόριθμος RetePlus δημιουργεί ένα σημασιολογικό δίκτυο με βάση τις σημασιολογικές σχέσεις ανάμεσα στους ελέγχους των συνθηκών των κανόνων. Για κάθε ταιρίασμα δημιουργείται ένα στιγμιότυπο των κανόνων και τοποθετείται στη λίστα. Έπειτα, η λίστα με βάση συγκεκριμένες αρχές διάταξης, επιλέγει τη σειρά εκτέλεσης των κανόνων. Με την πυροδότηση του στιγμιότυπου του κανόνα, εκτελείται η ενέργεια που περιέχει και υπάρχει η περίπτωση μεταβολών στη λειτουργική μνήμη με τους παρακάτω τρόπους :

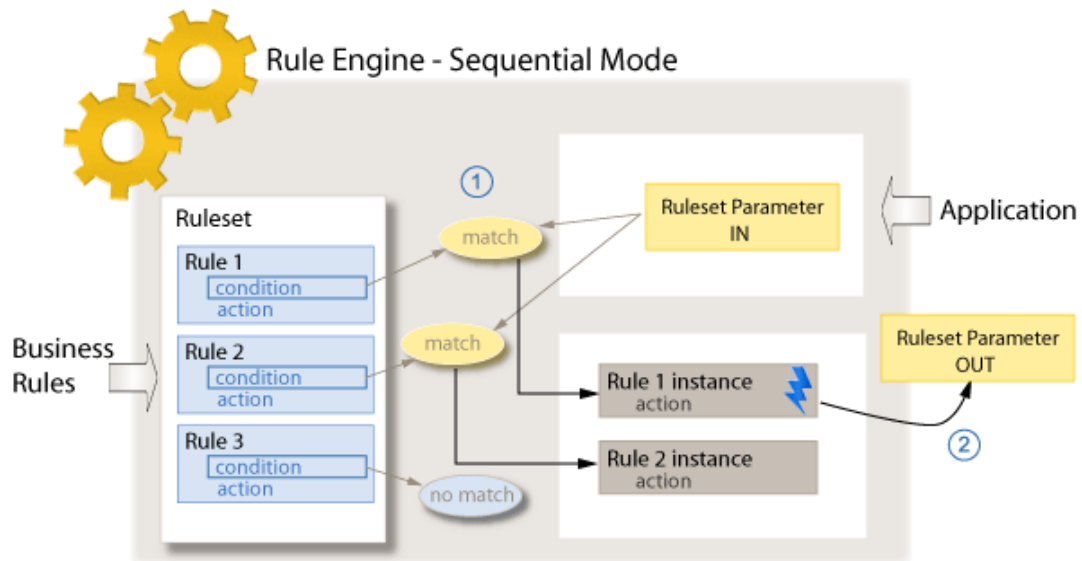
- ✘ Προσθήκη ενός αντικειμένου στη μνήμη
- ✘ Απομάκρυνση ενός αντικειμένου από τη μνήμη
- ✘ Μεταβολή των ιδιοτήτων ενός αντικειμένου.

Η διαδικασία συνεχίζει την πορεία της με κυκλικό και επαναληπτικό τρόπο μέχρι να αδειάσει η λίστα από τα στιγμιότυπα των κανόνων. Στον αλγόριθμο RetePlus, κάθε φορά που μεταβάλλεται η μνήμη, η μηχανή επαναλαμβάνει τη διαδικασία ταιριάσματος, με αποτέλεσμα να αναδιατάσσεται η λίστα και να μεταβάλλονται οι κανόνες που πρέπει να εκτελεστούν και η σειρά τους. Με αυτόν τον τρόπο ο αλγόριθμος RetePlus εμφανίζει υψηλή απόδοση υπολογισμού και συσχέτισης των δεδομένων.

✘ Ο Ακολουθιακός αλγόριθμος

Με τον ακολουθιακό αλγόριθμο, η μηχανή παραγωγής κανόνων της ILOG οδηγείται στην ακολουθιακή εκτέλεση των επιτρεπτών κανόνων για μια δεδομένη εφαρμογή. Κατά την ακολουθιακή εκτέλεση, η μηχανή κανόνων εκτελεί τη διαδικασία ταιριάσματος προτύπων στις παραμέτρους εισόδου του συνόλου κανόνων. Για κάθε ταιρίασμα, δημιουργείται ένα στιγμιότυπο του κανόνα και εκτελείται άμεσα.

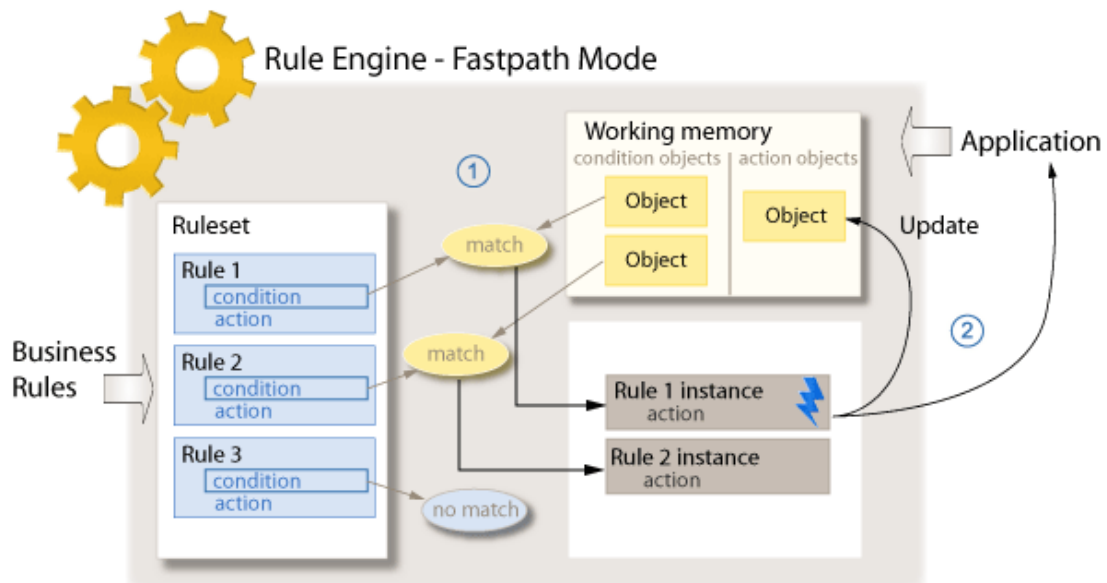
Κατά την πυροδότηση ενός στιγμιότυπου, υπάρχει περίπτωση να τεθεί τιμή σε μια ιδιότητα ή μια παράμετρο εξόδου. Οι κανόνες που εκτελούνται μετά τις παραπάνω μεταβολές, λαμβάνουν υπόψη τις μεταβολές, αλλά εκείνοι που έχουν ήδη εκτελεστεί δεν αποτιμώνται εξ αρχής. Εξαιτίας της συστηματοποιήμενης φύσης του, η ακολουθιακή μέθοδος εκτέλεσης εφαρμόζεται καλύτερα σε εφαρμογές τύπου συμμόρφωσης και επικύρωσης και κατά την αποτίμηση ροών κανόνων (*ruleflows*).



Σχήμα 8-13: Αρχιτεκτονική Προσέγγιση Εκτέλεσης Επιχειρηματικών Κανόνων με Βάση τον Ακολουθιακό Αλγόριθμο

✘ **Ο αλγόριθμος FastPath**

Η μέθοδος εκτέλεσης FastPath παρέχει αυξανόμενα ακολουθιακή μετάφραση και εκτέλεση ενός προγράμματος κανόνων.



Σχήμα 8-14: Αρχιτεκτονική Προσέγγιση Εκτέλεσης Επιχειρηματικών Κανόνων με Βάση τον Αλγόριθμο FastPath

Στον αλγόριθμο FastPath η μηχανή παραγωγής κανόνων μπορεί να χρησιμοποιήσει τη λειτουργική μνήμη ως μέσο αναφοράς των αντικειμένων της εφαρμογής ή των παραμέτρων εισόδου – εξόδου. Ο αλγόριθμος FastPath αναλαμβάνει την εκτέλεση της διαδικασίας ταιριάσματος προτύπων με παρόμοιο τρόπο με τον αλγόριθμο

RetePlus, σχηματίζοντας ένα δέντρο με βάση τις σημασιολογικές σχέσεις ανάμεσα στους ελέγχους των συνθηκών των κανόνων.

Για κάθε ταίριασμα, ένα στιγμιότυπο κανόνων δημιουργείται και πυροδοτείται άμεσα. Καθώς πυροδοτείται ένα στιγμιότυπο, υπάρχει περίπτωση να προκαλέσει μεταβολές στα αντικείμενα της λειτουργικής μνήμης, αλλά οι μεταβολές αυτές δε λαμβάνονται υπόψη και η μηχανή κανόνων δεν επαναλαμβάνει τη διαδικασία ταίριασματος προτύπων. Ο αλγόριθμος FastPath συνδυάζει στοιχεία του αλγορίθμου RetePlus και του ακολουθιακού αλγορίθμου για την εκτέλεση των κανόνων, με αποτέλεσμα να είναι ιδιαίτερα αποτελεσματικός σε εφαρμογές τύπου συσχέτισης και επικύρωσης. Όπως ο ακολουθιακός αλγόριθμος, η μέθοδος εκτέλεσης FastPath είναι stateless υπό την έννοια ότι αναλώνεται στο ταίριασμα προτύπων σε σχέση με ένα μεγάλο αριθμό κανόνων που εκτελούν ατομικά απλές διακρίσεις και διαχωρισμούς ή σχετικούς ελέγχους ένωσης (join tests). Επομένως, δεν υφίσταται η ανάγκη για ύπαρξη λίστας που να υποστηρίζει τη επιλογή της σειράς εκτέλεσης των κανόνων, με αποτέλεσμα να βελτιστοποιείται ο ακολουθιακός αλγόριθμος σε εφαρμογές επικύρωσης και συμμόρφωσης.

4.3.4.2 Εγκατάσταση

Η έκδοση του ILOG JRules που χρησιμοποιήσαμε στην εφαρμογή μας είναι η τελευταία που κυκλοφόρησε και είναι η ILOG JRules 6.7 version. Παρά το γεγονός ότι παρέχεται με χρηματικό αντίτιμο προς χρήση, κατά την περίοδο εργασίας μας χρησιμοποιήσαμε την εξάμηνη δοκιμαστική έκδοση (evaluation version) που παρέχεται μετά από εγγραφή στο site της εταιρείας. Η εγκατάσταση είναι ιδιαίτερα απλή με γραφικό περιβάλλον που καθοδηγεί το χρήστη στα κατάλληλα βήματα που πρέπει να ακολουθήσει. Κατά την εγκατάσταση του λογισμικού, περιλαμβάνονται τα παρακάτω δομικά στοιχεία :

- ✱ **Rule Studio**
- ✱ **Rule Team Server**
- ✱ **Rule Execution Server**
- ✱ **Rule Scenario Manager**
- ✱ **Rule Engine**

Για τη χρησιμοποίηση του Rule Studio αναγκαστήκαμε να χρησιμοποιήσουμε το περιβάλλον ανάπτυξης Eclipse με τα κατάλληλα απαραίτητα επιπρόσθετα χαρακτηριστικά που απαιτούνται για την ενσωμάτωση του Rule Studio. Συγκεκριμένα πρόκειται για την έκδοση Eclipse SDK 3.3 ή μεταγενέστερη με τα επιπλέον plugins EMF 2.3 (Eclipse Modeling Framework), GEF 3.3 (Graphical Editing Framework) και BIRT 2.2.0 (Business Intelligence and Reporting Tools). Η διαδικασία εγκατάστασης του Rule Studio στο περιβάλλον ανάπτυξης Eclipse που ακολουθήθηκε είναι η ακόλουθη :

4. Εκκίνηση του περιβάλλοντος Eclipse με τα απαραίτητα επιπρόσθετα χαρακτηριστικά για την υποστήριξη του Rule Studio (Eclipse SDK 3.3 + EMF 2.3 + GEF 3.3 + BIRT 2.2.0).
5. Στο **Help** menu, επιλέγουμε **Software Updates > Manage Configuration**.

6. Στο **Product Configuration** κουτί διαλόγου, επιλέγουμε **Add an Extension Location** από το δεξί πλαίσιο διαλόγου.
4. Στο **Browse for Folder** πλαίσιο διαλόγου, κατευθυνόμαστε στον κατάλογο JRules <InstallDir>/studio, και στη συνέχεια επιλέγουμε **OK**.
5. Επιλέγουμε το κουμπί **OK** για να επανεκκινήσουμε το περιβάλλον Eclipse προκειμένου να λάβουμε υπόψη και να ενσωματώσουμε τις μεταβολές που πραγματοποιήσαμε.
6. Μετά την επανεκκίνηση του Eclipse, επιλέγουμε **Help > About Eclipse Platform** και επαληθεύουμε ότι το Rule Studio έχει εγκατασταθεί σωστά.

5 Περιγραφή Περιπτώσεων

Η σύγχρονη ανταγωνιστική μορφή της παγκόσμιας αγοράς με τις συνεχείς μεταβολές και ο πολυδιάστατος και δύσκολος χαρακτήρας των επιχειρηματικών φαινομένων που υπαγορεύουν συνεχή επαγρύπνηση και ευελιξία, καθιστούν τον σχεδιασμό των επιχειρηματικών εφαρμογών και τη διαχείριση των υπολογιστικών πόρων στρατηγικής σημασίας παράγοντες για την επιβίωση και την ανάπτυξη. Στα πλαίσια της επιλογής του κατάλληλου τρόπου δόμησης και οργάνωσης των επιχειρησιακών διαδικασιών, η αυτοματοποίηση και η ευελιξία της επιχειρηματικής δραστηριότητας είναι ζητούμενα που απαιτούν εξονυχιστική και ερευνητικά στραμμένη προσέγγιση ως προς την αναζήτηση των κατάλληλων μεθοδολογιών σχεδιασμού και υλοποίησης. Η συνεχής προσαρμογή στις ευμετάβλητες καταστάσεις που υπαγορεύει η σύγχρονη επιχειρηματική πραγματικότητα αποτελεί τον οδηγό για τη διαχείριση των επιχειρηματικών διαδικασιών.

Στη συγκεκριμένη διπλωματική θέτουμε μια συγκεκριμένη επιχειρηματική εφαρμογή με τις προεκτάσεις της στο επίκεντρο της έρευνας μας και εξετάζουμε τις διάφορες μεθόδους υλοποίησης μιας αυτοματοποιημένης επιχειρηματικής διαδικασίας, ώστε να καταλήξουμε στο στόχο μας που είναι η επέκταση και ανάδειξη μιας αρχιτεκτονικής τεσσάρων επιπέδων, όπου η επιχειρηματική πολιτική και στρατηγική οδηγεί τις εξελίξεις ακολουθώντας τους ευμετάβλητους και εξαντλητικούς ρυθμούς επιβίωσης που υπαγορεύει η ανταγωνιστική πορεία της αγοράς. Στο τέλος επιδεικνύεται μια ολοκληρωμένη προσέγγιση προγραμματιστικής υλοποίησης σε πραγματικές υπολογιστικές συνθήκες μιας επιχειρηματικής εφαρμογής που υποτάσσεται στους επιχειρηματικούς κανόνες και πολιτικές, ακολουθεί την πορεία τους και χαρακτηρίζεται από ευελιξία, αυτόματη προσαρμογή στα δεδομένα και επαναχρησιμοποίηση πόρων.

Η επιχειρηματική εφαρμογή που επιλέγεται αναφέρεται στον τραπεζικό τομέα και πρόκειται για τη διαδικασία έγκρισης ή απόρριψης ενός δανείου ή έκδοσης μιας πιστωτικής κάρτας. Για την υλοποίηση εκμεταλλευόμαστε την προσανατολισμένη στις Υπηρεσίες Αρχιτεκτονική με τις διάφορες προεκτάσεις και τεχνολογίες που την έχουν καταστήσει δημοφιλή. Ο σχεδιασμός της αυτοματοποιημένης επιχειρηματικής διαδικασίας πραγματοποιείται σε γλώσσα BPEL. Χρησιμοποιούνται τρεις διαφορετικές προσεγγίσεις έκφρασης της επιχειρηματικής λογικής που στη συνέχεια συγκρίνονται καταλήγοντας σε εκείνη που εμφανίζει τα καλύτερα αποτελέσματα.

- ✱ **Προσέγγιση με βάση τον προγραμματιστικό κώδικα (code-based):** Στη συγκεκριμένη προσέγγιση, η επιχειρηματική λογική εντοπίζεται με τη μορφή προγραμματιστικού κώδικα σε μια αντικειμενοστραφή γλώσσα (Java) που συμμετέχει στην επιχειρηματική BPEL διαδικασία ως διαδικτυακή Υπηρεσία που καλείται στην αφαιρετική (abstract) της υπόσταση (μέσω WSDL γλώσσας) από τη διαδικασία επιτελώντας τη λειτουργία της. Η επιχειρηματική λογική παρουσιάζεται με κανόνες και σημεία αποφάσεων και ελέγχου ροής σε μορφή κώδικα, κάνοντας δυσανάγνωστη και δύσκολα μεταβαλλόμενη τη λειτουργία της.
- ✱ **Προσέγγιση με βάση την οργάνωση – μοντελοποίηση της επιχειρηματικής διαδικασίας (model-driven):** Στην οδηγούμενη από τον τρόπο οργάνωσης και μοντελοποίησης της επιχειρηματικής διαδικασίας προσέγγιση, η επιχειρηματική λογική πραγματοποιείται με βάση τα εργαλεία διαχείρισης επιχειρηματικών διαδικασιών που παρέχει η γλώσσα BPEL και

συνοψίζεται στη χρήση σημείων αποτίμησης και αποφάσεων (decision points) που ελέγχουν και μεταβάλλουν τη ροή εκτέλεσης. Οι διαδικτυακές υπηρεσίες περιορίζονται σε απλές σχετικά λειτουργίες και η σημασιολογία των επιχειρηματικών κανόνων εκφράζεται με τη μορφή σημείων αποφάσεων, όπου αποτιμώνται οι απαραίτητες συνθήκες και επιλέγεται η ροή της εργασίας που θα ακολουθηθεί.

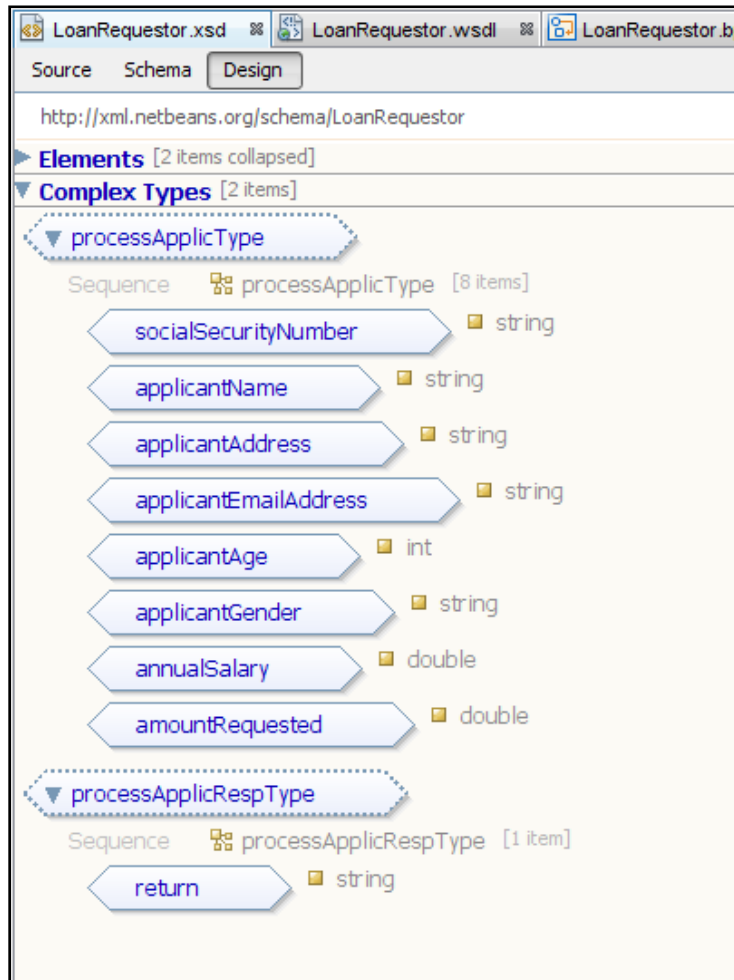
- ✘ **Προσέγγιση προσανατολισμένη στις υπηρεσίες (Service-oriented approach):** Η συγκεκριμένη προσέγγιση υλοποίησης συνδυάζει στο μέγιστο βαθμό τα χαρακτηριστικά της SOA αρχιτεκτονικής και μιας μηχανής παραγωγής κανόνων (business rules engine) ώστε να επιτευχθεί η απαιτούμενη ευελιξία, επαναχρησιμοποίηση και συντηρησιμότητα των επιχειρηματικών διαδικασιών. Η επιχειρηματική λογική και η εταιρική πολιτική εκφράζεται με τη μορφή επιχειρηματικών κανόνων που συντίθενται σε μια μηχανή παραγωγής κανόνων σε φυσική γλώσσα κατανοητή από μη εξειδικευμένα στον προγραμματισμό άτομα. Στη συνέχεια οι επιχειρηματικοί κανόνες εξωτερικεύουν τη λειτουργία και τη συμπεριφορά τους ακολουθώντας τη δομή της WSDL γλώσσας και τέλος καλούνται στα πλαίσια μιας επιχειρηματικής διαδικασίας ως διαδικτυακές υπηρεσίες.

5.1.1 Προσέγγιση με βάση τον προγραμματιστικό κώδικα (code-based) – Υλοποίηση

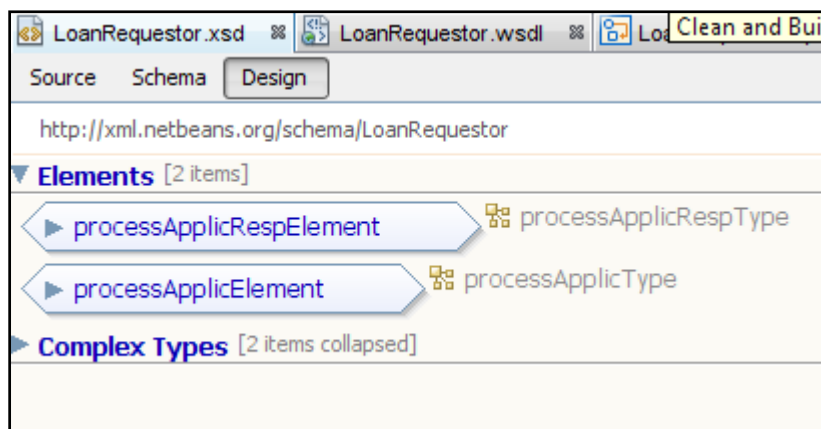
Η επιχειρηματική διαδικασία που χρησιμοποιείται συνίσταται στο σχεδιασμό μιας απλής επιχειρηματικής πορείας έγκρισης ή απόρριψης ενός δανείου. Η επιχειρηματική περίπτωση χρήσης ακολουθεί τα παρακάτω βήματα :

- ✘ Ο χρήστης πραγματοποιεί μια αίτηση για δάνειο συμπληρώνοντας μια φόρμα που περιέχει πληροφορίες σχετικές με τα προσωπικά στοιχεία αναγνώρισης του και με τα χαρακτηριστικά του δανείου.
- ✘ Όταν λαμβάνεται η αίτηση του δανείου, τα προσωπικά στοιχεία του χρήστη επαληθεύονται με βάση μια υπάρχουσα βάση δεδομένων και προκύπτει το αποτέλεσμα με βάση τις πληροφορίες του χρήστη και το ποσό δανείου που αιτείται.
- ✘ Ανακοινώνεται με τυπικότητα η απόφαση στο χρήστη, που σε περίπτωση απόρριψης ενημερώνεται και για την αιτία απόρριψης.

Ως προς την υλοποίηση αρχικά σχηματίζουμε στο περιβάλλον ανάπτυξης NetBeans 6.1 το κατάλληλο XML σχήμα `LoanRequestor.xsd` για την περιγραφή των απαραίτητων στοιχείων που αποτελούν τα δεδομένα εισόδου και εξόδου της επιχειρηματικής διαδικασίας και παρουσιάζουν τη μορφή πολύπλοκης δομής δεδομένων.



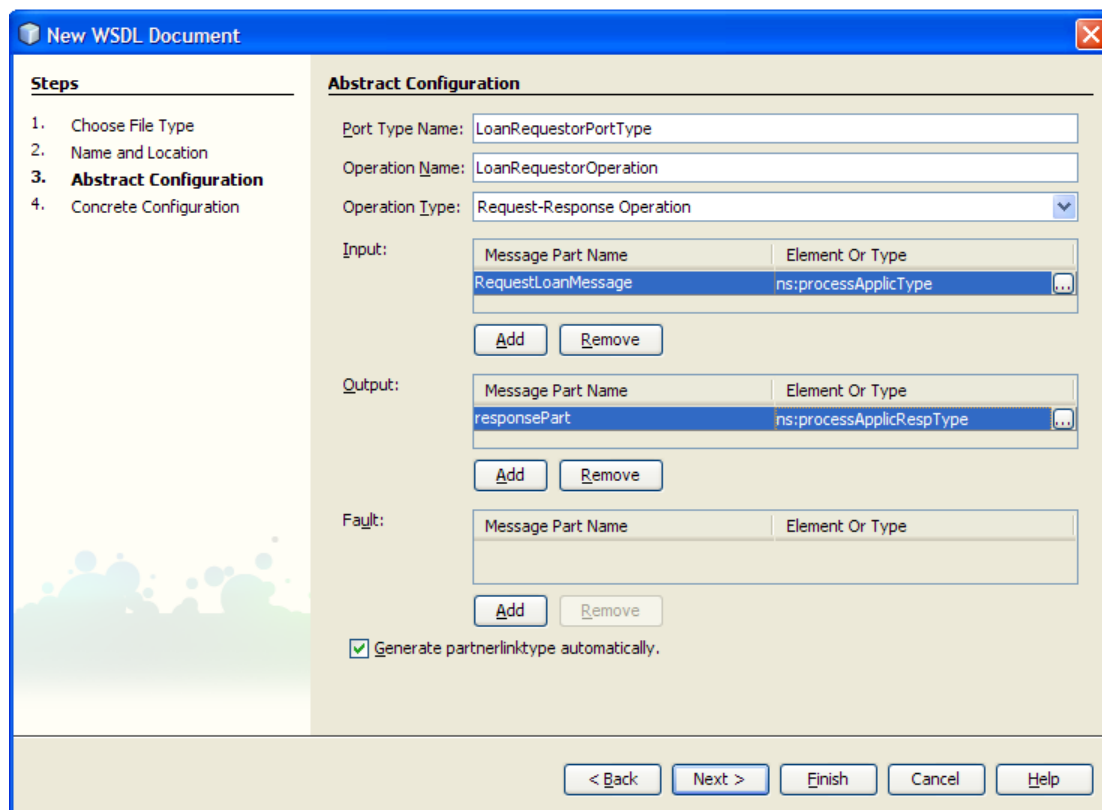
Σχήμα 8-1: Ορισμός Πολύπλοκων Τύπων του XML σχήματος LoanRequestor.xsd



Σχήμα 8-2: Ορισμός Στοιχείων του XML σχήματος LoanRequestor.xsd

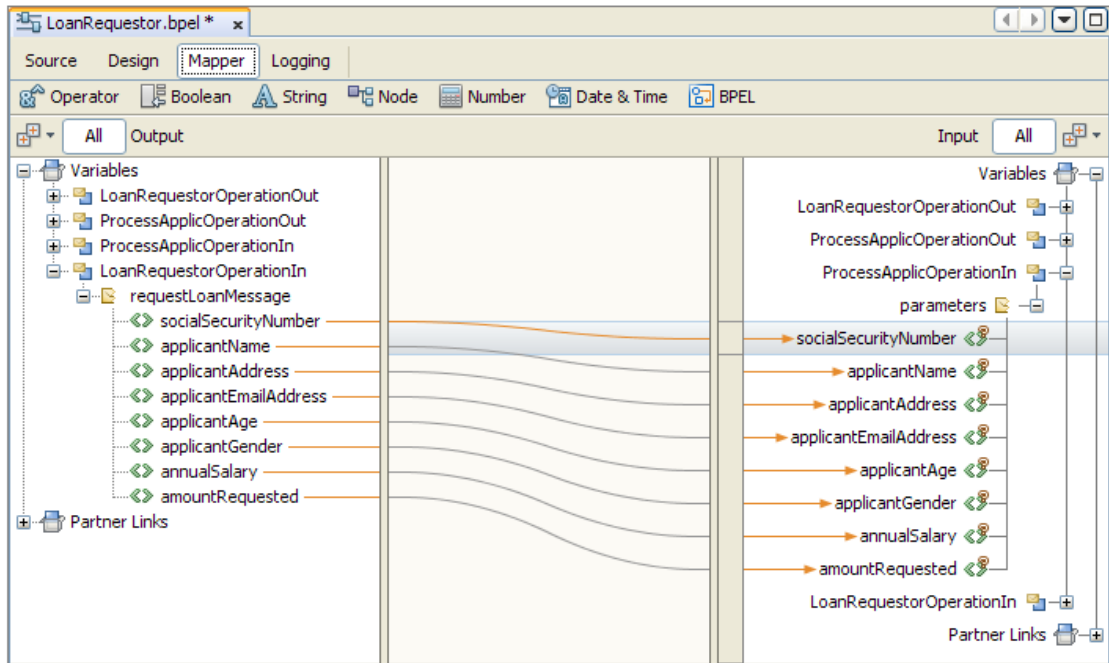
Στη συνέχεια δημιουργούμε μια BPEL εφαρμογή με τη μορφή μονάδας λογισμικού επιχειρηματικής λογικής (BPEL module) που μπορεί να χρησιμοποιηθεί από τους JBI παράγοντες και να εκτελεστεί στον εξυπηρετητή εφαρμογών. Ως τμήμα της BPEL εφαρμογής σχεδιάζουμε την επιχειρηματική διαδικασία και τις διαδικτυακές υπηρεσίες που καλεί. Αρχικά δημιουργούμε ένα WSDL αρχείο που έχει το ρόλο του

πελάτη και δέχεται ως είσοδο τα δεδομένα που απαιτούνται για την αίτηση του δανείου και επιστρέφει ως έξοδο το αποτέλεσμα της αποτίμησης των χαρακτηριστικών της αίτησης. Πρόκειται για το LoanRequestor WSDL αρχείο που επικοινωνεί μέσω XML μηνυμάτων (που ακολουθούν την τυποποίηση του XML σχήματος LoanRequestor.xsd) με τη διαδικασία και επιτελεί τη λειτουργία της αίτησης του δανείου με το μήνυμα requestLoanMessage και τη λειτουργία της αποδοχής της απάντησης με το μήνυμα responsePart .

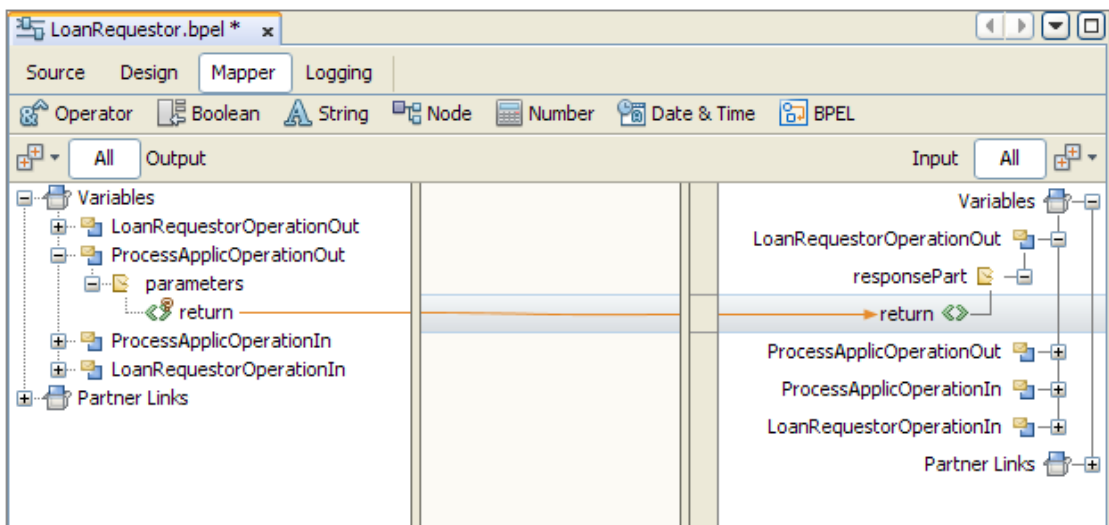


Σχήμα 8-3: Ορισμός Αφαιρετικής Δομής του WSDL αρχείου LoanRequestor

Το LoanRequestor WSDL αρχείο προκαλεί την εκκίνηση της επιχειρηματικής διαδικασίας μέσω μιας εντολής αποδοχής μηνυμάτων ReceiveFromCustomer που υπάρχει στη διαδικασία και αναμένει για την αποστολή δεδομένων εισόδου από τον συνεργάτη LoanRequestor (partner link). Εν συνεχεία με την προσθήκη στη διαδικασία της εντολής μορφής ανάθεσης (assign) CopyCustomerInfo μεταφέρουμε τα δεδομένα εισόδου στην διαδικτυακή υπηρεσία προς επεξεργασία, από την οποία λαμβάνουμε τα αποτελέσματα και μέσω μιας νέας εντολής ανάθεσης CopyLoanStatus τα εξάγουμε στον πελάτη (LoanRequestor WSDL αρχείο) προκειμένου να ενημερωθεί για την απάντηση μέσω της εντολής επιστροφής ReplyToCustomer. Η ύπαρξη της εντολής επιστροφής υπαγορεύεται από τη μορφή της επικοινωνίας που έχει επιλεγεί και εμφανίζει σύγχρονο χαρακτήρα.

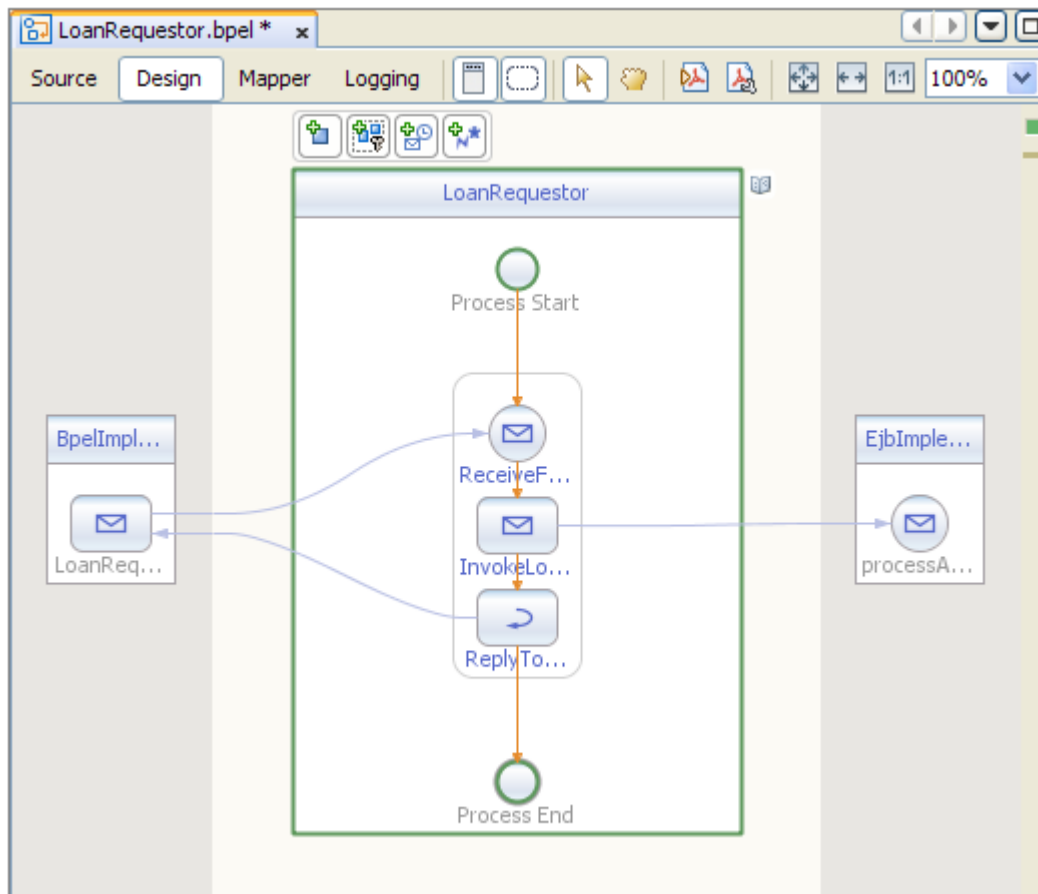


Σχήμα 8-4: Assign Δραστηριότητα CopyCustomerInfo



Σχήμα 8-5: Assign Δραστηριότητα CopyLoanStatus

Η κλήση της διαδικτυακής υπηρεσίας, που αποτελεί το σημαντικότερο βήμα για την επεξεργασία της αίτησης, πραγματοποιείται μέσω της εντολής `InvokeLoanProcessorEJB` που μεταφέρει τα δεδομένα που υπάρχουν στην αίτηση του δανείου στην διαδικτυακή υπηρεσία, όπου επεξεργάζονται τα δεδομένα και λαμβάνεται το αποτέλεσμα της αποτίμησης στο πλαίσιο της επιχειρηματικής λογικής που είναι ενσωματωμένη στην διαδικτυακή υπηρεσία. Προκύπτει επομένως η παρακάτω ολοκληρωμένη επιχειρηματική διαδικασία με βάση τη χρήση του γραφικού περιβάλλοντος BPEL Designer που παρέχει το NetBeans.

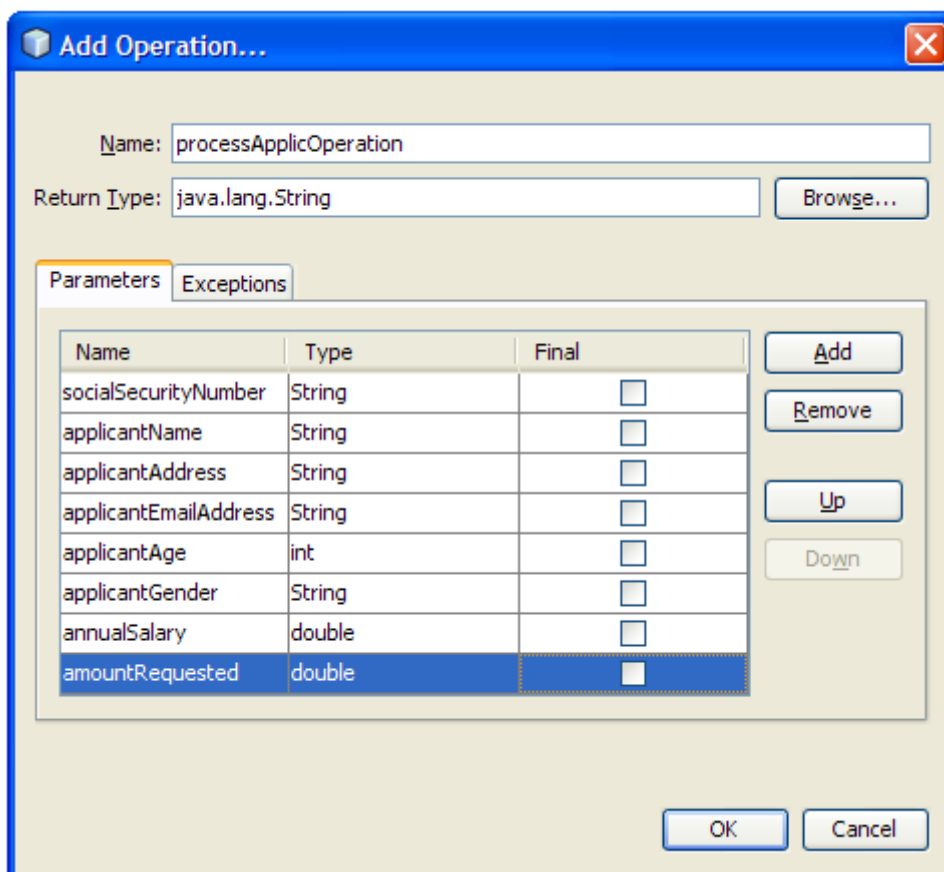


Σχήμα 8-6: Επιχειρηματική Διαδικασία LoanRequestor

Η επιχειρηματική λογική βρίσκεται εξολοκλήρου σε μορφή προγραμματιστικού κώδικα στη διαδικτυακή υπηρεσία και περιέχει τα απαραίτητα προγραμματιστικά εργαλεία ελέγχου της ροής και αποτίμησης των συνθηκών των σημείων αποφάσεων. Για την υλοποίηση της διαδικτυακής υπηρεσίας, χρησιμοποιούνται στην πλατφόρμα επιχειρησιακών εφαρμογών Java EE τα Enterprise Java Beans που είναι μονάδες λογισμικού που φέρουν επιχειρηματική λογική και επεξεργάζονται και εκτελούνται από τη Μηχανή Υπηρεσιών JBI Java EE Service Engine. Η τεχνολογία των EJB αποτελεί την server-side component αρχιτεκτονική της Java EE που επιτρέπει την γρήγορη ανάπτυξη κατακευματισμένων (distributed), συναλλακτικών (transactional), ασφαλών και μεταφέρσιμων εφαρμογών βασισμένες στην τεχνολογία της Java. Σε αντίθεση με άλλες τεχνολογίες, η τεχνολογία των EJB αποκρύπτει τις βασικές διαδικασίες του συστήματος που συνδέονται με τις κατακευματισμένες εφαρμογές, όπως διαχείριση των στιγμιότυπων, διαχείριση της τράπεζας αντικειμένων, διαχείριση πολλαπλών threads και διαχείριση τράπεζας συνδέσεων. Το περιβάλλον στο οποίο λειτουργεί το EJB ονομάζεται EJB Container. Ο EJB Container παρέχει υπηρεσίες όπως διαχείριση συναλλαγών και πόρων, μεταφερσιμότητα, έλεγχο εκδόσεων και υπηρεσίες ασφαλείας στα EJB που περιέχει. Η τεχνολογία των EJB επιτρέπει τη δημιουργία διαδικτυακών υπηρεσιών και διαχείρισης τους, ώστε να αποτελούν τμήμα πολύπλοκων επιχειρηματικών εφαρμογών.

Επομένως, δημιουργούμε της EJB εφαρμογή LoanProcessor, μέσα στην οποία επιλέγουμε τη δυνατότητα δημιουργίας της ομώνυμης διαδικτυακής υπηρεσίας που

είναι υπεύθυνη για την έκφραση των επιχειρηματικών κανόνων. Στη συνέχεια στον κώδικα που συνθέτει την διαδικτυακή υπηρεσία δημιουργούμε τη λειτουργία `processApplicOperation` για την επεξεργασία της αίτησης του δανείου. Ως προγραμματιστική συνάρτηση η λειτουργία της διαδικτυακής υπηρεσίας πρέπει να δέχεται μια σειρά δεδομένων εισόδου, τα οποία επιλέγονται να είναι εκείνα που εισάγει ο πελάτης κατά την υποβολή της αίτησης, δηλαδή τα προσωπικά του στοιχεία και το ποσό του δανείου που επιθυμεί.



Σχήμα 8-7: Ορισμός και Παράμετροι της Διαδικτυακής Υπηρεσίας `LoanProcessor`

Το επόμενο βήμα πρόκειται για την υλοποίηση των επιχειρηματικών κανόνων. Η μορφή τους είναι αποκλειστικά σε περιβάλλον κώδικα Java, γεγονός που αποτελεί εμπόδιο για τη δυνατότητα ενσωμάτωσης συνεχών μεταβολών. Συγχρόνως, η επιχειρηματική λογική σε μορφή κώδικα είναι δυσνόητη και ακατανόητη από τους επιχειρηματικούς αναλυτές που αποφασίζουν για το σχεδιασμό της εταιρικής πολιτικής και την κατάσταση της στρατηγικής, με αποτέλεσμα να δεσμεύονται πόροι για την απασχόληση μηχανικών λογισμικού και να καθιστούν ανελαστική την επιχειρηματική διαδικασία. Παρακάτω φαίνεται ο κώδικας που υλοποιήσαμε και πραγματοποιεί την επιχειρηματική λογική. Αρχικά γίνεται αποτίμηση της ηλικίας του πελάτη, προκειμένου να διαπιστωθεί η δυνατότητα πραγματοποίησης αίτησης δανείου. Στη συνέχεια ελέγχεται η επάρκεια των ετήσιων εσόδων από την απασχόληση του πελάτη, υπολογίζεται το μέγιστο ποσό χορήγησης με βάση την ηλικία και το χρονικό διάστημα αποπληρωμής του δανείου και τέλος λαμβάνεται η απόφαση έγκρισης ή απόρριψης.

```

int MINIMUM_AGE_LIMIT = 18;
int MAXIMUM_AGE_LIMIT = 65;
double MINIMUM_SALARY = 20000;
int AVERAGE_LIFE_EXPECTANCY = 70;
String result = "Loan Application APPROVED.";

// Check age of applicant
// If less than min age limit, rejected
if(applicantAge < MINIMUM_AGE_LIMIT) {
    result = "Loan Application REJECTED - Reason: Under-aged
" +
        applicantAge +
        ". Age needs to be over " +
        MINIMUM_AGE_LIMIT +
        " years to qualify.";
    System.out.println(result);
    return result;
}

// Check age of applicant
// If more than max age limit, rejected
if(applicantAge > MAXIMUM_AGE_LIMIT) {
    result = "Loan Application REJECTED - Reason: Over-aged
" +
        applicantAge +
        ". Age needs to be under " +
        MAXIMUM_AGE_LIMIT +
        " years to qualify.";
    System.out.println(result);
    return result;
}

// Check annual salary
// If less than min salary, rejected
if(annualSalary < MINIMUM_SALARY) {
    result = "Loan Application REJECTED - Reason: Annual
Salary $" +
        annualSalary +
        " too low. Annual Salary needs to be over $" +
        MINIMUM_SALARY +
        " to qualify.";
    System.out.println(result);
    return result;
}

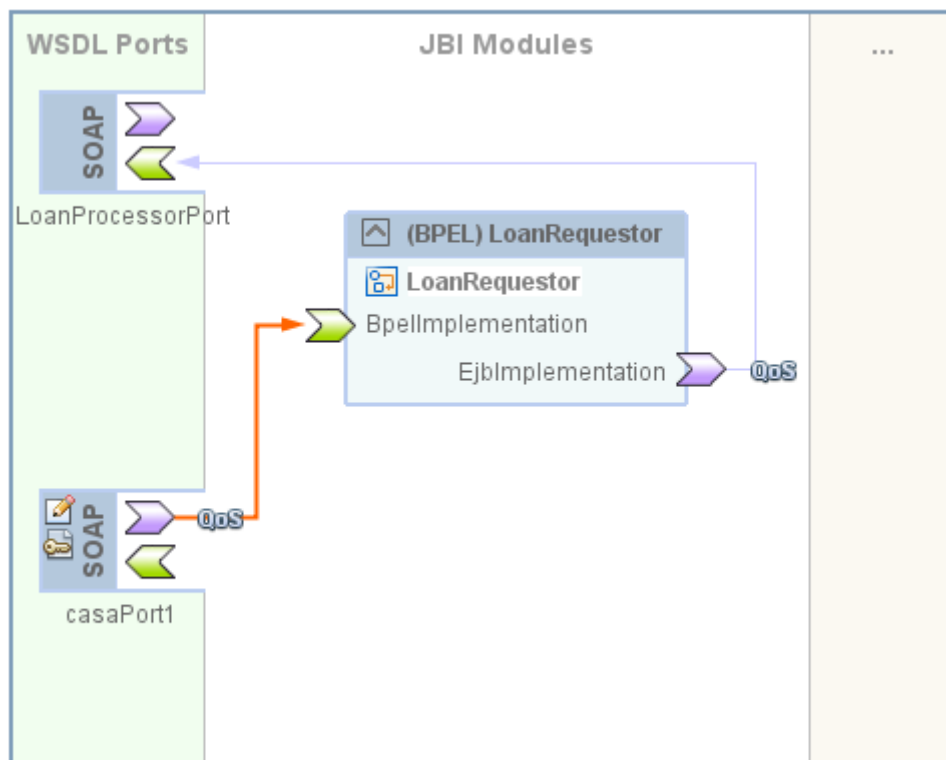
// Calculate the years to pay off loan based on applicantAge
int yearsToRepay = AVERAGE_LIFE_EXPECTANCY - applicantAge;

// Calculate the max amount of loan based on years to pay
off loan
double limit = annualSalary * yearsToRepay * 0.5;

// Check amount requested, if higher than limit, rejected
if(amountRequested > limit) {
    result = "Loan Application REJECTED - Reason: You are
asking for too much $" +
        amountRequested +
        ". Annual Salary $" +
        annualSalary +
        ", Age " +
        applicantAge +
        " years. Your limit is $" +
        limit;
    System.out.println(result);
    return result;
}
System.out.println(result);
return result;

```

Τέλος δημιουργείται η σύνθετη εφαρμογή `LoanRequestorCompositeApp` προκειμένου να ελεγχθεί η ορθή εκτέλεση της διαδικασίας και η ορθή επικοινωνία και ανταλλαγή μηνυμάτων μεταξύ των συναλλασσόμενων πλευρών, όπως πράγματι συμβαίνει με βάση το παρακάτω σχήμα όπου φαίνεται η αλληλεπίδραση και επικοινωνία μέσω του πρωτοκόλλου SOAP.



Σχήμα 8-8: Σύνθετη Εφαρμογή `LoanRequestor`

5.1.2 Προσέγγιση που οδηγείται από τη μοντελοποίηση της επιχειρηματικής διαδικασίας

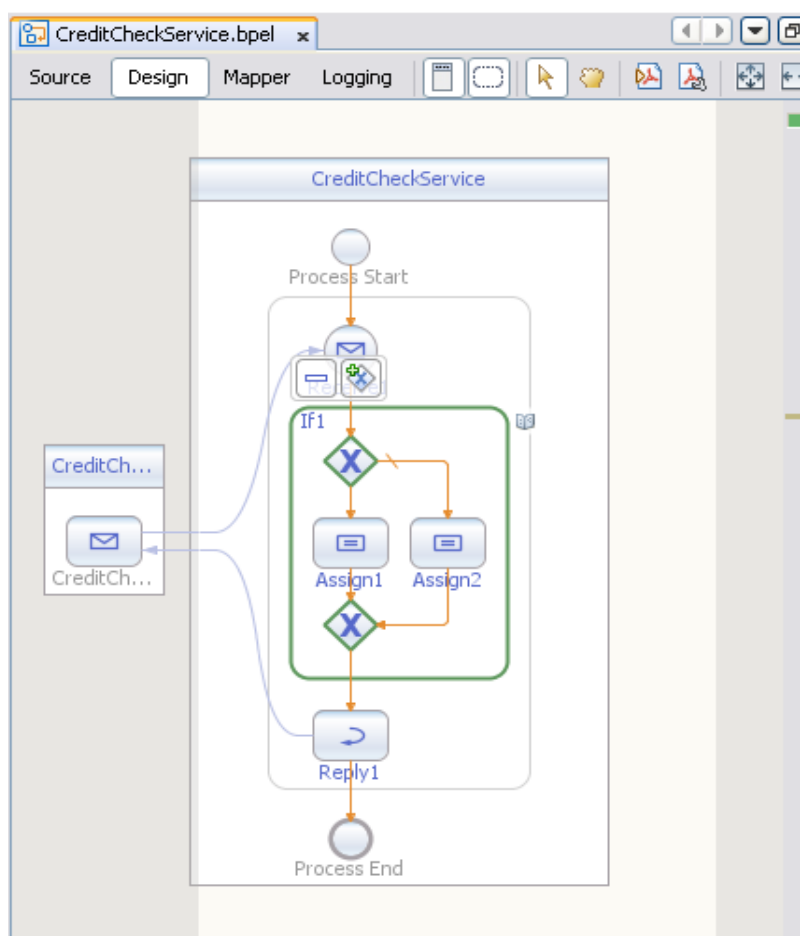
Για τη μελέτη των βασικών χαρακτηριστικών της προσέγγισης που στηρίζει την υλοποίηση της επιχειρηματικής λογικής στην ύπαρξη των κατάλληλων εργαλείων ελέγχου ροής και σχεδιασμού σημείων αποφάσεων σε επίπεδο επιχειρηματικής διαδικασίας, χρησιμοποιείται ως περίπτωση χρήσης η διαδικασία αίτησης και επεξεργασίας ενός δανείου. Για τις απαιτήσεις της συγκεκριμένης εφαρμογής, η επιχειρηματική διαδικασία αποτελεί σύνθεση διαφορετικών διαδικτυακών υπηρεσιών που έχουν υλοποιηθεί με τη σειρά τους ως διακριτές επιχειρηματικές διαδικασίες. Γίνεται με αυτόν τον τρόπο πραγματικότητα η έννοια της ενορχήστρωσης των διαδικτυακών υπηρεσιών προς εκπλήρωση μιας επιχειρηματικής διαδικασίας υψηλότερου επιπέδου.

Η πορεία της επιχειρηματικής διαδικασίας ξεκινά με την αίτηση που υποβάλλει ο χρήστης για τη χορήγηση δανείου. Η αίτηση διέρχεται στη συνέχεια μέσα από το στάδιο της επικύρωσης της πιστωτικής φερεγγυότητας του αιτούντα και τέλος επαληθεύεται η έγκριση ή απόρριψη του δανείου με βάση την αξία της κατοικίας του αιτούντα του δανείου. Το αποτέλεσμα της διαδικασίας γράφεται σε ένα εξωτερικό

αρχείο και αποθηκεύεται. Μέσα από την εφαρμογή δίνεται η δυνατότητα μελέτης και χρήσης τόσο του σύγχρονου όσο και του ασύγχρονου τρόπου επικοινωνίας, καθώς η κλήση της διαδικτυακής υπηρεσίας της εκτίμησης της αξίας της κατοικίας και σύγκρισης της με το ποσό του δανείου δεν απαιτεί άμεση επιστροφή απάντησης στη διαδικασία.

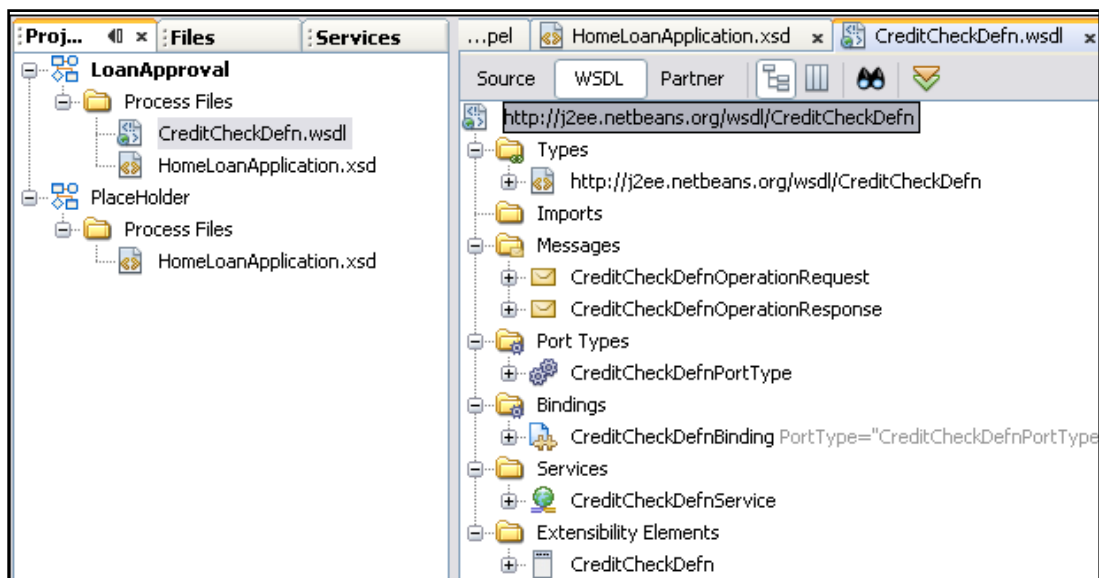
Αρχικά καθορίζεται το XML σχήμα (HomeLoanApplication.xsd) πάνω στο οποίο στηρίζεται η μορφή των μηνυμάτων που ανταλλάσσονται μεταξύ των μελών της διαδικασίας. Περιλαμβάνει στοιχεία και πληροφορίες σχετικές με τον υποψήφιο δανειολήπτη και με τις παραμέτρους του δανείου. Το δεύτερο βήμα είναι η δημιουργία της επιχειρηματικής εφαρμογής και της κύριας υπηρεσίας LoanApprovalService μέσα σε αυτήν που οργανώνει και ελέγχει την ροή της δανειακής διαδικασίας. Ακολουθεί η οριοθέτηση των διαδικτυακών υπηρεσιών που καλεί η διαδικασία και οι οποίες έχουν τη μορφή απλούστερων επιχειρηματικών διαδικασιών.

Η πρώτη υπηρεσία με τη μορφή BPEL διαδικασίας αντιπροσωπεύει την επιχειρηματική λογική του ελέγχου της πιστωτικής φερεγγυότητας του δανειολήπτη. Δημιουργείται αρχικά η BPEL διαδικασία με το όνομα CreditCheckService και παράγεται το WSDL αρχείο CreditCheckDefn.wsdl που καθορίζει τις συνθήκες του συμβολαίου που υπαγορεύει η αίτηση του δανείου και εμφανίζει το ρόλο του πελάτη της επιχειρηματικής διαδικασίας.



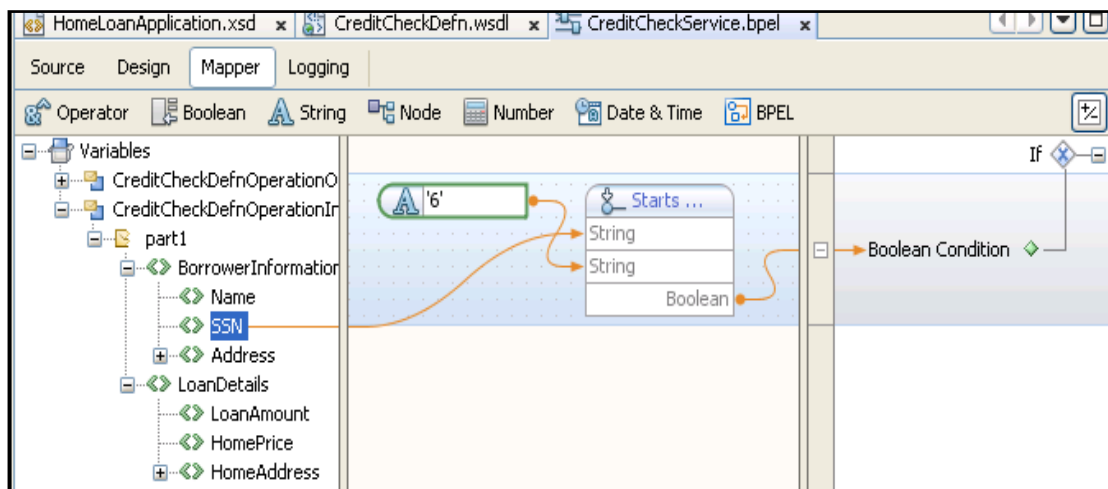
Σχήμα 8-9: Επιχειρηματική Διαδικασία CreditCheckService

Ως είσοδος της διαδικασίας χρησιμοποιείται το σύνολο των στοιχείων που παρέχει το CreditCheckDefn.wsdl αρχείο και τα οποία είναι απαραίτητα, καθώς σχετίζονται με τις παραμέτρους της αίτησης δανείου. Από την άλλη πλευρά ως έξοδο της διαδικασίας στον πελάτη CreditCheckDefn παρέχεται το αποτέλεσμα της αποτίμησης της πιστοληπτικής ικανότητας του πελάτη.



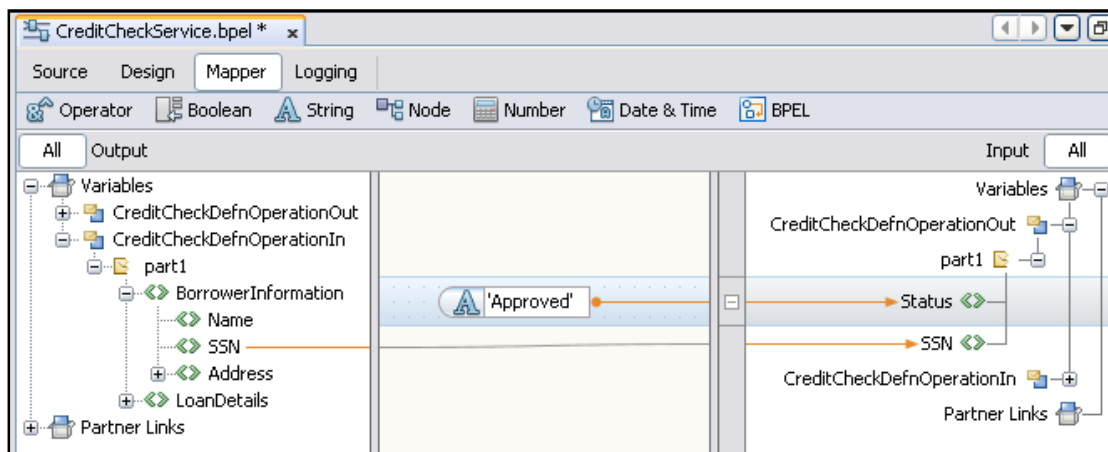
Σχήμα 8-10: WSDL δομή πελάτη CreditCheckDefn

Η επιχειρηματική λογική που χρησιμοποιείται από την επιχειρηματική διαδικασία είναι απλή και στηρίζεται σε μια δομή ελέγχου που αποτελεί το μοναδικό σημείο απόφασης και έχει τη μορφή if....then. Η επιχειρηματική διαδικασία δέχεται μέσω μιας δραστηριότητας Receive τα στοιχεία του χρήστη που αιτείται το δάνειο και στη συνέχεια ελέγχει την πιστωτική του κατάσταση με βάση τον μοναδικό κωδικό που τον χαρακτηρίζει. Στην περίπτωση που ο κωδικός του πελάτη ξεκινά από '6', το αποτέλεσμα της αποτίμησης είναι θετικό και ο χρήστης θεωρείται ότι χαρακτηρίζεται από πιστοληπτική αξιοπιστία. Σε αντίθετη περίπτωση γίνεται κατανοητό ότι η αίτηση για δάνειο απορρίπτεται, καθώς η πιστωτική του ικανότητα είναι αμφίβολη.

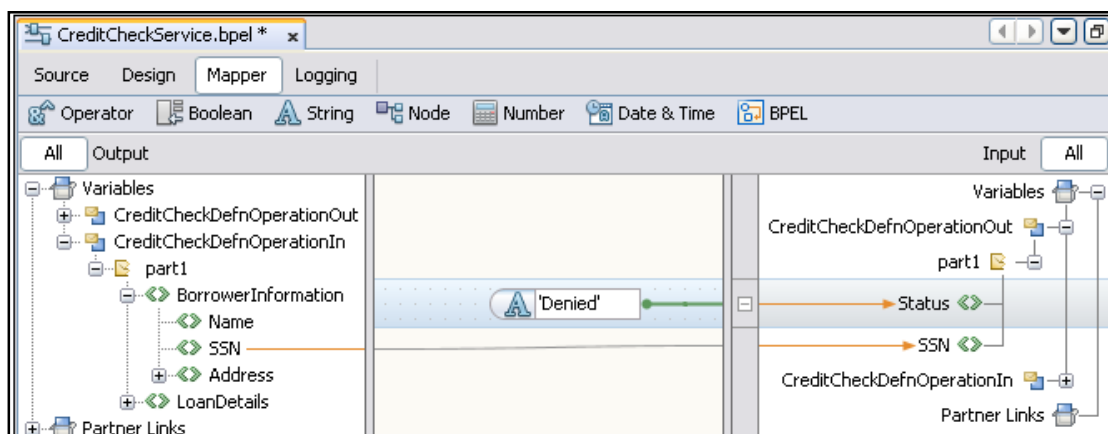


Σχήμα 8-11: Δομή Σημείου Αποφάσεων

Σε κάθε περίπτωση το αποτέλεσμα του ελέγχου οικονομικής αξιοπιστίας του πελάτη καταγράφεται ως απάντηση μέσω μιας δραστηριότητας Assign και τελικά ενημερώνεται ο χρήστης με μια σύγχρονη δραστηριότητα Reply. Το μήνυμα που παρουσιάζεται στο χρήστη είναι η λέξη 'Approved' σε περίπτωση θετικής απάντησης και 'Denied' σε περίπτωση αρνητικής.



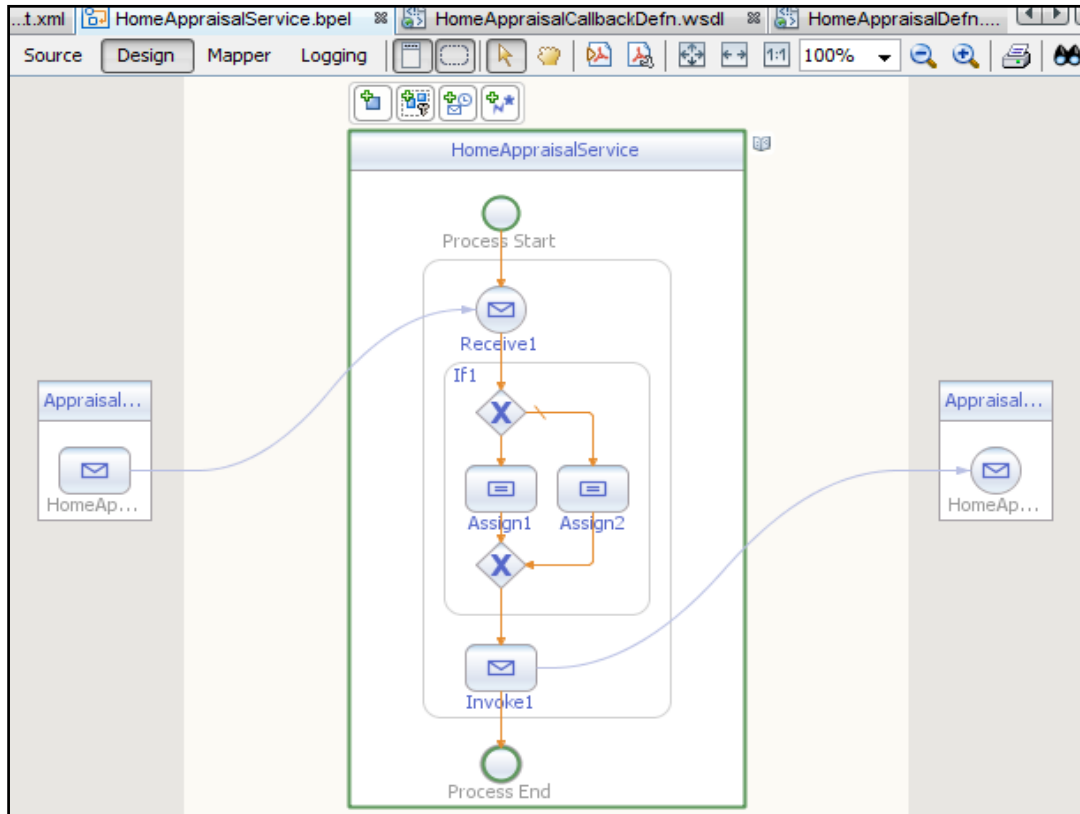
Σχήμα 8-12: Assign Δραστηριότητα σε Περίπτωση Θετικού Αποτελέσματος



Σχήμα 8-13: Assign Δραστηριότητα σε Περίπτωση Αρνητικού Αποτελέσματος

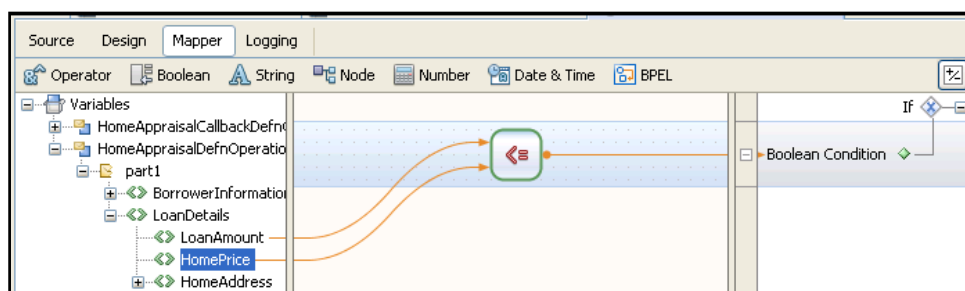
Η δεύτερη υπηρεσία με τη μορφή ασύγχρονης BPEL διαδικασίας αντιπροσωπεύει την επιχειρηματική λογική του ελέγχου της έγκρισης ή απόρριψης του δανείου με βάση την αξία της κατοικίας του πελάτη, η οποία θεωρείται ως ένας είδος εγγύησης. Δημιουργείται αρχικά η BPEL διαδικασία με το όνομα HomeAppraisalService και παράγεται το WSDL αρχείο HomeAppraisalDefn.wsdl που καθορίζει τις συνθήκες της αίτησης του δανείου και εμφανίζει το ρόλο του πελάτη της επιχειρηματικής διαδικασίας. Το συγκεκριμένο αρχείο WSDL παρουσιάζει στην αφαιρετική του υλοποίηση τη μορφή λειτουργίας μιας ροής (One-Way Operation), καθώς παρέχει μέσω μιας δραστηριότητας Receive μόνο τα δεδομένα εισόδου. Η έξοδος πραγματοποιείται μέσω μιας Reply δραστηριότητας και το αποτέλεσμα της αποτίμησης αποστέλλεται ως μήνυμα 'Approved' (σε περίπτωση θετικής απάντησης)

και 'Denied' (σε περίπτωση αρνητικής) στον συνεργάτη της διαδικασίας που έχει δημιουργηθεί με παρόμοιο τρόπο στη μορφή ενός WSDL αρχείου και ονομάζεται HomeAppraisalCallbackDefn.wsdl προκειμένου να γίνει δυνατή η ασύγχρονη μορφή επικοινωνίας. Ο χρήστης της επιχειρηματικής διαδικασίας καλεί επομένως την πρώτη λειτουργία που αποστέλλει τα δεδομένα εισόδου προς επεξεργασία και στη συνέχεια η διαδικασία καλεί τον χρήστη με τη χρήση του δεύτερου αρχείου WSDL γλώσσας υλοποίησης.



Σχήμα 8-14: Επιχειρηματική Διαδικασία HomeAppraisalService

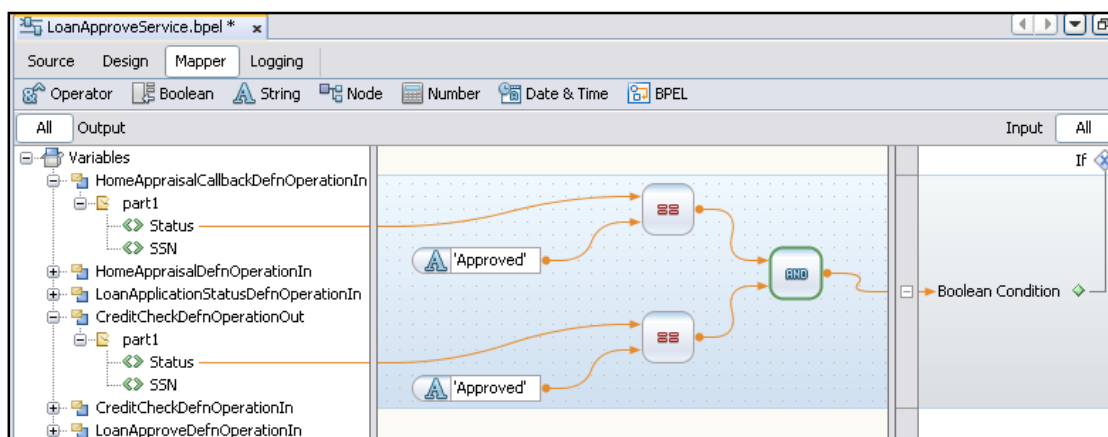
Η επιχειρηματική λογική της επιχειρηματικής διαδικασίας στηρίζεται στην ύπαρξη ενός σημείου απόφασης που ελέγχει τη ροή με βάση μια απλή σύγκριση ανάμεσα στην τιμή της κατοικίας και το ποσό του δανείου που αιτείται ο πελάτης. Σε περίπτωση που το ποσό του δανείου υπερβαίνει την αξία της κατοικίας που παρέχεται ως εγγύηση η αίτηση απορρίπτεται, ενώ σε αντίθετη περίπτωση γίνεται δεκτή.



Σχήμα 8-15: Επιχειρηματική Λογική Σημείου Αποφάσεων

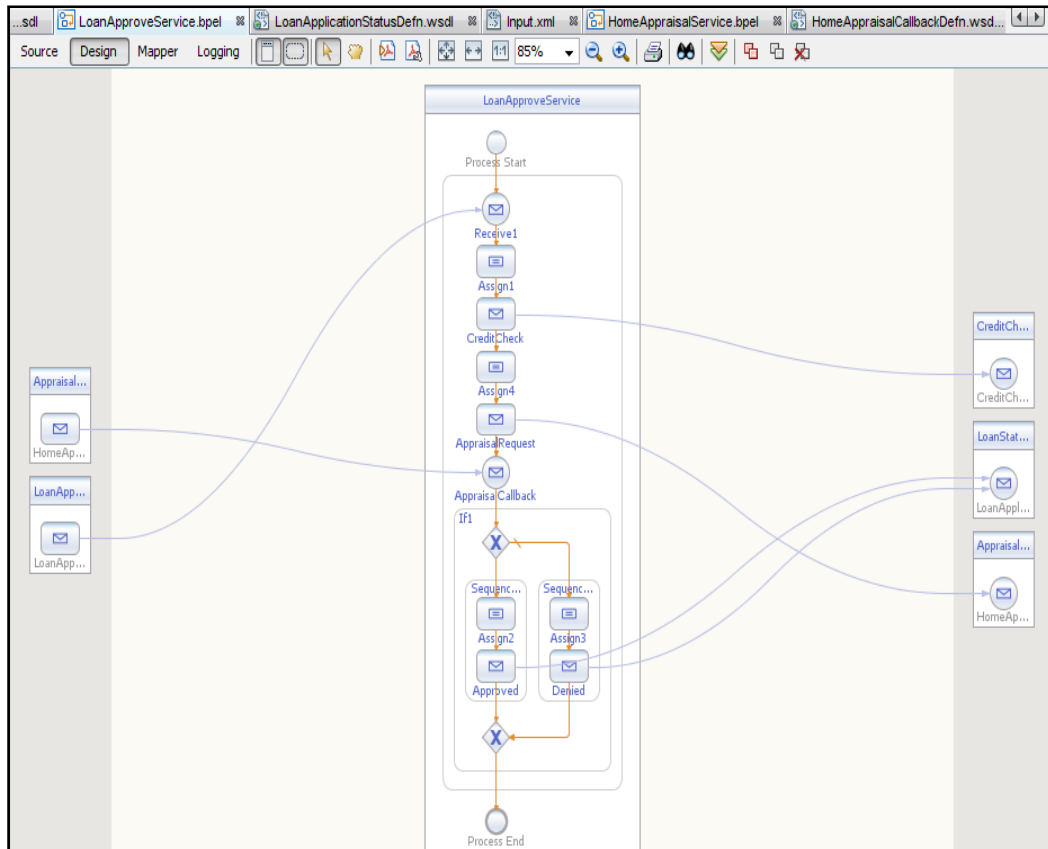
Στο τελευταίο στάδιο ανάπτυξης της επιχειρηματικής εφαρμογής, πραγματοποιείται η ενορχήστρωση των διαδικτυακών υπηρεσιών που δημιουργήθηκαν και αναπτύσσεται η κύρια επιχειρηματική διαδικασία με το όνομα LoanApprovalService. Η συγκεκριμένη επιχειρηματική διαδικασία ακολουθεί την παρακάτω ροή :

1. Ο χρήστης καλεί τη λειτουργία του αρχείου WSDL LoanApproveDefn.wsdl που αποστέλλει τα δεδομένα που σχετίζονται με τον πελάτη και το δάνειο που αιτείται.
2. Η διαδικασία καλεί με τη σειρά της μέσω του WSDL αρχείου CreditCheckDefn.wsdl τη διαδικτυακή υπηρεσία αποτίμησης και ελέγχου της πιστοληπτικής αξιοπιστίας του πελάτη που επιστρέφει ως απάντηση το αποτέλεσμα του ελέγχου.
3. Στη συνέχεια καλείται από τη διαδικασία η διαδικτυακή υπηρεσία εκτίμησης της αξίας της κατοικίας και σύγκρισης της με το ποσό του δανείου. Η κλήση γίνεται με ασύγχρονο τόπο και το αποτέλεσμα επιστρέφει μέσω της δραστηριότητας Callback.
4. Η επιχειρηματική λογική περιλαμβάνει την ύπαρξη ενός σημείου απόφασης που συνεκτιμώντας τα αποτελέσματα των διαδικτυακών υπηρεσιών, καταλήγει στην τελική απάντηση της έγκρισης ή απόρριψης του δανείου.

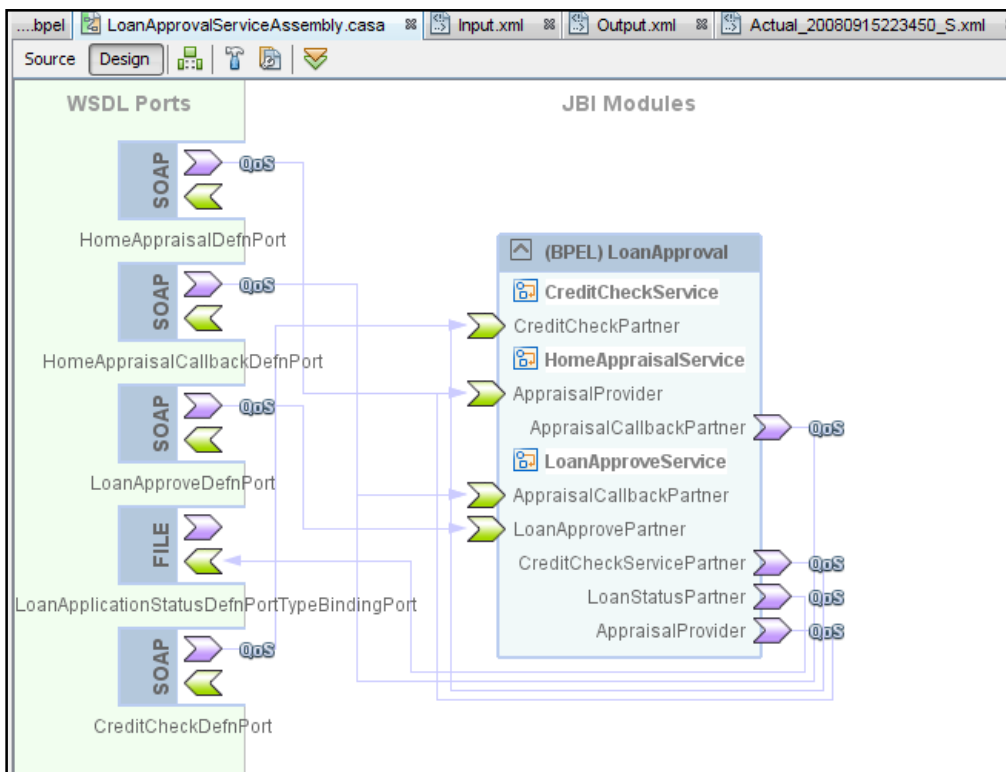


Σχήμα 8-16: Επιχειρηματική Λογική Σημείου Αποφάσεων Κύριας Διαδικασίας

5. Το αποτέλεσμα επιστρέφει με τη μορφή αρχείου, εφόσον έχει επιλεγεί ως πρωτόκολλο μεταφοράς των δεδομένων απάντησης το πρωτόκολλο μεταφοράς αρχείων (File Transport Protocol - FTP) σε ένα συνεργάτη που έχει την αφαιρετική μορφή ενός WSDL αρχείου και ονομάζεται LoanApplicationStatusDefn.wsdl.
6. Τέλος δημιουργείται η σύνθετη εφαρμογή LoadApprovalServiceAssembly προκειμένου να ελεγχθεί η ορθή εκτέλεση της διαδικασίας και η ορθή επικοινωνία και ανταλλαγή μηνυμάτων μεταξύ των συναλλασσόμενων πλευρών, όπως πράγματι συμβαίνει με βάση το παρακάτω σχήμα όπου φαίνεται η αλληλεπίδραση και επικοινωνία μέσω του πρωτοκόλλου SOAP.



Σχήμα 8-17: Κύρια Επιχειρηματική Διαδικασία LoanApprovalService



Σχήμα 8-18: Κύρια Σύνθετη Επιχειρηματική Εφαρμογή LoanApprovalService

5.1.3 Προσέγγιση προσανατολισμένη στις υπηρεσίες (Service-oriented Approach)

Η επιχειρηματική διαδικασία που χρησιμοποιείται συνίσταται στο σχεδιασμό μιας απλής επιχειρηματικής πορείας έγκρισης ή απόρριψης κατά την έκδοση μιας πιστωτικής κάρτας. Η επιχειρηματική περίπτωση χρήσης ακολουθεί τα παρακάτω βήματα :

- ✘ Ο χρήστης πραγματοποιεί μια αίτηση για έκδοση πιστωτικής κάρτας συμπληρώνοντας μια φόρμα που περιέχει πληροφορίες σχετικές με τα προσωπικά και πιστοληπτικά στοιχεία αναγνώρισης του και με το πιστωτικό όριο της κάρτας.
- ✘ Όταν λαμβάνεται η αίτηση για έκδοση κάρτας, τα προσωπικά στοιχεία του χρήστη επαληθεύονται με βάση μια υπάρχουσα βάση δεδομένων και προκύπτει το αποτέλεσμα με βάση τις πληροφορίες του χρήστη και το πιστωτικό όριο που αιτείται.
- ✘ Ανακοινώνεται η απόφαση στο χρήστη, που σε περίπτωση απόρριψης ενημερώνεται και για την αιτία απόρριψης.

Η πορεία που ακολουθείται όσον αφορά την τεχνική και προγραμματιστική πλευρά της εφαρμογής στηρίζεται στη χρήση αρχιτεκτονικών προσεγγίσεων βασισμένων στην SOA αρχιτεκτονική και μιας μηχανής παραγωγής κανόνων που ενσωματώνει την επιχειρηματική λογική και την εταιρική πολιτική σε επιχειρηματικούς κανόνες υλοποιήσιμους σε φυσική γλώσσα. Η επιχειρηματική λογική αναπτύσσεται εξολοκλήρου σε επιχειρηματικούς κανόνες και στη συνέχεια εξωτερικεύει τη λειτουργία της με τη μορφή διαδικτυακής υπηρεσίας που καλείται από την επιχειρηματική διαδικασία για την επεξεργασία και τον έλεγχο των δεδομένων εισόδου. Χρησιμοποιείται το περιβάλλον ανάπτυξης που περιέχεται στο Eclipse IDE για το σχεδιασμό της εφαρμογής των επιχειρηματικών κανόνων και των την παραμετροποίηση τους και στη συνέχεια εκτελείται στον εξυπηρετητή JBoss προς επαλήθευση της ορθής λειτουργίας του.

5.1.3.1 Καθορισμός μηνυμάτων επικοινωνίας

Για την ανταλλαγή δεδομένων μεταξύ της μηχανής παραγωγής επιχειρηματικών κανόνων και του προγραμματιστικού εργαλείου υλοποίησης των επιχειρηματικών διαδικασιών είναι απαραίτητο να αναπτυχθεί ένα μοντέλο αντικειμένων (object model) που, όπως έχει αναφερθεί παραπάνω, αποτελεί το βασικό αντικείμενο που αναγνωρίζει και διαχειρίζεται η μηχανή κανόνων. Στη συγκεκριμένη εφαρμογή αναπτύσσεται ένα απλό XML σχήμα που εκφράζει τα βασικά στοιχεία που απαιτείται να αποστείλει ο χρήστης για την αποτίμηση της αίτησης έκδοσης πιστωτικής κάρτας. Το XML σχήμα αποτελείται από πέντε βασικά στοιχεία (elements) :

- ✘ Το όνομα (name)
- ✘ Την ηλικία (age)
- ✘ Το ετήσιο εισόδημα (annualIncome)
- ✘ Την αξία των περιουσιακών στοιχείων (asset)
- ✘ Την κατάσταση της απάντησης (eligibilityStatus)

Τα τέσσερα πρώτα στοιχεία αποτελούν τα δεδομένα εισόδου που επεξεργάζεται η επιχειρηματική εφαρμογή, ενώ το τελευταίο αποτελεί την έξοδο που επιστρέφει ως απάντηση η εφαρμογή. Η μορφή του σχήματος φαίνεται παρακάτω :

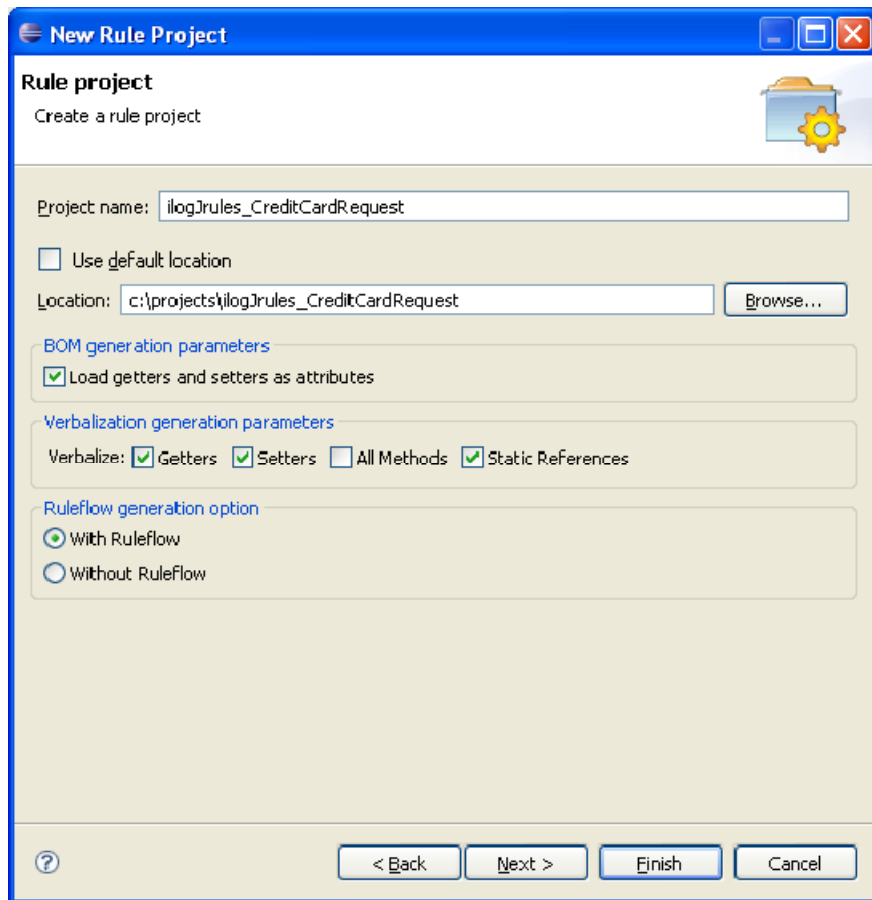
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns="http://www.ilog.com/rules/CreditCardRequest"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.ilog.com/rules/CreditCardRequest"
  elementFormDefault="unqualified"
  attributeFormDefault="unqualified">
  <xsd:element name="CreditCard" type="CreditCard"/>
  <xsd:complexType name="CreditCard">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string"/>
      <xsd:element name="age" type="xsd:int"/>
      <xsd:element name="annualIncome" type="xsd:double"/>
      <xsd:element name="asset" type="xsd:double"/>
      <xsd:element name="eligibilityStatus" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

5.1.3.2 Δημιουργία των επιχειρηματικών κανόνων και της Υπηρεσίας που ενσωματώνει τους επιχειρηματικούς κανόνες

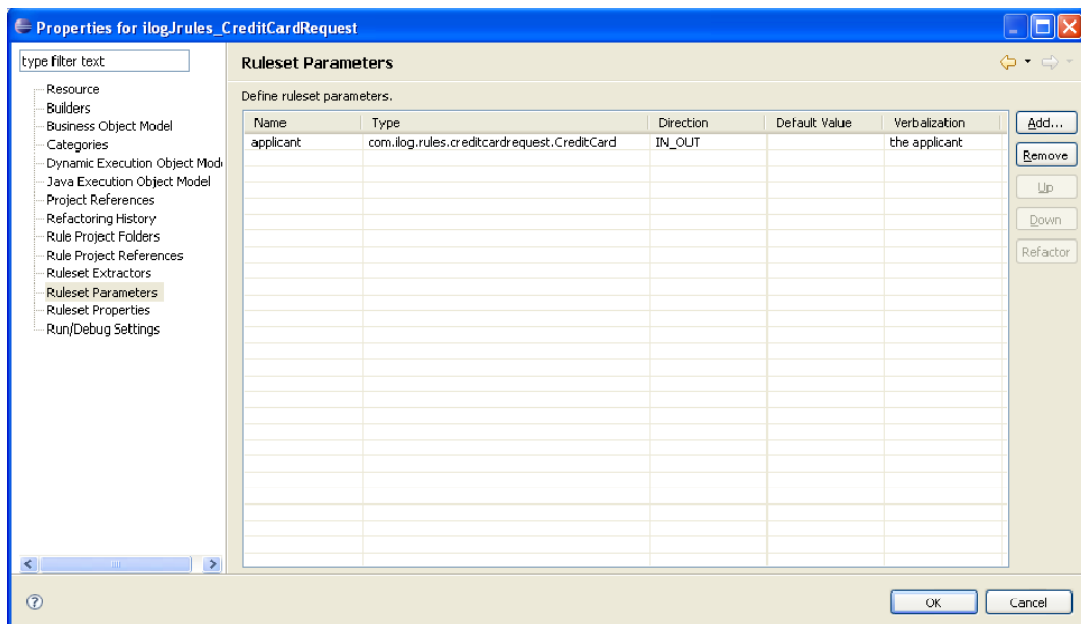
Η δημιουργία των επιχειρηματικών κανόνων απαιτεί τη χρήση του ILOG JRules και στη συνέχεια την εξωτερίκευση τους με τη μορφή διαδικτυακής Υπηρεσίας που μπορεί να κληθεί από την επιχειρηματική διαδικασία που αναπτύσσεται στο περιβάλλον ανάπτυξης που χρησιμοποιείται. Παρακάτω φαίνονται τα βήματα που ακολουθούνται σε σχέση με τη χρήση της μηχανής παραγωγής κανόνων :

1. **Δημιουργία της εφαρμογής στη μηχανή παραγωγή κανόνων (JRules Project):** Αρχικά δημιουργείται η εφαρμογή για την υλοποίηση των κανόνων στη μηχανή παραγωγής με βάση το μοντέλο αντικειμένων που έχει τη μορφή του XML σχήματος που έχουμε ήδη δημιουργήσει. Η εφαρμογή ονομάζεται **ilogJrules_CreditCardRequest** και μετατρέπει σε φυσική γλώσσα τις μεθόδους λήψης και οριοθέτησης των τιμών των παραμέτρων (setters and getters) με την επιλογή της δυνατότητας ‘Verbalize’. Παράλληλα δίνεται η δυνατότητα ανάπτυξης της ροής εκτέλεσης και αποτίμησης των επιχειρηματικών κανόνων μέσω της επιλογής ‘With Ruleflow’.
2. **Καθορισμός των παραμέτρων του συνόλου κανόνων της εφαρμογής (Ruleset parameters):** Ακολουθεί ο καθορισμός των παραμέτρων που συμμετέχουν στον ορισμό και τη δόμηση των επιχειρηματικών κανόνων. Με βάση το σύνολο των καταστάσεων των παραμέτρων που χρησιμοποιούνται πραγματοποιείται ο σχεδιασμός των κανόνων και η πορεία εκτέλεσης τους. Στο περιβάλλον ανάπτυξης Rule Studio που αποτελεί plug-in τμήμα του Eclipse IDE, ορίζεται ως μοναδική παράμετρος για την υλοποίηση της εφαρμογής ο ‘αιτών’ (applicant) που ακολουθεί την προτυποποίηση του XML

σχήματος που έχει δημιουργηθεί και αποτελεί τόσο παράμετρο εισόδου όσο και εξόδου.



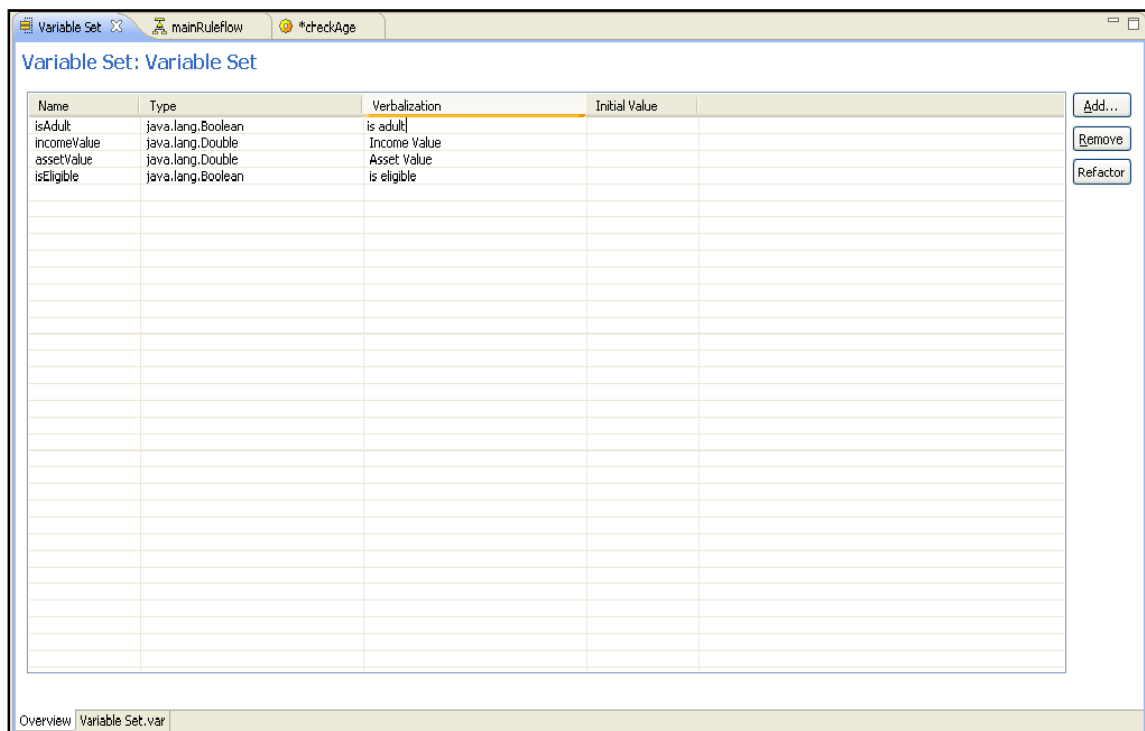
Σχήμα 8-19: Δημιουργία Εφαρμογής Επιχειρηματικών Κανόνων



Σχήμα 8-20: Δημιουργία Παραμέτρων Συνόλου Επιχειρηματικών Κανόνων

3. **Δημιουργία του συνόλου μεταβλητών (Variable Set):** Το επόμενο βήμα περιλαμβάνει τον ορισμό των μεταβλητών που αποτελούν αντικείμενο επεξεργασίας και μετατροπής των επιχειρηματικών κανόνων. Οι μεταβλητές που ορίζονται είναι οι εξής:

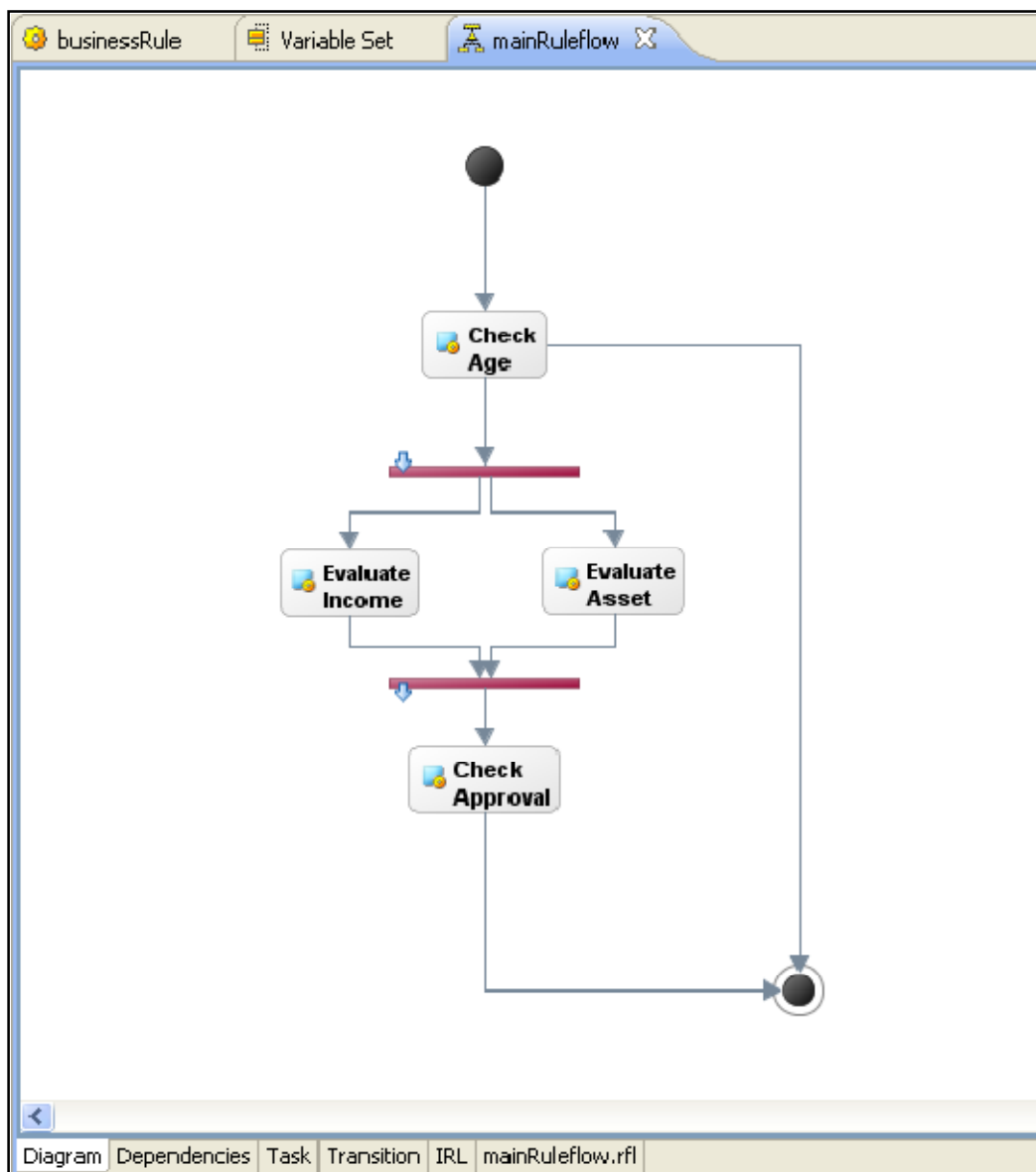
- ✓ **isAdult** που αντιστοιχεί στο δυαδικό τύπο δεδομένων με λογικές τιμές 0 ή 1 (Boolean) και η απόδοση του σε φυσική γλώσσα αντιστοιχεί στην ένδειξη της ενηλικίωσης ή όχι του χρήστη που πραγματοποιεί την αίτηση έκδοσης πιστωτικής κάρτας (is Adult),
- ✓ **incomeValue** που εμφανίζει τύπο δεδομένων διπλής λέξης (double) και η λεκτική του υπόσταση για τη χρήση στους επιχειρηματικούς κανόνες εκφράζει το εισόδημα του χρήστη (Income Value),
- ✓ **assetValue** που εμφανίζει τύπο δεδομένων διπλής λέξης (double) και η λεκτική του υπόσταση για τη χρήση στους επιχειρηματικούς κανόνες εκφράζει την αξία των περιουσιακών στοιχείων του πελάτη (Asset Value),
- ✓ **isEligible** που αντιστοιχεί στο δυαδικό τύπο δεδομένων με λογικές τιμές 0 ή 1 (Boolean) και η απόδοση του σε φυσική γλώσσα αντιστοιχεί στην ένδειξη της δυνατότητας ή όχι του πελάτη για έκδοση πιστωτικής κάρτας (is Eligible).



Σχήμα 8-21: Δημιουργία Μεταβλητών Συνόλου Επιχειρηματικών Κανόνων

4. **Σχεδιασμός της κύριας ροής αποτίμησης και εκτέλεσης των επιχειρηματικών κανόνων (ruleflow):** Στη συνέχεια καθορίζεται με βάση το γραφικό περιβάλλον που παρέχει το Rule Studio, η σειρά με την οποία πρόκειται να εκτελεστούν και να αποτιμηθούν οι επιχειρηματικοί κανόνες.

Στη συγκεκριμένη εφαρμογή χρησιμοποιούνται τέσσερις δραστηριότητες. Αρχικά ελέγχεται η ηλικία του πελάτη που αιτείται την έκδοση πιστωτικής κάρτας (κόμβος Check Age) και με βάση το αποτέλεσμα της αποτίμησης η ροή είτε συνεχίζεται και ακολουθεί ο έλεγχος του ετήσιου εισοδήματος και της περιουσιακής αξίας του πελάτη (σε περίπτωση που ο πελάτης είναι ενήλικος μέσω του κόμβου Evaluate Income και Evaluate Asset), είτε διακόπτεται και η διαδικασία τερματίζεται απορρίπτοντας την αίτηση. Τέλος, σε περίπτωση που ο πελάτης ικανοποιεί το ελάχιστο όριο ηλικίας και έχει πραγματοποιηθεί ο έλεγχος της οικονομικής κατάστασης του, ακολουθεί η απόφαση έγκρισης ή απόρριψης της έκδοσης πιστωτικής κάρτας (κόμβος Check Approval).

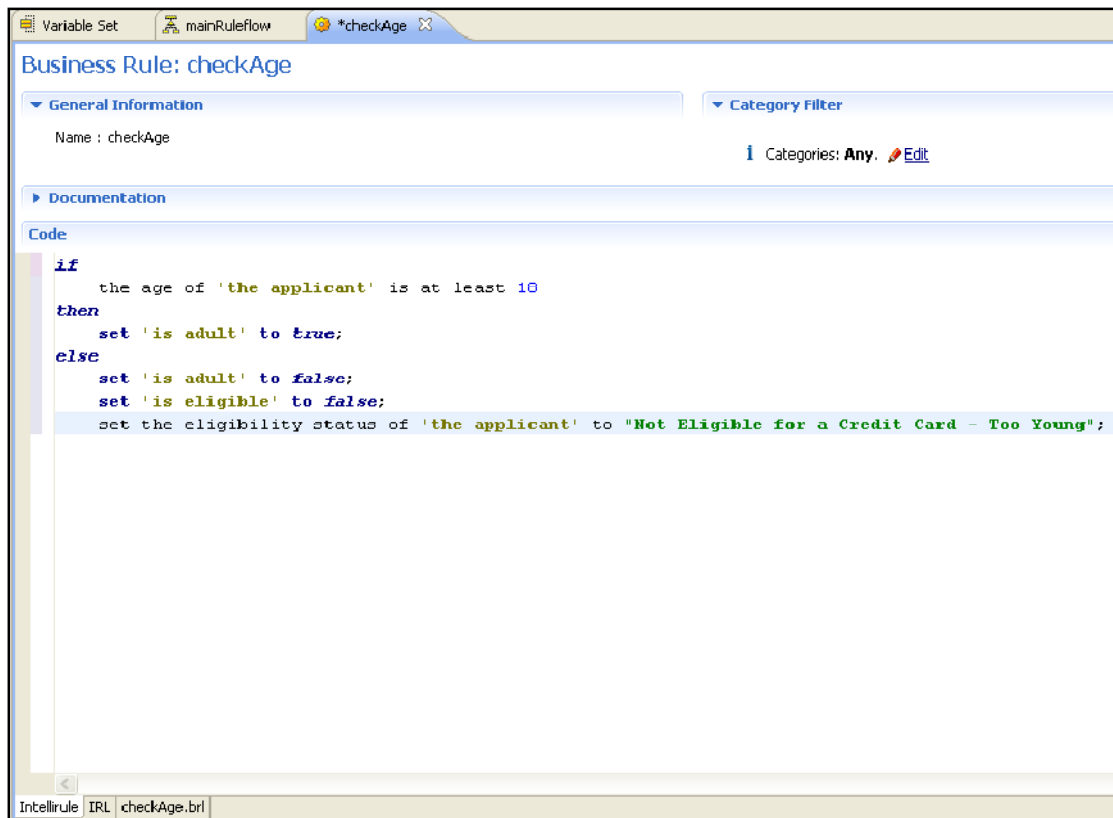


Σχήμα 8-22: Δημιουργία Ροής Αποτίμησης και Εκτέλεσης Επιχειρηματικών Κανόνων

5. **Δημιουργία των επιχειρηματικών κανόνων:** Ακολουθεί το βήμα της δημιουργίας των επιχειρηματικών κανόνων με βάση τους οποίους ελέγχεται

κα μπορεί να μετατραπεί η επιχειρησιακή πολιτική που ακολουθείται κατά τη διαδικασία έκδοσης πιστωτικής κάρτας. Η υλοποίηση που επιχειρείται στηρίζεται στην ύπαρξη των τεσσάρων παρακάτω κανόνων:

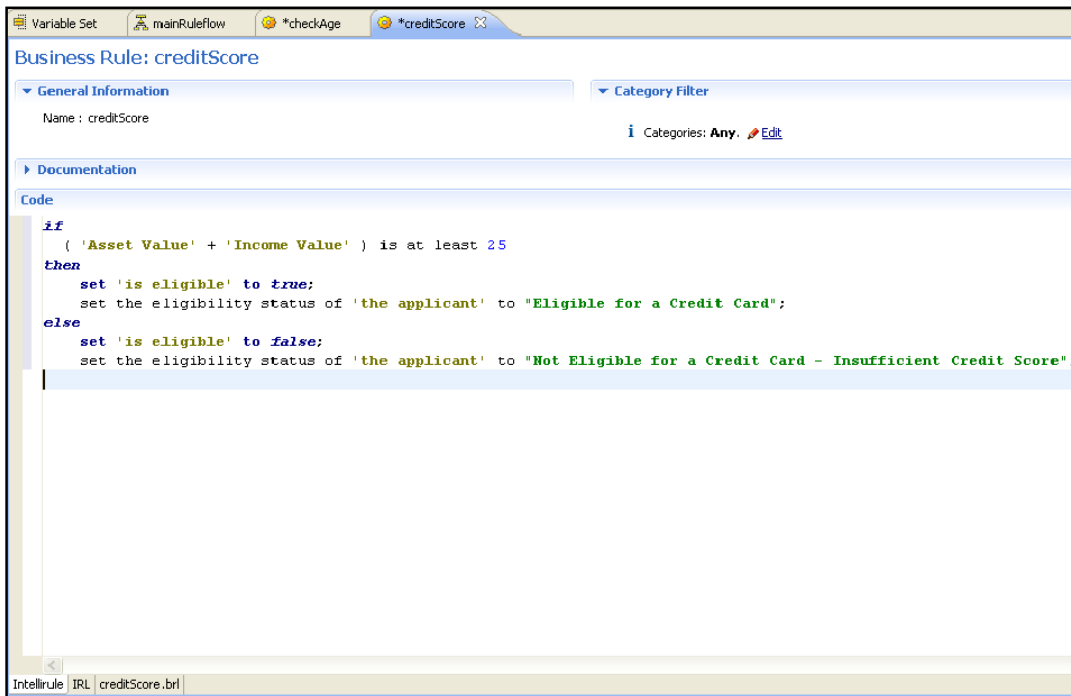
- ✓ **checkAge Rule:** Ο συγκεκριμένος επιχειρηματικός κανόνας ελέγχει την ηλικία του ατόμου που πραγματοποιεί την αίτηση. Σε περίπτωση που η ηλικία του αιτούντα είναι μικρότερη των δεκαοκτώ (18) ετών, η αίτηση απορρίπτεται με την αιτιολογία της εξαιρετικά νεαρής ηλικίας ("Not Eligible for a Credit Card – Too Young").



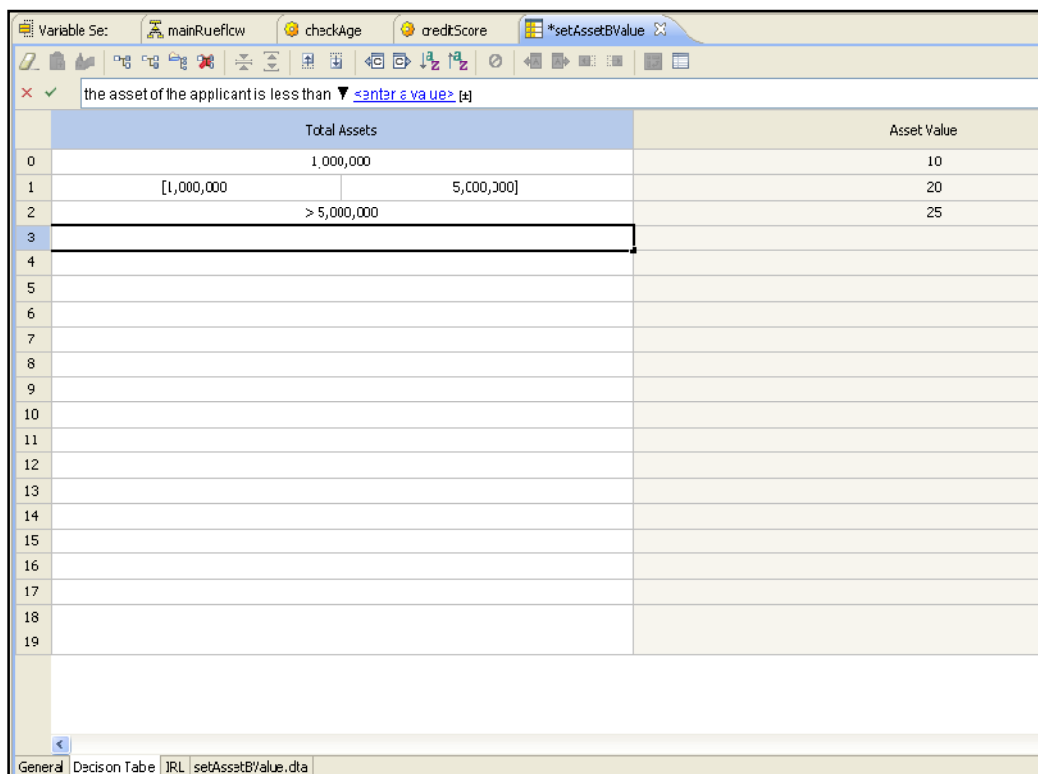
Σχήμα 8-23: Δημιουργία Επιχειρηματικού Κανόνα checkAge

- ✓ **creditScore Rule:** Ο συγκεκριμένος επιχειρηματικός κανόνας ελέγχει την πιστοληπτική αξιοπιστία του πελάτη που πραγματοποιεί την αίτηση. Σε περίπτωση που η τιμή της πιστωτικής ικανότητας του αιτούντα είναι μικρότερη των εικοσιπέντε (25), η αίτηση απορρίπτεται με την αιτιολογία της ανεπαρκούς πιστοληπτικής αξίας ("Not Eligible for a Credit Card – Insufficient Credit Score").
- ✓ **setAssetValue Rule:** Ο συγκεκριμένος κανόνας οδηγεί στην αποτίμηση της αξίας του περιουσιακού χαρτοφυλακίου του πελάτη που πραγματοποιεί την αίτηση έκδοσης πιστωτικής κάρτας. Η μορφή που ακολουθεί ο κανόνας στηρίζεται στο σχεδιασμό ενός πίνακα αποφάσεων (Decision Table), στον οποίο με βάση την χρηματική αξία των περιουσιακών στοιχείων δίνεται μια σχετική τιμή αξιολόγησης των περιουσιακών στοιχείων σε ένα καθορισμένο σύστημα τιμών. Η

πρώτη στήλη του πίνακα αποφάσεων καθορίζει το πεδίο τιμών της χρηματικής αξίας των περιουσιακών στοιχείων και η δεύτερη στήλη θέτει με βάση την προηγούμενη κατηγοριοποίηση την ανάλογη τιμή της περιουσιακής αξίας.



Σχήμα 8-24: Δημιουργία Επιχειρηματικού Κανόνα creditScore



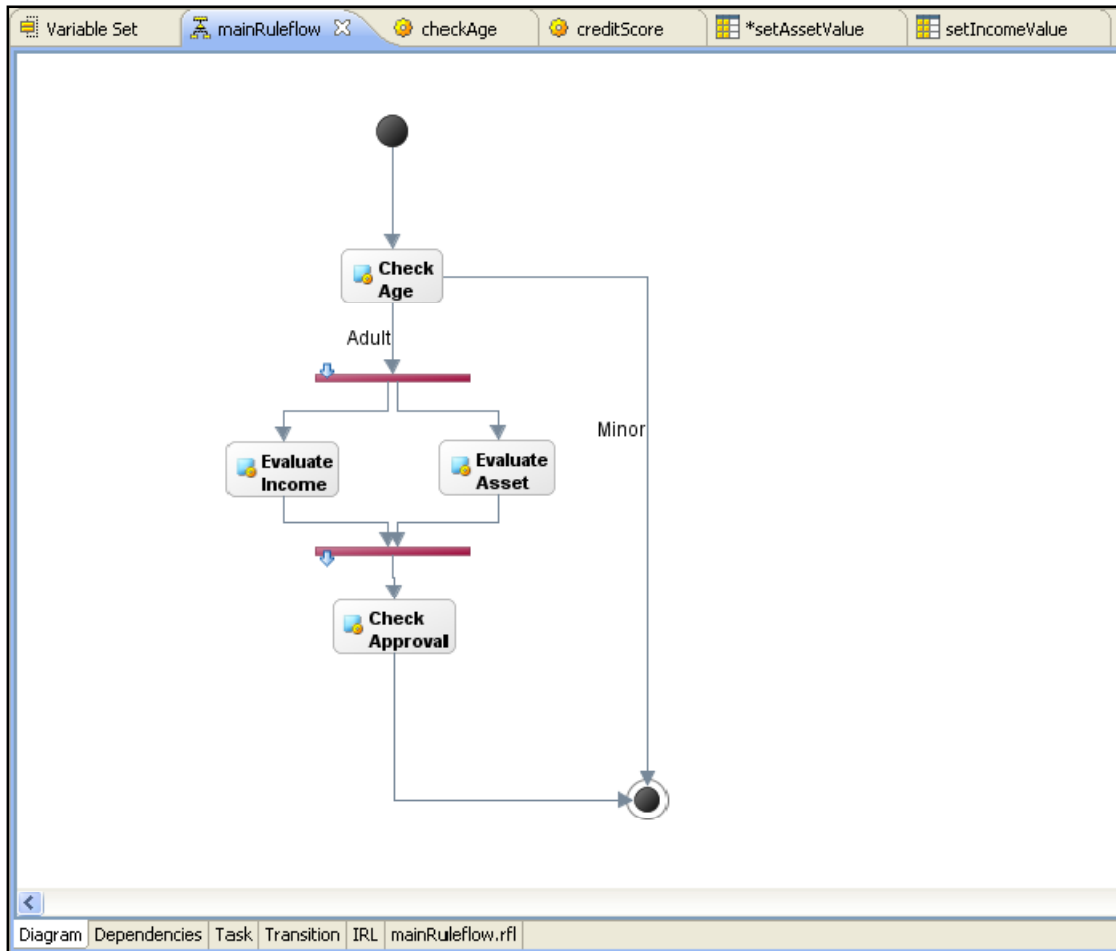
Σχήμα 8-25: Δημιουργία Πίνακα Αποφάσεων setAssetValue

- ✓ **setIncomeValue Rule:** Ο συγκεκριμένος κανόνας οδηγεί στην αποτίμηση της αξίας του ετήσιου εισοδήματος του πελάτη που πραγματοποιεί την αίτηση έκδοση πιστωτικής κάρτας. Η μορφή που ακολουθεί ο κανόνας στηρίζεται στο σχεδιασμό ενός πίνακα αποφάσεων (Decision Table), στον οποίο με βάση το ποσό του ετήσιου εισοδήματος δίνεται μια σχετική τιμή αξιολόγησης του εισοδήματος σε μία καθορισμένη κλίμακα τιμών. Η πρώτη στήλη του πίνακα αποφάσεων καθορίζει το πεδίο τιμών του ποσού του ετήσιου εισοδήματος και η δεύτερη στήλη θέτει με βάση την προηγούμενη κατηγοριοποίηση την ανάλογη τιμή της εισοδηματικής αξίας.

	Annual Income	Income Value
0	50,000	10
1	[50,000, 80,000]	20
2]80,000, 100,000[25
3	> 100,000	30
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		

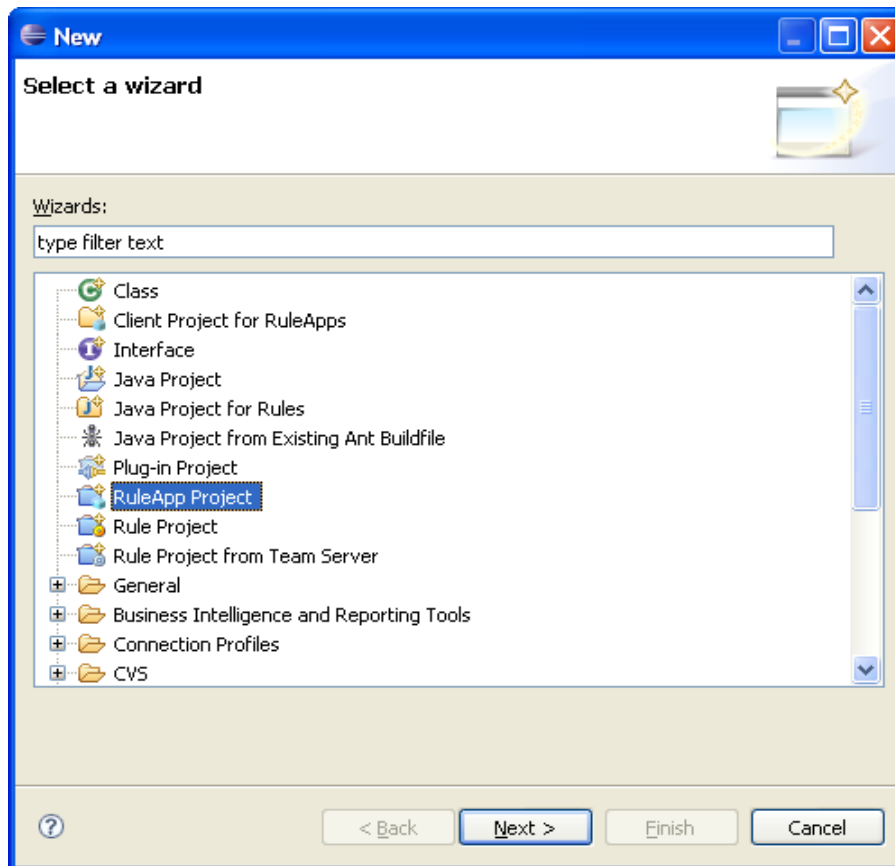
Σχήμα 8-26: Δημιουργία Πίνακα Αποφάσεων setIncomeValue

6. **Συσχέτιση των επιχειρηματικών κανόνων με τις δραστηριότητες της ροής εκτέλεσης των κανόνων:** Στη συνέχεια πραγματοποιείται η σύνδεση των δραστηριοτήτων της ροής αποτίμησης και εκτέλεσης των επιχειρηματικών κανόνων με τους κανόνες που δημιουργήθηκαν. Η συσχέτιση είναι προφανής, ενώ συγχρόνως αποδίδεται στο σημείο μεταβολής της ροής που ακολουθεί τον έλεγχο της ηλικίας η δομή if....else για την επιλογή του μονοπατιού εκτέλεσης.

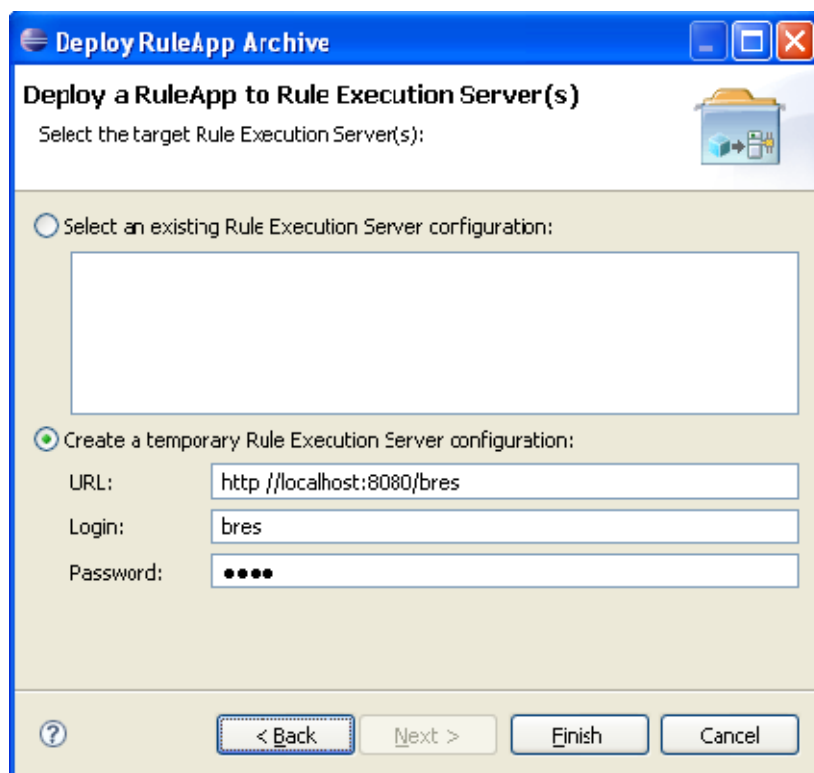


Σχήμα 8-27: Συσχέτιση και Αντιστοίχιση Ροής Εκτέλεσης και Επιχειρηματικών Κανόνων

- 7. Κατασκευή της εφαρμογής επιχειρηματικών κανόνων και ανάπτυξης της στον εξυπηρετητή:** Για την κατασκευή της εφαρμογής επιλέγεται από το περιβάλλον ανάπτυξης του Rule Studio η επιλογή RuleApp Project, δίνεται το όνομα **ilogJrules_CreditCardApp** και αναπτύσσεται (Deploy) στον εξυπηρετητή προς εκτέλεση στην θύρα του τοπικού δικτύου (localhost) που 'ακούει' η μηχανή παραγωγής κανόνων μέσω του εξυπηρετητή Rule Execution Server.

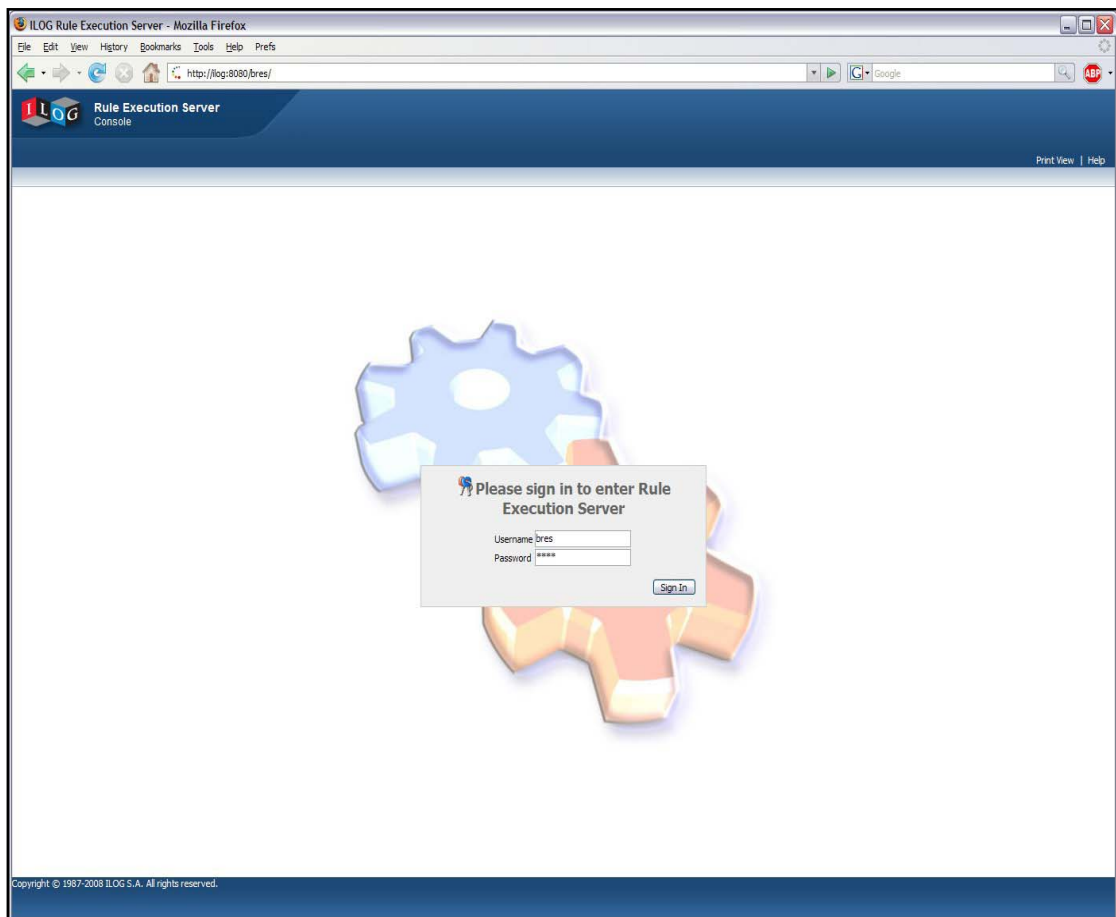


Σχήμα 8-28: Δημιουργία Εκτελέσιμης Εφαρμογής



Σχήμα 8-29: Ανάπτυξη Εφαρμογής στον Εξυπηρετητή

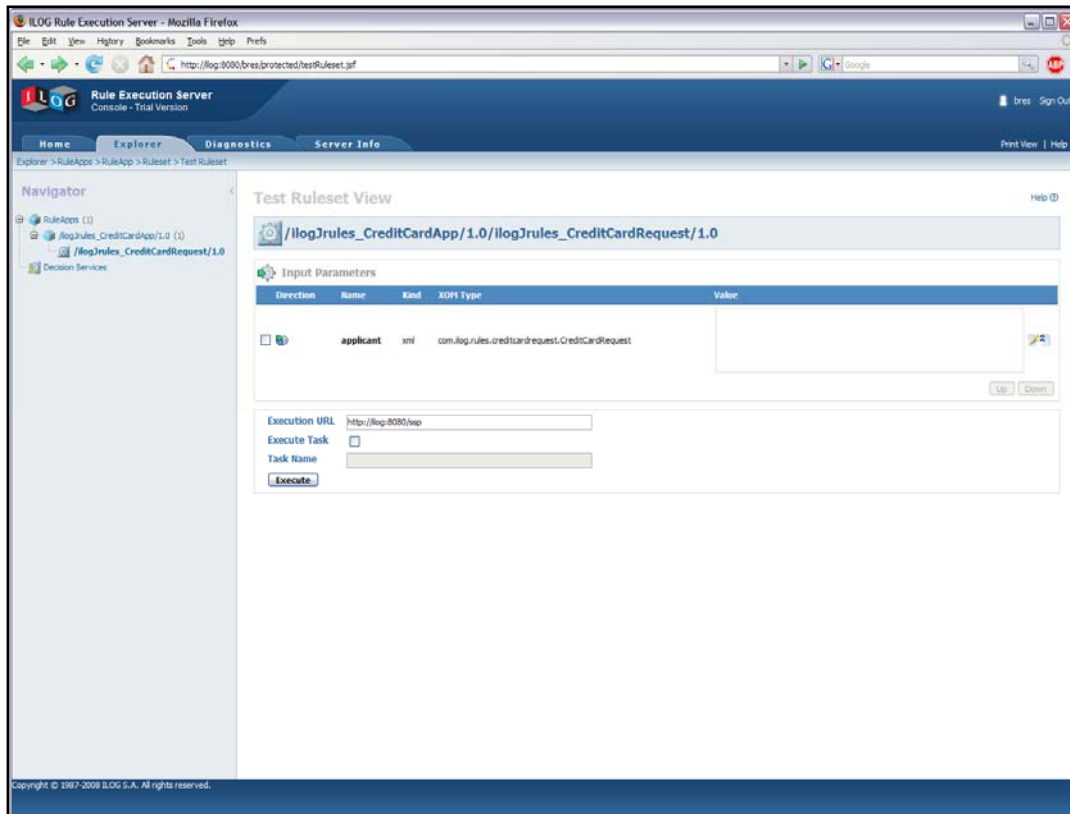
8. Επαλήθευση της ορθότητας εκτέλεσης και εξωτερίκευση των επιχειρηματικών κανόνων με βάση τη γλώσσα WSDL: Το τελευταίο στάδιο της υλοποίησης και εκτέλεσης των επιχειρηματικών κανόνων στη μηχανή παραγωγής είναι η επαλήθευση της ορθής εκτέλεσης τους και η έκθεση τους με βάση τη γλώσσα WSDL, ώστε να γίνει δυνατή η κλήση τους από την επιχειρηματική διαδικασία που πρόκειται να αναπτυχθεί στο περιβάλλον NetBeans 6.1. Η πρόσβαση στο γραφικό περιβάλλον που παρέχει ο εξυπηρετητής JRules Rule Execution Server προς εκτέλεση της εφαρμογής, πραγματοποιείται πληκτρολογώντας την τοποθεσία <http://localhost:8080/bres> στον φυλλομετρητή ιστού που χρησιμοποιείται και τοποθετώντας τα πιστοποιητικά `bres/bres` ως όνομα χρήστη (username) και κωδικό (password).



Σχήμα 8-30: Είσοδος στον Εξυπηρετητή Rule Execution Server

Στη συνέχεια μεταφερόμαστε στην τοποθεσία:

RuleApps→/ilogJrules_CreditCardApp/1.0→/ilogJrules_CreditCardRequest/1.0, όπου επιλέγεται η δυνατότητα ελέγχου του συνόλου των επιχειρηματικών κανόνων (Test Ruleset).



Σχήμα 8-31: Έλεγχος Ορθότητας Συνόλου Επιχειρηματικών Κανόνων στον Εξυπηρετητή Rule Execution Server

Τέλος εισάγουμε το παρακάτω μήνυμα προς επαλήθευση και έλεγχο του συνόλου κανόνων, εκτελούμε με επιτυχία την εφαρμογή και εξωτερικεύουμε τη συμπεριφορά και τη λειτουργία των επιχειρηματικών κανόνων σε WSDL μορφή μέσω της εντολής *'get WSDL for this ruleset version'*.

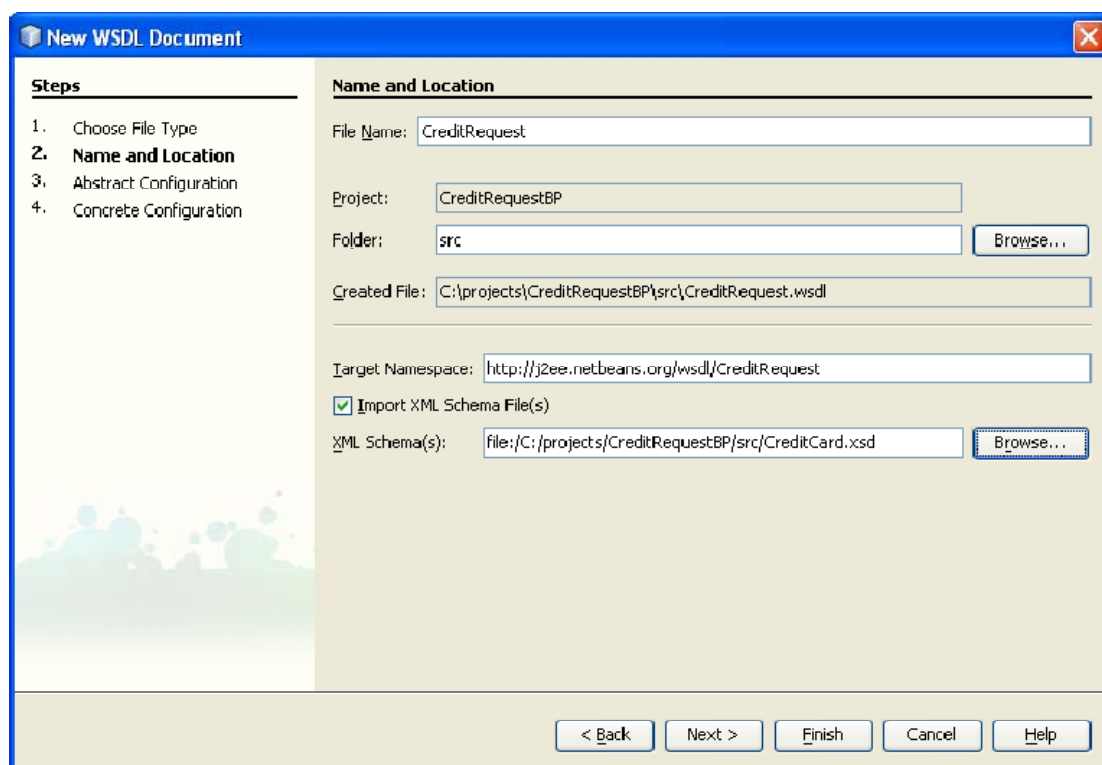
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CreditCard>
  <name>test</name>
  <age>50</age>
  <annualIncome>100000.00</annualIncome>
  <asset>1000000.00</asset>
  <eligibilityStatus>Don't Know</eligibilityStatus>
</CreditCard>
```

5.1.3.3 Δημιουργία της επιχειρηματικής διαδικασίας BPEL

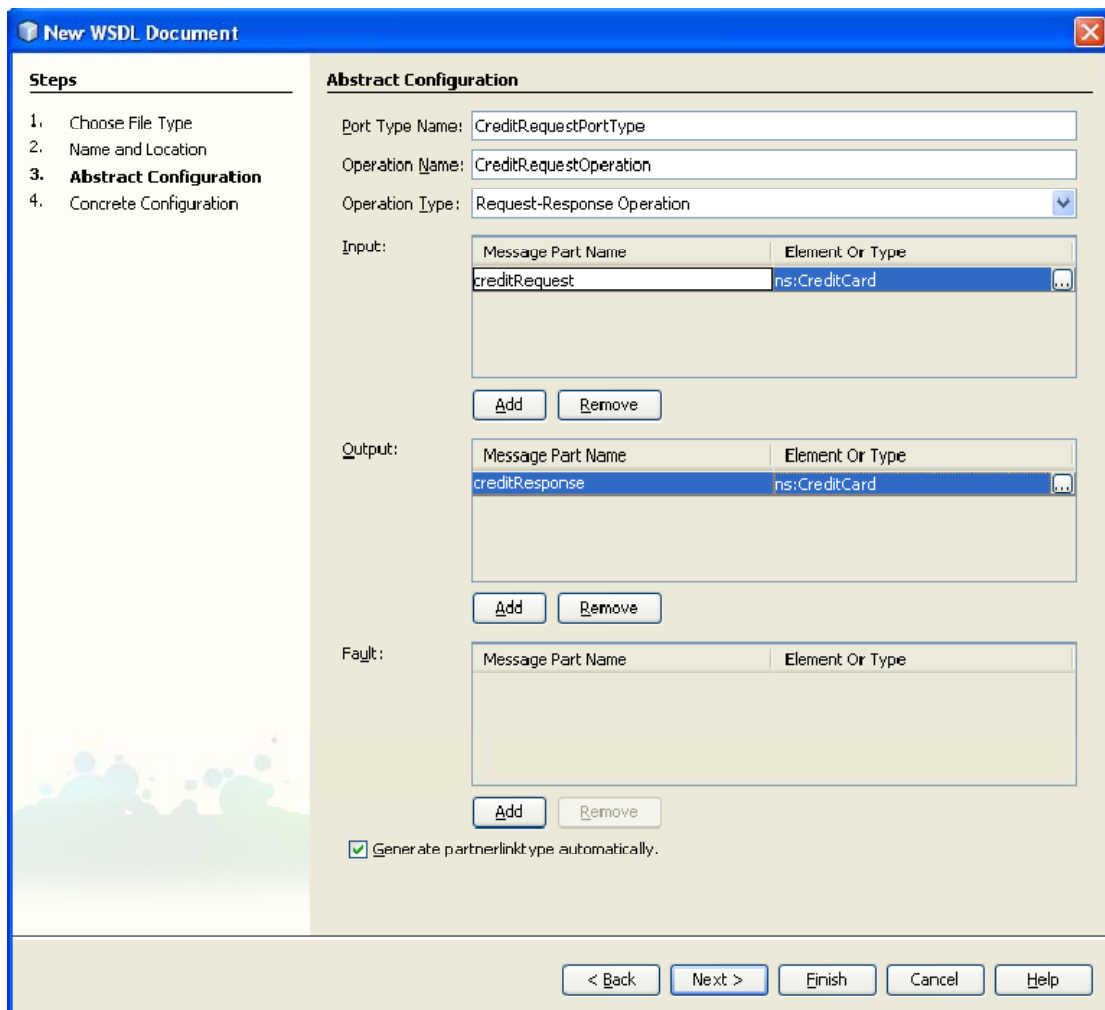
Ως προς την υλοποίηση της επιχειρηματικής διαδικασίας και της κλήσης του συνόλου των επιχειρηματικών κανόνων που έχουν σχεδιαστεί, αρχικά δημιουργούμε μια BPEL εφαρμογή με τη μορφή μονάδας λογισμικού επιχειρηματικής λογικής (BPEL module) που μπορεί να χρησιμοποιηθεί από τους JBI παράγοντες και να εκτελεστεί στον

εξυπηρετητή εφαρμογών με το όνομα CreditCardRequestBP. Στη συνέχεια εισάγουμε στο περιβάλλον ανάπτυξης NetBeans 6.1 το κατάλληλο XML σχήμα CreditCard.xsd για την περιγραφή των απαραίτητων στοιχείων που αποτελούν τα δεδομένα εισόδου και εξόδου της επιχειρηματικής διαδικασίας και παρουσιάζουν τη μορφή πολύπλοκης δομής δεδομένων.

Ακολουθεί ως τμήμα της BPEL εφαρμογής ο σχεδιασμός της επιχειρηματικής διαδικασίας με όνομα CreditRequestProcess, ο καθορισμός του ρόλου των συνεργατών που συμμετέχουν και η εισαγωγή της διαδικτυακής υπηρεσίας που καλεί. Αρχικά δημιουργούμε ένα WSDL αρχείο που έχει το ρόλο του πελάτη και δέχεται ως είσοδο τα δεδομένα που απαιτούνται για την αίτηση έκδοσης πιστωτικής κάρτας και επιστρέφει ως έξοδο το αποτέλεσμα της αποτίμησης των χαρακτηριστικών της αίτησης. Πρόκειται για το CreditRequest WSDL αρχείο που επικοινωνεί μέσω XML μηνυμάτων (που ακολουθούν την τυποποίηση του XML σχήματος CreditCard.xsd) με τη διαδικασία και επιτελεί τη λειτουργία της αίτησης έκδοσης πιστωτικής κάρτας με το μήνυμα CreditRequest και τη λειτουργία της αποδοχής της απάντησης με το μήνυμα CreditResponse.

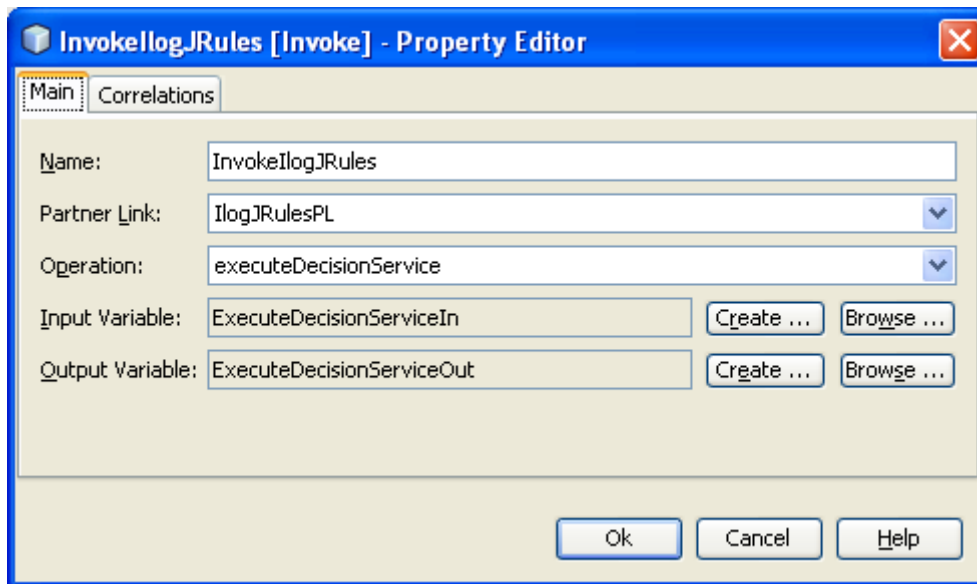


Σχήμα 8-32: Δημιουργία WSDL Αρχείου CreditRequest του Πελάτη της Διαδικασίας



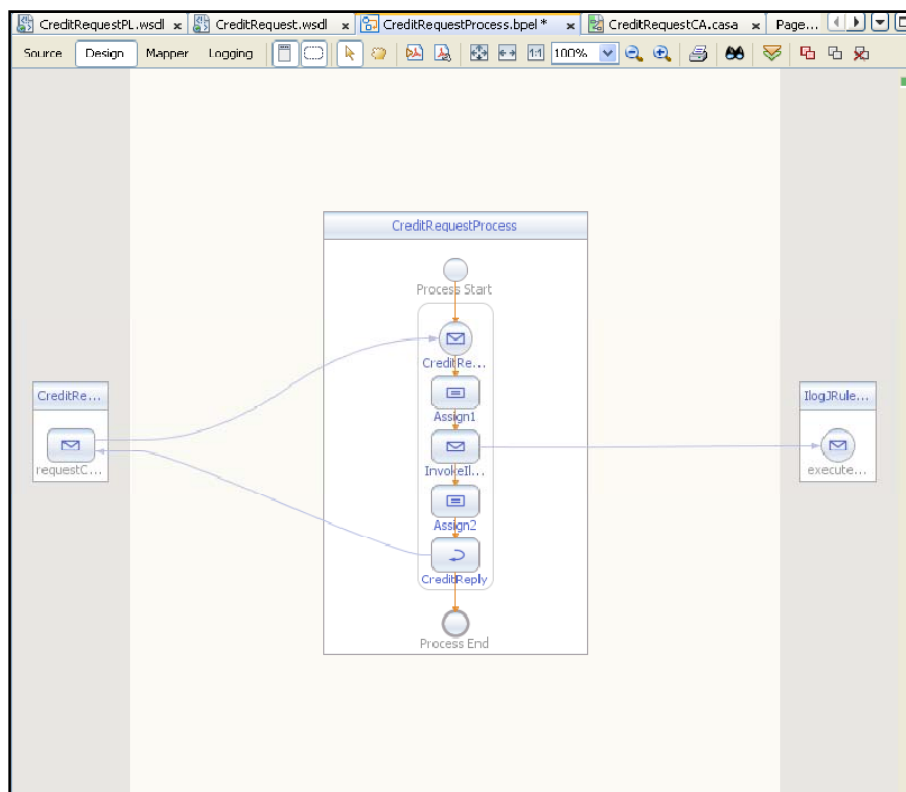
Σχήμα 8-33: Δημιουργία WSDL Αρχείου CreditRequest του Πελάτη της Διαδικασίας

Η επιχειρηματική διαδικασία μέσω ενός αντικειμένου Receive με το όνομα CreditReceive δέχεται τα δεδομένα που εισάγει ο χρήστης και με βάση την πορεία που ακολουθείται τα επεξεργάζεται και επιστρέφει ως απάντηση το αποτέλεσμα στο χρήστη μέσω του αντικειμένου Reply με το όνομα CreditReply. Για την επεξεργασία των δεδομένων είναι απαραίτητη η κλήση του συνόλου των επιχειρηματικών κανόνων που έχουν δημιουργηθεί στο ILOG JRules, ώστε να αποτιμηθούν τα στοιχεία που έχουν εισαχθεί για την αίτηση της έκδοσης πιστωτικής κάρτας. Η ενσωμάτωση της συγκεκριμένης υπηρεσίας είναι απλή και απαιτεί την δημιουργία ενός τύπου συνδέσμου – συνεργάτη (Partner Link Type) που σχεδιάζεται με βάση την αφαιρετική απεικόνιση των επιχειρηματικών κανόνων στη WSDL γλώσσα και καλείται μέσω του αντικειμένου δραστηριότητας Invoke με το όνομα InvokeILOGJRules.



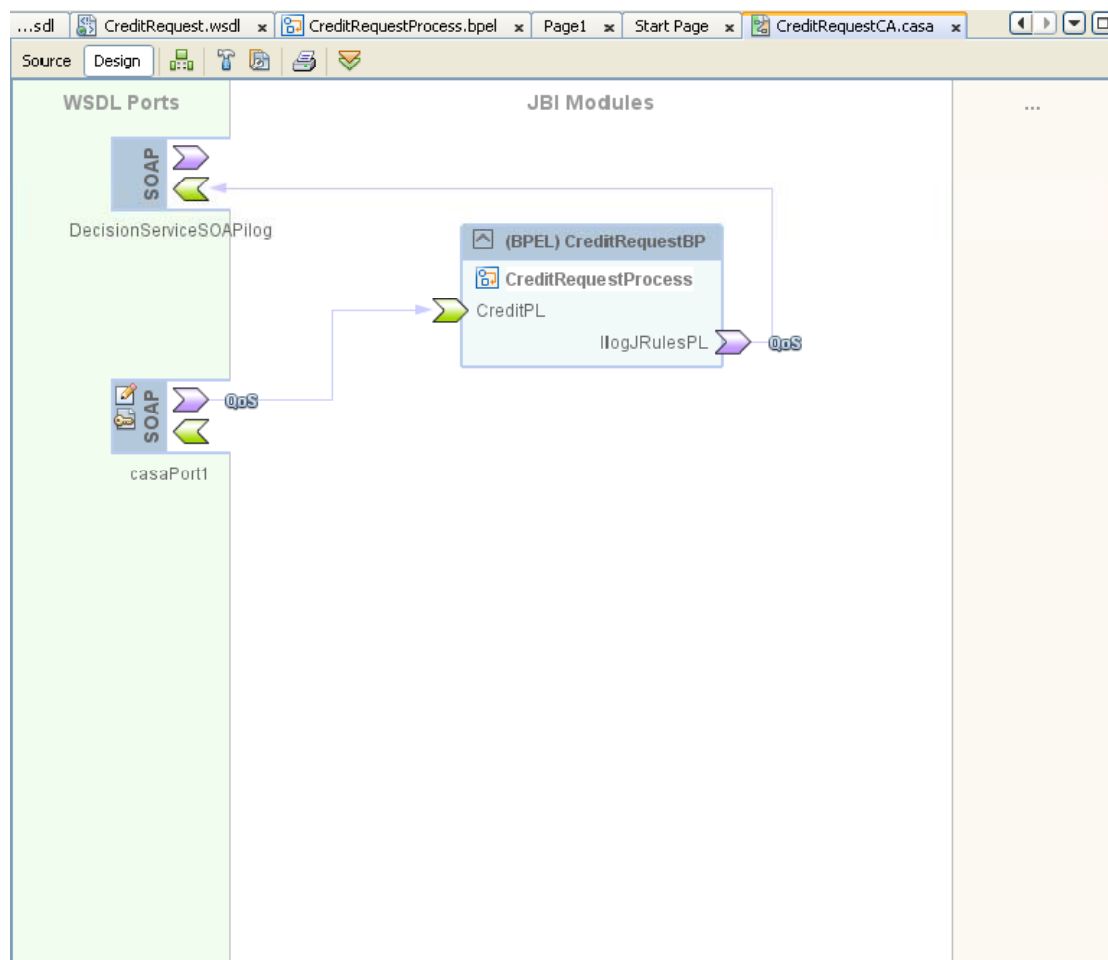
Σχήμα 8-34: Δημιουργία Δραστηριότητας Κλήσης του WSDL Αρχείου Ενσωμάτωσης των Επιχειρηματικών Κανόνων

Γίνεται φανερό ότι με το συγκεκριμένο τρόπο υλοποίησης, ενσωματώνεται η επιχειρησιακή πολιτική σε ένα σύνολο κανόνων σε φυσική γλώσσα που μπορεί να μεταβληθεί οποιαδήποτε στιγμή χωρίς να επηρεάζεται η μορφή της επιχειρηματικής διαδικασίας, με αποτέλεσμα να επιτυγχάνεται ευελιξία και προσαρμογή στα ευμετάβλητα δεδομένα της παγκόσμιας αγοράς.



Σχήμα 8-35: Δημιουργία Επιχειρηματικής Διαδικασίας CreditRequestProcess

Τέλος δημιουργείται η σύνθετη εφαρμογή CreditRequestCA προκειμένου να ελεγχθεί η ορθή εκτέλεση της διαδικασίας και η ορθή επικοινωνία και ανταλλαγή μηνυμάτων μεταξύ των συναλλασσόμενων πλευρών, όπως πράγματι συμβαίνει με βάση το παρακάτω σχήμα όπου φαίνεται η αλληλεπίδραση και επικοινωνία μέσω του πρωτοκόλλου SOAP.



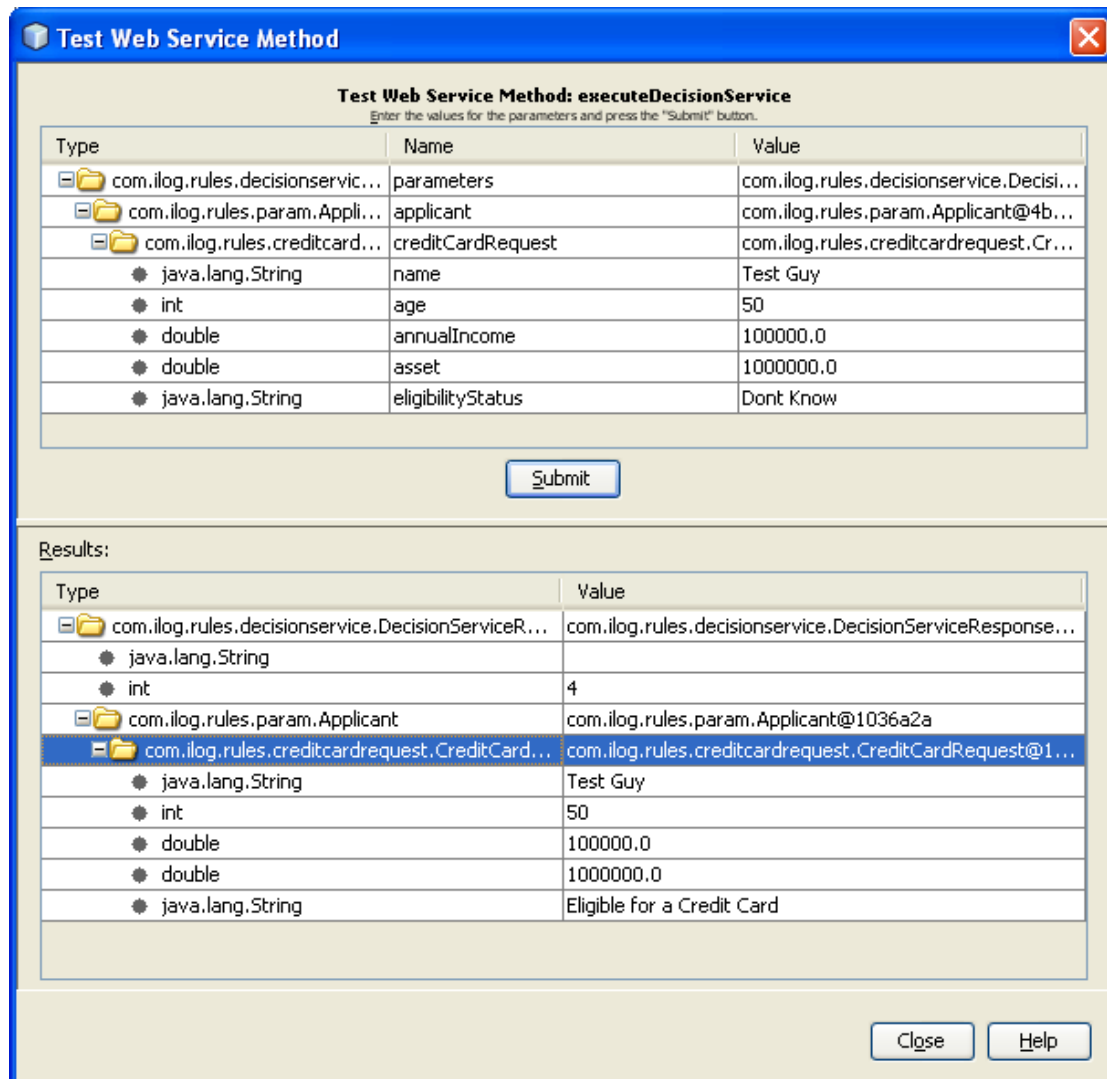
Σχήμα 8-36: Δημιουργία Σύνθετης Επιχειρηματικής Εφαρμογής CreditRequestBP

Παράλληλα, δίνεται ευκαιρία να αναδειχθεί μια επιπλέον δυνατότητα που προσφέρει το περιβάλλον ανάπτυξης NetBeans και ο εξυπηρετητής εφαρμογών GlassFish και η οποία σχετίζεται με τη δυνατότητα καταγραφής και δημοσίευσης των διαδικτυακών υπηρεσιών στην Αποθήκη Υπηρεσιών (Service Repository) που βρίσκεται στο NetBeans. Η διαδικασία που πρέπει να ακολουθηθεί περιλαμβάνει τα παρακάτω βήματα:

- ✓ Από την ετικέτα των Υπηρεσιών (Services Tab) που υπάρχει στο περιβάλλον ανάπτυξης NetBeans επιλέγουμε τη δυνατότητα προσθήκης διαδικτυακής Υπηρεσίας (Add Web Service).
- ✓ Στη συνέχεια επιλέγουμε είτε τη δυνατότητα προσθήκης της Υπηρεσίας μέσω του τοπικού καταλόγου αρχείων (όπως στην περίπτωση της Υπηρεσίας που περιέχει τους επιχειρηματικούς κανόνες) ή μέσω του URL της

Υπηρεσίας (όπως στην περίπτωση των υπόλοιπων Υπηρεσιών που υλοποιήθηκαν αποκλειστικά στο περιβάλλον ανάπτυξης NetBeans).

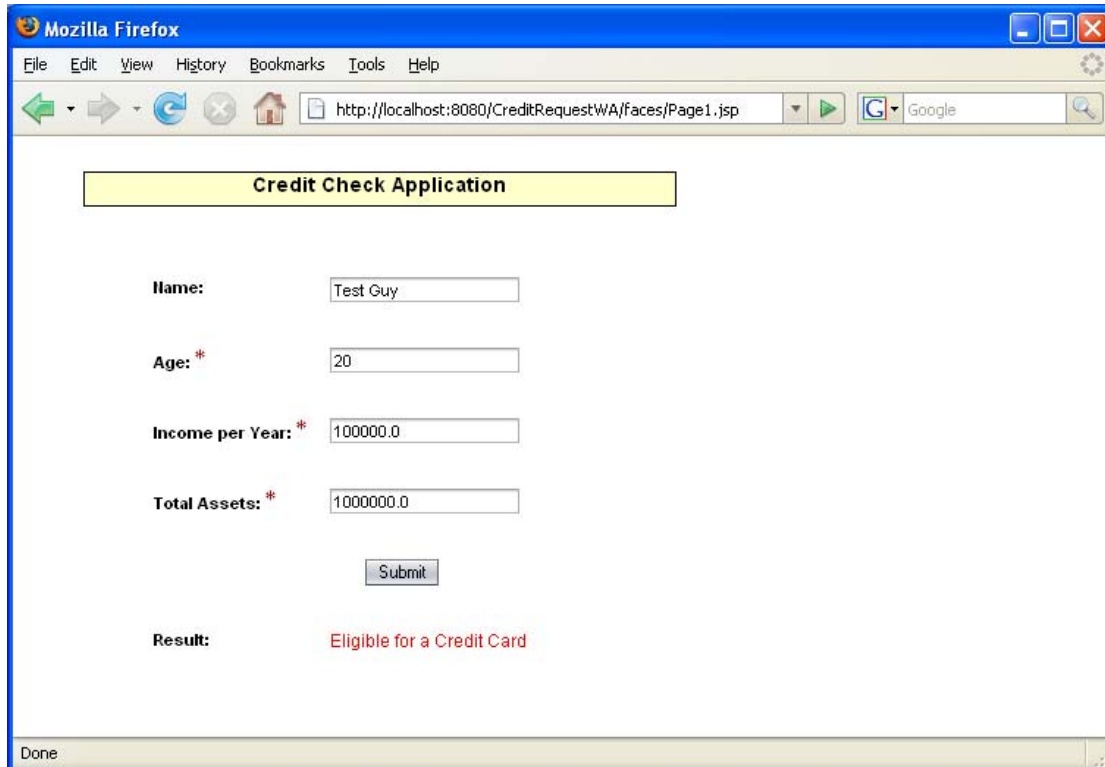
Το αποτέλεσμα είναι να δημιουργηθεί ένας κατάλογος Υπηρεσιών (Service Registry) ή με άλλα λόγια μια Αποθήκη Υπηρεσιών (Service Repository), ώστε κάθε φορά να επιλέγεται και να επαναχρησιμοποιείται η κατάλληλη υπηρεσία κατά την περαιτέρω σύνθεση σύνθετων επιχειρηματικών διαδικασιών, καθώς επίσης και να ελέγχεται συνεχώς η ορθή λειτουργία της.



Σχήμα 8-37: Καταγραφή και Έλεγχος Ορθής Εκτέλεσης της Επιχειρηματικής Διαδικασίας

Συγχρόνως, μπορούμε να δημιουργήσουμε μια μορφή γραφικής διεπαφής (frontend interface) μεταξύ της εφαρμογής και του χρήστη, με αποτέλεσμα να επιτύχουμε ένα ολοκληρωμένο περιβάλλον και μια πλήρη εφαρμογή ηλεκτρονικού επιχειρείν. Στη συγκεκριμένη υλοποίηση, θεωρήθηκε προτιμότερη επιλογή ο σχεδιασμός ενός διαδικτυακού περιβάλλοντος διεπαφής μέσω της τεχνολογίας του Java Server Faces που παρέχεται από την πλατφόρμα Java EE. Η δημιουργία βασίζεται σε γραφικό κυρίως περιβάλλον σχεδίασης διαδικτυακών εντολών και πλαισίων. Τα εργαλεία υλοποίησης είναι τα γνωστά από την γλώσσα HTML κουμπιά (buttons) και περιοχές

κειμένου (Text Fields) που συνδέονται με τα backend δεδομένα της διαδικτυακής Υπηρεσίας. Το τελικό αποτέλεσμα είναι η παρακάτω διαδικτυακή εφαρμογή που δέχεται ως δεδομένα τις παραμέτρους για την έκδοση πιστωτικής κάρτας και επιστρέφει το αποτέλεσμα της αίτησης με βάση την κλήση της σύνθετης διαδικτυακής Υπηρεσίας CreditRequestProcess:



The screenshot shows a Mozilla Firefox browser window displaying a web application titled "Credit Check Application". The browser's address bar shows the URL "http://localhost:8080/CreditRequestWA/faces/Page1.jsp". The application form contains the following fields and values:

Field	Value
Name:	Test Guy
Age: *	20
Income per Year: *	100000.0
Total Assets: *	1000000.0

Below the input fields is a "Submit" button. The "Result:" field displays the text "Eligible for a Credit Card" in red. The browser's status bar at the bottom shows "Done".

Σχήμα 8-38: Γραφική Διεπαφή Επιχειρηματικής Διαδικασίας

6 Επίλογος – Συμπεράσματα

Με βάση την υλοποίηση και τη μελέτη των παραπάνω περιπτώσεων ενσωμάτωσης της επιχειρηματικής λογικής σε επιχειρησιακές εφαρμογές μπορούμε να καταλήξουμε σε σημαντικά συμπεράσματα :

- ✘ Η προσέγγιση που στηρίζεται στην ενσωμάτωση της επιχειρηματικής λογικής σε επίπεδο προγραμματιστικού κώδικα μπορεί να χρησιμοποιηθεί μόνο σε περιπτώσεις που περιλαμβάνουν εξαιρετικά απλές επιχειρηματικές διαδικασίες και δεν υπάρχει η ανάγκη μεταβολής τους μετά την εκτέλεση της επιχειρησιακής εφαρμογής. Παράλληλα, εκτός από το γεγονός της ανελαστικότητας κατά τη διαδικασία μεταβολών λόγω της ανάγκης που δημιουργείται για επέμβαση στον κώδικα λογισμικού, προστίθεται και το πρόβλημα που δημιουργεί η διαχείριση τους μόνο από άτομα που έχουν σχέση με τεχνικά υπολογιστικά θέματα.
- ✘ Η προσέγγιση με χρήση σημείων αποφάσεων και διαχείρισης της επιχειρηματικής διαδικασίας με βάση ένα πρότυπο υλοποίησης ροών εργασίας, παρέχει μια μορφή ελαστικότητας και ευελιξίας, με αποτέλεσμα να εφαρμόζεται σε περιπτώσεις επιχειρησιακών εφαρμογών που απαιτούν συχνές μεταβολές και αποδέσμευση της επιχειρηματικής λογικής από τη διαδικαστική λογική (process logic). Πρόκειται για ικανοποιητική προσέγγιση που όμως δεν αντιμετωπίζει στο σύνολο του το πρόβλημα της καθολικής κατανόησης από άτομα χωρίς γνώσεις πληροφορικής, ενώ υπάρχει επίσης κάποια μορφή εξάρτησης από τη μορφή της διαδικασίας χωρίς να καλύπτει την περίπτωση συνεχών μεταβολών που επιδρούν στην επιχειρηματική λογική προσθέτοντας νέες συνθήκες στα σημεία αποφάσεων και νέους συνεργάτες που πρέπει να οριστούν.
- ✘ Η προσανατολισμένη στις Υπηρεσίες προσέγγιση εξυπηρετεί την περίπτωση που υπάρχει ανάγκη για δεδομένα από διαφορετικές επιχειρηματικές διαδικασίες ή εφαρμογές για τη λήψη αποφάσεων και την περίπτωση που υπάρχει ανάγκη για ενσωμάτωση της ευμετάβλητης επιχειρησιακής πολιτικής και του στρατηγικού σχεδιασμού. Συγχρόνως παρέχει πραγματική ευελιξία και προσαρμοστικότητα με την εισαγωγή της έννοιας των επιχειρηματικών κανόνων σε κατανοητή μορφή και φυσική γλώσσα, σε βάρος της απλότητας και της εύκολης υλικής εγκατάστασης. Πάντως μπορεί η προσανατολισμένη στις Υπηρεσίες προσέγγιση να δημιουργεί πολυπλοκότητα, αλλά το συγκεκριμένο είναι το μόνο ίσως πρόβλημα. Επίσης, αξίζει να αναφερθεί ότι η ύπαρξη ενός κοινού Ταμειυτήρα επιχειρηματικών κανόνων που χρησιμοποιείται κατά τη συγκεκριμένη αρχιτεκτονική προσέγγιση επιτρέπει την επαναχρησιμοποίηση κανόνων και διαδικασιών. Τέλος, και μόνο το γεγονός της προσαρμογής της επιχειρηματικής λογικής στα πλαίσια της επιχειρησιακής πολιτικής καθιστά τη συγκεκριμένη προσέγγιση πολλά υποσχόμενη.

Γίνεται λοιπόν, φανερό ότι κάθε προσέγγιση μπορεί να αποδειχθεί χρήσιμη ανάλογα με το είδος της επιχειρησιακής εφαρμογής που πρόκειται να χρησιμοποιηθεί. Πάντως, είναι δεδομένο ότι για τις σύγχρονες επιχειρηματικές διαδικασίες που χαρακτηρίζονται από ταχύ ρυθμό μεταβολών και ανάγκη για άμεση προσαρμοστικότητα χωρίς να απαιτείται συνεχής δέσμευση νέων υπολογιστικών πόρων, η προσανατολισμένη στις Υπηρεσίες προσέγγιση αποδεικνύεται η ιδανικότερη λύση. Οδηγεί σε επαναχρησιμοποίηση των πόρων, ευελιξία και προσαρμοστικότητα που αποτελούν τα σύγχρονα επιχειρηματικά ζητούμενα.

Προσέγγιση	Ανάπτυξη		Επαναχρησιμοποίηση		Ευελιξία	
	Διαχωρισμός μεταξύ επιχειρηματικής και Διαδικαστικής Λογικής	Ευκολία οργάνωσης και διαχείρισης της σειράς εκτέλεσης	Επαναχρησιμοποίηση Κανόνων εντός του πλαισίου της διαδικασίας	Επαναχρησιμοποίηση Κανόνων μεταξύ διαφορετικών εφαρμογών	Ικανότητα υποστήριξης υπάρχουσας επιχειρηματικής πολιτικής	Ευκολία άμεσης μεταβολής κανόνων (κατά τη διάρκεια εκτέλεσης)
Hardcoded	☺	☺ ☺ ☺ ☺	Μη εφικτό	Μη εφικτό	Μη εφικτό	Μη εφικτό
BPM-based	☺ ☺ ☺	☺ ☺ ☺ ☺	☺ ☺ ☺ ☺	☺	☺	☺ ☺ ☺
Service-Oriented Rule Based	☺ ☺ ☺ ☺	☺ ☺ ☺	☺ ☺ ☺ ☺	☺ ☺ ☺ ☺	☺ ☺ ☺ ☺	☺ ☺ ☺ ☺

Πίνακας 9-1: Σύγκριση προσεγγίσεων επιχειρηματικών κανόνων

7 Βιβλιογραφία

- [1] A.W. Scheer. "Architecture for Integrated Information Systems, Foundations of Enterprise Modeling", Berlin, Germany, 1992.
- [2] Allen, Paul. "Service Orientation: Winning Strategies and Best Practices", Cambridge University Press, Cambridge, UK, 2006.
- [3] B. Curtis, M.I. Kellner, J. Over. "Process Modeling", Communications of the ACM, Vol. 35, No. 9, pp. 75-90, 1992.
- [4] B. Orriens, J. Yang, and M. P. Papazoglou. "A Framework for Business Rule Driven Service Composition", In *Proceedings of the Fourth International Workshop on Conceptual Modeling Approaches for e-Business Dealing with Business Volatility*, 2003.
- [5] B. von Halle, "Business rules applied: Building Better Systems Using the Business Rule Approach", Wiley & Sons, 2002.
- [6] Bloomberg, Jason and Ronald Schmelzer, "SOA Orient or Be Doomed!: How Service Orientation Will Change Your Business", Hoboken, NJ: John Wiley and Sons, 2006.
- [7] "Building bug-free O-O software: An introduction to Design by Contract" available online at: <http://archive.eiffel.com/doc/manuals/technology/contract/>.
- [8] Business Activity Monitoring (BAM) available at <http://blogs.bpel-people.com/>
- [9] Business Process Modelling Language (BPML) available at <http://www.bpml.org/BPML.html>.
- [10] Brahim Medjahed, Boualem Benatallah, Athman Bouguettaya, Anne H. H. Ngu, Ahmed K. Elmagarmid, "Business-to-business interactions: issues and enabling technologies", The VLDB Journal (2003) 12: 59–85.
- [11] Casati, F. et al., "Business-oriented management of Web services", Commun. ACM 46(10), 55–60 (2003).
- [12] Cearly, D., Fenn, J., Plummer, D., "Gartner's Positions on the Five Hottest IT Topics and Trends in 2005", 2005.
- [13] Charfi, A., Mezini, M.: "Hybrid Web service composition: business processes meet business rules." In: ICSOC '04: Proceedings of the 2nd international conference on Service oriented computing, pp. 30–38. ACM Press, New York (2004)
- [14] Chappel, David A., "Enterprise Service Bus", Sebastopol, CA: O'Reilly Media, 2006.
- [15] D. A. Manolescu. "Orchestration Patterns in Service Oriented Architectures." URL: <http://www.orchestrationpatterns.com/OrchestrationPatterns.html>, January 2005.

- [16] Dirk Krafzig, Karl Banke, Dirk Slama, “Enterprise SOA. Service Oriented Architecture Best Practices.” Prentice Hall.
- [17] GlassFish Application Server documentation and characteristics available at <http://www.sun.com/software/products/appsrvr/>
- [18] GlassFishESB documentation available at <https://open-esb.dev.java.net/>
- [19] G. Wagner. “How to design a general rule markup language?”, In *Workshop XML Technologien fuer das Semantic Web (XSW)*, Berlin, June 2002.
- [20] H.J. Harrington. “Business Process Improvement: The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness”, McGraw-Hill, New York, NY, USA, 1991.
- [21] http://en.wikipedia.org/wiki/Information_Systems
- [22] http://en.wikipedia.org/wiki/Loose_coupling
- [23] Java EE tutorial, documentation and architecture available at java.sun.com/javaee/5/docs/tutorial/doc/JavaEETutorial.pdf
- [24] J.A. Zachman:”A framework for information systems architecture”, IBM Systems Journal, Vol. 26, no. 3, pp. 276-292, 1987
- [25] JBoss Application Server documentation and Architecture available at <http://www.jboss.org/jbossas/docs/>
- [26] JSR 94: JavaTM Rule Engine API, available at <http://jcp.org/en/jsr/detail?id=94>
- [27] JSR 208: Java Business Integration available at <http://jcp.org/en/jsr/detail?id=208>
- [28] Kenneth C. Laudon, Jane P. Laudon, «Πληροφοριακά Συστήματα Διοίκησης».
- [29] M. Koubarakis, D. Plexousakis:”A Formal Framework for Business Process Modeling and Design”, Information Systems, Vol. 27, pp. 299-319, 2002
- [30] Mike P. Papazoglou, Willem-Jan van den Heuvel, “Service oriented architectures: approaches, technologies and research issues”, The VLDB Journal (2007) 16:389–415.
- [31] M. P. Papazoglou. Service-oriented computing: concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering*, pages 3–12, December 2003.
- [32] NetBeans 6.1 tutorial and samples available at <http://www.netbeans.org/>
- [33] OASIS, Organization for the Advancement of Structured Information Standards <http://www.oasis-open.org>

- [34] Rosenberg, F. and S. Dustdar, "Business rules integration in BPEL: A service-oriented approach", Proceedings of the 7th International Conference on E-Commerce Technology, IEEE, 2005.
- [35] Rosenberg, F. and S. Dustdar, "Towards a distributed service-oriented business rule system", Proceedings of the Third European Conference on Web Services (ECOWS'05), IEEE, 2005.
- [36] T.H. Davenport, J.E. Short."The new industrial engineering: Information technology and business process redesign", Sloan Management Review , Vol. 31, No. 4, pp. 11-27, 1990
- [37] The Business Rules Group. "Defining Business Rules – What Are They Really?" available at [http://www.businessrulesgroup.org/first paper/br01c0.htm](http://www.businessrulesgroup.org/first%20paper/br01c0.htm), July 2000.
- [38] Thomas Erl , "Service Oriented Architecture, Concepts, Technology and Design", Prentice Hall.
- [39] U. Dayal, M. Hsu, R. Ladin."Business Process Coordination: State of the Art, Trends, and Open Issues", Proceedings of the 27th VLDB Conference, Rome, Italy, 2001.
- [40] Universal Description, Discovery, and Integration (UDDI) available at: <http://www.uddi.org>.
- [41] Web Services Atomic Transaction (WS-AtomicTransaction) available at: <http://msdn.microsoft.com/library/en-us/dnglobspec/html/WSAtomicTransaction.pdf>
- [42] Web Services Coordination (WS-Coordination) available at: <http://msdn.microsoft.com/library/en-us/dnglobspec/html/WS-Coordination.pdf>
- [43] Web Services Eventing (WS-Eventing) available at: <http://www-128.ibm.com/developerworks/webservices/library/specification/wseventing/>
- [44] WS Choreography Description Language available at: <http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/>
- [45] WS Choreography Model available at: <http://www.w3.org/TR/2004/WD-ws-chor-model-20040324/>
- [46] WS-Security (Web Services Security): http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
- [47] Yefim V. Natis, Roy W. Schulte, "Advanced SOA for Advanced Enterprise Projects", Sep 2006.