



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

**Προς μία Εκτεταμένη Αρχιτεκτονική  
Προσανατολισμένη σε Υπηρεσίες για την Διενέργεια  
Αυτοματοποιημένων Συναλλαγών**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ρουμπίνη Μάρθα Α. Μαρκοπούλου

**ΕΠΙΒΛΕΠΩΝ:** Ιωάννης Ψαρράς  
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2008





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

**Προς μία Εκτεταμένη Αρχιτεκτονική  
Προσανατολισμένη σε Υπηρεσίες για την Διενέργεια  
Αυτοματοποιημένων Συναλλαγών**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ρουμπίνη Μάρθα Α. Μαρκοπούλου

**ΕΠΙΒΛΕΠΩΝ:** Ιωάννης Ψαρράς  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18<sup>η</sup> Ιουλίου 2008.

.....  
Ιωάννης Ψαρράς  
Καθηγητής Ε.Μ.Π.

.....  
Δημήτριος Ασκούνης  
Επικουρος Καθηγητής Ε.Μ.Π.

.....  
Γρηγόριος Μέντζας  
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2008

.....  
Ρουμπίνη Μάρθα Α. Μαρκοπούλου

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

Copyright © Ρουμπίνη Μάρθα Α. Μαρκοπούλου, 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.

## Πρόλογος

Κατά τη διάρκεια των τελευταίων χρόνων διαμορφώνεται ένα ιδιαίτερα πολυσύνθετο πλαίσιο επιχειρηματικών συναλλαγών στο οποίο εντάσσονται και λειτουργούν επιχειρήσεις και οργανισμοί με διαφορετικά συστήματα τεχνολογίας, πολιτικές και επιχειρηματικά μοντέλα. Στα πλαίσια αυτού του ετερογενούς περιβάλλοντος επιχειρηματικών δραστηριοτήτων εμφανίζεται για τις επιχειρήσεις η ανάγκη ανάπτυξης ενός συνόλου ενεργειών και ιδιοτήτων που θα τις καταστήσουν ανταγωνιστικές και θα τις βοηθήσουν να προσφέρουν μια ποικιλία καινοτόμων υπηρεσιών αποτελεσματικά, γρήγορα και σε ελεγχόμενα κόστη. Ορισμένες από τις ιδιότητες αυτές είναι η προσαρμοστικότητα στις μεταβολές των επιχειρηματικών μοντέλων, η έγκαιρη αντίδραση σε αυτές, η επαναχρησιμοποίηση των μονάδων λειτουργικότητας και η αποδέσμευση της επιχειρηματικής λογικής από την τεχνική υποδομή. Επίσης βασική επιδίωξη είναι η διαλειτουργικότητα των ετερογενών συστημάτων και προτύπων που εφαρμόζουν οι επιχειρήσεις. Η διαλειτουργικότητα επιτρέπει την ανάπτυξη ενός κοινού κώδικα επικοινωνίας μεταξύ των συνεργατών και ενοποιεί διαφορετικής φύσης δεδομένα και διαδικασίες.

Τα Πληροφορικά Συστήματα που συγκεντρώνουν το σύνολο του λογισμικού, του υλικού και των δεδομένων ή των κανόνων τα οποία έχει στην ιδιοκτησία της μια επιχείρηση, μπορούν να οργανωθούν με τέτοιο τρόπο ώστε να επιτευχθούν τα καλύτερα αποτελέσματα ως προς τις διαστάσεις που αναφέρθηκαν προηγουμένως. Προς την κατεύθυνση ανανέωσης του πλαισίου των επιχειρηματικών συναλλαγών στην παρούσα διπλωματική παρουσιάζονται και εξετάζονται περιπτώσεις αρχιτεκτονικών και εφαρμογών τους και τέλος δίνεται μια ερευνητική πρόταση εκτεταμένης Αρχιτεκτονικής Προσανατολισμένης σε Υπηρεσίες (Service Oriented Architecture-SOA).



## Περίληψη

Η παρούσα διπλωματική κινείται στα πλαίσια της εφαρμογής των Πληροφοριακών Συστημάτων στις επιχειρήσεις και ειδικότερα στην οργάνωση και διεκπεραίωση των ηλεκτρονικών συναλλαγών στο περιβάλλον του ηλεκτρονικού επιχειρείν. Ανώτερος στόχος της είναι η προδιαγραφή μιας αρχιτεκτονικής ηλεκτρονικών συναλλαγών η οποία αφενός βασίζεται σε εφαρμοζόμενα αρχιτεκτονικά πρότυπα αφετέρου συνδυάζει τα βασικότερα χαρακτηριστικά των προτύπων αυτών προκειμένου να αποδώσει μια αποδοτική λύση σε ζητήματα διαλειτουργικότητας των Πληροφοριακών Συστημάτων.

Στα πλαίσια της διπλωματικής σκιαγραφείται η εικόνα που παρουσιάζουν σήμερα οι ηλεκτρονικές επιχειρηματικές συναλλαγές ενώ μεγάλη έμφαση δίνεται στο πρόβλημα της διαλειτουργικότητας των Πληροφοριακών Συστημάτων όπου και ορίζονται οι άξονες που προσδιορίζουν το εν λόγω πρόβλημα. Γίνεται αναφορά στις υπάρχουσες τεχνολογίες που υποστηρίζουν τις ηλεκτρονικές συναλλαγές και εξετάζονται λεπτομερώς οι δυο κύριες αρχιτεκτονικές (Service Oriented και Model Driven Architecture) που χρησιμοποιούνται από τις επιχειρήσεις σε αυτό τον τομέα. Κατόπιν ορίζονται και τεκμηριώνονται τα βασικά χαρακτηριστικά της προδιαγραφόμενης αρχιτεκτονικής ενώ επισημαίνεται ιδιαίτερα ο υβριδικός της χαρακτήρας. Αναλυτικότερα, συγκεντρώνονται και αξιολογούνται τα διάφορα εφαρμοζόμενα πρότυπα, γίνεται η αντιστοίχισή τους προς τα θέματα που πρέπει να επιλυθούν και επιλέγεται να εφαρμοστεί η λύση που κρίνεται ότι αποδίδει την βέλτιστη λύση.

Η προδιαγραφή της εκτεταμένης αρχιτεκτονικής αποσκοπεί στο να προσφέρει μια εύκολα εφαρμόσιμη, αποτελεσματική και μειωμένου κόστους λύση η οποία στηρίζεται σε ανοιχτά πρότυπα. Ειδικότερα, επιδίωξη αποτελεί η δυνατότητα υιοθέτησής της από τον κλάδο των μικρομεσαίων επιχειρήσεων για την απόκτηση ευκαιριών στο ανταγωνιστικό επιχειρηματικό περιβάλλον χωρίς την ανάγκη διάθεσης πόρων - χρηματικών κεφαλαίων ή ανθρωποωρών εκπαίδευσης - σε απαγορευτικό βαθμό.

### Λέξεις κλειδιά

Διαλειτουργικότητα, υπηρεσίες, επιχειρηματικές διαδικασίες, δεδομένα, ενοποίηση, επιχειρηματική αρχιτεκτονική, συναλλαγές, μοντέλα, εξυπηρετητής, πελάτης, SOA, MDA.





## **Abstract**

The present dissertation is based on the application of Information Systems within the enterprises. In particular, it focuses on the organization and implementation of electronic transactions within the environment of the electronic commerce. The main target is the justification of a new architecture for electronic transactions which follows applicable specifications but also uses a combination of their attributes for resolving the interoperability issues among heterogeneous cooperating Information Systems.

First of all, this dissertation represents the current situation of electronic business transactions and emphasizes on the dimensions of the interoperability problem that emerges between Information Systems. Here are presented the most widely used technologies that facilitate the electronic transactions and are examined in more details the two main architectures (Service Oriented and Model Driven Architecture) which prevalent today. Moreover, the principal attributes of the new architecture are defined and justified while great attention is given to its hybrid nature. After taking into consideration and evaluating all the known standards, these results are matched to the problems that remain to be resolved and finally the best solution is chosen.

The specification of the extended service architecture aims to offer an easily applicable and efficient solution of reduced cost which is based on open standards. Its importance lies in the fact that it could be used by small and middle sized firms. Under these circumstances especially these enterprises could gain advantage within the highly competitive business environment without being obliged to spend a high volume of resources- money assets or labor- on this purpose.

## **Keywords**

Interoperability, services, business processes, data, integration, business architecture, transactions, models, server, client, SOA, MDA.



## Ευχαριστίες

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 Ιωάννη Ψαρρά και τον υποψήφιο διδάκτορα Γιώργο Γκιώνη, οι οποίοι καταρχάς μου έδωσαν τη ευκαιρία να ασχοληθώ εκτενέστερα με το συγκεκριμένο ενδιαφέρον ζήτημα όπως επίσης με υποστήριξαν και με καθοδήγησαν με συνέπεια και υπευθυνότητα καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας.

Ιούλιος 2008

Ρουμπίνη Μάρθα Α. Μαρκοπούλου



## Περιεχόμενα

Πρόλογος .....	5
Περίληψη .....	7
Abstract .....	9
Ευχαριστίες .....	11
Περιεχόμενα .....	13
Πίνακας Ξενόγλωσσων Όρων .....	15
Πίνακας Εικόνων .....	17
Πίνακες .....	17
1. Εισαγωγή .....	19
1.1 Διατύπωση του προβλήματος .....	19
1.2 Διάρθρωση κειμένου .....	21
2. Πληροφοριακά Συστήματα στις Επιχειρήσεις .....	24
3. Ηλεκτρονικές Συναλλαγές και Ηλεκτρονικές Υπηρεσίες .....	28
3.1 Ορισμός και ανάλυση του πλαισίου των ηλεκτρονικών συναλλαγών .....	28
3.2 Οργάνωση ηλεκτρονικών συναλλαγών .....	31
3.2.1 Χαλαρή και Ισχυρή Σύζευξη .....	31
3.2.2 Ενορχήστρωση- Χορογραφία .....	34
3.2.3 Διαχείριση Επιχειρηματικών Διαδικασιών (Business process Management) .....	37
3.3 Διαλειτουργικότητα Πληροφοριακών Συστημάτων .....	38
3.3.1 Ορισμός διαλειτουργικότητας Πληροφοριακών Συστημάτων .....	38
3.3.2 Ενοποίηση των δεδομένων .....	42
3.3.3 Ενοποίηση των διαδικασιών .....	45
3.3.4 Ενοποίηση των υπηρεσιών .....	47
4. State of the Art Analysis .....	50
4.1 Αρχιτεκτονικές Φυσικής Όψης .....	50
4.1.1 Αρχιτεκτονική Πελάτη-Εξυπηρετητή .....	50
4.1.2 Κατανεμημένη Διαδικτυακή Αρχιτεκτονική .....	52
4.2 Generic Αρχιτεκτονικές .....	55
4.2.1 Αρχιτεκτονική Οδηγούμενη από τα Μοντέλα .....	55
4.2.2 Αρχιτεκτονική Προσανατολισμένη σε Υπηρεσίες .....	60
4.2.3 Αρχιτεκτονική Ψηφίδων .....	66

4.3 Εφαρμογές διαλειτουργικότητας .....	70
4.3.1 Μεσίτες Ενοποίησης Επιχειρηματικών Εφαρμογών (EAI Brokers) .....	70
4.3.2 Εξυπηρετητές Εφαρμογών (Application Servers) .....	71
4.3.3 Προσαρμογείς (Adapters) .....	72
4.3.4 Μεσολογισμικό Προσανατολισμένο στα Μηνύματα (Message Oriented Middleware) .....	73
4.3.5 Δίαυλος Επιχειρηματικών Υπηρεσιών (Enterprise Service Bus) .....	75
4.4 Αρχιτεκτονικές και εφαρμογές σε ερευνητικό επίπεδο .....	79
4.4.1 Ecosystem Oriented Architecture (EOA) .....	79
4.4.2 Genesis Hybrid Architecture .....	83
5. Ανάλυση σε επίπεδο αρχιτεκτονικής .....	86
5.1 Διαστάσεις αρχιτεκτονικών συστημάτων ηλεκτρονικών συναλλαγών .....	86
5.1.1 Επιχειρηματική Αρχιτεκτονική (Business Architecture) .....	87
5.1.2 Συναλλαγές (Transactions) .....	88
5.1.3 Ενορχήστρωση-χορογραφία των διαδικασιών (Process Orchestration – Choreography) .....	90
5.1.4 Ενοποίηση (Integration) .....	91
5.1.5 Ανταλλαγή των μηνυμάτων (Messaging) .....	91
5.2 Αξιολόγηση των SOA και MDA σε κάθε Διάσταση .....	94
5.3 Σύγκριση SOA και MDA και τελικά συμπεράσματα .....	110
6. Επεκτείνοντας την Αρχιτεκτονική που Προσανατολίζεται σε Υπηρεσίες .....	116
6.1 Στόχοι Εκτεταμένης Αρχιτεκτονικής-Κύριοι άξονες .....	116
6.2 Εκτεταμένη Αρχιτεκτονική Προσανατολισμένη σε Υπηρεσίες .....	118
6.2.1 Επίπεδο βασικών λειτουργιών xSOA .....	118
6.2.2 Επίπεδο σύνθεσης xSOA .....	119
6.2.3 Επίπεδο διαχείρισης xSOA .....	121
6.2.4 Επισκόπηση της Εκτεταμένης Αρχιτεκτονικής .....	124
6.3 Η δυναμική για απόδοση της επένδυσης σε αρχιτεκτονικές SOA, MDA & xSOA .....	127
7. Επίλογος .....	130
7.1 Σύνοψη και Συμπεράσματα .....	130
7.2 Πεδία έρευνας στον τομέα των ηλεκτρονικών συναλλαγών .....	131
8. Βιβλιογραφία .....	134

## Πίνακας Ξενόγλωσσων Όρων

Αρκτικόλεξο	Αγγλικοί Όροι	Ελληνικοί Όροι
ADL	Architecture Description Language	Γλώσσα Περιγραφής Αρχιτεκτονικών
API	Application Programming Interface	Προγραμματιστική Διεπαφή Εφαρμογής
BPEL	Business Process Execution Language	Γλώσσα Εκτέλεσης Επιχειρηματικών Διαδικασιών
BPEL4WS	Business Process Execution Language for Web Services	Γλώσσα Εκτέλεσης Επιχειρηματικών Διαδικασιών για Διαδικτυακές Υπηρεσίες
BPM	Business Process Management	Διαχείριση Επιχειρηματικών Διαδικασιών
B2B	Business To Business	Επιχείρηση προς Επιχείρηση
B2C	Business To Costumer	Επιχείρηση προς Πελάτη
B2G	Business To Government	Επιχείρηση προς Κυβέρνηση
CIM	Computational Independent Model	Μοντέλο Ανεξάρτητο Υπολογιστικής πλατφόρμας
CORBA	Common Object Request Broker Architecture	Αρχιτεκτονική Μεσίτη Κοινών Αιτήσεων Αντικειμένων
DCOM	Distributed Component Object Model	Μοντέλο Κατανεμημένου Αντικειμένου Ψηφίδας
DOM	Document Object Model	Μοντέλο Αντικειμένου Εγγράφου
DTD	Document Type Definitions	Ορισμοί Τύπων Εγγράφων
EAI	Enterprise Application Integration	Ενοποίηση Επιχειρηματικών Εφαρμογών
EBCDIC	Extended Binary Coded Decimal Interchange Code	Εκτεταμένος Δυαδικά Κωδικοποιημένος Δεκαδικός Κώδικας Ανταλλαγής
eXML	Commerce XML	Εκτεταμένη Γλώσσα Σημείωσης για το Εμπόριο
ebXML	Electronic Business using eXtensible Markup Language	Ηλεκτρονικό Εμπόριο χρησιμοποιώντας την Εκτεταμένη Γλώσσα Σημείωσης
EDI	Electronic Data Interchange	Ηλεκτρονική Ανταλλαγή Δεδομένων
EOA	Enterprise Oriented Architecture	Αρχιτεκτονική Προσανατολισμένη σε Υπηρεσίες
ERP	Enterprise Resource Planning	Προγραμματισμός Επιχειρηματικών Πόρων
ESB	Enterprise Service Bus	Διάυλος Επιχειρηματικών Υπηρεσιών
FTP	File Transfer Protocol	Πρωτόκολλο Μεταφοράς Αρχείων
HTML	HyperText Markup Language	Γλώσσα Σημείωσης Υπερκειμένου
HTTP	Hypertext Transfer Protocol	Πρωτόκολλο Μεταφοράς Υπερκειμένου
ICT	Information Communication Technology	Τεχνολογία Πληροφορικής και Επικοινωνιών
IDL	Interface Description Language	Γλώσσα Περιγραφής Διεπαφής
MDA	Model Driven Architecture	Αρχιτεκτονική Οδηγούμενη από τα Μοντέλα
MOM	Message Oriented Middleware	Μεσολογισμικό Προσανατολισμένο στα Μηνύματα

OMG	Object Management Group	Ομάδα Διαχείρισης Αντικειμένων
OWL	Web Ontology Language	Γλώσσα Οντολογιών Ιστού
PIM	Platform Independent Model	Μοντέλο Ανεξάρτητο από την Πλατφόρμα
PSM	Platform Specific Model	Μοντέλο Συγκεκριμένης Πλατφόρμας
RDF	Resource Description Framework	Πρότυπο Περιγραφής Πόρων
REST	REpresentational State Transfer	Επίπεδο Αναπαράστασης Μεταφοράς
RPC	Remote Procedure Calls	Απομακρυσμένη Διαδικασία Κλήσεων
SAX	Simple API for XML	Απλή Προγραμματιστική Διεπαφή Εφαρμογής για Εκτεταμένη Γλώσσα Σημείωσης
SOA	Service Oriented Architecture	Αρχιτεκτονική Προσανατολισμένη σε Υπηρεσίες
SOAP	Simple Object Access Protocol	Απλό Πρωτόκολλο Προσπέλασης Αντικειμένου
UML	Unified Modeling Language	Ενιαία Γλώσσα Μοντελοποίησης
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business	Κέντρο των Ηνωμένων Εθνών για την Διευκόλυνση του Εμπορίου και του Ηλεκτρονικού Επιχειρείν
WS-BPEL	Web Service Business Process Execution Language	Γλώσσα Εκτέλεσης Επιχειρηματικών Διαδικασιών Διαδικτυακών Υπηρεσιών
WS-CDL	Web Service Choreography Description Language	Γλώσσα Περιγραφής Χορογραφίας Διαδικτυακών Υπηρεσιών
WSCI	Web Service Choreography	Χορογραφία Διαδικτυακών Υπηρεσιών
WSCL	Web Services Conversation Language	Γλώσσα Συνομιλίας Διαδικτυακών Υπηρεσιών
WSDL	Web Services Description Language	Γλώσσα Περιγραφής Διαδικτυακών Υπηρεσιών
WSFL	Web Services Flow Language	Γλώσσα Ροών Εργασίας Διαδικτυακών Υπηρεσιών
Xcbl	Common Business Library	Κοινή Βιβλιοθήκη Επιχειρήσεων
XLANG	XML-based extension of Web Services Description Language	Επεκτάσεις των Διαδικτυακών Υπηρεσιών που βασίζονται σε Εκτεταμένη Γλώσσα Σημείωσης
XML	Extensible Markup Language	Εκτεταμένη Γλώσσα Σημείωσης
XPATH	XML Path Language	Γλώσσες Μονοπατιών Εκτεταμένης Γλώσσας Σημείωσης
XSD	XML Schema Definitions	Περιγραφές Σχήματος Εκτεταμένης Γλώσσας Σημείωσης
XSLT	Extensible Stylesheet Language Transformations	Επεκτάσιμη Γλώσσα Μετατροπών Μορφοποίησης
xSOA	Extended Service Oriented Architecture	Εκτεταμένη Αρχιτεκτονική Προσανατολισμένη σε Υπηρεσίες



## **Πίνακας Εικόνων**

Εικόνα 1-Επιχειρησιακά συστήματα .....	27
Εικόνα 2-Σενάριο ολοκληρωμένης επιχειρηματικής εφαρμογής .....	30
Εικόνα 3-Επίπεδα διαλειτουργικότητας .....	40
Εικόνα 4- Σενάριο ενοποίησης δεδομένων .....	43
Εικόνα 5- Τυπική αρχιτεκτονική δυο επιπέδων πελάτη-εξυπηρετητή .....	51
Εικόνα 6- Τυπική κατανεμημένη αρχιτεκτονική διαδικτύου .....	53
Εικόνα 7- Αρχιτεκτονική που οδηγείται από τα μοντέλα (Model Driven architecture) .....	56
Εικόνα 8- Αντιστοιχίσεις ανάμεσα στα μοντέλα PIM και PSM.....	57
Εικόνα 9-Μετατροπή από μοντέλο UML σε συγκεκριμένα μοντέλα και πλατφόρμες .....	58
Εικόνα 10-MDA vs Συμβατικό σενάριο.....	59
Εικόνα 11- Επίπεδα αρχιτεκτονικής προσανατολιζόμενης σε υπηρεσίες (SOA).....	62
Εικόνα 12-Service-Oriented Modeling Framework (SOMF) Version 1.1 .....	64
Εικόνα 13- SOA Meta Model, The Linthicum Group, 2007 .....	65
Εικόνα 14-Αρχιτεκτονική ενοποιημένου συστήματος .....	67
Εικόνα 15-MOM μήνυμα .....	74
Εικόνα 16- Ενοποίηση βασισμένη σε πρότυπα .....	76
Εικόνα 17-Enterprise Service Bus .....	79
Εικόνα 18- στοίβα υπηρεσιών στο Digital Business Ecosystem.....	82
Εικόνα 19- Αρχιτεκτονική του Genesis.....	84
Εικόνα 20-Πέντε διαστάσεις αρχιτεκτονικής .....	94
Εικόνα 21-The Web Services Development Life Cycle hierarchy[37] .....	107
Εικόνα 22-Χαρακτηριστικά & Υλοποίηση xSOA ανά επίπεδο .....	126

## **Πίνακες**

Πίνακα 1- Σύγκριση Διαστάσεων σε SOA και MDA .....	113
--	-----



# **1 ■ Εισαγωγή**

## **1.1 Διατύπωση του προβλήματος**

Οι τεχνολογικές εξελίξεις των τελευταίων δεκαετιών στον τομέα της Τεχνολογίας Πληροφορικής και των Επικοινωνιών αποτέλεσαν την αφετηρία των διαδικασιών εκσυγχρονισμού των επιχειρηματικών εφαρμογών και σχέσεων. Ιδιαίτερα κάτω από την εκρηκτική εξάπλωση του Διαδικτύου και την εμφάνιση του φαινομένου της παγκόσμιας αγοράς και του εμπορίου οι επιχειρήσεις προέβησαν σε κινήσεις 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 υψηλά επίπεδα ανταγωνιστικότητας, να διευκολύνουν τις επιχειρηματικές τους διαδικασίες και να μειώσουν το λειτουργικό κόστος τους.

Η δημιουργία νέων υπηρεσιών, η προσφορά καινοτόμων προϊόντων, η ανάπτυξη μιας άμεσης και προσοδοφόρας σχέσης με το επιχειρηματικό περιβάλλον καθώς και οι αλλαγές στα επιχειρηματικά μοντέλα αποτελούν ορισμένα μόνο από τα αποτελέσματα που προέκυψαν από την αξιοποίηση των πόρων πληροφορίας και τεχνολογίας. Επιπλέον οι υποδομές Τεχνολογιών Πληροφορικής και Επικοινωνιών (ICT) των επιχειρήσεων άνοιξαν τους ορίζοντες των επιχειρήσεων για επέκταση της σφαίρας δράσης τους σε παγκόσμιο επίπεδο και τη σύναψη συνεργασιών ή την ισχυροποίηση των ήδη υπαρχουσών.

Σήμερα, παρουσιάζονται δυο βασικές κατηγορίες επιχειρήσεων. Αυτές που έχουν επενδύσει μεγάλα ποσά στην αναβάθμιση των συστημάτων τους και στην αντίθετη πλευρά αυτές που δεν είχαν την οικονομική ευρωστία για μεγάλες επενδύσεις και κρατούν κάποια μικρής κλίμακας και δυνατοτήτων συστήματα. Και οι δυο κατηγορίες επιχειρήσεων έχουν ωστόσο ανάγκη να επαναχρησιμοποιήσουν τα συστήματά τους, να συνδεθούν με αυτά των συνεταίρων τους και να αναπτύξουν νέες λειτουργικότητες προκειμένου να ελαττώσουν τα κόστη τους και να εκσυγχρονιστούν. Επιπλέον θεωρείται ότι κάθε επιχείρηση πρέπει να ακολουθήσει

βιώσιμες λύσεις που θα της επιτρέψουν να προσαρμοστεί γρήγορα και αποτελεσματικά στις συνεχώς μεταβαλλόμενες επιχειρηματικές απαιτήσεις.

Ο τομέας της έρευνας σε συνεργασία με τις επιχειρήσεις καλείται να αναπτύξει λύσεις στα πλαίσια της οργάνωσης και της λειτουργίας των συναλλαγών. Ειδικότερα οι συναλλαγές θεωρούνται ως η κυριότερη δραστηριότητα των επιχειρήσεων εφόσον μέσω αυτών γίνεται αντιληπτή η λειτουργία των τελευταίων. Έτσι έχουν αναπτυχθεί και συνεχίζουν να αναπτύσσονται αρχιτεκτονικές και τεχνολογίες που μπορούν να χρησιμοποιηθούν εύκολα και με μειωμένο κόστος από τα ήδη υπάρχοντα συστήματα. Βασική επιδίωξη είναι η δημιουργία μιας ευέλικτης και εύκολα προσαρμοζόμενης στα εμπορικά δεδομένα επιχείρησης η οποία παράλληλα θα είναι ικανή να διαλειτουργήσει με τους συνεργάτες της σε επίπεδο συστημάτων. Υπό αυτές τις συνθήκες θα είναι ικανή να επιτύχει μεγαλύτερα κέρδη, καλύτερες υπηρεσίες και γρήγορη απόκριση στις απαιτήσεις των πελατών.

Στα ερευνητικά αυτά πλαίσια κινείται και η παρούσα διπλωματική όπου αρχικά εξετάζονται οι ευρέως χρησιμοποιούμενες λύσεις. Οι λύσεις αυτές θα αποτελέσουν τη βάση των ερευνητικών προτάσεων εφόσον η παγκόσμια χρήση τους τις έχει καθιερώσει ως de facto πρότυπα και δύσκολα μπορούν να αντικατασταθούν. Η δυσκολία των εγχειρήματος έγκειται στο ότι υπάρχουν ορισμένα τεχνολογικά και επιχειρηματικά σταθερά πλαίσια που έχουν πλέον εδραιωθεί και τα οποία οι επιχειρήσεις έχουν ανάγκη. Στην αντίθετη πλευρά βρίσκεται η δυναμική φύση των επιχειρηματικών μοντέλων που απαιτεί πολλές και άμεσες αλλαγές. Έτσι παρόλο που αναζητείται μια εκσυγχρονιστική λύση, αυτή πρέπει να λαμβάνει υπόψη τα καθιερωμένα πρότυπα, να στηριχθεί σε αυτά και να τα αναδιατάξει με μικρές προσθήκες, δίνοντας τις επιθυμητές λύσεις.

Στη συνέχεια δίνονται οι βασικότεροι άξονες του προβλήματος που ορίζουν και τη μεθοδολογία που πρόκειται να ακολουθηθεί:

- Αρχικά πρέπει να προδιαγραφούν τα πλαίσια του προβλήματος και οι βασικότερες έννοιες των επιχειρηματικών ηλεκτρονικών συναλλαγών.
- Στη συνέχεια χρειάζεται να δοθεί μια γενική επισκόπηση των προτύπων που χρησιμοποιούνται σήμερα για την πραγματοποίηση των ηλεκτρονικών συναλλαγών και να προσδιοριστούν οι εφαρμοζόμενες αρχιτεκτονικές.
- Επιπλέον απαραίτητο θα ήταν να αξιολογηθούν τα εφαρμοζόμενα

πρότυπα ως προς το βαθμό που βοηθούν στην διευκόλυνση των συναλλαγών και ειδικά αν συνεισφέρουν στη διαλειτουργικότητα των Πληροφορικών Συστημάτων.

- Τέλος, βάσει της ανάλυσης των απαιτήσεων του προβλήματος, των χαρακτηριστικών που προσφέρουν οι διάφορες αρχιτεκτονικές και πρότυπα και των στόχων που επιδιώκονται, προκύπτει μια γενική κρίση. Σε αυτό το σημείο δίνονται προτάσεις επέκτασης της Αρχιτεκτονικής Προσανατολισμένης σε Υπηρεσίες (Service Oriented Architecture).

Γενικότερα, το πρόβλημα με το οποίο ασχολείται η παρούσα διπλωματική εργασία είναι πολυδιάστατο αφού συγκεντρώνει ζητήματα που άπτονται τόσο του επιχειρηματικού τομέα όσο και του τομέα εφαρμογών της Τεχνολογίας Πληροφορικής και Επικοινωνιών. Στην επόμενη ενότητα δίνεται η διάρθρωση του κειμένου που θα ακολουθηθεί η οποία είναι άμεσα εξαρτώμενη και από τη μεθοδολογία που ορίστηκε για την αντιμετώπιση του προβλήματος.

## **1.2 Διάρθρωση κειμένου**

Στη συνέχεια παρουσιάζεται συνοπτικά η διάρθρωση του κειμένου αυτής της διπλωματικής εργασίας βάσει πάντα της μεθοδολογίας που ορίστηκε στην προηγούμενη ενότητα.

Αρχικά δίνονται ο πρόλογος, οι περιλήψεις σε ελληνικά και αγγλικά, το κείμενο ευχαριστιών και ο πίνακας περιεχόμενων, εικόνων και ξενόγλωσσων όρων.

Στη συνέχεια, περνώντας στο κυρίως κείμενο της διπλωματικής, το 1<sup>ο</sup> κεφάλαιο αποτελεί την εισαγωγή. Σε αυτή διατυπώνεται το πρόβλημα και δίνεται ένα γενικό πλαίσιο που βοηθά στην κατανόηση των απαιτήσεων και των στόχων του προβλήματος. Στο ίδιο κεφάλαιο συμπεριλαμβάνεται η ενότητα διάρθρωσης του κειμένου της διπλωματικής.

Το 2<sup>ο</sup> κεφάλαιο αποτελεί μια επισκόπηση του ρόλου των Πληροφοριακών Συστημάτων στην επιχείρηση. Αναλυτικότερα δίνεται ο ορισμός των συστημάτων αυτών καθώς και κάποιες λεπτομέρειες σχετικά με τη θέση τους και τη λειτουργικότητά τους στις επιχειρήσεις. Ουσιαστικά το κεφάλαιο αυτό παρουσιάζει τις οντότητες που συμμετέχουν στις ηλεκτρονικές συναλλαγές.

Στο 3<sup>ο</sup> κεφάλαιο ορίζεται το πλαίσιο των επιχειρηματικών ηλεκτρονικών συναλλαγών και αναλύονται ορισμένες βασικές έννοιες και τεχνικές που

εφαρμόζονται κατά την οργάνωση των συναλλαγών. Πολύ σημαντικό είναι επίσης το σημείο ορισμού του προβλήματος της διαλειτουργικότητας των Πληροφοριακών Συστημάτων και των τριών ειδών ενοποίησης που επιδιώκεται να λύσει το συγκεκριμένο πρόβλημα.

Το 4<sup>ο</sup> κεφάλαιο παρουσιάζει τα αποτελέσματα της βιβλιογραφικής έρευνας που έγινε προκειμένου να συλλεχθούν πληροφορίες σχετικά με τις τεχνολογίες και τα πρότυπα που εφαρμόζονται στις σύγχρονες ηλεκτρονικές συναλλαγές. Από την έρευνα αυτή προέκυψαν και παρατίθενται πολύ χρήσιμες πληροφορίες για αρχιτεκτονικές φυσικής όψης (αρχιτεκτονική client-server και distributed internet) καθώς επίσης και αρχιτεκτονικές ανώτερου αφαιρετικού επιπέδου (Service Oriented, Model Driven, Component Based). Επίσης δεν παραλείπεται να γίνει αναφορά στις γνωστότερες εφαρμογές των αρχιτεκτονικών αυτών ώστε να δοθεί μια ολοκληρωμένη εικόνα της σημερινής κατάστασης γύρω από την οργάνωση και υλοποίηση των συναλλαγών. Στην τελευταία παράγραφο του παρόντος κεφαλαίου δίνονται ορισμένα δείγματα έρευνας που εντάσσονται σε αυτό τον τομέα.

Συνεχίζοντας στο 5<sup>ο</sup> κεφάλαιο, δίνονται τα στοιχεία τα οποία θα δώσουν προτάσεις επέκτασης της SOA αρχιτεκτονικής. Αρχικά καθορίζονται οι βασικές διαστάσεις γύρω από τις οποίες κινείται η τεκμηρίωση και η προδιαγραφή μιας αρχιτεκτονικής ηλεκτρονικών συναλλαγών. Με γνώμονα τις διαστάσεις αυτές η προσοχή μας επικεντρώνεται στις δύο βασικές αρχιτεκτονικές-Service Oriented και Model Driven- οι οποίες θεωρούνται ως οι πλέον κατάλληλες για την ανάπτυξη και διεκπεραίωση εμπορικών συναλλαγών μεταξύ στο σύγχρονο ετερογενές επιχειρηματικό περιβάλλον. Έτσι ακολουθεί μια εκτενής ανάλυση των χαρακτηριστικών των αρχιτεκτονικών αυτών και εντέλει συνάγονται συμπεράσματα σχετικά με την καταλληλότητα καθεμιάς στην αντιμετώπιση των απαιτήσεων των ηλεκτρονικών συναλλαγών.

Το 6<sup>ο</sup> κεφάλαιο καλείται να παρουσιάσει ερευνητικές προτάσεις για την επέκταση της Αρχιτεκτονικής που Προσανατολίζεται σε Υπηρεσίες. Η επέκταση αυτή έχει μελετηθεί με ιδιαίτερο ενδιαφέρον από την ερευνητική κοινότητα.

Στο κεφάλαιο αυτό εξετάζονται συνοπτικά τα θέματα που άπτονται του συγκεκριμένου ερευνητικού πεδίου και τέλος προτείνονται ορισμένες λύσεις για την υλοποίηση των επεκτεινόμενων χαρακτηριστικών της αρχιτεκτονικής. Στην τελευταία παράγραφο αυτού του κεφαλαίου παρουσιάζονται τα οφέλη από την επένδυση (Return On Investment) που προσφέρουν οι Service Oriented και Model

Δίνen αρχιτεκτονικές για τις επιχειρήσεις αλλά και η επέκταση της SOA αρχιτεκτονικής που αναλύεται εδώ. Η αναφορά στα οφέλη αυτά είναι απαραίτητη καθώς δίνει την αξία των αρχιτεκτονικών από την επιχειρηματική σκοπιά.

Το 7<sup>ο</sup> κεφάλαιο αποτελεί και τον επίλογο της παρούσας εργασίας όπου δίνονται κάποια γενικότερα σχόλια και πληροφορίες σχετικά με πρόσθετα ερευνητικά πεδία.

Τέλος στο 8<sup>ο</sup> κεφάλαιο δίνεται η βιβλιογραφία που χρησιμοποιήθηκε κατά τη συγγραφή της διπλωματικής εργασίας.

# 2. Πληροφοριακά Συστήματα στις Επιχειρήσεις

Ορίζοντας σε γενικές γραμμές ένα Πληροφορικό Σύστημα θα λέγαμε ότι αποτελεί ένα σύνολο στοιχείων τα οποία συνεργάζονται με σκοπό την συλλογή, την επεξεργασία, την αποθήκευση και τη διανομή πληροφοριών οι οποίες είναι απαραίτητες για την λήψη αποφάσεων και τον έλεγχο μέσα στα πλαίσια ενός οργανισμού. Από την επιχειρηματική σκοπιά, τα πληροφοριακά συστήματα αποτελούν επενδύσεις των οργανισμών που έχουν ως απώτερο στόχο τη δημιουργία αξίας για την επιχείρηση.

Τα μοντέρνα Πληροφοριακά Συστήματα αποτελούν βασικό συστατικό στοιχείο των επιχειρησιακών δραστηριοτήτων. Τα τελευταία χρόνια έχει αναπτυχθεί μια ιδιαίτερα ισχυρή αλληλεξάρτηση ανάμεσα στους οργανισμούς και στα Πληροφορικά Συστήματα καθώς στρατηγική, κανόνες και επιχειρηματικές διαδικασίες είναι άμεσα εξαρτώμενες από το λογισμικό, το υλικό, τις βάσεις δεδομένων και τις τηλεπικοινωνίες που ορίζουν το τεχνολογικό υπόβαθρο των Πληροφοριακών Συστημάτων. Οι στρατηγικές επέκτασης και εξέλιξης των επιχειρήσεων αλλά και τα σχέδια επίτευξης μεγάλου μεριδίου στην αγορά και μείωσης του λειτουργικού κόστους επηρεάζονται σε μεγάλο βαθμό από την κατάσταση στην οποία βρίσκονται τα πληροφοριακά της συστήματα καθώς και από το πόσο ισχυρά είναι σε σχέση με εκείνα των ανταγωνιστών. Ιδιαίτερη σημασία δίνεται επίσης στην ικανότητα των συστημάτων αυτών να αντιμετωπίζουν την αυξανόμενη πολυπλοκότητα των επιχειρηματικών διεργασιών και να τις διεκπεραιώνουν με αποτελεσματικό τρόπο όσο μεγαλώνει η ακτίνα δράσης τους.

Οι επιχειρηματικές διεργασίες οι οποίες αναφέρονται παραπάνω αναφέρονται στον τρόπο με τον οποίο οργανώνεται, συντονίζεται και εστιάζεται η εργασία με στόχο τη παραγωγή ενός πολύτιμου προϊόντος ή υπηρεσίας[27]. Αποτελούν συγκεκριμένες ροές εργασιών, υλικών, πληροφοριών και γνώσεων αλλά αντιστοιχούν και στον τρόπο που γίνεται ο συντονισμός των ροών αυτών. Οι επιχειρηματικές



διεργασίες θα μπορούσαν να χαρακτηριστούν ως μέσο απόκτησης ανταγωνιστικού πλεονεκτήματος για τον οργανισμό ενώ αντίθετα θα μπορούσαν να έχουν και αρνητικά αποτελέσματα στην εξέλιξη του οργανισμού αν δεν είναι εκσυγχρονισμένες ή αποτελεσματικές. Σε ανώτερο επίπεδο συγκέντρωσης και συντονισμού των επιχειρηματικών εργασιών αναγνωρίζονται οι επιχειρηματικές εφαρμογές οι οποίες ουσιαστικά ολοκληρώνουν τις επιχειρηματικές διεργασίες. Έτσι ο οργανισμός επωφελείται από τις ολοκληρωμένες διεργασίες αποκομίζοντας μεγαλύτερα οφέλη.

Τα συστήματα των επιχειρήσεων είναι επιφορτισμένα με την διεκπεραίωση των επιχειρηματικών διεργασιών ενώ παράλληλα δρουν σε διατομεακό αλλά και διαλειτουργικό επίπεδο. Πολλές φορές δε, επεκτείνουν τα όρια της δραστηριότητάς τους και εκτός του οργανισμού προσφέροντας υπηρεσίες ανάμεσα στην επιχείρηση και τους παράγοντες του περιβάλλοντός της (προμηθευτές, καταναλωτές, πιστωτές κλπ.). 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 ξεχωριστό τομέα. Συγκεκριμένα αναγνωρίζονται τα παρακάτω είδη πληροφοριακών συστημάτων:

- Enterprise Resource Planning systems (ERP)  
(ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΙΧΕΙΡΗΣΙΑΚΩΝ ΠΟΡΩΝ )
- Supply chain management systems (SCM)  
(ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΤΗΣ ΕΦΟΔΙΑΣΤΙΚΗΣ ΑΛΥΣΙΔΑΣ)
- Customer relationship management systems (CRM)  
(ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΣΧΕΣΕΩΝ ΜΕ ΤΟΥΣ ΠΕΛΑΤΕΣ)
- Knowledge management systems (KMS)  
(ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΙΧΕΙΡΗΣΙΑΚΗΣ ΓΝΩΣΗΣ)

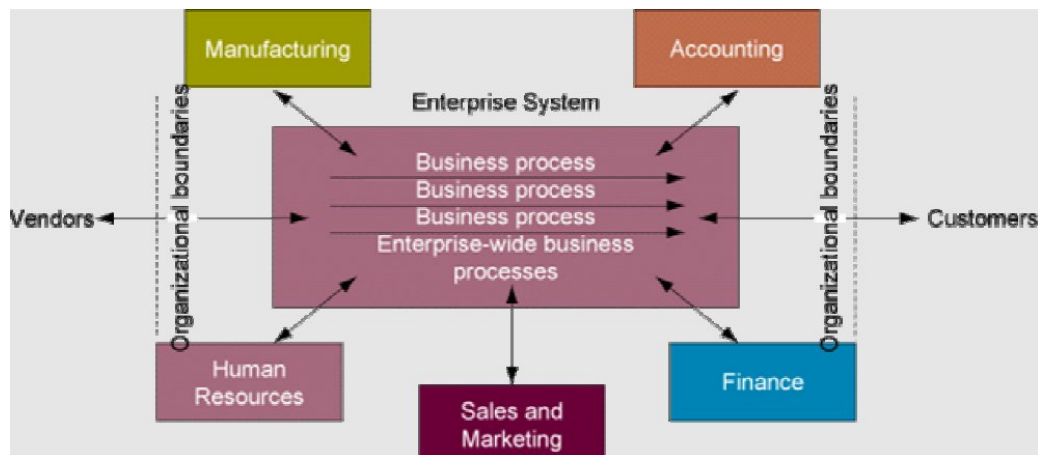
Τα συστήματα διαχείρισης πόρων (ERP) αποτελούν ένα ενιαίο πληροφοριακό σύστημα που συντονίζει και ολοκληρώνει τις επιχειρηματικές διεργασίες του οργανισμού. Κάτω από αυτό το σύστημα οι ροές πληροφοριών, δεδομένων και εργασιών ανάμεσα στα διάφορα αυτόνομα συστήματα υπολογιστή εκτελούνται με διαφάνεια ενώ πλέον οι πόροι είναι διαθέσιμοι στο μεγαλύτερο μέρος του οργανισμού. Τα πληροφορικά 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 εμπειρογνομosύνης. Οι υπηρεσίες που παρέχουν τα συστήματα αυτά καθώς και η σημασία της ένταξής τους στην τεχνολογική υποδομή του οργανισμού αποτελούν θέματα εκτενούς αναφοράς και δε θα αναλυθούν περαιτέρω εφόσον ξεφεύγουν από τα πλαίσια της διπλωματικής εργασίας.

Γεγονός είναι πλέον ότι οι οργανισμοί και οι επιχειρήσεις οργανώνουν σε μεγάλη κλίμακα τις ηλεκτρονικές επιχειρηματικές τους δραστηριότητες. Επιζητούν ακόμα μεγαλύτερη ολοκλήρωση των διεργασιών των διαφόρων τομέων και στοχεύουν στην σύνδεση των συστημάτων διαχείρισης πόρων, διαχείρισης της εφοδιαστικής αλυσίδας και σχέσεων με τους πελάτες μεταξύ του αλλά και με αυτά των πελατών, των προμηθευτών και των επιχειρηματικών εταίρων. Για αυτό το σκοπό έχουν αναπτυχθεί πακέτα επιχειρησιακού λογισμικού τα οποία, παρόλο που είναι ιδιαίτερα δαπανηρά και με μεγάλες απαιτήσεις πόρων, προσφέρουν πολλές υπηρεσίες όπως διαχείριση σχέσεων με τους πελάτες, διαχείριση της εφοδιαστικής αλυσίδας, υποστήριξη αποφάσεων κοκ. Επίσης μια σημαντική λύση ολοκλήρωσης αποτελούν οι πλατφόρμες υπηρεσιών οι οποίες προσφέρουν στη δημιουργία ολοκληρωμένων υπηρεσιών συνδυάζοντας επιχειρηματικές εφαρμογές από διάφορες λειτουργίες, επιχειρηματικές μονάδες ή εταιίρους. Για την οργάνωση και διαχείριση των αλλαγών των διεργασιών χρησιμοποιείται η τεχνική της διαχείρισης των επιχειρηματικών διεργασιών (business process management).

Προς την κατεύθυνση την ψηφιακής επιχείρησης και της ολοκλήρωσης των διεργασιών συντελεί ιδιαίτερα η ανάπτυξη του Internet, του Web αλλά και των γενικότερων εφαρμογών των ψηφιακών τεχνολογιών. Ένας γενικός όρος που χρησιμοποιείται για να αποδοθούν πληρέστερα οι δραστηριότητες αυτού του είδους είναι ο όρος *ηλεκτρονικό επιχειρείν* (*electronic business, e-business*). Το ηλεκτρονικό επιχειρείν περιλαμβάνει δραστηριότητες του ηλεκτρονικού εμπορίου που αντιστοιχούν στις διαδικασίες ηλεκτρονικών αγοραπωλησιών και στις σχετικές με αυτές διαδικασίες, 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 αυτής. Στο κεφάλαιο που ακολουθεί γίνεται μια εκτενέστερη αναφορά στις ηλεκτρονικές συναλλαγές που

πραγματοποιούνται στα πλαίσια της επιχειρηματικής δραστηριότητας και του ηλεκτρονικού επιχειρείν.



Εικόνα 1-Επιχειρησιακά συστήματα

# **3. Ηλεκτρονικές Συναλλαγές και Ηλεκτρονικές Υπηρεσίες**

## **3.1 Ορισμός και ανάλυση του πλαισίου των ηλεκτρονικών συναλλαγών**

Μέσα στην τελευταία δεκαετία τα πλαίσια δραστηριότητας των επιχειρήσεων παρουσιάζονται διευρυμένα και οι επιχειρηματικές συναλλαγές επωφελούνται πλέον από την 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.

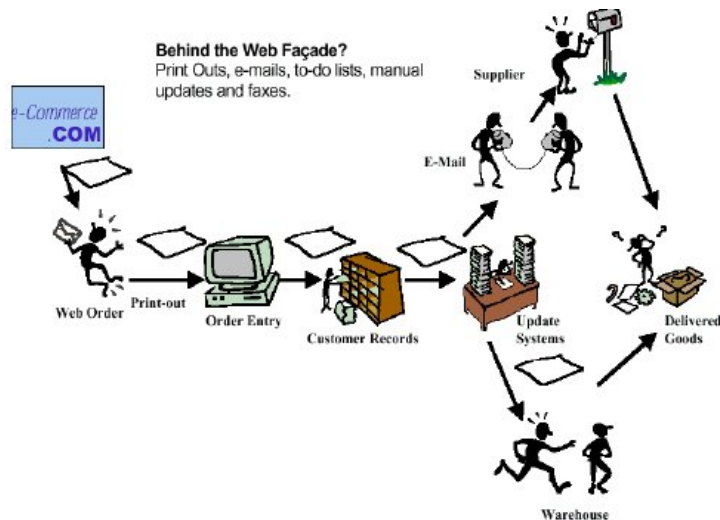
Αναλυτικότερα, η ραγδαία ανάπτυξη του Internet και του Web και η επιβλητική παρουσία τους σχεδόν σε κάθε πτυχή της καθημερινότητας επιτάσσουν την εισαγωγή και την καθιέρωσή τους και στον τομέα του εμπορίου. Έτσι εμφανίζονται στο προσκήνιο οι *ηλεκτρονικές συναλλαγές* οι οποίες θα μπορούσαν να οριστούν ως η πραγματοποίηση ανταλλαγής πληροφοριών μεταξύ δυο εταίρων με ηλεκτρονικά μέσα. Αυτές περιλαμβάνουν τις καθιερωμένες διαδικασίες αλληλεπίδρασης μεταξύ επιχειρήσεων (B2B), επιχειρήσεων και πελατών (B2C) αλλά και επιχειρήσεων με κρατικούς οργανισμούς (B2G), μόνο που διαφέρουν ως προς το είδος του περιβάλλοντος εκτέλεσής τους. Μέσω αυτών καθίσταται δυνατός ο εντοπισμός των επιθυμητών προϊόντων ή υπηρεσιών αλλά και η σύναψη εμπορικών σχέσεων με παραγωγούς και προμηθευτές. Επιπλέον οι συναλλαγές αυτές παρά την ηλεκτρονική τους φύση περιλαμβάνουν καθεμιά από τις γνωστές εμπορικές διαδικασίες όπως σύγκριση τιμών και ιδιοτήτων προϊόντων, αγορά, πώληση, έκδοση τιμολογίων κ.ο.κ. Χαρακτηριστικό σημείο διαχωρισμού από τις παραδοσιακές εμπορικές συναλλαγές αποτελεί η ύπαρξη των διαπροσωπικών των εφαρμογών σε αντίθεση από την ανθρώπινη παραδοσιακή επαφή, η απόσταση που μπορεί να χωρίζει

τους συναλλασσόμενους αλλά και οι διαφορές στις σημασίες που αποδίδουν σε βασικά επιχειρηματικά δεδομένα.

Οι περισσότερες εφαρμογές που λαμβάνουν μέρος στην πραγματοποίηση των ηλεκτρονικών συναλλαγών χαρακτηρίζονται από web front-end συστήματα, τα οποία μπορούν να παρέχουν και εξατομικευμένες λειτουργίες, από έναν εξυπηρετητή εφαρμογών (application server), ο οποίος μεσολαβεί ανάμεσα στην παρουσίαση και στις ίδιες τις εφαρμογές, και τέλος από ένα σύνολο back-end συστημάτων που είναι υπεύθυνα για την διαχείριση των εργασιών που απαιτούνται για την επιτυχία των συναλλαγών.

Βέβαια δεν πρέπει να παραλείψουμε το γεγονός ότι ακόμα και σήμερα υπάρχουν μελανά σημεία των ηλεκτρονικών συναλλαγών που αποδίδονται κυρίως στην ανάγκη ύπαρξης του ανθρώπινου παράγοντα. Πολλές φορές παρατηρούνται φαινόμενα ανθρώπινης παρέμβασης για την εισαγωγή ή μετατροπή δεδομένων, τον χειρισμό λαθών, τη σύνθεση διεργασιών. Τέτοια παραδείγματα παραπέμπουν σε συστήματα με έλλειψη ευελιξίας, πολύπλοκα και δύσκολα στον χειρισμό. Επίσης πολύ σημαντική αρνητική συνέπεια είναι και η αύξηση του κόστους αυτών των συστημάτων που μετακυλύεται στα κόστη των υπηρεσιών και των προϊόντων και κατ' επέκταση τη μείωση του περιθωρίου κέρδους των επιχειρήσεων [14].

Ένα άλλο ζήτημα που αξίζει να αναφερθεί έχει σχέση με την διαφορετική προέλευση των εφαρμογών που αποτελούν μέρος μιας ηλεκτρονικής διαδικασίας-συναλλαγής. Συνήθως οι διάφορες εφαρμογές παρέχονται από διαφορετικά συστήματα τα οποία μπορεί να βρίσκονται εντός των ορίων μια επιχείρησης ή στο εξωτερικό περιβάλλον αυτής. Ένα τέτοιο σενάριο παρουσιάζεται γραφικά στην Εικόνα 2. Στην περίπτωση όμως που είναι απαραίτητη η χρήση εφαρμογών ετερογενών συστημάτων τίθενται ζητήματα κατανόησης των διεπαφών των εφαρμογών (APIs), ετερογένειας των μοντέλων δεδομένων, ετερογένειας των υποκείμενων πρωτοκόλλων επικοινωνίας και των πλατφορμών εκτέλεσης, ζητήματα ασφάλειας και άλλα που συνθέτουν το πρόβλημα της διαλειτουργικότητας των συστημάτων. Το θέμα αυτό είναι ιδιαίτερα κρίσιμο και θα αναλυθεί σε επόμενη υποενότητα.



Εικόνα 2-Σενάριο ολοκληρωμένης επιχειρηματικής εφαρμογής

Γενικότερα, στην προσπάθεια αυτοματοποίησης των διαδικασιών που λαμβάνουν χώρα στις ηλεκτρονικές συναλλαγές χρειάζονται να ληφθούν υπόψη τα εξής βασικά κριτήρια[6]:

- *Σύζευξη μεταξύ των εταιρών (Coupling among partners)*: εκφράζει τον βαθμό (χαλαρή ή στενή σύζευξη) και την διάρκεια της σύζευξης (δυναμική ή μακράς διάρκειας) των εταιρών που συμμετέχουν σε μια ηλεκτρονική συναλλαγή.
- *Ετερογένεια (Heterogeneity)*: αναφέρεται στον βαθμό διαφοροποίησης των συναλλασσόμενων. Εκφράζεται σε επίπεδο τύπων δεδομένων, δομικών χαρακτηριστικών στο επίπεδο των επιχειρηματικών διαδικασιών και σημασιολογίας για την σύνθεση των εργασιών.
- *Αυτονομία (Autonomy)*: αναφέρεται στο βαθμό συμμόρφωσης των εταιρών 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)*: φανερώνει την ευκολία της εφαρμογής να προσαρμόζεται στις μεταβολές. Ειδικά στο δυναμικό επιχειρηματικό περιβάλλον εμφανίζονται συνέχεια νέες εφαρμογές, αναιρούνται κάποιες άλλες ενώ ορισμένες ανανεώνονται σε περιεχόμενο και δυνατότητες. Η επιρροή των αλλαγών αυτών εξαρτάται και από το βαθμό σύζευξης.
- *Ασφάλεια (Security)*: εκφράζει την ανάγκη ασφάλισης των εφαρμογών των διαφόρων συνεργατών μέσα από την εξασφάλιση αμοιβαίας ταυτοποίησης, εμπιστευτικότητας, ακεραιότητας της επικοινωνίας, αυθεντικότητας, μη αποκήρυξης.
- *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 σχέσεων που μπορεί να υποστηρίξει.

## **3.2 Οργάνωση ηλεκτρονικών συναλλαγών**

### **3.2.1 Χαλαρή και Ισχυρή Σύζευξη**

Μια από τις βασικότερες έννοιες που διαδραματίζει ουσιώδη ρόλο στην διασύνδεση καταναμημένων συστημάτων και την σύνθεση επιχειρηματικών διεργασιών είναι αυτή της σύζευξης (coupling) των αλληλεπιδρώντων μονάδων [11]. Ουσιαστικά υποδεικνύει τον βαθμό στον οποίο η λειτουργία μιας μονάδας λογισμικού, ενός συστήματος ή μιας γενικότερης διαδικασίας εξαρτάται από τις άλλες μονάδες. Βάσει αυτού του χαρακτηριστικού διακρίνονται δυο κατηγορίες σύζευξης η χαλαρή (loose) και η στενή (tight) σύζευξη. Ειδικότερα σε επιχειρηματικά περιβάλλοντα, το είδος της σύζευξης καθορίζει σε μεγάλο βαθμό την αλληλεπίδραση των συστημάτων μέσα στα όρια του οργανισμού αλλά και ανάμεσα σε ξένες επιχειρήσεις.

Το χαρακτηριστικό της στενής σύζευξης (tight coupling) αποδίδει την άμεση αλληλεξάρτηση που δημιουργείται ανάμεσα στα αλληλεπιδρώντα στοιχεία. Υπό συνθήκες στενής αλληλεξάρτησης υπάρχει ευκολότερος και πληρέστερος έλεγχος των διαδικασιών που λαμβάνουν χώρα ωστόσο εμφανίζονται προβλήματα ευελιξίας και επέκτασης των διαδικασιών ή των ορίων των συναλλαγών.

Η χαλαρή σύζευξη (loose coupling) από την άλλη πλευρά, δημιουργείται όταν επιδιώκεται ανεξαρτησία των συνεργαζόμενων μερών καθώς και μείωση της

πολυπλοκότητας των συναλλαγών. Παράλληλα προδίδει ευελιξία και μειώνεται ο κίνδυνος από αλλαγές των συνεργαζόμενων μονάδων. Ειδικότερα στις επαφές μεταξύ ετερογενών επιχειρήσεων είναι απαραίτητη η εύκολη εισαγωγή νέων σχέσεων χωρίς να επηρεάζονται οι ήδη υπάρχουσες, η κατάργηση κάποιων από τις ήδη εφαρμοζόμενες αλλά και η προσαρμογή των διαδικασιών των επιχειρήσεων στις μεταβαλλόμενες συνθήκες εμπορίου. Οι λειτουργίες που εκτελούνται από τους συνεργάτες δε θα πρέπει να μπλοκάρουν τις εργασίες των εταίρων τους όπως για παράδειγμα στην περίπτωση αναμονής μιας απάντησης ή κατά την χρήση των τεχνικών υποδομών ενός τρίτου.

Το χαρακτηριστικό της χαλαρής σύζευξης (loose coupling) έχει διάφορες διαστάσεις και συγκεκριμένα διακρίνουμε χαλαρή σύζευξη δυο εφαρμογών ως προς τον χρόνο χρησιμοποιώντας Message Oriented Middleware (αναλύεται περαιτέρω στην §4.3.4) υπονοώντας ότι η διαθεσιμότητα του ενός συστήματος δεν επηρεάζει το άλλο. Ένα άλλο επίπεδο χαλαρής σύζευξης διακρίνουμε ως προς τη μορφή (format) των δεδομένων όπου χρησιμοποιείται μεσολογισμικό για την μετατροπή των δεδομένων έτσι ώστε οι διαφορές στα μοντέλα δεδομένων να μη δημιουργούν προβλήματα ενοποίησης των διαδικασιών [19]. Τέλος σε επίπεδο υπηρεσιών (Web Services, SOA) η χαλαρή σύζευξη χαρακτηρίζεται ως διαφάνεια της υλοποίησης των εφαρμογών. Σε ένα κατανεμημένο περιβάλλον ξεχωρίζουν διάφορα επίπεδα πάνω στα οποία μπορούμε να εξετάσουμε το είδος της επιθυμητής σύζευξης. Αυτά αναλύονται στη συνέχεια ενώ γίνεται μια σύγκριση των δυο ειδών σύζευξης για καθένα από αυτά τα επίπεδα.

Ξεκινώντας από το φυσικό επίπεδο, όπου και εντοπίζεται η πιο πρόιμη μορφή σύζευξης, η στενή σύζευξη υπαγορεύει την ύπαρξη απευθείας συνδέσμου σε αντίθεση με τη χαλαρή σύζευξη όπου μεσολαβεί ένα φυσικό μέσο διασύνδεσης όπως παραδείγματος χάρη μια ουρά μηνυμάτων στην τεχνική Message Oriented Middleware (MOM -βλ. §4.3.4). Στην περίπτωση της στενής σύζευξης συναντούμε τις RPC κλήσεις όπου εκεί ο πελάτης και ο εξυπηρετητής συνδέονται απευθείας και η διαθεσιμότητα του εξυπηρετητή επηρεάζει τη λειτουργία του πελάτη.

Σε επίπεδο τύπου επικοινωνίας διακρίνουμε τη σύγχρονη και ασύγχρονη επικοινωνία. Η πρώτη περίπτωση αποτελεί κλασικό παράδειγμα στενής σύζευξης ενώ η δεύτερη χαλαρής σύζευξης. Παρόλα αυτά όταν επιδιώκεται χαλαρή σύζευξη θα πρέπει το υποκείμενο μεσολογισμικό να υποστηρίζει τον ασύγχρονο τρόπο επικοινωνίας. Όταν για παράδειγμα έχουμε RPC κλήσεις, εξυπηρετητής και πελάτης



μπορούν να ανταλλάσσουν μηνύματα ασύγχρονα όμως και πάλι θεωρείται ότι υπάρχει στενή σύζευξη αφού ο server πρέπει να είναι ενεργός και να δέχεται μηνύματα.

Διακρίσεις ανάμεσα σε στενή και χαλαρή σύζευξη παρατηρούνται και σε επίπεδο τύπου συστήματος, τόσο στη φάση ανάπτυξης της εφαρμογής όσο και στην αλλαγή και επανακαθορισμό του συστήματος. Ένα παράδειγμα αποτελεί η περίπτωση του ορισμού των διαπροσωπειών και των δεδομένων. Όσον αφορά στις διαπροσωπείες, παρατηρείται στενή σύζευξη των διαφόρων μονάδων εφόσον υπέχει ρητή αναφορά ονομάτων λειτουργιών και παραμέτρων. Μια αλλαγή επιφέρει συνήθως αλλαγή σε όλη την εφαρμογή ανάμεσα στις αλληλεξαρτώμενες λειτουργικές μονάδες. Από την πλευρά των δεδομένων μπορούμε να θεωρήσουμε ότι υπάρχουν στοιχεία χαλαρής σύζευξης εφόσον οι δομές (formats) των μηνυμάτων μπορούν να είναι πιο ευέλικτα και σε κάποιες περιπτώσεις μπορούν να εφαρμοστεί αναγνώριση του σχήματος (format) του μηνύματος όπως μέσω του XML Schema.

Επίσης σε επίπεδο αλληλεπίδρασης των διαφόρων μονάδων μπορεί να παρατηρηθεί διαφορά στη σύζευξη. Μια πιο στενή σύζευξη συνοδεύει την επικοινωνία Object oriented συστημάτων όπου εκεί είναι απαραίτητο να καταλάβει τη λογική (logic) κάθε αντικειμένου αλλά και τον τρόπο να μεταφέρεται από το ένα αντικείμενο στο άλλο. Στην περίπτωση των RPC συστημάτων η σύζευξη είναι στενότερη ενώ σε συστήματα βασισμένα στο MOM όπου το μόνο σημείο αλληλεπίδρασης είναι οι ουρές και το μήνυμα που παρέχεται ως είσοδος στον server, η σύζευξη είναι αρκετά χαλαρή.

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.

Τέλος, ο τρόπος που γίνεται η εύρεση των συνεργατών και των συνεργαζόμενων υπηρεσιών χαρακτηρίζεται από χαλαρή ή στενή σύζευξη. Η στατικά συνδεδεμένες υπηρεσίες αποτελούν ένα παράδειγμα στενής σύζευξης ενώ η δυναμική

σύνθεση των υπηρεσιών είναι αντιπροσωπευτική της χαλαρής σύζευξης (πχ. UDDI [58] στο περιβάλλον των Web Services).

Γενικότερα, ανάλογα με το περιβάλλον όπου λαμβάνουν χώρα οι επιχειρηματικές διαδικασίες και σύμφωνα με τα επιθυμητά επίπεδα ανεξαρτησίας, ευελιξίας και προσαρμοστικότητας των συναλλασσόμενων εφαρμογών, καθορίζεται η σύζευξη αυτών. Βέβαια δεν παραλείπεται το κόστος και η ιδιαίτερη πολυπλοκότητα που συνεπάγεται η εφαρμογή χαλαρής έναντι της στενής σύζευξης. Εντούτοις οι ετερογένεια των συστημάτων και των εφαρμογών των συνεργατών επιτάσσει τη στροφή προς τεχνικές χαλαρής σύζευξης.

### **3.2.2 Ενορχήστρωση- Χορογραφία**

Στη συνέχεια αναλύονται δυο πολύ βασικές τεχνικές οργάνωσης της ροής των συναλλαγών ή των υπηρεσιών και των αλληλεπιδράσεων των εφαρμογών που εμφανίζονται κατά τη διάρκεια των επιχειρηματικών δοσοληψιών εντός και εκτός των πλαισίων ενός οργανισμού[55].

- ***Ενορχήστρωση (Orchestration)***

Η προσέγγιση της *ενορχήστρωσης* ή *orchestration* όπως είναι ευρέως γνωστή, αναγνωρίζεται σε συστήματα εφαρμογών όπου η διαλειτουργικότητα των εφαρμογών επιτυγχάνεται βάσει μιας κεντρικά ελεγχόμενης ροής εργασιών (workflow). Δίνει τη δυνατότητα συγχώνευσης μεγάλων επιχειρηματικών διαδικασιών χωρίς να είναι απαραίτητη η ανάπτυξη ξανά των λύσεων που ενσωμάτωναν αρχικά τις διαδικασίες. Η λειτουργία της orchestration προσέγγισης αποσκοπεί στο να καλύψει το χάσμα με μια νέα ροή εργασιών (workflow). Έτσι η νέα αυτή ροή διεργασιών είναι πλέον πιο αφαιρετική και είναι εφικτή η μείωση της πολυπλοκότητας του περιβάλλοντος των δοσοληψιών.

Εξετάζοντας σε μεγαλύτερο βάθος τα χαρακτηριστικά της orchestration προσέγγισης, να αναφέρουμε καταρχάς ότι δεν λείπουν οι διάφοροι επιχειρηματικοί κανόνες, όροι και γεγονότα που ορίζουν ένα πρωτόκολλο αλληλεπίδρασης των συνεργαζόμενων μελών. Όλες οι λεπτομέρειες της λογικής της ροής εργασιών περιλαμβάνεται στον ορισμό της διαδικασίας. Σε αυτό τον ορισμό αναφέρονται επίσης και οι συμμετέχοντες στην διαδικασία. Η διαδικασία από μόνη της ονομάζεται

υπηρεσία διαδικασίας (*process service*) ενώ οι υπόλοιπες διαδικασίες που αλληλεπιδρούν με αυτή και καλούν ή καλούνται από αυτή ονομάζονται υπηρεσίες συνεργάτες (*partner services*). Οι εργασίες που λαμβάνουν μέρος μπορούν να χωριστούν σε βασικές ενέργειες (λήψη, κλήση, απάντηση, ρίψη, αναμονή)-*basic activities* (*receive, invoke, reply, throw, wait*)- οι οποίες επιτελούν βασικές εργασίες της ροής εργασιών και μπορούν να συντεθούν βάσει της λογικής δομής που προσφέρουν οι δομημένες ενέργειες -*structured activities* (*sequence, switch, while, flow, pick*). Γενικότερα από προτυποποιήσεις όπως η WS-Coordination[64] μέσα από την WS-BPEL [63], εξασφαλίζονται πρωτόκολλα συντονισμού για την υποστήριξη των πολύπλοκων εργασιών μεγάλης διάρκειας.

Η προσέγγιση της ενορχήστρωσης προσφέρει ένα περιβάλλον ενοποίησης με διάφορες εφαρμογές ενός οργανισμού έστω και αν αυτές βασίζονται σε διαφορετικές πλατφόρμες. Τέλος παρέχονται τρία βασικά χαρακτηριστικά. Υπάρχει δυνατότητα αλλαγής ή επέκτασης της λογικής της ροής εργασίας, διευκολύνεται η συγχώνευση των επιχειρηματικών διαδικασιών και ειδικά σε αρχιτεκτονικές που προσανατολίζονται σε υπηρεσίες (Service Oriented Architecture όπως αναλύεται στην § 4.2.2) δίνει τη δυνατότητα εξέλιξης σε επίπεδο ομοσπονδίων ετερόκλητων επιχειρήσεων.

- **Χορογραφία (Choreography)**

Η τεχνική της *χορογραφίας* ή *choreography* σε αγγλικούς όρους, ανταποκρίνεται στην ιδέα της απευθείας οργάνωσης των διεργασιών των οργανισμών για την εκτέλεση των συναλλαγών τους. Υπό ιδανικές συνθήκες οι οργανισμοί θα μπορούσαν να συμφωνήσουν πάνω στη δομή των εσωτερικών διεργασιών τους προκειμένου να διαλειτουργήσουν και να επιτύχουν αυτή την επιδίωξή τους βασισμένες 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[67].

Στα πλαίσια της προσέγγισης της χορογραφίας επιδιώκεται η ανταλλαγή μηνυμάτων εφόσον έχει εγκατασταθεί ένα πλαίσιο συνεργασίας των διαφόρων υπηρεσιών, ενώ παράλληλα θεωρείται ότι δεν υπάρχει οντότητα που να έχει τον

πλήρη έλεγχο της συνεργασίας αυτής. Πρόκειται ουσιαστικά για ένα καθολικό περιβάλλον διαλειτουργούμενων ανεξάρτητων οντοτήτων.

Στο περιβάλλον της χορογραφίας διακρίνονται στοιχεία ρόλων που στα Web Services ορίζουν τι κάνει μια εργασία και τι συγκεκριμένο μπορεί να κάνει μέσα στα πλαίσια μια συγκεκριμένης επιχειρηματικής λειτουργίας. Επίσης ορίζονται σύνολα ρόλων στο WSDL πρότυπο τα οποία σύνολα αποκαλούνται συμμετέχοντες (*participants*). Βασικό στοιχείο της χορογραφίας αποτελούν και οι σχέσεις (*relationships*) οι οποίες αντιπροσωπεύουν την κάθε απλή ανταλλαγή μηνύματος. Να σημειωθεί ότι τα μηνύματα είναι οι απλές δομικές μονάδες κάθε ενέργειας που ανήκει σε ένα σενάριο χορογραφίας. Επιπροσθέτως, υπάρχουν τα κανάλια επικοινωνίας (*channels*) τα οποία καθορίζουν τα χαρακτηριστικά των ανταλλαγών μηνυμάτων μεταξύ των ρόλων. Τέλος αναφέρονται άλλα δυο σημαντικά στοιχεία: οι αλληλεπιδράσεις (*interactions*) που αποτελούν τα δομικά στοιχεία της χορογραφίας εφόσον η ολοκλήρωση μιας αλληλεπίδρασης μαρτυρά πρόοδο της επικοινωνίας, καθώς και οι ενότητες εργασίας που περιλαμβάνουν κανόνες και περιορισμούς απαραίτητους για την επιτυχία μιας αλληλεπίδρασης.

Βασικές ιδιότητες της χορογραφίας είναι η επαναχρησιμοποίηση σε διαφορετικές επιχειρηματικές εργασίες, η δυνατότητα σύνθεσης των ξεχωριστών διεργασιών όσο και των ίδιων των χορογραφιών σε μεγαλύτερες οντότητες αλλά και οι ξεχωριστές μονάδες λειτουργικότητας που συνθέτουν μια χορογραφία και μπορούν να επαναχρησιμοποιηθούν.

Συνοψίζοντας, οι δύο προαναφερθείσες έννοιες εκφράζουν μοντέλα ανταλλαγής πολλαπλών μηνυμάτων σε περιβάλλοντα ηλεκτρονικών συναλλαγών. Η βασική διαφορά τους έγκειται στο ότι η ενορχήστρωση (*orchestration*) εκφράζει τη ροή εργασιών με βασικό συντελεστή έναν συγκεκριμένο οργανισμό αφού αυτός είναι που έχει τον έλεγχο και κατέχει το logic που αναπτύσσεται πίσω από την οργανωτική δομή του *orchestration*. Στην περίπτωση όμως της χορογραφίας (*choreography*), δεν υπάρχει ένας κύριος της επιχειρηματικής λογικής αλλά η όλη ιδέα δομείται στα πλαίσια συνεργασίας υπηρεσιών που προέρχονται από διαφορετικούς παρόχους.

### **3.2.3 Διαχείριση Επιχειρηματικών Διαδικασιών (Business process Management)**

Η διαχείριση επιχειρηματικών διαδικασιών ή business process management αποτελεί μια νέα τεχνολογική αντίληψη που έχει ως στόχο την διευκόλυνση του ελέγχου και της ενοποίησης των επιχειρηματικών συναλλαγών. Στην μοντέρνα εποχή οι επιχειρηματικές συναλλαγές απαιτούν την εμπλοκή δυο ή περισσότερων εφαρμογών σε πολύπλοκες, παράλληλες πολλές φορές διαδικασίες που ξεπερνούν τα στενά όρια μιας επιχείρησης ή οργανισμού και οι οποίες έχουν συνήθως μεγάλη διάρκεια. Υπό αυτές τις συνθήκες είναι απαραίτητες λειτουργίες που αντιμετωπίζουν τις διαφορές στις διαπροσωπείες εφαρμογών και τα ακολουθούμενα πρότυπα, ενώ παράλληλα επικεντρώνονται σε αυτοματοποίηση των διαδικασιών, σε ανάλυση αυτών και διαχωρισμό των διαφόρων επιπέδων. Η τεχνική BPM διακρίνει τις επιχειρηματικές διαδικασίες από τον υποκείμενο κώδικα ενοποίησης.

Στα πλαίσια της BPM τεχνικής οι υπηρεσίες μιας επιχείρησης που παράγουν αξία κωδικοποιούνται και η εκτέλεσή τους θεσπίζεται. Με τη χρήση διαφόρων εργαλείων είναι δυνατός ο ορισμός, η ανάλυση και η ενίσχυση της προτυποποίησης των διαδικασιών. Επιπλέον βάσει εργαλείων παρέχεται η δυνατότητα εικονικής κατασκευής, ανάλυσης και εκτέλεσης δια-λειτουργικών επιχειρηματικών διαδικασιών. Η διαχείριση των επιχειρηματικών διαδικασιών περιλαμβάνει την ανάλυση των διαδικασιών, τον ορισμό και επανακαθορισμό αυτών, την τοποθέτηση πόρων, τον προγραμματισμό, την μέτρηση της ποιότητας και της αποτελεσματικότητας των διαδικασιών και τη βελτιστοποίηση αυτών με τη χρήση πραγματικού-χρόνου μετρήσεων και μέτρων διαχείρισης της απόδοσης (performance management).

Η εφαρμογή των τεχνικών του BPM δεν πρέπει να απαιτεί πρόσθετη ανάπτυξη λογισμικού για την ενοποίηση της αλυσίδας των διαδικασιών ούτε μεγάλα κόστη συντήρησης. Οφείλει να χρησιμοποιεί διάφορα μέτρα για την παρακολούθηση της κατάστασης των διαδικασιών ατομικά αλλά και ως σύνολο.

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

## **3.3 Διαλειτουργικότητα Πληροφοριακών Συστημάτων**

### **3.3.1 Ορισμός διαλειτουργικότητας Πληροφοριακών Συστημάτων**

Αρκετές δεκαετίες πριν ο χώρος των επιχειρήσεων και των επιχειρηματικών διαδικασιών αντιμετώπιζε το πρόβλημα της απουσίας συγκεκριμένων μεθόδων και προτύπων ανάπτυξης και οργάνωσης των επιχειρηματικών εφαρμογών. Με το πέρασμα των χρόνων έγιναν προσπάθειες προς αυτή την κατεύθυνση οι οποίες απέδωσαν σημαντικά χρήσιμα αποτελέσματα. Ανάμεσα σε αυτά, κομβικής σημασίας ήταν τα σχεσιακά συστήματα διαχείρισης βάσεων δεδομένων- Relational Database Management Systems (RDBMS)- όπως επίσης και τα συστήματα διαχείρισης επιχειρηματικών πόρων (ERP) που αναπτύχθηκαν τελευταία και διευκολύνουν την οργάνωση των επιχειρηματικών εφαρμογών.

Οι μοντέρνες επιχειρήσεις είναι πλέον εξοπλισμένες άλλοτε με μεγάλα και πολύπλοκα Πληροφοριακά Συστήματα και άλλοτε με μικρότερα που επιτελούν βασικότερες επιχειρηματικές λειτουργίες. Το πρόβλημα που παρουσιάζεται ωστόσο στο επιχειρηματικό περιβάλλον είναι το λεγόμενο πρόβλημα της *διαλειτουργικότητας*. Σε γενικότερα πλαίσια που αφορούν τις επιχειρηματικές διαδικασίες, η διαλειτουργικότητα ορίζεται ως:

«...η δυνατότητα να αφήνεις ή να συμμετέχεις σε ένα επιχειρηματικό δίκτυο και να εγκαθιστάς ή να αφαιρείς συνδέσμους στο δίκτυο αυτό με μικρά, σε συνάρτηση προς το κέρδος, κόστη» [3].

Επίσης ένας ορισμός που αποδίδει από περισσότερο τεχνική σκοπιά τη διαλειτουργικότητα παρατίθεται στη συνέχεια:

«...η δυνατότητα ενός ή περισσοτέρων συστημάτων ή συστατικών μονάδων να ανταλλάσσουν και να χρησιμοποιούν πληροφορία η οποία έχει ανταλλαγή » [21].

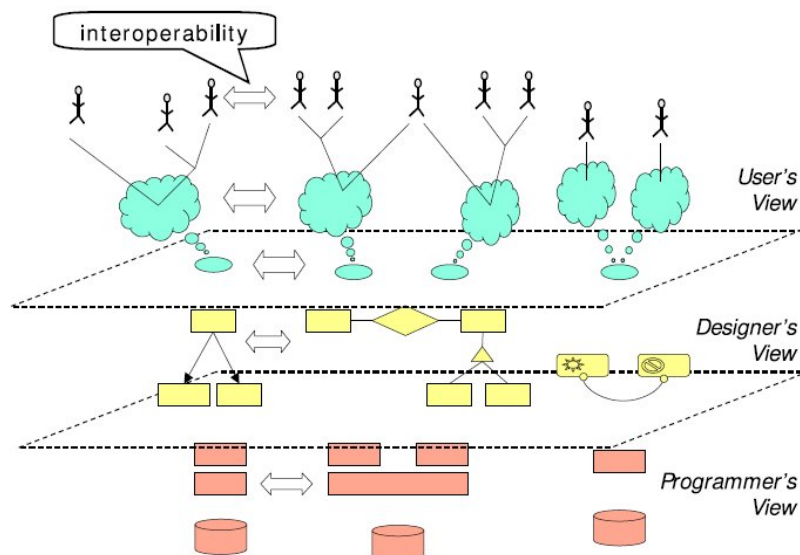
Βάσει και των δυο παραπάνω ορισμών, η διαλειτουργικότητα των πληροφοριακών συστημάτων προσδίδει στους οργανισμούς τη δυνατότητα να συνεργάζονται σε επίπεδο Πληροφοριακών Συστημάτων ανταλλάσσοντας πληροφορίες και γνώση και επεκτείνοντας τα όρια των επιχειρηματικών τους δραστηριοτήτων. Σε πολλές περιπτώσεις ωστόσο τα συνεργαζόμενα μέλη χαρακτηρίζονται από ετερογένεια και κατά συνέπεια προκύπτει το πρόβλημα της έλλειψης συμβατότητας, μερικής ή ολικής, που κάνει επιτακτική την ανάγκη εξεύρεσης λύσεων διαλειτουργικότητας. Ιδιαίτερα σε αυτό το σημείο θα μπορούσε να αναφερθεί ένας άλλος γενικότερος ορισμός της διαλειτουργικότητας:

«...η δυνατότητα ενοποίησης δεδομένων, λειτουργικότητας και διαδικασιών σύμφωνα με τη σημασιολογία τους»[48]

Στον προηγούμενο ορισμό διαφαίνονται και τα βασικά σημεία στα οποία επιδιώκεται η διαλειτουργικότητα των συστημάτων. Αυτά τα σημεία συνδέονται άμεσα με τις τρεις προκλήσεις που καλούνται να αντιμετωπίσουν οι μηχανικοί και οι διαχειριστές των Πληροφοριακών Συστημάτων και οι οποίες ορίζονται στα πλαίσια της επίτευξης διαλειτουργικότητας ως προς την οργάνωση, τη σημασιολογία και τις τεχνικές λεπτομέρειες των διαδικασιών, των υπηρεσιών και των δεδομένων. Αναλυτικότερα προκύπτουν οι εξής προκλήσεις[60]:

1. *Ενοποίηση των δεδομένων (data integration)*: διαδικασία παροχής ενός σχήματος δεδομένων που να γίνεται αντιληπτό από όλους τους συμμετέχοντες, εξαλείφοντας τις διαφορές που παρατηρούνται στην αναπαράσταση των δεδομένων των επιχειρηματικών εγγράφων λόγω της ετερογένειας των συστημάτων.
2. *Ενοποίηση των διαδικασιών (process integration)*: διαδικασία αναπαράστασης των επιχειρηματικών διαδικασιών των εταιρών σε μοντέλα διαδικασιών ώστε να είναι δυνατή η σύνθεσή τους σε εκτελέσιμες ροές υπηρεσιών.

3. *Ενοποίηση των υπηρεσιών (service integration)*: αναφέρεται γενικότερα στην σύνθεση ενός συνόλου ενεργειών που στόχο έχουν την τυποποίηση του ορισμού και της εκτέλεσης δυναμικών ροών εργασιών κατευθειάν από τους τελικούς χρήστες, καθώς και την παροχή υποδομών που εκθέτουν τις διαθέσιμες επιχειρηματικές εφαρμογές.



**Εικόνα 3-Επίπεδα διαλειτουργικότητας**

Αναλυτικά, αναφερόμενοι στο ζήτημα της διαλειτουργικότητας κινούμαστε στα πλαίσια των εξής πέντε τομέων υπηρεσιών[30]:

1. *Διευκόλυνση της επικοινωνίας*: βάσει μια συγκεκριμένης μορφής αναπαράστασης και σηματοδότησης της πληροφορίας, παρέχονται μοντέλα καναλιών επικοινωνίας τα οποία ορίζουν την προκαθορισμένη μορφή για την ανταλλαγή πληροφοριών. Αυτό το επίπεδο έχει σχέση με την διαλειτουργικότητα σε τεχνικό και σημασιολογικό επίπεδο.
2. *Διευκόλυνση της αφαιρετικής επεξεργασίας*: βασικά επιδιώκεται η κατανομή του φόρτου επεξεργασίας των πληροφοριών. Ουσιαστικά χρειάζεται να οριστεί η επεξεργασία αυτή ανεξάρτητα από την υπολογιστική πλατφόρμα (παράδειγμα αποτελεί η αρχιτεκτονική SOA). Το επίπεδο αυτό προσδιορίζει την τεχνική και σημασιολογική διαλειτουργικότητα.
3. *Διευκόλυνση της συνεργασίας με γνώση των διαδικασιών*: το σύστημα χρειάζεται να έχει τη δυνατότητα να εκφράζει τις συνεργατικές διαδικασίες και να ορίζει βήματα επεξεργασίας και επικοινωνίας σε αυτές. Αυτό έχει να



κάνει με τη σημασιολογική και πρακτική διαλειτουργικότητα. Ειδικές περιπτώσεις συνεργασιών μπορούν να αντιμετωπιστούν μόνο με προσεγγίσεις όπου υπάρχει γνώση των διαδικασιών και των πρακτικών.

4. *Διευκόλυνση της ανεξάρτητης εξέλιξης των συνεργαζόμενων μερών:* είναι σημαντική η απομόνωση που προσφέρουν οι υπηρεσιοστρεφείς προσεγγίσεις και η τεχνική της καθυστερημένης σύνδεσης (late binding) πάνω σε αφηρημένα κανάλια επικοινωνίας αλλά και οι αποθήκες δεδομένων ή οι οντολογίες που προσφέρουν μετα-πληροφορίες των υπηρεσιών (ιδιότητες, συμπεριφορά, αξιοπιστία).
5. *Διευκόλυνση της καθιέρωσης συνεργασιών:* είναι απαραίτητα μέσα για τη σύναψη συμβολαίων όπως για παράδειγμα οντολογίες που περιλαμβάνουν πληροφορίες για τους συνεργάτες, διαδικασίες διαπραγματεύσεων βάσει των διαφόρων επιπέδων διαλειτουργικότητας και ορισμοί σχέσεων εμπιστοσύνης μεταξύ των συνεργατών.

Επίσης πολύ βασικό στοιχείο της διαλειτουργικότητας αποτελεί η αυτονομία ως προς την υπολογιστική πλατφόρμα και το πρόγραμμα αλλαγών σε αυτή, σε τεχνικό επίπεδο. Σε επιχειρηματικό επίπεδο η αυτονομία ορίζεται ως προς την επιλογή των συστατικών, των εξωτερικά διαθέσιμων υπηρεσιών και της διάρκειας ζωής των εφαρμογών ενώ σε επίπεδο επιχειρηματικών πρακτικών η αυτονομία αφορά στην επιλογή των συνεργασιών των υπηρεσιών, στο είδος των συνεργατών και σε αποφάσεις αποχώρησης από συνεργασίες. Τέλος, στα πλαίσια της διαλειτουργικότητας ορίζονται στοιχεία όπως υποστήριξη της μεταφοράς, QoS, έλεγχος του λειτουργικού χρόνου συμμόρφωσης προς το συμβόλαιο συνεργασίας κλπ.

Στην παρούσα φάση ορίστηκε το θέμα της διαλειτουργικότητας των πληροφοριακών συστημάτων και σκιαγραφήθηκε το πρόβλημα που πηγάζει από την ανάγκη ύπαρξης αυτού του χαρακτηριστικού στα μοντέρνα Πληροφοριακά Συστήματα. Στις επόμενες ενότητες αναλύονται οι προκλήσεις που συνθέτουν το πρόβλημα της ετερογένειας των συστημάτων των επιχειρήσεων.

### 3.3.2 Ενοποίηση των δεδομένων

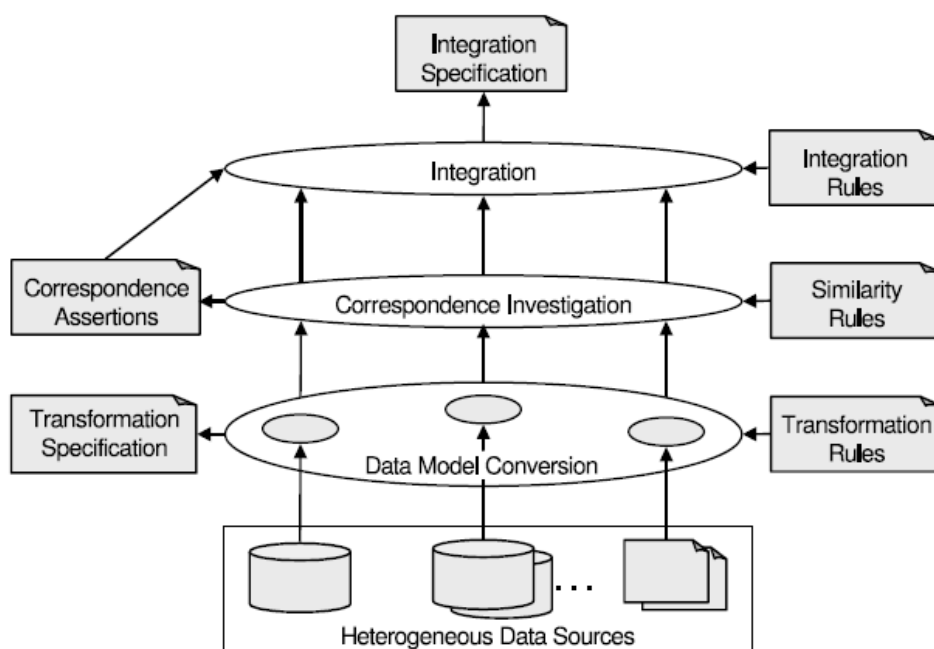
Βασικό συστατικό στο πλάνο επίτευξης της διαλειτουργικότητας αποτελεί η ενοποίηση των δεδομένων (Data Integration) που ανταλλάσσονται κατά τη διάρκεια των ηλεκτρονικών συναλλαγών. Τα διάφορα συστήματα χρησιμοποιούν και προσφέρουν υπηρεσίες οι οποίες περιλαμβάνουν την ανταλλαγή δεδομένων που ακολουθούν συγκεκριμένα μοντέλα δεδομένων, έχουν συγκεκριμένη σημασιολογία και είναι αποθηκευμένα με δεδομένο τρόπο. Ανάμεσα στα ετερογενή συστήματα αναμενόμενο είναι να παρατηρούνται διαφορές στα υιοθετούμενα πρότυπα αναπαράστασης (συντακτικά ή δομικά χαρακτηριστικά) και τις σημασιολογίες (ερμηνεία, έννοιες και αναμενόμενη χρήση των δεδομένων), γεγονός που καθιστά δύσκολη ή και αδύνατη τη διαλειτουργία των συστημάτων. Στη συνέχεια δίνεται μια διάκριση των ειδών ετερογένειας των δεδομένων[47]:

- *Συγκρούσεις δεδομένων (Data conflicts)*: διαφορές στην αναπαράσταση και την ερμηνεία των τιμών των δεδομένων όσον αφορά στις μονάδες μέτρησης, ακρίβεια και ορθογραφία.
- *Συγκρούσεις σχήματος (Schema conflicts)*: διαφορές στα σχήματα αναπαράστασης των ίδιων αντικειμένων χρησιμοποιώντας διαφορετικά ονόματα ή ιδιότητες στη θέση αντικειμένων κλπ.
- *Data versus Schema conflicts*: διαφωνίες ως προς το τι είναι δεδομένο και μετα-δεδομένο. Η τιμή σε ένα δεδομένο ενός σχήματος μπορεί να είναι αντικείμενο ή ιδιότητα σε ένα άλλο σχήμα.
- *Συγκρούσεις μοντέλων δεδομένων (Data model conflicts)*: διαφορές λόγω διαφορετικών μοντέλων δεδομένων.

Η έννοια της ενοποίησης των δεδομένων αναφέρεται στην διαδικασία παροχής πρόσβασης των εφαρμογών σε ενιαίας μορφής δεδομένα τα οποία έχουν ανάγκη, ξεπερνώντας το πρόβλημα των συγκρούσεων ετερογένειας. Συγκεκριμένα, υπό συνθήκες ενοποίησης δεδομένων (data integration), μια εφαρμογή έχει πρόσβαση στις απαραίτητες πληροφορίες σε οποιαδήποτε μορφή αναγνωρίζει αυτή χωρίς να περιορίζεται από το σχήμα, τη δομή, τη σημασιολογία ή την τοποθεσία που βρίσκονται τα δεδομένα που συνθέτουν τις πληροφορίες. Τρία βασικά

χαρακτηριστικά που πρέπει να προκύπτουν από την ενοποίηση των δεδομένων είναι: *συνέπεια, ακρίβεια και ορθότητα.*

Ένα σενάριο ενοποίησης παρουσιάζεται στη συνέχεια και αναπαρίσταται από την Εικόνα 3. Σύμφωνα με αυτό το σενάριο τα δεδομένα μετατρέπονται σε μια ομοιογενή μορφή πχ XML βάσει κανόνων μετατροπής. Επίσης σημαντικό είναι τα μεταφρασμένα δεδομένα και σχήματα να εμπλουτιστούν με πληροφορίες σημασιολογίας όπως μονάδες μέτρησης, ερμηνείες των όρων κοκ. Σε αυτή τη φάση είναι δυνατή η εξεύρεση αντιστοιχιών μεταξύ των αντικειμένων των ετερογενών πηγών, με τη χρήση σημασιολογικών περιγραφών και κανόνων ομοιότητας. Εντέλει οι ισχυρισμοί αντιστοιχίας που προκύπτουν εφαρμόζονται στα ετερογενή δεδομένα για να παράγουν μια ενοποιημένη μορφή.



**Εικόνα 4- Σενάριο ενοποίησης δεδομένων**

Οι μηχανισμοί που προσφέρουν την υλοποίηση της ενοποίησης μπορεί να είναι προσαρμογείς (adapters), wrappers, διαμεσολαβητές (mediators) και άλλοι μηχανισμοί μετατροπής των δεδομένων όπως επίσης και υπηρεσίες που προσφέρουν ενοποίηση των δεδομένων πχ. την πρόσβαση σε δεδομένα υπό μια ομογενή μορφή αναπαράστασης και χειρισμού των δεδομένων διατηρώντας πληροφορίες για κάθε ξεχωριστή πηγή (wrappers), υπηρεσίες ενοποίησης των συνόλων δεδομένων σε ανώτερο επίπεδο (mediators), υπηρεσίες που συγχωνεύουν τα προφίλ δυο πελατών,

υπηρεσίες που εξασφαλίζουν την συνέπεια των δεδομένων κλπ. Τα δεδομένα θα πρέπει να μετατραπούν ανεξάρτητα από τα πρότυπα στα οποία υπόκεινται, το λειτουργικό σύστημα που τα χειρίζεται και τη θέση στην οποία είναι αποθηκευμένα[44].

Σε παρούσα φάση χρησιμοποιούνται ευρέως τεχνολογίες μοντελοποίησης των δεδομένων όπως XML και UML, οι οποίες προσφέρουν αναπαράσταση των δεδομένων [6].

Ειδικότερα το πρότυπο της XML αναπαριστά τα δεδομένα μέσα στα ανταλλασσόμενα έγγραφα με στοιχεία (elements) τα οποία μπορεί να περιέχουν υποστοιχεία και ιδιότητες σε μια δενδρική δομή. Επίσης η δομή των εγγράφων αλλά και οι σχέσεις ανάμεσα στα στοιχεία περιγράφονται από το Document Type Definition ή XML Schema. Ένα παράδειγμα πλαισίου εφαρμογής της XML στην αναπαράσταση και τον ορισμό των δεδομένων αποτελεί η *xCBL (XML Common Business Library)*, η οποία περιλαμβάνει προτυποποιημένα XML έγγραφα που χρησιμοποιούνται κατά τις εμπορικές συναλλαγές. Προσφέρει ένα γενικότερο πλαίσιο περιγραφής του περιεχομένου των βασικών εμπορικών εγγράφων επωφελούμενη από την ύπαρξη κοινών ιδεών σε όλους τους εμπορικούς τομείς όπως πχ. εντολές εξόφλησης, παραγγελίες, τιμολόγια, ημερομηνίες, χρόνος, συναλλάγματα. Κατόπιν ο γενικός αυτός DTD κατάλογος μπορεί να εξειδικευτεί για να εξυπηρετήσει τη λειτουργία μιας εφαρμογής. Άλλο πλαίσιο που προσφέρει υπηρεσίες σε επίπεδο περιεχομένου είναι η *cXML (Commerce XML)* η οποία και πάλι καθορίζει ένα σύνολο από XML DTDs για περιγραφή στοιχείων με σκοπό να περιγράψει καταλόγους: Supplier, Index, Contract. Τα στοιχεία Supplier δίνουν πληροφορίες σχετικά με τους προμηθευτές, τα Index στοιχεία περιγράφουν τα στοιχεία των προμηθειών ενώ τα στοιχεία Contract αποδίδουν πληροφορίες των συμφωνιών μεταξύ των ανταλλασσόμενων μερών. Επίσης ένα χρησιμοποιούμενο πλαίσιο περιγραφής των δεδομένων που αφορούν τις εμπορικές συναλλαγές είναι το *RosettaNet* το οποίο σε επίπεδο περιγραφής των προϊόντων ορίζει κάποια λεξικά όπως το *RosettaNet Business Dictionary* στο οποίο περιγράφονται επιχειρηματικές ιδιότητες, το *RosettaNet Technical Dictionary* που περιέχει ιδιότητες που μπορούν να χρησιμοποιηθούν για την περιγραφή ιδιοτήτων προϊόντων και τέλος το *The RosettaNet Implementation Framework* το οποίο καθορίζει το περιεχόμενο των μηνυμάτων, τα πρωτόκολλα μεταφοράς και επικοινωνίας και κοινούς μηχανισμούς ασφαλείας. Πάλι στον ίδιο τομέα υφίσταται ένα ακόμα πλαίσιο το *ebXML*[12]

(*Electronic Business XML*) το οποίο στηρίζεται στην ιδέα ότι τα επιχειρηματικά έγγραφα αποτελούνται από δομικά συστατικά (components) και ειδικά αυτά μπορούν να διακριθούν σε Core components, Domain components και business information objects. Τα πρώτα είναι επαναχρησιμοποιήσιμες μονάδες πληροφορίας ενώ τα δυο άλλα παρέχονται από τις ίδιες επιχειρήσεις. Τέλος υπάρχουν και τα *UN/CFACT CCTS (Core Component Technical Specification)* τα οποία περιγράφουν οντότητες πληροφοριών που λαμβάνουν μέρος σε επιχειρηματικές συναλλαγές και αναφέρονται στα ανταλλασσόμενα έγγραφα.

Μια νέα μορφή περιγραφής των δεδομένων που λαμβάνουν μέρος στις επιχειρηματικές συναλλαγές είναι επίσης και οι γλώσσες αναπαράστασης οντολογιών όπως οι *RDF, OWL, DAML+OIL, Topic Maps* οι οποίες στηρίζονται στις οντολογίες που έχουν οριστεί για τους διάφορους τομείς και οι οποίες περιγράφουν με λεπτομέρεια του χρησιμοποιούμενους τύπους των δεδομένων καθώς και τις σχέσεις που τους συνδέουν αλλά και μπορούν να αναπαραστήσουν έννοιες που έχουν να κάνουν με περιορισμούς, σχέσεις ισοδυναμίας και διαχωρισμού καθώς και λογικούς συσχετισμούς.

Γενικότερα συνάγεται το συμπέρασμα ότι υπάρχουν ήδη πρότυπα που παρέχουν μια ενιαία βάση αναπαράστασης των δεδομένων με σκοπό την επίτευξη της ενοποίησης αυτών. Ωστόσο δεν μπορεί να εξασφαλιστεί η μη ύπαρξη διαφορών ανάμεσα στα χρησιμοποιούμενα μοντέλα δεδομένων και αυτό ενισχύει τους φόβους για αδυναμία ακρίβειας και συνέπειας των ανταλλασσόμενων πληροφοριών.

### **3.3.3 Ενοποίηση των διαδικασιών**

Η πραγματοποίηση των ηλεκτρονικών συναλλαγών στηρίζεται σε μεγάλο βαθμό στις επιχειρηματικές διαδικασίες (processes) των συμμετεχόντων. Με στόχο την τέλεια διασύνδεση των επιχειρηματικών εταίρων σε παγκόσμιο επίπεδο και την παροχή των λειτουργιών από εξωτερικούς παρόχους υπηρεσιών, υπάρχει ανάγκη ενοποίησης και αυτοματοποίησης των διαδικασιών τους ακολουθώντας ήδη υπάρχοντα μοντέλα διαδικασιών ώστε στη συνέχεια να μπορούν να προκύψουν εκτελέσιμες ροές υπηρεσιών.

Η ενοποίηση των διαδικασιών (process integration) έχει ως απώτερο στόχο την ανάπτυξη αυτοματοποιημένων διαδικασιών που αντιστοιχούν και προσφέρουν

λύσεις για επιχειρηματικές διαδικασίες, την ενσωμάτωση υπαρχουσών εφαρμογών σε επιχειρηματικές διαδικασίες και την ενοποίηση διαδικασιών με άλλες διαδικασίες. Ο διαδικασίες που ενοποιούνται ή ενσωματώνονται μπορεί να προέρχονται από την ίδια την επιχείρηση ή να ανήκουν σε εξωτερικές πηγές προερχόμενες από διαφορετικούς οργανισμούς[38].

Κάθε επιχειρηματική διαδικασία μπορεί να θεωρηθεί ως μια μονάδα εργασίας που ξεκινά τη λειτουργία της από μια απλή επιχειρηματική εργασία. Η διαδικασία αυτή στη συνέχεια οδηγείται από κανόνες, σκανδαλισμούς και υποδιαδικασίες, ενώ οι εργασίες και οι υποδιαδικασίες ανήκουν σε πόρους που αποτελούν μονάδες οργάνωσης και διαδραματίζουν συγκεκριμένο ρόλο στις διαδικασίες. Έτσι, ο ορισμός των κανόνων, των εργασιών, των υποδιαδικασιών και των πολιτικών των πόρων συνθέτουν την περιγραφή της διαδικασίας. Η εκτέλεση μιας διαδικασίας περιλαμβάνει την κλήση επιμέρους υπηρεσιών που μπορούν να βρίσκονται οπουδήποτε ενώ το περιεχόμενο που σχετίζεται με μια διαδικασία είναι αποθηκευμένο σε μια βάση δεδομένων. Ο διαχειριστής της διαδικασίας χειρίζεται την κατάσταση αυτής και δρομολογεί τις αιτήσεις ανάμεσα στις συμμετέχουσες εφαρμογές. Τα βήματα που ακολουθούνται κατά την εκτέλεση της διαδικασίας μπορεί να καθορίζονται εξ' αρχής πριν την εκτέλεση αυτής απεικονιζόμενα και με ένα διάγραμμα ροής αλλά πιθανό είναι επίσης να αναπαρίστανται βάσει κανόνων.

Ανεξάρτητα από τον τρόπο αναπαράστασης της εκτέλεσης των επιχειρηματικών διαδικασιών, η προσπάθεια αυτοματοποίησης των διαδικασιών ξεκινά με τον προσδιορισμό των βημάτων σε αφαιρετικό επίπεδο. Σε κάθε βήμα προσδιορίζονται οι εφαρμογές που θα εκτελεστούν και οι οποίες είτε πρέπει να αναπτυχθούν είτε είναι ήδη ανεπτυγμένες και πρέπει να εφαρμοστούν. Κατόπιν παράγονται οι περιγραφές και αυτές καταχωρούνται σε ένα σύστημα διαχείρισης των εργασιών. Κατά τη διάρκεια της εκτέλεσης δημιουργείται ένα στιγμιότυπο της διαδικασίας από το σύστημα διαχείρισης το οποίο συντονίζει την εκτέλεση των βημάτων, επιβλέπει και καταγράφει το ιστορικό. Παρόλα αυτά υπάρχει πιθανότητα κάποιες διαδικασίες να μην υποστηρίζονται από συστήματα διαχείρισης διεργασιών. Ειδικά αν θεωρήσουμε τη διάκριση των διαδικασιών ως προς την αυτοματοποίηση των εργασιών σε: ανθρωπο-κεντρικές (human centric) ή εφαρμογο-κεντρικές (application centric), και ως προς τη δομή των διαδικασιών: σε ισχυρά δομημένες, ημι-δομημένες και αδόμητες διαδικασίες βάσει της αυστηρότητας της δομής που ακολουθούν οι κανόνες και οι ακολουθίες που την συνθέτουν, μπορούμε να

διαπιστώσουμε ότι υπάρχει πρόβλημα ολικής υποστήριξης διαδικασιών. Υπαισέρχονται οι παράγοντες τις ανθρώπινης παρέμβασης και της μη δομημένης μορφής των κανόνων των διαδικασιών[57].

Με σκοπό την επίτευξη της ενοποίησης των διαδικασιών έχουν αναπτυχθεί διάφορες εφαρμογές μεσολογισμικού όπως: *μεσίτες μηνυμάτων (Message brokers)*, *διαχειριστές ουρών συναλλαγών (Transactional queue managers)* και *μηχανισμοί publish/subscribe*. Οι εφαρμογές αυτές διευκολύνουν την αυτοματοποίηση των διαδικασιών επιτρέποντας στις εφαρμογές να στείλουν μηνύματα και να αντιδρούν στα μηνύματα που λαμβάνουν από άλλες εφαρμογές. Συνήθως η λογική που υπαγορεύει τη συμπεριφορά αυτή καθορίζεται από τις ίδιες τις εφαρμογές, κατά συνέπεια υπάρχει αυτονομία και ευελιξία στις εφαρμογές και κατ' επέκταση οι αλλαγές των επιχειρηματικών αναγκών και πολιτικών μπορούν να αποτυπωθούν στη λογική της συμπεριφοράς των εφαρμογών.

Γενικότερα, πρόκειται για το επίπεδο σύνθεσης των ροών εργασιών που πρόκειται να εκτελεστούν στα πλαίσια μιας επιχειρηματικής συναλλαγής σε εκτελεστικό επίπεδο. Εν συντομία ορίζονται οι ροές αυτές οι οποίες αποτυπώνονται σε scripts ενώ υπάρχουν επίσης και οι διάφοροι κανόνες που ορίζουν δυναμικά την ροή κατά τη φάση εκτέλεσης, το επόμενο προς εκτέλεση βήμα αλλά και την σύνδεση της εκτέλεσης ενός βήματος σε έναν συγκεκριμένο πόρο. Οι αποφάσεις λαμβάνονται βάσει δεδομένων που βρίσκονται σε μηνύματα, βάσεις δεδομένων, εξαρτώνται από τη διαθεσιμότητα των πόρων και τις επιχειρηματικές πολιτικές.

### **3.3.4 Ενοποίηση των υπηρεσιών**

Αναφερόμενοι γενικότερα στις ηλεκτρονικές υπηρεσίες (e-services), αξίζει να δοθεί μια σύντομη περιγραφή της φύσης των υπηρεσιών αυτών μιας και το γενικότερο κεφάλαιο αναφέρεται και στις ηλεκτρονικές υπηρεσίες. Συγκεκριμένα πρόκειται για αυτόνομες, προερχόμενες από το διαδίκτυο εφαρμογές οι οποίες είναι ικανές όχι μόνο να επιτελούν επιχειρηματικές διαδικασίες αυτόνομα, αλλά και να συγκεντρώνουν άλλες ηλεκτρονικές υπηρεσίες προκειμένου να προσφέρουν ολοκληρωμένες επιχειρηματικές συναλλαγές[25].

Ειδικά το δυναμικό επιχειρηματικό περιβάλλον που οργανώνεται μέσα από το διαδίκτυο έχει ανάγκη από τη σύνθεση των διαφόρων υπηρεσιών ανάλογα με τη φύση της συναλλαγής που λαμβάνει χώρα. Οι προμηθευτές μπορούν να δημοσιοποιούν τις προσφερόμενες επιχειρηματικές τους διαδικασίες ενώ από την άλλη πλευρά οι πελάτες έχουν τη δυνατότητα να ψάχνουν σε κάποια αποθήκη υπηρεσιών για να εντοπίσουν εκείνες που ικανοποιούν τις απαιτήσεις τους μερικώς ή πλήρως. Στην περίπτωση όμως αυτή εμφανίζονται τα ζητήματα που έχουν να κάνουν με τη συνθεσιμότητα, την ορθότητα, τον συγχρονισμό και τον συντονισμό των διαδικασιών. Δεδομένης της ανάγκης σύνθεσης των διαφόρων υπηρεσιών, βασική είναι η ανάλυση της φύσης της διαδικασίας σύνθεσης και ιδιαίτερα η εξέταση των εννοιών της συνθεσιμότητας, της συμβατότητας, της προσαρμογής και της δυνατότητας αντικατάστασης. Τα χαρακτηριστικά που διαφοροποιούν την σύνθεση των εργασιών από την απλή ενοποίηση των ροών εργασίας (workflows) έχουν σχέση με τη δυναμική φύση του χώρου των υπηρεσιών ο οποίος ανανεώνεται συνεχώς, την ανάγκη άμεσης (on-the-fly) σύνθεσης των υπηρεσιών συναρτήσει των απαιτήσεων των καταναλωτών, την ύπαρξη υπηρεσιών με όμοια χαρακτηριστικά αλλά και διαφοροποιήσεις καθώς και την ύπαρξη πολλών διαστάσεων που αφορούν την ενοποίηση των υπηρεσιών όπως θέματα ασφαλείας, λειτουργιών, δεδομένων κλπ. Αναγνωρίζεται η ανάγκη ύπαρξης εργαλείων και τεχνικών για τη σύνθεση των υπηρεσιών σε ανώτερο επίπεδο αφαιρετικότητας καθώς η φύση της ενοποίησης των υπηρεσιών είναι πολύπλευρη και για παράδειγμα μπορεί είτε να οδηγείται από τα δεδομένα είτε από τις διαδικασίες είτε να είναι βασισμένη στις συναλλαγές αλλά και να συνδυάζει πολλές από αυτές τις οπτικές.

Η περίπτωση της ενοποίησης των υπηρεσιών (service integration) αποτελεί γενικότερη έννοια που αναφέρεται σε μια σειρά από ενέργειες όπως δημιουργία υπηρεσιών, περιγραφή υπηρεσιών, δημοσιοποίηση των υπηρεσιών σε αποθήκες του Διαδικτύου ή του Ενδοδικτύου, ανακάλυψη υπηρεσιών, κλήση των υπηρεσιών, σύνδεση (binding) και διαγραφή αυτών σε περίπτωση που δεν είναι διαθέσιμες[56]. Ουσιαστικά οι υπηρεσίες αυτές μπορούν να χρησιμοποιηθούν για την τυποποίηση του ορισμού και της εκτέλεσης των ροών εργασίας που δημιουργούνται δυναμικά ανάλογα με τις ανάγκες των τελικών χρηστών.

Σύμφωνα με πρότερη ερευνητική εμπειρία πάνω στο ζήτημα της σύνθεσης των ηλεκτρονικών υπηρεσιών[24], ιδιαίτερα σημαντική είναι η παράθεση μιας ολοκληρωμένης λύσης στο εν λόγω ζήτημα που θα περιλαμβάνει τις εργασίες της



εξεύρεσης, της εξέτασης συνέπειας και της σύνθεσης ενώ παράλληλα θα συνδυάζει τις αρετές της επαναχρησιμοποίησης και της δυνατότητας επέκτασης των υπηρεσιών.

Η σύνθεση των υπηρεσιών εκτείνεται σε τρεις φάσεις:

1. Σχεδιασμός (planning): φάση έρευνας πάνω στις υπηρεσίες που ικανοποιούν τα κριτήρια της σύνθεσης και της συμμόρφωσης. Ως αποτέλεσμα προκύπτει μια σύνθετη δομή υπηρεσιών.
2. Ορισμός (definition): φάση σύνθεσης της πραγματικής δομής υπηρεσιών. Προκύπτει η τεχνική περιγραφή της σύνθεσης των υπηρεσιών.
3. Υλοποίηση (implementation): φάση όπου γίνεται η σύνδεση με τις αντίστοιχες υπηρεσίες σύμφωνα και με την τεχνική περιγραφή της προηγούμενης φάσης.

Στις παραπάνω φάσεις χρησιμοποιούνται τρεις μορφές σύνθεσης:

- a. Η εξερευνητική σύνθεση (explorative composition): όπου η σύνθεση των υπηρεσιών προκύπτει on-the-fly και απαιτεί συντονισμό επιμέρους υπηρεσιών. Ο πελάτης ορίζει τις επιθυμητές υπηρεσίες και ένας μεσίτης υπηρεσιών κάνει συγκρίσεις με τις δημοσιοποιημένες υπηρεσίες παράγοντας έτσι πλάνα σύνθεσης. Από αυτά ο πελάτης μπορεί να επιλέξει κάποιο ανάλογο με τον βαθμό ικανοποίησης ορισμένων κριτηρίων.
- b. Η ημι-καθορισμένη σύνθεση (semi-fixed composition): εδώ κάποιες συνδέσεις με υπηρεσίες αποφασίζονται τη στιγμή της εκτέλεσης. Με την κλήση κάποιας σύνθετης εργασίας το πλάνο της σύνθεσης δομείται ανάλογο με τον συνδυασμό των δομικών υπηρεσιών που ορίστηκαν κατά την σύνθεση και των πιθανών διαθέσιμων υπηρεσιών. Ο ορισμός της σύνθεσης υπάρχει καταχωρημένος και αυτή μπορεί να συνδυαστεί με άλλες.
- c. Η καθορισμένη σύνθεση (fixed composition): όπως γίνεται αντιληπτό πρόκειται για προκαθορισμένη σύνθεση υπηρεσιών. Η δομή της σύνθεσης και οι υπηρεσίες που αποτελούν δομικά στοιχεία αυτής είναι στατικά συνδυασμένες και αιτήσεις σε αυτές τις σύνθετες υπηρεσίες στέλνονται σε αιτήσεις στις υπηρεσίες που είναι συστατικά αυτών.

Εν κατακλείδι, στα πλαίσια της σύνθεσης των υπηρεσιών λαμβάνουν χώρα εργασίες:

(1) περιγραφής της κλήσης των υπηρεσιών, (2) ελέγχου αντιστοιχίας και συμβατότητας, (3) περιγραφής της σύνθεσης των υπηρεσιών, (4) επίβλεψης εκτέλεσης των υπηρεσιών και του συντονισμού.

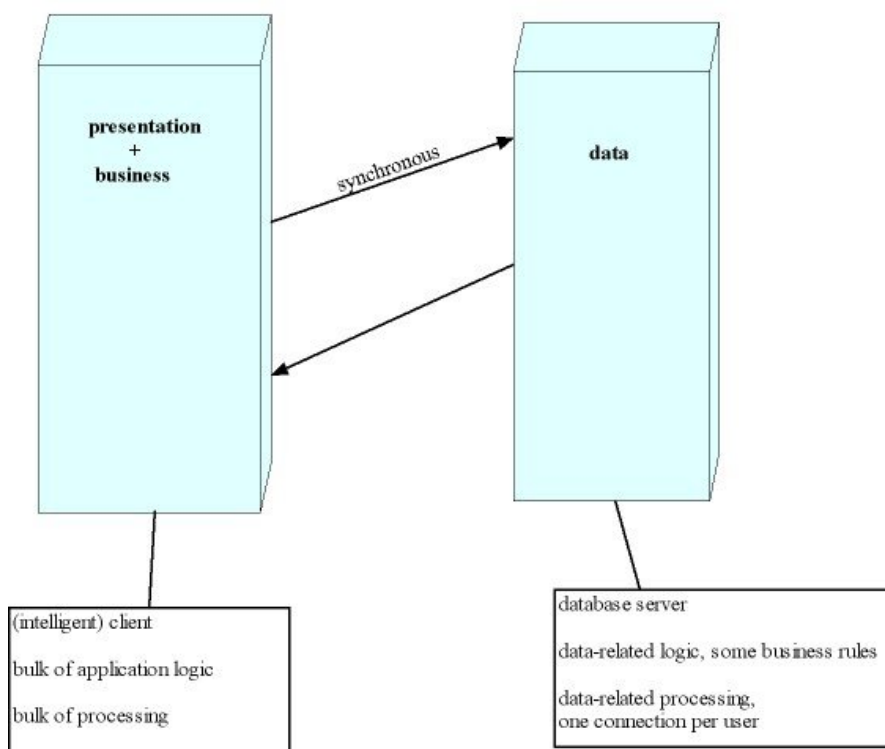
# 4 ■ State of the Art Analysis

## 4.1 Αρχιτεκτονικές Φυσικής Όψης

### 4.1.1 Αρχιτεκτονική Πελάτη-Εξυπηρετητή

Η αρχιτεκτονική *πελάτη-εξυπηρετητή* (*client-server*) είναι η πλέον διαδεδομένη μορφή αρχιτεκτονικής σε επίπεδο εφαρμογής. Παραπέμπει σε περιβάλλοντα όπου μονάδες εφαρμογών επικοινωνούν μεταξύ τους ζητώντας και λαμβάνοντας κομμάτια πληροφορίας.

Ιστορικά, σύμφωνα με το [55], αυτή η μορφή αρχιτεκτονικής εμφανίζεται σε πρώιμο στάδιο στα συστήματα ενός επιπέδου (*single-tier*) όπου τα συστήματα περιλάμβαναν τα δεδομένα, την παρουσίαση και την επιχειρηματική λογική σε έναν κόμβο που αποτελούσε και τον εξυπηρετητή (*server*). Αυτός εξυπηρετούσε τον πελάτη (*client*) στον οποίο απλά υλοποιούνταν το κομμάτι της παρουσίασης. Από τη δεκαετία των '90 εμφανίζεται πλέον η δυο επιπέδων (*two-tier*) αρχιτεκτονική *client-server*. Σύμφωνα με αυτή την προσέγγιση, οι ευθύνες πλέον της επιχειρηματικής λογικής και της επεξεργασίας μετατίθενται στον πελάτη. Η λειτουργία της επεξεργασίας περιλαμβάνει την παρουσίαση των δεδομένων όσο και τη λογική προσπέλασης αυτών. Ο εξυπηρετητής βάσει αυτής της προσέγγισης συγκεντρώνει τα δεδομένα, τους επιχειρηματικούς κανόνες και ορισμένες λειτουργίες που έχουν να κάνουν με την πρόσβαση των πελατών στα δεδομένα.



**Εικόνα 5- Τυπική αρχιτεκτονική δυο επιπέδων πελάτη-εξυπηρετητή**

Εξετάζοντας σε μεγαλύτερη λεπτομέρεια τα διάφορα χαρακτηριστικά της client –server αρχιτεκτονικής, ήδη αναφέραμε ότι ο πελάτης συγκεντρώνει τη λογική της εφαρμογής με εξαίρεση τους επιχειρηματικούς κανόνες που είναι συνδεδεμένοι με διαδικασίες και σκανδαλισμούς (triggers) στα συστήματα βάσεων δεδομένων.

Όσον αφορά στην επεξεργασία που λαμβάνει χώρα κατά την εφαρμογή, το μεγαλύτερο ποσοστό του φόρτου εργασίας μετατίθεται στον πελάτη ο οποίος συγκεντρώνει και τη λογική της εφαρμογής (application logic). Η επικοινωνία γίνεται σύγχρονα ενώ ο πελάτης (client) χρειάζεται να εγκαταστήσει τη δική του σύνδεση με τη βάση δεδομένων που βρίσκεται στον εξυπηρετητή, η οποία μάλιστα σύνδεση παραμένει ενεργή μέχρι ο χρήστης να αφήσει την εφαρμογή. Αν και ο εξυπηρετητής (server) είναι επιφορτισμένος με μικρότερο ποσοστό της εργασίας, αποτελεί συχνά το σημείο συμφόρησης (bottleneck) του συστήματος λόγω των πολλαπλών συνδέσεων των πελατών στη βάση αλλά και των σημαντικών απαιτήσεων σε πόρους. Το ίδιο επιβαρυνόμενοι είναι και οι πελάτες λόγω του όγκου των πόρων που είναι απαραίτητοι για την διεκπεραίωση των εργασιών.

Σε επίπεδο ασφάλειας, ο εξυπηρετητής και ειδικότερα τα συστήματα βάσεων δεδομένων προσφέρουν μια ασφαλή κεντροποιημένη διαχείριση των λογαριασμών των χρηστών και των ομάδων. Επιπλέον υπάρχει η δυνατότητα ελέγχου ασφαλείας

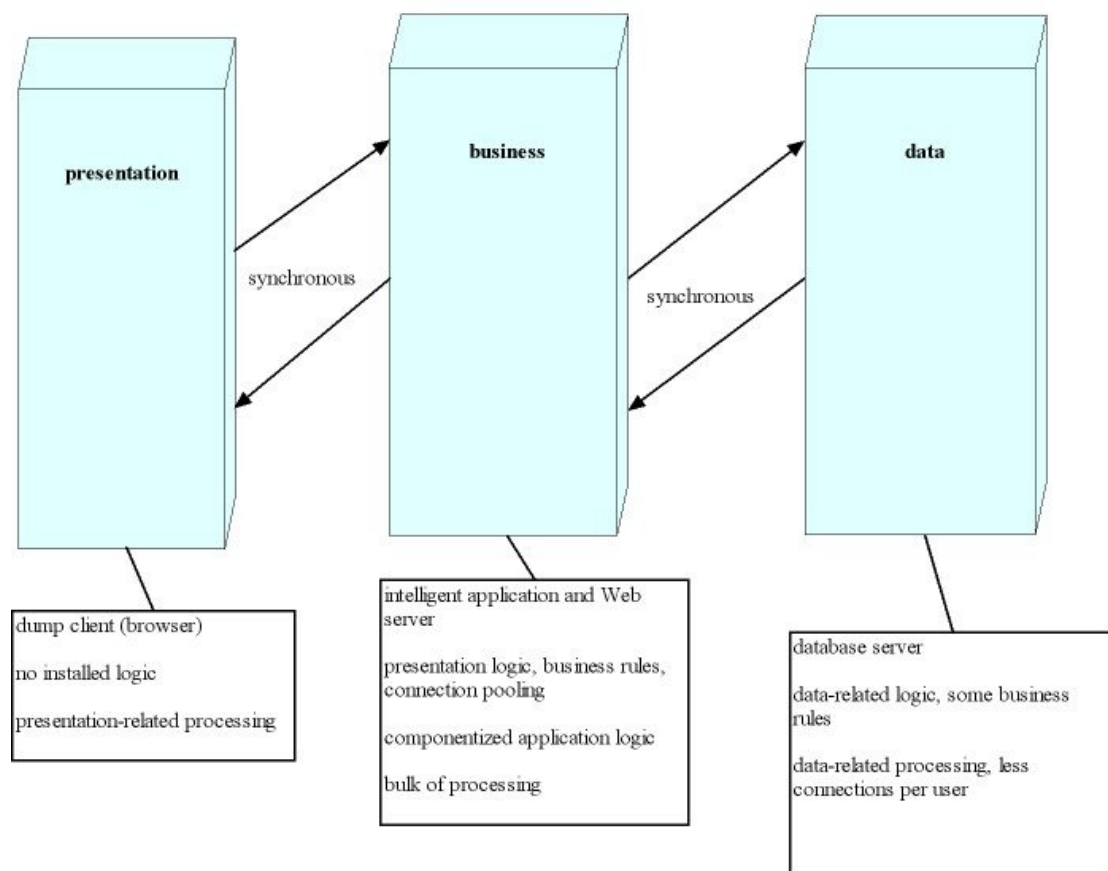
μέσα στο εκτελέσιμο του πελάτη αλλά και η τεκμηρίωση της καθαρότητας της εφαρμογής από τη χρήση πληροφοριών του λογαριασμού εισόδου του λειτουργικού συστήματος του πελάτη.

Τέλος, όσον αφορά στο θέμα της διαχείρισης των πόρων του συστήματος, η αρχιτεκτονική πελάτη-εξυπηρετητή αποδείχτηκε αρκετά δαπανηρή. Οι λόγοι που οδήγησαν σε αυτή τη διαπίστωση έχουν να κάνουν με τα κόστη διανομής και συντήρησης της λογικής της εφαρμογής. Ειδικότερα, εφόσον κάθε πελάτης είχε τον κώδικα της εφαρμογής ήταν επόμενο μια αναβάθμιση ή αλλαγή της εφαρμογής να απαιτεί την ενημέρωση όλων των καταναμημένων πελατών. Επιπλέον κάθε πελάτης που χρησιμοποιούσε σταθμούς εργασίας για την εκτέλεση των εφαρμογών μπορούσε να χαρακτηρίζεται από ετερογένεια στα προγράμματα λογισμικού ή στην προέλευση του υλικού. Επίσης δεν πρέπει να παραλείψουμε το πρόβλημα του μεγέθους των βάσεων δεδομένων και των ρυθμών επέκτασής τους με σκοπό την κάλυψη των αναγκών των πελατών.

#### **4.1.2 Καταναμημένη Διαδικτυακή Αρχιτεκτονική**

Η καταναμημένη διαδικτυακή αρχιτεκτονική (distributed internet architecture) εμφανίστηκε με τον ερχομό του Παγκόσμιου Ιστού (World Wide Web) και ενώ στο προσκήνιο βρίσκονταν οι πολύ-επίπεδες αρχιτεκτονικές (multi-tier architectures). Ήδη η αρχιτεκτονική δυο επιπέδων που περιγράφηκε προηγουμένως, λόγω του μεγάλου κόστους και των περιορισμών που έθετε, σύντομα αντικαταστάθηκε από τις πολύ-επίπεδες αρχιτεκτονικές (multi-tier architectures) οι οποίες έθεταν ως βασικούς άξονα την κατανομή του της λογικής της εφαρμογής ανάμεσα σε διάφορες μονάδες είτε στην πλευρά του εξυπηρετητή είτε του πελάτη. Ειδικά στην περίπτωση των μονάδων που βρίσκονταν στην πλευρά του εξυπηρετητή εμφανίστηκαν οι εξυπηρετητές εφαρμογών (application servers) οι οποίοι διευκόλυναν στη διαχείριση των πολλαπλών συνδέσεων στις βάσεις δεδομένων. Επιπλέον οι συνδέσεις στις βάσεις δεδομένων αντικαταστάθηκαν από τις remote procedure calls (RPC) με τις τεχνολογίες όπως η CORBA και DCOM που υποστήριζαν τις απομακρυσμένες κλήσεις ανάμεσα στις μονάδες του πελάτη και των εξυπηρετητών. Ωστόσο τα μειονεκτήματα ήταν ποικίλα όπως για παράδειγμα προβλήματα πόρων, μόνιμων συνδέσεων, ανάπτυξης και διαχείρισης.

Με την εμφάνιση του Παγκόσμιου Ιστού (World Wide Web) και των φυλλομετρητών (browsers) η λογική της εφαρμογής (application logic) μετατέθηκε στον server ενώ εμφανίστηκε παράλληλα ένα μεσαίο επίπεδο γνωστό ως Web Server ο οποίος ήταν επιφορτισμένος με την επικοινωνία εξυπηρετητή (server) και πελάτη (client) μέσω HTTP πρωτοκόλλου αντί για RPC. Με την ευρεία χρήση προγραμματιστικών τεχνικών βασιζόμενων σε ψηφίδες (component-based) και την εξέλιξη του μεσολογισμικού επιτεύχθηκε μειωμένη πολυπλοκότητα.



**Εικόνα 6- Τυπική καταναμημένη αρχιτεκτονική διαδικτύου**

Όπως ήδη αναφέρθηκε, η λογική της εφαρμογής (application logic) των εφαρμογών στη καταναμημένη αρχιτεκτονική μεταφέρθηκε στο μέρος του server καθιστώντας έτσι την αρχιτεκτονική καθαρά κεντροποιημένη. Επιπλέον οι μονάδες που δομούν τις καταναμημένες εφαρμογές βρίσκονται σε έναν ή περισσότερους application servers. Οι διάφορες αυτές μονάδες-ψηφίδες εφαρμογών είναι σχεδιασμένες ανάλογα με τις λειτουργίες που καλούνται να εκτελέσουν καθώς και ανάλογα με το πόσο επαναχρησιμοποιήσιμες πρόκειται να είναι. Οι μονάδες-ψηφίδες που βρίσκονται στον ίδιο εξυπηρετητή επικοινωνούν μεταξύ τους μέσω διεπαφών

(APIs) ενώ για την επικοινωνία μεταξύ διαφορετικών εξυπηρετητών χρησιμοποιούνται πρωτόκολλα RPC μέσω proxy. Επιπρόσθετα, κατά την σχεδίαση των μονάδων αυτών οι αλληλεπιδράσεις που πρόκειται αυτές να έχουν με άλλες μονάδες λαμβάνονται υπόψη και ενσωματώνονται στον κώδικα. Βέβαια πρόκειται για το χαρακτηριστικό χαλαρής σύζευξης (tight coupling) το οποίο πιθανότατα να δημιουργήσει προβλήματα σε περίπτωση που χρειάζεται να γίνουν αλλαγές στις διασυνδέσεις των διαφόρων μονάδων.

Σε επίπεδο επεξεργασίας δεδομένων, η κατανεμημένη αρχιτεκτονική υποστηρίζεται από πρωτόκολλά όπως το DCOM και από εμπορικές υλοποιήσεις της CORBA για απομακρυσμένη ανταλλαγή δεδομένων. Οι τεχνολογίες αυτές θεωρούνται αρκετά αξιόπιστες και αποτελεσματικές, αν και έχουν συναντήσει προκλήσεις, ενώ υποστηρίζουν μονάδες που διατηρούν ή όχι την κατάστασή τους (statefull και stateless) και επικοινωνούν με σύγχρονο τρόπο ανταλλάσσοντας δεδομένα.

Για την υποστήριξη της επικοινωνίας έχουν χρησιμοποιηθεί μονάδες, κώδικας από την πλευρά του εξυπηρετητή και Web τεχνολογίες όπως HTML και HTTP. Επίσης η XML πρόσφερε τη δυνατότητα πληρέστερης αναπαράστασης των δεδομένων και έδωσε κάποιο νόημα στο περιεχόμενο που μεταφέρεται μέσω των πρωτοκόλλων του Internet.

Εξετάζοντας το ζήτημα της ασφάλειας στην παρούσα κατανεμημένη αρχιτεκτονική, να επισημάνουμε ότι πλέον δεν ισχύει η εύκολη ταυτοποίηση των χρηστών όπως συνέβαινε στην αρχιτεκτονική δύο επιπέδων. Εδώ τα δεδομένα πρέπει να μεταφερθούν ανάμεσα σε πολλά διαφορετικά φυσικά επίπεδα χωρίς να αντιμετωπίσουν προβλήματα αλλοίωσης και διατηρώντας τα στοιχεία αναγνώρισης των χρηστών. Ένας τρόπος για να εξασφαλιστεί η ασφάλεια είναι η κρυπτογράφηση των δεδομένων κατά τη μεταφορά τους από τον Εξυπηρετητή Ιστού (Web Server).

Γενικότερα, η κατανεμημένη αρχιτεκτονική επιτάσσει συνεχή έλεγχο των διαφόρων στοιχείων που συνθέτουν το σύστημα επικοινωνίας. Πιο αναλυτικά, είναι απαραίτητο να ελέγχονται τα στιγμιότυπα των διαφόρων ψηφίδων, να εντοπίζονται πιθανά προβλήματα σε τοπικό ή απομακρυσμένο επίπεδο, να ελέγχονται οι απαιτήσεις σε πόρους και οι εργασίες διαχείρισης των βάσεων δεδομένων. Ειδική μέριμνα πρέπει να δίνεται για τον έλεγχο των Εξυπηρετητών Ιστού οι οποίοι είναι και τα σημεία επικοινωνίας των πελατών εντός και εκτός οργανισμού.

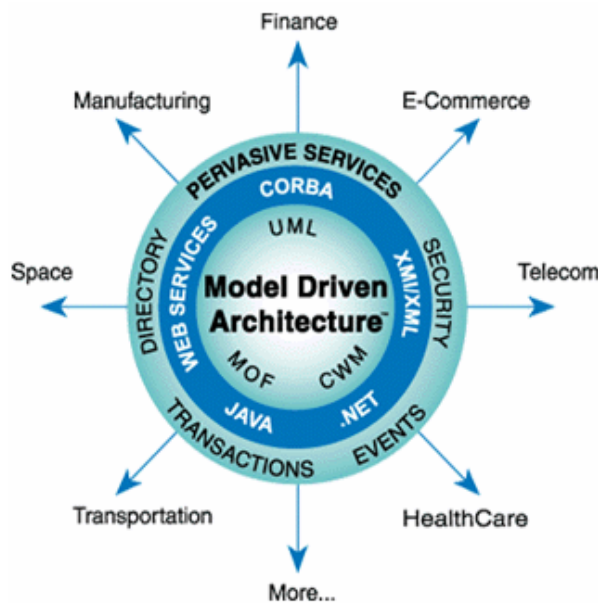
## **4.2 Generic Αρχιτεκτονικές**

### **4.2.1 Αρχιτεκτονική Οδηγούμενη από τα Μοντέλα**

Η Αρχιτεκτονική Οδηγούμενη από τα Μοντέλα, όπως θα μπορούσε να αποδοθεί στα ελληνικά ο όρος Model Driven Architecture (MDA), αποτελεί πρόταση του οργανισμού OMG[41] που στοχεύει στην επίτευξη της επιδιωκόμενης διαλειτουργικότητας των εφαρμογών και των μοντέλων των επιχειρήσεων.

Σύμφωνα με τον ορισμό που δίνεται από το OMG[41] η MDA *παρέχει μια ανοικτή και ουδέτερη ως προς τον πάροχο προσέγγιση ενάντια την πρόκληση αλλαγών των επιχειρήσεων και της τεχνολογίας*. Η MDA θέτει ως στόχο την ανάπτυξη συστημάτων που ικανοποιούν τις ανάγκες των χρηστών με περισσότερη ευελιξία ενώ παράλληλα υποστηρίζει τη συνεχή εξέλιξη των συστημάτων χωρίς να απαιτεί μεγάλες αναπροσαρμογές των στοιχείων της. Γεγονός είναι ότι δομείται πάνω στα ήδη θεσπισμένα και ευρέως χρησιμοποιούμενα πρότυπα λειτουργίας των συστημάτων και κατά αυτό τον τρόπο διευκολύνει την ενοποίηση και τη συνεργασία των ετερογενών συστημάτων μέσα και έξω από τα σύνορα των οργανισμών.

Βασικό στοιχείο της παρούσας αρχιτεκτονικής καθίσταται ο διαχωρισμός της επιχειρηματικής (business) και της λογικής των εφαρμογών (application logic) από το τεχνολογικό υπόβαθρο (λειτουργικό σύστημα, γλώσσες μοντελοποίησης και προγραμματισμού, εξυπηρετητές δεδομένων και repositories). Μοντέλα εφαρμογών που είναι ανεξάρτητα από την τεχνολογική πλατφόρμα όπως και η λειτουργικότητα ή η συμπεριφορά των διαφόρων συστημάτων τα οποία εκφράζονται σε μοντέλα αναπαράστασης όπως η UML, ή άλλα πρότυπα που αναγνωρίζονται από το OMG[41], μπορούν πλέον να εφαρμοστούν χωρίς περιορισμούς εικονικά σε οποιαδήποτε από τις υπάρχουσες πλατφόρμες όπως Web Services, .NET, CORBA, J2EE και άλλες. Το επίπεδο αφαιρετικότητας που προσφέρει η MDA είναι το βασικό πλεονέκτημά της καθώς παρέχοντας τη δυνατότητα διαχωρισμού επιχειρηματικών θεμάτων από την υλοποίηση σε τεχνικό επίπεδο, διευκολύνει στην περίπτωση αλλαγών ή εξέλιξης που επιτάσσει η τεχνολογία. Ουσιαστικά σε αυτή την περίπτωση χρειάζεται μόνο ο καθορισμός των τεχνολογικών λεπτομερειών βασιζόμενοι πάντα στα ίδια επιχειρηματικά πρότυπα και μοντέλα.



Εικόνα 7- Αρχιτεκτονική που οδηγείται από τα μοντέλα (Model Driven architecture)

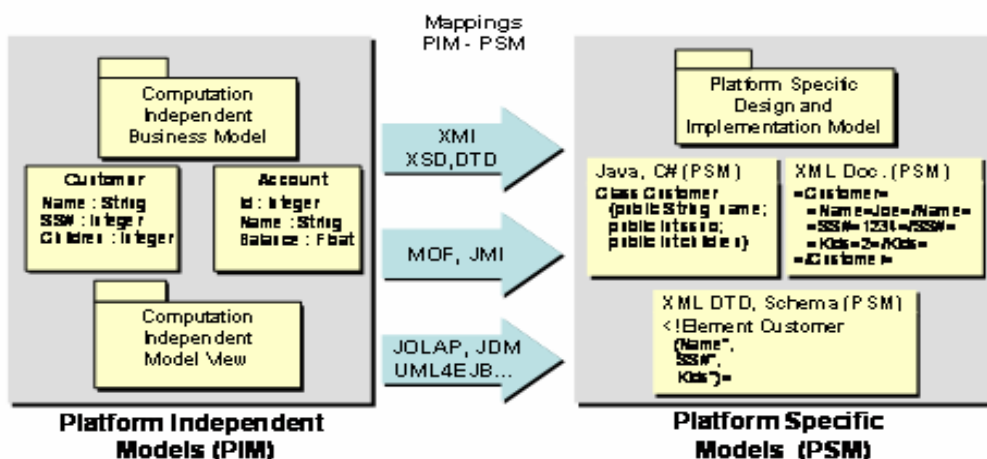
Η MDA αρχιτεκτονική δομείται σε τρία βασικά επίπεδα:

- Το πρώτο και ανώτερο αφαιρετικό επίπεδο ονομάζεται *Computational-Independent Model (CIM)* ή *Μοντέλο Ανεξάρτητο Υπολογιστικού Ορισμού* και έχει ως στόχο να αναπαραστήσει τον τομέα στο οποίο εφαρμόζεται το σύστημα. Περιγράφει το περιβάλλον εφαρμογής από επιχειρηματική σκοπιά και δεν καθορίζει τεχνικές λεπτομέρειες που αφορούν στην δομή ή τα σχέδια υλοποίησης.
- Το δεύτερο ενδιάμεσο επίπεδο είναι γνωστό ως *Platform-Independent Model (PIM)* ή *Μοντέλο Ανεξάρτητο Πλατφόρμας Υλοποίησης* και καθορίζει σχεδιαστικές λεπτομέρειες ανώτερου επιπέδου του IT συστήματος. Βασίζεται σε διαγράμματα χρήσης και μετα-μοντέλα εστιάζοντας περισσότερο στο ιδεατό μοντέλο του συστήματος. Για τον σκοπό αυτό χρησιμοποιεί σχεδιαστικά πρότυπα όπως Unified Modelling Language (UML), Object Constraint Language (OCL), XML Metadata Interchange (XMI), Meta Object Facility (MOF) και Common Warehouse Meta-model (CWM). Είναι σημαντικό να αναφερθεί ότι η δυνατότητα ανάλυσης, αυτοματοποίησης και μετατροπής των μοντέλων παρέχεται και μέσα από την ύπαρξη μετα-μοντέλων τα οποία περιγράφουν ξεκάθαρα τη σημασιολογία των μοντέλων που χρησιμοποιούνται όπως για παράδειγμα σε ένα UML διάγραμμα τη σημασία των κλάσεων, των ιδιοτήτων και των σχέσεων που αναπαρίστανται.



Ειδικότερα ο OMG[41] αναγνωρίζοντας τη σημασία των μετα-μοντέλων έχει θεσπίσει μια γλώσσα περιγραφής του συντακτικού των δομών μοντελοποίησης, τη MOF. Περνώντας και πάλι στο επίπεδο του PIM μοντέλου να επισημανθεί ότι πρόκειται στην πραγματικότητα για ένα επίπεδο αναπαράστασης της λειτουργικότητας του συστήματος ενώ παράλληλα μεταθέτει τους περιορισμούς και τις τεχνικές λεπτομέρειες στο επόμενο προς τα κάτω επίπεδο που αναλύεται στη συνέχεια. Στο PIM επίπεδο ουσιαστικά χρησιμοποιούνται οι κατάλληλες επιχειρηματικές έννοιες, τα απαραίτητα μοντέλα χρήσης (use case models) και προσδιορίζονται οι διαπροσωπείες.

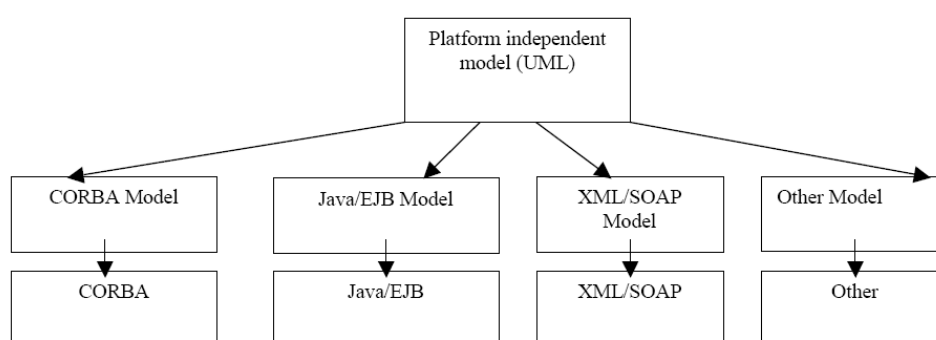
- Τέλος, τα μοντέλα του μεσαίου επιπέδου (PIM) προσδιορίζονται και από τεχνολογικής σκοπιάς στο κατώτερο επίπεδο της αρχιτεκτονικής μέσω του *Platform Specific Model (PSM)* ή *Μοντέλου Συγκεκριμένης Πλατφόρμας*. Σε αυτό το στάδιο το μοντέλο PIM μετατρέπεται σε μοντέλο εξαρτώμενο από την πλατφόρμα υλοποίησης. Η μετατροπή αυτή είναι δυνατόν να γίνει αυτόματα με τη χρήση των εργαλείων κατά τη φάση υλοποίησης της MDA, επίσης γνωστή και ως model-driven development (MDD), και έτσι είναι δυνατή η αυτόματη παραγωγή κώδικα για πλατφόρμες όπως Web Services, .NET, CORBA, J2EE.



Εικόνα 8- Αντιστοιχίσεις ανάμεσα στα μοντέλα PIM και PSM

Στην Εικόνα 8 παρουσιάζεται ένα παράδειγμα μετατροπής ενός μοντέλου PIM σε διάφορα μοντέλα PSM. Συγκεκριμένα, στο PIM δίνονται λεπτομέρειες που αφορούν στον τομέα εφαρμογής και ειδικότερα κλάσεις και ιδιότητες ενός πελάτη και ενός λογαριασμού, χωρίς ωστόσο να καθορίζονται στοιχεία αναπαράστασης σχετικά με την τεχνολογική πλατφόρμα που χρησιμοποιείται. Έτσι στην εικόνα 8 δίνονται οι εκτελούμενες μετατροπές ανάμεσα στα δυο μοντέλα (PIM και PSM) καθώς και τα πρότυπα που χρησιμοποιούνται για τις μετατροπές αυτές όπως παραδείγματος χάριν διακρίνεται η μετατροπή των UML αναπαραστάσεων σε XMI με χρήση των προτύπων XML Schema Definitions (XSD) or Document Type Definitions (DTD). Στη συνέχεια με τη βοήθεια εργαλείων είναι εύκολη η αναπαράσταση των UML κλάσεων από διαπροσωπείες σε κώδικα Java.

Συνήθως οι μετατροπές αυτές σε εκτελέσιμο κώδικα καθορίζονται από ένα σύνολο κανόνων το οποίο μπορεί να οριστεί και ως template.



**Εικόνα 9-Μετατροπή από μοντέλο UML σε συγκεκριμένα μοντέλα και πλατφόρμες**

Για την πληρέστερη κατανόηση της MDA θα μπορούσε να δοθεί ένα παράδειγμα[52] το οποίο παρουσιάζει τη διαδικασία δημιουργίας ενός server σε πλατφόρμα Java.

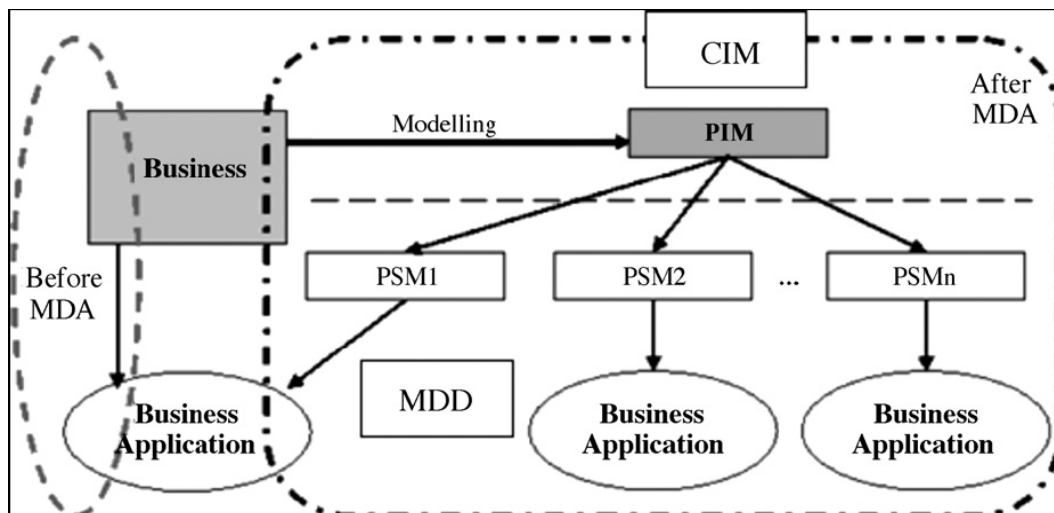
Σύμφωνα λοιπόν με τις αρχές της model driven αρχιτεκτονικής, σε πρώτο στάδιο δημιουργείται το PIM μοντέλο χρησιμοποιώντας τη UML. Αυτό μπορεί να περιλαμβάνει διαγράμματα κλάσεων, αντικειμένων, ακολουθίας κ.ο.κ.

Σε δεύτερο στάδιο δημιουργούνται τα μοντέλα των πρόσθετων υπηρεσιών που περιλαμβάνουν τις υπηρεσίες καταλόγου, συναλλαγών, χειρισμού συμβάντων και ασφαλείας (όλα αυτά αποδίδονται με τον όρο “*Pervasive Services*”). Επίσης δομείται

το μοντέλο λειτουργιών του τομέα. Ο κώδικας που παράγεται μετέπειτα καλεί τις υπηρεσίες της συγκεκριμένης πλατφόρμας υλοποίησης.

Στο επόμενο στάδιο, χρησιμοποιώντας ένα εργαλείο αντιστοίχισης μπορεί να επιλεγεί η επιθυμητή πλατφόρμα μεσολογισμικού, όπως εδώ η Java, και να προκύψει το PSM μοντέλο από το PIM. Το τελευταίο μοντέλο αποθηκεύεται επιπλέον σε MOF. Τέλος το εργαλείο MDA δημιουργεί όλα τα απαραίτητα αρχεία Java για τον server προκειμένου αυτός να τρέξει πάνω στη συγκεκριμένη πλατφόρμα. Επίσης πρέπει να επιλεγεί η γλώσσα προγραμματισμού εφόσον ο εξυπηρετητής εφαρμογών υποστηρίζει πολλές γλώσσες προγραμματισμού. Το εργαλείο θα ελέγχει τις πληροφορίες που δίνονται από την UML αναπαράσταση με σκοπό να δημιουργήσει τα απαραίτητα αρχεία με τα οποία ο server θα λειτουργήσει σωστά.

Παρακάτω δίνεται μια γενική αναπαράσταση των μετατροπών που εκτελούνται βάσει της Model driven αρχιτεκτονικής σε αντιδιαστολή προς το συμβατικό σενάριο αρχιτεκτονικής των IT συστημάτων.



Εικόνα 10-MDA vs Συμβατικό σενάριο

Συνοψίζοντας, η MDA δίνει τη δυνατότητα ανάπτυξης και χρήσης μοντέλων που αναπαριστούν περίπλοκες επιχειρηματικές ιδέες, μοντελοποιούν διαφορετικά στοιχεία τα οποία είναι απαραίτητα για την πραγματοποίηση της επικοινωνίας και των εφαρμογών των συστημάτων. Επιπλέον παρέχει το πλεονέκτημα της ανεξαρτησίας από την τεχνολογική πλατφόρμα (λειτουργικά συστήματα, τοπολογία των μονάδων hardware ή τεχνολογική υποδομή) και μεταθέτει την προσοχή στην

αναγκαιότητα των μοντέλων σε διάφορα αφαιρετικά επίπεδα. Παράλληλα, δίνεται η δυνατότητα καλύτερης και πληρέστερης εξέτασης των μετατροπών μεταξύ των διαφόρων μοντέλων οι οποίες μετατροπές μπορεί να απαιτούν γνώσεις των επιχειρηματικών διαδικασιών και των εφαρμοζόμενων τεχνολογιών. Έτσι από την λεπτομερή εξέταση των διαδικασιών μετατροπής επιτυγχάνονται καλύτερα επίπεδα ποιότητας και αποτελεσματικότητας και ενισχύεται η ιδέα της αυτοματοποίησης αλλά και της επαναχρησιμοποίησης για τις διάφορες εφαρμογές.

## **4.2.2 Αρχιτεκτονική Προσανατολισμένη σε Υπηρεσίες**

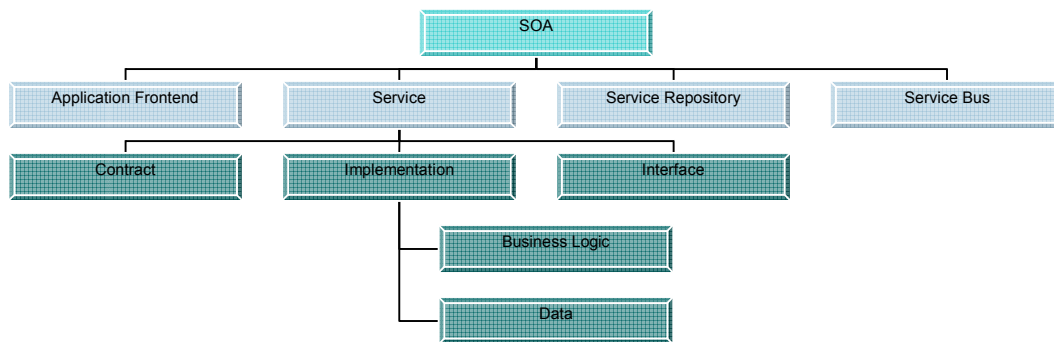
Οι μοντέρνες επιχειρήσεις, σε μια προσπάθεια να ανταπεξέλθουν στις δύσκολες ανταγωνιστικές συνθήκες της παγκόσμιας αγοράς, στοχεύουν να προσδώσουν στις επιχειρηματικές διαδικασίες τους χαρακτηριστικά όπως ευελιξία, αμεσότητα, αυτονομία και αποτελεσματικότητα. Η Αρχιτεκτονική Προσανατολισμένη σε Υπηρεσίες, Service Oriented Architecture (SOA), καλείται να υλοποιήσει τις επιδιώξεις αυτές των επιχειρήσεων δίνοντας έμφαση στις υπηρεσίες (services). Αυτές μπορούν να δομήσουν ένα σύνολο διαδικασιών (processes) οι οποίες συγκεντρώνουν τα χαρακτηριστικά της αυτονομίας και της χαλαρής σύζευξης (loose coupling). Έχουν ως απώτερο στόχο να επιλύσουν προβλήματα ενοποίησης (integration) εφαρμογών, διαχείρισης συναλλαγών, θέματα ασφάλειας ενώ παράλληλα επιδιώκουν την συνεργασία διαφορετικών legacy συστημάτων και την επικοινωνία αδιακρίτως πρωτοκόλλων και τεχνολογικών υποδομών. Έτσι θα υπάρξει και η δυνατότητα άμεσης σύνθεσης υπηρεσιών οι οποίες μπορούν να επαναχρησιμοποιηθούν καλύπτοντας τις μεταβαλλόμενες ανάγκες των επιχειρήσεων.

Σύμφωνα με τις αρχές της SOA, οι υπηρεσίες αποτελούν λειτουργικές μονάδες που είναι καλά ορισμένες και σημασιολογικά μη διαφορούμενες χρησιμεύοντας έτσι στην άμεση και κατ' απαίτηση διαλειτουργικότητα και ενοποίηση legacy συστημάτων. Επιπλέον οι βασικές αρχές που χαρακτηρίζουν τις υπηρεσίες είναι αναφορικά οι εξής:

- ✓ είναι επαναχρησιμοποιήσιμες
- ✓ μοιράζονται ένα επίσημο συμβόλαιο το οποίο περιγράφει την υπηρεσία και τους όρους της ανταλλαγής πληροφοριών

- ✓ είναι χαλαρά συζευγμένες (loosely coupled) εφόσον είναι σχεδιασμένες ώστε να μην υπάρχουν αλληλεξαρτήσεις
- ✓ αποκρύπτουν το λογικό υπόβαθρο
- ✓ μπορούν να συνθέσουν άλλες υπηρεσίες ενισχύοντας έτσι το χαρακτηριστικό της επαναχρησιμοποίησης και δημιουργώντας αφαιρετικά επίπεδα
- ✓ είναι αυτόνομες υπό την έννοια ότι ορίζουν οι ίδιες τη λογική των λειτουργιών τους
- ✓ είναι stateless δηλαδή δεν έχουν ανάγκη να χειρίζονται πληροφορίες κατάστασης παραμένοντας έτσι χαλαρά συζευγμένες
- ✓ μπορούν να εντοπιστούν από τους ενδιαφερόμενους οι οποίοι μπορούν να χρησιμοποιήσουν τη λογική τους

Στην Αρχιτεκτονική Προσανατολισμένη σε Υπηρεσίες μπορούμε να διακρίνουμε τέσσερις βασικές έννοιες: το εξωτερικό κομμάτι της εφαρμογής (application front-end), η υπηρεσία, την αποθήκη υπηρεσιών (service repository) και τον δίαυλο υπηρεσιών (service bus). Οι υπηρεσίες στη SOA αποτελούνται από το κομμάτι της υλοποίησης που εκφράζει την επιχειρηματική λογική και τα δεδομένα, από το κομμάτι τους συμβολαίου της υπηρεσίας το οποίο καθορίζει τη λειτουργικότητα, τη χρήση και τους περιορισμούς για τον πελάτη, και από τη διαπροσωπεία που αναπαριστά φυσικά την λειτουργικότητα της υπηρεσίας. Η αποθήκη υπηρεσιών αποθηκεύει τα συμβόλαια των υπηρεσιών και ο δίαυλος υπηρεσιών διασυνδέει το εξωτερικό κομμάτι της εφαρμογής με τις υπηρεσίες. Οι υπηρεσίες αυτές μπορούν να κληθούν από οποιοδήποτε χρήστη ανεξαρτήτως πλατφόρμας, λειτουργικού συστήματος και γλώσσας προγραμματισμού, και να αποτελέσουν μέρος μιας σύνθετης επιχειρηματικής διαδικασίας κατά την οποία η μια υπηρεσία επικοινωνεί με τις άλλες μέσω των διαπροσωπειών. Οι υπηρεσίες αυτές μπορεί να είναι καινούριες ή να προέρχονται από παλαιότερες εφαρμογές. Αναμφισβήτητα πάντως οι ενδιαφερόμενοι καταναλωτές των υπηρεσιών έχουν την δυνατότητα να συνθέσουν νέες και παλαιότερες υπηρεσίες σε νέες επιχειρηματικές λειτουργίες.



Εικόνα 11- Επίπεδα αρχιτεκτονικής προσανατολιζόμενης σε υπηρεσίες (SOA)

Στη συνέχεια δίνονται εν συντομία ορισμένες λεπτομέρειες σχετικά με τα παραπάνω στοιχεία της αρχιτεκτονικής SOA.

- **Εξωτερικό κομμάτι της εφαρμογής -Application front-end**

Πρόκειται για τη μονάδα που ξεκινά μια επιχειρηματική διαδικασία και ολοκληρώνει την εκτέλεση συλλέγοντας τα αποτελέσματα. Υπάρχουν διαφορετικοί τύποι front-end όπως αυτοί που έρχονται σε άμεση επαφή με τον χρήστη ή τα προγράμματα που καλούν τις λειτουργίες περιοδικά ή λόγω κάποιου συμβάντος.

- **Υπηρεσία-Service**

Η υπηρεσία περιλαμβάνει:

α) το συμβόλαιο (contract): προσδιορίζει τον σκοπό, τη λειτουργικότητα, τους περιορισμούς και την χρήση της υπηρεσίας. Επίσης μη υποχρεωτικά μπορεί να ορίζει τη διαπροσωπεία της υπηρεσίας σε IDL ή WSDL. Επίσης μπορεί να παρέχει σημασιολογία και παραμέτρους.

β) τη διαπροσωπεία (interface): περιγράφει την υλοποίηση της διαπροσωπείας της υπηρεσίας όπως παρουσιάζεται στους πελάτες αυτής (ο ορισμός της διαπροσωπείας αναφέρεται στο συμβόλαιο).

γ) την υλοποίηση (implementation): παρέχει φυσικά τη προδιαγεγραμμένη επιχειρηματική λογική και τα δεδομένα της υπηρεσίας. Αποτελείται από προγράμματα, δεδομένα πιστοποίησης και βάσεις δεδομένων.

δ) την επιχειρηματική λογική (business logic): ενσωματώνεται στην υλοποίηση και διατίθεται μέσω της διαπροσωπείας.

ε) τα δεδομένα (data): μπορεί να περιλαμβάνονται σε μια υπηρεσία. Ειδικά εάν αυτή ήταν δεδομενο-κεντρική.

- **Αποθήκη Υπηρεσιών-Service Repository**

Παρέχει πληροφορίες που είναι χρήσιμες για μια υπηρεσία όπως φυσική τοποθεσία της υπηρεσίας, πληροφορίες για τον πάροχο της υπηρεσίας, τεχνικούς περιορισμούς, κόστη χρήσης, θέματα ασφάλειας και διαθέσιμα επίπεδα υπηρεσιών. Επιπλέον εάν γίνεται λόγος για αποθήκες υπηρεσιών που βρίσκονται έξω από τα όρια μιας επιχείρησης και προορίζονται για διαλειτουργικότητα πολλών επιχειρήσεων, τότε περιέχονται πληροφορίες σχετικά με νομικά ζητήματα, θέματα παρουσίασης, ασφάλειας, εγγραφής χρήστη κλπ. Η υλοποίηση τους είναι απλή όπως για παράδειγμα μια ιδιωτική βάση δεδομένων.

- **Δίαυλος Υπηρεσιών-Service Bus**

Ο δίαυλος υπηρεσιών διασυνδέει application front-ends των εφαρμογών των πελατών με τις υπηρεσίες που θέλουν αυτές να χρησιμοποιήσουν. Διαφέρει ωστόσο από τις πλατφόρμες λογισμικού όπως είναι η CORBA καθώς δεν εξυπηρετεί μια συγκεκριμένη τεχνολογία αλλά προσφέρει ετερογένεια υποστηριζόμενων τεχνολογιών. Επίσης χαρακτηριστικά του service bus είναι η δυνατότητα διασύνδεσης που παρέχει στους πελάτες, η ετερογένεια υποστηριζόμενων τρόπων επικοινωνίας και κάποιες πρόσθετες υπηρεσίες ασφαλείας, μεταφοράς μηνυμάτων, συναλλαγών και ελέγχου.

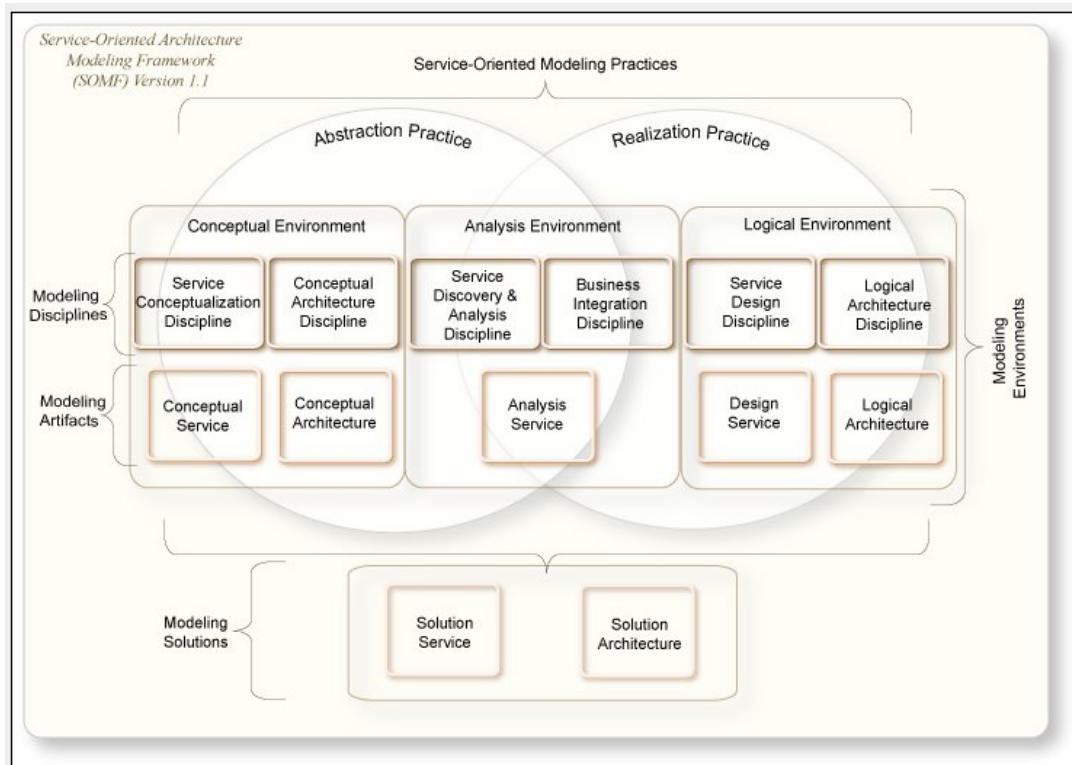
Σύγχρονες εφαρμογές της αρχιτεκτονικής παρουσιάζονται σε ένα ευρύ σύνολο τεχνολογιών όπως : SOAP, REST, RPC, DCOM, CORBA, Web Services, WCF (η υλοποίηση της Microsoft για τα Web Services είναι ένα μέρος του WCF). Η SOA μπορεί να εφαρμοστεί χρησιμοποιώντας ένα ή περισσότερα από αυτά τα πρωτόκολλα. Οι υπηρεσίες είναι εντελώς ανεξάρτητες και δε γνωρίζουν λεπτομέρειες των εφαρμογών που τις καλούν ενώ από την άλλη ούτε οι εφαρμογές γνωρίζουν τον τρόπο υλοποίησης των εργασιών των διαφόρων υπηρεσιών.

Επιπλέον μια ευρεία περιοχή εφαρμογής της SOA αρχιτεκτονικής είναι η δημιουργία εφαρμογών από τη σύνθεση χαλαρά συζευγμένων και διαλειτουργικών υπηρεσιών. Σε αυτά τα πλαίσια χρησιμοποιούνται τα συμβόλαια (contracts) που αναφέρθηκαν προηγουμένως, καθώς και η WSDL. Επίσης καθώς υπάρχει διαφάνεια

ως προς τις υποκείμενες τεχνολογικές πλατφόρμες (Java, .NET κοκ), οι διάφορες υπηρεσίες όπως πχ. κάποιες γραμμένες σε Java και εκτελέσιμες σε Java EE ή γραμμένες σε C# και εκτελέσιμες σε περιβάλλον .NET, μπορούν να ενοποιηθούν και να χρησιμοποιηθούν από τον ίδιο πελάτη. Εφαρμογές υλοποιημένες σε μια από τις παραπάνω πλατφόρμες μπορούν να καλέσουν και να χρησιμοποιήσουν υπηρεσίες από άλλες πλατφόρμες μέσω των Web Services.

Σε επίπεδο σύνθεσης των εφαρμογών, οι γλώσσες ανώτερου επιπέδου όπως πχ BPEL και τα πρότυπα WS-CDL[67] και WS-Coordination[64] χρησιμοποιούνται για την σύνθεση με προσέγγιση ενορχήστρωσης των υπηρεσιών σε επιχειρηματικές υπηρεσίες και κατόπιν την ενσωμάτωσή τους σε ροές εργασιών (workflows) και επιχειρηματικές διαδικασίες σύνθετων εφαρμογών.

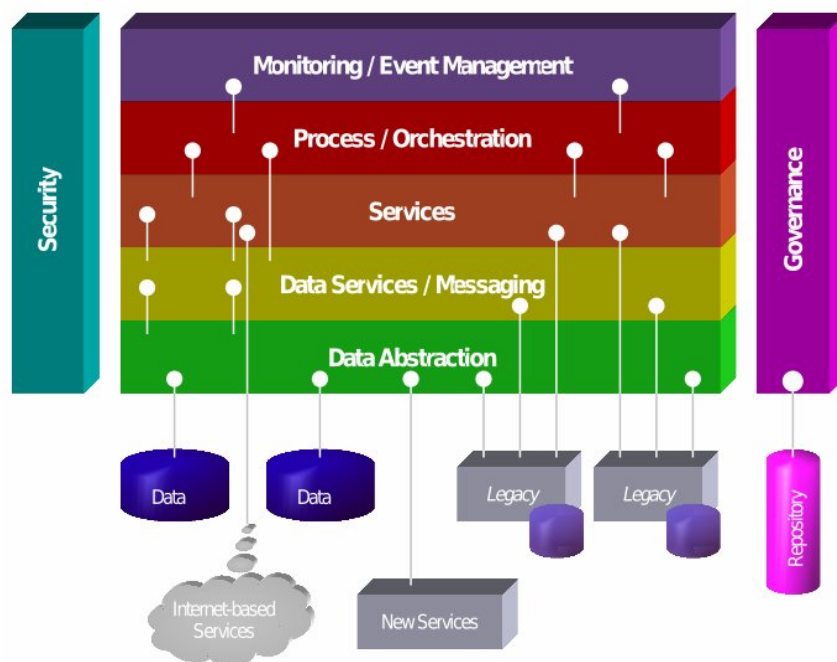
Σε πιο αφαιρετικό επίπεδο, η εφαρμογή της SOA για την πραγματοποίηση της διαλειτουργικότητας και της επαναχρησιμοποίησης μπορεί να επιτευχθεί μέσω της μοντελοποίησης των διαφόρων πρακτικών. Το Service-Oriented Modeling Framework (SOMF) στηρίζεται στο Service-Oriented Modeling που παρέχει τις βάσεις για την διαμόρφωση της ιδέας, την ανάλυση, τον σχεδιασμό και την δόμηση των service-oriented πόρων. Πρόκειται ουσιαστικά για ένα περιβάλλον παροχής μοντέλων ανάπτυξης service-oriented εφαρμογών.



Εικόνα 12-Service-Oriented Modeling Framework (SOMF) Version 1.1



Τέλος αναφερόμενοι σε θέματα ασφάλειας κατά τη μεταφορά των δεδομένων, η SOA βασίζεται σε πρότυπα ασφαλείας τα οποία περιλαμβάνονται στο πρότυπο WS-Security[70]. Ουσιαστικά σύμφωνα με τα πρότυπα αυτά, η λογική της ασφάλειας ενσωματώνεται μέσα στο επίπεδο της ανταλλαγής μηνυμάτων.



Εικόνα 13- SOA Meta Model, The Linthicum Group, 2007

Συνοψίζοντας, θα μπορούσε να δοθεί ένας γενικός ορισμός της SOA αρχιτεκτονικής έτσι όπως δίνεται από τον οργανισμό OASIS[40]: *είναι ένα παράδειγμα οργάνωσης και χρήσης των κατακεμημένων δυνατοτήτων που μπορεί να βρίσκονται 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.*”). Σε ένα περιβάλλον SOA για την χρήση των αυτόνομων υπηρεσιών δεν είναι απαραίτητη η γνώση της τεχνολογικής πλατφόρμας. Οι υπηρεσίες στη SOA αρχιτεκτονική εκθέτουν τη λειτουργικότητά τους μέσω των διαπροσωπειών τους και αυτές χρησιμοποιούνται για να διευκρινιστεί πώς λειτουργεί καθεμιά υπηρεσία και να επιλεγεί η κατάλληλη.

### 4.2.3 Αρχιτεκτονική Ψηφίδων

Με σκοπό τη διευκόλυνση, την ταχύτερη ανάπτυξη των εφαρμογών των συστημάτων και την ενοποίησή τους ξεπερνώντας τα όρια των οργανισμών, τίθεται το ζήτημα της επαναχρησιμοποίησης των διαφόρων ψηφίδων λογισμικού (software components) από εφαρμογές και περιβάλλοντα. Έτσι δημιουργείται η προοπτική μείωσης του χρόνου και του κόστους της ανάπτυξης των εφαρμογών, ωστόσο γεννάται παράλληλα το πρόβλημα της μη δυνατής πάντα ενσωμάτωσής τους σε οποιαδήποτε εφαρμογή. Η επίτευξη της διαλειτουργικότητας επίσης εμποδίζεται από την χρήση μεσολογισμικού που είναι άμεσα εξαρτώμενο από τον προμηθευτή και από την ανάπτυξη προτυποποιημένου και ευρέως εφαρμοζόμενου κώδικα, κάνοντας τη διαδικασία ενοποίησης των ετερογενών μονάδων ιδιαίτερα δαπανηρή και χρονοβόρα.

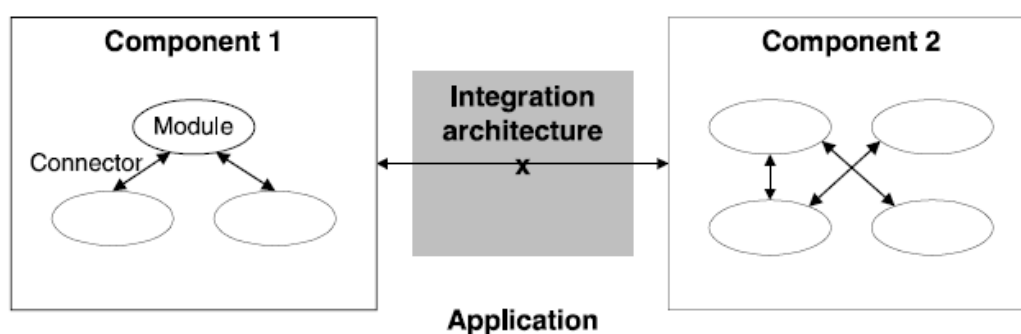
Προκειμένου να γίνει επιτυχής επιλογή του μεσολογισμικού είναι απαραίτητο να εξεταστούν τα προβλήματα διαλειτουργικότητας που εμφανίζονται στη χρήση των μονάδων λογισμικού. Ειδικότερα, είναι σημαντικό να αντιμετωπιστούν οι εξής προκλήσεις[10]:

1. Συγκέντρωση των χαρακτηριστικών σχετικών ιδιοτήτων των μονάδων που συμμετέχουν και των απαιτήσεων της εφαρμογής.
2. Οργάνωση των βασικών ιδιοτήτων σε μια ομοιόμορφη αναπαράσταση.
3. Ανάπτυξη αλγορίθμων ανάλυσης των ιδιοτήτων για τον εντοπισμό τύπων που δημιουργούν συγκρούσεις στη διαλειτουργία.
4. Ορισμός τεχνικών που μπορούν να εκφράσουν πιθανές υποδομές μεσολογισμικού.

Σε ευρύτερα πλαίσια αξίζει να αναφερθεί ότι η αρχιτεκτονική λογισμικού (software architecture) προδιαγράφει σε ανώτερο επίπεδο τα υπολογιστικά στοιχεία μέσω των οποίων πραγματοποιείται η αλληλεπίδραση καθώς και τους περιορισμούς που την συνοδεύουν.

Στο σημείο αυτό μπορεί να οριστεί και η αρχιτεκτονική ενοποίησης (integration architecture) η οποία είναι μια περιγραφή αρχιτεκτονικής λογισμικού που στοχεύει στην επίλυση των προβλημάτων διαλειτουργικότητας μεταξύ δυο τουλάχιστον μονάδων.

Μια ενδεικτική ορολογία που μπορεί να περιγράψει μια αρχιτεκτονική λογισμικού με τη συμμετοχή ψηφίδων φαίνεται στην εικόνα που ακολουθεί. Σύμφωνα με αυτή, μια ψηφίδα (component) είναι ένα σύστημα ή υποσύστημα που συμμετέχει. Μια δομική μονάδα (module) είναι ένα εσωτερικό υπολογιστικό στοιχείο μιας ψηφίδας ενώ ένας connector διασυνδέει δυο ψηφίδες και η εφαρμογή είναι το ενοποιημένο σύστημα των ψηφίδων.



Εικόνα 14-Αρχιτεκτονική ενοποιημένου συστήματος

Στη συνέχεια αναλύονται με περισσότερη λεπτομέρεια οι *connectors*, ή γενικότερα οι οντότητες που περιγράφουν αλληλεπιδράσεις μεταξύ των μονάδων. Περιγράφοντας τους τύπους των connectors καθίσταται δυνατή η επιλογή των κατάλληλων σχημάτων αλληλεπίδρασης αλλά επίσης χρήσιμο είναι να υπάρχει η δυνατότητα ορισμού connectors σε αρχιτεκτονικό επίπεδο και καθορισμού νέων τύπων αυτών(χρησιμοποιώντας τις γλώσσες ορισμού αρχιτεκτονικών- Architecture definition languages (ADL)).

Οι connectors εξυπηρετούν στην απλή μεταφορά δεδομένων και ελέγχου ανάμεσα στις μονάδες ενώ σε περίπτωση προβλημάτων διαλειτουργικότητας είναι απαραίτητοι πιο πολύπλοκοι connectors για να δομήσουν την υποδομή του μεσολογισμικού. Οι τελευταίοι connectors ονομάζονται στοιχεία ενοποίησης (integration elements) και κατατάσσονται στις εξής κατηγορίες:

- Translator: μετατρέπει δεδομένα και συναρτήσεις μεταξύ προτύπων των μονάδων συστήματος και υλοποιεί σημασιολογικές μετατροπές.

- Controller: συντονίζει και μεσολαβεί κατά την μεταφορά πληροφοριών μεταξύ των μονάδων χρησιμοποιώντας προκαθορισμένες διαδικασίες εξαγωγής αποφάσεων.
- Extender: προσθέτει νέα χαρακτηριστικά και λειτουργικότητα στα συστήματα των μονάδων με σκοπό να αναπτύξουν συμπεριφορά κατάλληλη για ενοποίηση. Χρησιμοποιείται σε περιπτώσεις που οι άλλοι δυο τύποι δε μπορούν να προσφέρουν πλήρη λειτουργικότητα πχ. έλεγχος ασφαλείας, buffering κλπ. [28].

Θέτοντας ως στόχο την επίτευξη διαλειτουργικότητας μεταξύ των ψηφίδων λογισμικού είναι απαραίτητη η σύγκριση ορισμένων χαρακτηριστικών. Ωστόσο λόγω των διαφόρων προβλημάτων που εμφανίζονται όπως πολλαπλότητα των χαρακτηριστικών, ύπαρξη άγνωστων τιμών χαρακτηριστικών, αφαιρετικότητα, προβλήματα πλεονασμού, μη πληρότητα των ορισμών ή ασυνέπεια των ορισμών, κρίνεται σημαντική η συστηματοποιημένη ομαδοποίηση και συσχέτιση των χαρακτηριστικών. Ενδεικτικά αναφέρονται δυο μέθοδοι που στηρίζονται: σε επίπεδα αφαιρετικότητας και σημασιολογικά δίκτυα. Τα χαρακτηριστικά αυτά μπορούν γενικά να οργανωθούν σε τρεις βασικές κατηγορίες: συστήματος, δεδομένων και ελέγχου. Στην πρώτη κατηγορία εντάσσονται τα χαρακτηριστικά που έχουν να κάνουν με το σύνολο του συστήματος και πώς αυτά τα χαρακτηριστικά διαμορφώνουν την ψηφίδα. Η κατηγορία των δεδομένων υποδεικνύει τον τρόπο με τον οποίο τα δεδομένα εντάσσονται και μεταφέρονται μέσα στην ψηφίδα ενώ τέλος τα χαρακτηριστικά ελέγχου σχετίζονται σε θέματα ελέγχου.

Ακολουθώντας τη μέθοδο κατηγοριοποίησης βάσει αφαιρετικών επιπέδων, προτείνονται τρία επίπεδα τα οποία αντιπροσωπεύουν το σημείο στο οποίο θα αποδοθεί ένα αρχιτεκτονικό χαρακτηριστικό κατά τη φάση σχεδιασμού [26].

Τα επίπεδα αυτά είναι:

- *Επίπεδο προσανατολισμού (Orientation level)*: τα χαρακτηριστικά σε αυτό το επίπεδο σχετίζονται με την αρχιτεκτονική της ψηφίδας σε υψηλό επίπεδο και δεν προσδιορίζουν λεπτομέρειες εκτέλεσης. Μπορούν να προσδιοριστούν από αναφορές του δημιουργού και διαμορφώνουν τη συνολική εικόνα της

ψηφίδας. Είναι δε εύκολο να προσδιοριστούν οι τιμές αυτών των χαρακτηριστικών.

- *Επίπεδο περιθωρίου δράσης (Latitude level)*: προσδιορίζουν πού και πώς πραγματοποιείται η επικοινωνία στο σύστημα ενώ για να προσδιοριστούν κάποιες τιμές των χαρακτηριστικών πιθανώς να χρειαστεί λεπτομερέστερη εξέταση του σχεδιασμού του συστήματος.
- *Επίπεδο εκτέλεσης (Execution level)*: τα χαρακτηριστικά σε αυτό το επίπεδο παρέχουν λεπτομέρειες σχετικά με την εκτέλεση. Οι τιμές τους καθορίζουν μέρη από την υλοποίηση του συστήματος επιτρέποντας την ανάλυση των δομών δεδομένων και των στρατηγικών επικοινωνίας. Οι τιμές αυτών των χαρακτηριστικών είναι ωστόσο δύσκολο να προσδιοριστούν εκτός και αν υπάρχει κατανοητός πηγαίος κώδικας. Παραδείγματα χαρακτηριστικών ελέγχου που είναι προσδιορισμένα σε επίπεδο εκτέλεσης είναι ο χρόνος σύνδεσης (binding), η μέθοδος επικοινωνίας και η συγχρονικότητα.

Σε επόμενο στάδιο τώρα, και βασιζόμενοι στην κατηγοριοποίηση των χαρακτηριστικών σε αφαιρετικά επίπεδα, προτείνεται ο καθορισμός των σχέσεων ανάμεσα στα χαρακτηριστικά. Για να επιτευχθεί η διασύνδεση των χαρακτηριστικών χρειάζεται να προσδιοριστούν οι σχέσεις στο εσωτερικό κάθε επιπέδου αλλά και ανάμεσα σε αυτά, επίσης να εξακριβωθούν ομοιόμορφες και συγκριτικές σχέσεις μεταξύ των χαρακτηριστικών αλλά και να προσδιοριστεί ένα ελάχιστο σύνολο αντιπροσωπευτικών χαρακτηριστικών ώστε να προσφερθεί μια πρώτη ανάλυση. Για την πραγματοποίηση αυτών των στόχων εφαρμόζονται τα σημασιολογικά δίκτυα βασιζόμενα στους ορισμούς των χαρακτηριστικών, τα αφαιρετικά επίπεδα και σε μελέτες περίπτωσης που αναπαριστούν τη σημασία στην κατανόηση της αρχιτεκτονικής [10].

Εν κατακλείδι, η παραπάνω ανάλυση στηρίχθηκε στην ανάγκη απομόνωσης των αρχιτεκτονικών χαρακτηριστικών που προσδιορίζουν τα θέματα διαλειτουργικότητας στις αρχιτεκτονικές ψηφίδων. Απαραίτητος είναι ο έγκαιρος προσδιορισμός των λεπτομερειών και η οργάνωση των πληροφοριών που αφορούν στα δεδομένα και τον έλεγχο των συστημάτων. Η παρούσα ανάλυση αποτελεί μέρος της διαδικασίας πριν την εφαρμογή της ενοποίησης και βοηθά στην αρχική επιλογή

του μεσολογισμικού. Με βάση τις κατηγοριοποιήσεις που παρουσιάστηκαν επιδιώκεται ο χειρισμός των διαφόρων χαρακτηριστικών με τέτοιο τρόπο ώστε να διατηρηθεί ένα υπόβαθρο και σε μετέπειτα ανάπτυξη ή επέκταση των εφαρμογών λογισμικού. Τέλος να αναφερθεί ότι υπάρχει και η διαδικασία μετά-ενοποίησης η οποία κυρίως ασχολείται με την αντιμετώπιση των προβλημάτων που τυχόν δημιουργούνται κατά την εφαρμογή αφαιρετικών αρχιτεκτονικών. Η τελευταία ωστόσο δεν αποτέλεσε στοιχείο περαιτέρω ανάπτυξης.

## **4.3 Εφαρμογές διαλειτουργικότητας**

### **4.3.1 Μεσίτες Ενοποίησης Επιχειρηματικών Εφαρμογών (EAI Brokers)**

Οι Μεσίτες Ενοποίησης Επιχειρηματικών Εφαρμογών ή Enterprise Application Integration (EAI) Brokers αποτελούν την περισσότερο διαδεδομένη λύση στο ζήτημα της ενοποίησης επιχειρηματικών διαδικασιών οι οποίες υποστηρίζονται από τα εμπορικά συστήματα των επιχειρήσεων. Προσφέρουν υπηρεσίες ενοποίησης τόσο μεταξύ απλών εφαρμογών αλλά επίσης συνθέτουν πλατφόρμες διαλειτουργικότητας προσφέροντας υπηρεσίες συναλλαγών, ασφάλειας, διαχείρισης πόρων. Μερικά παραδείγματα αυτής της τεχνολογίας αποτελούν οι IBM's WebSphere Integration Broker, PeopleSoft's AppConnect, και Sun ONE Integration Server.

Οι EAI Brokers διευκολύνουν την ανταλλαγή πληροφοριών μεταξύ δυο ή περισσότερων συστημάτων με ασύγχρονο και οδηγούμενο από τα γεγονότα τρόπο ενώ ταυτόχρονα είναι υπεύθυνοι για την ενοποίηση σε επίπεδο σημασιολογίας (semantics) και διαφορετικών τεχνολογικών πλατφορμών. Για τη διασύνδεση των διαφόρων legacy συστημάτων, των ERP ή CRM συστημάτων και χρησιμοποιούνται προσαρμογείς (adapters, βλ. §4.3.3) οι οποίοι στη συνέχεια συνδέονται επάνω στον μεσίτη ενοποίησης (integration broker). Ουσιαστικά οι προσαρμογείς (adapters) μεσολαβούν ανάμεσα στη διεπαφή των εφαρμογών και στο επίπεδο ανταλλαγής των μηνυμάτων.

Εν ολίγοις, πρόκειται για ένα μεσολογισμικό (middleware) το οποίο σε νεότερες εκδόσεις, υποστηρίζει την ανταλλαγή μηνυμάτων, την ενοποίηση των διεργασιών, τη διαχείριση των επιχειρηματικών διεργασιών, τη διαχείριση των εμπορικών συναλλαγών αλλά και την εφαρμογή επιχειρηματικών κανόνων. Περιλαμβάνει προσαρμογείς και εφαρμογές αλληλεπίδρασης των χρηστών μέσα από front-end διαδικασίες όπως JSP σελίδες.

### **4.3.2 Εξυπηρετητές Εφαρμογών (Application Servers)**

Κύριο χαρακτηριστικό των *application servers* ή *εξυπηρετητών εφαρμογών* είναι η ιδιότητάς τους να διαχωρίζουν την επιχειρηματική λογική από το επίπεδο της παρουσίασης και ο συντονισμός των διασυνδέσεων πόρων. Ανάμεσα στα χαρακτηριστικά ενός εξυπηρετητή εφαρμογών (application server) συγκαταλέγονται: ασφαλές περιβάλλον εκτέλεσης των συναλλαγών, κατάτμηση του επιπέδου εφαρμογής σε πολλούς εξυπηρετητές, διαχείριση ανάκαμψης σε περίπτωση αποτυχίας κάποιου εξυπηρετητή και καταμερισμό φορτίου (load balancing). Επίσης προσφέρουν συνδεσιμότητα σε εφαρμογές και πρόσβαση σε δεδομένα και διαδικασίες διαφόρων Πληροφοριακών Συστημάτων. Μερικά παραδείγματα τέτοιων εξυπηρετητών αποτελούν οι: SUN's J2EE , IBM's WebSphere και Microsoft's .NET. Ωστόσο πρέπει να τονιστεί ότι οι εξυπηρετητές εφαρμογών είναι ένα σύνολο από υπηρεσίες που προσφέρουν παρουσίαση, εκτέλεση και διαχείριση της επιχειρηματικής λογικής των διαδικτυακών εφαρμογών. Προσφέρουν τη δυνατότητα εκτέλεσης διαδικτυακών συναλλαγών παρουσιάζοντας δεδομένα και εφαρμογές που βρίσκονται στα διάφορα συστήματα μέσω μια μοναδικής διαπροσωπείας. Οι εξυπηρετητές εφαρμογών δεν είναι ικανοί να πετύχουν την απευθείας ενοποίηση των συστημάτων.

Εντούτοις, οι εξυπηρετητές αυτής της κατηγορίας μπορούν να προσφέρουν ενοποίηση διαφόρων μονάδων λογισμικού και για αυτό θεωρούνται ως υπόβαθρο της ενοποίησης των συστημάτων. Συγκεκριμένα, οι διάφορες μονάδες συστημάτων πχ. ERP, CRM κοκ., μπορούν να συνδεθούν με wrappers οι οποίοι αντιπροσωπεύουν ένα στρώμα λογισμικού που περιλαμβάνει τα δεδομένα και την επιχειρηματική λογική

και ορίζει τις υπηρεσίες του στην αντίστοιχη διεπαφή (API). Το API μεσολαβεί ανάμεσα στις κλήσεις των εφαρμογών πελατών, τα υποκείμενα δεδομένα και τον κώδικα, μετατρέποντας τις εισερχόμενες κλήσεις σε μορφή κατανοητή από τα τελευταία. Η εκτέλεση λαμβάνει χώρα ανάμεσα στους wrappers του εξυπηρετητή εφαρμογών ενώ αυτοί αναπαριστούν τα δεδομένα και τα μηνύματα των υποκείμενων συστημάτων σε μια μορφή αποδεκτή από τον εξυπηρετητή εφαρμογών (application server) υλοποιώντας ουσιαστικά μια αρχιτεκτονική facade.

Γενικότερα οι εξυπηρετητές εφαρμογών προσφέρουν επικοινωνία σε σύγχρονο και ασύγχρονο τρόπο και υπηρεσίες που χαρακτηρίζονται από αξιοπιστία, δυνατότητα επέκτασης, διαθεσιμότητα και αυτοματοποίησης των εργασιών ανάπτυξης εφαρμογών.

### 4.3.3 Προσαρμογείς (Adapters)

Επιδιώκοντας να διευκρινιστεί ο ρόλος που παίζουν οι *προσαρμογείς ή adapters*, αναφέρονται στη συνέχεια ορισμένα χαρακτηριστικά της λειτουργίας τους.

Καταρχάς, οι προσαρμογείς προσφέρουν συνδεσιμότητα, σημασιολογική αποσαφήνιση και υπηρεσίες μετάφρασης μεταξύ εφαρμογών και συνεργασιών[50]. Προσφέρουν υπηρεσίες μεταξύ ενός integration broker και μιας ψηφίδας (component) μια συγκεκριμένης εφαρμογής αλλά και στην περίπτωση των προσαρμογέων πόρων (resource adapters), παρέχουν μετάφραση των μηνυμάτων εφαρμογής από και προς ένα σύνολο προτύπων δεδομένων και πρωτοκόλλων επικοινωνίας. Συγκεκριμένα, τα μηνύματα που στέλνονται από και προς μια εφαρμογή μεταφράζονται από το σχήμα της πηγής σε ένα συγκεκριμένο πρότυπο και κατά τη λήψη από το πρότυπο στο σχήμα και το πρωτόκολλο που αναγνωρίζει η εφαρμογή.

Ένας προσαρμογέας υποστηρίζει μετατροπές τύπων δεδομένων από τη μια πηγή δεδομένων στην άλλη όπως πχ. ανάμεσα σε COBOL και XML, XML-to-XML λεξικά, IDL, ODL, συστήματα βάσεων δεδομένων κοκ. Επίσης παρέχει συναρτήσεις μετατροπών χρόνου και ημερομηνίας, EBCDIC/ASCII, συναρτήσεις δυαδικών ψηφίων και χαρακτήρων, EDI, και ένα σύνολο από συναρτήσεις διαχωρισμού/συνένωσης πεδίων δεδομένων. Μπορεί να υποστηρίζει XML parser και άλλα δημοφιλή πρωτόκολλα ηλεκτρονικού εμπορίου όπως: RosettaNet[61], ebXML[12], cXML και xCBL.



#### 4.3.4 Μεσολογισμικό Προσανατολισμένο στα Μηνύματα (Message Oriented Middleware)

Η ιδέα του *Μεσολογισμικού Προσανατολισμένου στα Μηνύματα*, ή *Message Oriented Middleware (MOM)*, είναι παρούσα και εξελίσσεται εδώ και δυο δεκαετίες περίπου. Πρόκειται για μια τεχνική που στόχο έχει την διασύνδεση εφαρμογών με χαρακτηριστικά χαλαρής σύζευξης και ασύγχρονης επικοινωνίας.

Για την επίτευξη της επικοινωνίας μεταξύ των εφαρμογών χρησιμοποιείται μια διεπαφή (API) ενώ η επικοινωνία αυτή πραγματοποιείται μέσω ενός συστήματος ανταλλαγής μηνυμάτων. Οι δράστες αυτού του συστήματος διακρίνονται σε παραγωγείς και καταναλωτές μηνυμάτων (producers και consumers) και η επικοινωνία τους λαμβάνει χώρα μέσω καναλιών που είναι γνωστά ως “δημοσίευσης-και-εγγραφής” (“publish-and-subscribe channels”) ή “καναλιών σημείου-με-σημείο” (“point-to-point channels”).

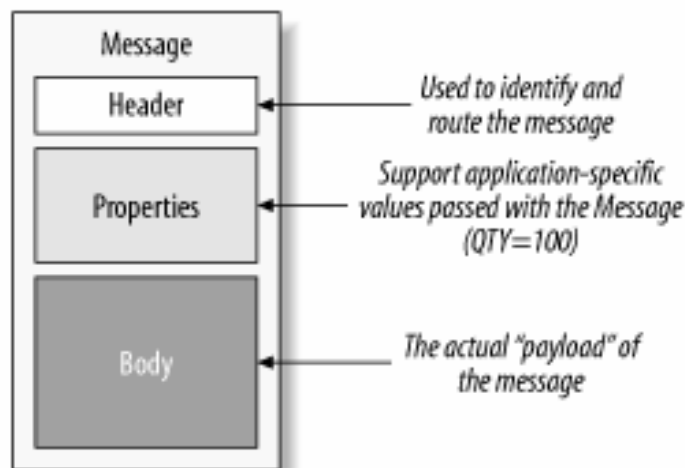
Αναφορικά με τους δράστες της επικοινωνίας, ο κάθε χρήστης μπορεί να είναι τόσο αποστολέας/παραγωγός όσο και παραλήπτης/καταναλωτής ενός μηνύματος. Ωστόσο σημαντικό στοιχείο είναι το ότι είναι χαλαρά συζευγμένοι ενώ ούτε ο αποστολέας του μηνύματος γνωρίζει ποιες ή πόσες εφαρμογές λαμβάνουν τα μηνύματα ούτε ο παραλήπτης γνωρίζει ποιες εφαρμογές στέλνουν τα δεδομένα. Αν χρειάζεται να σταλεί κάποια απάντηση, τότε ορίζεται σε ποιο κανάλι πρέπει να μεταφερθεί το μήνυμα. Ανάλογα με την υλοποίηση οι ορισμοί των εικονικών καναλιών επικοινωνίας είναι ενσωματωμένοι είτε μέσα στον κώδικα της εφαρμογής ή περιγράφονται και ορίζονται διαχειριστικά αποθηκεύοντας τις λεπτομέρειες σε μια υπηρεσία καταγραφής. Αλλά και πάλι η δημιουργία των καναλιών μηνυμάτων χρειάζεται να οριστεί στην εφαρμογή που χρησιμοποιεί το MOM.

Εστιάζοντας στα κανάλια επικοινωνίας, στο κανάλι “publish-and-subscribe” οι παραλήπτες που ενδιαφέρονται για κάποιο θέμα δηλώνουν ενδιαφέρον ενώ οι αποστολείς στέλνουν μηνύματα δημοσιεύοντας σε αυτό το θέμα. Τα μηνύματα λαμβάνονται από αυτούς που είχαν δηλώσει ενδιαφέρον. Στην περίπτωση του “point-to-point”, ένας μόνο παραλήπτης λαμβάνει το μήνυμα το οποίο καταχωρείται σε μια ουρά. Βέβαια μπορεί να υπάρχουν περισσότεροι του ενός παραλήπτες αλλά καθένας καταναλώνει ένα μοναδικό μήνυμα. Άλλες τεχνικές εφαρμόζονται σε περιπτώσεις απουσίας παραληπτών και για τα δυο μοντέλα καναλιών. Τέλος, η επιλογή του

μοντέλου εξαρτάται αποκλειστικά από τα επικοινωνούντα μέρη και τον σκοπό της επικοινωνίας.

Μια υλοποίηση των “publish-and-subscribe” υπαγορεύει τη διάρθρωση και τη διαχείριση των θεμάτων σε ιεραρχική διάταξη. Με αυτό τον τρόπο ο καταναλωτής μπορεί να δηλώσει ενδιαφέρον σε έναν κόμβο οπουδήποτε επιπέδου και κλάδου του δένδρου της ιεραρχίας και να λάβει τα μηνύματα που παραπέμπουν σε κόμβους ανώτερου επιπέδου της ιεραρχίας. Επίσης τις ιεραρχίες αυτές μπορούν να εκμεταλλευτούν οι διαχειριστές του συστήματος για να ορίσουν επίπεδα διάφορες πολιτικές ασφαλείας μέσω των μηχανισμών λιστών ελέγχου πρόσβασης (Access Control Lists).

Ολοκληρώνοντας τη σύντομη παρουσίαση του MOM να παρουσιαστεί η δομή ενός μηνύματος τύπου MOM. Τρία είναι τα βασικά σημεία του μηνύματος: οι επικεφαλίδες, οι ιδιότητες και το σώμα του μηνύματος. Οι επικεφαλίδες περιέχουν πληροφορίες για την κατεύθυνση, την κατεύθυνση απάντησης, τον τύπο του μηνύματος, το χρόνο λήξης του μηνύματος. Το μέρος των ιδιοτήτων περιλαμβάνει ένα σύνολο από ζευγάρια όνοματος/τιμής σύμφωνα με την εφαρμογή και χρειάζονται για εργασίες φιλτραρίσματος ενώ αποτελούν μέρη του κυρίως σώματος του μηνύματος.



**Εικόνα 15-MOM μήνυμα**

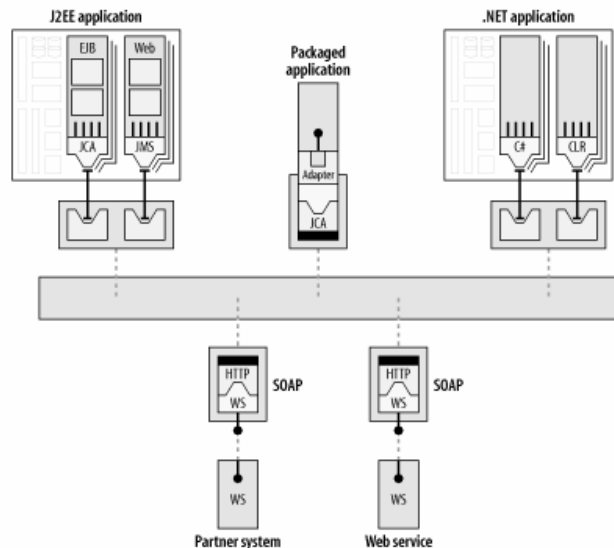
### 4.3.5 Δίαυλος Επιχειρηματικών Υπηρεσιών (Enterprise Service Bus)

Ο Δίαυλος Επιχειρηματικών Υπηρεσιών ή Enterprise Service Bus (ESB), αποτελεί μια πρόσφατη πρόταση για την επίτευξη της επιθυμητής διαλειτουργικότητας στο εκτεταμένο περιβάλλον μιας επιχείρησης. Το περιβάλλον αυτό περιλαμβάνει τόσο την ίδια την επιχείρηση όσο και τους συνεργάτες αυτής οι οποίοι χωρίζονται τόσο από επιχειρηματικά όσο και από φυσικά όρια ενώ οι διάφορες εφαρμογές ακόμα και του ίδιου οργανισμού μπορεί να είναι κατανεμημένες γεωγραφικά, να περιορίζονται από τείχη προστασίας και πολιτικές ασφαλείας. Ο ESB μέσα σε ένα τέτοιο περιβάλλον προσφέρει ευκαιρίες ενοποίησης χαλαρά συζευγμένων και κατανεμημένων εφαρμογών. Συνδυάζει ανταλλαγή μηνυμάτων, web services, μετατροπή δεδομένων, ευφυή δρομολόγηση με στόχο οι διάφορες κατανεμημένες διαδικασίες και μονάδες ενοποίησης να μπορέσουν απρόσκοπτα να συνδυαστούν και να συνεργαστούν προσφέροντας ολοκληρωμένες, αυτοματοποιημένες επιχειρηματικές υπηρεσίες.

Τα βασικότερα χαρακτηριστικά του Enterprise Service Bus εμφανίζονται στη συνέχεια.

- ✓ *Διεισδυτικότητα (pervasiveness)*: οι εφαρμογές μπορούν να ενταχθούν στον ESB, έτοιμες να εντοπιστούν και να συνεργαστούν με άλλες εφαρμογές ή υπηρεσίες που επίσης βρίσκονται στον ESB. Η επικοινωνία των εφαρμογών επιτυγχάνεται μέσω διεπαφών (APIs), πολλαπλών πρωτοκόλλων, περιβάλλοντα ανταλλαγής μηνυμάτων και προσαρμογέων εφαρμογών (application adapters).
- ✓ *Ενοποίηση βασισμένη σε πρότυπα (standards-based integration)*: η αρχιτεκτονική του ESB μπορεί να χρησιμοποιήσει μια πληθώρα τεχνολογιών και προτύπων όπως: μονάδες J2EE όπως Java Message Service (JMS) για διασύνδεση μέσω MOM, J2EE Connector Architecture (JCA ή J2CA) για διασύνδεση στους προσαρμογείς εφαρμογών (application adapters). Ακόμα λειτουργεί με εφαρμογές που προέρχονται από διάφορα περιβάλλοντα ανάπτυξης όπως .NET, COM, C#, and C/C++. Αλλά επίσης μπορεί να υποστηρίξει εφαρμογές που υποστηρίζουν SOAP και web-services APIs ενώ για τον χειρισμό των δεδομένων χρησιμοποιεί τα XML πρότυπα όπως XSLT, XPath, and XQuery για μετατροπή δεδομένων, ευφυή δρομολόγηση και

ερωτήματα πάνω στα δεδομένα. Τέλος πρότυπα τα οποία δέχεται είναι η Web Services Description Language (WSDL), Business Process Execution Language for Web Services (BPEL4WS), WS-Choreography[68] ακόμα και XML-based λεξικά όπως το ebXML[12] BPSS, για περιγραφή αφαιρετικών επιχειρηματικών διαδικασιών.



**Εικόνα 16- Ενοποίηση βασισμένη σε πρότυπα**

- ✓ *Υψηλή κατακευματισμένη ενοποίηση και επιλεκτική ανάπτυξη (Highly Distributed Integration and Selective Deployment):* ο ESB προσφέρει τα χαρακτηριστικά της ενορχήστρωσης, της δρομολόγησης των δεδομένων, της μετατροπής των δεδομένων και των προσαρμογών ως υπηρεσίες που μπορούν να συνδυαστούν σε ένα κατακευματισμένο περιβάλλον και καθεμιά μπορεί να αναπτυχθεί ξεχωριστά.
- ✓ *Κατακευματισμένη μετατροπή δεδομένων:* ο ESB προσφέρει μηχανισμούς μετατροπής των δεδομένων στο κατάλληλο σχήμα ενώ ταυτόχρονα αυτοί είναι κατακευματισμένοι και μπορούν να εντοπιστούν οπουδήποτε.
- ✓ *Επεκτασιμότητα μέσω κλιμακούμενων υπηρεσιών (Extensibility Through Layered Services):* ο ESB διαθέτει πολλές δυνατότητες ενοποίησης διαδικασιών και μάλιστα μέσω κλιμακούμενης τεχνολογίας μπορεί να προσφερθούν εξειδικευμένες υπηρεσίες.
- ✓ *Event-Driven SOA:* στην οδηγούμενη από τα γεγονότα αρχιτεκτονική οι εφαρμογές και οι υπηρεσίες θεωρούνται ως άκρα τα οποία μπορούν να κληθούν από γεγονότα. Δεν υπάρχει ενδιαφέρον για την υποδομή της επικοινωνίας ενώ δεν χρειάζεται να δοθεί σημασία στα πρωτόκολλα. Οι

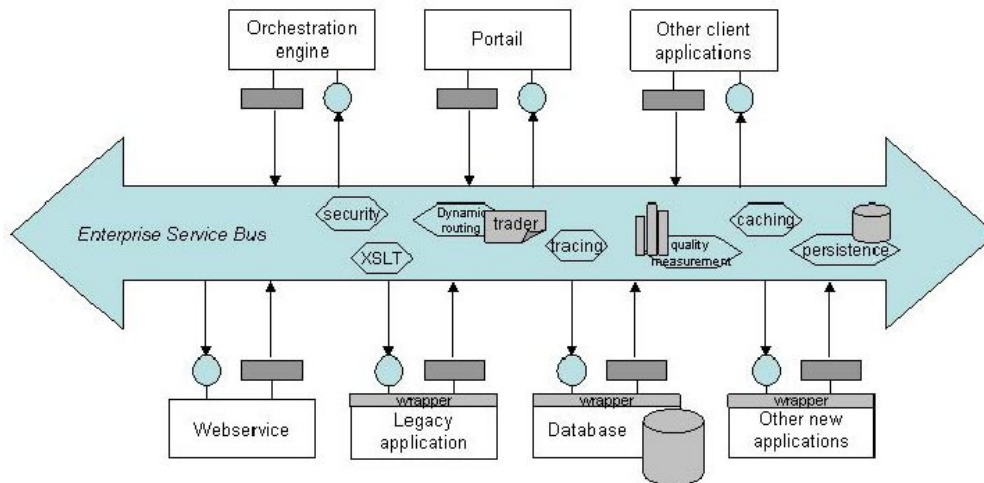
υλοποιήσεις των υπηρεσιών παίρνουν τα μηνύματα ως γεγονότα και τα επεξεργάζονται ενώ ο ESB μεταφέρει τα μηνύματα οπουδήποτε χρειάζεται. Οι υπηρεσίες μέσω του ESB και ειδικών εφαρμογών ενοποίησης μπορούν να συνθέσουν υπηρεσίες και ροές διαδικασιών οι οποίες να είναι επαναχρησιμοποιήσιμες και επανασυνδυαζόμενες με στόχο την αυτόματη ανάκληση των επιχειρηματικών διαδικασιών.

- ✓ *Ροή διεργασιών (process flow)*: πρόκειται για ροές που αποτελούνται από πεπερασμένα βήματα διεργασιών έως περίπλοκες επιχειρηματικές διαδικασίες που περιλαμβάνουν παράλληλα μονοπάτια εκτέλεσης διαδικασιών με σημεία διαχωρισμού και συνένωσης. Σύμφωνα με τις δυνατότητες που προσφέρει ο ESB μπορούν να υπάρξουν επιχειρηματικές διαδικασίες που εντάσσονται σε έναν συγκεκριμένο τομέα ή που εκτείνονται σε μια ολόκληρη επιχειρηματική μονάδα ή σε ένα μεγαλύτερο δίκτυο ολοκληρωμένων διεργασιών. Επιπλέον λόγω του ότι οι ροές των διαδικασιών δομούνται σε ανώτερο επίπεδο της κατανεμημένης αρχιτεκτονικής SOA, αυτές μπορεί να εκτείνονται σε μεγάλες κατανεμημένες τοπολογίες χωρίς να δημιουργείται εμπόδιο ως προς τα φυσικά όρια του δικτύου ή τα άλματα από πρωτόκολλο σε πρωτόκολλο ανάμεσα στις διάφορες εφαρμογές μέσα στον ESB.
- ✓ *Ασφάλεια και αξιοπιστία (security and reliability)*: οι διάφορες μονάδες του ESB παρέχουν ασφάλεια από τείχη προστασίας ενώ ταυτόχρονα η επικοινωνία μεταξύ των κόμβων του ESB αλλά και αυτών με τον ESB στηρίζεται σε κανόνες πιστοποίησης, χειρισμού πιστοποιητικών και ελέγχου πρόσβασης. Η αξιοπιστία επιτυγχάνεται με την εφαρμογή της τεχνολογίας MOM η οποία παρέχει ασύγχρονες επικοινωνίες, αξιόπιστη παράδοση δεδομένων και ακεραιότητα των συναλλαγών.
- ✓ *Αυτόνομο αλλά συσπειρωμένο περιβάλλον (Autonomous but Federated Environment)*: ο ESB δίνει τη δυνατότητα στις τοπικές μονάδες και τα τμήματα να είναι αυτόνομα στον έλεγχο των IT πόρων τους επιτρέποντας τοπική διακίνηση μηνυμάτων, οι μονάδες ενοποίησης και προσαρμογείς οι οποίοι είναι εγκατεστημένοι ελέγχονται και είναι ασφαλισμένοι τοπικά έχοντας ωστόσο τη δυνατότητα να συνδυαστούν και να αποτελέσουν ένα ολοκληρωμένο δίκτυο με ένα ολοκληρωμένο μοντέλο ασφαλείας. Τα κατανεμημένα χαρακτηριστικά του ESB επιτυγχάνονται από την αφαίρεση των άκρων των υπηρεσιών από τη φυσική υπόσταση και τα πρωτόκολλα και

με τη εφαρμογή του προτύπου ενορχήστρωσης μαζί με δρομολόγηση των δεδομένων ανάμεσα σε αυτά τα άκρα. Το ενοποιημένο περιβάλλον από την άλλη οφείλεται στην δυνατότητα που υπάρχει να προσπελαστούν τομείς εφαρμογών και όρια ασφαλείας.

- ✓ *Απομακρυσμένη πιστοποίηση και διαχείριση (Remote Configuration and Management)*: μια λύση σε αυτό το πρόβλημα θα μπορούσε να είναι η διατήρηση ενός προσχεδίου ενοποίησης που να περιλαμβάνει τους ορισμούς της διαπροσωπείας για τις εφαρμογές του τοπικού συστήματος, τη δρομολόγηση των μηνυμάτων κίνησης και του καναλιού διαχείρισης αλλά και παραχωρήσεις ασφαλείας. Επιπλέον θα ήταν χρήσιμοι προσαρμογείς για τις εφαρμογές, τα πρωτόκολλα και τη μετατροπή των δεδομένων.
- ✓ *XML ως ο "Native" τύπος δεδομένων του ESB*: παρόλο που ο ESB δίνει τη δυνατότητα μεταφοράς των δεδομένων οποιασδήποτε μορφής αυτό που θα προσέφερε περισσότερο θα ήταν η αναπαράσταση των μεταφερόμενων δεδομένων σε XML μορφή. Έτσι θα υπήρχε η ευκαιρία σύνθεσης νέων όψεων δεδομένων από διαφορετικές πηγές έτσι ώστε να εμπλουτιστούν και να επαναπροσδιοριστούν τα μηνύματα για εξελιγμένη από κοινού χρήση των δεδομένων μεταξύ των εφαρμογών.
- ✓ *Πραγματικού-χρόνου απόκριση των επιχειρηματικών δεδομένων (Real-Time Throughput of Business Data)*: πρόκειται για έναν τρόπο απόκρισης των δεδομένων κατά το πέρασμα από τις διάφορες διαδικασίες. Έτσι αποφεύγεται η επεξεργασία των δεδομένων κατά μεγάλα πακέτα ιδιαίτερα κατά τη διάρκεια της νύχτας.
- ✓ *Γνώση της κατάστασης λειτουργίας (operational awareness)*: μια πολύ σημαντική δυνατότητα για την παρακολούθηση της κατάστασης και της πορείας των επιχειρηματικών λειτουργιών. Σε περίπτωση που η XML χρησιμοποιείται ως native τύπος των δεδομένων καθίσταται ευκολότερη η πραγματικού-χρόνου εξέταση των δεδομένων καθώς μεταφέρονται στον ESB. Επιπλέον μπορούν να τοποθετηθούν σημεία εξέτασης και ελέγχου σε μια ροή διεργασιών, τα οποία παράλληλα χρησιμεύουν ως σημεία εντοπισμού του περιεχομένου των μηνυμάτων που διακινούνται. Αναφέρονται τέλος οι XML υπηρεσίες που προσφέρουν σημεία για συγκέντρωση και αποθήκευση των δεδομένων και μεταφορά αυτών σε άλλες εφαρμογές ή στους χρήστες για περαιτέρω χρήση τους και ανάλυση.

- ✓ *Αυξητική υιοθέτηση (incremental adoption)*: πρόκειται για το χαρακτηριστικό της αυξητικής εφαρμογής ολοκληρωμένων εφαρμογών στον ESB. Έτσι δημιουργείται η προοπτική μείωσης του κόστους υλοποίησης ολοκληρωμένων εφαρμογών και λόγω της επαναχρησιμοποίησης ήδη υπάρχοντων έργων σε σχέδια ολοκληρωμένων εφαρμογών.



Εικόνα 17-Enterprise Service Bus

## 4.4 Αρχιτεκτονικές και εφαρμογές σε ερευνητικό επίπεδο

### 4.4.1 Ecosystem Oriented Architecture (EOA)

Σε ερευνητικό επίπεδο γύρω από το ζήτημα της αλληλεπίδρασης των επιχειρηματικών οργανισμών, αναπτύσσεται τα τελευταία χρόνια η ιδέα του ψηφιακού επιχειρηματικού οικοσυστήματος (Digital Business Ecosystem - DE) το οποίο εξαρχής στόχευε στην ενίσχυση του ICT (Information and Communication Technology) υπόβαθρου των επιχειρήσεων και απέδιδε την κοινή εξέλιξη του επιχειρησιακού οικοσυστήματος μαζί με την μερική ψηφιακή του αναπαράσταση. Η ιδέα του Digital Business Ecosystem γεννήθηκε το 2002 προσθέτοντας τον όρο “digital” μπροστά από τους όρους “business ecosystem” του Moore (1996) στο Unit

ICT for Business of the Directorate General Information Society of the European Commission (Nachira, 2003). Ουσιαστικά πρόκειται για τον συνδυασμό των χαρακτηριστικών τριών διαφορετικών εννοιών: του ψηφιακού οικοσυστήματος, του επιχειρηματικού οικοσυστήματος και του οικοσυστήματος γενικότερα. Σύμφωνα με την ερμηνεία των όρων αυτών, καταρχάς λαμβάνουμε υπόψη την τεχνική υποδομή των P2P κατανεμημένων αρχιτεκτονικών που διασυνδέουν υπηρεσίες και πληροφορίες στα πλαίσια των διαδικτυακών συναλλαγών και κατανέμουν τους ψηφιακούς πόρους. Επίσης απολαμβάνουμε τις υπηρεσίες και τα αγαθά που παράγονται και προσφέρονται από το οικοσύστημα των συνεργαζόμενων επιχειρήσεων της ελεύθερης αγοράς αλλά και επηρεαζόμαστε από τις αλληλεπιδράσεις των ρόλων και των δυνατοτήτων των στοιχείων που συνθέτουν το οικοσύστημα. Εδώ διακρίνουμε επιπλέον τις δυνάμεις εξέλιξης, αυτό-οργάνωσης και αυτόνομης βελτίωσης του εξεταζόμενου περιβάλλοντος.

Στα πλαίσια των εργασιών της Ευρωπαϊκής Κοινότητας και υπό την έννοια του Ψηφιακού Περιβάλλοντος [13], που αναφέρθηκε προηγουμένως, προτείνεται μια νέα αρχιτεκτονική η οποία επιδιώκει να συμβάλει στην οργάνωση του επιχειρηματικού ψηφιακού οικοσυστήματος. Η Service Oriented αρχιτεκτονική, που έχει ήδη περιγραφεί σε προηγούμενο κεφάλαιο, δε μπορεί να ανταποκριθεί στους κανόνες που διέπουν το νέο δυναμικό ψηφιακό περιβάλλον. Σε αυτό το σημείο η Ecosystem Oriented αρχιτεκτονική έρχεται να αντιμετωπίσει τις προκλήσεις που θέτει το ψηφιακό οικοσύστημα και οι οποίες έχουν σχέση με την αποκεντροποιημένη μορφή των ψηφιδών του συστήματος, τον χειρισμό κατανεμημένων βάσεων γνώσης, την αυτό-οργάνωση και την αυτόνομη ανάκαμψη από αποτυχίες.

Σε αντίθεση με τον αποκεντροποιημένο δυναμικό χαρακτήρα της EOA, η SOA προέρχεται από ένα περιβάλλον επικεντρωμένο στις διαδικασίες, τις εφαρμογές και τα αντικείμενα και τις τεχνολογίες των EAI, RPC και CORBA με νεότερο προσανατολισμό στις υπηρεσίες. Ωστόσο και σε αυτό το σημείο πρόκειται για μια αρχιτεκτονική που αρχικά τουλάχιστον εφαρμόστηκε περισσότερο για μια συγκεκριμένη αλυσίδα παραγωγής, και σε συγκεκριμένους επιχειρηματικούς τομείς, ενδο-επιχειρησιακά και με κεντρική διαχείριση των εργασιών. Η αρχιτεκτονική EOA καλείται να ανταποκριθεί στις ανάγκες του ψηφιακού επιχειρηματικού οικοσυστήματος ξεφεύγοντας από τα στενότερα πλαίσια της SOA. Στοχεύει στην ενοποίηση των υπηρεσιών που προσφέρονται από τις διάφορες μικρομεσαίες



επιχειρήσεις χωρίς αυτές να επιβαρύνονται με κόστη επιπρόσθετων επενδύσεων και χωρίς την ανθρώπινη παρέμβαση.

Αναφορικά με την υλοποίηση της ΕΟΑ αρχιτεκτονικής, πρέπει να ληφθεί υπόψη η αναγκαιότητα ορισμού διαφόρων μη στατικών και περίπλοκων λειτουργικών μοντέλων από τους ίδιους τους συμμετέχοντες οι οποίοι στη συνέχεια θα μπορούν να χρησιμοποιούν οποιοδήποτε από αυτά τα μοντέλα τους εξυπηρετεί. Έτσι συνολικά είναι απαραίτητος ένας μηχανισμός δημοσίευσης των μοντέλων αυτών, ανεύρεσης του μοντέλου που είναι το πλέον κατάλληλο για κάποια συγκεκριμένη εργασία καθώς και υιοθέτησης αυτού του μοντέλου ελεύθερα και χωρίς περιορισμούς παρά μόνο για προστασία του περιβάλλοντος δραστηριοτήτων.

Προκειμένου να καταστεί δυνατός ο χειρισμός των διαφόρων μοντέλων, των αλληλεξαρτήσεων τους και των σχέσεων που αναπτύσσουν είναι απαραίτητη η ύπαρξη μιας αποθήκης μοντέλων (model repository). Η λειτουργία αυτής της μονάδας δε θα μοιάζει με αυτή ενός UDDI[58] των Web Services όπου εκεί εμφανίζονται οι ορισμοί των διαπροσωπειών των υπηρεσιών, αλλά το νέο αυτό repository θα μπορεί να χειριστεί επιχειρησιακά μοντέλα και όχι απλά προγραμματιστικούς ορισμούς. Σε αυτή την κατεύθυνση είναι δυνατό να συμβάλει το πρότυπο XMI καθώς είναι ανεξάρτητο τεχνολογικής πλατφόρμας και μπορεί να περιγράψει σημασιολογικά πλούσιους ορισμούς μοντέλων.

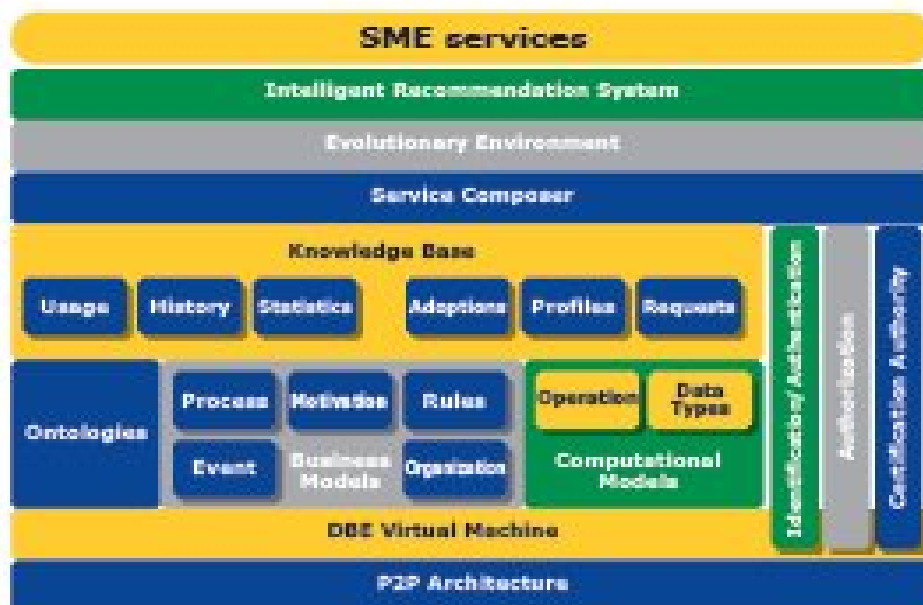
Οι λειτουργίες που υποστηρίζονται από το repository βασίζονται σε ένα σύνολο πληροφοριών που συνοδεύουν τα επιχειρησιακά μοντέλα, όπως επιχειρησιακά μοντέλα υπηρεσιών (service business models), τεχνικούς ορισμούς (technical specifications), επιχειρησιακά δεδομένα (business data), άκρα υπηρεσιών (service endpoints). Αυτές οι πληροφορίες των διαφόρων υπηρεσιών σύμφωνα με την παρούσα αρχιτεκτονική βρίσκονται μέσα σε κάποιο αρχείο καταγραφής της εκάστοτε υπηρεσίας (service registry) και δίνουν τη δυνατότητα σε ανθρώπους και ψηφίδες λογισμικού να περιηγηθούν και να ανακαλέσουν τις πληροφορίες αυτές.

Επιπλέον στα πλαίσια του ορισμού της ΕΟΑ, ορίζεται ένα σύνολο βασικών λειτουργιών που βασικό στόχο έχει την διευκόλυνση των συμμετεχόντων κατά την δημοσίευση και την ενοποίηση των επιμέρους υπηρεσιών. Η υλοποίηση δεν παρέχεται άμεσα αλλά ο ίδιος ο χρήστης μπορεί να υλοποιήσει το μοντέλο της υπηρεσίας που του παρέχεται. Παραδείγματα υπηρεσιών είναι: πληρωμή, επιχειρηματικά συμβόλαια και διαπραγματεύσεις, τιμολόγηση, εμπιστοσύνη, νόμιμη συμβατότητα και άλλες.

Η αποθήκη μοντέλων και το αρχείο καταγραφής των υπηρεσιών μπορούν να αποτελέσουν προβληματικό σημείο και σε περίπτωση αποτυχίας τους να αφήσουν εκκρεμές το σύστημα. Για να αντιμετωπιστεί ένα τέτοιο σενάριο αποτυχίας λαμβάνεται μέριμνα για την αποκεντροποιημένη αποθήκευση των δεδομένων ενώ παράλληλα αποφεύγεται ο κεντρικός έλεγχος των αποθηκευμένων δεδομένων και δίνεται περισσότερη έμφαση στα χαρακτηριστικά της αυτόνομης διάρθρωσης και της προσαρμοστικότητας που αποδίδονται στο σύστημα.

Συνοψίζοντας, εφόσον η Service Oriented αρχιτεκτονική δε μπορεί να υποστηρίξει τα ψηφιακά επιχειρηματικά οικοσυστήματα, ερευνάται η εφαρμογή μιας αρχιτεκτονικής που δίνει περισσότερη έμφαση στα εξής στοιχεία:

1. μοντέλα υπηρεσιών, επιχειρησιακά και υπολογιστικά με απώτερο στόχο την επαναχρησιμοποίησή τους
2. υλοποίηση των υπηρεσιών, περιγραφές υλοποίησης υπηρεσιών και δεδομένα αυτών
3. στιγμιότυπα των υπηρεσιών που αναπαριστούν το όνομα και τα άκρα κλήσεων των υπηρεσιών.



Εικόνα 18- στοίβα υπηρεσιών στο Digital Business Ecosystem

## 4.4.2 Genesis Hybrid Architecture

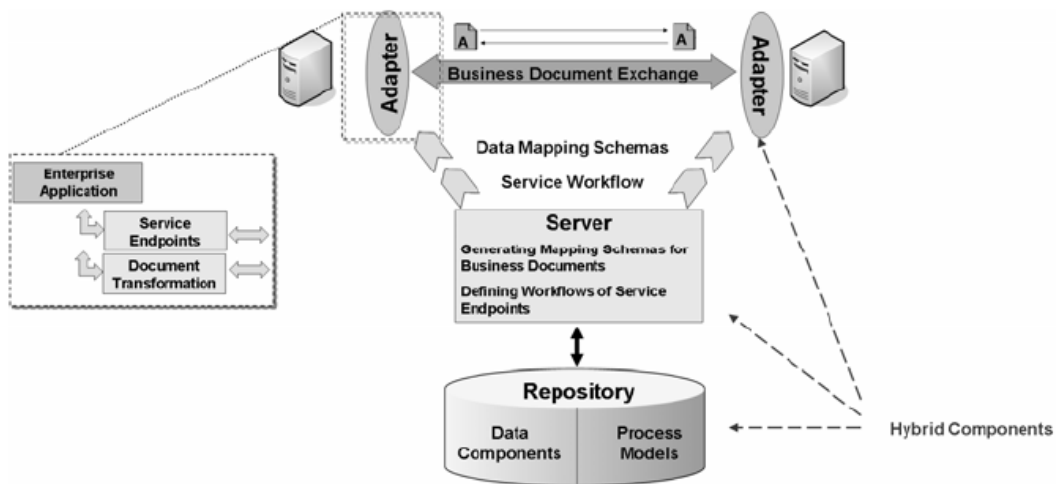
Η υβριδική αρχιτεκτονική του έργου Genesis[15] εντάσσεται στα γενικότερα ερευνητικά πλαίσια που στοχεύουν στην χαλαρή σύζευξη (loose coupling) και στην άμεση κλήση και εκτέλεση των επιχειρηματικών διεργασιών σε ένα μωσαϊκό εφαρμογών των επιχειρήσεων επιδιώκοντας την επιθυμητή διαλειτουργικότητα των συστημάτων των οργανισμών. Η παρούσα αρχιτεκτονική χρησιμοποιώντας ευρέως διαδεδομένα πρότυπα αρχιτεκτονικών επικοινωνίας, προτύπων αναπαράστασης των υπηρεσιών καθώς και γενικότερων τεχνολογιών προτείνει μια υβριδική λύση η οποία εκμεταλλεύεται τα πλεονεκτήματα των Peer-to- Peer (P2P) αρχιτεκτονικών για την πραγματοποίηση ηλεκτρονικών συναλλαγών μεταξύ ομότιμων συνεργατών. Παράλληλα επωφελείται από την ύπαρξη μια κεντρικής μονάδας εξυπηρέτησης.

Συγκεκριμένα, η προτεινόμενη αρχιτεκτονική προδιαγράφει την ύπαρξη τριών βασικών λειτουργικών μονάδων[16]:

A) *Αποθήκη (Repository)*: περιέχει σημασιολογικές πληροφορίες σχετικά με τα χαρακτηριστικά των πλατφορμών των χρηστών ως μοντέλα διαδικασιών, διασυνδέοντας άκρα υπηρεσιών και μοντέλα δεδομένων. Επίσης μπορεί να περιλαμβάνει επιχειρησιακά μοντέλα και κανόνες που διέπουν τις συναλλαγές.

B) *Εξυπηρετητής (Server)*: είναι επιφορτισμένος με την επεξεργασία των στοιχείων που είναι απαραίτητα για την εκτέλεση των συναλλαγών άμεσα μεταξύ των χρηστών. Εκτελεί την σύνδεση των ροών εργασίας βάσει των μοντέλων διαδικασιών και των άκρων των υπηρεσιών που υποστηρίζουν τα συμμετέχοντα μέλη. Επιπλέον προσφέρει αντιστοιχίσεις των δεδομένων, αναγκαίες για την επίτευξη της επιθυμητής διαλειτουργικότητας σε επίπεδο δεδομένων. Τέλος είναι ικανός να εξυπηρετήσει ως προς τη διαχείριση της ροής των συναλλαγών όπου αυτό κρίνεται απαραίτητο.

Γ) *Προσαρμογέας (Adapter)*: προσφέρει την τεχνική υποδομή για την πραγματοποίηση της απευθείας επικοινωνίας ανάμεσα στους πελάτες του συστήματος. Εκτελεί τις δυναμικές ροές εργασιών (workflows) που παρέχονται από τον εξυπηρετητή, κάνει μετατροπή των δεδομένων στα δικά τους πρότυπα και καλεί τα κατάλληλα άκρα των υπηρεσιών της δικής τους τεχνολογικής υποδομής.



Εικόνα 19- Αρχιτεκτονική του Genesis

Επικεντρώνοντας στα τεχνικά ζητήματα υλοποίησης της αρχιτεκτονικής του GENESIS, αναφέρουμε ότι η αποθήκη (Repository) της πλατφόρμας χρειάζεται να υποστηρίζει μοντέλα διαδικασιών τα οποία περιγράφονται σε XML όπως XMI, BPMN[5] μοντέλα, ebXML[12] BPSS. Επιπλέον, είναι σημαντικό να περιέχει μοντέλα επιχειρηματικών εγγράφων σε XML μορφή όπως είναι τα UN/CEFACT Core Components και οι Naming and Design Rules. Τέλος δεν πρέπει να λείπουν και περιγραφές πάλι σε XML μορφή που προδιαγράφουν προφίλ των χρηστών καθώς και τις συμφωνίες που τηρούνται στα πλαίσια των συναλλαγών π.χ. ebXML[12] Collaborations Protocol Profiles and Agreement. Προκειμένου τα σημασιολογικά περιεχόμενα της αποθήκης να είναι εύκολα διαχωρίσιμα, προτείνεται η διάκριση αυτών σε τρία επίπεδα:

- επίπεδο μοντελοποίησης των επιχειρηματικών διεργασιών, στο οποίο οι διαδικασίες χωρίζονται σε εσωτερικές του οργανισμού και εξωτερικές που περιγράφουν στοιχεία που έχουν να κάνουν με τις εξωτερικές επαφές των οργανισμών.
- επίπεδο μοντελοποίησης των επιχειρηματικών εγγράφων, όπου τα μοντέλα των εγγράφων δομούνται από μονάδες προτυποποιημένων δεδομένων όπως τα UN/CEFACT Core Components.
- επίπεδο μοντελοποίησης των επιχειρηματικών και νομικών κανόνων οι οποίοι ρυθμίζουν τις συναλλαγές. Εδώ είναι απαραίτητο ένα μεταμοντέλο που ορίζει στοιχεία εφαρμογής των κανόνων ανάλογα και με τον τομέα δράσης.

Από την πλευρά του ο εξυπηρετητής εκτελεί δυο βασικές λειτουργίες και συγκεκριμένα τη σύνθεση των δυναμικών ροών εργασίας το οποίο επιτυγχάνεται

με τη χρήση των μοντέλων διαδικασιών, την αντιστοίχιση των άκρων των υπηρεσιών κάθε μέρους που συμμετέχει και τη δημιουργία εκτελέσιμων όπως για παράδειγμα σε BPEL μορφή. Επιπρόσθετα ο εξυπηρετητής αναλαμβάνει τον ορισμό των επιχειρηματικών εγγράφων ανάλογα με τα μοντέλα δεδομένων που υποστηρίζει καθένα από τα εμπλεκόμενα άκρα της συναλλαγής. Αυτά εν τέλει παρέχονται στον πελάτη σε XML μορφή.

Ο προσαρμογέας (adapter) στα άκρα της επικοινωνίας παραλαμβάνει τα εκτελέσιμα σε BPEL μορφή καθώς και τα σχήματα μετασχηματισμού των εγγράφων και εκτελεί σε κατάλληλη BPEL μηχανή τη ροή των εργασιών ενώ παράλληλα μετατρέπει τα επιχειρηματικά έγγραφα στα δικά του κατανοητά πρότυπα.

Εν κατακλείδι, πρόκειται για μια αρχιτεκτονική που προσπαθεί να συνδυάσει τα πλεονεκτήματα των P2P και Client-Server αρχιτεκτονικών, βασιζόμενη σε θεσπισμένα και ευρέως εφαρμοζόμενα πρότυπα σε επίπεδο δεδομένων και πρωτοκόλλων επικοινωνίας. Βασική της επιδίωξη είναι η εύκολη, άμεση και μειωμένου κόστους σύνθεση των εργασιών των ηλεκτρονικών εφαρμογών με απώτερο σκοπό την διαλειτουργικότητα των ετερογενών Πληροφορικών Συστημάτων. Παράλληλα λόγω του ανοιχτού χαρακτήρα της αρχιτεκτονικής, επωφελούνται ιδιαίτερα οι μικρομεσαίες επιχειρήσεις οι οποίες πλέον είναι ικανές να υιοθετήσουν ανοιχτά πρότυπα και να ανταγωνιστούν μεγάλους οργανισμούς στα πλαίσια του ηλεκτρονικού εμπορίου (B2B, B2G, B2C).

# 5 ■ Ανάλυση σε επίπεδο αρχιτεκτονικής

## 5.1 Διαστάσεις αρχιτεκτονικών συστημάτων ηλεκτρονικών συναλλαγών

Η προδιαγραφή, τεκμηρίωση και ανάπτυξη μιας αρχιτεκτονικής συστήματος ηλεκτρονικών συναλλαγών συνθέτουν μια ιδιαίτερα σύνθετη διαδικασία. Η διαδικασία αυτή στηρίζεται σε μεγάλο βαθμό στην αρχιτεκτονική του λογισμικού ή όπως αναφέρεται σε αγγλικούς όρους, “software architecture”. Ουσιαστικά με τον όρο αυτό αποδίδεται η δομή του συστήματος που πρόκειται να υλοποιηθεί και συγκεκριμένα οι μονάδες λογισμικού που θα χρησιμοποιηθούν, οι ορατές ιδιότητες των μονάδων αυτών καθώς και οι σχέσεις που τις συνδέουν. Η ανάλυση της αρχιτεκτονικής λογισμικού για ένα σύστημα συντελεί αποτελεσματικά στην απλοποίηση της ανάπτυξης του συστήματος και τον έλεγχο της σωστής λειτουργίας ενώ τέλος θεωρείται ότι προσφέρει στην επαναχρησιμοποίηση των τεχνικών και των μονάδων που προδιαγράφονται.

Στην πραγματικότητα, στα πλαίσια της αρχιτεκτονικής λογισμικού, το σύστημα αναλύεται και εξετάζεται σε διάφορα αφαιρετικά επίπεδα και αντιμετωπίζονται χωριστά οι διαφορετικές όψεις, περιορισμοί και επιδιώξεις που ορίζονται από τους χρήστες του συστήματος. Με αυτό τον τρόπο οι αποφάσεις που πρέπει να ληφθούν για την υλοποίηση κατανέμονται σε διαφορετικά επίπεδα ευθύνης και αποδίδονται παραδείγματος χάριν στους ιδιοκτήτες των συστημάτων, στους μηχανικούς ανάπτυξης του λογισμικού, στις ομάδες υποστήριξης του λογισμικού, στους επιχειρηματίες που θα χρησιμοποιήσουν το σύστημα κοκ.

Σύμφωνα με το [20] διακρίνονται διάφορες όψεις σύμφωνα με τις οποίες οργανώνεται η αρχιτεκτονική λογισμικού. Ορισμένες από αυτές είναι οι εξής:

- Λειτουργική/ Λογική όψη (Functional/logic view)
- Όψη κώδικα/μονάδας (Code/module view)
- Όψη ανάπτυξης/ δομική (Development/structural view)

- Όψη συγχρονικότητας /διαδικασίας/ νήματος (Concurrency/process/thread view)
- Όψη φυσική/ ανάπτυξης (Physical/deployment view)
- Όψη δράσης των χρηστών/ ανάδρασης (User action/feedback view)
- Όψη των δεδομένων (Data view)

Στην περίπτωση της αρχιτεκτονικής συστήματος ηλεκτρονικών συναλλαγών θα βασιστούμε σε ορισμένες από τις προαναφερθείσες όψεις για να καθορίσουμε τις αρχιτεκτονικές διαστάσεις που θα μας απασχολήσουν.

### **5.1.1 Επιχειρηματική Αρχιτεκτονική (Business Architecture)**

Καταρχάς, οι επιχειρήσεις είναι οι κύριοι διαχειριστές ενός συστήματος ηλεκτρονικών συναλλαγών. Οι όροι “επιχειρήσεις” και “οργανισμοί” χρησιμοποιούνται για να αναπαραστήσουν το σύνολο των διαδικασιών, των υπηρεσιών, των δεδομένων, των επιχειρηματικών στόχων, των πολιτικών και των κανόνων που εντάσσονται στο περιβάλλον μια επιχείρησης. Παράλληλα, οι προηγούμενες έννοιες αντιπροσωπεύουν τις φυσικές οντότητες του περιβάλλοντος αυτού όπως πελάτες, προμηθευτές, συνεργάτες αλλά και τα πληροφορικά συστήματα ή τις βάσεις δεδομένων κλπ. Φυσικά δε μπορούν να παραληφθούν και οι σχέσεις που συνδέουν τις διάφορες οντότητες και οι οποίες είναι βασικό συστατικό του επιχειρηματικού περιβάλλοντος. Τα στοιχεία που μόλις αναφέρθηκαν αντιστοιχούν στην *επιχειρηματική αρχιτεκτονική (business architecture)* του συστήματος και θα μπορούσαν να αντιστοιχηθούν στις εξής όψεις: λειτουργική/λογική, ανάπτυξης/δομική, δεδομένων και αλληλεπίδρασης. Όπως προκύπτει, η επιχειρηματική αρχιτεκτονική καλύπτει βασικές παραμέτρους της οργάνωσης του συστήματος και αξίζει να θεωρηθεί ως μια από τις διαστάσεις ανάλυσης και λήψης αποφάσεων της αρχιτεκτονικής.

Ειδικότερα, η επιχειρηματική αρχιτεκτονική βοηθά κυρίως τον σχεδιαστή των ηλεκτρονικών συναλλαγών να κατηγοριοποιήσει τα ζητήματα του επιχειρηματικού τομέα όπου εντάσσονται οι συναλλαγές αυτές καθώς και να διακρίνει πιθανή

χρησιμότητα των υπηρεσιών στις επιχειρήσεις. Περιγράφει περιπτώσεις χρήσης παρουσιάζοντας τον τρόπο που αναπαρίστανται οι επιχειρηματικές διαδικασίες και προσπαθεί να αποδεσμεύσει την επιχειρηματική αρχιτεκτονική από την τεχνολογική πλατφόρμα. Στα πλαίσια αυτής της αρχιτεκτονικής διάστασης οι σχεδιαστές προσπαθούν να συγκεντρώσουν τους πόρους της επιχείρησης (σε δεδομένα και υποδομές πχ. βάσεις δεδομένων), τις επιχειρηματικές λειτουργίες και να τις αποτυπώσουν σε διαδικασίες και υπηρεσίες. Επιπλέον, οι επιχειρηματικές αυτές λειτουργίες ξεκινούν και ολοκληρώνουν την εκτέλεσή τους βάσει γενικών επιχειρησιακών κανόνων, όρων που καθορίζονται από τα εμπλεκόμενα μέλη και σίγουρα σε συνάρτηση με τους στόχους και τις επιδιώξεις των εταίρων. Επίσης πολύ σημαντικό είναι να λαμβάνεται υπόψη η δυναμική φύση των διαδικασιών αυτών έτσι ώστε σε περίπτωση ανάγκης να ξεκινούν ad hoc διαδικασίες. Εν ολίγοις, σύμφωνα με το [1] με τον όρο “*business architecture*” αποδίδεται η δομή της επιχείρησης καθώς και η ροή εργασιών που είναι σε άμεση συνάρτηση με τις επιχειρησιακές ιδέες, διαδικασίες και κανόνες.

### **5.1.2 Συναλλαγές (Transactions)**

Περνώντας από το ανώτερο αφαιρετικό επίπεδο σε αυτό των *συναλλαγών (transactions)*, εξετάζεται η όψη της δράσης των χρηστών η οποία επίσης καθορίζει σε μεγάλο βαθμό τη λειτουργία του συστήματος συναλλαγών. Οι συναλλαγές αποτελούν τον σκοπό δημιουργίας του συγκεκριμένου συστήματος και η πορεία αυτών επηρεάζει την αποτελεσματικότητά του. Προσεγγίζει περισσότερο την τεχνική σκοπιά του συστήματος συναλλαγών αλλά και πάλι ορίζει τον τρόπο αλληλεπίδρασης των χρηστών και μη τεχνικές απαιτήσεις του συστήματος όπως ανάκαμψη από λάθη και βασικές ιδιότητες των συναλλαγών. Σε αυτό το επίπεδο πρέπει να ξεκαθαριστεί ποιος ελέγχει την εξέλιξη των συναλλαγών και τα πρωτόκολλα που χρησιμοποιούνται για την διατήρηση της συνέπειας των δεδομένων και την διατήρηση της ομαλής εξέλιξης των συναλλαγών. Οι συναλλαγές ορίζουν επίσης ένα πολύ σημαντικό τομέα προδιαγραφής αρχιτεκτονικής που χρήζει μεγαλύτερης ανάλυσης. Ιδιαίτερες αποφάσεις και περιορισμοί πρέπει να ληφθούν σε αυτό το επίπεδο και για αυτό μπορεί να εξεταστεί ως μια ανεξάρτητη διάσταση της αρχιτεκτονικής.



Αναλυτικότερα, η διαδικασία των συναλλαγών επιτρέπει σε διάφορες αυτόνομες διαδικασίες να συνδέονται μεταξύ τους αυτόματα ως μια μοναδική αδιαίρετη διαδικασία. Γενικότερα ένα σύστημα που είναι υπεύθυνο για την εκτέλεση κάποιας συναλλαγής εξασφαλίζει ότι όλες οι λειτουργίες που λαμβάνουν χώρα στην εκάστοτε συναλλαγή ολοκληρώνονται χωρίς λάθη ή καμία από αυτές δεν ολοκληρώνεται επιτυχώς. Από τη στιγμή που κάποια λειτουργία είναι ολοκληρωμένη αλλά παρουσιαστούν προβλήματα κατά την προσπάθεια των άλλων, το σύστημα που επεξεργάζεται τη συναλλαγή εκτελεί τη διαδικασία “roll back” για όλες τις επιμέρους λειτουργίες, επιτυχημένες και μη, δηλαδή προσπαθεί να επανέλθει στη συνεπή γνωστή κατάσταση που βρισκόταν προτού ξεκινήσει η συναλλαγή. Από τη στιγμή που όλες οι λειτουργίες ολοκληρωθούν επιτυχώς, το σύστημα θεωρεί ολοκληρωμένη τη συναλλαγή και όλες οι αλλαγές γίνονται μόνιμες. Στα καταναμεμημένα συστήματα που επικοινωνούν μέσω των ηλεκτρονικών συναλλαγών προσέχεται ιδιαίτερα να τηρούνται οι ιδιότητες ACID[53]:

Ατομικότητα (Atomicity): πρέπει να γίνουν όλες οι λειτουργίες της συναλλαγής σωστά.

Συνέπεια (Consistency): η εκτέλεση μιας συναλλαγής απομονωμένα, χωρίς ταυτόχρονη εκτέλεση άλλων συναλλαγών με κοινές πηγές στοιχείων, εξασφαλίζει τη συνέπεια.

Απομόνωση (Isolation): ακόμα και αν εκτελούνται ταυτόχρονα πολλές συναλλαγές εξασφαλίζεται από το σύστημα ότι η κάθε συναλλαγή δε γνωρίζει για την ταυτόχρονη εκτέλεση της άλλης.

Αντοχή (Durability): αφού μια συναλλαγή ολοκληρωθεί με επιτυχία οι αλλαγές που επέφερε στις διάφορες πηγές δεδομένων είναι μόνιμες.

Αν και παραπάνω παρουσιάστηκαν τα γενικότερα πλαίσια των συναλλαγών σε ένα σύστημα, ιδιαίτερα σε αυτά των βάσεων δεδομένων, η μικρή αυτή εισαγωγή θέτει τις βάσεις κατανόησης των χαρακτηριστικών των επιχειρηματικών συναλλαγών.

Χρησιμοποιώντας ένα παράδειγμα περισσότερο τεχνικού περιεχομένου, οι διάφορες επιχειρηματικές πλατφόρμες όπως παραδείγματος χάριν η J2EE, προσφέρουν τη δυνατότητα διεκπεραίωσης συναλλαγών μέσα στη λειτουργικότητά τους. Συγκεκριμένα, η πλατφόρμα J2EE προσφέρει τη Java Transaction API (JTA)

που αποτελεί στην πραγματικότητα μια διεπαφή (API) για πρόσβαση στις δυνατότητες συναλλαγής του συμβατού λογισμικού.

### **5.1.3 Ενορχήστρωση-χορογραφία των διαδικασιών (Process Orchestration –Choreography)**

Όπως ήδη αναφέρθηκε η διάσταση των συναλλαγών (transactions) καθορίζει την αποτελεσματικότητα της λειτουργίας του συστήματος προδιαγράφοντας τα πλαίσια των συναλλαγών δηλαδή τους ρόλους των συμμετεχόντων, τα βήματα που ακολουθούνται και τις απαραίτητες λειτουργίες που φροντίζουν για τη συνέπεια των ανταλλασσόμενων δεδομένων. Ωστόσο δεν προδιαγράφεται ο τρόπος που οι επιμέρους διαδικασίες συντίθενται, ποιος εκτελεί και ελέγχει τη σύνθεση αυτών, ποιοι είναι οι κανόνες που διέπουν τον συντονισμό των επιμέρους διαδικασιών. Τα στοιχεία αυτά καλύπτονται από τη διάσταση της *ενορχήστρωσης-χορογραφίας των διαδικασιών (process orchestration –choreography)*. Οι προσεγγίσεις αυτές έχουν ήδη παρουσιαστεί αναλυτικά στην ενότητα 3.2.2 και για αυτό δε θα αναφερθούν εδώ με πολλές λεπτομέρειες. Εν συντομία δίνεται η διάκριση ανάμεσα στην ενορχήστρωση των διαδικασιών που επιβλέπεται από μια κεντρική αρχή και την χορογραφία που ορίζεται απευθείας ανάμεσα στους συνεργάτες. Περισσότερα θα δοθούν και στο επόμενο κεφάλαιο όπου εξετάζονται οι τεχνικές αυτές από τη σκοπιά των SOA και MDA.

Γενικότερα, ο συντονισμός και ο έλεγχος της διαδικασίας σύνθεσης των επιμέρους διαδικασιών είναι πολύ σημαντικό συστατικό για την επίτευξη ολοκληρωμένων συναλλαγών που εξυπηρετούν τους στόχους των εταίρων από τη στιγμή μάλιστα που υπάρχει ανάγκη για ad hoc σύνθεση διαδικασιών και σύναψη συνεργασιών αλλά και όταν η σύνθεση αυτή ορίζεται εξ' αρχής. Ο συντονισμός των διαδικασιών, είτε με τη μεσολάβηση κάποιας κεντρικής αρχής είτε χωρίς αυτή, αποτελεί βασικό στοιχείο αρχιτεκτονικής του συστήματος συναλλαγών αφού ουσιαστικά συνθέτει το περιβάλλον αλληλεπίδρασης των συμμετεχόντων και της εκτέλεσης των ροών εργασίας.

#### **5.1.4 Ενοποίηση (Integration)**

Μια επιπλέον διάσταση που δε μπορεί να παραλείψει κανείς είναι φυσικά αυτή της *ενοποίησης (integration)*. Η αρχιτεκτονική ενός συστήματος επιχειρηματικών συναλλαγών, που έχει ως απώτερο στόχο τη διασύνδεση ετερογενών κατά την πλειοψηφία τους πληροφορικών συστημάτων, έχει ανάγκη από ένα πλαίσιο ενοποίησης. Η ενοποίηση αυτή, όπως αναλύεται εκτενέστερα στην ενότητα 3.3, διαχωρίζεται σε ενοποίηση δεδομένων, σε ενοποίηση διαδικασιών και ενοποίηση υπηρεσιών. Καθεμιά από αυτές τις μορφές-προκλήσεις ενοποίησης αποσκοπούν στην επίτευξη της επιθυμητής διαλειτουργικότητας των πληροφοριακών συστημάτων και καθορίζουν σε μεγάλο βαθμό την ανάπτυξη κοινού κώδικα επικοινωνίας των συστημάτων. Οι αποφάσεις που σχετίζονται με αυτό το ζήτημα είναι ιδιαίτερα κρίσιμες για την αποτελεσματικότητα του συστήματος συναλλαγών και καθορίζουν σε μεγάλο βαθμό τη δομή του αλλά και την τεχνολογία που θα χρησιμοποιηθεί από αυτό. Μπορεί να θεωρηθεί ως ένα στάδιο μετάβασης από την αφαιρετική-λογική αναπαράσταση του συστήματος στην υλοποίησή του και κατά συνέπεια, είναι πολύ σημαντικό μια αρχιτεκτονική να εξασφαλίζει τη συγκεκριμένη διάσταση.

Έτσι, η ενοποίηση δύναται να οριστεί ως διάσταση αρχιτεκτονικής και τοποθετείται στα κατώτερα επίπεδα αφαιρετικότητας, εφόσον ορίζει τον τρόπο αναπαράστασης των δεδομένων του συστήματος, τον τρόπο ορισμού και κλήσης των διαδικασιών και γενικότερα την οργάνωση των υποδομών ώστε να είναι προσβάσιμες από τους εταίρους. Η χρήση κοινών προτύπων αναπαράστασης ή η εφαρμογή μονάδων μετατροπής (EAI μεσίτες, εξυπηρετητές εφαρμογών, προσαρμογείς, ESB) είναι κάποιοι από τους τρόπους επίτευξης της ενοποίησης.

#### **5.1.5 Ανταλλαγή των μηνυμάτων (Messaging)**

Στο κατώτερο επίπεδο της προδιαγραφής μιας αρχιτεκτονικής εξετάζονται θέματα που έχουν να κάνουν με την επικοινωνία και την ανταλλαγή μηνυμάτων μεταξύ των συμμετεχόντων. Το επίπεδο αυτό σχετίζεται περισσότερο με τη φυσική όψη αφού εξαρτάται από την υποδομή επικοινωνίας που χρησιμοποιείται (ουρές μηνυμάτων κλπ.) αλλά και με την όψη κώδικα αν λάβουμε υπόψη τα πρότυπα XML και SOAP που χρησιμοποιούνται. Σε αυτό το σημείο ορίζεται η διάσταση της *ανταλλαγής μηνυμάτων (messaging)*. Ο ορισμός της είναι απαραίτητος καθώς

αντιπροσωπεύει ένα βασικό στοιχείο του επιπέδου υλοποίησης μιας αρχιτεκτονικής συναλλαγών και μπορεί να διαφοροποιείται ανάλογα με τις απαιτήσεις των χρηστών και τις υποδομές που έχουν στη διάθεσή τους.

Σε περισσότερο τεχνικό επίπεδο, ο σύγχρονος τρόπος επικοινωνίας επιτυγχάνεται μέσω υποδομών RPC ενώ οι ασύγχρονη επικοινωνία ακολουθεί το MOM (βλ. §4.3.4). Βέβαια δεν απορρίπτονται οι περιπτώσεις όπου υλοποιείται σύγχρονη επικοινωνία χρησιμοποιώντας MOM και ασύγχρονη επικοινωνία βασιζόμενη σε RPC. Η τελευταία δομή ανταλλαγής μηνυμάτων είναι περισσότερο κατάλληλη όταν αναμένεται άμεση απάντηση ενώ η MOM προορίζεται κυρίως για ανεξάρτητη, ασύγχρονη επικοινωνία.

Ένα από τα πρωταρχικά πρότυπα ανταλλαγής μηνυμάτων υπήρξε το πρότυπο *αίτησης-απάντησης* (*request-response*) το οποίο υλοποιεί κατά κανόνα έναν τρόπο σύγχρονης επικοινωνίας. Σύμφωνα με αυτό η πηγή στέλνει ένα μήνυμα στον προορισμό ο οποίος απαντά αμέσως μόλις λάβει το μήνυμα.

Μια δεύτερη μορφή επικοινωνίας εκφράζεται μέσα από το πρότυπο ασύγχρονου τρόπου, το λεγόμενο *πυροδότησε-και-ζέχνα* (*fire-and-forget*). Στην προκειμένη περίπτωση είτε γίνεται αποστολή ενός μηνύματος σε ένα μόνο παραλήπτη, είτε μηνυμάτων πολύ-εκπομπής (*multi-cast*) ή ακόμα ευρείας εκπομπής (*broadcast*).

Ακόμα ένα ευρέως γνωστό και πιο σύγχρονο πρότυπο είναι το *δημοσίευσης-και-εγγραφής* (*publish-and subscribe*) το οποίο εισάγει έναν νέο τρόπο ασύγχρονης επικοινωνίας όπου καθένας μπορεί να λάβει ή να αποστείλει κάποιο μήνυμα. Εδώ ο *εγγραφόμενος* (*subscriber*) ενημερώνει τον *εκδότη* (*publisher*) σχετικά με το μήνυμα που επιθυμεί να λάβει και ο δεύτερος, εφόσον υπάρχει η ζητούμενη πληροφορία, την αποστέλλει σε όλους τους εγγεγραμμένους παραλήπτες του θέματος.

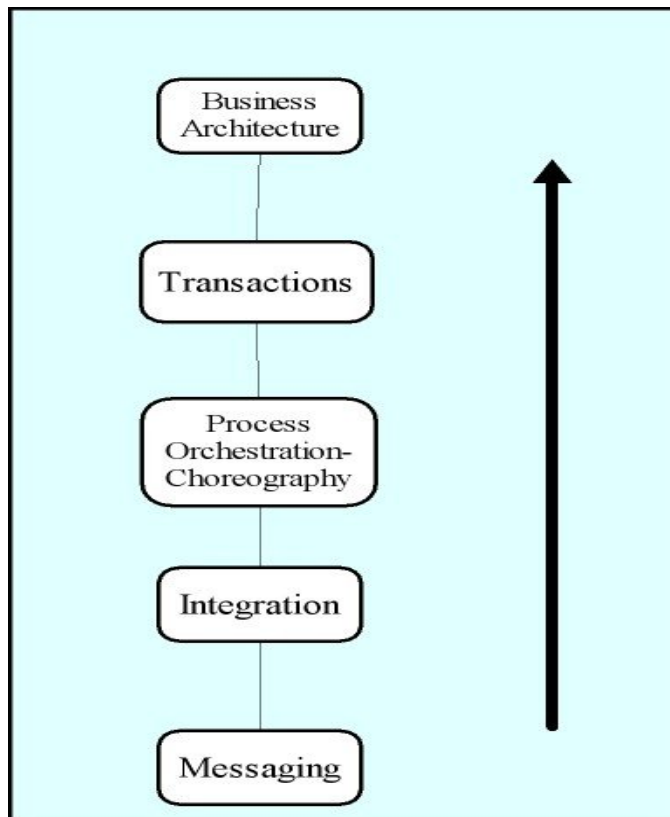
Άλλα παραδείγματα είναι τα εξής[11]:

*Επανάληψη κλήσεων και υπηρεσίες περιοδικών ελέγχων* (“*callbacks and polling services*”): όπου ο αποστολέας στέλνει το μήνυμα στον εξυπηρετητή (*server*) ο οποίος απαντά πιθανότατα ότι έλαβε το μήνυμα με ένα νέο. Στη συνέχεια ο εξυπηρετητής επεξεργάζεται το μήνυμα και στέλνει την απάντηση στον πελάτη (*client*) μέσω RPC/ORB. Υπάρχει πιθανότητα ο πελάτης περιοδικά να ελέγχει τον εξυπηρετητή για τη διαθεσιμότητα του αποτελέσματος.

*Προσομοιωμένες σύγχρονες υπηρεσίες με ουρές (“simulated synchronous services with queues”)*: όπου χρησιμοποιείται ένα σύστημα ουρών για αλληλεπίδραση όπως OS/390 με πρόσβαση MQSeries. Συνήθως εδώ χρησιμοποιούνται wrappers εξυπηρετητή-πελάτη (service-client wrappers, βλ. §4.3.2) που προσομοιώνουν τη σύγχρονη αλληλεπίδραση συνδυάζοντας δυο υποκείμενες ουρές, μια με αιτήσεις και μια με απαντήσεις. Αποδίδονται διακριτικά στα ζευγάρια αίτηση/απάντηση και αποτελεί ουσιαστικά ένα μέσο μεταφοράς χωρίς να επηρεάζει το σύστημα ανταλλαγής μηνυμάτων.

Συνοψίζοντας, διακρίνονται πέντε διαστάσεις που θα μπορούσαν να χρησιμοποιηθούν για την προδιαγραφή και την τεκμηρίωση μιας αρχιτεκτονικής σε ένα σύστημα ηλεκτρονικών συναλλαγών. Αυτή η διάκριση βέβαια δεν είναι και η μοναδική που θα μπορούσε να γίνει αφού η αρχιτεκτονική λογισμικού ορίζεται βάσει των εκάστοτε συνθηκών και περιορισμών. Έτσι δεν υπάρχει ένα συγκεκριμένο και ευρέως καθιερωμένο σύνολο από όψεις ή διαστάσεις αναφορικά με τις οποίες εξετάζεται η κάθε αρχιτεκτονική. Παρακάτω δίνεται ένα διάγραμμα με τις πέντε διαστάσεις που ορίστηκαν προηγουμένως και φαίνεται η πορεία ανάπτυξης της αρχιτεκτονικής από τα κατώτερα στρώματα υλοποίησης στα ανώτερα αφαιρετικά επίπεδα.

Να σημειωθεί εδώ ότι θα μπορούσαμε να ορίσουμε και ακόμα μια διάσταση που έχει να κάνει με την πρόσβαση και την ασφάλεια των συναλλαγών, καθώς αποτελεί μια ξεχωριστή λειτουργία του συστήματος με ειδικούς περιορισμούς και απαιτήσεις. Ωστόσο η ανάλυση αυτής της διάστασης ξεφεύγει από τα πλαίσια της παρούσας διπλωματικής.



Εικόνα 20-Πέντε διαστάσεις αρχιτεκτονικής

## **5.2 Αξιολόγηση των SOA και MDA σε κάθε Διάσταση**

Στην παρούσα ενότητα εξετάζονται οι επιμέρους διαστάσεις που διαμορφώνουν τα πλαίσια των αρχιτεκτονικών σε συστήματα ηλεκτρονικών συναλλαγών. Συγκεκριμένα, εφόσον έχουν δοθεί τα πλαίσια καθεμιάς από τις διαστάσεις αυτές, προσδιορίζονται οι εφαρμόσιμες λύσεις που ακολουθούνται από τις αρχιτεκτονικές Model Driven (MDA) και Service Oriented (SOA). Αν ακολουθήσουμε μια πορεία από τα κατώτερα και πιο τεχνικά επίπεδα προς τα ανώτερα και περισσότερα λογικά επιχειρηματικά επίπεδα –αντίθετη από αυτή που ακολουθήσαμε κατά τον ορισμό των διαστάσεων-αναφέρονται ξανά οι πέντε διαστάσεις:

- Ανταλλαγή μηνυμάτων (Messaging)
- Ενοποίηση (Integration )
- Ενορχήστρωση- Χορογραφία Διαδικασιών (Process Orchestration – Choreography)

- Συναλλαγές (Transactions)
- Επιχειρηματική αρχιτεκτονική (Business Architecture)

### **Ανταλλαγή μηνυμάτων (Messaging)**

Παρακάτω δίνονται ορισμένες λεπτομέρειες σχετικά με το επίπεδο ανταλλαγής μηνυμάτων επικοινωνίας. Τα στοιχεία αυτά είναι από τα πλέον καθιερωμένα πρότυπα επικοινωνίας που χρησιμοποιούνται επίσης από τις αρχιτεκτονικές που μας εξετάζονται εδώ.

Αναφορικά με τα πρωτόκολλα που χρησιμοποιούνται, επισημαίνεται ο πρωταρχικός ρόλος του SOAP (Simple Object Access Protocol). Το πρωτόκολλο αυτό χρησιμοποιείται για την ανταλλαγή πληροφοριών σε XML μορφή. Αξίζει να σημειωθεί σε αυτό το σημείο ότι η XML είναι ανεξάρτητη πλατφόρμας μεσολογισμικού και μπορεί να χρησιμοποιηθεί σε πολλές πλατφόρμες, εργαλεία επεξεργασίας και χειρισμού των δεδομένων όπως τα διάφορα APIs SAX και DOM. Ωστόσο το πρόβλημα της XML είναι η ίδια η απλότητά της καθώς πολλές φορές χρειάζεται να θεσπιστούν κοινά πρότυπα αναπαράστασης των δεδομένων σε ανώτερο αφαιρετικό επίπεδο (DTDs και Schemas).

Επανερχόμενοι στο πρωτόκολλο SOAP, αυτό μπορεί να χρησιμοποιηθεί πάνω από ποικίλα πρωτόκολλα μεταφοράς όπως HTTP, SMTP και FTP. Σε ανώτερο επίπεδο μπορεί να χρησιμοποιηθούν πρότυπα όπως το ebXML[12], που εξυπηρετούν τις ανάγκες αναπαράστασης του περιεχομένου των μηνυμάτων ιδιαίτερα για τις εμπορικές εφαρμογές. Ειδικότερα στα Web Services, το SOAP εξυπηρετεί την επικοινωνία ανάμεσα σε ετερογενή συστήματα. Επιπλέον, επισημαίνεται ότι το μέρος header (επικεφαλίδα) του μηνύματος SOAP μπορεί να υλοποιήσει διάφορες λειτουργίες που είναι απαραίτητες κατά τις ηλεκτρονικές συναλλαγές. Παραδείγματα αυτών είναι[55]:

- ✓ Οδηγίες επεξεργασίας που μπορούν να εκτελεστούν από τους μεσάζοντες της επικοινωνίας ή τον τελικό παραλήπτη
- ✓ Συνδυασμός πληροφοριών δρομολόγησης ή ροής εργασιών με το μήνυμα
- ✓ Υλοποίηση μέτρων ασφαλείας μαζί με το μήνυμα
- ✓ Κανόνες αξιόπιστης παράδοσης του μηνύματος

- ✓ Πληροφορίες χειρισμού του περιεχομένου και της συναλλαγής
- ✓ Πληροφορίες συσχετισμού μηνυμάτων αιτήματος με μηνύματα απάντησης

Γενικότερα, το πρωτόκολλο SOAP ήρθε ως αντικαταστάτης των RPC πρωτοκόλλων επιτρέποντας σε κλήσεις μεταξύ κατανεμημένων μονάδων να μπορούν να σειριοποιηθούν σε XML αρχεία, να μεταφερθούν και στην παραλαβή να αποσειριοποιηθούν στη καθαρή (native) μορφή που αναγνωρίζουν οι επιμέρους μονάδες.

Η αρχιτεκτονική SOA υπαγορεύει την ανταλλαγή μηνυμάτων σε μορφή αρχείου κειμένου και διευκολύνει τον μεγάλο όγκο φορτίου μεταφερόμενης πληροφορίας, τις λειτουργίες πολύπλοκων διαπροσωπειών και την περιορισμένη ένταση μεταφοράς μεταξύ των υπηρεσιών. Εν γένει το πρωτόκολλο επικοινωνίας SOAP εκπληρώνει τους στόχους της Service Oriented αρχιτεκτονικής για ανεξάρτητες μονάδες επικοινωνίας υποστηρίζοντας τη δημιουργία “ευφών”, επεκτάσιμων μηνυμάτων σε μορφή αρχείων κειμένου.

Για το συγκεκριμένο πρότυπο επικοινωνίας έχουν θεσπιστεί κάποια εργαλεία των Web Services και ειδικότερα:

*WS-BaseNotification*

*WS-BrokeredNotification*

*WS-Topics*

*WS-Eventing*

Σύμφωνα με την MDA αρχιτεκτονική τα διάφορα αφαιρετικά μοντέλα μπορούν να υλοποιηθούν σε πολλαπλές πλατφόρμες με τα κατάλληλα πρότυπα αντιστοιχήσεων. Στα πλαίσια αυτά ο όρος *πλατφόρμα* αναφέρεται στις τεχνολογικές και αρχιτεκτονικές λεπτομέρειες που δεν σχετίζονται άμεσα με τη βασική λειτουργικότητα της μονάδας λογισμικού. Κατά τη διάρκεια των διαφόρων συναλλαγών μεταξύ των εταιρών η λειτουργικότητα αυτή παραμένει αναλλοίωτη ανεξάρτητα από τον τρόπο που εκτελείται η διαδικασία για παράδειγμα από την COBRA, τα Enterprise Java Beans ή το SOAP. Οι τεχνικές λεπτομέρειες κατά συνέπεια αφαιρούνται από το μοντέλο PIM. Αν για παράδειγμα οι διάφορες διαδικασίες υλοποιηθούν μέσω SOAP θεωρείται ότι περνάμε πλέον στο platform-



specific μοντέλο. Αν και το PSM μοντέλο είναι και αυτό μια αναπαράσταση σε UML που δίνει τις επιχειρηματικές και τεχνολογικές σημασιολογίες, αντικατοπτρίζει τα συγκεκριμένα τεχνολογικά στοιχεία εκτέλεσης που εντάσσονται στην εκάστοτε πλατφόρμα. Ως εκ τούτου, η ανταλλαγή των πληροφοριών είναι καθαρά χαρακτηριστικό της πλατφόρμας που χρησιμοποιείται κάθε φορά και δεν μπορεί να προσδιοριστεί σε ενιαία βάση. Ωστόσο χρήσιμο είναι να αναφερθεί ότι το XML πρότυπο χρησιμοποιείται ευρέως για την αναπαράσταση των μηνυμάτων στις περισσότερες περιπτώσεις MDA εφαρμογών.

Περισσότερη έμφαση στην οδηγούμενη από τα μοντέλα αρχιτεκτονική δίνεται σε διαστάσεις ανώτερου επιπέδου οι οποίες πρόκειται να αναλυθούν στη συνέχεια.

### **Ενοποίηση (Integration)**

Για τη μελέτη της διάστασης της ενοποίησης θα επικεντρωθούμε στις τρεις κύριες όψεις αυτής. Όπως έχει αναλυθεί στην ενότητα 3.3, η ενοποίηση των επιχειρηματικών εφαρμογών εντοπίζεται σε επίπεδο δεδομένων (data integration), διαδικασιών (process integration) και υπηρεσιών (service integration). Κάθε ένα από αυτά εξετάζονται στη συνέχεια σε περιβάλλον service-oriented και model driven αρχιτεκτονικών.

Μεγάλη σημασία δίνεται ιδιαίτερα στη σημασιολογία και την αναπαράσταση των ανταλλασσόμενων δεδομένων και την ανάγκη αντιστοίχισης των προτύπων που ακολουθούνται ανάλογα με τον οργανισμό στον οποίο ανήκουν και τα σχήματα που υιοθετούν αυτοί. Ειδικότερα, η αρχιτεκτονική MDA παρέχει τη δυνατότητα αναπαράστασης των δεδομένων και της σημασιολογίας τους μέσα από τα μετα-μοντέλα δεδομένων. Εκμεταλλευόμενοι αυτά τα μοντέλα ορισμού των δεδομένων εμφανίζεται η ευκαιρία ορισμού δεδομένων που σχετίζονται με ποιοτικά χαρακτηριστικά (QoS) και χαρακτηριστικά που αποδίδονται στην οργάνωση των συναλλαγών ή σε θέματα πιστοποίησης και ασφαλείας. Η SOA αρχιτεκτονική από τον ορισμό της δεν παρέχει αυτή την ανεξάρτητη προτύπων αναπαράσταση των δεδομένων και κατά συνέπεια αντιμετωπίζει δυσκολίες σε επίπεδο ενοποίησης των δεδομένων (data integration). Επίσης δεν πρέπει να παραλειφθεί η αναφορά στα προβλήματα της επέκτασης των δεδομένων ή των αλλαγών που εμφανίζονται σε

ορισμένες περιπτώσεις. Μια ουδέτερη προτύπων και σχημάτων αναπαράσταση θα μπορούσε να λύσει πολλά από τα προβλήματα αυτά.

Όπως έχει ήδη αναφερθεί στην ενότητα ανάλυσης του προβλήματος της διαλειτουργικότητας (§ 3.3), η *ενοποίηση των διαδικασιών* δυο συναλλασσόμενων συστημάτων εξωτερικά ή εσωτερικά του οργανισμού αποτελεί καίριο ζήτημα. Οι δυο αρχιτεκτονικές καλούνται να αντιμετωπίσουν τις απαιτήσεις σύνθεσης των διαδικασιών των εταιρών για την πραγματοποίηση των ηλεκτρονικών συναλλαγών. Αναφερόμενοι για ακόμα μια φορά στον ορισμό που δίνουν οι [60] η ενοποίηση των διαδικασιών αποτελεί *μια διαδικασία αναπαράστασης των επιχειρηματικών διαδικασιών των εταιρών σε μοντέλα διαδικασιών ώστε να είναι δυνατή η σύνθεσή τους σε εκτελέσιμες ροές υπηρεσιών*.

Εξετάζοντας αρχικά τη Model Driven αρχιτεκτονική, έχει αναφερθεί ήδη ότι αυτή προσφέρει μια αφαιρετική αναπαράσταση της επιχειρηματικής λογικής του συστήματος, των πληροφοριών/δεδομένων, της λογικής παρουσίασης όσο και των διαδικασιών που λαμβάνουν χώρα κατά τη διάρκεια των συναλλαγών. Επιδιώκει έτσι να γεφυρώσει το χάσμα που υπάρχει ανάμεσα στην επιχειρηματική λογική και στις IT υποδομές του οργανισμού. Η αφαιρετική αναπαράσταση των επιχειρηματικών διαδικασιών που υποστηρίζονται από τα μοντέλα CIM και PIM καθιστούν εύκολη την ενοποίηση των διαδικασιών. Στα ίδια πλαίσια ορίζονται και κατανοούνται πολύ εύκολα οι διαδικασίες που λαμβάνουν χώρα για την πραγματοποίηση των συναλλαγών και επιπλέον, διευκολύνεται ιδιαίτερα η μετατροπή, προσθήκη ή αφαίρεση διαδικασιών αλλά και γενικότερα οποιαδήποτε αλλαγή στην επιχειρηματική προοπτική (business perspective) αποτυπώνεται γρήγορα και απλά στις αναπαριστώμενες σχέσεις. Οι μετατροπές που είναι απαραίτητες στην ενοποίηση των διαδικασιών πραγματοποιούνται σε αφαιρετικό επίπεδο με τη βοήθεια της MOF μέσω του XMI. Χαρακτηριστικά αναφέρεται ότι το μοντέλο είναι ο κώδικας και κατά αυτόν τον τρόπο αποφεύγονται οι λειτουργίες μετάφρασης και διερμηνείας (compiling και translation). Οι χρήστες δεν εξαρτώνται πλέον από καθαρό κώδικα ο οποίος έχει υψηλές απαιτήσεις συντήρησης.

Γενικότερα, υποστηρίζεται ότι μια διαδικασία δε μπορεί να θεωρηθεί αποκομμένη από το περιβάλλον της. Οι ιεραρχίες και ο συγχρονισμός των υπηρεσιών έχουν ανάγκη από επιχειρηματικούς κανόνες και περιορισμούς ενώ για τη σύνθεση των διαφόρων μονοπατιών εκτέλεσης με συγκεκριμένους απώτερους στόχους, είναι

απαραίτητο ένα αρχιτεκτονικό μοντέλο και γνώση των στοιχείων διασύνδεσης έτσι ώστε να αποφεύγεται η σχεδίαση ενός ξεχωριστού μοντέλου διαδικασίας για κάθε ροή ελέγχου χωρίς γνώση για σημεία διαλειτουργικότητας. Επίσης σημαντική κρίνεται και η ύπαρξη μιας αποθήκης υπηρεσιών που θα παρέχει πληροφορίες για την διαθεσιμότητα των διαδικασιών με κριτήριο το QoS. Τέλος, υπό την απουσία ενός μοντέλου είναι δύσκολο να χειριστεί κανείς τις συνέπειες των αλλαγών στην αρχιτεκτονική ή την είσοδο νέων παρόχων υπηρεσιών για τον σχεδιασμό των διαδικασιών[29].

Από την άλλη πλευρά, η Service Oriented αρχιτεκτονική μειονεκτεί στον τομέα της αφαιρετικής αναπαράστασης του συστήματος και των υπηρεσιών μέσω μοντέλων και χρησιμοποιεί πρότυπα περιγραφής τα οποία είναι λιγότερο αφαιρετικά. Στην περίπτωση της SOA μπορεί να θεωρηθεί ότι οι επιχειρηματικές διαδικασίες αποτελούν μηχανισμό εκμετάλλευσης των υπηρεσιών χαμηλότερου επιπέδου. Έτσι δίνεται μεγαλύτερη έμφαση στην εκτέλεση των υπηρεσιών και οι διαδικασίες είναι μέσο εκμετάλλευσης. Αξίζει να σημειωθεί ότι σύμφωνα με τους ειδικούς σε θέματα SOA, η επιτυχία της αρχιτεκτονικής εξαρτάται σε μεγάλο βαθμό από την προσαρμοστικότητα που εμφανίζει τόσο σε τεχνολογικές όσο και σε οργανωσιακές αλλαγές. Έτσι προκύπτουν πολύ σημαντικά ζητήματα όπως θέματα ταυτοποίησης των υπηρεσιών που προσφέρει η SOA καθώς και θέματα διασποράς των υπηρεσιών για την επιθυμητή ευελιξία της σύνθεσης των διαδικασιών η οποία ωστόσο μπορεί να στοιχίσει σε πολυπλοκότητα. Εφόσον τα ζητήματα αυτά λυθούν σε ικανοποιητικό βαθμό, μέσα από την σύνθεση των διαφόρων διαθέσιμων υπηρεσιών οι οργανισμοί είναι σε θέση να φέρουν εις πέρας πολύπλοκες επιχειρηματικές διαδικασίες μειώνοντας τον βαθμό πολυπλοκότητας, αυξάνοντας την ευελιξία τους σε αλλαγές και τη δυνατότητα προσαρμογής τους ανάλογα με τις μεταβαλλόμενες απαιτήσεις και συνθήκες.

Όσον αφορά στη διάσταση της *ενοποίησης των υπηρεσιών*, έτσι όπως ορίζεται από το πρόβλημα της διαλειτουργικότητας, η Service oriented αρχιτεκτονική θεωρείται ως η πλέον κατάλληλη. Από τον ορισμό της δίνει βάση στις υπηρεσίες οι οποίες μπορούν να ορίζονται αυτόνομα ως βασικές μονάδες λειτουργιών, παρέχει τη δυνατότητα δημοσίευσής τους και την ad hoc σύνθεση μέσα από την κλήση των άκρων που παρέχουν. Το χαρακτηριστικό αυτό είναι ιδιαίτερα σημαντικό στα μοντέρνα δυναμικά περιβάλλοντα που συνθέτουν οι εμπορικές ηλεκτρονικές

συναλλαγές με την αυξημένη συμμετοχή διαφόρων εταίρων και την πολυπλοκότητα των συναλλαγών. Στην Service Oriented αρχιτεκτονική η ευελιξία αποκτά μεγαλύτερες διαστάσεις και αναφέρεται ως : ευελιξία για την αντικατάσταση υπηρεσιών σε περίπτωση αποτυχίας, ευελιξία αναβάθμισης και αλλαγής των υπηρεσιών χωρίς να επηρεάζονται οι επιχειρηματικές διαδικασίες, ευελιξία αλλαγής των παρόχων των υπηρεσιών, ευελιξία επαναχρησιμοποίησης υπαρχουσών υπηρεσιών για νέες υπηρεσίες και προϊόντα. Υπό συνθήκες προσανατολισμού στις υπηρεσίες δημιουργούνται ευκαιρίες προμήθειας από τρίτους εντείνοντας τον ανταγωνισμό και κάνοντας αποτελεσματικές τις αλυσίδες αξίας ενώ παράλληλα μειώνονται τα κόστη.

Σε αντίθεση με την SOA, η MDA δεν υποστηρίζει την ιδέα των υπηρεσιών εφόσον η ίδια η UML δεν έχει αυτή τη δυνατότητα. Έτσι παραμένει σε μια σύνθετη μορφή διαδικασιών μειώνοντας την ευελιξία που παρέχουν οι αυτόνομες δομικές μονάδες υπηρεσιών. Επιπροσθέτως, αν και η MDA αρχιτεκτονική θεωρείται ότι συνταιριάζει την επιχειρηματική προοπτική (business perspective) με τις εταιρικές IT υποδομές, η SOA αρχιτεκτονική καλείται να εκφράσει τις επιταγές της επιχειρηματικής λογικής μέσα από τις υπηρεσίες-services που ενοποιεί.

Η υλοποίηση πλέον των εφαρμογών, όντας διαχωρίσιμη από τον ορισμό της αλληλουχίας και της σύνθεσης των επιχειρηματικών διαδικασιών, λαμβάνει χώρα στα κατώτερα επίπεδα αφαίρεσης. Στην μεν MDA αρχιτεκτονική υπάρχει πάλι το μοντέλο PSM που αναφέρεται σε συγκεκριμένη τεχνολογική πλατφόρμα- CORBA, Java2EE, .NET- ενώ αντίστοιχα στη SOA αρχιτεκτονική χρησιμοποιούνται adapters, wrappers, application servers για να δημιουργήσουν τα κατάλληλα περιβάλλοντα και αντιστοιχίσεις των ετερογενών προτύπων. Ένα λειτουργικό πλαίσιο υλοποίησης των αρχών της SOA προσφέρει ο Δίαυλος Επιχειρηματικών Υπηρεσιών (ESB) ο οποίος παρουσιάστηκε σε προηγούμενο κεφάλαιο. Στην περίπτωση αυτή προσφέρεται πλήρης ανεξαρτησία από το τεχνολογικό περιβάλλον του συστήματος και δεν χρειάζεται να οριστεί συγκεκριμένο PSM μοντέλο. Αποφεύγεται έτσι ο ορισμός μιας πληθώρας μοντέλων ανάλογα με την τεχνολογική πλατφόρμα εφόσον ο ESB λειτουργεί ως μεσολογισμικό (middleware) ανάμεσα στα (legacy) συστήματα συμβάλλοντας καθοριστικά στην πραγματοποίηση των συναλλαγών. Δεν επιτελεί μόνο τη λειτουργία της εκτέλεσης των εφαρμογών αλλά προσφέρει υπηρεσίες που

είναι απαραίτητες για την ολοκληρωμένη ενοποίηση των δεδομένων, υπηρεσιών και διαδικασιών.

### Ενορχήστρωση-Χορογραφία Διαδικασιών (Process Orchestration-Choreography)

Σε προηγούμενο κεφάλαιο έχει γίνει αναφορά στα πρωτόκολλα αλληλεπίδρασης των συνεργαζόμενων επιχειρηματικών διαδικασιών. Συγκεκριμένα γίνεται λόγος για τις έννοιες “Ενορχήστρωση”(Orchestration) και “Χορογραφία”(Choreography) που στοχεύουν στον ορισμό και τη διαχείριση των διαφόρων ενεργειών των επιχειρησιακών διαδικασιών ώστε να επιτευχθεί η συνεργασία μεταξύ των επιμέρους υπηρεσιών. Εν συντομία, επαναλαμβάνονται ορισμένα βασικά σημεία από το κάθε πρωτόκολλο και παράλληλα αυτά εξετάζονται στα πλαίσια των αρχιτεκτονικών SOA και MDA.

Στη Service Oriented αρχιτεκτονική, η τεχνική της *ενορχήστρωσης* περιγράφει πως οι διάφορες υπηρεσίες μπορούν να αλληλεπιδρούν σε επίπεδο ανταλλαγής μηνυμάτων με βάση πάντα την επιχειρηματική λογική και τη σειρά εκτέλεσης των αλληλεπιδράσεων αυτών. Η συνεργασία των υπηρεσιών αυτών επιβλέπεται από μια κεντρική αρχή. Ως αποτέλεσμα μπορεί να προκύψουν μεγάλης διάρκειας και με πολλά βήματα μοντέλα διεργασιών. Στην περίπτωση αυτή οι διαδικασίες που εκτελούνται αντιμετωπίζονται σε στενά πλαίσια από ένα από τα μέλη που λαμβάνουν χώρα στην διαδικασία. Γενικότερα διευκολύνεται ο έλεγχος των συνεργασιών, η ροή εργασίας μπορεί να αλλάξει ή να τροποποιηθεί εύκολα, ευνοείται η ενοποίηση των διαδικασιών και δημιουργούνται επιχειρήσεις που στηρίζονται σε συνεργασίες. Ωστόσο στα αρνητικά σημεία αυτού του πρωτοκόλλου έγκειται η εξάρτηση από μια μοναδική κεντρική αρχή η οποία αν καταρρεύσει αφήνει εκκρεμείς τις συναλλαγές αλλά και το πρόβλημα που μπορεί να δημιουργηθεί εάν αυξηθεί σημαντικά ο αριθμός των εμπλεκόμενων μελών καθιστώντας ιδιαίτερα περίπλοκο τον κεντρικό συντονισμό τους. Επίσης τίθεται το ζήτημα ασφάλειας των επιχειρηματικών διαδικασιών μιας και η αρχή αυτή ορίζει και έχει πρόσβαση στη σύνθεση των συναλλαγών.

Από την άλλη πλευρά, η τεχνική της *χορογραφίας* σχετίζεται με το σύνολο των ανταλλαγών μηνυμάτων, με ρόλους αλληλεπίδρασης και συμφωνίες που θεσπίζονται ανάμεσα στα συμμετέχοντα μέλη. Η χορογραφία ευνοεί τις συνεργασίες

και αντιμετωπίζει ισότιμα όλα τα συμμετέχοντα μέλη. Ορίζει πρόσθετες παρατηρήσιμες συμπεριφορές μέσα στο σύνολο των συμμετεχόντων και επιτηρεί την ανταλλαγή των μηνυμάτων που μπορεί να εμπλέκουν πολλές πηγές και άλλα μέλη. Κανένα μέλος δε θεωρείται ότι καθορίζει τον διάλογο. Σύμφωνα με το [55] η τεχνική της χορογραφίας διακρίνεται για τη δυνατότητα σύνθεσης, επαναχρησιμοποίησης και επεκτασιμότητας ενώ μπορεί να αυξήσει την οργανωτική ευελιξία και να διευκολύνει την ανακάλυψη. Οι οργανισμοί έχουν τη δυνατότητα να συμμετάσχουν σε διάφορες συνεργασίες που μπορούν να αλλάξουν ή να επεκτείνουν δυναμικά επιχειρηματικές διαδικασίες μέσω χορογραφιών.

Η ενορχήστρωση υλοποιείται από μια οικογένεια XML-based γλωσσών ορισμού διαδικασιών -XLANG, WSFL, BPML[4]- από τις οποίες η πιο αντιπροσωπευτική είναι η WS-BPEL (Business Process Execution Language for Web Services). Στην περίπτωση της χορογραφίας έχουμε τις εξής γνωστές γλώσσες: WSCL, WSCI, BPSS ενώ χρησιμοποιείται ιδιαίτερα η WS-CDL[67] (Web Services Choreography Description Language) η οποία ορίζει την κοινά παρατηρήσιμη συμπεριφορά όλων των συμμετεχόντων.

Εξετάζοντας τις παραπάνω έννοιες από τη σκοπιά της Model Driven αρχιτεκτονικής, παρατηρείται ότι αυτές δεν εφαρμόζονται εδώ. Οι υπηρεσίες που προσφέρει η SOA είναι από τη φύση τους αυτόνομες και δεν συντίθενται εξ' ορισμού τους σε κάποια πλαίσια επιχειρηματικών διαδικασιών. Έτσι έχουν ανάγκη από πρωτόκολλα που ορίζουν τις συνεργασίες τους και τις εντάσσουν σε αυτές. Όσον αφορά στη Model Driven Architecture, εδώ οι επιχειρηματικές διαδικασίες ορίζονται από τα ίδια τα μοντέλα του επιπέδου PIM και ως εκ τούτου δεν έχουν ανάγκη από ενορχήστρωση ή χορογραφία. Παρόλα αυτά θα μπορούσε να θεωρηθεί ότι για τη σύνθεση των διαδικασιών των εταίρων και την εξυπηρέτηση των συναλλαγών τους, είναι αναγκαία κάποια μορφή σύνθεσης των επιμέρους τμημάτων των διαδικασιών. Σε αυτή την περίπτωση ένα θεσπισμένο πρότυπο τύπου BPEL θα συνέβαλε σημαντικά.

Τέλος, σύμφωνα με την έρευνα πάνω σε συστήματα διαχείρισης ροών εργασίας μεταξύ οργανισμών[23] επιδιώκεται μια συμφωνία όλων των μελών σε παγκόσμια χορογραφία. Αυτό το μοντέλο στη συνέχεια θα χρησιμοποιηθεί για να προκύψουν τοπικές χορογραφίες για κάθε μέλος χωριστά, ενώ τέλος καθένα από αυτά

μπορούν να εκτελέσουν τη δική τους ενορχήστρωση που συμμορφώνεται με το παγκόσμια χορογραφία.

### Συναλλαγές (Transactions)

Σε προηγούμενο κεφάλαιο της διπλωματικής εργασίας αναλύθηκε ο ρόλος και κάποια ιδιαίτερα χαρακτηριστικά των ηλεκτρονικών συναλλαγών μεταξύ των επιχειρήσεων. Σε αυτό το σημείο αναφερόμαστε στην διεκπεραίωση των συναλλαγών με περισσότερους τεχνικούς όρους, βασιζόμενοι στα πρότυπα και τις ιδέες που έχουν καθιερωθεί. Παρακάτω δίνεται μια ερμηνεία του πλαισίου εκτέλεσης των συναλλαγών ως διάσταση των αρχιτεκτονικών SOA και MDA.

Στα πλαίσια της Service Oriented Architecture, και ειδικότερα για τα Web Services, σύμφωνα με το [51] το υποκείμενο περιβάλλον εκτέλεσης παρέχει τις απαιτούμενες δυνατότητες επεξεργασίας της συναλλαγής. Παρόλα αυτά πιθανότατα να χρειάζεται περιεχόμενο ανταλλαγής πληροφοριών έτσι ώστε να οι διάφορες υπηρεσίες να μπορούν να ομαδοποιηθούν σε μεγαλύτερες ομάδες λειτουργικότητας και να συντονιστούν ανάμεσα σε διαφορετικά περιβάλλοντα εκτέλεσης. Συγκεκριμένα για τα πρότυπα συναλλαγών που εφαρμόζονται στα Web Services, αναφέρεται ότι επεκτείνουν τις τεχνολογίες επεξεργασίας των συναλλαγών προσαρμόζοντας το 2-phase control commit πρωτόκολλο και ορίζοντας νέες επεκτάσεις πρωτοκόλλων για συναλλαγές που υποστηρίζουν την επαναφορά από αποτυχία και έχουν μεγάλη διάρκεια.

Τα πρότυπα στην περιοχή αυτή είναι τα εξής:

#### ***WS-Transactions:***

- **WS-AtomicTransactions[62]:** καθορίζει ACID συναλλαγές για 2-phase commit πρωτόκολλο και σύντομες εκτελέσεις.
- **WS-BusinessActivity:** καθορίζει συναλλαγές για πρωτόκολλα αβέβαιης εκτέλεσης (commit) και αναίρεσης που βασίζεται σε επανόρθωση (compensation-based undo) για μεγαλύτερης διάρκειας εκτελέσεις.
- **WS-Coordination[64]:** καθορίζει τον χειρισμό και τον συντονισμό για την οικογένεια πρωτοκόλλων WS-Transaction και των παραλλαγών τους.

### ***WS-Composite Application Framework (WS-CAF)[69]:***

- **WS-Context:** καθορίζει ένα σύστημα χειρισμού περιεχομένου για γενικευμένο περιεχόμενο.
- **WS-CoordinationFramework:** καθορίζει έναν τρόπο χειρισμού και έναν συντονιστή για τον προσδιορισμό του βασικού περιεχομένου και τα πρωτόκολλα συναλλαγών που ορίζονται στο WS-TransactionManagement.
- **WS-TransactionManagement:** ορίζει τρία πρωτόκολλα συναλλαγών για τον συντονιστή: ACID, επανόρθωσης συναλλαγών μεγάλης διάρκειας και χειρισμού των επιχειρηματικών διαδικασιών.

Ένα άλλο παράδειγμα αποτελεί το *Business Transaction Protocol (BTP)*[7] που προτείνει ο OASIS [40] και ορίζει ότι σε μια συναλλαγή οι αναγκαίες πηγές δεν ελέγχονται από ένα μοναδικό μέρος. Αντιθέτως τα διάφορα μέρη ελέγχουν τις δικές τους πηγές και συνεργάζονται υπό έναν συγκεκριμένο τρόπο με σκοπό την ολοκλήρωση της εργασίας που εξυπηρετεί η συναλλαγή. Σύμφωνα με το συγκεκριμένο πρότυπο, τα *atoms* είναι επιχειρηματικές συναλλαγές πάνω στις οποίες όλοι οι εταίροι πρέπει να συμφωνήσουν προτού να ολοκληρωθεί η συναλλαγή και τα *cohesions* που εξασφαλίζουν ότι ένας κεντρικός συντονιστής ελέγχει την κατάσταση κάθε μέρους της συναλλαγής και ακόμα και αν ένα μέλος δε μπορεί να ολοκληρώσει, ο συντονιστής αποφασίζει αν θα ολοκληρώσουν τα υπόλοιπα μέλη. Το BTP ορίζει ένα πρωτόκολλο ελαφριάς σύζευξης (*loosely coupled*) που εξασφαλίζει ότι οι πολλαπλές αλληλεπιδράσεις των Web Services διαδίδονται και κατανέμονται σωστά. Τα παραπάνω πρότυπα παρουσιάζουν μια κεντρικοποιημένη κατανομή καθώς στηρίζονται στην ιδέα ενός συντονιστή που εκτείνεται ευρέως αλλά και καθορίζουν ατομικές και επανρθώσιμες συναλλαγές μεγάλης διάρκειας.

Περνώντας στην Model Driven Architecture, σύμφωνα με το [39], σε ένα μετα-επίπεδο, το σύνολο των μοντέλων και των σχέσεων μπορούν να περιοριστούν ώστε να ικανοποιήσουν ένα σύνολο από απαιτήσεις συνέπειας. Εδώ δεν απαιτείται κάποιου είδους ατομικότητας ή βασικών στοιχείων συναλλαγής που θα εξασφάλιζαν ότι η συνέπεια διατηρείται σε όλα τα παρατηρήσιμα σημεία. Κύρια επιδίωξη είναι οι μεγάλης διάρκειας συναλλαγές όπου τα θέματα συνέπειας αναγνωρίζονται, υπόκεινται σε μια διαχείριση και τους δίνεται κάποια προτεραιότητα. Η ασυνέπεια μπορεί να υπάρχει και πρέπει να τη διαχειρίζεται κάποιος για μεγάλες περιόδους του



κύκλου ζωής του λογισμικού. Αυτή η διαδικασία ονομάζεται *διαχείριση ασυνέπειας (inconsistency management)*.

### **Επιχειρηματική Αρχιτεκτονική (Business Architecture)**

Όπως ήδη έχει αναλυθεί στην ενότητα διάκρισης των διαστάσεων μιας αρχιτεκτονικής ηλεκτρονικών συναλλαγών, στο ανώτερο επίπεδο αφαιρετικότητας της αρχιτεκτονικής αυτής βρίσκεται η επιχειρηματική αρχιτεκτονική. Στη συνέχεια αναφέρονται ξανά ορισμένα χαρακτηριστικά αυτής αλλά κυρίως εξετάζονται τα στοιχεία που καλύπτουν οι αρχιτεκτονικές SOA και MDA.

Πέρα από τα γενικότερα χαρακτηριστικά που περιγράφουν μια επιχειρηματική αρχιτεκτονική, μπορούμε να προσδιορίσουμε πώς η SOA ανταποκρίνεται συγκεκριμένα στις απαιτήσεις των επιχειρήσεων και πώς εξυπηρετεί τους σκοπούς αυτών. Οι εφαρμογές που είναι Service Oriented βασίζονται στην αρχή ότι όλες οι επιχειρήσεις έχουν κάποιο επιχειρηματικό σχεδιασμό. Αυτός ο επιχειρηματικός σχεδιασμός περιγράφει πως λειτουργεί η επιχείρηση- τις διαδικασίες που εκτελεί, την οργανωτική δομή των ανθρώπων και των κεφαλαίων της επιχείρησης, τους μεσοπρόθεσμους και μακροπρόθεσμους στόχους αυτής, τις επιρροές από την αγορά και την οικονομία που καθορίζουν πως η επιχείρηση επιτυγχάνει τους στόχους της όπως επίσης και τους κανόνες και τις πολιτικές που επηρεάζουν τη λειτουργία της επιχείρησης[35]. Τα θεμέλια του επιχειρηματικού σχεδιασμού είναι οι επιχειρηματικές διαδικασίες που αποτελούν μέρος μιας επιχείρησης και συντελούν στον τρόπο που η επιχείρηση λειτουργεί και απευθύνεται στους πελάτες, στις ευκαιρίες και στους εσωτερικούς και εξωτερικούς κινδύνους[31].

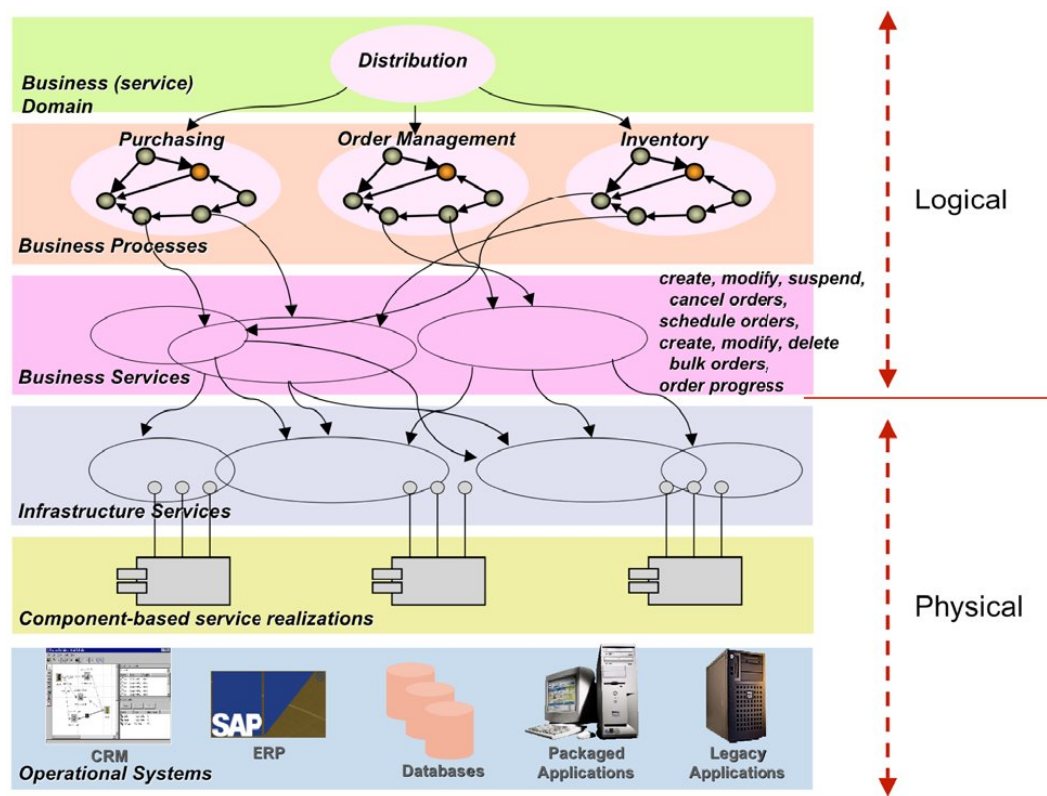
Ειδικά στην περίπτωση της Service Oriented αρχιτεκτονικής, οι υπηρεσίες χρησιμοποιούνται για να την υποστηρίξει η ανάπτυξη γρήγορων, μειωμένου κόστους και εύκολων συνθέσεων από κατανεμημένες εφαρμογές[34]. Η SOA είναι ένας λογικός τρόπος σχεδιασμού ενός συστήματος λογισμικού για την παροχή υπηρεσιών σε εφαρμογές τελικών χρηστών ή σε άλλες υπηρεσίες που είναι κατανεμημένες σε ένα δίκτυο, μέσω δημοσιευμένων και εντοπιζόμενων διαπροσωπειών. Ένα σωστά δομημένο σύστημα SOA μπορεί να ισχυροποιήσει ένα επιχειρηματικό περιβάλλον με ευμετάβλητη υποδομή και ένα περιβάλλον

επεξεργασίας παρέχοντας ανεξάρτητες, επαναχρησιμοποιήσιμες, αυτοματοποιημένες επιχειρηματικές διαδικασίες (ως υπηρεσίες).

Ακολουθώντας τη μεθοδολογία που επιτάσσει η ανάπτυξη που βασίζεται στις υπηρεσίες (service-based), πολύ σημαντικό είναι κανείς να ορίσει, να κατασκευάσει, να τελειοποιήσει και να εδραιώσει μη σταθερές επιχειρηματικές διαδικασίες από εσωτερικά ή εξωτερικά διαθέσιμα Web Services. Αυτό που απαιτείται είναι μια μεθοδολογία που βασίζεται στην αρχιτεκτονική των υπηρεσιών και η οποία επιτρέπει στις επιχειρήσεις να σχεδιάζουν και να αναπτύσσουν αποτελεσματικά υπηρεσίες όπως επίσης μπορεί να ενσωματώνει ευκολότερα τις αλλαγές σε εφαρμογές που βασίζονται στις υπηρεσίες σύμφωνα με τις αλλαγές του επιχειρηματικού σχεδιασμού.

Οι επιχειρηματικές διαδικασίες και υπηρεσίες σε υπηρεσιοστρεφείς εφαρμογές εφαρμόζονται ως μονάδες οικονομικές, λειτουργικές και δεδομένων τα οποία παρέχονται από πηγές όπως ERP, βάσεις δεδομένων, CRM και άλλα συστήματα. Κατά συνέπεια είναι χρήσιμο να ξεχωρίζουμε ανάμεσα σε επιχειρηματικές διαδικασίες και επιχειρηματικές υπηρεσίες, ως το λογικό κομμάτι του κύκλου ανάπτυξης των υπηρεσιών, ενώ το φυσικό κομμάτι του κύκλου αυτού αποτελεί το υπόβαθρο των υλοποιήσεων των υπηρεσιών και των μονάδων που αντιστοιχεί τις λογικές υπηρεσίες στις υπάρχουσες πηγές[36].

Παρακάτω δίνεται μια αναπαράσταση το διαφόρων επιπέδων μιας SOA αρχιτεκτονικής. Το λογικό επιχειρηματικό κομμάτι περιλαμβάνει επιχειρηματικές διαδικασίες και υπηρεσίες όπως μια διαδικασία χειρισμού παραγγελίας που παρέχει επιχειρηματικές υπηρεσίες όπως μετατροπή, διακοπή, ακύρωση, εντολές ερωτημάτων κλπ. Οι υπηρεσίες του υπόβαθρου είναι συνήθως υπηρεσίες διαχείρισης και επίβλεψης που περιέχονται σε ένα container και στο αντίστοιχο υπόβαθρο όπως αυτές που παρέχουν τεχνική χρησιμότητα: πρόσβαση, ασφάλεια, δυνατότητα συναλλαγής και πιστοποίησης αλλά και αυτές που χειρίζονται τις πηγές. Αυτές οι υπηρεσίες θέτουν το τεχνικό υπόβαθρο διευκολύνοντας την ενοποίηση των υπηρεσιών (που ήδη έχει αναφερθεί προηγουμένως) με την εισαγωγή ευφυούς δρομολόγησης άλλων μηχανισμών μετατροπών που αποτελούν συνήθως μέρος του Enterprise Service Bus. Όσον αφορά στο μέρος της υλοποίησης, παρέχονται όλες οι οικονομικές και λειτουργικές διαδικασίες αλλά και τα δεδομένα από πηγές όπως ERP, CRM, βάσεις δεδομένων και άλλα συστήματα.



Εικόνα 21-The Web Services Development Life Cycle hierarchy[37]

Τέλος ένα πολύ σημαντικό πεδίο έρευνας που προσφέρει η Service Oriented αρχιτεκτονική έχει να κάνει με τις δυνατότητες σε επίπεδο διαχείρισης των υπηρεσιών. Αυτές οι δυνατότητες που προσφέρονται είναι αυτόνομες, ικανοποιούν απαιτήσεις των IT συστημάτων και λύνουν προβλήματα με ελάχιστη ανθρώπινη παρέμβαση. Αυτές οι λειτουργίες εκτελούν αναδρομικούς ελέγχους στο σύστημα, συλλέγουν λεπτομέρειες από το σύστημα και αναπτύσσουν δράσεις[17][22][32]. Παραδείγματα υπηρεσιών (autonomic services) που εκτελούν αυτούς τους ελέγχους είναι οι εξής:

- Αυτό-οριζόμενες υπηρεσίες διαχείρισης (Self-configuring management services): ορίζονται από μόνες τους ώστε να προσαρμόζονται εύκολα σε διαφορετικά περιβάλλοντα στα οποία μπορούν να εγκατασταθούν και να λειτουργήσουν για βελτιστοποίηση των λειτουργιών τους.
- Αυτό-προσαρμοζόμενες υπηρεσίες διαχείρισης (Self-adapting management services): υπηρεσίες που δυναμικά προσαρμόζονται σε αλλαγές στο περιβάλλον χρησιμοποιώντας διάφορες πολιτικές. Οι αλλαγές αυτές μπορεί να

περιλαμβάνουν δημιουργία νέων στιγμιοτύπων, κατάργηση των ήδη υπαρχόντων ή αλλαγή στα χαρακτηριστικά του συστήματος.

- *Αυτό-θεραπευόμενες υπηρεσίες διαχείρισης (Self-healing management services)*: οι υπηρεσίες αυτές μπορούν να ανακαλύψουν, να διαγνώσουν και να αντιδράσουν σε διασπάσεις. Ανακαλύπτουν δυσλειτουργίες του συστήματος και ξεκινούν διορθωτική δράση βασιζόμενη σε πολιτικές χωρίς να επηρεάζουν το IT περιβάλλον. Οι διορθωτικές διαδικασίες αφορούν σε ένα προϊόν που αλλάζει την κατάστασή του ή προκαλεί αλλαγές σε άλλες μονάδες του περιβάλλοντος.
- *Αυτό-βελτιούμενες υπηρεσίες διαχείρισης (Self-optimizing management services)*: αυτές μπορούν να ελέγξουν και να ρυθμίσουν τις πηγές αυτόματα, να ρυθμίσουν τη λειτουργία τους ώστε να ικανοποιήσουν τις ανάγκες των χρηστών ή της επιχείρησης. Έχουν τη δυνατότητα να χειριστούν τις πηγές με τέτοιο τρόπο ώστε να αυξήσουν τη χρησιμότητα, να μειώσουν το υπολογιστικό βάρος.
- *Αυτό-προστατευόμενες υπηρεσίες διαχείρισης (Self-protecting management services)*: αυτή η κατηγορία υπηρεσιών βοηθά στον εντοπισμό, την αναγνώριση και την προστασία του συστήματος ενάντια σε απειλές πχ. από πρόσβαση χωρίς δικαίωμα, επίθεση ιών κλπ. Οι υπηρεσίες αυτές ενεργοποιούν διορθωτικές κινήσεις και ενισχύουν την άμυνα και τις privacy πολιτικές των επιχειρήσεων.

Ακόμα μια μεγάλη πρόκληση για τη Service Oriented αρχιτεκτονική είναι η προσθήκη ευαισθησίας στο περιεχόμενο και η έμφαση στο προφίλ του χρήστη στις υπηρεσίες. Αυτό ωστόσο έχει ανάγκη και από μετα-μοντέλα και μετα-πληροφορίες για την αναπαράσταση των πληροφοριών που χρησιμοποιούνται. Σε αυτή την περίπτωση θα ήταν ιδιαίτερα χρήσιμες οι ιδέες της model driven αρχιτεκτονικής που παρουσιάζονται στη συνέχεια.

Η Οδηγούμενη από τα Μοντέλα (Model Driven) Αρχιτεκτονική προσφέρει το πλεονέκτημα της μοντελοποίησης των επιχειρηματικών διαδικασιών στα επίπεδα CIM και PIM ενώ μεταθέτει την υλοποίηση της εφαρμογής στο επίπεδο PSM. Βασίζεται ιδιαίτερα στα μετα-δεδομένα και τα μετα-μοντέλα που περιέχουν όλη την απαραίτητη πληροφορία για τη φύση των δεδομένων, τη δομή αυτών όπως επίσης και

τις διαδικασίες που εφαρμόζει η επιχείρηση. Όλοι οι ρόλοι, οι κανόνες, οι στόχοι, τα χαρακτηριστικά των συνεργασιών των επιχειρήσεων καλούνται να αναπαρασταθούν σε μοντέλα της MDA.

Ειδικότερα τα ανώτερα αφαιρετικά επίπεδα (CIM και PIM) προσφέρουν τη μοντελοποίηση των επιχειρηματικών διεργασιών και αναπαριστούν την επιχειρηματική αρχιτεκτονική (business architecture). Σε αυτά συγκεντρώνονται όλες οι διαδικασίες, συντίθενται και ορίζονται οι ροές εργασιών βάσει των κανόνων, των πολιτικών, των στόχων που ακολουθούν οι επιχειρήσεις. Έτσι οι επιχειρηματικές διαδικασίες αποκτούν αξία, μπορούν να εφαρμοστούν σε διαφορετικά τεχνολογικά υπόβαθρα και να αλληλεπιδράσουν με άλλες διαδικασίες ετερογενών συστημάτων. Πολύ σημαντικό ρόλο σε αυτές τις δραστηριότητες έχουν τα μετα-μοντέλα δεδομένων και διεργασιών καθώς διευκολύνεται η αφαιρετική αναπαράσταση της λειτουργίας της επιχείρησης, η δυναμική σύνθεση των ροών και η ενοποίηση των διεργασιών χωρίς να παρουσιάζονται προβλήματα διαλειτουργικότητας. Αυτή η αφαιρετική αναπαράσταση αποδεικνύεται ιδιαίτερα χρήσιμη στις περιπτώσεις αλλαγών των επιχειρηματικών πρακτικών όπου πολύ απλά η αλλαγή αυτή γρήγορα και με μειωμένο κόστος μεταφέρεται σε επίπεδο μοντέλου. Ουσιαστικά, η Model Driven αρχιτεκτονική θέτει τα θεμέλια για την ευρεία και απρόσκοπτη χρήση και διαχείριση σε σημασιολογικό επίπεδο πάνω στο οποίο στηρίζονται οι επιχειρηματικές εφαρμογές και τα οποία έχουν ανάγκη οι συναλλαγές.

Επιπλέον σε πρακτικότερο επίπεδο η Οδηγούμενη από τα Μοντέλα Αρχιτεκτονική παρέχοντας τα διάφορα μοντέλα, προσεγγίζει την “επιχειρηματική γλώσσα” (business language) χαρακτηριστικό το οποίο συντελεί στη γεφύρωση του κενού μεταξύ επιχειρήσεων και IT, CEO και CIO, σύμφωνα με το[18].

Ένα βασικό ωστόσο θέμα που παρουσιάζεται στην MDA αφορά στο ότι εδώ δεν ορίζονται οι διάφορες αυτόνομες υπηρεσίες που υπάρχουν στη SOA. Παρόλο που οι διαδικασίες είναι αφαιρετικά δοσμένες, εντούτοις δεν εξυπηρετείται η τάση που επιτάσσει η σημερινή υπηρεσιο-κεντρική επιχειρηματική δραστηριότητα.

## 5.3 Σύγκριση SOA και MDA και τελικά συμπεράσματα

Στα πλαίσια της ανάγκης ύπαρξης μιας ενιαίας αρχιτεκτονικής πλατφόρμας για την εξυπηρέτηση των ηλεκτρονικών συναλλαγών και την επίτευξη της επιδιωκόμενης διαλειτουργικότητας, τίθεται το ζήτημα της γενικής σύγκρισης των αρχιτεκτονικών SOA και MDA που παρουσιάστηκαν προηγουμένως. Η σύγκριση αυτή αποσκοπεί στο να ξεκαθαρίσει χαρακτηριστικά της καθεμιάς αρχιτεκτονικής που δομούν το καταλληλότερο υπόβαθρο των ηλεκτρονικών συναλλαγών μεταξύ ετερογενών συστημάτων. Για αυτό το σκοπό οι δυο προαναφερθείσες αρχιτεκτονικές εξετάζονται βάσει κριτηρίων που θεωρούνται σημαντικά για την αποτελεσματικότητα των συναλλαγών και τη διαλειτουργία των συστημάτων. Ειδικότερα, βάση της σύγκρισης αποτελεί η ανάλυση των αρχιτεκτονικών διαστάσεων που έχει προηγηθεί και σύμφωνα με αυτή προκύπτουν τα σημεία που δίνονται παρακάτω.

Ακολουθώντας την ανοδική πορεία από τα κατώτερα προς τα ανώτερα επίπεδα αρχιτεκτονικής, στο επίπεδο **ανταλλαγής μηνυμάτων (messaging)** οι διαφορές των αρχιτεκτονικών δεν είναι ιδιαίτερα έντονες. Η μεταφορά των μηνυμάτων εκτελείται πάνω από τα γνωστά πρωτόκολλα επιπέδου εφαρμογής HTTP, SMTP και FTP. Το SOAP πρωτόκολλο έχει θεσπιστεί ευρέως ως το πλέον κατάλληλο πρότυπο ανταλλαγής μηνυμάτων σε μορφή XML ενώ ήδη αναπτύσσονται διάφορες υπηρεσίες που διευκολύνουν τη διαχείριση των ανταλλασσόμενων μηνυμάτων (*WS-BaseNotification*, *WS-BrokeredNotification*, *WS-Topics*, *WS-Eventing*[65]) για την Προσανατολισμένη σε Υπηρεσίες Αρχιτεκτονική. Στην περίπτωση της Οδηγούμενης από τα Μοντέλα Αρχιτεκτονικής, οι ανταλλαγή των μηνυμάτων εξαρτάται από τη τεχνολογική πλατφόρμα και τα πρότυπα που αυτή υποστηρίζει. Γενικότερα, παρατηρείται ότι η SOA αρχιτεκτονική βασίζεται σε ευρέως διαδεδομένα πρότυπα και διευκολύνει την ανταλλαγή αλλά και τη διαχείριση των μηνυμάτων επικοινωνίας ενώ στην MDA η υλοποίηση εξαρτάται από την εκάστοτε τεχνολογική πλατφόρμα. Στην τελευταία περίπτωση η υλοποίηση μπορεί να μεταφερθεί σε άλλη πλατφόρμα εφόσον υπάρχουν τα εξαρτώμενα από την πλατφόρμα μοντέλα PSM στα οποία αποδίδονται οι λεπτομέρειες υλοποίησης των ανεξαρτήτων πλατφόρμας μοντέλων PIM.

Βασιζόμενοι κυρίως στα κριτήρια που σχετίζονται με τα τρία προβλήματα ενοποίησης- **δεδομένων, διαδικασιών και υπηρεσιών (data, process, και service integration)**- αναφέρθηκαν πρωτίτερα χαρακτηριστικά των δυο αρχιτεκτονικών και κρίθηκε η καταλληλότητα της καθεμιάς. Από την ανάλυση που προηγήθηκε προκύπτει ότι η SOA αρχιτεκτονική εξυπηρετεί τις απαιτήσεις σε ευρέως επαναχρησιμοποιούμενες υπηρεσίες με ιδιότητες όπως μικρή πολυπλοκότητα, αυτονομία, δυναμική σύνθεση και ευκολία συντήρησης, απόκρυψη των υπηρεσιών μέσω διαπροσωπειών κοκ. Ωστόσο με αιώτερο σκοπό τη δημιουργία αποδοτικών μοντέλων υπηρεσιών, υπάρχει ανάγκη τεχνικών μοντελοποίησης SOA που θα συμβάλλουν στο σχεδιασμό υπηρεσιών υποστηρίζοντας πληροφορίες εντοπισμού και σύνδεσης (*binding*), ιδιότητες σύνθεσης υπηρεσιών, χαρακτηριστικά προσανατολισμού στο νόημα και τα δεδομένα ή τα προφίλ των χρηστών και άλλες ιδιότητες. Η MDA μπορεί να επιδράσει καθοριστικά σε αυτό το ζήτημα δανείζοντας την ιδέα της αφαιρετικής αναπαράστασης δεδομένων, διαδικασιών, παρουσίασης και ειδικότερα μπορούν να αναφερθούν ορισμένα σημεία που δεν υποστηρίζονται πλήρως από τη SOA ως προς το πώς μπορούν να υλοποιηθούν. Αυτά χρειάζονται να υποστηριχθούν από μοντέλα όπως:

- συμβόλαια υπηρεσιών (*service contracts*)- περιγράφουν αλληλεπιδράσεις, ρόλους, κανόνες
- ορισμοί υπηρεσιών (*service specifications*)- καθορίζουν λειτουργίες υπηρεσιών, διαπροσωπείες υπηρεσιών που χρειάζονται ή προσφέρονται
- δεδομένα υπηρεσιών (*service data*) - προσφέρουν ορισμούς της ανταλλασσόμενης δομημένης πληροφορίας
- κλήση υπηρεσιών και χειρισμός γεγονότων (*service invocation and event handling*) – υποστηρίζουν σύγχρονη και ασύγχρονη κλήση υπηρεσιών
- παράμετροι υπηρεσιών (*service parameters*)- οι οποίες περνούν ανάμεσα σε προμηθευτές και καταναλωτές

Επίσης σύμφωνα με τον OMG[41] κρίνεται απαραίτητη η μοντελοποίηση για τη σύνθεση υπηρεσιών από άλλες με την εισαγωγή αποθηκών μεταμοντέλων και μεταδεδομένων, τον καθορισμό καναλιών υπηρεσιών για χρήσεις υπηρεσιών μεταξύ καταναλωτών και προμηθευτών, την τυποποίηση και επέκταση υπηρεσιών, την

υποστήριξη ανταλλαγής μοντέλων υπηρεσιών μέσω XMI (XML Metadata Interchange)[46].

Από τη μελέτη της διάστασης της **ενορχήστρωσης και χορογραφίας των διαδικασιών (process orchestration και choreography)**, προέκυψε ότι η SOA αρχιτεκτονική παρέχει τη δυνατότητα της σύνθεσης κεντρικά ή μη αντίστοιχα των επιχειρηματικών διεργασιών. Κάτω από αυτές τις συνθήκες εξυπηρετεί τους στόχους πολλών επιχειρήσεων για δυναμική σύνθεση των εμπορικών τους συναλλαγών. Η σύνθεση αυτή στηρίζεται σε κοινώς αποδεκτά πλαίσια όπως WS-BPEL ή WS-CDL [67]. Αντίθετα, για την MDA δεν γίνεται κάποια ιδιαίτερη αναφορά λόγω του ότι από τη φύση της χρησιμοποιεί μοντέλα επιμέρους διεργασιών τα οποία συνθέτει υπό μια μορφή ενορχήστρωσης. Οι ιδιότητες της σύνθεσης μέσω των πρωτοκόλλων ενορχήστρωσης και χορογραφίας προσδίδουν χαρακτηριστικά δυναμικότητας στη σύνθεση των εμπορικών συναλλαγών που αποτελούν πολύ βασική επιδίωξη του σύγχρονου ευμετάβλητου επιχειρηματικού περιβάλλοντος.

Σε επίπεδο **συναλλαγών (transactions)**, η SOA αρχιτεκτονική υποστηρίζει πρότυπα που φροντίζουν για την διενέργεια μεγάλης διάρκειας συναλλαγών που εκτείνονται σε πολλά ετερογενή περιβάλλοντα αυξάνοντας την ευελιξία των συναλλαγών. Βάσει αυτών των προτύπων εξασφαλίζεται η συνεπής ολοκλήρωση των συναλλαγών ή αντίθετα η επανάκαμψη σε περίπτωση προβλήματος αλλά επίσης προσφέρεται ένα σύνολο από υπηρεσίες ελέγχου της διαδικασίας της συναλλαγής. Η MDA από την άλλη πλευρά δε θεσπίζει ρητά κάποια γενικά πρότυπα χειρισμού των συναλλαγών. Η διατήρηση της συνέπειας των συναλλαγών εξαρτάται από τα ίδια τα μοντέλα και τις σχέσεις και καθορίζεται σε στενότερα πλαίσια από έναν κεντρικό διαχειριστή.

Όσον αφορά στη διάσταση της **επιχειρηματικής αρχιτεκτονικής (business architecture)**, η Model Driven αρχιτεκτονική, που προτείνει ο OMG[41], στοχεύει σε μια ανοιχτή προσέγγιση προσδιορισμού και ανάπτυξης εφαρμογών η οποία αποδεσμεύει την επιχειρηματική λογική και τη λογική της εφαρμογής από την υποκείμενη τεχνολογική πλατφόρμα. Κεντρικός άξονας της αρχιτεκτονικής αυτής αποτελούν οι επιχειρηματικές διαδικασίες που λαμβάνουν χώρα στις ηλεκτρονικές συναλλαγές. Στα ίδια περίπου πλαίσια κινείται και η Service Oriented αρχιτεκτονική



παρέχοντας τρόπους ορισμού, ανάπτυξης και δυναμικής κλήσης ανεξάρτητων υπηρεσιών. Πιο συγκεκριμένα ωστόσο, η αρχιτεκτονική SOA επικεντρώνεται στην χρήση των ανεξάρτητων μονάδων λειτουργικότητας, όπως θα μπορούσαν να χαρακτηριστούν οι υπηρεσίες. Κατά αυτό τον τρόπο η SOA εντάσσεται στη γενικότερη τάση που υιοθετείται από το διαδίκτυο σύμφωνα με την οποία οι δομικές μονάδες- υπηρεσίες είναι αυτόνομες και εύκολα καλούμενες μέσω γνωστών διαπροσωπειών από οποιαδήποτε επιχειρηματική διαδικασία. Ως εκ τούτου, η δυνατότητα επαναχρησιμοποίησης των εφαρμογών, η διαθεσιμότητα, η ευκολία, η δυνατότητα άμεσης εφαρμογής και η απρόσκοπτη επέκταση των εφαρμογών αποτελούν ορισμένα βασικά χαρακτηριστικά που μπορούν να αποδοθούν και στις δύο αρχιτεκτονικές. Εξίσου σημαντική είναι η συμβολή της SOA αρχιτεκτονικής στην κάλυψη των αναγκών σε υπηρεσίες παρακολούθησης, συμμόρφωσης, διαχείρισης, σύναψης συμφωνιών, συλλογής στατιστικών απόδοσης και μια ποικιλίας άλλων μη-λειτουργικών απαιτήσεων. Ωστόσο για την υλοποίηση των υπηρεσιών αυτών είναι απαραίτητη η προσθήκη περιεχομένου στις συναλλαγές. Τα βασικά αυτά ζητήματα μας απασχολούν στο επόμενο κεφάλαιο.

Παρόλα αυτά βασικό σημείο διαφοροποίησης είναι ο προσανατολισμός προς τις υπηρεσίες ή τα μοντέλα των διαδικασιών και δεδομένων για την SOA και την MDA αντίστοιχα. Πολλές φορές ωστόσο οι αρχιτεκτονικές αυτές δεν εξυπηρετούν στο έπακρο τους στόχους που θέτουν και για αυτό κρίνεται αναγκαία η σύγκριση των λειτουργιών που επιτελούν και σε ορισμένες περιπτώσεις ο δανεισμός στοιχείων της μίας ή της άλλης αρχιτεκτονικής.

Στη συνέχεια παρουσιάζεται ο πίνακας που συνοψίζει τα σημεία που αναφέρθηκαν στην ανάλυση και τη σύγκριση που προηγήθηκαν.

**Πίνακας 1-Σύγκριση Διαστάσεων σε SOA και MDA**

<b>Διαστάσεις Αρχιτεκτονικών</b>	<b>SOA</b>	<b>MDA</b>
<b>Messaging</b>		
Χρήση ευρέως διαδεδομένων προτύπων	✓	✓
SOAP, XML, HTTP, SMTP, FTP	✓	✓
WS-* πρότυπο	✓	x

<b>Integration</b>		
Αφαιρετική αναπαράσταση διαδικασιών	x	✓
Ανεξαρτησία από τεχνολογική πλατφόρμα- εύκολη μεταφορά ανάμεσα σε διαφορετικές πλατφόρμες	✓	✓ (PSM)
Παροχή ενιαίου τεχνικού υπόβαθρου ενοποίησης ( enterprise service bus)	✓	x
Δυνατότητα εντοπισμού και ad hoc σύνθεσης διαδικασιών ή υπηρεσιών	✓	x
Ενοποίηση σε MOF μέσω XMI	x	✓
<b>Process Orchestration-Choreography</b>		
BPEL, XLANG, WSFL, BPML, WS-BPEL(orchestration) WSCL, WSCI, BPSS WS-CDL(choreography)	✓	x
<b>Transactions</b>		
Διαχείριση συναλλαγών μέσω προτύπων: WS-Transactions, WS-CAF	✓	x
Εξάρτηση από τα μοντέλα και τις σχέσεις	x	✓
<b>Business Architecture</b>		
Αφαιρετική αναπαράσταση συστήματος- διαδικασιών	x	✓ (CIM,PIM)
Αυτονομία των υπηρεσιών- ευκολία σύνθεσης ετερογενών-εξωγενών υπηρεσιών	✓	x
Διαφάνεια λειτουργικότητας των υπηρεσιών- Εύκολη κλήση μέσω διαπροσωπειών	✓	✓
Autonomic Services	✓	x
Αλλαγές του επιχειρηματικού περιβάλλοντος αποτυπώνονται άμεσα στις εφαρμογές και αντίστροφα μέσω των αφαιρετικά αναπαριστώμενων σχέσεων	x	✓
Χρήση μεταδεδομένων για τον προκαθορισμένο τρόπο περιγραφής των λειτουργιών, των δεδομένων και των ποιοτικών χαρακτηριστικών της συναλλαγής- Δυνατότητα αναπαράστασης δεδομένων- μοντέλα δεδομένων και	x	✓

σημαιολογία- καθολική αντίληψη		
Χρήση συμβολαίων για τις ηλεκτρονικές συναλλαγές	✓	✓
Προσέγγιση του business perspective με τις IT software services	✓	
Μοντέλα επιχειρησιακού χαρακτήρα που γεφυρώνουν το χάσμα business-IT		✓
Διαθεσιμότητα, άμεση εφαρμογή, επεκτασιμότητα	✓	✓

Λαμβάνοντας υπόψη τα χαρακτηριστικά όλων των παραπάνω διαστάσεων, διαπιστώνεται ότι καθεμιά από τις δυο αρχιτεκτονικές καλύπτει συγκεκριμένες ανάγκες των σύγχρονων επιχειρήσεων και των συναλλαγών αυτών. Συνοπτικά, υπό συνθήκες ισχυρά ανταγωνιστικού επιχειρηματικού περιβάλλοντος, η δυναμική και αυτόνομη ιδέα των υπηρεσιών που διαμορφώνει η Service Oriented τεχνολογία είναι ιδιαίτερα σημαντική. Ταυτόχρονα η επιθυμητή ενοποίηση για την εκτέλεση των επιχειρηματικών συναλλαγών έχει ανάγκη από την αφαιρετική αναπαράσταση του συστήματος, των διαδικασιών του και των δεδομένων που τις υποστηρίζουν. Τα ανοιχτά πρότυπα που χρησιμοποιούνται από τη SOA μπορούν να εξυπηρετήσουν τις ανάγκες για μεταφορά μηνυμάτων επικοινωνίας, διεκπεραίωση συναλλαγών και σύνθεση διεργασιών και ενοποίηση, εφόσον υποστηρίζονται από τα κατάλληλα μετα-μοντέλα και μετα-δεδομένα που προσφέρουν περισσότερη σημασιολογία στις συναλλαγές.

# 6 ■ Επεκτείνοντας την Αρχιτεκτονική που Προσανατολίζεται σε Υπηρεσίες

## 6.1 Στόχοι Εκτεταμένης Αρχιτεκτονικής-Κύριοι άξονες

Στο προηγούμενο κεφάλαιο προσδιορίστηκαν τα βασικότερα χαρακτηριστικά των αρχιτεκτονικών SOA και MDA ως προς τις πέντε διαστάσεις που καθορίζουν μια αρχιτεκτονική ηλεκτρονικών συναλλαγών. Κατόπιν ακολούθησε η σύγκριση των χαρακτηριστικών αυτών και κρίθηκε η σημασία και η αποτελεσματικότητά τους για την διεκπεραίωση των ηλεκτρονικών συναλλαγών μεταξύ των επιχειρηματικών εταίρων. Από την προαναφερθείσα διαδικασία προέκυψε το συμπέρασμα ότι ο προσανατολισμός προς τις υπηρεσίες που υποστηρίζει η Αρχιτεκτονική που Προσανατολίζεται σε Υπηρεσίες είναι ο πλέον κατάλληλος για το σύγχρονο επιχειρηματικό περιβάλλον. Οι στόχοι που θέτει αυτό εκπληρώνονται σε μεγάλο βαθμό από το δυναμικό και ευέλικτο χαρακτήρα της SOA ενώ τα μειονεκτήματα που παρουσιάζει η τελευταία μπορούν να αντιμετωπιστούν με την προσθήκη χαρακτηριστικών από την Οδηγούμενη από τα Μοντέλα Αρχιτεκτονική ή την επέκταση των καθιερωμένων λειτουργιών της SOA κυρίως με την προσθήκη μη λειτουργικών χαρακτηριστικών. Ειδικότερα, η συμβολή της MDA είναι πολύ καθοριστική στην αφαιρετική αναπαράσταση των δεδομένων και των διαδικασιών τα οποία αποκτούν πλέον περιεχόμενο και διευκολύνουν τον χειρισμό των συναλλαγών και την ενοποίηση των ετερογενών συστημάτων. Προς την κατεύθυνση ενσωμάτωσης περιεχομένου στις διαδικασίες συναλλαγών κινούνται επίσης εφαρμογές των οντολογιών και της Τεχνητής Νοημοσύνης, σύμφωνα με δημοσιευμένες ερευνητικές προτάσεις.

Στην παρούσα φάση, βάσει των παραπάνω συμπερασμάτων και ορισμένων παραμέτρων λειτουργικότητας που θα αναφερθούν στη συνέχεια, δίνονται τα

χαρακτηριστικά μιας Εκτεταμένης Αρχιτεκτονικής που προσανατολίζεται σε Υπηρεσίες για σύστημα ηλεκτρονικών συναλλαγών. Η προδιαγραφή αυτή κινείται στα πλαίσια της πρότασης για μια extended Service Oriented Architecture (xSOA) που εξετάζεται από ένα σύνολο ερευνητών, αλλά επιπλέον προσπαθεί να ορίσει πρότυπα και προσεγγίσεις υλοποίησης για τα χαρακτηριστικά της επέκτασης [38].

Δίνοντας λίγα εισαγωγικά στοιχεία για την xSOA όπως αυτά αναφέρονται στο [38], η Εκτεταμένη Αρχιτεκτονική που προσανατολίζεται σε Υπηρεσίες θέτει ως βασική της επιδίωξη να συγκεντρώσει και να δομήσει λογικά τις λειτουργικές απαιτήσεις πολύπλοκων εφαρμογών που χρησιμοποιούν το υπολογιστικό πρότυπο της ιδέας προσανατολισμού προς τις υπηρεσίες. Ορισμένες από αυτές τις απαιτήσεις αφορούν στην ενορχήστρωση, την “ευφυή” δρομολόγηση, τη διαχείριση των υπηρεσιών κοκ. Σύμφωνα με την προτεινόμενη επέκταση ορίζονται βασικά διακριτά επίπεδα αφαιρετικότητας καθένα από τα οποία ορίζει περιορισμούς, ρόλους και ευθύνες για ένα συγκεκριμένο τομέα λειτουργικότητας και παράλληλα υποστηρίζεται από τα αντίστοιχα χαρακτηριστικά του προηγούμενου επιπέδου. Τα επίπεδα που αναγνωρίζονται είναι τα εξής:

- Το κατώτερο επίπεδο βασικών λειτουργιών της SOA που παρέχονται και από τον Δίαυλο Επιχειρηματικών Υπηρεσιών (ESB).
- Το αμέσως ανώτερο αφαιρετικά επίπεδο όπου εντάσσονται οι λειτουργίες σύνθεσης.
- Το ανώτατο επίπεδο στο οποίο συναντώνται οι διαδικασίες διαχείρισης.

Τα επίπεδα αυτά θα παρουσιαστούν αναλυτικότερα στην επόμενη ενότητα.

Κύριος σκοπός του προτεινόμενου εκτεταμένου μοντέλου αρχιτεκτονικής είναι να προσφέρει ένα περιβάλλον συναλλαγών κατάλληλο για την εξυπηρέτηση των συνεχώς αυξανόμενων απαιτήσεων ενοποίησης των επιχειρηματικών εφαρμογών. Επιπροσθέτως, πολύ σημαντικό είναι να εξασφαλιστεί η αυτονομία των υπηρεσιών και των δεδομένων που χρησιμοποιούνται κατά τις συναλλαγές ώστε να αποτυπώνουν τις ιδιαιτερότητες κάθε επιχειρηματικού σχεδιασμού, να αλλάζουν ανάλογα με τις ισχύουσες συνθήκες και να συντίθενται απλά και γρήγορα βάσει κανόνων και απαιτήσεων των πελατών. Δεν πρέπει να παραληφθεί δε, η ανάγκη διαχείρισης και παρακολούθησης των συναλλαγών ώστε να εφαρμοστούν διορθωτικές κινήσεις και ενέργειες μείωσης της πολυπλοκότητας των εφαρμογών.

Στο κύριο μέρος ανάλυσης των χαρακτηριστικών της Εκτεταμένης Αρχιτεκτονικής αναφέρονται οι λειτουργίες που εντάσσονται σε κάθε επίπεδο, ορισμένες από τις ερευνητικές προσεγγίσεις και τέλος δίνονται προτάσεις υλοποίησης των προδιαγραφών αυτών. Βασικά, χρησιμοποιούνται ανοιχτά πρότυπα που έχουν καθιερωθεί από μεγάλους οργανισμούς προτυποποίησης έτσι ώστε οι επιχειρήσεις να έχουν εύκολη και ελεύθερη πρόσβαση σε αυτά και να βασίζονται σε τεχνολογίες οι οποίες είναι ευρέως αποδεκτές και εφαρμόσιμες.

Τέλος, να σημειωθεί ότι η προδιαγραφή αυτή δεν είναι πλήρης καθώς άπτεται ενός ιδιαίτερα εκτεταμένου ερευνητικού πεδίου.

## **6.2 Εκτεταμένη Αρχιτεκτονική Προσανατολισμένη σε Υπηρεσίες**

### **6.2.1 Επίπεδο βασικών λειτουργιών xSOA**

Στο κατώτερο επίπεδο αφαιρετικότητας συναντώνται οι λειτουργίες που είναι ήδη γνωστές από τη συμβατική μορφή της SOA αρχιτεκτονικής. Σύμφωνα με το κλασικό σενάριο, ο πάροχος μιας υπηρεσίας καθορίζει την περιγραφή αυτής της μονάδας λειτουργικότητας και τη δημοσιεύει σε μια αρχή εύρεσης υπηρεσιών. Ο πελάτης που επιθυμεί να καταναλώσει την υπηρεσία αυτή, αφού την ανακαλύψει, καλεί κατευθείαν την περιγραφή από την υπηρεσία μέσω ανταλλαγής μετα-δεδομένων (Metadata Exchange) είτε μέσω ενός καταλόγου ή από μιας αποθήκης όπως το UDDI. Στη συνέχεια ο πελάτης χρησιμοποιεί την περιγραφή της υπηρεσίας για να συνδεθεί με τον πάροχο της υπηρεσίας και την καλεί ή αλληλεπιδρά με την υλοποίηση αυτής [38]. Ως επέκταση του παραπάνω σεναρίου έχουν αναφερθεί διάφορες ερευνητικές προτάσεις. Ο σκοπός των προτεινόμενων επεκτάσεων είναι κυρίως να ληφθούν υπόψη πρόσθετες μη-λειτουργικές ιδιότητες και ιδιότητες συμπεριφοράς των υπηρεσιών. Επίσης επιδιώκεται η καθιέρωση προτύπων για μηχανισμούς ανεύρεσης, δημοσίευσης και γνωστοποίησης μεταξύ κατανεμημένων, δυναμικών και ετερογενών περιβαλλόντων.

Η πλειοψηφία των λύσεων που προτείνονται έχει να κάνει με την εισαγωγή περιεχομένου και σημασιολογίας στις περιγραφές των υπηρεσιών και τις λειτουργίες ανεύρεσης και σύνθεσης. Πράγματι, ορισμένες μη-λειτουργικές απαιτήσεις αλλά και

χαρακτηριστικά που παίζουν σημαντικό ρόλο στην επιλογή των υπηρεσιών και τη σύνθεση αυτών, μπορούν να ενσωματωθούν στις βασικές λειτουργίες με την προσθήκη οντολογιών και μετα-μοντέλων μη-τεχνικών χαρακτηριστικών. Αυτά εφόσον δημοσιεύονται σε αποθήκες μετα-μοντέλων ή σε on-line πόρους εύρεσης οντολογιών μπορούν να διατίθενται και να ανακαλούνται από οποιονδήποτε ενδιαφερόμενο.

Ορισμένα από τα παραδείγματα που παρουσιάζονται στο [38] αναφέρουν την εφαρμογή των αρχών του Σημασιολογικού Ιστού και της Τεχνητής Νοημοσύνης για την περιγραφή των ιδιοτήτων και των δυνατοτήτων των Web Services [45]. Επιπλέον, υποστηρίζεται η σύνθεση υπηρεσιών και η ανακάλυψη αυτών μέσω διερεύνησης βασισμένης σε λέξεις κλειδιά, σε συμπεριφορές ή οντολογίες [49]. Ακόμα μια υλοποίηση αναφέρεται στο [59] σχετικά με την εφαρμογή αλγορίθμων αντιστοίχισης σημασιολογιών και συντακτικού για τις περιγραφές σε WSDL που βοηθούν στην ανακάλυψη των κατάλληλων Web Services.

### **6.2.2 Επίπεδο σύνθεσης xSOA**

Στο επίπεδο σύνθεσης της εκτεταμένης αρχιτεκτονικής xSOA, οι διάφορες αυτόνομες υπηρεσίες συντίθενται βάσει κανόνων και πολιτικών σε πιο σύνθετες εφαρμογές. Στη συνέχεια αυτές χρησιμοποιούνται είτε ως εφαρμογές των πελατών είτε ως βασικές υπηρεσίες που θα χρησιμοποιηθούν για τη σύνθεση άλλων πολυπλοκότερων υπηρεσιών. Σύμφωνα με το [38] οι συνθέσεις αυτές χαρακτηρίζονται από κάποιες βασικές λειτουργίες- ιδιότητες. Αυτές παρατίθενται στη συνέχεια ενώ δίνονται παράλληλα τα στοιχεία που χρειάζεται να ενσωματωθούν στην επέκταση.

Καταρχάς, πρωτεύουσας σημασίας είναι τα *μετα-δεδομένα*, η *πρότυπη ορολογία* και τα *μοντέλα αναφοράς* (*Meta-data, standard terminology and reference models*) που χρησιμοποιούνται για τη δομή και το περιεχόμενο των μηνυμάτων, τις λειτουργίες των μηνυμάτων, τα άκρα των υπηρεσιών και τα διακριτά πρωτόκολλα του δικτύου. Τα μετα-δεδομένα προσφέρουν επίσης στην αφαιρετική αναπαράσταση των ιδιοτήτων, των δυνατοτήτων και των γενικότερων χαρακτηριστικών. Υπό αυτά τα πλαίσια και σε συνδυασμό με κοινώς αποδεκτές ορολογίες/πρότυπα παρέχεται στις υπηρεσίες και τις επιχειρήσεις που τις προσφέρουν η δυνατότητα να γνωρίζουν το

νόημα, το σκοπό, την ακολουθία των ανταλλασσόμενων πληροφοριών και των επιχειρηματικών συνεργασιών.

Ένα δεύτερο εξίσου πολύ σημαντικό ερευνητικό πεδίο αποτελεί η *συμμόρφωση (conformance) των υπηρεσιών*. Ο όρος αυτός χρησιμοποιείται σε επίπεδο σύνταξης των δεδομένων, συμπεριφοράς των σύνθετων λειτουργιών όπου χρειάζεται να τηρείται κάποια τάξη στις σχέσεις συνεργασίας και σημασιολογίας όπου εξασφαλίζεται ότι οι υπηρεσίες και οι διαδικασίες ορίζονται κάτω από σημασιολογικές περιγραφές του τομέα διατηρώντας το νόημά τους και μπορούν να αξιολογηθούν.

Από το επίπεδο της σύνθεσης των υπηρεσιών δε θα μπορούσαν βεβαίως να λείπουν οι λειτουργίες *συντονισμού (coordination)* της εκτέλεσης των μονάδων υπηρεσιών, δηλαδή των διαδικασιών, ελέγχου της ροής των δεδομένων και της ροής ελέγχου. Οι λειτουργίες αυτές υποστηρίζονται ακόμα περισσότερο από τις εργασίες *παρακολούθησης (monitoring)*. Ανάμεσα σε αυτές συγκαταλέγονται οι λειτουργίες παρακολούθησης και ελέγχου των πληροφοριών που παράγονται από τις υπηρεσίες, των στιγμιότυπων των επιχειρηματικών διαδικασιών, των στατιστικών των τελευταίων, της κατάστασης ή περιλήψεων των στιγμιότυπων, ακόμα και λειτουργίες τερματισμού, επανεκκίνησης, διακοπής επιλεγμένων στιγμιότυπων. Τέτοιες δυνατότητες παρέχονται από συγκεκριμένα εργαλεία όπως το WebLogic της BEA και το BusinessWare της Vitria.

Τέλος, αξίζει να αναφερθεί ένα πρόσθετο στοιχείο λειτουργικότητας που αφορά στην *ενίσχυση των πολιτικών (Policy enforcement)* που ακολουθούνται από τις υπηρεσίες. Ένα παράδειγμα θα μπορούσε να είναι οι πολιτικές που καθορίζουν τον χειρισμό ενός συστήματος ή οργανώνουν την αλληλεπίδραση των Web Services [2].

Εξετάζοντας συνολικά το επίπεδο της σύνθεσης των υπηρεσιών, είναι ήδη γνωστό από το κεφάλαιο ανάλυσης των αρχιτεκτονικών διαστάσεων ότι μερικά πρότυπα που βοηθούν προς αυτή την κατεύθυνση είναι τα: WS-BPEL, WS-CDL, BPEL. Αυτά συμβάλλουν καθοριστικά στη σύνθεση μεγάλων συνεργασιών από υπηρεσίες. Ωστόσο είναι απαραίτητο να χρησιμοποιηθούν εργαλεία που εξετάζουν πρόσθετες παραμέτρους όπως κατάσταση και συμπεριφορά των χαλαρά συζευγμένων εφαρμογών [38]. Επιπροσθέτως ιδιαίτερη βαρύτητα δίνεται στη δυναμική φύση των συνθέσεων και επιδιώκεται η εφαρμογή λιγότερο δομημένων μοντέλων διαδικασιών και αυτοματοποιημένων μεθόδων προγραμματισμού. Πρότυπα όπως τα WS-



Transaction, WS-Coordination και BTP που έχουν ήδη παρουσιαστεί δίνουν μια μερική λύση για τη σύνθεση και τον συντονισμό των συναλλαγών.

Γενικότερα, για την βελτίωση της σύνθεσης των υπηρεσιών συμβάλλουν ιδιαίτερα τα μοντέλα ρόλων και πολιτικών που ενισχύουν το σημασιολογικό περιεχόμενο των συντιθέμενων διαδικασιών και δίνουν τις βάσεις συμμόρφωσης των συνεργατών. Βέβαια δεν αμελούνται και οι διαδικασίες δημιουργίας των ροών των διαδικασιών μέσω BPEL. Ένα παράδειγμα παρόμοιας προσέγγισης δίνεται από τους Charfi και Mezini [9] οι οποίοι ορίζουν δυο διαστάσεις, μία σχετικά με τη ροή των διαδικασιών μέσω BPEL ή BPL και μια δεύτερη που αφορά στη χρήση κανόνων οι οποίοι προσαρτώνται στις επιχειρηματικές διαδικασίες.

Ένα τελευταίο ερευνητικό παράδειγμα που πρέπει να σημειωθεί υποδεικνύει την χρήση της Τεχνητής Νοημοσύνης για τη σύνθεση Web Services με τη συμβολή και της γλώσσας οντολογιών OWL. Η πρόταση αυτή υποστηρίζει τη γενικότερη τάση εισαγωγής περιεχομένου στη διαδικασία σύνθεσης των υπηρεσιών. Στα ίδια πλαίσια, η Model Driven Architecture ανάδειξε την αναγκαιότητα ύπαρξης μοντέλων που αναπαριστούν αφαιρετικά τις διάφορες διαδικασίες και τα δεδομένα. Έτσι ουσιαστικά δίνει το έναυσμα για την ενσωμάτωση περιεχομένου στις διαδικασίες σύνθεσης. Το τελευταίο μπορεί να επιτευχθεί είτε με τις γλώσσες οντολογιών είτε με τα μετα-μοντέλα συνθέσεων.

### **6.2.3 Επίπεδο διαχείρισης xSOA**

Η Αρχιτεκτονική που Προσανατολίζεται σε Υπηρεσίες (SOA) είναι άρρητα συνδεδεμένη με την ύπαρξη χαλαρά συζευγμένων εφαρμογών που προέρχονται από διάφορα ετερογενή και κατανεμημένα περιβάλλοντα και συντίθενται δυναμικά ανάλογα με τις απαιτήσεις των πελατών. Κατά συνέπεια, η εφαρμογή μιας τέτοιας αρχιτεκτονικής θέτει ιδιαίτερες προκλήσεις και σύνθετες απαιτήσεις. Η διαχείριση των εφαρμογών που εκτελούνται σε ένα περιβάλλον SOA είναι από τις βασικότερες φροντίδες των επιχειρήσεων. Σύμφωνα με το [38], με τον όρο *διαχείριση των Web Services* ή και των υπηρεσιών γενικότερα, αποδίδουμε τη λειτουργικότητα που απαιτείται για την ανακάλυψη της ύπαρξης, διαθεσιμότητας, απόδοσης, υγείας, των μονάδων χρήσης, της επεκτασιμότητας, όπως επίσης και τον έλεγχο και την διάρθρωση των παραμέτρων, την υποστήριξη καθ' όλη τη διάρκεια ζωής και τη

συντήρηση των Web Services ή των επιχειρηματικών διαδικασιών στα πλαίσια της SOA.

Στην περίπτωση του xSOA, σύμφωνα με την προηγούμενη πηγή, το επίπεδο διαχείρισης είναι επιφορτισμένο με ενέργειες ελέγχου και διαχείρισης της επικοινωνίας και των υποδομών μεταξύ υπηρεσιών, ανάμεσα σε διαφορετικούς παρόχους, πλατφόρμες, τοπολογίες και τεχνολογίες. Οι υπηρεσίες που καλύπτονται από τη διαχείριση αφορούν σε υπηρεσίες συμφωνίας του επιπέδου των υπηρεσιών (Service-level agreement -SLA) που πιθανότατα να περιλαμβάνουν και QoS, όπως επίσης και αναφορές υπηρεσιών. Επιπλέον οι υπηρεσίες διαχείρισης πιθανότατα να παρέχουν στατιστικά στοιχεία απόδοσης και χρησιμοποίησης των υπηρεσιών, μετρήσεις για τους χρόνους απόκρισης των συναλλαγών, μετρήσεις για το φορτίο και για την πορεία των συναλλαγών αλλά και ενέργειες επιδιορθώσεων. Ακόμα στα ίδια πλαίσια εντάσσονται και οι υπηρεσίες παροχής υπηρεσιών και πόρων στο προσωπικό, δυναμικής διάθεσης/απόσυρσης μονάδων υλικού, εγκατάστασης/απεγκατάστασης λογισμικού ανάλογα με το φόρτο εργασίας, ακόμα η εφαρμογή πολιτικών δρομολόγησης και ασφαλείας και ασφαλούς αποστολής μηνυμάτων μέσω SOAP. Στο ίδιο επίπεδο περιλαμβάνονται υπηρεσίες διαχείρισης της κατάστασης και του κύκλου ζωής των υπηρεσιών όπως για παράδειγμα λειτουργίες έναρξης/τερματισμού των υπηρεσιών, αλλαγής των παραμέτρων κλπ. Επίσης μια τελευταία συμβολή του επιπέδου διαχείρισης σχετίζεται με τη δυνατότητα επέκτασης του περιβάλλοντος των υπηρεσιών. Γενικότερα, οι λειτουργίες που προσφέρονται από το επίπεδο διαχείρισης έχουν ως απώτερο σκοπό τη παροχή μιας ολοκληρωμένης εικόνας των ξεχωριστών επιχειρηματικών διαδικασιών και συναλλαγών, την εξασφάλιση συνέπειας των συνθέσεων υπηρεσιών και την αποστολή ειδοποιήσεων σε περίπτωση που υπάρξει κάποια αλλαγή κατάστασης ή συμβεί κάποια συνθήκη. Η συνολική επίβλεψη της λειτουργικότητας των σύνθετων διαδικασιών μειώνει παράλληλα τον κίνδυνο εμφάνισης λαθών.

Βάσει των στοιχείων που δίνονται στο [38] οι παραδοσιακές εφαρμογές διαχείρισης δεν ανταποκρίνονται στις απαιτήσεις των επιχειρήσεων καθώς μένουν μόνο στο επίπεδο των εφαρμογών και δεν προσθέτουν την επιχειρηματική γνώση που απαιτείται και που μπορεί να επιτευχθεί σε επίπεδο υπηρεσιών-και όχι εφαρμογών. Χαρακτηριστικά οι [8], με τη διαχείριση που προσανατολίζεται στις επιχειρήσεις, ρίχνουν το βάρος στην εκτίμηση του αντίκτυπου της εκτέλεσης μιας υπηρεσίας από

την επιχειρηματική σκοπιά. Έτσι οι επιχειρηματικοί στόχοι είναι αυτοί στους οποίους προσαρμόζονται οι εκτελέσεις των υπηρεσιών.

Οι υπηρεσίες που αναφέρθηκαν παραπάνω θέτουν ιδέες επέκτασης της αρχιτεκτονικής SOA οι οποίες ενισχύουν ακόμα περισσότερο τον δυναμικό χαρακτήρα, εξασφαλίζουν την ομαλή ενσωμάτωση των αλλαγών και τονίζουν ιδιαίτερα τη σημασία της επιχειρηματικής λογικής. Όπως προκύπτει, οι υπηρεσίες αυτές είναι απαραίτητο να αναπτυχθούν και να ελεγχθούν από κεντρικές έμπιστες αρχές ενώ οι διάφορες υπηρεσίες που χρησιμοποιούνται από ετερογενή συστήματα μπορούν να μοντελοποιηθούν. Κατά αυτό τον τρόπο χρησιμοποιούνται τυποποιημένες υπηρεσίες οι οποίες όμως δύναται να υλοποιηθούν βάσει των εκάστοτε παραμέτρων συνεργασίας. Ιδιαίτερα χρήσιμα επίσης μπορούν να αποδειχθούν τα συμβόλαια μεταξύ των επιχειρήσεων που θέτουν τους όρους συμφωνίας και συνθέτουν το περιβάλλον της συνεργασίας. Τα συμβόλαια αυτά μπορούν να χρησιμοποιηθούν για τον έλεγχο της εξέλιξης των συνεργασιών, κατά πόσο οι συνεργάτες τηρούν τις πολιτικές και τους κανόνες αλλά και πόσο ανταποκρίνονται στην ποιότητα των υπηρεσιών που προσφέρουν. Ωστόσο λόγω της ετερογένειας της σημασιολογίας των όρων που χρησιμοποιούνται και προκειμένου να μην υπάρχει πρόβλημα αντιστοίχισης, θα μπορούσαν να ανταλλαχθούν μετα-δεδομένα τα οποία πληροφορούν για την ερμηνεία των ανταλλασσόμενων δεδομένων.

Ένα ερευνητικό παράδειγμα που εντάσσεται στο επίπεδο διαχείρισης είναι η δημιουργία ενός προτύπου διαπραγμάτευσης της εμπιστοσύνης που επιδεικνύει κανείς στα Web Services [54]. Αυτό βασίζεται σε μια Model Driven προσέγγιση θεσπίζοντας μοντέλα εμπιστοσύνης και υιοθετώντας μηχανές καταστάσεων που ενσωματώνουν πολιτικές ασφαλείας. Σύμφωνα με το Ludwig et al. (IBM)[33], για την επίτευξη της καλύτερης διαχείρισης των συνεργασιών μια λύση θα μπορούσε να ήταν η κατασκευή μοντέλων συμβολαίων για συμφωνίες - πρότυπο WS-Agreement- τα οποία θα διαχειρίζεται μια κεντρική αρχή και είναι αναγκαία σε όσους ζητούν κάποια υπηρεσία. Ανά πάσα στιγμή αυτά παρέχουν πληροφορίες για την κατάσταση της συμφωνίας.

## 6.2.4 Επισκόπηση της Εκτεταμένης Αρχιτεκτονικής

Βασιζόμενοι στην ανάλυση των επιπέδων που προδιαγράφονται στην Εκτεταμένη Αρχιτεκτονική που προσανατολίζεται σε Υπηρεσίες, παρακάτω δίνεται μια συνοπτική περιγραφή του συστήματος ηλεκτρονικών συναλλαγών. Αυτή στηρίζεται στις προτάσεις επέκτασης που αναφέρθηκαν προηγουμένως στα τρία επίπεδα της Εκτεταμένης Αρχιτεκτονικής που προσανατολίζεται σε υπηρεσίες (xSOA).

Κάθε επιχείρηση ή οργανισμός, με σκοπό να επιτύχει την επιθυμητή ενοποίηση των συστημάτων του με αυτά των συνεργατών του, έχει ανάγκη από μια αφαιρετική αναπαράσταση του συστήματός του και των διαδικασιών που υλοποιεί. Αξιοποιώντας τις δυνατότητες αφαιρετικής αναπαράστασης που παρέχει η MDA μέσω της UML, κάθε επιχείρηση μπορεί να ορίσει τις διαδικασίες, να δώσει σημασιολογικό περιεχόμενο σε αυτές και να ορίσει μετα-μοντέλα δεδομένων, κανόνων, πολιτικών και υπηρεσιών σε MOF. Έτσι θα έχει τη δυνατότητα ανταλλαγής των μετα-δεδομένων μέσω XML χρησιμοποιώντας το πρότυπο XMI. Επίσης υπό αυτή την ουδέτερη προτύπων αναπαράσταση διευκολύνεται η ανταλλαγή δεδομένων με διαφορετικό περιεχόμενο και επιτυγχάνεται η ενοποίηση των δεδομένων. Ακόμα, για την οργάνωση των διαδικασιών της, κάθε επιχείρηση μπορεί να χρησιμοποιήσει μοντέλα υπηρεσιών οι οποίες είναι ήδη δημοσιευμένες ή να ορίσει τις δικές της υπηρεσίες και να τις δημοσιεύσει. Βασικός άξονας της επιχειρηματικής αρχιτεκτονικής όπως ήδη έχει αναφερθεί είναι οι υπηρεσίες αλλά εδώ για την διευκόλυνση της ενοποίησης οι επιχειρήσεις καλούνται να δημιουργήσουν τα δικά τους μοντέλα υπηρεσιών. Συνολικά, αυτά ευνοούν την επαναχρησιμοποίηση και την υλοποίηση σε οποιαδήποτε πλατφόρμα όπως επίσης ενσωματώνουν πολύ εύκολα τις αλλαγές στα επιχειρηματικά μοντέλα.

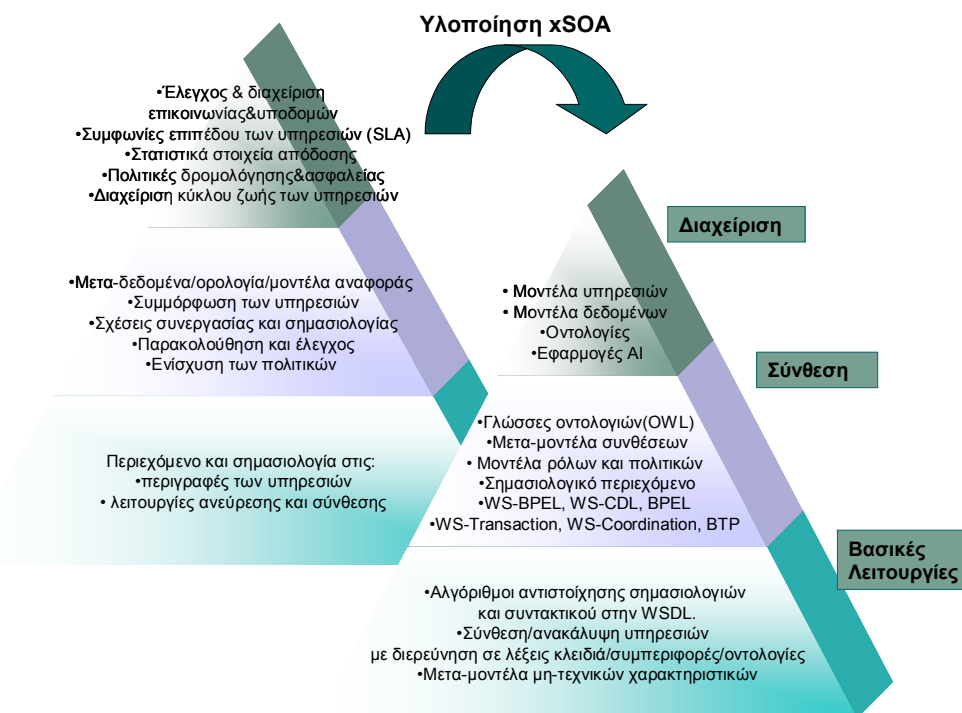
Η δημοσίευση των μοντέλων υπηρεσιών ή των μοντέλων δεδομένων, κανόνων και πολιτικών μπορεί να γίνει σε μια κεντρική μονάδα αποθήκευσης (repository). Αυτή θα πρέπει να υποστηρίζεται από υπηρεσίες που προσφέρει ένας εξυπηρετητής (server) ο οποίος θα λειτουργεί ως μεσίτης υπηρεσιών (service broker) όπως στην περίπτωση των Web Services-από την πλευρά της SOA. Πέρα ωστόσο από τις υπηρεσίες αυτές, ο server μπορεί να προσφέρει πρόσθετες υπηρεσίες όπως ενορχήστρωση (orchestration) των συναλλαγών μεταξύ των συμμετεχόντων, υπηρεσίες παρακολούθησης της ροής των διαδικασιών (monitoring), υπηρεσίες QoS

και ελέγχου της πορείας των συναλλαγών (πρότυπα *WS-Transactions* και *WS-CAF*). Οι υπηρεσίες αυτές αντιστοιχούν στα επίπεδα σύνθεσης και διαχείρισης του xSOA που αναφέρθηκε στην προηγούμενη υποενότητα.

Σε τελικό στάδιο, εφόσον οι συμμετέχοντες στη συναλλαγή πληροφορηθούν σχετικά με τα μοντέλα των υπηρεσιών που θέλουν να καλέσουν, τα μοντέλα δεδομένων των συνεργατών τους και τον τρόπο που συντίθενται οι διάφορες διαδικασίες, είναι σε θέση να ενοποιήσουν τις διαδικασίες και τα δεδομένα τους και να ολοκληρώσουν την συναλλαγή προχωρώντας σε επίπεδο υλοποίησης επικοινωνώντας άμεσα μεταξύ τους. Έτσι έχουν συνυπογράψει ένα επιχειρηματικό συμβόλαιο και πλέον μπορούν να καλέσουν τις αντίστοιχες υπηρεσίες ανάλογα με τη ροή εργασιών που έχουν ορίσει, να κάνουν τις αντιστοιχήσεις μεταξύ των τύπων δεδομένων ώστε να αναπτύξουν έναν κοινό κώδικα επικοινωνίας. Επίσης καθ' όλη τη διάρκεια της συναλλαγής ο κεντρικός εξυπηρετητής θα μπορεί να εποπτεύει την εξέλιξη των συναλλαγών αλλά και να καλούνται υπηρεσίες διαχείρισης των διαδικασιών.

Εξετάζοντας το παραπάνω αρχιτεκτονικό μοντέλο, διακρίνονται τέσσερις βασικές οντότητες που συμμετέχουν στο σύστημα: οι δυο επιχειρηματικοί συνεργάτες, ο εξυπηρετητής και η αποθήκη περιγραφών και μοντέλων υπηρεσιών. Πρόκειται για ένα καταναμημένο σύστημα που χρησιμοποιεί το πρωτόκολλο SOAP πάνω από το HTTP για τη μεταφορά των μηνυμάτων -διάσταση ανταλλαγής μηνυμάτων. Επιπλέον, το εν λόγω σύστημα βασίζεται στην τεχνολογία πελάτη-εξυπηρετητή αλλά ενσωματώνει στην αρχιτεκτονική του και στοιχεία από την επικοινωνία μεταξύ ομότιμων (P2P) οντοτήτων - όπως και τα Web Services. Η διαφοροποίηση από τα Web Services είναι ότι εδώ ενσωματώνεται η σημασιολογία των διαδικασιών που μπορεί να καθορίζει τις υπηρεσίες που θα κληθούν ανάλογα με τις απαιτήσεις των εταίρων και γίνεται εύκολη η ενοποίηση σε επίπεδο δεδομένων μέσω των μετα-μοντέλων. Η σύνθεση των συναλλαγών σύμφωνα με αυτό το μοντέλο αρχιτεκτονικής δεν έχει το στατικό χαρακτήρα των Web Services αλλά μπορεί ανά πάσα στιγμή να δομηθεί ανάλογα με τις ανάγκες των συνεργατών ή κάποιο στιγμιαίο συμβάν. Επιπλέον πολύ βασική είναι και η συνεισφορά των πρόσθετων υπηρεσιών που εξαρτώνται από τα προφίλ των χρηστών και μπορούν να ενταχθούν δυναμικά στη ροή των διαδικασιών. Τέλος δεν τίθεται ζήτημα αντιστοίχισης δεδομένων που ορίζονται σε διαφορετικά σχήματα εφόσον ανταλλάσσονται πλέον τα μοντέλα των δεδομένων και μπορούν να γεφυρωθούν οι διαφορές.

Συνοψίζοντας, το μοντέλο αρχιτεκτονικής που παρουσιάστηκε έχει ως σκοπό να συνδυάσει τα πλεονεκτήματα της Service Oriented αρχιτεκτονικής μαζί με τις ιδέες μοντελοποίησης της Model Driven αλλά κυρίως να ανταποκριθεί στον υπηρεσιοστρεφή χαρακτήρα των σύγχρονων εμπορικών συναλλαγών. Παράλληλα παρέχονται προτάσεις για την ενσωμάτωση μη-λειτουργικών χαρακτηριστικών (QoS, πολιτικές, κανόνες, στατιστικά απόκρισης, ανάκαμψη από προβλήματα κλπ.) στις επιχειρηματικές συναλλαγές και ιδιαίτερο ρόλο καλείται να παίξει το περιεχόμενο των συναλλαγών αυτών. Η λειτουργικότητα του Enterprise Service Bus θα μπορούσε να επεκταθεί με υπηρεσίες που αντιστοιχούν στα επίπεδα σύνθεσης και διαχείρισης χρησιμοποιώντας ωστόσο μοντέλα και γλώσσες οντολογιών για την απόδοση περιεχομένου στις διαδικασίες, τις υπηρεσίες και τα δεδομένα.



**Εικόνα 22-Χαρακτηριστικά & Υλοποίηση xSOA ανά επίπεδο**

## **6.3 Η δυναμική για απόδοση της επένδυσης σε αρχιτεκτονικές SOA, MDA & xSOA**

Σε όλο το περιεχόμενο της παρούσας διπλωματικής εργασίας διαφαίνεται η μεγάλη σημασία που έχει η εφαρμογή των ιδεών των SOA και MDA για τις επιχειρήσεις. Τα κίνητρα που οδήγησαν τις επιχειρήσεις στην υιοθέτηση των service-oriented και model-driven ιδεών ισχυροποιούνται ολοένα και περισσότερο ενώ τα οφέλη που αποκομίζονται από την εφαρμογή αυτών είναι ιδιαίτερα σημαντικά. Η καθοριστική σημασία της Αρχιτεκτονικής που προσανατολίζεται σε Υπηρεσίες σε συνδυασμό με την Οδηγούμενη από τα Μοντέλα Αρχιτεκτονικής και των οντολογιών φανερώνεται μέσα από τις πρόσθετες υπηρεσίες που προσφέρει η Εκτεταμένη SOA. Η αξία της εφαρμογής των παραπάνω αρχιτεκτονικών παρουσιάζεται εν συντομία στην παρούσα παράγραφο.

Στο σύγχρονο επιχειρηματικό περιβάλλον οι οργανισμοί επικεντρώνονται περισσότερο στην προσφορά άμεσων, καινοτόμων και πληρέστερων υπηρεσιών προς τους πελάτες τους ξεφεύγοντας από τα στενά πλαίσια της παροχής προϊόντων και την επιδίωξη του κέρδους. Εντάσσοντας τα δυο τελευταία στις γενικότερες επιδιώξεις τους, οι επιχειρήσεις αποσκοπούν κυρίως στην αύξηση της ευελιξίας τους.

Ενάντια στην επιδίωξη της ευελιξίας τίθεται η εισαγωγή συνεχώς νέων προτύπων και τεχνολογιών που αυξάνουν την πολυπλοκότητα. Οι επιχειρήσεις επενδύουν σημαντικούς πόρους για την απόκτηση και την εφαρμογή αλλά και στην συντήρηση ή αντικατάστασή τους. Αυτό το πρόβλημα βέβαια αναγκάζει τις μικρομεσαίες επιχειρήσεις στην απώλεια μέρους της αγοράς εφόσον δε κατέχουν τους απαιτούμενους πόρους. Επιπλέον, πολύ σημαντική είναι και η πολυπλοκότητα που προσθέτουν οι διάφορες επιχειρηματικές διαδικασίες, με σημεία που χρειάζονται ιδιαίτερο χειρισμό ή σημεία που είναι περισσότερο εξειδικευμένα σε στοιχεία των πελατών και άρα με διαφορετικό περιεχόμενο και στόχους. Δεν πρέπει επίσης να παραληφθεί ότι οι περισσότερες επιχειρήσεις έχουν ανάγκη από πρόσθετες διαδικασίες και υπηρεσίες. Για αυτό το λόγο χρειάζεται να ενοποιούν τα συστήματά τους με άλλα ετερογενή, στοιχείο το οποίο προσδίδει ακόμα περισσότερη πολυπλοκότητα στις διαδικασίες. Τέλος, τίθεται το ζήτημα της επαναχρησιμοποίησης των μονάδων λειτουργικότητας και της συντήρησης των ήδη υπαρχουσών.

Τη λύση στα προβλήματα που αναφέρθηκαν προηγουμένως έρχεται να δώσει η εφαρμογή των ιδεών SOA, MDA και xSOA. Η αποδέσμευση της επιχειρηματικής λογικής από την τεχνολογική πλατφόρμα μειώνει κατά πολύ τα κόστη συντήρησης, αντικατάστασης και ενοποίησης των επιμέρους τμημάτων κώδικα, της χρησιμοποιούμενης πλατφόρμας, και των ετερογενών συστημάτων. Οι επενδύσεις αφορούν σε βασικές λειτουργικές μονάδες του συστήματος όπως ESB, αποθήκες δεδομένων ή υπηρεσιών κοκ. οι οποίες εξυπηρετούν το σύνολο των επιχειρηματικών συναλλαγών και της ενοποίησης. Επιπλέον οι διάφορες μονάδες λειτουργικότητας όπως είναι οι υπηρεσίες παρέχουν τη δυνατότητα ευρείας επαναχρησιμοποίησης στον χρόνο εκτέλεσης. Στην περίπτωση αυτή μειώνεται ακόμα και ο κίνδυνος αστοχίας καθώς χρησιμοποιούνται ήδη δοκιμασμένες μονάδες.

Ανάμεσα στους μη τεχνικούς περιορισμούς που συναντούν καθημερινά οι επιχειρήσεις συγκαταλέγεται και η ανάγκη εξέλιξης. Οι υποδομές της Τεχνολογίας Πληροφορικής (IT) αυτών ωστόσο χρειάζεται να ακολουθούν την ίδια πορεία. Ένας από τους λόγους που πρέπει να συμβαίνει αυτό είναι ο κίνδυνος αποτυχίας των IT συστημάτων, η οποία αν συμβεί έχει σοβαρές λειτουργικές και οικονομικές επιπτώσεις. Επιπλέον πολύ βασικός λόγος είναι η δυνατότητα αλλαγών των τεχνολογικών υποδομών οι οποίες επιτάσσουν μια σειρά αλλαγών και επανεκπαίδευσης του προσωπικού. Ο περιορισμός της εξέλιξης αντιμετωπίζεται απλά μέσα από τη διάσπαση των μεγάλων έργων σε μικρότερα που προσφέρει η SOA και εξυπηρετεί επίσης την αυτόνομη αντιμετώπιση των αλλαγών.

Επιπροσθέτως οι παραπάνω αρχιτεκτονικές προσφέρουν πολύ σημαντικά οφέλη όσον αφορά στη μείωση διαφόρων κινδύνων. Η εφαρμογή των συμβολαίων υπηρεσιών και η αφαιρετική αναπαράσταση των επιχειρηματικών υπηρεσιών και διαδικασιών μειώνουν τους κινδύνους να μην εκπληρωθούν οι προσδοκίες κάποιου τμήματος της επιχείρησης και να εκτιμηθούν ή να προγραμματιστούν λανθασμένα οι πόροι και οι απαιτήσεις του συστήματος, αντίστοιχα. Εν τω μεταξύ το επίπεδο της διαχείρισης των διαδικασιών και η χαλαρή σύζευξη (loose coupling) που χαρακτηρίζει τη σύνθεση των διαδικασιών ευνοούν τακτικές διαίρει και βασίλευε (divide and conquer). Έτσι μειώνονται σημαντικά οι πιθανότητες λαθών καθώς τα έργα δομούνται από μικρές λειτουργικές μονάδες και είναι εύκολος ο χειρισμός τους. Επιπλέον έχει ήδη αναφερθεί η πολύ σημαντική συμβολή στην ενοποίηση βασικών υποσυστημάτων που εφαρμόζονται ευρέως. Με τις SOA και MDA μειώνεται το ρίσκο και στον συγκεκριμένο τομέα.



Σε γενικά πλαίσια, οι αρχιτεκτονικές που διαχωρίζουν την επιχειρηματική (business) από τη λογική της υλοποίησης (implementation logic) και υποστηρίζουν την ενοποίηση ετερογενών συστημάτων, την αυτονομία των υπηρεσιών και την αφαιρετική αναπαράσταση των διαδικασιών και των πληροφοριών προσφέρουν καθοριστικά στο εκσυγχρονισμό των επιχειρηματικών συναλλαγών. Επίσης μειώνουν τα κόστη και διευκολύνουν τη διαδικασία βελτίωσης των συστημάτων και των αρχιτεκτονικών καθώς τα άτομα που ανήκουν στο περιβάλλον της επιχείρησης κατανοούν τις IT λειτουργίες και τις διαχειρίζονται με επιτυχία.

# 7 ■ Επίλογος

## 7.1 Σύνοψη και Συμπεράσματα

Οι ηλεκτρονικές επιχειρηματικές συναλλαγές συνθέτουν έναν πολυδιάστατο χώρο στον οποίο οι εξελίξεις της Τεχνολογίας Πληροφορικής και Επικοινωνιών (ICT) ενσωματώνονται στο περιβάλλον των επιχειρήσεων και των οργανισμών. Η παρούσα διπλωματική επεδίωξε να ασχοληθεί με καίρια ζητήματα του χώρου αυτού όπως είναι η διαλειτουργικότητα και η ενοποίηση των Πληροφοριακών Συστημάτων για την αποτελεσματικότητα των εμπορικών συνεργασιών.

Αρχικά παρουσιάστηκαν τα ευρέως εφαρμοζόμενα τεχνολογικά πρότυπα και οι αρχιτεκτονικές που υιοθετούνται από την πλειοψηφία των επιχειρήσεων για την οργάνωση των διαδικασιών, των δεδομένων και των τεχνικών υποδομών τους, σύμφωνα πάντα με τα επιχειρηματικά μοντέλα που ακολουθούν. Ορίστηκαν οι κυριότερες έννοιες που χρησιμοποιούνται στις ηλεκτρονικές συναλλαγές και ειδικότερα η έννοια της διαλειτουργικότητας που αποτελεί το βασικότερο στόχο του εκσυγχρονισμού των δομών των συναλλαγών.

Ιδιαίτερη βαρύτητα δόθηκε στην ανάλυση των επιμέρους χαρακτηριστικών της Προσανατολισμένης σε Υπηρεσίες (Service Oriented) και της Οδηγούμενης από τα Μοντέλα (Model Driven) Αρχιτεκτονικής. Αυτές θεωρούνται ως οι καταλληλότερες και πιο αποτελεσματικές για τη δόμηση των συναλλαγών των επιχειρήσεων. Επιπροσθέτως διαχωρίστηκαν οι κυριότερες διαστάσεις της αρχιτεκτονικής ηλεκτρονικών συναλλαγών και κατόπιν δόθηκαν ορισμένες προτάσεις για επέκταση της Αρχιτεκτονικής που Προσανατολίζεται σε Υπηρεσίες. Οι προτάσεις αυτές βασίστηκαν σε σχετικές γνωστές ερευνητικές εργασίες αλλά επίσης προτάθηκαν ορισμένα στοιχεία υλοποίησης όπως προέκυψαν από την ανάλυση που προηγήθηκε στην παρούσα διπλωματική. Τέλος σχολιάστηκαν τα οφέλη που αποκομίζουν οι επιχειρήσεις από τον συνδυασμό των αρχιτεκτονικών SOA και MDA αλλά και από την εφαρμογή της Εκτεταμένης Αρχιτεκτονικής που Προσανατολίζεται σε Υπηρεσίες.

Συμπερασματικά, αξίζει να αναφερθεί ότι ο τομέας των ηλεκτρονικών συναλλαγών, που υποστηρίζονται από τα Πληροφοριακά Συστήματα, προσφέρει μια πληθώρα τεχνολογιών. Ωστόσο, προκειμένου οι επιχειρήσεις να μπορέσουν να προσαρμοστούν στις διαρκώς μεταβαλλόμενες συνθήκες χρειάζεται να προσαρμόζονται δυναμικά και άμεσα στις αλλαγές. Οι διαδικασίες συντήρησης και αλλαγών στον κώδικα ή οι μετατροπές των επιχειρηματικών μοντέλων και των υποδομών ώστε να ανταποκρίνονται στις εξελισσόμενες συνθήκες, μπορούν να αυξήσουν τα κόστη των επιχειρήσεων και να τις επιβαρύνουν σημαντικά. Έτσι κρίνεται επιτακτικό να δοθεί μια ανεξάρτητη προτύπων και ευέλικτη δομή στα Πληροφοριακά Συστήματα και τις αρχιτεκτονικές που υποστηρίζουν οι συναλλαγές. Επιπλέον η υπηρεσιοστρεφής αρχιτεκτονική των συναλλαγών έχει ανάγκη από επέκταση των δυνατοτήτων που προσφέρει και ενσωμάτωση περιεχομένου στις διαδικασίες και τα δεδομένα. Απώτερος σκοπός είναι να καλυφθούν μη-λειτουργικές απαιτήσεις και να βελτιωθεί η διαχείριση, η σύνθεση και η εξεύρεση των υπηρεσιών. Οι αρχιτεκτονικές Service Oriented και Model Driven παρέχουν ενδιαφέρουσες λύσεις στα παραπάνω θέματα σε συνδυασμό επίσης με στοιχεία από την τεχνολογία των οντολογιών και της τεχνητής νοημοσύνης. Η επέκταση της Service Oriented αρχιτεκτονικής που παρουσιάστηκε στο 6<sup>ο</sup> κεφάλαιο αυτής της διπλωματικής σκιαγραφεί κάποιες πιθανές λύσεις και δίνει το έναυσμα για εκτεταμένη ερευνητική δραστηριότητα.

## **7.2 Πεδία έρευνας στον τομέα των ηλεκτρονικών συναλλαγών**

Όπως έχει ήδη αναφερθεί ο τομέας των ηλεκτρονικών συναλλαγών προσφέρει μια μεγάλη ποικιλία από ερευνητικά θέματα. Όσο μεγαλύτερη είναι η προσπάθεια ενοποίησης ετερογενών συστημάτων και συνεργασίας μεταξύ επιχειρηματικών οργανισμών, τόσο περισσότερες παράμετροι πρέπει να εξεταστούν και να συνδυαστούν για να προκύψει η βέλτιστη λύση. Επιπλέον, η εξέλιξη στην τεχνολογία λογισμικού αλλά και η πρόοδος στην επιστήμη των επικοινωνιών και της πληροφορίας θέτουν νέες προκλήσεις για περαιτέρω έρευνα. Αν και η παρούσα διπλωματική κινείται πάνω στην έρευνα των αρχιτεκτονικών των ηλεκτρονικών

συναλλαγών, παρακάτω δίνονται ορισμένα πρόσθετα θέματα που χρήζουν ιδιαίτερης προσοχής αφού μπορούν να συνδυαστούν με τις αρχιτεκτονικές των συναλλαγών.

Ένα από τα σημαντικότερα ερευνητικά πεδία που μπορεί να παίξει καθοριστικό ρόλο στην διευκόλυνση των επιχειρηματικών συναλλαγών είναι η ανάπτυξη του Σημασιολογικού Ιστού (Semantic Web). Ο Σημασιολογικός Ιστός προσφέρει την δυνατότητα αναπαράστασης του περιεχομένου των πληροφοριών που βρίσκονται διασκορπισμένες και ανταλλάσσονται στο Διαδίκτυο μέσω οντολογιών συγκεκριμένων τομέων. Ήδη γίνεται μια προσπάθεια ενσωμάτωσης των οντολογιών στα Web Services που αποδίδεται με τον όρο *Semantic Web Services*. Στην περίπτωση αυτή με τη βοήθεια των γλωσσών οντολογιών όπως η Web ontology Language (OWL[42]), μπορούν να δημιουργηθούν και να διαμοιραστούν περιγραφές των Web Services που γίνονται αντιληπτές από τη μηχανή. Αυτό επιτυγχάνεται ενσωματώνοντας τις ιδέες που έχουν οριστεί σε οντολογίες μέσα στα web Services. Επιπλέον, γενικευμένες οντολογίες υπηρεσιών όπως η OWL Services (OWL-S)[43] και η Web Services Modeling Ontology (WSMO)[66] σε συνδυασμό με γλώσσες κανόνων θέτουν τα θεμέλια για τη σημασιολογική αναπαράσταση της λειτουργικότητας και της συμπεριφοράς των υπηρεσιών.

Εκτός από την εφαρμογή των οντολογιών στις ηλεκτρονικές συναλλαγές, επίσης ένα πολύ βασικό ζήτημα είναι η ασφάλεια των εκτελούμενων συναλλαγών. Ένας τρόπος ελέγχου είναι η εφαρμογή φίλτρων στα πακέτα TCP/IP όπως γίνεται με τα τείχη προστασίας (firewalls) και τα Intrusion Detection Systems (IDS). Ειδικότερα όμως στην περίπτωση της SOA, η οποία χρησιμοποιεί την XML ως πρότυπο αναπαράστασης των ανταλλασσόμενων πληροφοριών πάνω από το HTTP πρωτόκολλο, ο παραδοσιακός αυτός τρόπος δεν αποφέρει αποτελέσματα. Μια λύση σε αυτό το πρόβλημα αποτελεί η ενσωμάτωση ειδικών στοιχείων πιστοποίησης μέσα στο περιεχόμενο του XML μηνύματος καθώς και άλλα συστήματα ελέγχου του XML περιεχομένου. Γενικότερα, υπάρχει ανάγκη να θεσπιστούν πρότυπα που να εξασφαλίζουν την αυθεντικότητα και ταυτοποίηση των συναλλασσόμενων μερών καθώς κατά τις συναλλαγές αυτές πολλές φορές μεταφέρονται εμπιστευτικά δεδομένα και στοιχεία ανταγωνισμού-πχ. επιχειρηματικά μοντέλα-που δεν πρέπει να υποκλαπούν.

Τέλος, στην ενότητα 5.2 αναφέρθηκε ο όρος *autonomic services* και συγκεκριμένα οι εξής υπηρεσίες:

Αυτό-οριζόμενες υπηρεσίες διαχείρισης

Αυτό-προσαρμοζόμενες υπηρεσίες διαχείρισης

Αυτό-θεραπευόμενες υπηρεσίες διαχείρισης

Αυτό-βελτιούμενες υπηρεσίες διαχείρισης

Αυτό-προστατευόμενες υπηρεσίες διαχείρισης

Πρόκειται για υπηρεσίες που αναπτύσσουν αυτόνομη συμπεριφορά και καλύπτουν ιδιαίτερες ανάγκες χωρίς την ανθρώπινη παρέμβαση. Αυτές εντάσσονται σε ένα νέο ερευνητικό πεδίο που αναπτύσσεται γνωστό ως “*Autonomic computing*” που έχει ως σκοπό την ανάπτυξη υπολογιστικών συστημάτων που μπορούν να αναπτύξουν συμπεριφορά αυτό-διαχείρισης ώστε να μειωθεί η πολυπλοκότητα των συνεχώς εξελισσόμενων συστημάτων. Όσον αφορά στις υπηρεσίες αυτού του είδους, αυτές μπορούν να ανταποκρίνονται στον δυναμικό χαρακτήρα των σύγχρονων συναλλαγών και να απαλλάσσουν από το πρόσθετο βάρος της διαχείρισης. Περισσότερες λεπτομέρειες σχετικά με τη λειτουργία των υπηρεσιών αυτών δόθηκαν στην ενότητα 5.2.

Τα ερευνητικά θέματα που παρουσιάστηκαν εδώ συνθέτουν ένα μικρό δείγμα από την ποικιλία των ζητημάτων που έχουν σχέση με τον σχεδιασμό, την οργάνωση, τη διαχείριση και την υλοποίηση των ηλεκτρονικών συναλλαγών καθώς και με τις αρχιτεκτονικές που τις υποστηρίζουν. Τα συγκεκριμένα παραδείγματα αναφέρονται σε βασικά θέματα που προσελκύουν συνεχώς το ενδιαφέρον της ερευνητικής και επιχειρηματικής κοινότητας.

# 8 ■ Βιβλιογραφία

- [1] Adra Al Mosawi, Liping Zhao and Linda Macaulay, School of Informatics, The University of Manchester, “A Model Driven Architecture for Enterprise Application Integration”, Proceedings of the 39th Hawaii International Conference on System Sciences – 2006.
- [2] Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services: Concepts, Architectures and Applications. Springer, Heidelberg (2004)
- [3] ATHENA project, [www.athena-ip.org](http://www.athena-ip.org) , Μάρτιος 2006.
- [4] Business Process Modeling Language (BPML) : <http://www.bpmi.org/BPML.htm>
- [5] Business Process Modeling Notation (BPMN):  
<http://www.bpmn.org/Documents/BPMN%20V1-0%20May%2003%202004.pdf>
- [6] Business-to-business interactions: issues and enabling technologies  
Brahim Medjahed, Boualem Benatallah, Athman Bouguettaya, Anne H. H. Ngu, Ahmed K. Elmagarmid. The VLDB Journal (2003) 12: 59–85.
- [7] Business Transaction Protocol (BTP) available at: <http://www.oasis-open.org/committees/businesstransactions/documents/primer/Primerhtml/BTP%20Primer%20D1%2020020602.html>
- [8] Casati, F. et al.: Business-oriented management of Web services. Commun. ACM 46(10), 55–60 (2003)
- [9] 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)
- [10] Davis L., R.F. Gamble\*, J. Payton, “ The impact of component architectures on interoperability” The Journal of Systems and Software 61 (2002) 31–45
- [11] Dirk Krafzig, Karl Banke, Dirk Slama, “Enterprise SOA. Service Oriented Architecture Best Practices”. Prentice Hall.
- [12] ebXML Technical Architecture Project Team: ebXML Technical Architecture Specification, v1.0.4. Technical report, eb-XML.org, February, 2001.
- [13] European Commission, Information Society and Media, “Digital Business Ecosystems”.
- [14] Fabio Casati and Ming-Chien Shan, “Process Automation as the Foundation for E-Business”.

- [15] GENESIS project, Deliverable D3.1 Analysis of the Data Modeling State of the Art, available online at [www.genesis-ist.eu](http://www.genesis-ist.eu)
- [16] G. Gionis, Y. Charalabidis, D. Askounis, S. Koussouris<sup>1</sup>, F. Lampathaki, “Realising the Business Perspective of eTransactions among Heterogeneous Partners: The Practical Power of Hybrid Architectural Approaches”.
- [17] H. Kreger, et. al “Management Using Web Services: A Proposed Architecture and Roadmap”, IBM, HP and Computer Associates, June 2005, available at: [www-128.ibm.com/developerworks/library/specification/ws-mroadmap](http://www-128.ibm.com/developerworks/library/specification/ws-mroadmap).
- [18] [http://www.theenterprisearchitect.eu/archive/2007/09/13/modeldriven\\_soa](http://www.theenterprisearchitect.eu/archive/2007/09/13/modeldriven_soa)
- [19] [http://en.wikipedia.org/wiki/Loose\\_coupling](http://en.wikipedia.org/wiki/Loose_coupling)
- [20] [http://en.wikipedia.org/wiki/Software\\_architecture#Architecture\\_frameworks](http://en.wikipedia.org/wiki/Software_architecture#Architecture_frameworks)
- [21] Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer, Glossaries. New York, NY: 1990.
- [22] J. O. Kephart, D. M. Chess “The Vision of Autonomic Computing”, IEEE Computer, January 2003.
- [23] Jan Mendling and Michael Hafner, “From WS-CDL Choreography to BPEL Process Orchestration”
- [24] Jian Yang and Mike P. Papazoglou, “Service Components for Managing the Life-Cycle of Service Compositions”
- [25] Jian Yang, Mike P. Papazoglou, Willem-Jan den Heuvel, “Tackling the Challenges of Service Composition in E-Marketplaces”
- [26] Kelkar, A., Gamble, R., 1999. Understanding the architectural characteristics behind middleware choices. In: Proceedings of the 1st Conference on Information Reuse and Integration.
- [27] Kenneth C. Laudon, Jane P. Laudon, «Πληροφοριακά Συστήματα Διοίκησης».
- [28] Keshav, R., 1999. Architecture integration elements: connectors that form middleware, M.S. thesis, University of Tulsa, Tulsa, OK.
- [29] Konrad Pfadenhauer, Schahram Dustdar, Burkhard Kittl. “Comparison of Two Distinctive Model Driven Web Service Orchestration Proposals”, Proceedings of the 2005 Seventh IEEE International Conference on E-Commerce Technology Workshops (CECW’05) 0-7695-2384-6/05 © 2005 IEEE
- [30] Kutvonen, Lea, “Addressing Interoperability Issues in Business Process Management”

- [31] Leymann F., and D. Roller “Production Workflow - Concepts and Techniques”, PTR Prentice Hall, 2000.
- [32] Leymann F., “Combining Web Services and the Grid: Towards Adaptive Enterprise Applications”, Proc. CAiSE/ASMEA’05 (Porto, Portugal, June 2005).
- [33] Ludwig, H., Dan, A., Kearney, R.: Crona: an architecture and library for creation and monitoring of WS-Agreements. In: ICSOC ’04: Proceedings of the 2nd international conference on Service oriented computing, pp. 65–74. ACM Press, New York (2004)
- [34] M.P. Papazoglou and G. Georgakopoulos, “Service-Oriented Computing, CACM”, October 2003, 46(10).
- [35] M.P. Papazoglou, P.M.A. Ribbers “e-Business: Organizational and Technical Foundations”, J. Wiley & sons, Ltd, February 2006.
- [36] M. P. Papazoglou, W. van den Heuvel ““Business Process Development Lifecycle Methodology” to appear in Communications of ACM, 2006.
- [37] Michael P. Papazoglou, Paolo Traverso, Schahram Dustdar, Frank Leymann, “Service-Oriented Computing Research Roadmap”.
- [38] Mike P. Papazoglou · Willem-Jan van den Heuvel, “Service oriented architectures: approaches, technologies and research issues”. The VLDB Journal (2007) 16:389–415.
- [39] Model-Driven Software Development: “Model-Driven development: The good, the bad and the ugly”, IBM Journal. Volume 45, Number , 2006. Available online at: <https://www.research.ibm.com/journal/sj/453/hailpern.html> .
- [40] OASIS, Organization for the Advancement of Structured Information Standards <http://www.oasis-open.org>
- [41] Object Management Group: <http://www.omg.org/mda> .
- [42] Ontology Web Language (OWL) available online at <http://www.w3.org/2004/OWL/>
- [43] OWL-S: Semantic Markup for Web Services available online at <http://www.w3.org/Submission/OWL-S>
- [44] Parent, C., Spaccapietra, S.: Issues and Approaches of Database Integration. Commun. ACM 41(5), 166–178 (1998).
- [45] Patil, A.A., et al.: Meteor-S: web service annotation framework. In: WWW ’04: Proceedings of the 13th international conference on World Wide Web, pp. 553–562. ACM Press, New York (2004)



- [46] Ramakrishna Raju, “ *An Evaluation Of SOA Frameworks*”.
- [47] Renato Fileto and Claudia Bauzer Medeiros, “A Survey on Information Systems Interoperability”.
- [48] Ruinan Gong, Qing Li, Ke Ning, Yuliu Chen and David O’Sullivan “Business Process Collaboration Using Semantic Interoperability: Review and Framework”, 1st Asian Semantic Web Conference, September 2006 Beijing, China.
- [49] Sahin, O.D., et al.: SPiDeR: P2P-Based Web Service Discovery In: Proceedings of the Third International Conference on Service Oriented Computing, pp. 157–170. Springer, Amsterdam (2005)
- [50] Seacord, R.C., Plakosh, D., Lewis, G.A.: *Modernizing Legacy Systems*. Carnegie Mellon, SEI. Addison-Wesley, Reading (2003).
- [51] Selda Güner “Architectural Approaches, Concepts and Methodologies of Service Oriented Architecture”, Master Thesis.
- [52] Siegel Jon, Ph.D., and the OMG Staff Strategy Group, “Developing in OMG’s Model Driven Architecture (MDA)” (<ftp://ftp.omg.org/pub/docs/omg/01-12-01.pdf>).
- [53] Silberchartz, Korth, Sudarshan. “Συστήματα Βάσεων Δεδομένων. Η πλήρης θεωρία των Βάσεων Δεδομένων”, ελληνική έκδοση από τις “Εκδόσεις Μ.Γκιούρδας”.
- [54] Skogsrud, H., Benatallah, B., Casati, F.: Trust-serv: modeldriven lifecycle management of trust negotiation policies for Web services. In: WWW ’04: Proceedings of the 13th international conference on World Wide Web, pp. 53–62. ACM Press, New York (2004)
- [55] Thomas Erl , “Service Oriented Architecture, Concepts, Technology and Design”. Prentice Hall.
- [56] Tsalgatidou, A., Pilioura, T., An Overview of Standards and Related Technology in Web Services, *International Journal of Distributed and Parallel Data Bases*, Special Issue on E-Services, 12(2); p. 135-162, Sep 2002.
- [57] Umeshwar Dayal, Meichun Hsu, Rivka Ladin, “Business Process Coordination: State of the Art, Trends, and Open Issues”.
- [58] Universal Description, Discovery, and Integration (UDDI) available at: <http://www.uddi.org>.
- [59] van den Heuvel, W.J.: *Integrating Modern Business Applications with Legacy Systems: A Software Component Perspective*. MIT Press, Cambridge, February (2007)

[60] Yannis Charalabidis, George Gionis, Dimitris Askounis, Peter Mayer, Kostas Kalaboukas, Richard Stevens, Harald Kuhn, Creating a Platform for End-to-End Business to Business and Government to Business Electronic Transactions in the new European Union: The GENESIS Project, eChallenges 2007 Conference, Den Haag, October 24th – 26th, 2007.

[61] Yendluri, P.:RosettaNet implementation framework (RNIF),Version 2.0, Technical report. RosettaNet, 2000.

[62] Web Services Atomic Transaction (WS-AtomicTransaction) available at: <http://msdn.microsoft.com/library/en-us/dnglobspec/html/WSAtomicTransaction.pdf>

[63] Web Services Business Process Execution Language (WSBPEL)[http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsbpel](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel)

[64] Web Services Coordination (WS-Coordination) available at: <http://msdn.microsoft.com/library/en-us/dnglobspec/html/WS-Coordination.pdf>

[65] Web Services Eventing (WS-Eventing) available at: <http://www-128.ibm.com/developerworks/webservices/library/specification/wseventing/>

[66] Web Services Modeling Ontology (WSMO) available at: <http://www.wsmo.org/>

[67] WS Choreography Description Language available at: <http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/>

[68] WS Choreography Model available at: <http://www.w3.org/TR/2004/WD-ws-chor-model-20040324/>

[69] WS-Composite Application Framework (WS-CAF) available at: <http://developers.sun.com/techttopics/webservices/wscaf/primer.pdf>

[70] WS-Security (Web Services Security): [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wss](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss)

[71] World Wide Web Consortium (W3C), <http://www.w3.org>