



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Τομέας Ηλεκτρικών Βιομηχανικών Διατάξεων και
Συστημάτων Αποφάσεων

Εργαστήριο Συστημάτων Αποφάσεων και Διοίκησης

Ενσωμάτωση καινοτομικών ψηφιακών υπηρεσιών σε βιομηχανικά προϊόντα

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Εμμανουήλ Κ. Ντάνου

Επιβλέπων Καθηγητής

Δημήτριος Ασκούνης

Αθήνα, Δεκέμβριος 2019



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Τομέας Ηλεκτρικών Βιομηχανικών Διατάξεων και
Συστημάτων Αποφάσεων

Εργαστήριο Συστημάτων Αποφάσεων και Διοίκησης

Ενσωμάτωση καινοτομικών ψηφιακών υπηρεσιών σε βιομηχανικά προϊόντα

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Εμμανουήλ Κ. Ντάνου

Συμβουλευτική Επιτροπή: Δημήτριος Ασκούνης
Ιωάννης Ψαρράς
Βασίλειος Ασημακόπουλος

Εγκρίθηκε από την επταμελή επιτροπή την 9η Δεκεμβρίου Δύο Χιλιάδες Δέκα Εννέα

.....
Δημήτριος Ασκούνης
Καθηγητής

.....
Ιωάννης Ψαρράς
Καθηγητής

.....
Βασίλειος Ασημακόπουλος
Καθηγητής

.....
Γρηγόριος Μέντζας
Καθηγητής

.....
Χρυσόστομος Δούκας
Αναπληρωτής Καθηγητής

.....
Θεοδώρα Βαρβαρίγου
Καθηγήτρια

.....
Ιωάννης Χαραλαμπίδης
Αναπληρωτής Καθηγητής
(Πανεπιστήμιο Αιγαίου)

Αθήνα, Δεκέμβριος 2019

.....
Εμμανουήλ Κ. Ντάνος

Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Ηλεκτρονικών Υπολογιστών ΕΜΠ

Copyright © Εμμανουήλ Κ. Ντάνος , 2019

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.

Περίληψη

Οι σύγχρονες βιομηχανικές επιχειρήσεις απομακρύνονται από το μοντέλο του αποκλειστικού παραγωγού υλικών προϊόντων, και προχωρούν στον σταδιακό εμπλουτισμό των φυσικών προϊόντων με συναφείς υπηρεσίες. Ο εμπλουτισμός αυτός φτάνει μέχρι τη δημιουργία ενιαίων «Συστημάτων Προϊόντος – Υπηρεσίας» (ΣΠΥ), τα οποία αντικαθιστούν τα αποκλειστικώς φυσικά προϊόντα στις καταναλωτικές και τις βιομηχανικές αγορές.

Η ενσωμάτωση υπηρεσιών στα σύγχρονα βιομηχανικά προϊόντα υποστηρίζεται από Τεχνολογίες Πληροφορικής και Επικοινωνιών. Οι ΤΠΕ, επιτρέπουν την παροχή προηγμένων υπηρεσιών, αφού παρέχουν τα απαραίτητα κανάλια επικοινωνίας, αυτοματισμούς, υλοποιούν την απαραίτητη επιχειρησιακή λογική και παρέχουν δυνατότητες επεξεργασίας αποθήκευσης και μετάδοσης δεδομένων.

Παρόλα αυτά, δεν υπάρχουν επαρκείς μεθοδολογίες και εργαλεία που να υποστηρίζουν την ανάπτυξη λογισμικού για τις ιδιαίτερες συνθήκες των ΣΠΥ. Σχεδόν όλες οι μεθοδολογίες ανάπτυξης ΣΠΥ στην υφιστάμενη βιβλιογραφία εστιάζουν σε στρατηγικές ή επιχειρησιακές πλευρές του ΣΠΥ και δεν υποστηρίζουν ικανοποιητικά την ανάπτυξη λογισμικού για την ενσωμάτωση ψηφιακών υπηρεσιών.

Με βάση την ανάλυση της τρέχουσας βιβλιογραφίας, ορίζεται ένα σύνολο από απαιτήσεις για κατάλληλες μεθοδολογίες και εργαλεία: υποστήριξη πολλών τύπων εφαρμογών, υποστήριξη ανάπτυξης με μοντέλα, παροχή Ολοκληρωμένου Περιβάλλοντος Ανάπτυξης, διασύνδεση με την πλατφόρμα εκτέλεσης των διαδικασιών και υποστήριξη υπηρεσιοστραφών αρχιτεκτονικών (Service – Oriented Architectures) και μεθόδων διαχείρισης επιχειρησιακών διαδικασιών (Business Process Management).

Για την ικανοποίηση των παραπάνω απαιτήσεων, και με βάση τη μεθοδολογία ανάπτυξης υπηρεσιών MDSEA (Model-Driven Service Engineering Architecture) προτείνεται μία κατάλληλη μεθοδολογία ανάπτυξης λογισμικού με μοντέλα. Η μεθοδολογία υποστηρίζει παράλληλα «δομικές» και «συμπεριφορικές» αναπαραστάσεις των υπηρεσιών και μπορεί να οδηγήσει τη δημιουργία λογισμικού, υπηρεσιών web, συνθέσεων υπηρεσιών και εκτελέσιμων επιχειρησιακών διαδικασιών. Μαζί με τη μεθοδολογία περιγράφεται η αρχιτεκτονική ενός Ολοκληρωμένου Περιβάλλοντος Ανάπτυξης το οποίο θα την υποστηρίζει. Το περιβάλλον ανάπτυξης θα διασυνδέεται με τα εργαλεία σχεδιασμού του συστήματος υπηρεσιών και την πλατφόρμα παροχής υπηρεσιών στην οποία θα εκτελείται το παραγόμενο λογισμικό.

Με σκοπό την αξιολόγηση της μεθοδολογίας και του προτεινόμενου εργαλείου, αναπτύχθηκε πρωτότυπη Πλατφόρμα Ανάπτυξης στα πλαίσια ευρύτερου ερευνητικού έργου. Η αξιολόγηση έγινε σε τρεις πιλοτικές εφαρμογές ενσωμάτωσης ψηφιακών υπηρεσιών σε βιομηχανικά προϊόντα, και διαπιστώθηκε η βελτιωμένη επικοινωνία και συνεργασία μεταξύ των εμπλεκόμενων στη διαδικασία ανάπτυξης λογισμικού.

Λέξεις κλειδιά: Σύστημα Προϊόντος – Υπηρεσίας, Μηχανική Λογισμικού, Χρήση Μοντέλων, Ενσωμάτωση εργαλείων ανάπτυξης, Ολοκληρωμένο Περιβάλλον Ανάπτυξης, Βιομηχανία

Abstract

Modern manufacturing business are moving away from producing mainly physical products and start enriching them with relevant value-added services. This can lead to the creation of unified “Product-Service Systems” (PSS), which can replace purely physical offerings in consumer and industrial markets.

The integration of services in modern industrial products is supported by Information and Communication Technology (ICT). ICT, and software applications in particular, enable the provision of advanced services by offering the necessary communication channels, automation, business logic and data processing, storage and transmission capabilities.

Nevertheless, there are no satisfactory methodologies and tools for the development of software for the particular circumstances of a Product – Service System. Nearly all PSS development methodologies in the current literature focus on strategic or operational aspects of the PSS, while mostly ignoring the development of the necessary software for the integration and implementation of digital services.

Based on an analysis of the current literature, a series of general requirements for appropriate methods and tools are defined: support for multiple types of software applications, support for model-driven development, provision of an Integrated Development Environment, interoperability with the service provision ICT infrastructure and support for Service – Oriented Architectures and Business Process Management technologies.

In order to satisfy these requirements, and based on the MDSEA (Model-Driven Service Engineering Architecture) framework, a suitable software development methodology is proposed. The methodology supports modelling and development using both “structural” and “behavioural” perspectives of the service system. The methodology can drive the development of standalone software, web services, service compositions and executable business processes. Additionally, the architecture of a suitable Integrated Development Environment is described. The development environment is interoperable with external service system modelling and engineering tools, as well as with the service provision ICT system.

In order to evaluate the methodology and the proposed tool, a prototype Development Platform was developed in the framework of a larger research project. The methodology and tools were applied in three industrial pilots involving the integration of digital services in industrial products and the development of Product – Service Systems. The evaluation demonstrated improved communication and collaboration between participants in the software engineering process.

Keywords: Product-Service System, Software Engineering, Model-Driven, Tool Integration, Integrated Development Environment, Manufacturing

Πίνακας περιεχομένων

1	ΕΙΣΑΓΩΓΗ	1
1.1	Το ΠΡΟΒΛΗΜΑ	1
1.2	ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ ΚΑΙ Η ΣΥΜΒΟΛΗ ΤΗΣ ΔΙΔΑΚΤΟΡΙΚΗΣ ΔΙΑΤΡΙΒΗΣ	3
1.3	Η ΔΟΜΗ ΤΗΣ ΔΙΑΤΡΙΒΗΣ.....	5
2	ΒΙΟΜΗΧΑΝΙΑ ΚΑΙ ΥΠΗΡΕΣΙΟΠΟΙΗΣΗ	7
2.1	Η ΕΝΝΟΙΑ ΤΗΣ ΥΠΗΡΕΣΙΑΣ ΚΑΙ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΥΠΗΡΕΣΙΩΝ	7
2.2	ΥΠΗΡΕΣΙΕΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑ.....	13
2.3	ΕΝΣΩΜΑΤΩΣΗ ΥΠΗΡΕΣΙΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΑ ΠΡΟΪΟΝΤΟΣ - ΥΠΗΡΕΣΙΑΣ	13
2.4	ΟΙΚΟΣΥΣΤΗΜΑΤΑ ΒΙΟΜΗΧΑΝΙΚΩΝ ΥΠΗΡΕΣΙΩΝ	17
2.5	Η ΜΕΤΑΒΑΣΗ ΠΡΟΣ ΕΝΑ ΜΟΝΤΕΛΟ ΠΑΡΟΧΗΣ ΥΠΗΡΕΣΙΩΝ.....	21
3	ΘΕΩΡΗΤΙΚΟ ΚΑΙ ΤΕΧΝΟΛΟΓΙΚΟ ΥΠΟΒΑΘΡΟ	27
3.1	ΥΠΗΡΕΣΙΕΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ	27
3.2	ΥΠΗΡΕΣΙΟΣΤΡΑΦΕΙΣ ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΕΠΙΧΕΙΡΗΣΙΑΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ.....	30
3.2.1	<i>Υπηρεσιοστραφείς αρχιτεκτονικές</i>	30
3.2.2	<i>Συστήματα διαχείρισης επιχειρησιακών διαδικασιών</i>	34
3.2.3	<i>Συμπεράσματα</i>	37
3.3	ΜΕΘΟΔΟΛΟΓΙΕΣ ΚΑΙ ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ.....	38
3.3.1	<i>Ανάπτυξη λογισμικού με χρήση μοντελοποίησης: η μεθοδολογία MDA (Model-Driven Architecture)</i>	38
3.3.2	<i>Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης λογισμικού</i>	43
3.4	ΜΕΘΟΔΟΛΟΓΙΕΣ ΣΧΕΔΙΑΣΜΟΥ ΣΥΣΤΗΜΑΤΩΝ ΥΠΗΡΕΣΙΩΝ.....	46
3.4.1	<i>Στάδια ανάπτυξης νέας υπηρεσίας και ΣΠΥ</i>	46
3.4.2	<i>Μηχανική υπηρεσιών (Service Engineering) για την ανάπτυξη και σχεδιασμό νέων υπηρεσιών</i>	47
3.4.3	<i>Μεθοδολογίες ανάπτυξης Συστημάτων Προϊόντος Υπηρεσίας και υποδομών λογισμικού</i> <i>49</i>	
3.4.4	<i>Η μεθοδολογία ανάπτυξης Συστημάτων Υπηρεσιών MDSEA</i>	54
3.5	ΑΠΑΙΤΗΣΕΙΣ ΓΙΑ ΜΕΘΟΔΟΛΟΓΙΕΣ ΚΑΙ ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΕΦΑΡΜΟΓΕΣ ΥΠΗΡΕΣΙΟΠΟΙΗΣΗΣ ΣΤΗ ΒΙΟΜΗΧΑΝΙΑ	61
3.5.1	<i>Απαίτηση 1: Ευελιξία και γενικότητα εργαλείων και μεθοδολογίας</i>	62
3.5.2	<i>Απαίτηση 2: Χρήση κατάλληλης μεθοδολογίας μοντελοποίησης</i>	62
3.5.3	<i>Απαίτηση 3: Υποστήριξη δομικής και συμπεριφορικής προσέγγισης</i>	65
3.5.4	<i>Απαίτηση 4: Ολοκλήρωση εργαλείων ανάπτυξης</i>	66
3.5.5	<i>Απαίτηση 5: Ολοκλήρωση ροής και εργαλείων ανάπτυξης με το σύστημα εκτέλεσης του λογισμικού υπηρεσιών</i>	68
3.5.6	<i>Απαίτηση 6: Τεχνολογική υποδομή: SOA και BPM</i>	68
3.5.7	<i>Συμπεράσματα</i>	69
3.6	ΜΕΘΟΔΟΛΟΓΙΕΣ ΚΑΙ ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΣΥΣΤΗΜΑΤΑ SOA ΚΑΙ BPM.....	70
3.6.1	<i>Μεθοδολογίες και εργαλεία</i>	70
3.6.2	<i>Αξιολόγηση</i>	78
4	ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΙ ΕΡΓΑΛΕΙΟ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΥΠΗΡΕΣΙΩΝ	81
4.1	ΜΕΘΟΔΟΛΟΓΙΑ	81
4.1.1	<i>Γενικά</i>	81
4.1.2	<i>Μετασχηματισμός μοντέλων στην MDA</i>	81
4.1.3	<i>Δομική ροή εργασίας: UML προς Java</i>	84

4.1.4	Συμπεριφορική ροή εργασίας: Μοντέλα BPMN 2.0 προς εκτελέσιμες επιχειρησιακές διαδικασίες και συνθέσεις υπηρεσιών	87
4.2	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΥΠΗΡΕΣΙΩΝ	91
4.2.1	Στόχοι και πλαίσιο συστήματος.....	91
4.2.2	Προδιαγραφές Πλατφόρμας Ανάπτυξης Λογισμικού Υπηρεσιών.....	92
4.2.3	Προτεινόμενη Αρχιτεκτονική.....	103
4.2.4	Υποστήριξη των ροών εργασίας από την προτεινόμενη αρχιτεκτονική.....	108
5	ΕΦΑΡΜΟΓΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ	113
5.1	ΕΙΣΑΓΩΓΗ.....	113
5.2	ΤΟ ΕΡΕΥΝΗΤΙΚΟ ΈΡΓΟ MSEE.....	113
5.2.1	Στόχοι και αποτελέσματα	113
5.2.2	Αρχιτεκτονική της Πλατφόρμας Υπηρεσιών MSEE	115
5.2.3	Ανάπτυξη λογισμικού υπηρεσιών στα πλαίσια του MSEE	117
5.2.4	Πλατφόρμα Ανάπτυξης Υπηρεσιών Λογισμικού.....	121
5.3	ΠΡΩΤΟΤΥΠΟ	123
5.3.1	Περιβάλλον Eclipse.....	126
5.3.2	Κύριες Λειτουργίες της Πλατφόρμας ανάπτυξης.....	126
5.3.3	Διαλειτουργικότητα με άλλα στοιχεία της πλατφόρμας ΤΠΕ του MSEE	141
5.4	ΑΞΙΟΛΟΓΗΣΗ.....	152
5.4.1	Προσέγγιση αξιολόγησης	152
5.4.2	Διαδικασία αξιολόγησης	154
5.4.3	Αποτελέσματα	155
6	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ	159
6.1	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	159
6.2	ΠΡΟΟΠΤΙΚΕΣ.....	164
	ΒΙΒΛΙΟΓΡΑΦΙΑ	168
	ΠΑΡΑΡΤΗΜΑ.....	185
	ΕΝΔΕΙΚΤΙΚΟΙ ΚΑΝΟΝΕΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ ACCELEO (MOFM2T) UML ΠΡΟΣ JAVA.....	185

Πίνακας πινάκων

ΠΙΝΑΚΑΣ 1: ΛΟΓΙΚΗ ΥΛΙΚΩΝ ΑΓΑΘΩΝ ΚΑΙ ΛΟΓΙΚΗ ΥΠΗΡΕΣΙΩΝ	11
ΠΙΝΑΚΑΣ 2: ΚΡΙΤΗΡΙΑ ΠΡΟΣΔΙΟΡΙΣΜΟΥ ΓΙΑ ΤΟ ΕΠΙΠΕΔΟ ΕΝΣΩΜΑΤΩΣΗΣ ΥΠΗΡΕΣΙΩΝ (MARTINEZ, BASTL, KINGSTON, & EVANS, 2010)	23
ΠΙΝΑΚΑΣ 3: ΠΑΡΑΛΛΗΛΗ ΑΝΑΠΤΥΞΗ ΒΙΟΜΗΧΑΝΙΚΟΥ ΠΡΟΪΟΝΤΟΣ ΚΑΙ ΤΕΧΝΙΚΗΣ ΥΠΗΡΕΣΙΑΣ (AURICH, FUCHS & WAGENKNECHT, 2006).....	47
ΠΙΝΑΚΑΣ 4: ΑΝΤΙΣΤΟΙΧΙΣΗ ΟΡΩΝ ΤΗΣ MDA ΜΕ ΕΝΝΟΙΕΣ ΤΗΣ MDSEA	56
ΠΙΝΑΚΑΣ 5: ΑΠΑΙΤΗΣΕΙΣ ΓΙΑ ΜΙΑ ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΙ ΤΟ ΑΝΤΙΣΤΟΙΧΟ ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ ΓΙΑ ΕΝΣΩΜΑΤΩΣΗ ΥΠΗΡΕΣΙΩΝ ΣΕ ΒΙΟΜΗΧΑΝΙΚΑ ΠΡΟΪΟΝΤΑ.	69
ΠΙΝΑΚΑΣ 6: ΑΞΙΟΛΟΓΗΣΗ ΥΦΙΣΤΑΜΕΝΩΝ ΠΡΟΣΕΓΓΙΣΕΩΝ ΩΣ ΠΡΟΣ ΤΗΝ ΚΑΤΑΛΛΗΛΟΤΗΤΑ ΓΙΑ ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΒΙΟΜΗΧΑΝΙΚΕΣ ΥΠΗΡΕΣΙΕΣ	78
ΠΙΝΑΚΑΣ 7: ΣΧΕΔΙΑΣΤΙΚΕΣ ΕΙΣΟΔΟΙ ΚΑΙ ΑΝΑΜΕΝΟΜΕΝΕΣ ΕΞΟΔΟΙ ΤΗΣ ΡΟΗΣ ΕΡΓΑΣΙΩΝ	84
ΠΙΝΑΚΑΣ 8: ΑΝΤΙΣΤΟΙΧΙΣΗ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΜΟΝΤΕΛΟΥ TIM ΣΕ ΣΤΟΙΧΕΙΑ ΤΗΣ JAVA	85
ΠΙΝΑΚΑΣ 9: ΚΥΡΙΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ΑΝΑΠΤΥΞΗΣ, ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΜΕΘΟΔΟΙ ΥΛΟΠΟΙΗΣΗΣ	128
ΠΙΝΑΚΑΣ 10: ΑΡΙ ΠΕΛΑΤΗ ΑΠΟΘΕΤΗΡΙΟΥ.....	144

Πίνακας Σχημάτων

ΣΧΗΜΑ 1: Η ΣΥΜΒΟΛΗ ΤΗΣ ΔΙΔΑΚΤΟΡΙΚΗΣ ΔΙΑΤΡΙΒΗΣ	5
ΣΧΗΜΑ 2: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΙΧΕΙΡΗΣΙΑΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ (WESKE, 2007)	37
ΣΧΗΜΑ 3: ΓΕΝΙΚΗ ΕΠΙΣΚΟΠΗΣΗ ΤΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ MDSEA ΤΟΥ MSEE (CHEN, DUCQ, DOUMEINGTS, ZACHARIEWICZ, & ALIX, 2012; DUCQ, CHEN, & ALIX, 2012; AGOSTINHO, BAZOUN, ZACHAREWICZ, DUCQ, BOYE, & JARDIM-GONCALVES, 2013).....	56
ΣΧΗΜΑ 4: ΣΤΟΙΧΕΙΑ ΤΟΥ ΕΠΙΠΕΔΟΥ BSM ΣΤΗ ΜΕΘΟΔΟΛΟΓΙΑ MDSEA ΚΑΙ ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΜΕΘΟΔΟΙ, ΕΡΓΑΛΕΙΑ ΚΑΙ ΠΡΟΤΥΠΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ (DUCQ, CHEN, & ALIX, 2012)	57
ΣΧΗΜΑ 5: ΣΤΟΙΧΕΙΑ ΤΟΥ ΕΠΙΠΕΔΟΥ TIM ΣΤΗ ΜΕΘΟΔΟΛΟΓΙΑ MDSEA ΚΑΙ ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΜΕΘΟΔΟΙ, ΕΡΓΑΛΕΙΑ ΚΑΙ ΠΡΟΤΥΠΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ (DUCQ, CHEN, & ALIX, 2012)	58
ΣΧΗΜΑ 6: ΣΤΟΙΧΕΙΑ ΤΟΥ ΕΠΙΠΕΔΟΥ TSM ΣΤΗ ΜΕΘΟΔΟΛΟΓΙΑ MDSEA ΚΑΙ ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΜΕΘΟΔΟΙ, ΕΡΓΑΛΕΙΑ ΚΑΙ ΠΡΟΤΥΠΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ (DUCQ, CHEN, & ALIX, 2012).....	59
ΣΧΗΜΑ 7: ΓΕΝΙΚΗ ΔΙΑΔΙΚΑΣΙΑ TIM ΠΡΟΣ ΕΚΤΕΛΕΣΙΜΑ ΑΡΧΕΙΑ	83
ΣΧΗΜΑ 8: ΠΑΡΑΔΕΙΓΜΑ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ ΕΝΟΣ ΜΟΝΤΕΛΟΥ TIM ΣΕ TSM ΓΙΑ ΤΗ ΓΛΩΣΣΑ JAVA.....	86
ΣΧΗΜΑ 9: ΠΑΡΑΔΕΙΓΜΑ ΠΛΑΙΣΙΟΥ ΚΩΔΙΚΑ JAVABEAN ΠΟΥ ΕΧΕΙ ΠΑΡΑΧΘΕΙ ΑΠΟ ΜΟΝΤΕΛΟ TSM	87
ΣΧΗΜΑ 10: ΠΑΡΑΔΕΙΓΜΑ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ ΜΟΝΤΕΛΟΥ BPMN 2.0 ΑΠΟ TIM ΣΕ TSM ΜΕΣΩ ΑΝΤΙΣΤΟΙΧΙΣΗΣ ΚΑΙ ΕΜΠΛΟΥΤΙΣΜΟΥ	89
ΣΧΗΜΑ 11: ΕΝΔΕΙΚΤΙΚΟΣ ΚΩΔΙΚΑΣ BPMN 2.0 ΣΕ ΜΟΡΦΗ XML	90
ΣΧΗΜΑ 12: ΤΟ ΠΛΑΙΣΙΟ ΚΑΙ ΟΙ ΑΛΛΗΛΕΠΙΔΡΑΣΕΙΣ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΥΠΗΡΕΣΙΩΝ	92
ΣΧΗΜΑ 13: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΛΑΤΦΟΡΜΑΣ ΑΝΑΠΤΥΞΗΣ.....	104
ΣΧΗΜΑ 14: ΣΤΟΙΧΕΙΑ ΠΡΟΣΒΑΣΗΣ ΚΑΙ ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑΣ ΑΠΟΘΕΤΗΡΙΟΥ ΜΟΝΤΕΛΩΝ	107
ΣΧΗΜΑ 15: Η ΔΟΜΙΚΗ ΡΟΗ ΕΡΓΑΣΙΑΣ ΟΠΩΣ ΥΛΟΠΟΙΕΙΤΑΙ ΑΠΟ ΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΑ ΣΤΟΙΧΕΙΑ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ΑΝΑΠΤΥΞΗΣ, ΟΡΓΑΝΩΜΕΝΗ ΣΕ ΒΗΜΑΤΑ.....	109
ΣΧΗΜΑ 16: Η ΣΥΜΠΕΡΙΦΟΡΙΚΗ ΡΟΗ ΕΡΓΑΣΙΑΣ ΟΠΩΣ ΥΛΟΠΟΙΕΙΤΑΙ ΑΠΟ ΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΑ ΣΤΟΙΧΕΙΑ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ΑΝΑΠΤΥΞΗΣ, ΟΡΓΑΝΩΜΕΝΗ ΣΕ ΒΗΜΑΤΑ.....	111
ΣΧΗΜΑ 17: ΣΤΟΙΧΕΙΑ ΤΗΣ ΠΡΩΤΟΤΥΠΗΣ ΠΛΑΤΦΟΡΜΑΣ ΤΠΕ ΤΟΥ MSEE (MSEE, MSEE SERVICE-SYSTEM FUNCTIONAL AND MODULAR ARCHITECTURE, 2012)	116
ΣΧΗΜΑ 18: ΔΙΑΧΩΡΙΣΜΟΣ ΤΗΣ ΑΝΑΠΤΥΞΗΣ ΤΩΝ ΥΠΟΔΟΜΩΝ ΤΠΕ ΣΤΗ ΜΕΘΟΔΟΛΟΓΙΑ MDSEA ΚΑΤΑ ΤΗ ΔΙΑΔΙΚΑΣΙΑ ΤΗΣ ΕΝΣΩΜΑΤΩΣΗΣ ΥΠΗΡΕΣΙΩΝ (MSEE, SERVICE CONCEPTS, MODELS AND METHOD: MODEL DRIVEN SERVICE ENGINEERING, 2012).....	119

ΣΧΗΜΑ 19: ΠΑΡΑΛΛΗΛΕΣ ΡΟΕΣ ΑΝΑΠΤΥΞΗΣ ΤΩΝ ΥΠΟΔΟΜΩΝ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΥΠΗΡΕΣΙΩΝ, ΣΤΑ ΠΛΑΙΣΙΑ ΤΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ MDSEA.....	120
ΣΧΗΜΑ 20: ΔΙΑΣΥΝΔΕΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ΑΝΑΠΤΥΞΗΣ ΜΕ ΤΟ ΥΠΟΛΟΙΠΟ ΣΥΣΤΗΜΑ.....	122
ΣΧΗΜΑ 21: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ ΠΡΩΤΟΤΥΠΗΣ ΠΛΑΤΦΟΡΜΑΣ ΑΝΑΠΤΥΞΗΣ	124
ΣΧΗΜΑ 22: ΔΟΜΙΚΗ ΡΟΗ ΕΡΓΑΣΙΑΣ ΣΤΗ ΠΡΩΤΟΤΥΠΗ ΠΛΑΤΦΟΡΜΑ ΑΝΑΠΤΥΞΗΣ	129
ΣΧΗΜΑ 23: ΠΑΡΑΔΕΙΓΜΑ ΕΠΕΚΤΑΣΗΣ ΜΕΤΑ-ΜΟΝΤΕΛΟΥ UML ΓΙΑ ΒΙΟΜΗΧΑΝΙΚΕΣ ΥΠΗΡΕΣΙΕΣ ΣΤΗ ΓΛΩΣΣΑ JAVA	130
ΣΧΗΜΑ 24: ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΙ ΕΜΠΛΟΥΤΙΣΜΟΣ ΔΙΑΓΡΑΜΜΑΤΟΣ ΚΛΑΣΕΩΝ UML ΣΤΗ ΠΛΑΤΦΟΡΜΑ ΑΝΑΠΤΥΞΗΣ.....	131
ΣΧΗΜΑ 25: ΚΑΝΟΝΕΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ ΜΟΝΤΕΛΟΥ TSM ΣΕ JAVA ΣΤΗ ΠΛΑΤΦΟΡΜΑ ΑΝΑΠΤΥΞΗΣ	134
ΣΧΗΜΑ 26: ΠΛΑΙΣΙΟ ΚΩΔΙΚΑ ΚΛΑΣΗΣ JAVA ΠΟΥ ΠΑΡΑΧΘΗΚΕ ΑΥΤΟΜΑΤΑ ΑΠΟ ΤΗΝ ΠΛΑΤΦΟΡΜΑ ΑΝΑΠΤΥΞΗΣ	135
ΣΧΗΜΑ 27: ΕΠΕΞΕΡΓΑΣΙΑ ΚΩΔΙΚΑ JAVA ΠΟΥ ΠΑΡΑΧΘΗΚΕ ΑΠΟ ΕΝΑ ΜΟΝΤΕΛΟ TSM ΣΤΗ ΠΛΑΤΦΟΡΜΑ ΑΝΑΠΤΥΞΗΣ	135
ΣΧΗΜΑ 28: ΣΥΜΠΕΡΙΦΟΡΙΚΗ ΡΟΗ ΕΡΓΑΣΙΑΣ ΣΤΗ ΠΡΩΤΟΤΥΠΗ ΠΛΑΤΦΟΡΜΑ ΑΝΑΠΤΥΞΗΣ.....	137
ΣΧΗΜΑ 29: ΕΠΕΞΕΡΓΑΣΙΑ ΜΟΝΤΕΛΟΥ BPMN ΣΤΗ ΠΡΩΤΟΤΥΠΗ ΠΛΑΤΦΟΡΜΑ ΑΝΑΠΤΥΞΗΣ	139
ΣΧΗΜΑ 30: ΑΝΑΖΗΤΗΣΗ ΚΑΙ ΕΝΣΩΜΑΤΩΣΗ ΥΦΙΣΤΑΜΕΝΩΝ WEB SERVICES ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΥΠΗΡΕΣΙΩΝ ΤΟΥ MSEE ΣΤΗΝ ΕΠΙΧΕΙΡΗΣΙΑΚΗ ΔΙΑΔΙΚΑΣΙΑ	140
ΣΧΗΜΑ 31: ΡΟΗ ΕΡΓΑΣΙΑΣ ΓΙΑ ΤΗΝ ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΜΕΣΩ ΔΙΑΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑΣ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ΤΟΥ MSEE	142
ΣΧΗΜΑ 32: MODEL REPOSITORY VIEW - ΓΡΑΦΙΚΗ ΔΙΕΠΑΦΗ ΑΠΟΘΕΤΗΡΙΟΥ	145
ΣΧΗΜΑ 33: ΕΚΤΕΛΕΣΗ ΤΟΥ ΟΔΗΓΟΥ ΕΙΣΑΓΩΓΗΣ ΜΟΝΤΕΛΟΥ	146
ΣΧΗΜΑ 34: ΣΥΝΔΕΣΗ ΣΤΟ ΑΠΟΘΕΤΗΡΙΟ ΜΟΝΤΕΛΩΝ	146
ΣΧΗΜΑ 35: ΕΠΙΛΟΓΗ ΜΟΝΤΕΛΩΝ ΓΙΑ ΕΙΣΑΓΩΓΗ	147
ΣΧΗΜΑ 36: ΕΠΙΛΟΓΗ ΠΡΟΟΡΙΣΜΟΥ ΓΙΑ ΤΗΝ ΕΙΣΑΓΩΓΗ ΜΟΝΤΕΛΩΝ	147
ΣΧΗΜΑ 37: ΚΑΤΗΓΟΡΙΕΣ ΥΠΗΡΕΣΙΩΝ ΣΤΗ ΠΛΑΤΦΟΡΜΑ ΠΑΡΟΧΗΣ ΥΠΗΡΕΣΙΩΝ	150
ΣΧΗΜΑ 38: ΠΕΡΙΓΡΑΦΕΣ ΥΠΗΡΕΣΙΩΝ ΣΥΓΚΕΚΡΙΜΕΝΗΣ ΚΑΤΗΓΟΡΙΑΣ	150
ΣΧΗΜΑ 39: ΡΟΗ ΕΡΓΑΣΙΑΣ SA-WSDL ΠΡΟΣ ΚΩΔΙΚΑ ΠΕΛΑΤΗ JAVA	152

1 Εισαγωγή

1.1 Το Πρόβλημα

Με σκοπό τη δημιουργία ανταγωνιστικού πλεονεκτήματος, οι σύγχρονες βιομηχανικές επιχειρήσεις μπορούν να εφαρμόσουν τη στρατηγική της «υπηρεσιοποίησης» (servitisation). Αυτή απαιτεί την απομάκρυνση από το μοντέλο του αποκλειστικού παραγωγού υλικών προϊόντων, και τον σταδιακό εμπλουτισμό των φυσικών προϊόντων που παράγουν με συναφείς υπηρεσίες. Ο εμπλουτισμός αυτός φτάνει μέχρι τη δημιουργία ενιαίων «Συστημάτων Προϊόντος – Υπηρεσίας» (ΣΠΥ), τα οποία αντικαθιστούν τα αποκλειστικώς φυσικά προϊόντα στις καταναλωτικές και τις βιομηχανικές αγορές.

Παραδοσιακά, μια βιομηχανική εταιρεία παραγωγής ορίζεται ως αυτή που «κατασκευάζει» και παρέχει φυσικά προϊόντα στους πελάτες. Μαζί με τα προϊόντα αυτά, η επιχείρηση μπορεί επίσης να παρέχει ένα περιορισμένο σύνολο των υποστηρικτικών υπηρεσιών, όπως είναι η εργασία παράδοσης ή συντήρησης. Στη περίπτωση της υπηρεσιοποίησης, οι προσφερόμενες υπηρεσίες αποκτούν κεντρικό ρόλο στη προσφερόμενη προστιθέμενη αξία, σε σχέση με την απλή απόκτηση του προϊόντος.

Για παράδειγμα, η παραδοσιακή κύρια δραστηριότητα ενός κατασκευαστή βιομηχανικού εξοπλισμού είναι ο σχεδιασμός παραγωγή και πώληση εργαλειομηχανών, μαζί με βασικές υπηρεσίες παράδοσης, εγκατάστασης και συντήρησης. Ωστόσο, ο κατασκευαστής μπορεί να ενσωματώσει περαιτέρω υπηρεσίες στο προϊόν του: μελετητικές και συμβουλευτικές υπηρεσίες πριν την αγορά, υπηρεσίες προμήθειας αναλωσίμων, υπηρεσίες απόσυρσης και ανακύκλωσης στο τέλος ζωής. Ο κατασκευαστής μπορεί να φτάσει στο σημείο να προσφέρει το προϊόν το με τη μορφή υπηρεσίας, π.χ. μέσω της μίσθωσής του προϊόντος στον πελάτη (στις εγκαταστάσεις του πελάτη), ή τη μίσθωσή των λειτουργιών του (σε εγκαταστάσεις του κατασκευαστή).

Η ενσωμάτωση υπηρεσιών στα σύγχρονα βιομηχανικά προϊόντα υποστηρίζεται από Τεχνολογίες Πληροφορικής και Επικοινωνιών. Από τη μία πλευρά, το απαραίτητο επίπεδο συνεργασίας ανθρώπων και οργανισμών, καθώς και των νέων μηχανισμών διοίκησης και διαχείρισης, απαιτεί την ανταλλαγή δεδομένων και την διαλειτουργικότητα πληροφοριακών συστημάτων. Από την άλλη, οι ΤΠΕ επιτρέπουν την παροχή προηγμένων υπηρεσιών, αφού παρέχουν τα απαραίτητα κανάλια επικοινωνίας και αυτοματισμούς, ενώ υλοποιούν την απαραίτητη επιχειρησιακή λογική και παρέχουν δυνατότητες επεξεργασίας αποθήκευσης και μετάδοσης δεδομένων. Έτσι, για παράδειγμα, επιτρέπουν δυνατότητες όπως την απομακρυσμένη παρακολούθηση μισθωμένων οχημάτων, την επιτήρηση και εντοπισμό βλαβών σε υπηρεσίες συντήρησης, την αλληλεπίδραση με τον πελάτη όπου και αν βρίσκεται, την αυτόματη χρέωση και πληρωμή, και άλλες λειτουργίες.

Η ανάπτυξη εφαρμογών λογισμικού υπηρεσιών για την παροχή προηγμένων ψηφιακών υπηρεσιών στα βιομηχανικά προϊόντα και στα «Συστήματα Προϊόντος – Υπηρεσίας» περιλαμβάνει ιδιαίτερες προκλήσεις:

- Η ανάπτυξη ενός σύγχρονου Συστήματος Προϊόντος – Υπηρεσίας απαιτεί την παράλληλη ανάπτυξη του προϊόντος, της υπηρεσίας και των δομών που θα εξασφαλίζουν την παροχή τους. Κατά συνέπεια, η ανάπτυξη των υποδομών ΤΠΕ και του επιχειρησιακού λογισμικού πρέπει να συγχρονίζεται με τις δραστηριότητες αυτές, έτσι ώστε να διασφαλίζεται η μετάφραση των επιχειρησιακών απαιτήσεων σε απτές και λειτουργικές εφαρμογές.
- Σε περίπλοκα συνεργατικά δίκτυα υπάρχουν πολλαπλοί εμπλεκόμενοι, όπως παραγωγοί, προμηθευτές, πάροχοι υπηρεσιών, τελικοί χρήστες κτλ., με αποτέλεσμα περισσότερο περίπλοκες διαδικασίες συλλογής απαιτήσεων, σχεδιασμού και υλοποίησης του τελικού λογισμικού. Υπάρχει η σαφής ανάγκη για επιτάχυνση και εξομάλυνση των συγκεκριμένων διαδικασιών.
- Η ανάπτυξη λογισμικού για την υποστήριξη βιομηχανικών υπηρεσιών απαιτεί τη συνεργασία στελεχών με διαφορετικές εξειδικεύσεις, ορολογίες, και αντιλήψεις του προβλήματος. Τυπικά, στη διαδικασία αυτή εμπλέκονται ειδικοί του συγκεκριμένου βιομηχανικού κλάδου (domain experts) και διοικητικά στελέχη, αλλά και τεχνικοί όπως αναλυτές, προγραμματιστές κ.α. Αυτό οφείλεται στο ότι κατά την ανάπτυξη του υποστηρικτικού λογισμικού απαιτείται η μετάφραση του επιχειρησιακού σχεδιασμού (π.χ. του τρόπου παροχής ενός συστήματος προϊόντος – υπηρεσίας) σε τεχνικές προδιαγραφές και, τελικά, λογισμικό. Συνεπώς πρέπει να διασφαλιστεί η απρόσκοπτη επικοινωνία ανάμεσα στους προγραμματιστές, τους ειδικούς στο πεδίο εφαρμογής και τα διοικητικά στελέχη.
- Οι βιομηχανικές επιχειρήσεις δεν έχουν, κατά κανόνα, την ανάπτυξη λογισμικού ως κύρια δραστηριότητά τους. Οι λειτουργίες ανάπτυξης λογισμικού αποτελούν μόνο ένα μέρος της δραστηριότητάς τους, καθώς το λογισμικό που αναπτύσσουν πρέπει να υποστηρίξει τα κύρια προϊόντα και υπηρεσίες που προσφέρουν. Για τον λόγο αυτό, η ανάπτυξη λογισμικού είναι δευτερεύουσα λειτουργία, η οποία λειτουργεί συχνά υπό περιορισμούς χρόνου και πόρων, ή ανατίθεται σε τρίτους, με τους οποίους πρέπει να υπάρχει απρόσκοπτη συνεργασία.
- Στις σύγχρονες αγορές υπάρχει σημαντική πίεση για την μείωση του χρόνου διάθεσης στην αγορά (time-to-market) νέων προϊόντων και υπηρεσιών, δηλαδή τη μείωση του χρόνου από τη σύλληψή τους μέχρι το λανσάρισμα. Κατά συνέπεια, είναι απαραίτητη η μείωση του χρόνου σχεδιασμού, ανάπτυξης, δοκιμών και θέσης σε λειτουργία των απαιτούμενων εφαρμογών και υποδομών.
- Μία ευρεία γκάμα προϊόντων και υπηρεσιών απαιτεί μία εξίσου ευρεία σειρά υποστηρικτικών λογισμικών, ενώ, λόγω ανταγωνιστικών πιέσεων, οι παρεχόμενες υπηρεσίες/ προϊόντα πρέπει να ενημερώνονται συνεχώς, άρα πρέπει να αναβαθμίζονται και οι σχετικές εφαρμογές. Ο χρόνος και το κόστος ανάπτυξης μπορεί να ελεγχθεί με την επαναχρησιμοποίηση κώδικα, την εφαρμογή αυτοματισμών και μεθόδων ταχείας διόρθωσης σφαλμάτων και θέσης σε λειτουργία των εφαρμογών.
- Η παροχή των βιομηχανικών υπηρεσιών περιλαμβάνει πολλαπλές τεχνολογικές πλατφόρμες και συστήματα. Ένα σύστημα ΤΠΕ για την υποστήριξη σύγχρονων Συστημάτων Προϊόντος – Υπηρεσίας αξιοποιεί σύγχρονες διαδικτυακές τεχνολογίες, υπηρεσιοστραφείς αρχιτεκτονικές, βάσεις δεδομένων, έμπειρα συστήματα, συστήματα επίγνωσης πλαισίου κτλ. Επιπλέον, καθώς πολλές από τις τελικές

εφαρμογές υποστηρίζουν φυσικά προϊόντα, περιλαμβάνουν την χρήση ενσωματωμένου λογισμικού (embedded software) αισθητήρων, ενεργοποιητών, αναγνώστων barcode και άλλων ηλεκτρονικών και ηλεκτρομηχανικών διατάξεων. Οι εφαρμογές αυτές πρέπει να μπορούν να συνδυάσουν επιχειρησιακή λογική (όπως συστήματα υποστήριξης αποφάσεων, προγραμματισμού, λογιστήριο, λήψη παραγγελιών κ.α.) με λειτουργίες ενσωματωμένων συστημάτων. Η προσέγγιση αυτή απαιτεί την απαραίτητη ευελιξία από τις αντίστοιχες μεθοδολογίες και περιβάλλοντα ανάλυσης και ανάπτυξης λογισμικού.

Για την αντιμετώπιση των παραπάνω προκλήσεων, είναι κρίσιμη η ανάπτυξη των κατάλληλων εργαλείων και μεθοδολογιών, οι οποίες θα επιτρέπουν την ανάπτυξη του απαραίτητου λογισμικού κάτω από αυτές τις ιδιαίτερες συνθήκες.

1.2 Το Αντικείμενο και η συμβολή της Διδακτορικής Διατριβής

Αντικείμενο της παρούσας διδακτορικής διατριβής είναι η μελέτη και ο σχεδιασμός μεθόδων και εργαλείων πληροφορικής για την ενσωμάτωση ψηφιακών υπηρεσιών σε βιομηχανικά προϊόντα, με απώτερο στόχο τη δημιουργία και λειτουργία προηγμένων βιομηχανικών Συστημάτων Προϊόντος – Υπηρεσίας (ΣΠΥ).

Η ενσωμάτωση υπηρεσιών σε βιομηχανικά προϊόντα και η δημιουργία Συστημάτων Προϊόντος - Υπηρεσίας είναι κρίσιμος παράγοντας για την ανταγωνιστικότητα της βιομηχανίας. Η επιτυχής παροχή προηγμένων υπηρεσιών και σύγχρονων ΣΠΥ εξαρτάται από την εφαρμογή Τεχνολογιών Πληροφορικής και Επικοινωνιών (ΤΠΕ).

Ωστόσο, υπάρχει ένα κρίσιμο μεθοδολογικό και τεχνικό κενό. Τη στιγμή αυτή, δεν έχουν μελετηθεί επαρκώς οι μεθοδολογίες για την ανάπτυξη λογισμικού ψηφιακών υπηρεσιών για βιομηχανικά προϊόντα, λαμβάνοντας ταυτόχρονα υπόψη τις ιδιαίτερες συνθήκες της υπηρεσιοποίησης στη βιομηχανία. Οι υφιστάμενες μεθοδολογίες για την ανάπτυξη Συστημάτων Προϊόντος Υπηρεσίας δίνουν δευτερεύουσα σημασία στην ανάπτυξη των υποδομών ΤΠΕ του Συστήματος Προϊόντος - Υπηρεσίας. Το αποτέλεσμα είναι ότι ο σχεδιασμός και ανάπτυξη λογισμικού για την ενσωμάτωση ψηφιακών υπηρεσιών σε προϊόντα στα ιδιαίτερα πλαίσια υλοποίησης ενός ΣΠΥ δεν έχει εξεταστεί σχεδόν καθόλου. Ομοίως, δεν υπάρχει καμία ερευνητική προσπάθεια που να μελετά τα εργαλεία ανάπτυξης λογισμικού σε σχέση με τις ανάγκες της υποστήριξης των βιομηχανικών ΣΠΥ.

Έτσι, τη στιγμή αυτή χρησιμοποιούνται για τους σκοπούς της ενσωμάτωσης ψηφιακών υπηρεσιών, συμβατικές και γενικού σκοπού, μεθοδολογίες σχεδιασμού και ανάπτυξης λογισμικού, προερχόμενες από τα αντίστοιχα ειδικά πεδία της επιστήμης υπολογιστών (π.χ. βάσεις δεδομένων, υποστήριξη απόφασης, ενσωματωμένο λογισμικό κ.α.). **Η χρήση τους όμως γίνεται χωρίς να έχει διερευνηθεί ποτέ η εφαρμογή μεθοδολογιών και εργαλείων εξειδικευμένων, στις ιδιαίτερες ανάγκες της υπηρεσιοποίησης στη βιομηχανία.**

Το κενό εντοπίζεται σε όλα τα βήματα της αλυσίδας του προβλήματος:

- Δεν έχουν καθοριστεί τα χαρακτηριστικά που πρέπει να έχουν οι μεθοδολογίες και τα εργαλεία ανάπτυξης λογισμικού για την ενσωμάτωση ψηφιακών υπηρεσιών.

- Δεν έχουν προταθεί και μελετηθεί μεθοδολογίες ανάπτυξης λογισμικού προσαρμοσμένες στις ανάγκες της υπηρεσιοποίησης στη βιομηχανία.
- Δεν έχει μελετηθεί η μορφή των εργαλείων που θα υποστηρίζουν τη διαδικασία ανάπτυξης λογισμικού για την υπηρεσιοποιημένη βιομηχανία, πόσο μάλλον εργαλείων που να υποστηρίζουν συγκεκριμένες, εξειδικευμένες μεθοδολογίες.
- Λόγω των παραπάνω ελλείψεων, δεν έχει υπάρξει, μέχρι στιγμής πρακτική αξιολόγηση καινοτομικών μεθόδων ή / και εργαλείων ανάπτυξης λογισμικού σε πραγματικές (ή έστω πειραματικές) συνθήκες βιομηχανικής υπηρεσιοποίησης.

Η παρούσα διατριβή επιχειρεί να θέσει τα θεμέλια για τη συμπλήρωση αυτού του ερευνητικού, μεθοδολογικού και τεχνολογικού κενού. Συγκεκριμένα, οι ερευνητικές συμβολές της διατριβής είναι οι παρακάτω:

- Μετά από επισκόπηση του ερευνητικού πεδίου της υπηρεσιοποίησης και των Συστημάτων Προϊόντος – Υπηρεσίας διατυπώνονται έξι βασικές απαιτήσεις για συνδυασμούς μεθοδολογιών και εργαλείων ανάπτυξης λογισμικού για την ενσωμάτωση ψηφιακών υπηρεσιών σε βιομηχανικά προϊόντα. Με βάση την παραπάνω μελέτη διαπιστώνεται, επίσης, η απουσία μεθοδολογιών, συνδυασμένων με εργαλεία, που τις ικανοποιούν πλήρως. Στη συνέχεια, περιγράφεται εξειδικευμένη μεθοδολογία ανάπτυξης λογισμικού βασισμένη στη παράλληλη δομική και συμπεριφορική μοντελοποίηση του συστήματος υπηρεσιών και των υποδομών ΤΠΕ. Η χρήση δομικών και συμπεριφορικών μοντέλων επιτρέπει την πληρέστερη αποτύπωση των συσχετίσεων μεταξύ των οντοτήτων του συστήματος υπηρεσιών, της συμπεριφοράς και των αλληλεπιδράσεών τους. Η μεθοδολογία οδηγεί από αφαιρετικά μοντέλα των ψηφιακών υπηρεσιών, σε λεπτομερή σχεδιασμό του λογισμικού πάνω στις τεχνολογίες – στόχους του συστήματος, και τελικά σε κώδικα και εκτελέσιμα αρχεία.
- Περιγράφεται η αρχιτεκτονική και η λειτουργία μίας καινοτομικής Πλατφόρμας Ανάπτυξης Λογισμικού, η οποία υποστηρίζει και υλοποιεί την προτεινόμενη μεθοδολογία ανάπτυξης. Η Πλατφόρμα Ανάπτυξης είναι ένα πλήρες περιβάλλον ανάπτυξης λογισμικού με μοντέλα, με δυνατότητες συνεργατικής ανάπτυξης, και διασυνδεδεμένο με το σύστημα εκτέλεσης των ψηφιακών υπηρεσιών.
- Αξιολογείται η συμβολή της προτεινόμενης μεθοδολογίας και του εργαλείου σε πραγματικές συνθήκες ενσωμάτωσης ψηφιακών υπηρεσιών στη βιομηχανία με την ακόλουθη μεθοδολογία:
 - Δημιουργήθηκε πρωτότυπο Ολοκληρωμένο Περιβάλλον Ανάπτυξης Λογισμικού, βασισμένο στην προτεινόμενη αρχιτεκτονική.
 - Μέσω του εργαλείου, έγινε η εφαρμογή της μεθοδολογίας σε τρεις περιπτώσεις πιλοτικής βιομηχανικής υπηρεσιοποίησης, για την ανάπτυξη λογισμικού υπηρεσιών για Συστήματα Προϊόντος – Υπηρεσίας.
 - Τέλος, αποτυπώθηκε και αναλύθηκε η αξιολόγηση της νέας μεθόδου και εργαλείου από τους τελικούς χρήστες.

Η συμβολή της διατριβής παρουσιάζεται συνοπτικά στο παρακάτω σχήμα:



Σχήμα 1: Η συμβολή της διδακτορικής διατριβής

1.3 Η δομή της διατριβής

Η παρούσα Διατριβή αποτελείται από τα εξής κεφάλαια:

- Το Κεφάλαιο 1 αποτελεί την παρούσα εισαγωγή στο αντικείμενο και τους στόχους της διατριβής
- Το Κεφάλαιο 2 αναλύει το γενικότερο πλαίσιο της υπηρεσιοποίησης στη βιομηχανία.
- Το Κεφάλαιο 3 παραθέτει το θεωρητικό και τεχνικό υπόβαθρο στους τομείς των μεθοδολογιών ανάπτυξης Συστημάτων Προϊόντος – Υπηρεσίας και των σχετικών μεθοδολογιών και εργαλείων.
- Το Κεφάλαιο 4 αναλύει την προτεινόμενη μεθοδολογία και τα εργαλεία ανάπτυξης λογισμικού.
- Το Κεφάλαιο 5 περιγράφει την αξιολόγηση της μεθοδολογίας και του πρωτότυπου περιβάλλοντος ανάπτυξης λογισμικού, στα πλαίσια του ερευνητικού έργου MSEE (Manufacturing Service Ecosystem)
- Το Κεφάλαιο 6 αναπτύσσει τα συμπεράσματα της παρούσας έρευνας, και προτείνει πιθανές μελλοντικές κατευθύνσεις.

2 Βιομηχανία και Υπηρεσιοποίηση

Στο σύγχρονο παγκοσμιοποιημένο οικονομικό περιβάλλον, οι ανταγωνιστικές πιέσεις και η πρόοδος της τεχνολογίας μεταμορφώνουν το τοπίο της βιομηχανικής παραγωγής. Τα τελευταία χρόνια υπάρχει έντονο ενδιαφέρον, μεταξύ άλλων, για αυξημένη αυτοματοποίηση, νέες μορφές διοίκησης και οργάνωσης της παραγωγής, ευέλικτες εφοδιαστικές αλυσίδες και υψηλή προσαρμοστικότητα στις απαιτήσεις των τελικών χρηστών

Μία από αυτές τις προσεγγίσεις είναι «υπηρεσιοποίηση» (servitisation) των προσφερόμενων αγαθών των βιομηχανικών επιχειρήσεων. Αυτό επιτυγχάνεται με την ενσωμάτωση υπηρεσιών στα αγαθά αυτά. Οι νέες υπηρεσίες μπορεί απλώς να υποστηρίζουν το υλικό αγαθό, να προσφέρουν νέες δυνατότητες, ή ακόμα να αλλάζουν τον τρόπο διάθεσής του (π.χ. με τη μορφή μίσθωσης αντί πώλησης). Η μετάβαση αυτή συσχετίζεται με αλλαγές στον τρόπο λειτουργίας της επιχείρησης, με τον μετασχηματισμό και ισχυροποίηση των σχέσεων μεταξύ των συμμετεχόντων στις εφοδιαστικές αλυσίδες, και την αναζήτηση συστηματικών εργαλείων και μεθοδολογιών για την καινοτομία στις υπηρεσίες.

Η πρακτική αυτή υποστηρίζεται μέσω εργαλείων, εφαρμογών και υποδομών Τεχνολογίας Πληροφορικής και Επικοινωνιών, οι οποίες είναι πλέον απαραίτητες για την παροχή προηγμένων υπηρεσιών στους τελικούς πελάτες της βιομηχανίας (καταναλωτικούς ή βιομηχανικούς). Το ερευνητικό έργο που περιγράφεται στη παρούσα διατριβή εστιάζει στη μελέτη και σχεδιασμό των κατάλληλων εργαλείων για την ανάπτυξη των υποδομών λογισμικού για βιομηχανικές υπηρεσίες και Συστήματα Προϊόντος – Υπηρεσίας.

2.1 Η έννοια της υπηρεσίας και του συστήματος υπηρεσιών

Στην οικονομική θεωρία, ένα «αγαθό» είναι οτιδήποτε ικανοποιεί ανάγκες και παρέχει ωφέλεια στους αποδέκτες του (Sullivan & Sheffrin, 2003). Τα αγαθά συχνά ταξινομούνται συχνά σε σχέση με τον βαθμό «υλικότητάς» τους. Στο ένα άκρο, τα αμιγώς υλικά αγαθά μπορούν να περιλαμβάνουν πρώτες ύλες, εμπορεύματα, μηχανήματα κ.τ.λ. Στο άλλο άκρο βρίσκονται αμιγώς άυλα αγαθά, όπως π.χ. η πληροφορία, προϊόντα πνευματικής ιδιοκτησίας και, κρισιμότερα, οι υπηρεσίες.

Οι υπηρεσίες πάντα αποτελούσαν ανταλλασσόμενο είδος και φορέα αξίας στην οικονομία. Ωστόσο, τα τελευταία χρόνια παρατηρείται ότι η σημασία των υπηρεσιών στην οικονομία γίνεται όλο και μεγαλύτερη. Η συμμετοχή των υπηρεσιών στη παγκόσμια οικονομική δραστηριότητα αυξάνεται με δραματικό ρυθμό τα τελευταία χρόνια. Από το 1995 μέχρι το 2014, το ποσοστό συμμετοχής των υπηρεσιών στο παγκόσμιο ΑΕΠ αυξήθηκε από 61% σε 70% (World Bank, 2015). Σύμφωνα με την αναφορά του Παγκόσμιου Οργανισμού Εργασίας για την παγκόσμια απασχόληση και τις κοινωνικές προοπτικές για το 2015 («World Employment And Social Outlook: Trends 2015»), το μεγαλύτερο μέρος των νέων θέσεων εργασίας, σε παγκόσμιο επίπεδο, θα βρίσκεται στον τομέα των υπηρεσιών του ιδιωτικού τομέα, αλλά και στους δημόσιους τομείς της υγείας, εκπαίδευσης και δημόσιας διοίκησης. Αντίθετα, το μέρος της συνολικής απασχόλησης στον βιομηχανικό τομέα θα παραμείνει σταθερό σε παγκόσμιο επίπεδο. (International Labour Organization, 2015)

Οι υπηρεσίες είναι άυλα οικονομικά αγαθά, τα οποία, χωρίς να έχουν φυσική υπόσταση (αλλά με την χρήση ενίοτε υλικών μέσων και παγίων), παρέχουν ωφέλειες στους αποδέκτες

τους. Οι Sampson και Froehle επισημαίνουν ότι στη παραδοσιακή βιβλιογραφία δεν τυγχάνουν πάντα ικανοποιητικού ορισμού. Ορισμένες προσεγγίσεις ορίζουν ως υπηρεσίες κάθε οικονομική δραστηριότητα/ αγαθό που δεν εμπίπτουν στους παραδοσιακούς πρωτογενείς και δευτερογενείς τομείς (π.χ. βιομηχανία, κατασκευές, γεωργία κτλ.). Άλλες προσπαθούν να βασιστούν σε ειδικά χαρακτηριστικά, όπως π.χ. η άυλη φύση των υπηρεσιών, η ανυπαρξία αποθεμάτων, η μη-μεταφορά ιδιοκτησίας κτλ. Άλλοι ορισμοί εστιάζουν στον τρόπο παραγωγής, π.χ. ότι οι υπηρεσίες παράγονται από ανθρώπους (ωστόσο υπάρχουν μηχανικά παρεχόμενες και αυτοματοποιημένες υπηρεσίες), ότι αφορούν διαδικασίες που προκαλούν αλλαγές στους παραλήπτες ή στους πόρους τους κτλ.

Με τη σειρά τους, οι Sampson και Froehle υποστηρίζουν ότι «στις διαδικασίες των υπηρεσιών, ο πελάτης προσφέρει σημαντικές εισόδους στη διαδικασία παραγωγής», και ότι το παραπάνω αποτελεί αναγκαία και ικανή συνθήκη για τον ορισμό μίας διαδικασίας ως υπηρεσία. Κατά πρώτον, το έναυσμα για την παροχή της υπηρεσίας δίδεται από τον πελάτη και η υπηρεσία δεν μπορεί να παρασχεθεί χωρίς τη συμμετοχή του. Επίσης, ένας πελάτης μπορεί να φέρει απτά υλικά, αντικείμενα και πόρους, τα οποία συμμετέχουν και μετασχηματίζονται κατά την παροχή της υπηρεσίας (π.χ. το κεφάλαιο βάση του οποίου παρέχονται οι τραπεζικές υπηρεσίες, το κτίσμα για το οποίο παραλαμβάνει υπηρεσίες κατασκευής κ.α.). Τέλος, ο πελάτης παρέχει πληροφορίες απαραίτητες για την παροχή της υπηρεσίας π.χ. δεδομένα περιουσιακής κατάστασης για την παροχή υπηρεσιών φοροτεχνικού. Οι ερευνητές παρατηρούν επίσης ότι η έμμεση συμμετοχή των πελατών σε δραστηριότητες π.χ. με την παροχή απόψεων ή απαιτήσεων για ένα βιομηχανικό προϊόν (το οποίο μπορεί να μην καταναλώσουν) αλλά και η πράξη της επιλογής ενός αγαθού για κατανάλωση δεν αποτελούν επαρκείς συνθήκες για τον ορισμό μίας δραστηριότητας ως υπηρεσία. Αντίθετα, η πράξη της «πώλησης» μπορεί να θεωρηθεί υπηρεσία, καθώς ικανοποιεί όλες τις παραπάνω προϋποθέσεις: ενεργοποιείται με την παρουσία και συμμετοχή του πελάτη, παρέχει πληροφόρηση για τον τρόπο παροχής της υπηρεσίας (το τι θέλει να αγοράσει κτλ.) και μπορεί να φέρει και άλλα ιδιόκτητα στοιχεία (π.χ. το μηχάνημα για το οποίο θέλει να αγοράσει υπηρεσίες επισκευών) (Sampson & Froehle, 2006).

Στο παραπάνω πνεύμα, οι Vargo και Lusch ορίζουν την υπηρεσία ως την εφαρμογή ειδικευμένων προσόντων (γνώσεων και δεξιοτήτων) μέσω πράξεων και διαδικασιών προς όφελος μίας άλλης οντότητας, ή της οντότητας που παρέχει την υπηρεσία (Vargo & Lusch, 2004). Από την άλλη, οι Dumas κ.α. διατυπώνουν την άποψη ότι ένα στιγμιότυπο μίας υπηρεσίας (service instance) είναι η «υπόσχεση» από ένα μέρος (πάροχος) της επιτέλεσης μίας λειτουργίας για λογαριασμό ενός άλλου (παραλήπτης), σε συγκεκριμένο τόπο και χρόνο, και διαμέσου ορισμένου καναλιού, με την υλοποίηση της υπόσχεσης να θεωρείται η «παροχή» της υπηρεσίας (service delivery) (Dumas, O'Sullivan, Heravizadeh, David, & ter Hofstede).

Εκτείνοντας τον παραπάνω συλλογισμό, οι Ferrario και Guarino θεωρούν ότι :

«Μία υπηρεσία είναι παρούσα σε κάποιον χρόνο t , και θέση l , αν και μόνο αν, ένας παράγοντας (actor) έχει δεσμευθεί ρητώς να εγγυηθεί την εκτέλεση κάποιου τύπου δράσης στη θέση l , όταν συμβεί κάποιο ενεργοποιό συμβάν, για το συμφέρον κάποιου άλλου παράγοντα κατόπιν πρότερης συμφωνίας, και με συγκεκριμένο τρόπο»

Με τον τρόπο αυτό, οι ερευνητές διαχωρίζουν τη «δέσμευση υπηρεσίας» από το «περιεχόμενο» της υπηρεσίας (δηλαδή τις ενέργειες για την παροχή της) και τη «διαδικασία» της υπηρεσίας (δηλαδή τις επιχειρησιακές διαδικασίες/δομές που αφορούν την εκτέλεση των απαραίτητων ενεργειών). Επιπλέον, έτσι διαχωρίζεται η «διαθεσιμότητα» της υπηρεσίας από την δέσμευση για την παροχή της, π.χ. όπως στη περίπτωση που ενώ υπάρχει μία σύμβαση υπηρεσιών, αυτή δεν υλοποιείται. (Ferrario & Guarino, 2009).

Παρά τη δυσκολία να οριστεί επακριβώς η έννοια της «υπηρεσίας», εξετάζοντας τους παραπάνω ορισμούς, είναι δυνατόν να εντοπιστούν τα ιδιαίτερα χαρακτηριστικά που τη διακρίνουν από άλλα αγαθά:

- **Είναι άυλα αγαθά:** οι υπηρεσίες δεν έχουν φυσική υπόσταση, αν και μπορούν να περιλαμβάνουν φυσικά πάγια ή υλικά κατά την παροχή τους. Μία «καθαρή» υπηρεσία δεν περιλαμβάνει καμία μεταφορά ιδιοκτησίας από τον πάροχο στον παραλήπτη. Για παράδειγμα, κατά τη λειτουργία ενός κομμωτηρίου ή ενός δικηγορικού γραφείου δεν υπάρχουν φυσικά αντικείμενα που να αλλάζουν χέρια. Μία υπηρεσία όμως μπορεί να περιλαμβάνει την παράδοση υλικών αγαθών, ή να συνοδεύει φυσικά αγαθά, χωρίς όμως να χάνει την άυλη φύση της. Για παράδειγμα η «υπηρεσία» της αλλαγής λιπαντικών ενός οχήματος περιλαμβάνει την αγορά του λιπαντικού, το οποίο περνά στην ιδιοκτησία του πελάτη, αλλά το επιθυμητό αποτέλεσμα της υπηρεσίας δεν είναι απλώς η κατοχή του υλικού, αλλά η αντικατάσταση του παλαιού λιπαντικού με το νέο. Επιπλέον, η παροχή δωρεάν επισκευής του οχήματος εντός της περιόδου εγγύησης μετά την αγορά συνδέεται με το υλικό πάγιο (το όχημα), αλλά δεν υφίσταται εξαιτίας του οχήματος – αλλά από τις εργασίες που ενδεχομένως να γίνουν σε αυτό. Σε άλλες περιπτώσεις, η υπηρεσία μπορεί να περιλαμβάνει την παροχή υλικών ή άυλων αγαθών – η διαφορά από την απλή εμπορία αγαθών είναι ότι τα αγαθά αυτά εντάσσονται σε ένα πλαίσιο υπηρεσίας. Κάτι τέτοιο σημαίνει ότι πέρα από τα ίδια τα αγαθά παρέχεται και επιπλέον αξία με τη μορφή υπηρεσίας π.χ. με την αποκλειστική διάθεση των προϊόντων αυτών στον πελάτη, ή με τη δημιουργία πλαισίου συνεργασίας για την παραγωγή τους (π.χ. σε ένα συμβουλευτικό μελετητικό έργο).
- **Δεν αποθηκεύονται και καταναλώνονται τη στιγμή της παραγωγής τους:** καθώς οι υπηρεσίες είναι άυλες, δεν μπορούν να παραχθούν και στη συνέχεια να αποθηκευτούν για μελλοντική χρήση. Ακόμα και αν ένας πάροχος υπηρεσιών προετοιμαστεί για την παροχή της υπηρεσίας π.χ. με την αγορά παγίων, την εκπαίδευση του προσωπικού κτλ., οι υπηρεσίες που δεν παράγονται πριν την επαφή με τον πελάτη – ο οποίος τις καταναλώνει τη στιγμή της παραγωγής τους. Έτσι, μία υπηρεσία μπορεί να καταναλωθεί αποκλειστικά και μόνο σε συγκεκριμένο χρόνο και κάτω από τις συνθήκες που επιτρέπουν την κατανάλωσή της π.χ. όταν ο πελάτης αιτηθεί την υπηρεσία από τον πάροχο. Επιπλέον, μόλις η υπηρεσία παρασχεθεί στον παραλήπτη της, η υπηρεσία «χάνεται» ολοκληρωτικά, καθώς έχει πλέον «καταναλωθεί» από αυτόν.
- **Είναι αναπόσπαστες από τους παρόχους και τους παραλήπτες τους:** ένα υλικό αγαθό μπορεί να παραχθεί και στη συνέχεια να περνά από έναν αριθμό ενδιάμεσων εμπλεκόμενων μέχρι να καταλήξει στον καταναλωτή του. Το υλικό αγαθό, άπαξ και

παραχθεί, μπορεί να διαχωριστεί από τον παραγωγό του – ο οποίος δεν χρειάζεται να έρθει απαραίτητα σε επαφή με τον καταναλωτή. Αντιθέτως, μία υπηρεσία δεν μπορεί να διαχωριστεί από τον πάροχο της, ο οποίος μόνο αυτός μπορεί να την δώσει στον πελάτη. Ομοίως, ο πελάτης δεν μπορεί να παραλάβει την υπηρεσία χωρίς να έρθει σε επαφή με τον πάροχο της, σε όλα τα στάδια της παροχής της υπηρεσίας – από το αρχικό αίτημα για την παροχή της, κατά την παροχή αλλά και κατά την αποπληρωμή. Κατά συνέπεια, η παροχή μίας υπηρεσίας δημιουργεί μία σχέση ανάμεσα στον παραγωγό της υπηρεσίας και τον τελικό παραλήπτη.

- **Χαρακτηρίζονται από μοναδικότητα και ετερογένεια:** Οι υπηρεσίες, σε σχέση με τα υλικά αγαθά, χαρακτηρίζονται από μοναδικότητα ανάλογα με τις συνθήκες παροχής τους, τους παρόχους και τις ανάγκες των τελικών παραληπτών. Αντιθέτως, τα υλικά αγαθά τείνουν να έχουν μεγαλύτερη τυποποίηση, και να παραμένουν αμετάβλητα από την παραγωγή μέχρι την κατανάλωσή τους. Η διάκριση αυτή δεν είναι απόλυτη, καθώς υπάρχουν υλικά αγαθά τα οποία παράγονται με διαφορετικό τρόπο κάθε φορά ανάλογα με τις απαιτήσεις του πελάτη και τις συνθήκες παραγωγής τους, και υπηρεσίες οι οποίες παρέχονται πάντα με τον ίδιο τρόπο. Προφανώς, υπάρχουν εύλογα οικονομικά κίνητρα για την τυποποίηση του τρόπου παροχής υπηρεσιών. Ωστόσο, η πράξη της παροχής μίας υπηρεσίας αφορά μία μοναδική στιγμή (ή χρονική περίοδο), και μπορεί να γίνεται υπό διαφορετικές συνθήκες, με άλλους παρόχους και παραλήπτες κάθε φορά, άρα και το περιεχόμενό της μπορεί να διαφέρει (πολύ ή λίγο) κάθε φορά που παρέχεται.
- **Οι πελάτες συμμετέχουν στη παραγωγή της υπηρεσίας:** Οι πελάτες ενδέχεται να παρέχουν εισόδους, όπως πληροφόρηση, τα αντικείμενα, υλικά και πάγια για τα οποία παρέχεται η υπηρεσία. (Sampson & Froehle, 2006)

Με βάση τα παραπάνω, μπορούν να διατυπωθούν επιπλέον χαρακτηριστικά της υπηρεσίας σαν οικονομικό αγαθό:

- Κατά κανόνα, οι υπηρεσίες δεν μπορούν να προετοιμαστούν εκ των προτέρων - πρέπει να παραχθούν μόνο τη στιγμή που απαιτηθούν.
- Οι υπηρεσίες δεν μεταβιβάζονται και δεν μπορούν να επιστραφούν εφόσον καταναλωθούν.
- Σε αντίθεση με τη μαζική παραγωγή υλικών αγαθών, η μαζική παραγωγή υπηρεσιών είναι περισσότερο δύσκολη, καθώς η παροχή τους συχνά εξαρτάται από τον ανθρώπινο παράγοντα και τις μοναδικές συνθήκες κάτω από τις οποίες παρέχονται - για τον λόγο αυτό, η διασφάλιση ποιότητας υπηρεσιών παρουσιάζει ιδιαίτερες προκλήσεις.

Γίνεται εμφανές ότι ενώ υπάρχουν χαρακτηριστικά των υπηρεσιών, αυτά δεν ορίζουν μονοσήμαντα την έννοια της υπηρεσίας – μπορούν να είναι παρόντα ή απόντα σε μία δεδομένη περίπτωση.

Αναλύοντας περαιτέρω τα χαρακτηριστικά της έννοιας της υπηρεσίας, οι Constantin και Lusch διαχωρίζουν τους πόρους (πάγια, υλικά, δεξιότητες κτλ.), μίας οικονομικής οντότητας σε «τελεστέους πόρους» (operand resources) και «τελεστές πόρους» (operant resources). Οι πρώτοι αφορούν πόρους πάνω στους οποίους γίνονται ενέργειες για την παραγωγή αξίας,

π.χ. πρώτες ύλες, γη, κεφάλαιο κτλ. Αντιθέτως, οι δεύτεροι είναι πόροι μέσω των οποίων παράγονται αποτελέσματα απευθείας, όπως διαδικασίες, γνώσεις, δεξιότητες, οργανωτικές δομές κ.α. (Constantin & Lusch, 1994).

Με βάση τους παραπάνω ορισμούς, οι Vargo και Lusch αποσαφηνίζουν τις ιδιαιτερότητες των υπηρεσιών έναντι των υλικών αγαθών μέσω της περιγραφής δύο διαφορετικών προσεγγίσεων της επιχείρησης για την παραγωγή αξίας και του μάρκετινγκ: της παραδοσιακής «λογικής υλικών αγαθών» και της αναδυόμενης «λογικής υπηρεσιών» (Vargo & Lusch, 2004):

	Παραδοσιακή λογική υλικών αγαθών	Αναδυόμενη λογική υπηρεσιών
Ανταλλασσόμενα αγαθά	Υλικά αγαθά	Εφαρμογή γνώσεων και δεξιοτήτων
Ρόλος των αγαθών	Τα αγαθά είναι «τελεστέοι πόροι» και τελικά προϊόντα.	Τα αγαθά είναι φορείς «τελεστών πόρων», δηλαδή ενδιάμεσα στάδια στη παραγωγή αξίας ή ωφέλειας στον πελάτη
Ρόλος του πελάτη	Ο πελάτης είναι παραλήπτης των αγαθών	Ο πελάτης είναι συμπαραγωγός της υπηρεσίας, καθώς η υπηρεσία δημιουργείται με τη συμμετοχή του πελάτη
Προσδιορισμός της έννοιας της «αξίας»	Η αξία προσδιορίζεται από τον παραγωγό, είναι ενσωματωμένη στο αγαθό και ορίζεται ως «ανταλλακτική» αξία.	Η αξία γίνεται αντιληπτή και προσδιορίζεται από τον πελάτη με βάση την «αξία από τη χρήση». Συγκεκριμένα, η αξία προκύπτει από την επωφελή εφαρμογή των «τελεστών πόρων», οι οποίοι μεταδίδονται ενίοτε και μέσω «τελεστών πόρων» (π.χ. η άυλη υπηρεσία της πληροφόρησης παρέχεται μέσω του υλικού εντύπου). Η επιχείρηση μπορεί να κάνει μόνο «προτάσεις αξίας» στον πελάτη (value propositions).
Αλληλεπίδραση επιχείρησης – πελάτη	Ο πελάτης αντιμετωπίζεται από την επιχείρηση σαν «τελεστής πόρος»: η επιχείρηση ενεργεί στον πελάτη (μέσω ενεργειών μάρκετινγκ) για την παραγωγή αξίας, δηλαδή πωλήσεων και συναλλαγών	Ο πελάτης είναι «τελεστής πόρος»: Οι πελάτες είναι ενεργοί συμμετέχοντες στη παραγωγή αξίας (ως συμπαραγωγοί αξίας).
Πηγή οικονομικής ανάπτυξης	Ο πλούτος παράγεται από απτά αγαθά και πόρους. Ο πλούτος είναι η ιδιοκτησία, έλεγχος και παραγωγή «τελεστών πόρων»	Ο πλούτος παράγεται από την εφαρμογή και ανταλλαγή εξειδικευμένων γνώσεων και ικανοτήτων. Αντιπροσωπεύει το δικαίωμα στη χρήση «τελεστών πόρων».

Πίνακας 1: Λογική υλικών αγαθών και λογική υπηρεσιών

Με βάση την παραπάνω διάκριση, οι Vargo και Lusch υποστηρίζουν ότι τα βασικά ανταλλασσόμενα σε μία οικονομία είναι υπηρεσίες οι οποίες αποσκοπούν στην παραγωγή ωφέλειας για τους εμπλεκόμενους, και ότι τα υλικά (απτά) αγαθά αποτελούν απλώς ένα μέσο για την παροχή της υπηρεσίας, τόσο σαν εργαλείο για την υλοποίησή της ή σαν φορέα της υπηρεσίας.

Με βάση τα παραπάνω, διαπιστώνεται ότι η παραγωγή των υπηρεσιών γίνεται με τη συμμετοχή πολλαπλών εμπλεκόμενων πλευρών, δεξιοτήτων, πόρων και διαδικασιών, δηλαδή ετερογενή στοιχεία που απαρτίζουν ένα «Σύστημα Υπηρεσίας».

Σύμφωνα με τους Spohrer κ.α. ένα «Σύστημα Υπηρεσίας» είναι μία δυναμική διαμόρφωση (configuration) πόρων για τη συμπαραγωγή αξίας, η οποία περιλαμβάνει πόρους, όπως άνθρωπος, οργανισμούς, κοινή πληροφορία (γλώσσα, κανόνες, μέτρα, μεθόδους) και τεχνολογία. Ένα σύστημα υπηρεσιών μπορεί να βελτιώσει την κατάσταση του ιδίου ή ενός άλλου συστήματος μέσω κοινής χρήσης ή εφαρμογής των πόρων του.

Κάθε σύστημα υπηρεσίας έχει μοναδική ταυτότητα και αποτελεί «στιγμιότυπο» συγκεκριμένης «κλάσης» συστήματος υπηρεσίας, (π.χ. άνθρωποι, επιχειρήσεις, συνεργασίες, δημόσιες υπηρεσίες κτλ.). Τα συστήματα υπηρεσιών είναι συνδεδεμένα εσωτερικά (δηλαδή στα συστατικά τους μέρη) και εξωτερικά (με εξωτερικές οντότητες) με άλλα συστήματα υπηρεσιών μέσω προτάσεων αξίας. Έτσι, «ατομικά», μοναδιαία, συστήματα υπηρεσιών μπορούν να παράγουν «σύνθετα» συστήματα υπηρεσιών.

Οι αλληλεπιδράσεις μεταξύ συστημάτων υπηρεσιών περιλαμβάνουν τη δημιουργία «προτάσεων» για τη «συμπαραγωγή αξίας» (τη παραγωγή και παροχή μίας υπηρεσίας), τη συμφωνία στη πρόταση (τυπική και κωδικοποιημένη ή άτυπη) και την πραγματοποίηση της συμφωνίας. Στα πλαίσια αυτά προτείνονται και δύο ειδικές περιπτώσεις προτάσεων: η «συν-δημιουργία» ενός νέου στιγμιότυπου συστήματος υπηρεσίας (λ.χ. μίας νέας επιχείρησης ή μίας σύμπραξης δημοσίου – ιδιωτικού τομέα) και μίας νέας κλάσης συστήματος υπηρεσίας (λ.χ. μίας νέας μορφής σχέσης απασχόλησης). Στα πλαίσια αυτά, τα συστήματα υπηρεσίας έχουν ένα κύκλο ζωής που περιλαμβάνει την αρχή, τη λειτουργία, και το τέλος τους. (Maglio, Vargo, Caswell, & Spohrer, 2009).

Μέσα σε ένα σύστημα υπηρεσιών, οι εμπλεκόμενοι συμμετέχουν σε ανταλλαγές δεξιοτήτων, όπου μοιράζονται τέσσερα είδη πόρων: πληροφορία, ανθρωποπροσπάθεια, κίνδυνο, και υλικά. Τα διάφορα συστήματα υπηρεσιών μπορούν να ταξινομηθούν σε αδρές γραμμές με βάση τον τύπο ανταλλαγών που επικρατούν. Για παράδειγμα, το μοίρασμα πληροφορίας είναι κοινό σε συμβουλευτικά έργα, ενώ το μοίρασμα του κινδύνου εμφανίζεται στη παροχή ασφαλιστικών υπηρεσιών. Ωστόσο, τα τέσσερα αυτά είδη εμφανίζονται, σε κάποιο βαθμό, σε κάθε σχέση παροχής υπηρεσιών, και προσδιορίζονται από την κατανομή των δεξιοτήτων / πόρων που εμπλέκονται και από τον τρόπο που παράγεται προστιθέμενη αξία για κάθε ένα από τους εμπλεκόμενους (Maglio & Spohrer, 2008).

Σύμφωνα με τους Spohrer και Kwan, μία «κοσμοθεωρία συστημάτων υπηρεσιών» βασίζεται σε τρεις θεμελιώσεις αρχές: κατά πρώτον, αντιλαμβάνεται την οικονομική και επιχειρηματική δραστηριότητα σαν ένα σύστημα από αλληλοσυνδεόμενα συστήματα υπηρεσιών, τα οποία μπορούν με τη σειρά τους να είναι σύνθετα ή μοναδιαία. Κατά

δεύτερον, τα συστήματα υπηρεσιών συνδέονται με «προτάσεις προστιθέμενης αξίας» ανάμεσα στα μέρη τους, δηλαδή τα πλαίσια της αξίας που συν-δημιουργείται και για τα δύο μέρη.

Τέλος, τα συστήματα υπηρεσιών έχουν μηχανισμούς διακυβέρνησης (governance mechanisms), με τους οποίους είναι δυνατή η παρακολούθηση της απόδοσης, η επίλυση διαφορών, η ανάληψη κινδύνου κτλ. (Spohrer & Kwan, 2008)

2.2 Υπηρεσίες και βιομηχανία

Παραδοσιακά, μια βιομηχανική εταιρεία είναι αυτή που «κατασκευάζει» και παρέχει φυσικά προϊόντα στους πελάτες της. Μαζί με τα προϊόντα αυτά, η επιχείρηση μπορεί επίσης να παρέχει ένα περιορισμένο σύνολο υποστηρικτικών υπηρεσιών, όπως, για παράδειγμα, εργασίες παράδοσης, εγκατάστασης ή συντήρησης.

Η ενσωμάτωση υπηρεσιών σε βιομηχανικά προϊόντα είναι η αυξανόμενη έμφαση των βιομηχανικών επιχειρήσεων στην παροχή φυσικών προϊόντων και συσχετισμένων υπηρεσιών αντί φυσικών προϊόντων μόνο, μέχρι του σημείου να προσφέρει ένα ενιαίο «Σύστημα Προϊόντος - Υπηρεσίας» (ΣΠΥ, Product Service System - PSS) αντί των υλικών αγαθών. (Neely, 2008). Η προσθήκη αξίας στα υλικά αγαθά με την παράλληλη παροχή υπηρεσιών απαιτεί σταδιακές αλλαγές στον τρόπο λειτουργίας της εταιρείας – και μεγάλο μέρος της μέχρι τώρα σχετικής έρευνας έχει επικεντρωθεί σε αυτό το πεδίο. Κατά τα τελευταία τριάντα χρόνια, το φαινόμενο αυτό έχει ενταθεί, και έχει περιγραφεί ως «υπηρεσιοποίηση» της βιομηχανίας (“servitisation of manufacturing”) (Vandermerwe & Rada, 1988).

Σε γενικές γραμμές, η βιβλιογραφία σχετικά με την ενσωμάτωση υπηρεσιών είναι σχετικά νέα, και εξελίσσεται μαζί με την κατανόηση του φαινομένου. Αρχικά, η ενσωμάτωση υπηρεσιών αντιμετωπιζόταν περισσότερο σαν δραστηριότητα «προστιθέμενης αξίας», όπου οι υπηρεσίες συμπλήρωναν υφιστάμενα φυσικά προϊόντα (Vandermerwe & Rada, 1988). Παράλληλα, υπήρχε έμφαση στις σταδιακές αλλαγές στις λειτουργίες και την κουλτούρα ενός οργανισμού για τη στροφή προς την παροχή υπηρεσιών. Η πλέον πρόσφατη βιβλιογραφία τείνει να υιοθετήσει στοιχεία της «λογικής υπηρεσιών» (Vargo & Lusch, 2007) και εστιάζει στην υπηρεσία με την έννοια της «εξυπηρέτησης» χωρίς τον διαχωρισμό σε αγαθά και υπηρεσίες. Στη θεώρηση αυτή, αποτελεί μία δυναμική δραστηριότητα, στην οποία η προστιθέμενη αξία προκύπτει (συν-δημιουργείται) από την αλληλεπίδραση του πελάτη, του παρόχου και των προμηθευτών.

2.3 Ενσωμάτωση υπηρεσιών και συστήματα προϊόντος - υπηρεσίας

Οι βιομηχανικές επιχειρήσεις έχουν διαφορετικά επίπεδα ενσωμάτωσης υπηρεσιών στα παραδοσιακά βιομηχανικά προϊόντα. Ο βαθμός ενσωμάτωσης περιγράφεται καλύτερα ως ένα φάσμα, που περιλαμβάνει, στο ένα άκρο, την παροχή αμιγώς υλικών προϊόντων, και στο άλλο του άκρο, την παροχή αμιγώς άυλων υπηρεσιών. (Oliva & Kallenberg, 2003).

Σύμφωνα με τους Vandermerwe και Rada, η ενσωμάτωση υπηρεσιών είναι η τάση κατά την οποία οι εταιρείες συνειδητά αναπτύσσουν προσφερόμενες υπηρεσίες που υποστηρίζουν τα προϊόντα τους ώστε να αποκτήσουν ανταγωνιστικό πλεονέκτημα. Με την προσθήκη των υπηρεσιών στα υφιστάμενα κύρια προϊόντα τους, οι επιχειρήσεις διαφοροποιούνται από τον

ανταγωνισμό, αυξάνουν την εξάρτηση των πελατών τους και δημιουργούν εμπόδια στον ανταγωνισμό. Ο συγκεκριμένος ορισμός παρουσιάζεται περισσότερο στα πλαίσια της «προσθήκης αξίας», όπου η ενσωμάτωση υπηρεσιών αφορά την προσθήκη υπηρεσιών σε προϊόντα με σκοπό τη δημιουργία νέων ροών εσόδων – οι οποίες θα αντιστοιχούν στις νέες υπηρεσίες. Ειδικά σε αδιαφοροποίητα προϊόντα, η προσθήκη συνοδευτικών υπηρεσιών μπορεί να αποτελέσει άξονα διαφοροποίησης και ανταγωνισμού πέρα από την τιμή (Vandermerwe & Rada, 1988; Tukker A. , 2004).

Ένα Σύστημα Προϊόντος – Υπηρεσίας προσφέρεται από την επιχείρηση σαν ενιαία οντότητα που αποτελείται από υπηρεσίες και υλικά αγαθά, αναπόσπαστα μεταξύ τους. Ο Tukker παρατηρεί ότι ένα ΣΠΥ «αποτελείται από υλικά προϊόντα και άυλες υπηρεσίες, σχεδιασμένα και συνδυασμένα με τέτοιο τρόπο, ώστε, μαζί, να μπορούν να ικανοποιούν συγκεκριμένες ανάγκες του πελάτη». (Tukker A. , 2004). Η εμφάνιση των «Συστημάτων Προϊόντος-Υπηρεσίας» μπορεί να ερμηνευθεί ως απάντηση στην ανάγκη για μία συνεκτική στρατηγική η οποία να περιλαμβάνει ταυτόχρονα τα προϊόντα και τις υπηρεσίες, αλλά και τις υποδομές και τα δίκτυα συνεργασίας που τις υποστηρίζουν. (Cavaliere & Pezzota, 2012).

Οι Müller και Stark Θεωρούν ότι ένα ΣΠΥ αποτελείται από 1) προϊόντα 2) υπηρεσίες, 3) το δίκτυο των εμπλεκόμενων (πάροχοι, προμηθευτές, πελάτες κτλ.), οι οποίοι διασυνδέονται μεταξύ τους με 4) τους μηχανισμούς παροχής του ΣΠΥ (ενοποιημένες διαδικασίες, συν-δημιουργία αξίας κτλ.), οι οποίες υποστηρίζονται από 5) υποδομές (πάγια, εγκαταστάσεις, ΤΠΕ κτλ.), ενώ το ΣΠΥ περιγράφεται από 6) συμβάσεις, δηλαδή δεσμεύσεις που μοιράζουν τον κίνδυνο, προδιαγράφουν τις δραστηριότητες κτλ. (Müller & Stark, 2008).

Εάν εξετασθούν με βάση το προαναφερθέν «φάσμα ενσωμάτωσης υπηρεσιών», τα ΣΠΥ μπορούν να αντιστοιχούν σε διάφορα επίπεδα ενσωμάτωσης. Για παράδειγμα, ένας συνδυασμός ΣΠΥ μπορεί να περιλαμβάνει υπηρεσίες υποστήριξης ενός φυσικού αγαθού (π.χ. συντήρηση ή συμβουλευτικές υπηρεσίες στους τελικούς χρήστες). Ένας άλλος μπορεί να περιλαμβάνει την παροχή του προϊόντος στους τελικούς πελάτες χωρίς τη μεταβίβαση ιδιοκτησίας (π.χ. μέσω μίσθωσης). Σε άλλες περιπτώσεις, ο τελικός καταναλωτής μπορεί να μην έχει καν πρόσβαση στο προϊόν, αλλά να απολαμβάνει τα τελικά αποτελέσματα της χρήσης του με τη μορφή υπηρεσίας (π.χ. με συνδυαστικές «λύσεις» φυσικού προϊόντος – υπηρεσίας) (Tukker A. , 2004).

Τα επιχειρηματικά μοντέλα που βασίζονται στην ενσωμάτωση υπηρεσιών και τα ΣΠΥ μπορούν:

- Να ικανοποιούν τις ανάγκες του πελάτη με ολοκληρωμένο τρόπο και προσαρμοσμένο στις ιδιαίτερες ανάγκες του. Έτσι ο πελάτης μπορεί να συγκεντρωθεί στις κύριες δραστηριότητές του.
- Να δημιουργούν μοναδικές και προσαρμοσμένες σχέσεις με τους πελάτες, με αποτέλεσμα την καλύτερη συγκράτησή τους.
- Ενδεχομένως, να επιτρέπουν γρηγορότερους ρυθμούς καινοτομίας, αφού μπορούν να παρακολουθήσουν καλύτερα τις ανάγκες των πελατών (Tukker A. , 2004).

Οι Hockerts και Weaver (Hockerts & Weaver, 2002) διαχωρίζουν τρεις προσεγγίσεις στα ΣΠΥ:

- Τη προσέγγιση «ολοκλήρωσης» υπηρεσιών, η οποία περιλαμβάνει την παροχή τόσο προϊόντων όσο και υπηρεσιών
- Τη προσέγγιση «προϊόντος», η οποία εστιάζει στη παράδοση προϊόντων και άμεσα συσχετισμένων υπηρεσιών
- Τη προσέγγιση «συστήματος προϊόντος – υπηρεσίας», η οποία εστιάζει στην εισαγωγή νέων υπηρεσιών που ενσωματώνονται στο ίδιο το προϊόν.

Ο Neely προτείνει δύο κατηγορίες, σε σχέση περισσότερο με την προστιθέμενη αξία προς τον καταναλωτή. Η πρώτη αφορά σε συστήματα ΣΠΥ προσανατολισμένα στη «χρήση», όπου ο πάροχος της υπηρεσίας διατηρεί την ιδιοκτησία του φυσικού προϊόντος. Η δεύτερη κατηγορία αφορά ΣΠΥ προσανατολισμένα στο «αποτέλεσμα» όπου το φυσικό προϊόν αντικαθίσταται εξ ολοκλήρου με υπηρεσία. (Neely, 2008)

Εξειδικεύοντας την παραπάνω προσέγγιση, μπορούν να διαχωριστούν τρεις βασικές κατηγορίες ΣΠΥ με βάση τη στόχευση των προσφερόμενων υπηρεσιών, οι οποίες μπορούν να αναλυθούν περαιτέρω σε οκτώ υποκατηγορίες:

- Υπηρεσίες με προσανατολισμό προϊόντος: το επιχειρηματικό μοντέλο είναι ακόμα προσαρμοσμένο στα φυσικά προϊόντα, με κάποιες επιπρόσθετες υπηρεσίες
 - Υπηρεσίες συσχετισμένες με το προϊόν: ο πάροχος πουλά ένα προϊόν και παρέχει υπηρεσίες που απαιτούνται κατά τη χρήση του. Αυτές μπορεί να περιλαμβάνουν, για παράδειγμα, υπηρεσίες συντήρησης, χρηματοδότησης, αναλώσιμα, υπηρεσίες απόσυρσης κτλ.
 - Συμβουλευτικές υπηρεσίες: ο πάροχος προσφέρει συμβουλευτικές υπηρεσίες για την καλύτερη χρήση του προϊόντος. Για παράδειγμα, μπορεί να προτείνει βελτιωμένες εσωτερικές διαδικασίες και δομές, λύσεις για την εφοδιαστική αλυσίδα κτλ.
- Υπηρεσίες με προσανατολισμό χρήσης: τα παραδοσιακά προϊόντα παίζουν κεντρικό ρόλο, αλλά το επιχειρηματικό μοντέλο δεν εστιάζει στη πώλησή τους. Το προϊόν μπορεί να βρίσκεται στη κατοχή του παρόχου και μπορεί να χρησιμοποιείται από πολλούς χρήστες.
 - Αποκλειστική μίσθωση προϊόντος (leasing): Ο πάροχος διατηρεί την κυριότητα του προϊόντος και είναι αρμόδιος για τη συντήρηση και έλεγχο του. Ο ενοικιαστής πληρώνει ένα τακτικό ποσό για τη χρήση του προϊόντος, η οποία είναι, τυπικά, αποκλειστική και απεριόριστη.
 - Μη αποκλειστική μίσθωση προϊόντος: Ο πάροχος διατηρεί η κυριότητα του προϊόντος και είναι αρμόδιος για τη συντήρηση και τον έλεγχο του. Ωστόσο, ο πελάτης δεν έχει αποκλειστική και απεριόριστη χρήση, καθώς υπάρχουν και άλλοι χρήστες που το χρησιμοποιούν με τη σειρά τους.
 - Ταυτόχρονη χρήση (product pooling): Παρόμοιο με τη μη αποκλειστική μίσθωση, αλλά με τη διαφορά ότι η χρήση από πολλούς πελάτες μπορεί να είναι ταυτόχρονη.
- Υπηρεσίες με προσανατολισμό αποτελέσματος: Ο πελάτης και ο πάροχος συμφωνούν πάνω στο τελικό επιθυμητό αποτέλεσμα, χωρίς την εμπλοκή συγκεκριμένου προϊόντος.

- Εξωτερική διαχείριση/ανάθεση: Μέρος της δραστηριότητας του πελάτη ανατίθεται σε κάποιον εξωτερικό ανάδοχο. Η σύμβαση περιλαμβάνει τον καθορισμό κριτηρίων απόδοσης την παροχή της υπηρεσίας, οπότε εντάσσεται στις υπηρεσίες με προσανατολισμό αποτελέσματος. Ωστόσο, δεν αλλάζει ριζικά ο τρόπος με τον οποίο εκτελείται η συγκεκριμένη δραστηριότητα, δεδομένου ότι δεν υπήρχε ισχυρή σύνδεση με κάποιο φυσικό προϊόν (ή πάγιο) για την επίτευξή της.
- Αμοιβή ανά μονάδα υπηρεσίας: ενώ το επιχειρηματικό μοντέλο στηρίζεται σε φυσικό προϊόν, δεν περνά στη κατοχή του πελάτη. Αντιθέτως, αγοράζει το αποτέλεσμα χρήσης του προϊόντος, με βάση τον όγκο χρήσης που κάνει.
- Λειτουργικό αποτέλεσμα: ο πάροχος συμφωνούν με τον πελάτη πάνω στη παράδοση κάποιου αποτελέσματος, ενδεχομένων διατυπωμένο με αφαιρετικούς ή γενικούς όρους χωρίς να γίνεται αναφορά σε συγκεκριμένο φυσικό προϊόν ή τεχνολογικό σύστημα. Σε αυτή την περίπτωση όμως, εμπλέκονται δραστηριότητες που τυπικά συνδέονται με συγκεκριμένα πάγια. Για παράδειγμα, ένα συμβόλαιο διαχείρισης κλιματισμού μπορεί να διατυπώνεται σε όρους αποτελέσματος (υγρασίας, θερμοκρασίας κτλ.) και να υλοποιείται μέσω εξοπλισμού κυριότητας του παρόχου, χωρίς να εμπλέκεται ο πελάτης στη μελέτη, επιλογή, εγκατάσταση και λειτουργία του.

Όπως γίνεται αντιληπτό, η μετάβαση από το τις αρχικές προς τις τελικές κατηγορίες αντιστοιχεί σε μείωση της εξάρτησης από το φυσικό προϊόν σαν κύριο συστατικό του συστήματος προϊόντος – υπηρεσίας. Ο πάροχος αποκτά σταδιακά μεγαλύτερη ευθύνη και ελευθερία στην ικανοποίηση των τελικών αναγκών του πελάτη. Ωστόσο, αυτό συνεπάγεται μεγαλύτερη ασάφεια στη διατύπωση των όρων της συνεργασίας (όπως π.χ. φαίνεται από τη διαφορά μεταξύ των σαφών τεχνικών προδιαγραφών ενός παγίου εξοπλισμού και των ενδεχομένων υποκειμενικών μέτρων ποιότητας για το παραγόμενο αποτέλεσμα). Ενδέχεται να μην είναι εύκολο για τους παρόχους να προσδιορίσουν το τι πρέπει να παραδώσουν, και στους πελάτες να ελέγξουν ότι τα παραδοτέα ικανοποιούν τις ανάγκες τους (Tukker A. , 2004).

Τέλος, ο κύκλος ζωής ενός ΣΠΥ περιλαμβάνει μία σειρά από στάδια αντίστοιχα των κύκλων ζωής ξεχωριστών προϊόντων και υπηρεσιών. Αυτό διανύει έναν κύκλο ζωής που αποτελεί μία σύνθεση των κύκλων ζωής του υλικού και άυλου τμήματός του, δηλαδή των φυσικών προϊόντων και των υπηρεσιών που το απαρτίζουν, και τον μηχανισμών για τη λειτουργία και διοίκησή του.

Είναι δυνατόν να περιγραφούν τέσσερις βασικοί τρόποι με τους οποίους συσχετίζονται οι κύκλοι ζωής υπηρεσιών και φυσικών προϊόντων.

- Μονόδρομη αλληλεπίδραση:
 - Προτεραιότητα προϊόντος: Η κύρια εστίαση της εταιρείας είναι στον κύκλο ζωής των φυσικών προϊόντων, οι οποίοι οδηγούν την εξέλιξη της υπηρεσιών. (π.χ. τυπικές περιπτώσεις υποστηρικτικών υπηρεσιών όπως συντήρηση κτλ.). Η υπηρεσία εξελίσσεται μόνο όταν εξελίσσεται το φυσικό προϊόν.

- Προτεραιότητα υπηρεσίας: Η εστίαση της εταιρείας είναι στην υπηρεσία – η υπηρεσία οδηγεί τον κύκλο ζωής του προϊόντος.
- Αμφίδρομη αλληλεπίδραση: Η υπηρεσία και το φυσικό αγαθό εξελίσσονται παράλληλα, αλλά με σχετικά χαλαρή συσχέτιση μεταξύ τους. Από την άποψη της διοίκησης, τα προϊόντα και οι κύκλοι ζωής τους είναι ξεχωριστοί (π.χ. αναπτύσσονται, παράγονται και διοικούνται από διαφορετικές ομάδες.
- Ενιαία εξέλιξη: Η υπηρεσία και το φυσικό αγαθό αντιμετωπίζονται ως ενιαίο προϊόν. Στη περίπτωση αυτή, σχεδιάζονται ως ενιαίο σύνολο και περνούν τα στάδια του κύκλου ζωής μαζί. Η διαδικασία αυτή απαντάται γενικά στη περίπτωση Συστημάτων Προϊόντος – Υπηρεσίας. (Wiesner, Guglielmina, Gusmeroli, & Doumeings, 2014)

Φυσικά, η παραπάνω ταξινόμηση δεν αποτυπώνει πλήρως την πολυπλοκότητα και πολλαπλότητα των προσεγγίσεων που εφαρμόζονται στη πραγματικότητα. Έτσι, π.χ. μία επιχείρηση μπορεί να ξεκινήσει ως πάροχος αποκλειστικά φυσικών αγαθών και υποτυπωδών υπηρεσιών υποστήριξης. Στη περίπτωση αυτή, η αλληλεπίδραση είναι κυρίως μονόδρομη. Ωστόσο, μπορεί να αποφασίσει να δώσει μεγαλύτερη εστίαση στη παροχή υπηρεσιών, οπότε και να προχωρήσει σε ένα μοντέλο αμφίδρομης αλληλεπίδρασης για το ίδιο προϊόν – έτσι ώστε οι υπηρεσίες και το φυσικό αγαθό να εξελίσσονται παράλληλα. Η εταιρεία μπορεί, στη συνέχεια, να αντιμετωπίσει το προϊόν ως σύστημα προϊόντος – υπηρεσίας, και να υιοθετήσει κοινές δομές και βήματα για την πορεία του.

2.4 Οικοσυστήματα Βιομηχανικών Υπηρεσιών

Η βιομηχανική δραστηριότητα δεν είναι απομονωμένη: οι επιχειρήσεις και οι οργανισμοί που συμμετέχουν σε έναν ή περισσότερους σχετιζόμενους κλάδους αλληλοσυνδέονται με διάφορα είδη σχέσεων. Οι σχέσεις μίας βιομηχανικής επιχείρησης με εξωτερικούς εμπλεκόμενους μπορεί να αφορούν πελάτες, προμηθευτές, συνεργάτες, κρατικούς φορείς, αλλά και τρίτους οργανισμούς όπως ΜΚΟ, συλλογικά και κλαδικά όργανα, κτλ.

Η επιχείρηση αλληλοεπιδρά με άλλες οντότητες στα πλαίσια της λειτουργίας και της προσπάθειας επίτευξης των στόχων της. Στην απλούστερη δυνατή περίπτωση, μία επιχείρηση έρχεται σε επαφή με τους πελάτες της, στους οποίους παρέχει αγαθά. Στη περίπτωση αυτή, η επιχείρηση εξάγει και επεξεργάζεται τις πρώτες ύλες χωρίς την εμπλοκή άλλων. Ωστόσο, κατά κανόνα, οι πρώτες ύλες και τα εξαρτήματα που συνθέτουν τα τελικά προϊόντα αγοράζονται από άλλους προμηθευτές. Επιπλέον, η επιχείρηση αγοράζει υπηρεσίες, με σκοπό την κατασκευή των προϊόντων, την παροχή τους σε πελάτες ή την υποστήριξη της λειτουργίας της. Επεκτείνοντας το παραπάνω μοντέλο, καθώς η επιχείρηση δραστηριοποιείται στα πλαίσια οργανωμένης κοινωνίας, έρχεται σε επαφή με κρατικούς φορείς και δημόσιες υπηρεσίες όπως, οι οποίες, π.χ. ελέγχουν την βιομηχανική δραστηριότητα, ρυθμίζουν τη διακίνηση και χρήση των προϊόντων, εισπράττουν φόρους κτλ.

Χρησιμοποιώντας την έννοια της «αλυσίδας αξίας», η συνεργασία όλων των εμπλεκόμενων στη παραγωγή του προϊόντος προσθέτει αξία, μέχρι που η πλήρης αξία του καταλήγει στον τελικό χρήστη. Παρόλα αυτά, ο όρος «αλυσίδα αξίας» περιγράφει τη συνεργασία αυτή σαν μονοδιάστατη διαδικασία, στην οποία διαδοχικοί εμπλεκόμενοι συνθέτουν ένα προϊόν που καταλήγει σε έναν τελικό αποδέκτη. Ωστόσο, ένα περισσότερο ακριβές μοντέλο θα αποτύπωνε το γεγονός ότι οι εμπλεκόμενοι στην παραγωγή αγαθών συνδέονται με

πολλαπλές σχέσεις και αλληλεξαρτήσεις, όχι απαραίτητα μονοδιάστατες, δηλαδή μόνο προς την κατεύθυνση του τελικού καταναλωτή. Για τον λόγο αυτό, το σύνολο των αλληλεξαρτώμενων εταιρών σε έναν κλάδο χαρακτηρίζεται συχνά ως «οικοσύστημα». Στη περίπτωση των βιομηχανικών επιχειρήσεων, το «βιομηχανικό οικοσύστημα» περιέχει όλους τους εμπλεκόμενους σε έναν κλάδο ή στη παραγωγή ενός προϊόντος.

Από τη βιβλιογραφία σχετικά με την ενσωμάτωση υπηρεσιών, γίνεται σαφές ότι η επιτυχής παροχή υπηρεσιών από τη βιομηχανική επιχείρηση απαιτεί:

- Βελτίωση της επικοινωνίας και συνεργασίας με πελάτες και προμηθευτές της επιχείρησης (π.χ. με μεταφορά τεχνογνωσίας ή υπηρεσιών για προβλήματα του πελάτη).
- Την «συν-δημιουργία» της προστιθέμενης αξίας.
- Τον διαμοιρασμό των κινδύνων μεταξύ των πελατών, εταιρείας και προμηθευτών.
- Τη γρήγορη απόκριση και συντονισμό πολλαπλών εμπλεκόμενων για την κάλυψη νέων καταστάσεων και σύνθετων αναγκών του πελάτη.

Οι παραπάνω απαιτήσεις μεταφράζονται στην ανάγκη για στενότερη συνεργασία στα πλαίσια του δικτύου πελατών, προμηθευτών και συνεργατών μίας επιχείρησης. Πράγματι, η ιστορική εμπειρία δείχνει ότι η ενσωμάτωση υπηρεσιών συχνά επιτυγχάνεται με την επέκταση και εμπλουτισμό της συνεργασίας με τους συνεργάτες και πελάτες της επιχείρησης. (Schmenner, 2009). Συγκεκριμένα, οι επιχειρήσεις τείνουν να εστιάζουν λιγότερο σε μεμονωμένες συναλλαγές, και περισσότερο στη δημιουργία μακροχρόνιων σχέσεων με τους προμηθευτές, τους συνεργάτες και τους πελάτες τους. Για να προσφέρει περίπλοκων και προηγμένων ΣΠΥ, μία επιχείρηση μπορεί να χρησιμοποιεί ένα δίκτυο από προμηθευτές υλικών και παρόχους υπηρεσιών.

Το φαινόμενο αυτό επιβεβαιώνεται πολλαπλά στη σχετική βιβλιογραφία. Π.χ. η Ng κ.α. δίνουν έμφαση στην συν-δημιουργία της αξίας και εισάγει ένα πλαίσιο για την παροχή πολύπλοκων υπηρεσιών. Σύμφωνα με αυτό μία υπηρεσία μπορεί να παρέχεται από ένα περίπλοκο δίκτυο εταιρών, και να περιλαμβάνει την ταυτόχρονη διαχείριση προσωπικού, πληροφορίας, υλικών και εξοπλισμού καθώς και τον διαμοιρασμό του κινδύνου σε πολλαπλούς εταίρους. (Ng, Parry, Wild, McFarlane, & Tasker, 2011). Η αποδοτικότητα της παροχής υπηρεσιών μπορεί να βελτιωθεί μέσω μίας δικτυωμένης οργάνωσης με αποδοτικότερες διαδικασίες για τη διαχείριση πόρων, γνώσης και προσωπικού. (Meier, Volker, & Funke, Industrial Product-Service Systems (IPS). Paradigm shift by mutually determined products and services, 2011). Η επιχείρηση μπορεί να παρέχει τα προϊόντα και τις υπηρεσίες της με πολλούς διαφορετικούς τρόπους και συνδυασμούς. Για τη σωστή εξυπηρέτηση του κάθε πελάτη και των ιδιαιτεροτήτων του, κάθε μορφή παροχής μίας υπηρεσίας, μπορεί να απαιτεί μοναδικούς συνδυασμούς από προϊόντα, υπηρεσίες προϊόντων και μηχανισμούς παροχής υπηρεσιών, άρα και μοναδικές συνεργασίες. (Baines T., et al., 2009). Ο όρος «επιχείρηση υπηρεσιών» (service enterprise) χρησιμοποιείται για να περιγράψει το περίπλοκο σύστημα των διασυνδεδεμένων και αλληλεξαρτώμενων δραστηριοτήτων που αναλαμβάνονται από ένα πολύπλοκο δίκτυο εμπλεκόμενων για την επίτευξη κοινών και σημαντικών στόχων» (Purchase, Parry, Valerdi, Nightingale, & Mills, 2011). Με βάση τα παραπάνω, η «ενσωμάτωση υπηρεσιών» είναι συνδεδεμένη με την

έννοια του «βιομηχανικού οικοσυστήματος», δηλαδή το πολύπλοκο δίκτυο σχέσεων, συνεργασιών και ανταγωνιστικών συγκρούσεων μέσα σε έναν βιομηχανικό κλάδο. (Neely, Benedettini, & Visnjic, 2011; Saccani, Visintin, & Rapaccini, 2014; Bikfalvi, Lay, Maloca, & Waser, 2013)

Οι Bastl κ.α. εξετάζοντας συνολικά τη σχετική βιβλιογραφία, παραθέτουν τους ακόλουθους τέσσερεις για τις συνέπειες στον τρόπο με τον οποίο η επιχείρηση εντάσσεται στο δίκτυο προμηθευτών, συνεργατών και πελατών της:

- Μία ολοκληρωμένη λύση ασκεί μεγαλύτερη πίεση στις εταιρείες και στο δίκτυο συνεργατών τους να αναπτύξουν και να ενοποιήσουν πολλαπλές ετερογενείς δεξιότητες ώστε να προσφέρουν προϊόντα με υποστηρικτικές υπηρεσίες. Για παράδειγμα, η Harley Davidson, για την παροχή ενός πλήρους πακέτου υπηρεσιών, συνεργάζεται με τις οργανώσεις κατόχων μοτοσικλετών, τους εμπορικούς αντιπροσώπους, τις κρατικές αρχές αδειοδότησης οχημάτων (για την έκδοση αδειών κυκλοφορίας), χρηματοπιστωτικά ιδρύματα (για τη χρηματοδότηση αγορών με δόσεις) και παρόχους υπηρεσιών εφοδιαστικής αλυσίδας (για τη μεταφορά των μοτοσικλετών στο δίκτυο πωλήσεων).
- Η ενσωμάτωση υπηρεσιών χαρακτηρίζεται από αυξημένο «πελατοκεντρικό» προσανατολισμό, σαν συνέπεια της μετάβασης από υπηρεσίες που σχετίζονται μόνο με το προϊόν σε υπηρεσίες που αφορούν τις διαδικασίες του πελάτη. Αυτό σημαίνει ότι η έμφαση δεν είναι πλέον μόνο στη καλή λειτουργία και χρήση του προϊόντος, αλλά και στην αποδοτικότητα/ αποτελεσματικότητα των διαδικασιών του χρήστη σε σχέση με τη συνολικά προσφερόμενη λύση. Ο πελατοκεντρικός προσανατολισμός είναι ανάλογος με το επίπεδο της συνεργασίας μεταξύ των εμπλεκόμενων μερών.
- Το γεγονός ότι παρέχονται υπηρεσίες κάνει το είδος των συναλλαγών μεταξύ πελατών και προμηθευτών διαφορετικό από ότι στη περίπτωση της προμήθειας υλικών αγαθών. Η παροχή υπηρεσιών επιτυγχάνεται καλύτερα με την δημιουργία σχέσεων, παρά με την εκτέλεση μεμονωμένων συναλλαγών. Η ανάπτυξη σχέσεων βελτιώνει την ποιότητα της παροχής υπηρεσιών καθώς οι ανάγκες του πελάτη γίνονται καλύτερα και αμεσότερα αντιληπτές.
- Τέλος, η μετατόπιση προς ολοκληρωμένες λύσεις σημαίνει ότι οι εμπλεκόμενοι πελάτες, προμηθευτές και συνεργάτες τείνουν να συνάπτουν μακροπρόθεσμες σχέσεις. Συνεπώς, η παροχή ολοκληρωμένων λύσεων απαιτεί τη δημιουργία συνεργασιών με βαθύτερο χρονικό ορίζοντα.

Οι σχέσεις αγοραστών - προμηθευτών σε συνθήκες ενσωμάτωσης υπηρεσιών μπορούν να αναλυθούν με βάση το μοντέλο των Cannon και Perrault σχετικά με τις διασυνδέσεις πελατών και προμηθευτών. Το μοντέλο αυτό προσδιορίζει πέντε βασικές διασυνδέσεις μεταξύ προμηθευτή και αγοραστή, δηλαδή την ανταλλαγή πληροφορίας, τις λειτουργικές διασυνδέσεις, τις νομικές δεσμεύσεις, τις συνεργατικές συνήθειες, και την προσαρμογή των εμπλεκόμενων οργανισμών στη μεταξύ τους σχέση. Οι Bastl κ.α., παρατηρούν ότι οι σχέσεις της επιχείρησης με το δίκτυο συνεργατών της αλλάζουν καθώς αυξάνεται η έμφαση στην ενσωμάτωση υπηρεσιών. Συγκεκριμένα, παρατηρούν τις ακόλουθες, προβλεπόμενες αλλαγές ανά τομέα:

- Ανταλλαγή πληροφορίας:
 - Αφορά στην ανταλλαγή και κοινοποίηση πληροφοριών που μπορεί να ενδιαφέρουν όλα τα μέρη. Αναμένεται αύξηση της αμφίδρομης επικοινωνίας, συχνότερες και περισσότερο ανοικτές επαφές. Επίσης, η καλύτερη ανταλλαγή πληροφορίας ενδέχεται να είναι κρίσιμη για την αντιμετώπιση της αβεβαιότητας.
- Λειτουργικές διασυνδέσεις
 - Διασυνδέσεις συστημάτων και διαδικασιών των δύο μερών για τη διευκόλυνση των λειτουργιών στις οποίες συνεργάζονται. Ο συντονισμός των επιχειρησιακών λειτουργιών των εμπλεκόμενων στην εφοδιαστική αλυσίδα συνδέεται με περισσότερο αποτελεσματική απόκριση στις ανάγκες του πελάτη. Οι διασυνδέσεις μπορούν να δημιουργούν γραμμές μεταξύ προμηθευτών και τελικών πελατών. Η διασύνδεση όμως μπορεί να επιφέρει μεγαλύτερα κόστη κατά τη διάλυση των σχέσεων προμηθευτή – αγοραστή.
- Νομικές δεσμεύσεις
 - Λεπτομερείς και δεσμευτικές συμβατικές συμφωνίες που ορίζουν τις υποχρεώσεις και ρόλους των εμπλεκόμενων μερών. Στα πλαίσια της στενότερης συνεργασίας, αναμένεται η ύπαρξη περισσότερων μηχανισμών διαχείρισης των σχέσεων στην εφοδιαστική αλυσίδα. Σε αυτό συμβάλλει η ανάγκη για δημιουργία λειτουργικών διασυνδέσεων και ανταλλαγή πληροφορίας, (που περιγράφηκαν προηγουμένως) αλλά και η ανάγκη για ισότιμη κατανομή των κινδύνων.
- Συνεργατικές συνήθειες
 - Αφορούν στις προσδοκίες που έχουν τα εμπλεκόμενα μέρη από τη συνεργασία τους για την επίτευξη κοινών και ατομικών στόχων. Οι προσδοκίες αυτές εμφανίζονται στην εταιρική κουλτούρα των εμπλεκόμενων μερών λόγω της στενής συνεργασίας και εκφράζονται με συνεπείς συμπεριφορές των εμπλεκόμενων.
- Προσαρμογή πελάτη και προμηθευτή
 - Οι προσαρμογές στη σχέση πελάτη - προμηθευτή αφορούν επενδύσεις για την προσαρμογή των διαδικασιών, των προϊόντων και της λειτουργίας του οργανισμού στις ανάγκες και δυνατότητες ενός εταίρου, η οποίες αυξάνονται λόγω της στενής αλληλεξάρτησης και των λειτουργικών προσαρμογών.

(Cannon & Perreault, 1999; Bastl, Johnson, Lightfoot, & Evans, 2012)

Τα δίκτυα συνεργασιών και ανταλλαγής υπηρεσιών στη βιομηχανία επιτρέπουν την εισαγωγή του όρου «οικοσύστημα βιομηχανικών υπηρεσιών». Η γενικότερη έννοια του οικοσυστήματος υπηρεσιών έχει οριστεί ως εξής : Τα οικοσυστήματα υπηρεσιών είναι «*σχετικά αυτό-περιεχόμενα, αυτορρυθμιζόμενα συστήματα από εμπλεκόμενους, (που αξιοποιούν και ολοκληρώνουν πόρους – resource integrators), συνδεδεμένων μέσω κοινών θεσμικών λογικών (institutional logics) και αμοιβαία δημιουργία αξίας μέσω της ανταλλαγής υπηρεσιών*» (Vargo S. , 2012).

Κατ' επέκταση, ένα «οικοσύστημα βιομηχανικών υπηρεσιών» είναι ένα τέτοιο οικοσύστημα υπηρεσιών, το οποίο αποτελείται κατά κύριο λόγο από συνεργαζόμενες επιχειρήσεις του βιομηχανικού κλάδου, οι οποίες έχουν σαν κύριο αντικείμενο την παραγωγή υλικών αγαθών για την καταναλωτική ή τη βιομηχανική αγορά.

Συγκεκριμένα, ένα οικοσύστημα βιομηχανικών υπηρεσιών είναι:

«Μία μη-ιεραρχική μορφή συνεργασίας, όπου διάφοροι οργανισμοί και άτομα συνεργάζονται με κοινούς ή συμπληρωματικούς στόχους πάνω σε συνδυασμούς προστιθέμενης αξίας προϊόντων και συσχετισμένων υπηρεσιών. Αυτό περιλαμβάνει την προώθηση, ανάπτυξη και παροχή νέων ιδεών, προϊόντων, διαδικασιών και αγορών. Διαδικτυακές αρχιτεκτονικές και πλατφόρμες επιτρέπουν την ενεργή συμμετοχή όλων των εμπλεκόμενων σε όλα τα στάδια του κύκλου ζωής προϊόντος και υπηρεσίας.» (Wiesner, Westphal, Hirsch, & Thoben, 2013)

Παραπλήσια έννοια, η οποία όμως υποδηλώνει εντονότερη συνεργασία μεταξύ των εμπλεκόμενων, είναι η «εικονική βιομηχανική επιχείρηση» (Virtual Manufacturing Enterprise – VME). Η ανάγκη για την παραγωγή σύνθετων προϊόντων (με υλικά αγαθά και υπηρεσίες, όπως τα ΣΠΥ) οδηγεί στη δημιουργία «εικονικών επιχειρήσεων» (virtual enterprises – VEs), δηλαδή βραχυπρόθεσμες συμμαχίες, οι οποίες αναπτύσσουν προϊόντα με συνεργατικό τρόπο, και οι οποίες συνεισφέρουν ευέλικτα πόρους, συστήματα παραγωγής και τεχνογνωσία (Romero, Rabelo, & Molina, 2012). Εναλλακτικά, μία εικονική επιχείρηση μπορεί να οριστεί ως μία εφοδιαστική αλυσίδα που σχηματίζεται κατά περίπτωση για ένα τελικό προϊόν. Στην αλυσίδα αυτή, οι συμμετέχοντες λειτουργούν δυναμικά σαν ένα δίκτυο προμηθευτών, κατασκευαστών, παρόχων υπηρεσιών και πελατών. Ο στόχος της εικονικής επιχείρησης είναι ο ταχύς και αποτελεσματικός συνδυασμός δεξιοτήτων για την παραγωγή συγκεκριμένων προϊόντων.

Κατά τους Zhang και Li, οι εικονικές επιχειρήσεις τείνουν να έχουν κάποια ιδιαίτερα χαρακτηριστικά (Zhang & Li, 1999):

- Υπάρχει μία κεντρική επιχείρηση που μπορεί και να μην διαθέτει η ίδια υποδομές παραγωγής.
- Η κεντρική επιχείρηση παράγει προϊόντα εξαρτώμενη από άλλες βιομηχανικές επιχειρήσεις.
- Το ποια θα είναι η κεντρική επιχείρηση εξαρτάται από το τελικό προϊόν. Αυτό συμβαίνει καθώς η εικονική επιχείρηση είναι προσανατολισμένη στο κεντρικό προϊόν.
- Η κεντρική επιχείρηση και οι εταίροι της σχηματίζουν μία οργανωτική δομή για την παραγωγή του προϊόντος, η οποία μπορεί να μην είναι μόνιμη.
- Μία επιχείρηση μπορεί να παίζει κεντρικό ρόλο σε μία συγκεκριμένη «εικονική επιχείρηση» και ρόλο εταίρου σε άλλη.

2.5 Η μετάβαση προς ένα μοντέλο παροχής υπηρεσιών

Υπάρχουν πολλοί λόγοι για τη μετάβαση μίας επιχείρησης σε στρατηγικές ενσωμάτωσης υπηρεσιών. Στη περίπτωση κλάδων με μη διαφοροποιημένα προϊόντα, η προσθήκη υπηρεσιών επιτρέπει τον ανταγωνισμό μέσω διαφοροποίησης και όχι μόνο μέσω κόστους.

Σε άλλες περιπτώσεις, οι επιχειρήσεις εκτιμούν ότι οι επιπρόσθετες υπηρεσίες και προσφέρουν μεγαλύτερα περιθώρια κέρδους (Gebauer, Fleisch, & Friedli, 2005) και ότι οι υπηρεσίες ενισχύουν τη διαφοροποίησή τους (Vandermerwe & Rada, 1988).

Οι Oliva και Kallenberg, με βάση την υφιστάμενη βιβλιογραφία, διατυπώνουν την άποψη ότι το σκεπτικό για τη μετάβαση σε λειτουργίες ενσωμάτωσης υπηρεσιών οργανώνεται σε τρεις άξονες:

- Οικονομικά κίνητρα:
 - Οι υπηρεσίες που αφορούν σε ήδη εγκατεστημένα προϊόντα με μακρύ χρόνο ζωής μπορούν να προσφέρουν αυξημένα και μακροχρόνια έσοδα.
 - Οι υπηρεσίες έχουν κατά κανόνα μεγαλύτερα περιθώρια κέρδους από τα αγαθά.
 - Σταθερότητα εσόδων, καθώς οι υπηρεσίες είναι περισσότερο ανθεκτικές στους οικονομικούς κύκλους που οδηγούν τις επενδύσεις και τις αγορές εξοπλισμού κ.α.
- Απαιτήσεις της αγοράς: Οι πελάτες βιομηχανικών προϊόντων απαιτούν περισσότερες υπηρεσίες και καλύτερη εξυπηρέτηση.
- Διατήρηση ανταγωνιστικού πλεονεκτήματος: Η μίμηση υπηρεσιών από ανταγωνιστές είναι δυσκολότερη καθώς οι μηχανισμοί με τους οποίους παρέχονται είναι λιγότερο ορατοί και εξαρτώνται περισσότερο από την ανθρώπινη εργασία. Συνεπώς μπορούν να αποτελέσουν μία σταθερότερη πηγή ανταγωνιστικού πλεονεκτήματος. (Oliva & Kallenberg, 2003)

Η θέση ενός επιχειρηματικού μοντέλου στο φάσμα της ενσωμάτωσης υπηρεσιών μπορεί να προσδιοριστεί με βάση τέσσερα κριτήρια (Martinez, Bastl, Kingston, & Evans, 2010):

- Η «πηγή αξίας των δραστηριοτήτων», η οποία αντιστοιχεί στον κύριο κίνητρο που ωθεί τους πελάτες στον συγκεκριμένο πάροχο αγαθών-υπηρεσιών.
- Ο «πρωτεύων ρόλος των παγίων», ο οποίος αφορά τη φύση της ανάγκης για το πάγιο, η οποία μπορεί να κυμαίνεται σε δύο άκρα: τη αξιοποίηση του παγίου (με τη σχετική έμφαση στη διαδικασία παροχής της σχετικής αξίας προς τον πελάτη) και την απλή κατοχή του παγίου.
- Ο «τύπος προσφερόμενου προϊόντος», ο οποίος αναδεικνύει το φάσμα των συνδυασμών αγαθού – υπηρεσίας, και εκτείνεται από αποκλειστικά υλικά αγαθά μέχρι αποκλειστικά άυλες υπηρεσίες
- Η «στρατηγική παραγωγής» που καθορίζει το επίπεδο προσαρμογής των προσφερόμενων συνδυασμών αγαθού – υπηρεσίας

Τα παραπάνω κριτήρια αποτυπώνονται στον ακόλουθο πίνακα:

Επίπεδο ενσωμάτωσης υπηρεσιών	Υψηλή ενσωμάτωση υπηρεσιών	Χαμηλή ενσωμάτωση υπηρεσιών
Πηγή αξίας δραστηριοτήτων	Βασισμένη σε σχέσεις	Βασισμένη σε συναλλαγές
Πρωτεύων ρόλος των παγίων	Αξιοποίηση παγίου	Κατοχή παγίου

Τύπος προσφερόμενου προϊόντος	Συνολική ολοκλήρωση υπηρεσιών	Φυσικό αγαθό και συσχετισμένες υπηρεσίες
Στρατηγική παραγωγής	Μαζική ή ατομική εξατομίκευση (mass / pure customisation)	Μαζική παραγωγή

Πίνακας 2: Κριτήρια προσδιορισμού για το επίπεδο ενσωμάτωσης υπηρεσιών (Martinez, Bastl, Kingston, & Evans, 2010)

Η μετάβαση στην ενσωμάτωση υπηρεσιών απαιτεί την αλλαγή προσανατολισμού της επιχείρησης και μπορεί να συναντήσει πολλαπλά εμπόδια και δυσκολίες. Για μία καθαρά βιομηχανική επιχείρηση, η παροχή υπηρεσιών απαιτεί νέες οργανωτικές αρχές, δομές και διαδικασίες. Απαιτούνται νέες ικανότητες, μετρικές και κίνητρα. Στο πλέον βασικό επίπεδο, πρέπει να γίνει μετατόπιση από ένα μοντέλο «διεκπεραίωσης συναλλαγών» (π.χ. πωλήσεις φυσικών προϊόντων) σε ένα μοντέλο «δημιουργίας σχέσεων» με τους πελάτες. Μάλιστα, η ανάπτυξη των απαραίτητων νέων δυνατοτήτων απαιτεί την επένδυση οικονομικών και διοικητικών πόρων που υπό άλλες συνθήκες θα χρησιμοποιούνταν στην έρευνα / ανάπτυξη και την παραγωγή, δηλαδή τις παραδοσιακές πηγές ανταγωνιστικού πλεονεκτήματος μίας βιομηχανικής επιχείρησης. (Oliva & Kallenberg, 2003)

Εναλλακτικά, η μετάβαση μπορεί να περιγραφεί ως μία «αλληλεπιδραστική» διαδικασία κατά την οποία ο πελάτης και ο προμηθευτής κινούνται μαζί προς την ενσωμάτωση υπηρεσιών. Καθώς ο προμηθευτής προχωρά σε μεγαλύτερη ενσωμάτωση υπηρεσιών, η διεπαφή περνά από τα ακόλουθα στάδια:

- 1^ο στάδιο: μικρή ενσωμάτωση υπηρεσιών: Οι αλληλεπιδράσεις αφορούν μεμονωμένες συναλλαγές υπηρεσιών και περιφερειακές υπηρεσίες στο κύριο προϊόν
- 2^ο στάδιο: παράδοση αγαθού και υπηρεσίας
- 3^ο στάδιο: προσαρμογή του αγαθού και της υπηρεσίας στις ιδιαίτερες ανάγκες του πελάτη
- 4^ο στάδιο: το αγαθό και η υπηρεσία σχεδιάζονται μαζί με σκοπό την παροχή συνολικών λύσεων.

Οι Martinez κ.α. με βάση την έρευνά τους στο πεδίο των επιχειρήσεων βιομηχανικού εξοπλισμού εντοπίζουν βασικά εμπόδια σε πέντε διαστάσεις (κουλτούρα, παράδοση ενοποιημένου προϊόντος, εσωτερικές διαδικασίες και δεξιότητες, στρατηγική ευθυγράμμιση, σχέσεις με προμηθευτές και συνεργάτες):

- Εδραιωμένη κουλτούρα προϊόντος – υπηρεσίας
 - Οι παραδοσιακοί βιομηχανικοί παραγωγή έχουν ισχυρό τεχνολογικό προσανατολισμό, κάτι που εμποδίζει τη μετάβαση σε μία κουλτούρα με προσανατολισμό υπηρεσιών.
- Παράδοση ενοποιημένου προϊόντος (αγαθών – υπηρεσιών).
 - Η επαφή προμηθευτή και πελάτη πρέπει να γίνεται μέσω πολλαπλών σημείων: η μεγαλύτερη συνεργασία κατά τον κύκλο ζωής του προϊόντος

επιβάλλει την εμπλοκή περισσότερων μονάδων και προσωπικού και από τις δύο πλευρές.

- Απαιτείται εστίαση στη παράδοση ενοποιημένου αγαθού – υπηρεσίας και όχι κυρίως αγαθού. Η έλλειψη μηχανισμών ανταπόκρισης μπορεί να δυσχεραίνει την παροχή του ενοποιημένου προϊόντος.
- Εσωτερικές διαδικασίες και δεξιότητες
 - Απαιτείται ο σχεδιασμός του ενοποιημένου προϊόντος που απαιτεί παράλληλο σχεδιασμό του προϊόντος και των σχετικών υπηρεσιών.
 - Τα κριτήρια και οι μετρικές απόδοσης πρέπει να ανασχεδιαστούν με έμφαση και στην παράδοση υπηρεσιών, ώστε να αντανακλούν την συνολική αποτελεσματικότητα της επιχείρησης στην παροχή του ενοποιημένου προϊόντος. Επιπλέον, τα κριτήρια απόδοσης για την παροχή του ενοποιημένου προϊόντος/υπηρεσίας πρέπει να ευθυγραμμίζονται μεταξύ των εμπλεκόμενων οργανισμών.
 - Κατά τη μετάβαση, απαιτείται η δημιουργία συγκεκριμένων και διακριτών υποδομών (οργανωτικών μονάδων, δικτύων, τεχνογνωσίας) για την παροχή των συνδυασμένων υπηρεσιών.
- Στρατηγική ευθυγράμμιση
 - Η απουσία εσωτερικής συνεργασίας, καθώς και κοινής γλώσσας / ευθυγράμμισης νοοτροπίας με τους πελάτες επιβραδύνει τις προσπάθειες για μετάβαση στο νέο μοντέλο. Η ανάπτυξη της κοινής γλώσσας και νοοτροπίας είναι κρίσιμη για την επίτευξη κοινής αντίληψης για τα προβλήματα που πρέπει να λυθούν, τη διερεύνηση και την αξιολόγηση των εφαρμοζόμενων λύσεων.
- Σχέσεις επιχείρησης με τους προμηθευτές και συνεργάτες της
 - Οι αλλαγές στις σχέσεις της επιχείρησης με τους πελάτες μπορεί να μην αντανακλώνται στις σχέσεις της επιχείρησης με τους προμηθευτές της. Οι προμηθευτές μπορεί να παράσχουν μέρη του υλικού προϊόντος ή μέρη των υπηρεσιών (π.χ. επιτόπια υποστήριξη, εγκατάσταση και άλλες υπηρεσίες).
 - Αν οι σχέσεις της εταιρείας με τους προμηθευτές της στηρίζονται σε μία λογική «μεμονωμένων συναλλαγών», τότε ενδέχεται να εμφανίζονται δυσκολίες στην υποστήριξη του ενοποιημένου προϊόντος από το εξωτερικό της δίκτυο. Η επιτυχημένη παροχή ενοποιημένου προϊόντος/ υπηρεσίας απαιτεί καλή επικοινωνία και «διαφανή» συνεργασία με τους προμηθευτές σε πρότερα σημεία της συγκεκριμένης αλυσίδας αξίας. Οι ενδεχόμενες δυσλειτουργίες της συνεργασίας με τους προμηθευτές θα μεταδίδονται και θα εκδηλώνονται ως δυσλειτουργίες των παρεχόμενων υπηρεσιών προς τον τελικό πελάτη. (Martinez, Bastl, Kingston, & Evans, 2010).

Ωστόσο, ενδέχεται να υπάρξουν προβλήματα με τη σταδιακή μετάβαση. Οι παραδοσιακές βιομηχανικές επιχειρήσεις έχουν ήδη παγιωμένες δομές, κουλτούρα και σχέσεις με τους εταίρους τους. Έτσι, ενώ μία επιχείρηση μπορεί να εκτελεί μικρές αλλαγές και μεταρρυθμίσεις, η συνολική της κουλτούρα και λειτουργία μπορεί να παραμείνει καθηλωμένη στη παροχή αγαθών και να δυσκολεύεται να μεταβεί σε περισσότερο δυναμικούς τρόπους εργασίας. Σε αυτή την περίπτωση ενδέχεται να βοηθήσει η δημιουργία

νέας ή αυτόνομης οργανωτικής δομής για την παροχή υπηρεσιών, με έμφαση στην ευελιξία και την αποδοτικότητα. (Barnett, Parry, Saad, Newnes, & Goh, 2013)

Στη μετάβαση προς την παροχή υπηρεσιών, πρέπει να υπάρξουν αλλαγές τόσο στον τρόπο λειτουργίας όσο και στην κουλτούρα του οργανισμού. Υιοθετώντας μία προσέγγιση με έμφαση στις υπηρεσίες επιβάλλει τη μετατόπιση από τον πληθυντικό όρο «υπηρεσίες» στον ενικό όρο «εξυπηρέτηση» - αντανακλώντας με αυτό τον τρόπο τη χρήση των πόρων της επιχείρησης προς όφελος του πελάτη (Vargo & Lusch, 2007). Επίσης, οι υπηρεσίες είναι άυλες και απαιτούν επικοινωνία με τους πελάτες. Συνεπώς, το προσωπικό που εμπλέκεται στη παροχή υπηρεσιών πρέπει να αναλαμβάνει περισσότερες πρωτοβουλίες, να αντιμετωπίζει καλύτερα την πίεση, να έχει περισσότερο αναπτυγμένες διαπροσωπικές ικανότητες και να συνεργάζεται πιο αποτελεσματικά από τους συναδέλφους τους στη παραγωγή. (Schneider, 1995). Οι Ng κ.α. προσθέτουν στο παραπάνω επισημαίνοντας ότι η οργάνωση, η νοοτροπία και κουλτούρα τόσο των προμηθευτών όσο και των πελατών, πρέπει να δίνουν έμφαση στη συνεργασία και επικοινωνία. Μέσω της στενότερης συνεργασίας και της αμοιβαίας εξάρτησης μπορούν να επιτευχθούν αμοιβαίες ωφέλειες. (Ng, Parry, Wild, McFarlane, & Tasker, 2011; Duffy & Fearn, 2004).

3 Θεωρητικό και Τεχνολογικό Υπόβαθρο

3.1 Υπηρεσίες και Τεχνολογίες Πληροφορικής και Επικοινωνιών

Οι Τεχνολογίες Πληροφορικής και Επικοινωνιών είναι πλέον κρίσιμος παράγοντας για την παροχή προηγμένων υπηρεσιών και τη δημιουργία των σύγχρονων Συστημάτων Προϊόντος – Υπηρεσίας. Οι ΤΠΕ παρέχουν τα απαραίτητα κανάλια επικοινωνίας, υλοποιούν περίπλοκες επιχειρησιακές λογικές, παρέχουν αυτοματισμούς και διακινούν, αποθηκεύουν, και αναλύουν μεγάλες ποσότητες δεδομένων.

Για παράδειγμα, μία χαρακτηριστική περίπτωση βιομηχανικού Συστήματος Προϊόντος – Υπηρεσίας είναι η απομακρυσμένη παρακολούθηση βιομηχανικού εξοπλισμού και η παροχή υπηρεσιών συντήρησης όποτε εμφανιστούν συγκεκριμένες συνθήκες βλάβης. Η εταιρεία μπορεί επίσης να ενημερώνει αυτόματα τον χρήστη και να προγραμματίζει ραντεβού συντήρησης. Τα πεδία εφαρμογής ΤΠΕ αφορούν:

1. Συλλογή δεδομένων από το υποστηριζόμενο προϊόν
2. Μετάδοση των δεδομένων λειτουργίας και σφάλματος στον κατασκευαστή
3. Αποθήκευση και αρχειοθέτηση δεδομένων
4. Ανάλυση δεδομένων
5. Απόκριση στη βλάβη

Η λειτουργία του συγκεκριμένου ΣΠΥ γίνεται μόνο με την εφαρμογή τεχνολογιών αισθητήρων, τεχνολογιών διαδικτύου και συστημάτων υποστήριξης αποφάσεων, ενώ πρέπει να υποστηρίζονται από τις απαραίτητες υπολογιστικές και επικοινωνιακές υποδομές. (Baines & Lightfoot, 2013).

Σε γενικές γραμμές, η υφιστάμενη βιβλιογραφία υποδεικνύει ότι οι ΤΠΕ επιτρέπουν στις επιχειρήσεις να έχουν ανταγωνιστικό πλεονέκτημα μέσω διαφοροποίησης των υπηρεσιών που υποστηρίζουν τα προϊόντα και τους πελάτες τους. Συγκεκριμένα, οι ΤΠΕ επιτρέπουν την παραγωγή επιπλέον προστιθέμενης αξίας για τον πελάτη, καλύτερη ποιότητα, ολοκληρωμένες λύσεις και μειωμένα κόστη. Νέες υπηρεσίες μπορούν να έρθουν στην αγορά πιο γρήγορα με τη χρήση ΤΠΕ, ενώ κάνουν δυνατή την παροχή νέων, μοναδικών υπηρεσιών, π.χ. με τη συλλογή και επεξεργασία δεδομένων σε πραγματικό χρόνο. Ωστόσο, ενώ είναι εύκολο να αντιγραφούν οι υποδομές ΤΠΕ, το πραγματικό πλεονέκτημα διαφοροποίησης έγκειται στις νέες διαδικασίες και τρόπους παραγωγής που επιτρέπουν οι ΤΠΕ.

Οι Kowalkowski, Kindstrom και Gebauer, σε μελέτη οκτώ βιομηχανικών εταιρειών σε διάφορα στάδια ενσωμάτωσης υπηρεσιών, παρατηρούν ότι, αρχικά, οι ΤΠΕ σε μία επιχείρηση εισάγονται κυρίως γιατί επιτρέπουν τη μείωση του κόστους και την αύξηση της αποδοτικότητας, π.χ. μέσω καλύτερων συστημάτων επικοινωνίας, αυτοματοποίηση λειτουργικών κτλ. Στη συνέχεια όμως, οι ΤΠΕ χρησιμοποιούνται για να επιτύχουν διαφοροποίηση μέσω της προσφοράς καινοτόμων υπηρεσιών που γίνονται πλέον εφικτές με τη χρήση τεχνολογικών μέσων. (Kowalkowski, Kindstrom, & Gebauer, 2013).

Εξετάζοντας τη σχετική βιβλιογραφία, οι Grace, Finnegan και Butler παρατηρούν ότι τα εργαλεία ΤΠΕ επιδρούν, όχι μόνο στη παροχή των υπηρεσιών, αλλά στη διαδικασία της «συν-δημιουργίας αξίας» με τον πελάτη, δηλαδή τη συνεργασία με στόχο τη βελτίωση του

περιεχομένου της υπηρεσίας και την προστιθέμενη αξία που αποκομίζουν τα δύο μέρη (βλ. ενότητα 2.1). Η συν-δημιουργία αξίας μπορεί να επηρεαστεί από τη χρήση εργαλείων επικοινωνίας με ΤΠΕ, με την αφαίρεση χρονικών, χωρικών και υλικών φραγμών που εμποδίζουν την αποτελεσματική συνεργασία, την ανταλλαγή δεδομένων με ηλεκτρονικό τρόπο κατά τη διάρκεια της παροχής υπηρεσιών, αλλά και την αύξηση της κοινωνικής απόστασης λόγω της μείωσης των διαπροσωπικών επαφών (Grace, Finnegan, & Butler, 2008).

Στο ίδιο πνεύμα, οι Heiskala, Hiekkänen και Korhonen παρατηρούν ότι η εισαγωγή των ΤΠΕ μετατοπίζει την εστίαση από την ανθρώπινη προσπάθεια για την παροχή της υπηρεσίας στη διαχείριση, συντήρηση, σχεδιασμό και ανάπτυξη των υποδομών για την παροχή της. Η παροχή της υπηρεσίας, η οποία γίνεται με μοναδικό τρόπο για κάθε πελάτη, περιορίζεται πλέον στα όρια που προδιαγράφει ο σχεδιασμός του τεχνολογικού συστήματος για την παροχή της. Το σύστημα ΤΠΕ που την παρέχει λειτουργεί μόνο εντός των παραμέτρων για τα οποία έχει σχεδιαστεί, και δεν είναι πάντα εύκολο να προσαρμοστεί στις ανάγκες του πελάτη, όπως οι παραδοσιακές υπηρεσίες. Για τον λόγο αυτό πρέπει να δίνεται έμφαση στον σωστό σχεδιασμό των υποδομών ΤΠΕ και στην κατανόηση του τρόπου με τον οποίο ο πελάτης προσλαμβάνει την προστιθέμενη αξία των υπηρεσιών. Ένας τρόπος με τον οποίο μπορεί να επιτευχθεί αυτό είναι η συμμετοχή του πελάτη (ή πελατών) στην διαδικασία καινοτομίας, σχεδιασμού και ανάπτυξης, σε αντίθεση με τις παραδοσιακές «κλειστές» μεθόδους για την ανάπτυξη υπηρεσιών. Έτσι, μέρος της «συν-δημιουργίας αξίας» μετατοπίζεται στις φάσεις σχεδιασμού και ανάπτυξης. Επιπλέον, η χρήση των ΤΠΕ στη παροχή υπηρεσιών υποχρεώνει την επιχείρηση σε επαναπροσδιορισμό των απαιτούμενων δεξιοτήτων. Η επιχείρηση που παρέχει υπηρεσίες με τη χρήση ΤΠΕ δεν χρειάζεται μόνο δεξιότητες, πόρους και γνώση για την παροχή των υπηρεσιών (π.χ. τεχνικοί, σύμβουλοι, πωλητές, και εργαλεία), αλλά και για τον σχεδιασμό, ανάπτυξη, συντήρηση, αναβάθμιση και διαχείριση των υποδομών ΤΠΕ που θα τις παρέχουν (π.χ. αναλυτές, προγραμματιστές, διαχειριστές συστήματος, πακέτα λογισμικού, κέντρα δεδομένων και άλλα). (Heiskala, Hiekkänen, & Korhonen, 2011)

Τέλος, το απαιτούμενο επίπεδο συνεργασίας σε μία εικονική επιχείρηση είναι δυνατό μόνο μέσω της χρήσης τεχνολογιών πληροφορικής και επικοινωνιών. (Martinez, Fouletier, Park, & Favrel, 2001) (Esposito & Evangelista, 2014). Οι ΤΠΕ επιτρέπουν τη συλλογή δεδομένων σε πραγματικό χρόνο, την παρακολούθηση παραγωγικών λειτουργιών και εφοδιαστικών δικτύων, κρίσιμων δεικτών απόδοσης (key performance indicators – KPIs) και τη συνολική διαχείριση (Kokkinakos, Markaki, Panopoulos, Kousouris, & Askounis, 2013).

Κατά τον Chae, οι υπηρεσίες που στηρίζονται σε ΤΠΕ («IT-Enabled Services») δεν νοούνται χωρίς την υποδομή ΤΠΕ και το απαραίτητο λογισμικό, τα οποία ενεργούν ως μέσα για την παροχή τους. Έτσι, η εξέλιξη του τρόπου παροχής τους απαιτεί τη συνεχή τροποποίηση του συστήματος ΤΠΕ που την υποστηρίζει, με αποτέλεσμα τη δημιουργία νέων εκδόσεων του σχετικού λογισμικού και νέων στοιχείων (π.χ. τεχνολογίες, λειτουργίες, χρήστες κτλ.). Ένα λογισμικό για την υποστήριξη υπηρεσιών δεν μπορεί να εννοηθεί ποτέ ως ολοκληρωμένο - όσο τουλάχιστον εξελίσσεται η υπηρεσία ως αποτέλεσμα των μεταβαλλόμενων συνθηκών της αγοράς και του ανταγωνισμού.

Σε αντίθεση με τις παραδοσιακές προσεγγίσεις για την ανάπτυξη προϊόντων, υπηρεσιών, συστημάτων προϊόντων – υπηρεσιών και λογισμικού, η ανάπτυξη υπηρεσιών βασισμένων σε ΤΠΕ ξεπερνά τα στενά όρια του οργανισμού και απαιτεί τη συνεργασία πολλαπλών εμπλεκόμενων, συμπεριλαμβανομένων των πελατών, συνεργατών και ειδικών. Η διαδικασία ανάπτυξης είναι δυναμική, και μη γραμμική, ενώ δεν υπάρχει ένας κεντρικός σχεδιαστής και σημείο ελέγχου.

Τέλος, η ανάπτυξη των υπηρεσιών αυτών βασίζεται σε μία διαδικασία «δημιουργίας ποικιλίας, επιλογής και συγκράτησης». Η καινοτομία απαιτεί τη δημιουργία μίας ποικιλίας από στοιχεία όπως διαδικασίες, σύνολα δεδομένων, νέες λειτουργικότητες κ.α., από τα οποία επιλέγονται συγκεκριμένα για χρήση σε ένα νέο σύστημα υπηρεσίας, ενώ τα πλέον επιτυχημένα συγκρατούνται για επαναχρησιμοποίηση. Συνεπώς, συχνά εμφανίζεται η ανάγκη για διατήρηση και αναχρησιμοποίηση των στοιχείων ΤΠΕ του συστήματος υπηρεσίας. (Chae, 2014)

Οι μορφές υπηρεσιών που καλείται να υποστηρίξει ένα επιχειρησιακό λογισμικό είναι πολλαπλές. Οι ΤΠΕ μπορούν να υποστηρίξουν άμεσες επαφές πελάτη και παρόχου, υποστηρίζοντας μόνο τον πάροχο (π.χ. στην ανάκτηση δεδομένων για τον πελάτη) κατά την παροχή της υπηρεσίας, αλλά και τους δύο στη διευκόλυνση της συνεργασίας τους (π.χ. τη χρήση ενός συστήματος από κοινού κατά την εξυπηρέτηση). Ωστόσο, οι ΤΠΕ υποστηρίζουν και έμμεσες αλληλεπιδράσεις: η υπηρεσία παρέχεται στον πελάτη από τον πάροχο διαμέσου ενός συστήματος ΤΠΕ, είτε υπό τον χειρισμό προσωπικού του παρόχου (π.χ. επικοινωνία μέσω συστήματος συνομιλίας κειμένου - chat) είτε αυτόματα, με το προσωπικό του παρόχου να διαχειρίζεται απλώς τα συστήματα ΤΠΕ (Froehle & Roth, 2004).

Μία εναλλακτική τυπολογία υπηρεσιών που παρέχονται μέσω ΤΠΕ είναι η παρακάτω (Schumann, Wunderlich, & Wangenheim, 2012):

- Αυτοεξυπηρέτηση
 - Βασισμένη σε τεχνολογικές υποδομές κυρίως του παρόχου: π.χ. η χρήση μηχανημάτων αυτόματης ανάληψης.
 - Βασισμένη σε τεχνολογικές υποδομές του παρόχου και του πελάτη: π.χ. η χρήση ηλεκτρονικής τραπεζικής, υπηρεσίες διαμέσου φορητών συσκευών.
- Εξυπηρέτηση από άλλους
 - Απομακρυσμένες / αλληλεπιδραστικές υπηρεσίες
 - Βασισμένη σε τεχνολογικές υποδομές κυρίως του παρόχου: π.χ. η απομακρυσμένη ρομποτική χειρουργική.
 - Βασισμένη σε τεχνολογικές υποδομές του παρόχου και του πελάτη: π.χ. η απομακρυσμένη διάγνωση και διαχείριση βιομηχανικού εξοπλισμού.

Συνθέτοντας τις παραπάνω προσεγγίσεις, ο Glushko προτείνει τον παρακάτω κατάλογο από λειτουργικότητες (Glushko, 2010):

- Επαφές πρόσωπο με πρόσωπο πελάτη και παρόχου: π.χ. τεχνική υποστήριξη πελάτη και ανάκτηση πληροφορίας από σύστημα ΤΠΕ, προσαρμογή προϊόντος σε

συνεργασία με τον πωλητή κτλ., όπου η παροχή της υπηρεσίας γίνεται με προσωπική επαφή η οποία υποστηρίζεται τεχνολογικά.

- Αυτοεξυπηρέτηση με τη χρήση τεχνολογίας: π.χ. αυτόματη εγκατάσταση ενημερώσεων λογισμικού, αυτόματη παρακολούθηση λειτουργίας και ενεργοποίησης προληπτικής συντήρησης κτλ., στις οποίες η υπηρεσία παρέχεται από τεχνολογικά συστήματα που διαχειρίζεται ο πάροχος.
- Υπηρεσίες πολλαπλών καναλιών: υπηρεσίες παρεχόμενες μέσω διαφορετικών τρόπων: π.χ. υπηρεσίες πληροφόρησης που παρέχονται και μέσω διαδικτύου, αλλά και με προσωπικές επαφές, τηλεφωνική εξυπηρέτηση, αυτοεξυπηρέτηση με τεχνολογικά μέσα κτλ.
- Υπηρεσίες πολλαπλών συσκευών και πλατφορμών: π.χ. φορητές συσκευές, συστήματα αισθητήρων, κεντρικά συστήματα ΤΠΕ και βάσεις δεδομένων κ.α., στις οποίες η υπηρεσία παρέχεται μέσω πολλαπλών τεχνολογικών διεπαφών και μέσων
- Υπηρεσίες με έντονη χρήση back-office: υπηρεσίες που, πριν φτάσουν στον πελάτη, απαιτούν τον συντονισμό πολλαπλών εμπλεκόμενων και τον συνδυασμό ενός αριθμού διαδικασιών π.χ. υπηρεσίες συντήρησης όπου απαιτείται συντονισμός των κατασκευαστών ανταλλακτικών, παρόχων τεχνικών υπηρεσιών, μεταφορέων κ.α. Οι υπηρεσίες αυτές μπορεί να απαιτούν την ενεργοποίηση σημαντικών πόρων ΤΠΕ, όπως αυτοματισμούς, δίκτυα επικοινωνιών, βάσεις δεδομένων και συστήματα υποστήριξης αποφάσεων.
- Υπηρεσίες με βάση την επίγνωση πλαισίου στο οποίο παρέχεται η υπηρεσία, όπως χρόνος, τοποθεσία, δραστηριότητα κτλ.: π.χ. εντοπισμός και διανομή προϊόντων με τεχνολογίες RFID και χρήση GPS για την υπόδειξη κοντινών εξουσιοδοτημένων συνεργείων.

Μία υπηρεσία μπορεί να χρησιμοποιεί ένα ή περισσότερα από τα παραπάνω μοντέλα παροχής υπηρεσιών με ΤΠΕ. Επιπλέον, υπάρχουν απαιτήσεις για την καταγραφή και την ανάλυση μεγάλου όγκου δεδομένων για την αυτοματοποιημένη παροχή υπηρεσιών, με στόχο τη διασφάλιση ποιότητας των υπηρεσιών και τη συνολική τους διοίκηση (Neff, Hamel, Herz, Uebernickel, & Brenner, 2013).

3.2 Υπηρεσιοστραφείς Αρχιτεκτονικές και Διαχείριση Επιχειρησιακών Διαδικασιών

Το λογισμικό παροχής υπηρεσιών θα λειτουργεί στο πλαίσιο των σύγχρονων τεχνολογιών και αρχιτεκτονικών ΤΠΕ. Στις επιχειρησιακές εφαρμογές σήμερα, δύο εδραιωμένες τεχνολογικές προσεγγίσεις είναι ιδιαίτερα συμβατές με τις απαιτήσεις της υπηρεσιοποίησης στη βιομηχανία: η «Υπηρεσιοστραφής Αρχιτεκτονική» (Service-Oriented Architecture – SOA) και η «Διαχείριση Επιχειρησιακών Διαδικασιών» (Business Process Management – BPM). Οι προσεγγίσεις αυτές αναλύονται στις ακόλουθες ενότητες.

3.2.1 Υπηρεσιοστραφείς αρχιτεκτονικές

Η «Υπηρεσιοστραφής Αρχιτεκτονική» (Service-Oriented Architecture, SOA) αποτελεί μία από τις σημαντικότερες σύγχρονες προσεγγίσεις στην υλοποίηση περίπλοκων συστημάτων πληροφορικής. Η SOA είναι μία μορφή αρχιτεκτονικής πληροφοριακών συστημάτων βασισμένη σε χαλαρά διασυνδεδεμένες «υπηρεσίες». Μία υπηρεσία (κατά την έννοια της

SOA) είναι μία καθορισμένη λειτουργία που εκτελεί ένας «πάροχος υπηρεσίας» (service provider) για έναν «καταναλωτή» (service consumer). Τόσο ο πάροχος όσο και ο καταναλωτής είναι «πράκτορες λογισμικού» (software agents). Η εκτελούμενη λειτουργία μπορεί να είναι χαμηλού επιπέδου (atomic) ή να αποτελεί σύνθεση περισσότερων τέτοιων λειτουργιών. Η κάθε μία υπηρεσία διαθέτει μία σαφώς καθορισμένη διεπαφή, η οποία είναι προσβάσιμη από τις εφαρμογές στα πλαίσια του συγκεκριμένου πληροφοριακού συστήματος ή μεταξύ διαφορετικών πληροφοριακών συστημάτων (Parazoglou & Georgakopoulos, 2003). Στις υπηρεσιοστραφείς αρχιτεκτονικές, οι περιγραφές των υπηρεσιών (διεπαφές, διαλειτουργικότητα κ.α.) μπορούν να αποθηκεύονται σε ένα μητρώο υπηρεσιών (service registry), και μέσω αυτού να «δημοσιεύονται» στο σύστημα, επιτρέποντας σε άλλους εμπλεκόμενους (όπως π.χ. χρήστες, προγραμματιστές, άλλες εφαρμογές, υπηρεσίες κ.α.) να τις «ανακαλύπτουν» (service discovery) και να τις χρησιμοποιούν. Οι υποστηρικτικές λειτουργίες μίας αρχιτεκτονικής τύπου SOA, υλοποιούνται από εξειδικευμένο λογισμικό, το οποίο επιτρέπει την εκτέλεση των υπηρεσιών, τη σύνθεση και συντονισμό τους, καθώς και την εποπτεία και διαχείριση του συνολικού συστήματος. Ο προσανατολισμός υπηρεσιών θα πρέπει, κατά ελάχιστο, να υποστηρίζει την εγκατάσταση, δημοσίευση και καταχώρηση της περιγραφής μίας υπηρεσίας στο κατάλληλο μητρώο, την πρόσβαση στην υπηρεσία, καθώς και τον συντονισμό και την εκτέλεση συνθέσεων υπηρεσιών. Η υποδομή αυτή αναφέρεται ως Υπηρεσιοστραφές Ενδιάμεσο Λογισμικό (Service Oriented Middleware – SOM) (Issarny, et al., 2011). Τη στιγμή αυτή, οι περισσότερες πρακτικές εφαρμογές υπηρεσιοστραφών αρχιτεκτονικών βασίζονται σε πρότυπα και τεχνολογίες web services, (Rosen, Lublinsky, Smith, & Balcer, Applied SOA: Service Oriented Architecture and Design Strategies, 2012), όπως το REST, το SOAP, η XML, η WSDL και άλλες.

Έτσι, σε ένα πληροφοριακό σύστημα βασισμένο στις αρχές της SOA, είναι αυτό στο οποίο όλες οι λειτουργικότητες είναι ορισμένες ως ανεξάρτητες υπηρεσίες, με καλά καθορισμένες διεπαφές, οι οποίες μπορούν να κληθούν με συγκεκριμένη σειρά ώστε να υλοποιήσουν επιχειρησιακές διαδικασίες. Αναλύοντας τον παραπάνω ορισμό, εντοπίζονται συγκεκριμένα στοιχεία που αξίζει να αναλυθούν περαιτέρω:

- Όλες οι λειτουργίες ορίζονται ως υπηρεσίες: αυτές περιλαμβάνουν επιχειρησιακές λειτουργίες (π.χ. δημιουργία παραγγελίας), επιχειρησιακές συναλλαγές που αποτελούνται από λειτουργίες χαμηλότερου επιπέδου (π.χ. παραλαβή τραπεζικού extrait) και λειτουργίες υπηρεσιών συστήματος (π.χ. ταυτοποίηση χρήστη).
- Από την πλευρά της εφαρμογής που τις χρησιμοποιεί, οι υπηρεσίες δεν παρουσιάζουν μνήμη και μεταβάσεις μεταξύ καταστάσεων – είναι δηλαδή «stateless». Για τους χρήστες τους, οι υπηρεσίες είναι πάντα διαθέσιμες και λειτουργούν με τον ίδιο τρόπο, ανεξάρτητα από τις προηγούμενες κλήσεις τους.
- Όλες οι υπηρεσίες είναι ανεξάρτητες: Λειτουργούν ως «μαύρα κουτιά» για τους εξωτερικούς εμπλεκόμενους, οι οποίοι δεν γνωρίζουν και δεν ενδιαφέρονται για τον τρόπο με τον οποίο εκτελούν την λειτουργία τους, αλλά μόνο για το τελικό αποτέλεσμα.
- Σε γενικές γραμμές, οι διεπαφές είναι «επικλητές»: σε επίπεδο αρχιτεκτονικής είναι αδιάφορο αν είναι τοπικές (εντός του συστήματος) ή απομακρυσμένες. Επιπλέον, ο

τρόπος και οι τεχνολογίες διασύνδεσης είναι αδιάφορες. Η υπηρεσία μπορεί να βρίσκεται στην ίδια ή σε άλλη εφαρμογή ή σε άλλο πληροφοριακό σύστημα. Αυτό συμβαίνει καθώς σε μία αρχιτεκτονική τύπου SOA, η διαχείριση της κλήσης της εφαρμογής και του τρόπου επικοινωνίας δεν είναι ευθύνη της εφαρμογής που την καλεί, αλλά του συστήματος που αποτελεί το υπόβαθρο της αρχιτεκτονικής. (Channabasavaiah, Holley, & Tuggle, 2003)

Μία αρχιτεκτονική SOA αποτελείται από μία σειρά επιπέδων, τα οποία διαχωρίζουν τις επιχειρησιακές πλευρές από το λογισμικό και τις υποδομές ΤΠΕ. Σύμφωνα με τους Rosen κ.α., τα επίπεδα αυτά είναι, συνοπτικά, και από το χαμηλότερο στο υψηλότερο:

Επίπεδο πόρων και συστημάτων: Περιλαμβάνει το υλικό και λογισμικό συστήματος, βάσεις δεδομένων, παλαιότερες εφαρμογές «μονολιθικού τύπου» κτλ. Στο επίπεδο αυτό αντιστοιχούν τα «λειτουργικά δεδομένα» δηλαδή πληροφορία που υφίσταται σε βάσεις δεδομένων και εφαρμογές.

Επίπεδο ολοκλήρωσης: Το επίπεδο αυτό περιλαμβάνει το λογισμικό υποδομής της αρχιτεκτονικής SOA (το οποίο π.χ. αναλαμβάνει τη διαλειτουργικότητα των υπηρεσιών και την εκτέλεση των διαδικασιών που συνθέτουν κτλ. Στο επίπεδο αυτό υπάρχουν δεδομένα «μετασχηματισμού» (transformational data) δηλαδή δεδομένα που μεταδίδονται και μετασχηματίζονται κατά την υλοποίηση της διαλειτουργικότητας μεταξύ των στοιχείων σε ανώτερα επίπεδα. Στο επίπεδο αυτό δραστηριοποιούνται τα «ενδιάμεσα λογισμικά SOA» (SOA middleware) α οποία παρέχουν υποδομές για πληροφοριακά συστήματα βασισμένα στη SOA όπως π.χ. το λογισμικό Oracle Service Bus¹, IBM Websphere² κ.α.

Επίπεδο υπηρεσιών: Περιλαμβάνει τις υπηρεσίες και τις διεπαφές τους. Το επίπεδο αυτό επίσης περιλαμβάνει μηχανισμούς διασύνδεσης με το αμέσως προηγούμενο επίπεδο. Οι υπηρεσίες στο επίπεδο αυτό μπορούν να είναι «στοιχειώδεις» ή «σύνθετες», δηλαδή να είναι συνθέσεις άλλων υπηρεσιών. Στις SOA, οι υπηρεσίες ΤΠΕ του επιπέδου αυτού αντιστοιχούν σε στοιχειώδεις επιχειρησιακές λειτουργίες. Τα δεδομένα στο επίπεδο αυτό αφορούν σε δεδομένα που ανταλλάσσονται μεταξύ των υπηρεσιών. Το επίπεδο υπηρεσιών εξυπηρετεί το αμέσως παραπάνω επίπεδο, αυτό των επιχειρησιακών διαδικασιών.

Επίπεδο επιχειρησιακών διαδικασιών: Οι επιχειρησιακές διαδικασίες είναι μία σειρά από λειτουργίες που εκτελούνται με βάση συγκεκριμένους κανόνες (π.χ. παραλαβή και καταχώρηση αίτησης, πώληση αγαθών, πρόσληψη υπαλλήλου κτλ.). Οι διαδικασίες αυτές μπορούν να περιγράφονται με μία γλώσσα διαδικασιών (BPMN, BPEL κτλ.) και να εκτελούνται από ένα σύστημα διαχείρισης επιχειρησιακών διαδικασιών (Business Process Management System). Οι ακολουθίες αυτές απαντώνται με τον όρο «εννοχηστρώσεις» (orchestrations). Τα δεδομένα στο επίπεδο αυτό αφορούν «επιχειρησιακά έγγραφα», καθώς αντιστοιχούν σε έγγραφα της επιχειρησιακής λογικής που εξυπηρετούν (π.χ. παραγγελία, μισθοδοσία μήνα, κτλ.) (Rosen, Lublinsky, Smith, & Balcer, Applied SOA: Service Oriented Architecture and Design Strategies, 2012).

¹ <http://www.oracle.com/technetwork/middleware/service-bus/overview/index.html>

² <http://www.ibm.com/software/websphere>

Η ομάδα Open Group από την άλλη διαχωρίζει πέντε επίπεδα, με το κάθε ένα να στηρίζει το επόμενο (The Open Group, 2011):

Επιχειρησιακό επίπεδο: Περιλαμβάνει όλες τις υποδομές που υποστηρίζουν τη λειτουργία του συστήματος.

Επίπεδο συστατικών υπηρεσιών: Περιλαμβάνει την τεχνική υλοποίηση των υπηρεσιών, όπως π.χ τις εφαρμογές και δεδομένα που τις συνθέτουν.

Επίπεδο υπηρεσιών: Περιλαμβάνει τις υπηρεσίες, όπως αυτές εκφράζονται από τις διεπαφές τους και την υποστηριζόμενη λειτουργικότητα.

Επίπεδο επιχειρησιακών διαδικασιών: Περιλαμβάνει τις συνθέσεις υπηρεσιών, οι οποίες εξυπηρετούν τις επιχειρησιακές ανάγκες.

Επίπεδο καταναλωτή: Στο επίπεδο αυτό, εξωτερικοί παράγοντες (όπως π.χ. χρήστες ή άλλα συστήματα) αλληλεπιδρούν με την υπηρεσιοστραφή αρχιτεκτονική.

Η προσέγγιση της SOA, δηλαδή της σύνθεσης περίπλοκων εφαρμογών από ατομικές, αυτοπεριεχόμενες, «υπηρεσίες», έχει μία σειρά από ωφέλειες:

- Επιτυγχάνεται μείωση της επανάληψης στην ανάπτυξη λογισμικού και την αναχρησιμοποίηση υφιστάμενων λειτουργιών με αποτελεσματικό τρόπο. Καθώς οι επιχειρησιακές εφαρμογές επεκτείνονται και τροποποιούνται, η εκ νέου ανάπτυξη ήδη υπάρχουσας λειτουργικότητας είναι εξαιρετικά ασύμφορη. Η προσέγγιση της SOA επιτρέπει την ανάπτυξη της λειτουργικότητας μία μόνο φορά και στη συνέχεια τη χρήση από πολλαπλά συστήματα και σε διάφορες συνθήκες. Για παράδειγμα, σε μία επιχείρηση που κάνει καταθέσεις σε λογαριασμούς προμηθευτών, αλλά και υπαλλήλων, συμφέρει η ανάπτυξη της λειτουργίας «αποστολή εμβάσματος σε τραπεζικό λογαριασμό» μία μόνο φορά, και η χρησιμοποίησή της τόσο από τις εφαρμογές των προμηθειών, όσο και από το σύστημα μισθοδοσίας.
- Ευνοείται ο διαχωρισμός των υποδομών από τις εφαρμογές, και η χρήση των υποδομών ως αδιαφοροποίητο πόρο (commodity). Η SOA υποστηρίζει την υλοποίηση ενός συστήματος που θα διαμορφώνει ένα κοινό πλαίσιο και θα διαχειρίζεται την επικοινωνία και την ενορχήστρωση των διάφορων εφαρμογών. Έτσι, είναι δυνατόν να συντεθούν περίπλοκα συστήματα με τη χρήση των εφαρμογών ως υπηρεσιών, ανεξάρτητων από την υποδομή στο υπόβαθρο (υλικό, λογισμικό υποδομής SOA κτλ.) το οποίο μπορεί να διαχειρίζεται ανεξάρτητα.
- Επιτυγχάνεται αυξημένη ταχύτητα και μειωμένο κόστος στην ανάπτυξη νέων εφαρμογών, καθώς αυτό γίνεται δυνατό μέσω της σύνθεσης υπηρεσιών από υφιστάμενες βιβλιοθήκες υπηρεσιών.
- Γίνεται καλύτερη διαχείριση κινδύνου κατά την ανάπτυξη νέων εφαρμογών. Ο κίνδυνος κατανέμεται στις ανεξάρτητες υπηρεσίες που συνθέτουν τις εφαρμογές, στις διαδικασίες συντονισμού μεταξύ τους και τους μηχανισμούς υποβάθρου της αρχιτεκτονικής. Για παράδειγμα, μία νέα λειτουργία μπορεί να δημιουργηθεί με τη σύνθεση ήδη δοκιμασμένων υπηρεσιών, εστιάζοντας τον κίνδυνο μόνο στον καλό σχεδιασμό των διαδικασιών «ενορχήστρωσης» τους.

- Η χρήση της SOA περιλαμβάνει την αποτύπωση των επιχειρησιακών διαδικασιών και την αντιστοίχιση τους με τις διαδικασίες των πληροφοριακών συστημάτων. Οι εφαρμογές ΤΠΕ δεν είναι πλέον «μαύρα κουτιά» που απλώς παρέχουν το σύνολο της λειτουργικότητας. Πλέον, μόνο οι στοιχειώδεις, («ατομικές») υπηρεσίες είναι αδιαφανείς, και η πλήρης επιχειρησιακή λειτουργικότητα παράγεται από την σαφώς καθορισμένη σύνθεσή τους.
- Διευκολύνεται έτσι η δυνατότητα παρακολούθησης της δραστηριότητας και των λειτουργιών της επιχείρησης (συλλογή δεδομένων αποδοτικότητας, υπολογισμός δεικτών κτλ.)
- Διευκολύνεται η αναδιοργάνωση επιχειρησιακών διαδικασιών αφού είναι εύκολη η τροποποίηση και ανασύνθεση των αντίστοιχων διαδικασιών ΤΠΕ. (Channabasavaiah, Holley, & Tuggle, 2003)

3.2.2 Συστήματα διαχείρισης επιχειρησιακών διαδικασιών

Τα υπηρεσιοστραφή συστήματα, και ιδιαίτερα οι συνθέσεις υπηρεσιών, έχουν στενή σχέση με τις τεχνικές και προσεγγίσεις της «Διαχείρισης Επιχειρησιακών Διαδικασιών» (Business Process Management – BPM). Οι τεχνικές BPM βασίζονται στο ότι κάθε προϊόν, υπηρεσία και αποτέλεσμα που παράγει ένας οργανισμός είναι το αποτέλεσμα μίας σειράς δραστηριοτήτων. Οι δραστηριότητες αυτές συνθέτουν «επιχειρησιακές διαδικασίες» (business processes) που συνθέτουν τη λειτουργία της επιχείρησης. Στις σύγχρονες επιχειρήσεις και οργανισμούς, οι δραστηριότητες που συνθέτουν τις επιχειρησιακές τους διαδικασίες εκτελούνται από ανθρώπους, εξοπλισμό και συστήματα ΤΠΕ.

Συνοπτικά, μία επιχειρησιακή διαδικασία είναι ένα σύνολο δραστηριοτήτων που εκτελούνται συντονισμένα σε με σκοπό την επίτευξη ενός επιχειρησιακού στόχου, εκτελείται από έναν οργανισμό και, ενδεχομένως, αλληλοεπιδρά με επιχειρησιακές διαδικασίες άλλων οργανισμών. Η «Διαχείριση Επιχειρησιακών Διαδικασιών» περιλαμβάνει έννοιες, μεθοδολογίες και τεχνικές για τον σχεδιασμό, διοίκηση, εκτέλεση και ανάλυση των επιχειρησιακών διαδικασιών. Τέλος, ένα «Σύστημα Διαχείρισης Επιχειρησιακών Διαδικασιών» είναι ένα σύστημα λογισμικού το οποίο, με την καθοδήγηση μοντέλων και αναπαραστάσεων διαδικασιών, συντονίζει την υλοποίησή τους. (Weske, 2007).

Η εστίαση στις επιχειρησιακές διαδικασίες πηγάζει από τη θεώρηση ότι οι επιχειρησιακές διαδικασίες είναι η λειτουργία ενός οργανισμού. Έτσι, λογικά, η καλή τους εκτέλεση θα επιτρέψει στον οργανισμό την επίτευξη των στόχων του φθηνότερα, ευκολότερα και με υψηλότερη ποιότητα. Οι επιχειρησιακές διαδικασίες μπορεί να είναι

- Οι κύριες διαδικασίες του οργανισμού, οι οποίες αφορούν την κύρια λειτουργία και αποστολή του.
- Οι υποστηρικτικές διαδικασίες, οι οποίες υποστηρίζουν την εκτέλεση των κύριων διαδικασιών (π.χ. οικονομική διαχείριση και διοίκηση ανθρώπινου δυναμικού).
- Οι διοικητικές διαδικασίες που επιβλέπουν την οργάνωση, σχεδιασμό και εκτέλεση όλων των υπόλοιπων διαδικασιών, την υλοποίηση των λειτουργιών του οργανισμού και την εκπλήρωση της αποστολής του.

Η μοντελοποίηση μίας επιχειρησιακής διαδικασίας περιλαμβάνει πληροφορία από μία σειρά πεδίων. Κατά κύριο λόγο, περιλαμβάνει τη δημιουργία ενός μοντέλου της ακολουθίας των ενεργειών για την εκτέλεση της εργασίας. Ωστόσο, η ακολουθία ενεργειών δεν αρκεί για μία πλήρη επιχειρησιακή μοντελοποίηση. Αυτή προϋποθέτει την αναπαράσταση (με άμεσο ή έμμεσο τρόπο) επιπλέον πληροφορίας. Έτσι, απαιτείται αρχικά μία αναπαράσταση των εργασιών που εκτελούνται, με δεδομένα όπως η περιγραφή της συγκεκριμένης εργασίας, οι είσοδοι, οι έξοδοι, και οι πόροι που χρησιμοποιούνται για την εκτέλεσή της. Επιπλέον, περιλαμβάνεται η περιγραφή των απαιτήσεων και της ανταλλαγής πληροφορίας κατά την εκτέλεση της επιχειρησιακής διαδικασίας, όπως η ανταλλαγή μηνυμάτων και η αποθήκευση δεδομένων. Ένα μοντέλο επιχειρησιακής διαδικασίας περιλαμβάνει επίσης πληροφορία για την οργανωτική δομή του οργανισμού ή οργανισμών που εμπλέκονται, όπως π.χ. σε ποια τμήματα και υποτμήματα ανήκουν οι εργαζόμενοι που εκτελούν τη διαδικασία ή ποιοι την ελέγχουν. Τέλος, η επιχειρησιακή μοντελοποίηση περιλαμβάνει τα πληροφοριακά συστήματα που υποστηρίζουν την εκτέλεση της διαδικασίας όπως βάσεις δεδομένων, διεπαφές χρήστη, διεπαφές διαλειτουργικότητας με άλλα συστήματα κ.α.

Παραδοσιακά, μία επιχείρηση περιγράφεται με βάση τις λειτουργίες της (π.χ. πωλήσεις ή προμήθειες). Οι λειτουργίες μπορούν να αναλυθούν σε υπολειτουργίες (π.χ. οι προμήθειες σε παραγγελίες, πληρωμές και διαχείριση συμβάσεων), αλλά δεν παρέχεται πληροφορία για το πότε ενεργοποιείται η κάθε λειτουργία. Η μοντελοποίηση επιχειρησιακών διαδικασιών αναλύει τις λειτουργίες σε επιμέρους εργασίες, τις συσχετίζει και περιγράφει τη σειρά και τις προϋποθέσεις εκτέλεσής τους. Σε έναν οργανισμό, μία λειτουργία μπορεί να περιλαμβάνει πολλές διαδικασίες που εκτελούνται σειριακά ή παράλληλα, ή συνδέονται με διαδικασίες σε άλλες λειτουργίες (στην εκκίνηση, ολοκλήρωση ή μέσω ανταλλαγής μηνυμάτων).

Στη μοντελοποίηση επιχειρησιακών διαδικασιών χρησιμοποιούνται αφαιρετικές σημειογραφίες όπως η BPMN (Business Process Model and Notation (BPMN³) (White, 2008) και άλλες. Στη σημειογραφία BPMN (Business Process Model and Notation), ένα μοντέλο επιχειρησιακή διαδικασίας είναι ένας κατευθυνόμενος γράφος. Οι ακμές του γράφου αναπαριστούν τις σχέσεις μεταξύ των κόμβων του γράφου (ροή εργασιών ή ανταλλαγή πληροφορίας). Οι κόμβοι του γράφου μπορεί να είναι

- «Εργασίες»: Μοντέλα των δραστηριοτήτων της επιχειρησιακής διαδικασίας με μία είσοδο και μία έξοδο
- «Συμβάντα»: Αναπαριστούν την αλλαγή καταστάσεων μίας διαδικασίας. Περιλαμβάνει την αρχή, το τέλος και άλλα γεγονότα όπως η λήψη ενός μηνύματος, το πέρασμα του χρόνου κ.α.
- «Πύλες»: αναπαριστούν σημεία ροής ελέγχου, όπως διακλαδώσεις υπό όρους, πιθανότητες, κ.α.

Ένα «στιγμιότυπο» διαδικασίας (process instance) είναι μία διαδικασία κατά την εκτέλεσή της. Ένα στιγμιότυπο μπορεί να σχετίζεται με ένα μόνο μοντέλο διαδικασίας. Από την άλλη, ένα μοντέλο διαδικασίας μπορεί να αποτελεί το πρότυπο για απεριόριστο αριθμό στιγμιότυπων της διαδικασίας, τα οποία εκτελούνται παράλληλα. Για παράδειγμα, το

³ <http://www.bpmn.org/>

μοντέλο της επεξεργασίας μίας αίτησης είναι μοναδικό. Αντίθετα σε μία δεδομένη στιγμή, μπορεί να γίνεται η επεξεργασία πολλών δεκάδων αιτήσεων με βάση το μοντέλο αυτό, σε διάφορα στάδια εκτέλεσης, που αντιστοιχούν στα στάδια εκτέλεσης που αναπαρίστανται στο μοντέλο.

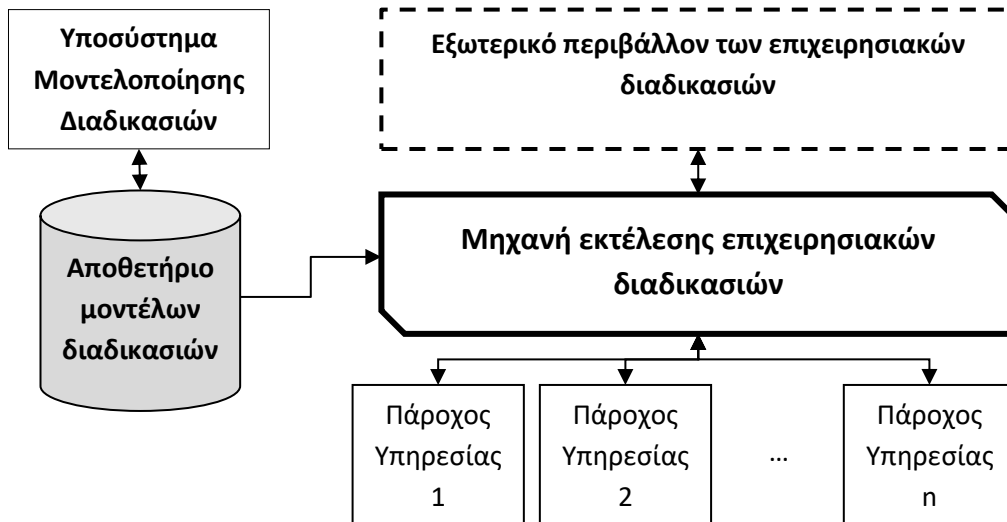
Στα πλαίσια της επιχειρησιακής μοντελοποίησης, περιλαμβάνεται και η μοντελοποίηση των επιχειρησιακών δεδομένων και της ροής τους μέσα στη διαδικασία. Περιλαμβάνει εσωτερικά δεδομένα των εργασιών, δεδομένα που τηρούνται εξωτερικά της διαδικασίας, δεδομένα που ανταλλάσσονται μεταξύ βημάτων και εργασιών, και δεδομένα που χρησιμοποιούνται για τον έλεγχο της ροής (π.χ. που ορίζουν αν εκπληρώνονται ορισμένοι επιχειρησιακοί κανόνες). Από την άλλη, η μοντελοποίηση της οργάνωσης αφορά την αναπαράσταση των οργανωσιακών μονάδων, και την συσχέτισή τους με τις εργασίες της διαδικασίας. Η συσχέτιση μπορεί να αφορά έννοιες όπως η ανάθεση μίας εργασίας, ο έλεγχος και έγκριση ενός αποτελέσματος κ.α.

Τα πληροφοριακά συστήματα διαχείρισης διαδικασιών χειρίζονται την εκτέλεση επιχειρησιακών διαδικασιών, βασισμένα στην αφαιρετική μοντελοποίησή τους και τις περιγραφές της αλληλεπίδρασης με άλλες διαδικασίες ή συστήματα. Η ανάλυση και μοντελοποίηση των επιχειρησιακών διαδικασιών επιτρέπει την αντιστοίχιση των εργασιών που τις συνθέτουν με εφαρμογές και υπηρεσίες λογισμικού. Οι μηχανές εκτέλεσης επιχειρησιακών διαδικασιών (Business Process Engines) ή ροής εργασιών (Workflow Engines) είναι εφαρμογές που εκτελούν τις εκτελέσιμες επιχειρησιακές διαδικασίες καλώντας τις αντίστοιχες εφαρμογές (συμπεριλαμβανομένων υπηρεσιών SOA), εφαρμόζοντας επιχειρησιακούς κανόνες, ελέγχοντας στοιχεία αλληλεπίδρασης χρήστη κ.α. Το αποτέλεσμα είναι ότι η υποδομή πληροφορικής του οργανισμού αντικατοπτρίζει και ευθυγραμμίζεται με το επιχειρησιακές διαδικασίες που καλείται να υποστηρίξει.

Συνοπτικά, ένα τέτοιο σύστημα διαχείρισης επιχειρησιακών διαδικασιών αποτελείται από το ακόλουθα μέρη:

- Υποσύστημα μοντελοποίησης διαδικασιών: εργαλεία και περιβάλλοντα για τη δημιουργία μοντέλων, τον ορισμό των δραστηριοτήτων και της ροής εργασιών κ.α.
- Περιβάλλον επιχειρησιακών διαδικασιών: είναι ο «εξωτερικός κόσμος» για τη διαδικασία, ο οποίος προσφέρει το έναυσμα για τη δημιουργία «στιγμιότυπων» των διαδικασιών (process instances), και με τον οποίο η διαδικασία αλληλοεπιδρά.
- Αποθετήριο μοντέλων επιχειρησιακών διαδικασιών: τηρεί τα μοντέλα που δημιουργεί και επεξεργάζεται το υποσύστημα μοντελοποίησης.
- «Μηχανή εκτέλεσης» επιχειρησιακών διαδικασιών (process engine): είναι το κεντρικό στοιχείο του συστήματος διαχείρισης επιχειρησιακών διαδικασιών. Η «μηχανή εκτέλεσης» είναι υπεύθυνη για την δημιουργία των στιγμιότυπων (instantiation) και για την εκτέλεση των διαδικασιών, με βάση τα μοντέλα τους. Καθώς εκτελεί τη διαδικασία, επικοινωνεί με εξωτερικές οντότητες που παρέχουν λειτουργικότητα, όπως διεπαφές χρήστη, βάσεις δεδομένων, συστήματα υπολογισμών κ.α.
- Πάροχοι υπηρεσιών: Πόροι (εξωτερικοί της μηχανής εκτέλεσης) που πραγματοποιούν επιχειρησιακές δραστηριότητες. Μπορούν να περιλαμβάνουν

ανθρώπους, εξοπλισμό, web services (π.χ. σε ένα σύστημα βασισμένο σε αρχές SOA) κ.α.



Σχήμα 2: Αρχιτεκτονική συστήματος διαχείρισης επιχειρησιακών διαδικασιών (Weske, 2007)

Ο κύριος οδηγός των δραστηριοτήτων διαχείρισης διαδικασιών είναι η επίτευξη «ευελιξίας» του οργανισμού. Η μοντελοποίηση επιτρέπει την αποτύπωση της διαδικασίας, και τη βελτίωσή της μέσω ποιοτικών (π.χ. συζήτησης με τους εμπλεκόμενους) και ποσοτικών (π.χ. προσομοίωσης και μαθηματικής βελτιστοποίησης) μεθόδων. Η αφαιρετικότητα της μοντελοποίησης επιτρέπει την αποσύνδεση των βημάτων της διαδικασίας από συγκεκριμένους πόρους (π.χ. ανθρώπους ή συστήματα), ανοίγοντας τον δρόμο για μεγαλύτερη ευελιξία στη συμμετοχή τους. Η διαχείριση διαδικασιών αποτελεί επίσης παράγοντα για την επίτευξη καινοτομικότητας, αφού η ευελιξία που προσφέρεται επιτρέπει την εύκολη εφαρμογή νέων τρόπων εργασίας. Ακόμη, η χρήση συστηματικών μεθόδων μοντελοποίησης και διαχείρισης διαδικασιών επιτρέπουν τον ορισμό και παρακολούθηση συγκεκριμένων δεικτών απόδοσης (όπως χρόνος, ποιότητα, κόστος κ.α.), και συνεπώς, τη διοίκηση και βελτίωση της λειτουργίας του οργανισμού σε στέρεες βάσεις (Kirchner, 2017).

3.2.3 Συμπεράσματα

Οι υπηρεσιοστραφείς αρχιτεκτονικές συστημάτων και οι τεχνολογίες BPM είναι κατάλληλες για το περιβάλλον ΤΠΕ της υπηρεσιοποιημένης βιομηχανίας.

- Η «υπηρεσιοστρέφεια» και η διαχείριση επιχειρησιακών διαδικασιών είναι ώριμες, ευρέως χρησιμοποιούμενες προσεγγίσεις. Υπάρχει σημαντική εμπειρία από πρακτικές εφαρμογές σε επιχειρησιακά συστήματα, καθώς και σημαντικός αριθμός εργαλείων και μεθοδολογιών.
- Τα υποτμήματα των υπηρεσιοστραφών συστημάτων διαθέτουν «χαλαρή διασύνδεση» (loose coupling), με σκοπό την αναχρησιμοποίηση τους σε πολλούς συνδυασμούς. Όταν οι απαραίτητες «ατομικές» υπηρεσίες γίνουν διαθέσιμες (π.χ. αφού υλοποιηθούν από τους προγραμματιστές ή προσφερθούν από τρίτους) μπορούν να συνδυαστούν για να δημιουργήσουν νέα λειτουργικότητα. Από την άλλη, οι τεχνικές BPM προσφέρουν γλώσσες και πρότυπα μοντελοποίησης ώστε να

εκφραστούν επιχειρησιακές διαδικασίες ως ακολουθίες εργασιών και ψηφιακών υπηρεσιών, καθιστώντας τις δύο προσεγγίσεις συμπληρωματικές. Έτσι, είναι εφικτή η δημιουργία τροποποιημένων εκδόσεων υφιστάμενων συστημάτων αλλάζοντας τη δομή ή τις υπηρεσίες που τις αποτελούν, προσφέροντας έτσι έναν υψηλό βαθμό ευελιξίας.

- Οι τεχνολογίες SOA και BPM στοχεύουν στην βελτίωση της ευθυγράμμισης των επιχειρησιακών λειτουργιών με το λογισμικό που τις υποστηρίζει. Ο τρόπος που το επιτυγχάνουν αυτό είναι μέσω της αντιστοίχισης των στοιχείων του επιχειρησιακού πεδίου (εμπλεκόμενοι διαδικασίες, οντότητες, σχέσεις, μηνύματα κ.α.) με τα στοιχεία του λογισμικού. Επιπλέον, με αυτόν τον τρόπο διευκολύνεται η διαχείριση και η αναδιαμόρφωση, όποτε απαιτηθεί, του συστήματος παροχής επιχειρησιακών υπηρεσιών.
- Τα συστήματα SOA και BPM ενθαρρύνουν τον εννοιολογικό διαχωρισμό και την αφαιρετική αναπαράσταση των στοιχείων τους. Η γνώση της εσωτερικής υλοποίησης των υπηρεσιών λογισμικού δεν έχει τόση σημασία, όσο η γνώση της λειτουργικότητας και των διεπαφών τους. Πράγματι, μόνο τα δύο τελευταία στοιχεία αρκούν για την χρήση και την επικοινωνία με μία υπηρεσία λογισμικού. Οι συνθέσεις υπηρεσιών και οι εκτελέσιμες επιχειρησιακές διαδικασίες κρύβουν την πολυπλοκότητα των υποδομών ΤΠΕ που τις υλοποιούν.
- Τέλος, για την ανάπτυξη συστημάτων SOA και BPM, καθώς και για τον συνδυασμό τους, χρησιμοποιούνται συχνά προσεγγίσεις μηχανικής λογισμικού με μοντελοποίηση (model-driven software engineering). Αυτό οφείλεται στην έμφαση των τεχνολογιών SOA και BPM στην αφαιρετική απεικόνιση επιχειρησιακών λειτουργιών και συστημάτων ΤΠΕ. Τη στιγμή αυτή, οι εφαρμογές των προσεγγίσεων μοντελοποίησης στα πεδία του SOA και BPM αποτελούν αντικείμενο σημαντικού ερευνητικού ενδιαφέροντος (Ameller, et al., 2015).

3.3 Μεθοδολογίες και εργαλεία ανάπτυξης λογισμικού

3.3.1 Ανάπτυξη λογισμικού με χρήση μοντελοποίησης: η μεθοδολογία MDA (Model-Driven Architecture)

Από την πλευρά της, η επιστήμη υπολογιστών έχει ήδη προσφέρει σημαντικό αριθμό από μεθοδολογίες και τεχνικές οι οποίες επιχειρούν να μεταφέρουν τις επιχειρησιακές απαιτήσεις στη διαδικασία ανάπτυξης. Πράγματι, καθώς το νέο λογισμικό υπάρχει ως μέσο και υπόβαθρο για την εκτέλεση συγκεκριμένων επιχειρησιακών διαδικασιών και στόχων, η ανεπιτυχής μεταφορά των επιχειρησιακών απαιτήσεων στη διαδικασία ανάπτυξης μπορεί να αποτελέσει σημαντικό πρόβλημα στην επίτευξη των στόχων του.

Οι δυσκολίες εντοπίζονται στη μετάφραση των περιγραφών υψηλού επιπέδου των διαδικασιών σε περιγραφές του απαιτούμενου λογισμικού, στα προβλήματα της ενσωμάτωσης τεχνολογικών παραμέτρων σε επιχειρησιακές διαδικασίες, την επικοινωνία και συντονισμό των εμπλεκόμενων στελεχών κτλ.

Έτσι, απαιτείται η συστηματικοποίηση της ανάπτυξης λογισμικού με τέτοιο τρόπο ώστε να διασφαλίζεται η ορθότητα των παραγόμενων αποτελεσμάτων σε σχέση με τους απώτερους

επιδιωκόμενους στόχους τους, τηρώντας εύλογους περιορισμούς χρόνου, κόστους και προσπάθειας.

Έχει προταθεί σημαντικός αριθμός από μεθοδολογίες και εργαλεία για την επίτευξη του σκοπού αυτού. Οι τεχνικές αυτές βρίσκονται σε διάφορα στάδια θεωρητικής διερεύνησης και πρακτικής εφαρμογής στη βιομηχανία και στην έρευνα. Πολλές χαρακτηρίζονται από έντονο τεχνολογικό προσανατολισμό ενώ άλλες δίνουν έμφαση σε οργανωτικές πλευρές, όπως διαδικασίες ανάπτυξης, οργάνωση και λειτουργία ομάδων κτλ. Τη στιγμή αυτή εφαρμόζονται μοντέλα τύπου «καταρράκτη» (waterfall models), επαναληπτικά και σπειροειδή (iterative/ spiral development), η οικογένεια μεθοδολογιών Agile, η μεθοδολογία rational Unified Process, κτλ. Σε περισσότερο τεχνικό επίπεδο, εφαρμόζονται μεθοδολογίες όπως ο δομημένος προγραμματισμός (structured programming), ο αντικειμενοστραφής προγραμματισμός (object-oriented programming), τα διαγράμματα ροής, κ.α. Ωστόσο, παρά τη σημαντική πρόοδο που έχει γίνει, το πρόβλημα παραμένει ανοικτό και αποτελεί πεδίο σημαντικού ερευνητικού ενδιαφέροντος.

Στα πλαίσια των τεχνικών αυτών περιλαμβάνεται η εφαρμογή μεθόδων μοντελοποίησης για τη διαχείριση της πολυπλοκότητας του πραγματικού κόσμου και των επιχειρησιακών απαιτήσεων καθώς και της αναπαράστασης των προδιαγραφών, στοιχείων και συμπεριφορών συστημάτων λογισμικού.

Ένα μοντέλο είναι μία αφαιρετική αναπαράσταση της πραγματικότητας (υφιστάμενης ή δυνητικής), με σκοπό τη μελέτη και την κατανόησή της. Ένα μοντέλο μπορεί να περιέχει γνώση για την δομή, τα συστατικά, τις ιδιότητες, την λειτουργία και τις σχέσεις των συστατικών μερών του εξωτερικού κόσμου. Έτσι, επιτρέπει την συστηματική αναπαράσταση της πραγματικότητας μέσα στην (περιορισμένη) ανθρώπινη διάνοηση, την επεξεργασία της αναπαράστασης αυτής, την αποθήκευση και την μετάδοσή της. Η «μοντελοποίηση» είναι η διαδικασία της κατασκευής ενός μοντέλου ως αναπαράσταση ενός συστήματος, μίας διαδικασίας ή ενός φαινομένου.

Η μοντελοποίηση γίνεται διαμέσου της κατάλληλης «γλώσσας μοντελοποίησης», δηλαδή της σημειογραφίας με την οποία αποτυπώνονται με αφαιρετικό τα στοιχεία του μοντέλου και οι μεταξύ τους σχέσεις.

Η κατασκευή ενός μοντέλου προϋποθέτει μία «οντολογία». Κάθε εσωτερικά συνεπής αναπαράσταση του φυσικού κόσμου, πρέπει να έχει μία σαφή τοποθέτηση για το είδος, τις κατηγορίες, τις σχέσεις και τις πιθανές ιδιότητες των πραγμάτων που θα περιγράψει. Μία οντολογία είναι ο προσδιορισμός μίας τυπικής, περιεκτικής και σαφούς «εννοιολογίας», δηλαδή ο προσδιορισμός ενός συστήματος εννοιών για την αναπαράσταση της πραγματικότητας. Μία οντολογία παρέχει ένα κοινά αποδεκτό (από τους δημιουργούς και τους χρήστες του μοντέλου) «κατάλογο» του συνόλου των εννοιών που περιλαμβάνει ένα πεδίο της πραγματικότητας ή θεωρούνται κρίσιμες για το πρόβλημα που πρέπει να αντιμετωπίσει, και το λεξιλόγιο για την περιγραφή τους.

Μία μορφή οντολογίας με υψηλό βαθμό τυπικότητας και αναλυτικότητας είναι το «μεταμοντέλο». Ένα μεταμοντέλο είναι μία λεπτομερής περιγραφή του τρόπου με τον οποίο κατασκευάζεται ένα μοντέλο για ένα συγκεκριμένο πεδίο της πραγματικότητας. Πιο

αναλυτικά, ένα μεταμοντέλο καθορίζει, με τον πλέον τυπικό και αναλυτικό τρόπο, τα στοιχεία, σχέσεις, ιεραρχίες και ιδιότητες που μπορεί να περιλαμβάνει ένα δεδομένο μοντέλο, και τους κανόνες που πρέπει να ακολουθεί για να θεωρείται έγκυρο. Για παράδειγμα, ένα οργανόγραμμα μίας επιχείρησης είναι ένα μοντέλο της δομής της, που περιλαμβάνει «τμήματα», «εργαζόμενους» και «προϊσταμένους τμημάτων». Το μεταμοντέλο του οργανογράμματος καθορίζει ότι ένα οργανόγραμμα περιλαμβάνει τα στοιχεία «τμήμα», «εργαζόμενος», «προϊστάμενος τμήματος», ότι το ο «εργαζόμενος» συνδέεται με ένα «τμήμα», ότι ο «προϊστάμενος» είναι επίσης συνδεδεμένος με ένα, το πολύ, τμήμα, και ότι ο «προϊστάμενος» είναι «εργαζόμενος», αλλά δεν είναι ο κάθε «εργαζόμενος», «προϊστάμενος». Έτσι θέτει τους κανόνες για τη δημιουργία οποιουδήποτε έγκυρου και συνεπούς οργανογράμματος με βάση τους κανόνες αυτούς.

Στην ανάπτυξη λογισμικού, η χρήση μοντέλων αποτελεί το κοινό χαρακτηριστικό μίας σειράς μεθοδολογιών με την κοινή ονομασία «Model – Driven Engineering» (MDE – «Μηχανική Οδηγούμενη από Μοντέλα»). Η MDE στηρίζεται στην αναλυτική περιγραφή του «τομέα εφαρμογής» (domain) του νέου λογισμικού με τη χρήση συστηματικών μεθόδων μοντελοποίησης. Το «Μοντέλο Τομέα» ή «Μοντέλο Πεδίου» (Domain Model) επιτρέπει τη συμφωνία όλων των εμπλεκόμενων σχετικά με τις έννοιες, τα στοιχεία, τις ιδιότητες και τις σχέσεις στο πραγματικό πρόβλημα. Έτσι, μπορεί να χρησιμοποιηθεί σαν είσοδος σε διαδικασίες σχεδιασμού και ανάπτυξης λογισμικού.

Καθώς οι απεικονίσεις μέσω μοντελοποίησης μπορούν να υπόκεινται σε αυστηρούς κανόνες, η MDE σχετίζεται και με το πεδίο των «τυπικών μεθόδων» (formal methods). Οι «τυπικές μέθοδοι» αποσκοπούν στην λύση του προβλήματος της ανάπτυξης λογισμικού από επιχειρησιακές προδιαγραφές, εφαρμόζοντας μαθηματικές μεθόδους για την παραγωγή, πιστοποίηση και δοκιμή λογισμικού π.χ. μέσω του μετασχηματισμού μοντέλων από μία μορφή σε μία άλλη.

Πάνω στις βασικές ιδέες της MDE έχουν βασιστεί πολλές μεθοδολογίες και συστήματα ανάπτυξης λογισμικού. Μια από τις σημαντικότερες πρωτοβουλίες για τη δημιουργία μίας ολοκληρωμένης προσέγγισης είναι η «Model-Driven Architecture» (Αρχιτεκτονική Οδηγούμενη Από Μοντέλα – MDA). Η MDA είναι μία πρωτοβουλία της ένωσης Object Management Group (OMG), και έχει σαν στόχο τη δημιουργία ενός πλαισίου για τη δόμηση και μετασχηματισμό των προδιαγραφών που οδηγούν την ανάπτυξη λογισμικού. Σημειώνεται ότι ο όρος «αρχιτεκτονική» αναφέρεται στο σύνολο προτύπων και διαδικασιών που σχηματίζουν την MDA, και όχι στην αρχιτεκτονική συστημάτων. (Mukerji & Miller, 2003)

Η MDA προτάθηκε από το OMG το 2001, ως ένα πλαίσιο που ενσωμάτωνε μία σειρά από υφιστάμενα πρότυπα. Το πλαίσιο αυτό είχε σαν στόχο την παροχή μεθόδων για την υποστήριξη της διαλειτουργικότητας, την ανάπτυξη νέου λογισμικού και κώδικα, προσομοίωση και πιστοποίηση νέων συστημάτων, και άλλων ωφελειών. Η MDA έχει εξελιχθεί μέχρι σήμερα, με την 2^η έκδοση του πλαισίου να δημοσιεύεται το 2014.

Η MDA περιλαμβάνει μία σειρά από προτεινόμενα εργαλεία, πρότυπα και τεχνικές, οι οποίες μπορούν να αποτελέσουν τη βάση μεθοδολογιών ανάπτυξης βασισμένων επάνω της. Τα κυριότερα από αυτά, και τα οποία έχουν άμεση συσχέτιση με την ανάπτυξη λογισμικού, είναι:

- Unified Modelling Language – UML⁴: είναι ένα σύνολο από γλώσσες μοντελοποίησης, οι οποίες αποσκοπούν στην αποτύπωση διαφόρων πλευρών περίπλοκων συστημάτων. Οι πρώτες εκδόσεις της UML χρονολογούν από τα μέσα της δεκαετίας του 1990, αλλά στηρίζεται σε παλαιότερες μορφές μοντελοποίησης. Τη στιγμή αυτή βρίσκεται στη 2^η έκδοσή της, και συνεχίζει να εξελίσσεται. Περιλαμβάνει δομικά διαγράμματα (κλάσεις, συστατικά, αντικείμενα κτλ.) και διαγράμματα συμπεριφοράς (δραστηριότητες, επικοινωνία, περιπτώσεις χρήσης κτλ.). Η UML βασίζεται σε καλά προσδιορισμένα μετα-μοντέλα.
- Meta-Object Facility – MOF⁵: Είναι ένα τυποποιημένο πρότυπο για τη δημιουργία και καθορισμό μετα-μοντέλων. Ως εκ τούτου, μπορεί να χαρακτηριστεί ως μετα- μετα-μοντέλο. Το μετα-μοντέλο της UML βασίζεται στο MOF.
- XML Metadata Interchange - XMI⁶: Πρότυπο βασισμένο στην XML με στόχο την ανταλλαγή μεταδεδομένων ανάμεσα σε εργαλεία βασισμένα στην UML και σε αποθετήρια βασισμένα στο MOF. Η XMI χρησιμοποιείται και για την ανταλλαγή μεταδεδομένων κατά τη διαδικασία ανάπτυξης λογισμικού.
- Common Warehouse Metamodel - CWM⁷: Ένα μετα-μοντέλο για διαφόρους τύπους αποθετηρίων μεταδεδομένων, το οποίο βασίζεται στη UML, το MOF και το XMI.
- Query, View, Transformation Standard - QVT⁸: ένα πρότυπο για γλώσσες «μετασχηματισμού μοντέλων». Οι γλώσσες αυτές επιτρέπουν τον μετασχηματισμό μοντέλων από μία μορφή μοντελοποίησης σε μία άλλη (model-to-model), εφόσον ακολουθεί κανόνες που θέτει το MOF 2.0. (Δεν περιλαμβάνονται λ.χ. μετασχηματισμοί τύπου model-to-text)
- MOF – Model – to – text - MOFM2T⁹: αποτελεί πρότυπο της MDA για τον μετασχηματισμό μοντέλων βασισμένων στο MOF 2.0 προς κείμενο, π.χ. προς πηγαίο κώδικα.

Μία από τις κεντρικές αρχές της MDA είναι η δημιουργία σαφών διακρίσεων ανάμεσα στις περισσότερο αφαιρετικές (επιχειρησιακές – business- oriented) απεικονίσεις ενός συστήματος και στις περισσότερο τεχνικές απεικονίσεις. Ο διαχωρισμός αυτός είναι απαραίτητος για την δυνατότητα μετακίνησης των εφαρμογών από μία τεχνολογία σε άλλη. Για τον λόγο αυτό, η MDA διακρίνει τρεις διαφορετικές οπτικές γωνίες για τη μοντελοποίηση ενός συστήματος (France & Rumpe, 2007):

- Computation – Independent Model (Μοντέλο Ανεξάρτητο Υπολογισμών): Το μοντέλο (ή μοντέλα) στο επίπεδο αυτό εστιάζουν στο περιβάλλον μέσα στο οποίο θα λειτουργήσει το υπό μελέτη σύστημα. Το CIM ανταποκρίνεται ως ένα βαθμό στην έννοια του «μοντέλου τομέα» (domain model), αλλά το ξεπερνά, καθώς αναπαριστά και τη συμπεριφορά του συστήματος. Το CIM δεν περιέχει αναφορές σε στοιχεία και προδιαγραφές σχετικές με ΤΠΕ. Στη διαδικασία της MDA, οι επιχειρησιακές

⁴ <http://www.uml.org/>

⁵ <http://www.omg.org/mof/>

⁶ <http://www.omg.org/spec/XMI/>

⁷ <http://www.omg.org/cwm/>

⁸ <http://www.omg.org/spec/QVT/>

⁹ <http://www.omg.org/spec/MOFM2T/1.0/>

απαιτήσεις του CIM πρέπει να αντανακλώνται και να συνδέονται με τα στοιχεία των επόμενων μοντέλων (PIM και PSM).

- Platform – Independent Model (Μοντέλο Ανεξάρτητο Πλατφόρμας): Η «πλατφόρμα» είναι, σύμφωνα με την MDA, «ένα σύνολο από υποσυστήματα και τεχνολογίες που παρέχουν ένα ενιαίο σύνολο λειτουργικότητας μέσω διεπαφών και τρόπων χρήσης». Το μοντέλο (ή μοντέλα) στο επίπεδο αυτό αφορούν σε χαρακτηριστικά του συστήματος που είναι ανεξάρτητα της τεχνολογικής βάσης του τελικού συστήματος (π.χ. γλώσσα προγραμματισμού, περιβάλλον εκτέλεσης κτλ.), αλλά αφορούν στοιχεία ΤΠΕ, όπως π.χ. δεδομένα.
- Platform – Specific Model (Μοντέλο Συγκεκριμένης Πλατφόρμας): το μοντέλο (ή μοντέλα) που προκύπτει από την ενσωμάτωση στοιχείων που εξαρτώνται από την τεχνολογική πλατφόρμα στο μοντέλο PIM. Σε περίπτωση που το PSM δεν περιλαμβάνει όλες τις απαραίτητες πληροφορίες για τη δημιουργία του συστήματος, τότε θεωρείται «αφηρημένο» και εξαρτάται από επιπλέον πληροφορίες ή στοιχεία που περιέχουν τις πληροφορίες αυτές.

Η διαδικασία ανάπτυξης λογισμικού στηρίζεται στον «κάθετο» μετασχηματισμό των μοντέλων αυτών από την μία μορφή στην άλλη. Συγκεκριμένα, ένα μοντέλο επιπέδου CIM μετασχηματίζεται σε ένα μοντέλο PIM. Στη συνέχεια, το μοντέλο επιπέδου PIM μετασχηματίζεται σε μοντέλο επιπέδου PSM. Ο μετασχηματισμός αυτός λαμβάνει χώρα με τη βοήθεια πληροφοριών σχετικά με την τεχνική πλατφόρμα – στόχο, το οποίο περιέχεται σε ένα «μοντέλο πλατφόρμας» (Platform Model). Με τη σειρά του, το μοντέλο PSM περιέχει αρκετή πληροφορία ώστε να οδηγήσει (αυτόματα, χειροκίνητα ή με κάποιο συνδυασμό μεθόδων) την κατασκευή του τελικού συστήματος. Η δημιουργία του τελικού συστήματος μπορεί π.χ. να περιλαμβάνει την αυτόματη παραγωγή πηγαίου κώδικα βάσει του μοντέλου PSM.

Η MDA δεν προσδιορίζει με ακρίβεια τους μηχανισμούς με τους οποίους θα γίνονται οι μετασχηματισμοί των μοντέλων. Η εφαρμογή συγκεκριμένων τεχνικών εξαρτάται από το πεδίο εφαρμογής και τις ιδιαίτερες συνθήκες του προβλήματος. Έτσι, έχουν προταθεί και εξετάζονται πολλαπλές μέθοδοι μοντελοποίησης και μετασχηματισμού μοντέλων σύμφωνες με τις αρχές της MDA. (Scheidgen, 2006; Truyen, *The Fast Guide to Model Driven Architecture* The Basics of Model Driven Architecture, 2006; Jilani, Usman, & Halim, 2010; Kardoš & Drozdová, 2010; De Castro, Marcos, & Vara, 2011).

Η χρήση της MDA για την ανάπτυξη λογισμικού έχει μία σειρά από πλεονεκτήματα και μειονεκτήματα (Sacevski & Veseli, 2007). Κάποια πλεονεκτήματα είναι τα ακόλουθα:

- Η χρήση αφαιρετικών αναπαραστάσεων με αυστηρώς καθορισμένα μοντέλα, επιτρέπει τη διαχείριση της πολυπλοκότητας από τους εμπλεκόμενους στη διαδικασία ανάπτυξης.
- Η MDA ενσωματώνει την αποτύπωση των επιχειρησιακών απαιτήσεων στη διαδικασία ανάπτυξης λογισμικού, και προδιαγράφει ένα πλαίσιο για τη μεταφορά τους σε τεχνικές απαιτήσεις και περιγραφές.
- Η αποτύπωση των προδιαγραφών του λογισμικού σε διάφορα στάδια εξειδίκευσης και τεχνικής προσαρμογής βελτιώνει τη διαχείριση του κύκλου ζωής ανάπτυξης

λογισμικού, καθώς αυξάνει τη συνέπεια μεταξύ των απεικονίσεων ενός συστήματος, διευκολύνει τη διαχείριση αλλαγών κτλ.

- Το πλαίσιο της MDA ενθαρρύνει τη χρήση αυτόματων ή ημι-αυτόματων διαδικασιών μετασχηματισμού μοντέλων και παραγωγής κώδικα.
- Αυξάνεται η «μεταφερισιμότητα» (portability) του λογισμικού, καθώς περιλαμβάνει συστηματικές αναπαραστάσεις του συστήματος, οι οποίες όμως είναι ανεξάρτητες από την τελική τεχνολογική πλατφόρμα (μοντέλα επιπέδου CIM - PIM).
- Βελτιώνει τη διαλειτουργικότητα μεταξύ συστημάτων, αφού επιτρέπει την συμφωνία στις επιχειρησιακές, (και σε μεγάλο βαθμό τεχνικές) πλευρές συστημάτων που βασίζονται σε άλλες τεχνολογίες.
- Τα παραπάνω συνεισφέρουν στη μείωση του κόστους και χρόνου ανάπτυξης, ενώ βελτιώνουν την ποιότητα του παραγόμενου λογισμικού.

Από την άλλη, ορισμένα μειονεκτήματα και κίνδυνοι στη χρήση της MDA είναι τα παρακάτω:

- Τα πολλαπλά ενδιάμεσα στάδια και προϊόντα (μοντέλα πολλών επιπέδων) δημιουργούν επιπλέον φόρτο εργασίας και ανάγκη για προσεκτικό έλεγχο της συνέπειας μεταξύ τους.
- Μία αλλαγή σε ένα μοντέλο μπορεί να επηρεάζει και άλλες αναπαραστάσεις του συστήματος σε διάφορα επίπεδα, άρα απαιτεί την προσαρμογή τους ώστε να διατηρηθεί η συνέπεια.
- Η διαδικασία ανάπτυξης που προδιαγράφεται από την MDA είναι, σε μεγάλο βαθμό, «κάθετη», δηλαδή κινείται με μονόδρομο τρόπο από υψηλότερα σε χαμηλότερα επίπεδα αφαίρεσης. Αλλαγές σε αργότερα στάδια του κύκλου ανάπτυξης προϋποθέτουν αλλαγές σε ενδιάμεσες αναπαραστάσεις χαμηλότερου επιπέδου. Αυτές με τη σειρά τους μπορεί να υποχρεώνουν σε αλλαγές στις αναπαραστάσεις υψηλότερου επιπέδου στις οποίες βασίζονται (π.χ. όταν αφορούν την αρχιτεκτονική του συστήματος).
- Υπάρχουν ακόμα περιορισμοί στην αυτοματοποίηση των μετασχηματισμών μοντέλων και της παραγωγής του τελικού κώδικα.
- Η MDA δεν παρέχει τυποποιημένες γλώσσες μοντελοποίησης και μετασχηματισμούς. Συνεπώς, απαιτείται εκπαίδευση των εμπλεκόμενων στα πρότυπα, διαδικασίες, μετασχηματισμούς και γλώσσες μοντελοποίησης που χρησιμοποιούνται στη συγκεκριμένη εφαρμογή της μεθόδου.

3.3.2 Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης λογισμικού

Τα Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης (Integrated Development Environments – IDEs) είναι εφαρμογές που αποσκοπούν στην υποστήριξη της ανάπτυξης λογισμικού μέσω της παροχής ενός ενοποιημένου περιβάλλοντος εργασίας. Το ενοποιημένο αυτό περιβάλλον μπορεί να περιέχει εργαλεία για τη σύνταξη κώδικα, δοκιμές και την αναζήτηση σφαλμάτων, την δημιουργία εκτελέσιμων αρχείων, την διαχείριση πολλαπλών αρχείων και πόρων (όπως μοντέλα, κώδικας, τεκμηρίωση, βιβλιοθήκες κτλ.), τη διαχείριση εκδόσεων λογισμικού, τη συνεργασία με άλλους χρήστες κ.α.

Υπό άλλες συνθήκες, τα εργαλεία αυτά θα ήταν διάσπαρτα και ο προγραμματιστής θα χρειαζόταν να τα συλλέξει, εγκαταστήσει, παραμετροποιήσει και χρησιμοποιήσει σαν

ξεχωριστές μονάδες. Η διαδικασία αυτή μπορεί να είναι χρονοβόρα, να εισάγει τεχνικά προβλήματα λόγω π.χ. ασυμβατότητας και να αυξάνει την προσπάθεια για διαχείριση. Έτσι, η ενοποίηση των εργαλείων σε ένα ενιαίο περιβάλλον εξυπηρετεί την βελτίωση την παραγωγικότητας του χρήστη και την ποιότητα του παραγόμενου κώδικα.

Αξίζει να σημειωθεί ότι η διάκριση ανάμεσα στα IDE και στα ξεχωριστά εργαλεία ανάπτυξης λογισμικού δεν είναι εντελώς ξεκάθαρη. Τα περιβάλλοντα ανάπτυξης μπορούν να τοποθετηθούν σε ένα φάσμα, στο ένα άκρο του οποίου βρίσκονται IDE με τον μέγιστο βαθμό ενσωματωμένης λειτουργικότητας, με το άλλο άκρο να καταλαμβάνεται από χαλαρά συνδεδεμένες ομάδες εργαλείων. Ανάμεσα στα δύο άκρα μπορούν να τοποθετηθούν ενδιάμεσες πλατφόρμες και συνθέσεις. Αυτές μπορεί να περιλαμβάνουν π.χ. τις «αλυσίδες εργαλείων» (toolchains) οι οποίες αποτελούνται από ξεχωριστά εργαλεία που ενσωματώνονται σε μία τυποποιημένη ροή εργασίας, ή IDE τα οποία προσαρμόζονται σε συγκεκριμένες ανάγκες (π.χ. γλώσσες, ροές εργασίας, άλλη λειτουργικότητα) με τη χρήση «πρόσθετων» (plugins).

Οι Yang και Han θεωρούν ότι, καθώς ένα περιβάλλον ανάπτυξης λογισμικού έχει σαν κύριο στόχο την υποστήριξη της παραγωγικότητας του προγραμματιστή, η επιτυχία ενός IDE βασίζεται στη βελτίωση δύο ειδών διεπαφών: τη διεπαφή χρήστη – εργαλείων και τις διεπαφές μεταξύ των λειτουργιών που ενσωματώνει. Η βελτίωση της παραγωγικότητας συνδέεται με τη μείωση καθυστερήσεων από την πλευρά του χρήστη (π.χ. την εξάλειψη της ανάγκης για το άνοιγμα του ίδιου αρχείου με πολλαπλές εφαρμογές) και από την πλευρά των εργαλείων (π.χ. με τον καλύτερο συντονισμό των λειτουργιών μεταξύ τους). Επιπλέον, οι ερευνητές προσθέτουν σαν ισότιμο παράγοντα τη δυνατότητα δημιουργίας εξειδικευμένων συνδυασμών λειτουργιών, προσαρμοσμένων σε συγκεκριμένες ροές εργασίας και στόχους. (Yang & Han, 1996)

Τη στιγμή αυτή, οι ωφέλειες των IDE έχουν οδηγήσει σε σημαντική ποικιλία πρακτικών εφαρμογών, ενώ η τεχνολογία τους έχει φτάσει σε υψηλό βαθμό ωριμότητας. Τα IDE μπορούν να συνδέονται με συγκεκριμένες γλώσσες προγραμματισμού και τεχνολογίες (π.χ. το jDeveloper¹⁰ για τη γλώσσα Java ή το Xcode για πλατφόρμες iOS¹¹), ή να παρέχουν ένα πλαίσιο προσαρμοσμένο σε διάφορες τεχνικές πλατφόρμες ή μεθοδολογίες ανάπτυξης (π.χ. το πακέτο Eclipse¹² και το Microsoft Visual Studio¹³). Η ποικιλία εκτείνεται και στους τρόπους διάθεσης, αφού υπάρχουν παράλληλα εμπορικά πακέτα, αλλά και εφαρμογές ανοικτού κώδικα για τις περισσότερες γλώσσες και πλατφόρμες.

Τα IDEs μπορούν να ενσωματώνουν μία σειρά από λειτουργικότητες σε ένα ενιαίο πακέτο. Ωστόσο, η ενσωμάτωση αυτή, και ο βαθμός ολοκλήρωσης των διαφόρων πλευρών της διαδικασίας ανάπτυξης μπορεί να εννοηθεί αποτελεσματικότερα σαν την ενσωμάτωση πολλαπλών εξειδικευμένων εργαλείων, με το κάθε ένα να εξυπηρετεί μία συγκεκριμένη εργασία ή πλευρά της. Τα χαρακτηριστικά ενός IDE και το σημείο που καταλαμβάνει στο

¹⁰ <http://www.oracle.com/technetwork/developer-tools/jdev/overview/index.html>

¹¹ <https://developer.apple.com/xcode/>

¹² <http://eclipse.org>

¹³ <http://www.visualstudio.com>

φάσμα της ολοκλήρωσης μπορούν να εκτιμηθούν με τη βοήθεια συγκεκριμένων μορφών ενοποίησης των επιμέρους εργαλείων. Έτσι, μία από τις πλέον δημοφιλείς κατηγοριοποιήσεις διατυπώνει πέντε διαστάσεις με τις οποίες μπορεί να γίνει κατανοητή η ολοκλήρωση αυτή (Wasserman, 1990) :

- «Ολοκλήρωση πλατφόρμας» : Τα εργαλεία και οι λειτουργικότητες πρέπει να διαλειτουργούν μεταξύ τους. Αυτό μπορεί να επιτυγχάνεται με την ένταξη της λειτουργικότητας σε ένα ενιαίο υπόβαθρο λογισμικού (που αποτελείται π.χ. από ενδιάμεσο λογισμικό, υποδομές ΤΠΕ κτλ.), το οποίο παρέχει το υπόβαθρο της διαλειτουργικότητας αυτής.
- «Ολοκλήρωση παρουσίασης»: Από την πλευρά του χρήστη, τα διάφορα εργαλεία πρέπει να έχουν κοινή όψη και τρόπο χρήσης. Η ενιαία παρουσίαση και συμπεριφορά της αλληλεπίδρασης ανθρώπου – μηχανής επιτρέπουν στον χρήστη να μάθει νέα εργαλεία πιο γρήγορα και αποτελεσματικά αφού μπορεί να χρησιμοποιήσει συμβάσεις που γνωρίζει ήδη.
- «Ολοκλήρωση δεδομένων»: Είναι απαραίτητη η κοινή χρήση δεδομένων και η επικοινωνία μεταξύ των ενσωματωμένων εργαλείων, καθώς αυτά χρησιμοποιούνται διαδοχικά ή παράλληλα κατά τη διαδικασία ανάπτυξης λογισμικού. Σε περιβάλλοντα ανάπτυξης με μικρή ολοκλήρωση, αυτό επιτυγχάνεται με τη μεταφορά αρχείων. Σε περισσότερο ολοκληρωμένα περιβάλλοντα αυτό μπορεί να επιτυγχάνεται με την κοινή χρήση βάσεων δεδομένων και αποθετηρίων.
- «Ολοκλήρωση ελέγχου»: Τα διάφορα εργαλεία και στοιχεία του περιβάλλοντος ανάπτυξης θα πρέπει να μπορούν να αλληλοενημερώνονται για διάφορα συμβάντα. Για παράδειγμα, η ολοκλήρωση της μεταγλώττισης του πηγαίου κώδικα παράγει ειδοποιήσεις σχετικά με σφάλματα μεταγλώττισης, τα οποία μπορούν να μεταδίδονται και υποδεικνύονται στο εργαλείο της σύνταξης. Η «ολοκλήρωση ελέγχου» αναφέρεται στον βαθμό που τα διάφορα εργαλεία επιτυγχάνουν την επικοινωνία αυτή.
- «Ολοκλήρωση διαδικασίας»: Τα εργαλεία και ο τρόπος διασύνδεσής τους πρέπει να μπορούν να εξυπηρετήσουν τις ανάγκες και τα βήματα μίας σαφώς καθορισμένης διαδικασίας ανάπτυξης λογισμικού. Ένα ολοκληρωμένο περιβάλλον ανάπτυξης θα μπορεί να παρέχει υποστήριξη στην διαχείριση και εποπτεία της διαδικασίας αυτής, διασυνδέοντας τη λειτουργικότητα αυτή με τις υπόλοιπες λειτουργικότητες ανάπτυξης που το απαρτίζουν.

Σε ένα ενιαίο περιβάλλον ανάπτυξης μπορούν να ενσωματωθούν διάφορων τύπων εργαλεία. Μία κατηγορία εργαλείων είναι τα «οριζόντια εργαλεία», τα οποία υποστηρίζουν τη διαδικασία γενικά, όπως π.χ. τεκμηρίωση, διαχείριση έργου, κτλ. Από την άλλη, τα «κάθετα εργαλεία» αφορούν συγκεκριμένες φάσεις της διαδικασίας ανάπτυξης, όπως π.χ. σύνταξη κώδικα, διαχείριση μοντέλων, μεταγλώττιση κτλ. Τα κάθετα εργαλεία μπορούν με τη σειρά τους να διαχωριστούν σε κατηγορίες, ανάλογα με το επίπεδο σχεδιασμού και ανάπτυξης στο οποίο στοχεύουν. Η πρώτη κατηγορία αφορά σε εργασίες σε υψηλό (περισσότερο αφαιρετικό) επίπεδο, π.χ. ανάλυση απαιτήσεων και σχεδιασμό, ενώ η δεύτερη κατηγορία αφορά σε εργασίες σε χαμηλότερο, λιγότερο αφαιρετικό επίπεδο, όπως η σύνταξη κώδικα και οι δοκιμές. (Wasserman, 1990)

Σαν ερευνητικό αντικείμενο, τα IDEs και η ενσωμάτωση εργαλείων (Tool Integration) εντάσσονται στο ευρύτερο πεδίο της ανάπτυξης λογισμικού με υποβοήθηση υπολογιστών (Computer- Aided Software Engineering – CASE). Τη στιγμή αυτή, υπάρχει σημαντικό ενδιαφέρον για το αντικείμενο, και η σχετική έρευνα καλύπτει μία σειρά από οπτικές γωνίες. Οι Asplund και Törngren παρατηρούν ότι η πρόσφατη έρευνα καλύπτει πεδία όπως τα χαρακτηριστικά ενός ολοκληρωμένου περιβάλλοντος, η ανάπτυξη των διάφορων τύπων ολοκλήρωσης, προτάσεις για πλαίσια υλοποίησης για ολοκληρωμένα περιβάλλοντα, προτάσεις για μοντέλα αναφοράς, προϊόντα λογισμικού, αξιολόγηση προτύπων, προδιαγραφών και μοντέλων αναφοράς κ.α. Οι συγγραφείς επισημαίνουν ότι υπάρχουν κρίσιμα ερωτήματα που πρέπει να οδηγήσουν μελλοντική έρευνα. Μεταξύ άλλων, αναφέρουν ότι υπάρχει ανάγκη να διερευνηθεί το πώς η ολοκλήρωση εργαλείων λογισμικού εξαρτάται από τις απαιτήσεις διαφορετικών πεδίων εφαρμογής, υπό ποιες συνθήκες αυξάνεται η παραγωγικότητα της ανάπτυξης λογισμικού μέσω αυτοματοποίησης, και, πως μπορούν να υποστηρίξουν τους εμπλεκόμενους στη διαδικασία ανάπτυξης, όπως μεταξύ άλλων, οι ειδικοί του πεδίου εφαρμογής, οι υπεύθυνοι έργων και οι πελάτες. (Asplund & Törngren, 2015)

3.4 Μεθοδολογίες σχεδιασμού συστημάτων υπηρεσιών

3.4.1 Στάδια ανάπτυξης νέας υπηρεσίας και ΣΠΥ

Όπως και στα παραδοσιακά προϊόντα, ο κύκλος ζωής ενός ΣΠΥ περιλαμβάνει μία σειρά από στάδια. Έτσι, μπορεί να έχει έναν μία αρχή (σύλληψη και σχεδιασμός), μέση (παροχή και διοίκηση), και τέλος (τροποποίηση ή απόσυρση), δηλαδή εντάσσεται στα πλαίσια ενός «κύκλου ζωής υπηρεσίας». Η διαδικασία μπορεί να ανατροφοδοτείται, π.χ. κατά τη διάρκεια της παροχής μίας υπηρεσίας μπορεί να ληφθεί η απόφαση για τον ανασχεδιασμό της, άρα η επιστροφή σε πρωιμότερα στάδια σχεδιασμού και ανάπτυξης. Ένα ΣΠΥ ξεκινά με την παράλληλη ανάλυση απαιτήσεων των πελατών και χρηστών, την ανάπτυξη και εξατομίκευση των προϊόντων και υπηρεσιών, τη διάθεση στην αγορά, και την υποστήριξη των πελατών που το χρησιμοποιούν. Οι φάσεις αυτές μπορούν να επαναληφθούν π.χ. κατά την τροποποίηση του ΣΠΥ, ενώ σε άλλες περιπτώσεις καταλήγουν στην τελική απόσυρση του ΣΠΥ και παύση της λειτουργίας του. (Berkovich, Leimeister, & Krcmar, 2011).

Μάλιστα, το 2003, οι Bullinger κ.α., θέτουν δύο βασικά ερωτήματα ως κεντρικά ζητήματα της μηχανικής υπηρεσιών: «Πως θα έπρεπε να αναπτύσσουμε υπηρεσίες;» και «Πως μπορεί να υποστηριχθεί αυτή η διαδικασία;». Η προσέγγισή τους ξεκινά από τη βάση των αρχών ανάπτυξης νέων προϊόντων, και θεωρεί ότι μία υπηρεσία αναπτύσσεται σε τρία επίπεδα:

- Μοντέλο προϊόντος: τα περιεχόμενα της παρεχόμενης υπηρεσίας και τα αναμενόμενα αποτελέσματα
- Μοντέλο διαδικασίας: ο τρόπος με τον οποίο γίνεται η παροχή των προϊόντων
- Μοντέλο πόρων: οι πόροι που απαιτούνται για την εκτέλεση των απαιτούμενων διαδικασιών

Αναπτύσσοντας το μοντέλο αυτό περαιτέρω, για την ανάπτυξη μίας νέας υπηρεσίας, χρειάζεται ο καθορισμός των απαιτούμενων πόρων (ανθρώπινων, υλικών και άυλων), ο καθορισμός των δραστηριοτήτων στις οποίες συμμετέχουν, του παραγόμενου προϊόντος, το

κανάλι/ κανάλια πώλησης και παροχής της υπηρεσίας καθώς και οι τελικοί παραλήπτες (τμήμα αγοράς και ανεξάρτητοι πελάτες).

Όσο αφορά σε μεθοδολογίες ανάπτυξης υπηρεσιών, οι ερευνητές παρατηρούν ότι στη περίπτωση υπηρεσιών με μικρή μεταβλητότητα (π.χ. υψηλή τυποποίηση) είναι σχετικά ευκολότερη η εφαρμογή συστηματικών μεθόδων ανάπτυξης από άλλα πεδία, όπως π.χ. των υλικών προϊόντων ή του λογισμικού. Αντίθετα, όπου απαιτείται ιδιαίτερη προσαρμογή είναι δυσκολότερη η εφαρμογή παραδοσιακών μεθόδων (Bullinger, Fährnrich, & Meiren, 2003).

Η ανάπτυξη ενός ΣΠΥ έχει ιδιαίτερο ενδιαφέρον και προκλήσεις, γιατί το σύνθετο αυτό αγαθό απαιτεί την ταυτόχρονη ανάπτυξη των δύο πλευρών του (υλικής και άυλης) καθώς και των αλληλεπιδράσεων μεταξύ τους. Συγκεκριμένα, περιλαμβάνει την ανάπτυξη του υλικού προϊόντος/ προϊόντων (σχεδιασμός, μέθοδοι παραγωγής, ποιότητα κτλ.), υπηρεσιών (οργανωτικές δομές, τρόπος παροχής, ανθρώπινοι πόροι κτλ.), αλλά και υποστηρικτικών υποδομών/ λειτουργιών, όπως οι υποδομές ΤΠΕ.

Σύμφωνα με τους Aurich, Fuchs και Wagenknecht, τα παράλληλα στάδια της παράλληλης ανάπτυξης προϊόντος και των τεχνικών υπηρεσιών εντάσσονται σε δύο παράλληλες και σχετιζόμενες ροές, όπως φαίνεται στο παρακάτω σχήμα:

Ανάπτυξη Προϊόντος	Αρχική Ιδέα και αξιολόγηση ιδεών	Ανάπτυξη σύλληψης και προετοιμασία προδιαγραφών	Δημιουργία προϊόντος: χαρακτηριστικά, διεργασίες παραγωγής	Λεπτομερής σχεδιασμός λεπτομερή σχέδια προϊόντος, παραγωγής	Ανάπτυξη πρωτοτύπου, δοκιμές, βελτιώσεις	Προετοιμασία Παραγωγής, εγκατάσταση / μίσθωση δυναμικότητας, δοκιμαστική παραγωγή
Ανάπτυξη Υπηρεσίας	Ταυτοποίηση απαιτήσεων, ανάλυση μάρκετινγκ	Ανάλυση βιωσιμότητας, κόστος, τεχνικής ικανότητας	Ανάπτυξη μεθόδων παροχής, απαιτήσεων ΤΠΕ	Μοντελοποίηση συστήματος προϊόντος - υπηρεσίας, ανταλλαγών δεδομένων	Σχεδιασμός υλοποίησης, προετοιμασία πόρων, και δεξιοτήτων, σχέδιο διάθεσης στην αγορά	Δοκιμές υπηρεσίας, βελτιώσεις, προετοιμασία για διάθεση στην αγορά

Πίνακας 3: Παράλληλη ανάπτυξη βιομηχανικού προϊόντος και τεχνικής υπηρεσίας (Aurich, Fuchs & Wagenknecht, 2006)

Οι ερευνητές συμπεραίνουν ότι ο σχεδιασμός βιομηχανικών τεχνικών υπηρεσιών πρέπει να γίνεται με συστηματικές μεθοδολογίες, οι οποίες θα διασυνδέονται με τις διαδικασίες ανάπτυξης προϊόντος. Επιπλέον, οι δύο παράλληλες ροές εργασιών πρέπει να διασυνδέονται για να σχηματίζουν μία ολοκληρωμένη διαδικασία ανάπτυξης ΣΠΥ, ενώ, τέλος, είναι απαραίτητη η συμμετοχή των εμπλεκόμενων στο ευρύτερο συνεργατικό δίκτυο παροχής υπηρεσιών (π.χ. πάροχοι υπηρεσιών, ή τοπικοί αντιπρόσωποι) ώστε να υπάρχει ευελιξία στην ικανοποίηση των μεταβαλλόμενων απαιτήσεων του πελάτη. (Aurich, Fuchs, & Wagenknecht, 2006).

3.4.2 Μηχανική υπηρεσιών (Service Engineering) για την ανάπτυξη και σχεδιασμό νέων υπηρεσιών

Ο κεντρικός ρόλος των υπηρεσιών στη σύγχρονη οικονομία επιβάλλει τη συστηματικότερη μελέτη της υπηρεσίας σαν εμπορεύσιμο αγαθό, μηχανισμό προστιθέμενης αξίας και καινοτομίας καθώς και πηγή ανταγωνιστικού πλεονεκτήματος για την επιχείρηση. Η

προσπάθεια αυτή έχει απαραίτητα διεπιστημονικά χαρακτηριστικά: Σύμφωνα με τους Maglio κ.α., μία «επιστήμη υπηρεσιών» θα πρέπει να συνδυάζει την κατανόηση της οργάνωσης και διοίκησης, του ανθρώπινου παράγοντα, της επιχειρηματικότητας και της τεχνολογίας, ώστε 1) να εξηγήσει την προέλευση και την ανάπτυξη των συστημάτων υπηρεσιών 2) να λύσει θεμελιώδη προβλήματα (όπως π.χ. πως επιτυγχάνεται η βελτίωση της ποιότητας και αποδοτικότητας της παροχής υπηρεσιών) και 3) να παράγει επαγγελματίες και επιστήμονες (Maglio, Srinivasan, Kreulen, & Spohrer, 2006).

Πέρα από τη διαπίστωση της ανάγκης, και του ενδεχόμενου εύρους του πεδίου, η ιδέα μίας επιστήμης των υπηρεσιών έχει αποτελέσει έναυσμα για έρευνα στα επί μέρους επιστημονικά πεδία που εμπλέκει, όπως τα πληροφοριακά συστήματα, η διοίκηση, το μάρκετινγκ κτλ. (Böhmman, Leimeister, & Möslin, 2014).

Το 2010, οι Ostrom κ.α., διερεύνησε τις ανάγκες του πεδίου των υπηρεσιών και διατύπωσε μία σειρά από δέκα ερευνητικές προτεραιότητες για μία «επιστήμη των υπηρεσιών», οργανωμένες ως εξής:

- Στρατηγική διάσταση
 - Ανάπτυξη και ενσωμάτωση των υπηρεσιών σε έναν οργανισμό
 - Βελτίωση της ευημερίας μέσα από υπηρεσίες
 - Δημιουργία και συντήρηση μίας κουλτούρας υπηρεσιών
- Ανάπτυξη υπηρεσιών
 - Προώθηση της καινοτομίας στις υπηρεσίες
 - Βελτίωση του σχεδιασμού των υπηρεσιών
 - Βελτιστοποίηση των δικτύων υπηρεσιών και των αλυσίδων αξίας
- Εκτέλεση υπηρεσιών
 - Αποτελεσματικό μάρκετινγκ και πώληση υπηρεσιών
 - Βελτίωση της εμπειρίας του πελάτη μέσω της «συνδημιουργίας» αξίας
 - Μέτρηση και βελτιστοποίηση της προστιθέμενης αξίας της υπηρεσίας
- Γενικά:
 - Χρήση της τεχνολογίας για καλύτερες υπηρεσίες

Σε αυτά τα πλαίσια, εντάσσεται η έννοια της «μηχανικής υπηρεσιών» (Service Engineering), δηλαδή ο συστηματικός σχεδιασμός και ανάπτυξη συστημάτων υπηρεσιών. Σύμφωνα με τους Bullinger κ.α., η μηχανική υπηρεσιών μπορεί να εννοηθεί σαν μία τεχνική επιστήμη, η οποία ασχολείται με τη συστηματική ανάπτυξη και σχεδιασμό υπηρεσιών, χρησιμοποιώντας κατάλληλες διαδικασίες, μεθόδους και εργαλεία. Η μηχανική υπηρεσιών υιοθετεί μία περισσότερο τεχνική και μεθοδολογική προσέγγιση, ώστε να χρησιμοποιήσει υφιστάμενη τεχνογνωσία από την περιοχή της παραδοσιακής ανάπτυξης προϊόντων, για την ανάπτυξη καινοτόμων υπηρεσιών. (Bullinger, Fähnrich, & Meiren, 2003)

Από την πλευρά τους, οι Böhmman κ.α. χρησιμοποιούν τον όρο «μηχανική συστημάτων υπηρεσιών» (service systems engineering) για το ίδιο πεδίο, αλλά με θεμελιώδη πλέον έννοια το σύστημα υπηρεσιών, και με ειδική έμφαση στη χρήση πληροφοριακών συστημάτων και Τεχνολογιών Πληροφορικής και Επικοινωνιών για την υλοποίηση του. Παρατηρώντας την ανάγκη για την ανάπτυξη εργαλείων και μεθοδολογιών όσο το δυνατόν

ανεξάρτητων από συγκεκριμένα πεδία εφαρμογής (π.χ. υγεία, ενέργεια κτλ.), εντοπίζουν μία σειρά από βασικές προκλήσεις σε τρεις άξονες (Böhmman, Leimeister, & Mösllein, 2014):

- Σχεδιασμός Αρχιτεκτονικών Υπηρεσιών (Engineering Service Architectures) : Αφορά στον σχεδιασμό των κατάλληλων συνδυασμών και διατάξεων πόρων, παραγόντων και δραστηριοτήτων συμπαραγωγής προστιθέμενης αξίας.
 - Καινοτομία: μηχανισμοί για την καινοτομία σε συστήματα υπηρεσιών, με στόχο την καλύτερη ανταπόκρισή τους στις ανάγκες των εμπλεκόμενων πλευρών, την ευελιξία, και την προσαρμοστικότητα σε νέες και εξωτερικές συνθήκες
 - «Κυβερνο – υλικά» συστήματα (cyber-physical systems): η συνεχώς αυξανόμενη χρήση Τεχνολογιών Πληροφορικής και Επικοινωνιών στην παροχή υπηρεσιών σημαίνει ότι τα συστήματα υπηρεσιών θα πρέπει να σχεδιάζονται και να αναπτύσσονται ώστε να συνδυάζουν με επιτυχία τεχνολογικούς και ανθρώπινους πόρους.
 - Προχωρημένα μοντέλα, Μέθοδοι και Εργαλεία για την Ανάπτυξη Αρχιτεκτονικών Υπηρεσιών: Τεχνικές και εργαλεία για την ανάπτυξη συστημάτων υπηρεσιών με έμφαση στη τυποποίηση, ευελιξία, τμηματοποίηση και αναδιάταξη στοιχείων των συστημάτων υπηρεσιών, καθώς και μέθοδοι για τη μοντελοποίηση και προσομοίωσή τους.
- Σχεδιασμός Αλληλεπιδράσεων Συστημάτων Υπηρεσιών (Engineering Service Systems Interactions): τα Συστήματα Υπηρεσιών έχουν αλληλεπιδράσεις μεταξύ τους και με το εξωτερικό περιβάλλον. Αυτές οι αλληλεπιδράσεις διευκολύνονται με τις νέες τεχνολογίες (π.χ. με τεχνολογίες διαδικτύου, αισθητήρες, κινητές επικοινωνίες, γεωεντοπισμός κτλ.). Άλλες αφορούν τα εμπλεκόμενα μέρη, όπως μηχανισμοί επίλυσης διαφορών μεταξύ των εμπλεκόμενων, αξιοπιστία και διαχείριση κινδύνων κτλ. Τέλος, κρίσιμη είναι και η ανάπτυξη μεθοδολογιών και εργαλείων για τη μελέτη και σχεδιασμό αλληλεπιδράσεων γενικότερα, αλλά και ειδικότερα με τη χρήση ΤΠΕ καθώς και υπό-πεδία όπως η αλληλεπίδραση ανθρώπου – υπολογιστή και η ευχρηστία.
- Σχεδιασμός Κινητοποίησης Πόρων (Engineering Resource Mobilization): Αφορά στην αξιοποίηση τριών βασικών ειδών πόρων που εμπεριέχονται σε ένα σύστημα υπηρεσιών:
 - Ανθρώπινο δυναμικό π.χ. μέσω της ανάθεσης εργασιών ή της διασύνδεσης εκπροσώπων με τους πελάτες κ.α.
 - Υλικοί πόροι, όπως π.χ. συστήματα προγραμματισμού για τη χρήση μηχανημάτων και άλλων παγίων κ.α.
 - Πληροφοριακοί πόροι, όπως ανοικτά δεδομένα, οργανωσιακή γνώση και περιεχόμενο παραγόμενο από τους χρήστες (π.χ. όπως κριτικές για την ποιότητα υπηρεσιών) κ.α.

3.4.3 Μεθοδολογίες ανάπτυξης Συστημάτων Προϊόντος Υπηρεσίας και υποδομών λογισμικού

Η τάση για την ενσωμάτωση υπηρεσιών στη σύγχρονη βιομηχανία φέρνει στο προσκήνιο την ανάγκη για συστηματική αντιμετώπιση του προβλήματος της ανάπτυξης νέων Συστημάτων

Προϊόντος – Υπηρεσίας. Η έρευνα στο συγκεκριμένο πεδίο στηρίζεται σε μεγάλο βαθμό στις αντίστοιχες προσπάθειες για να την διερεύνηση και συστηματοποίηση της ανάπτυξης νέων υπηρεσιών. Ωστόσο, ενώ ένα ΣΠΥ περιλαμβάνει υπηρεσίες, αυτές είναι αλληλένδετες με τα αντίστοιχα φυσικά προϊόντα και απαρτίζουν ένα ενιαίο σύστημα, το οποίο αναπτύσσεται συνολικά. Συνεπώς, η εφαρμογή μεθόδων που αποσκοπούν αποκλειστικά στην ανάπτυξη νέων υπηρεσιών αγνοούν κρίσιμες πλευρές των ΣΠΥ και παρέχουν μόνο μερική υποστήριξη.

Για τον λόγο αυτό, τα τελευταία χρόνια έχει υπάρξει ιδιαίτερο ενδιαφέρον για την ανάπτυξη συστηματικών μεθόδων για τη δημιουργία ΣΠΥ, οι οποίες θα καλύπτουν τις παράλληλες ροές ανάπτυξης των συστατικών του στοιχείων. Οι περισσότερες από αυτές, καθώς προέρχονται από το πεδίο της διοίκησης, του μάρκετινγκ και της επιχειρησιακής έρευνας, και εστιάζουν στην ανάλυση του ΣΠΥ από την πλευρά του επιχειρησιακού σχεδιασμού σε υψηλό επίπεδο. Έτσι, καλύπτουν ζητήματα όπως οι επιχειρησιακές διαδικασίες, ο σχεδιασμός των νέων προϊόντων, η οργάνωση των ανθρώπινων πόρων κτλ.

Καθώς, όμως, οι ΤΠΕ αποτελούν βασικό παράγοντα για την επίτευξη της προστιθέμενης αξίας σε σύγχρονες υπηρεσίες και ΣΠΥ, ένας μικρός αριθμός από αυτές επιχειρεί να γεφυρώσει το χάσμα ανάμεσα στον επιχειρησιακό σχεδιασμό και την ανάπτυξη των απαραίτητων υποδομών. Οι μεθοδολογίες αυτές αναλύονται στη παρούσα ενότητα.

Ο Morelli διαπιστώνει, ήδη από το 2002, ότι ο σχεδιασμός ενός ΣΠΥ έχει διεπιστημονικά χαρακτηριστικά, και για τον λόγο αυτό, εξετάζει πεδία που μπορεί να συνεισφέρουν μεθοδολογικά στον σχεδιασμό ενός ΣΠΥ. Διαπιστώνει ότι στην ανάπτυξη ενός ΣΠΥ περιλαμβάνονται και οι αντίστοιχες τεχνολογικές υποδομές, μαζί με το προσωπικό, τις δραστηριότητες μάρκετινγκ, τις σχέσεις με τους πελάτες κτλ. Διαπιστώνει μάλιστα ότι οι περισσότερο ενδιαφέρουσες προσεγγίσεις προέρχονται από τον χώρο της διοίκησης, του μάρκετινγκ και των πληροφοριακών συστημάτων. Παράλληλα, παρουσιάζει παράδειγμα μοντελοποίησης ενός ΣΠΥ με διάγραμμα ροής και δραστηριότητες, όπου γίνεται διάκριση μεταξύ των δραστηριοτήτων που λαμβάνουν χώρα σε πληροφοριακά συστήματα και στον φυσικό κόσμο, καθώς και οι διεπαφές μεταξύ των δύο πεδίων (Morelli, 2002).

Οι Aurich, Fuchs και Wagenknecht, διατυπώνουν ένα γενικό πλαίσιο για μία συστηματική μεθοδολογία ανάπτυξης Συστημάτων Προϊόντος – Υπηρεσίας για προϊόντα της βιομηχανικής αγοράς, βασισμένη στο γεγονός ότι η ανάπτυξη των άυλων και υλικών πλευρών του ΣΠΥ γίνεται με τη μορφή παράλληλων ροών εργασίας, στα πλαίσια ενός ενοποιημένου κύκλου ζωής. Οι ερευνητές προτείνουν, λοιπόν, τη διασύνδεση των σταδίων ανάπτυξης του ΣΠΥ, αναλύοντας τις επιχειρησιακές διαδικασίες σε τμήματα του ΣΠΥ, με ορισμένα από αυτά να οδηγούν σε απαιτήσεις για την ανταλλαγή δεδομένων μέσω ΤΠΕ. Ωστόσο, οι ερευνητές περιορίζονται σε ανάλυση υψηλού επιπέδου της διαδικασίας και δεν προτείνουν τεχνικές ή μηχανισμούς για την ανάπτυξη προδιαγραφών και συστημάτων ΤΠΕ. (Aurich, Fuchs, & Wagenknecht, 2006).

Οι Maussang, κ.α. διατυπώνουν τις γενικές αρχές μίας διαδικασίας ανάπτυξης ΣΠΥ σε δύο φάσεις. Στη πρώτη φάση συλλέγονται ανάγκες της αγοράς, προσδιορίζονται οι απαραίτητες υπηρεσίες βάσει κριτηρίων, και διατυπώνονται οι απαιτήσεις για τις λειτουργίες και υποδομές που θα τις παράσχουν. Από το σημείο αυτό, οι ερευνητές προτείνουν την απεικόνιση των ζητούμενων λειτουργιών με τη χρήση απλών διαγραμμάτων λειτουργιών και

συσχετίσεων σε υψηλό επίπεδο FBD (Functional Block Diagram), η FAST (Function Analysis System Technique)¹⁴, και Internal Functional Analysis¹⁵. Οι συγκεκριμένες μέθοδοι χαρακτηρίζονται από σχετική χαλαρότητα και χαμηλό επίπεδο φορμαλισμού, σε σχέση με περισσότερο αυστηρές μεθοδολογίες. Στη συνέχεια οι προδιαγραφές αυτές μπορούν να παραδοθούν σε αναλυτές και μηχανικούς ώστε να προχωρήσουν στον αναλυτικό σχεδιασμό των απαραίτητων υποδομών (Maussang, Sakao, Zwolinski, & Brissaud, 2007).

Οι Müller και Stark προτείνουν μία τεχνική για την υποστήριξη της ανάλυσης απαιτήσεων ενός ΣΠΥ. Η μεθοδολογία περιλαμβάνει τρεις φάσεις: 1) ανάλυση του πλαισίου του συστήματος 2) εξαγωγή προδιαγραφών 3) αποτύπωση των προδιαγραφών σε τυπική μορφή. Η πρώτη φάση περιλαμβάνει την ανάλυση του ΣΠΥ σε επίπεδα πελάτη και σχεδιασμού. Στο επίπεδο πελάτη αναλύονται οι ανάγκες και η προστιθέμενη αξία. Στο επίπεδο σχεδιασμού αναλύονται τα επιθυμητά παραδοτέα, ο κύκλος ζωής, οι εμπλεκόμενοι, τα κύρια προϊόντα, οι νομικές δεσμεύσεις και, τέλος, οι υποδομές που θα στηρίξουν το ΣΠΥ, όπως τεχνολογίες ΤΠΕ, λογισμικό κτλ. Όπως και οι απαιτήσεις των υπόλοιπων στοιχείων του ΣΠΥ, οι απαιτήσεις των υποδομών ΤΠΕ αναλύονται σε λίστες οι οποίες περιγράφουν την απαιτούμενη λειτουργικότητα και τα χαρακτηριστικά τους. Η ανάλυση αυτή περιορίζεται σε λειτουργικές περιγραφές και διατυπώσεις επιθυμητών χαρακτηριστικών και αφήνει τον αναλυτικό σχεδιασμό στο τεχνικό προσωπικό που θα υλοποιήσει το σύστημα. (Müller & Stark, 2008)

Οι Becker, Beverungen και Knackstedt, διερευνώντας το 2010, τις υφιστάμενες μεθοδολογίες επιχειρησιακής μοντελοποίησης σε σχέση με την ικανότητά τους να βοηθήσουν την ανάπτυξη των ΣΠΥ, διαπιστώνουν ότι οι γλώσσες μοντελοποίησης που δεν προέρχονται από τον χώρο της πληροφορικής τείνουν να μην αποτυπώνουν τα πληροφοριακά συστήματα και τις επιχειρησιακές μονάδες που παρέχουν τις υπηρεσίες. Ομοίως, οι γλώσσες μοντελοποίησης που δεν προέρχονται από τεχνικό υπόβαθρο δεν παρέχουν επαρκή απεικόνιση των υλικών και άλλων στοιχείων που χρειάζονται για το σχεδιασμό ενός ΣΠΥ, όπως στοιχεία για τον κύκλο ζωής του, και οι γλώσσες που δεν προέρχονται από την περιοχή του μάρκετινγκ ενδέχεται να μην αποτυπώνουν το είδος και την ένταση των αλληλεπιδράσεων με τον πελάτη. Για τον λόγο αυτό δεν μπορούν να εντοπιστούν ώριμες μεθοδολογίες και γλώσσες που να καλύπτουν επαρκώς όλες τις απαιτούμενες συνιστώσες και διαδικασίες ενός ΣΠΥ σε επαρκή βαθμό (Becker, Beverungen, & Knackstedt, 2010).

Το 2009, οι Berkovich, Leimeister και Krcmar υποστηρίζουν ότι η ανάπτυξη ενός ΣΠΥ απαιτεί ταυτόχρονα την ανάπτυξη νέου προϊόντος, υπηρεσίας και λογισμικού. Καταγράφουν μία σειρά από απαραίτητες διαστάσεις που πρέπει να καλύπτει η ανάλυση απαιτήσεων για ένα ΣΠΥ, όπως η διεπιστημονικότητα, η διαχείριση αλλαγών, οι διαφορετικές ορολογίες των εμπλεκόμενων τεχνικών πεδίων, η εμπλοκή του πελάτη κτλ. Συμπεραίνουν ότι οι υφιστάμενες τεχνικές ανάλυσης απαιτήσεων δεν μπορούν από μόνες τους να καλύψουν τις απαιτήσεις των ΣΠΥ, και ότι χρειάζεται επιπλέον ερευνητική προσπάθεια (Berkovich, Leimeister, & Krcmar, 2009).

¹⁴ <http://www.scav-csva.org/fast.php>

¹⁵ <http://accelinn.com/en/38-unified-efficiency/functional-methodology/functional-analysis/143-internal-functional-analysis>

Το 2011 οι ίδιοι ερευνητές παρουσιάζουν ένα πλαίσιο για το πεδίο της «μηχανικής απαιτήσεων» (requirements engineering) για τα ΣΠΥ. Αυτό επιτυγχάνεται διατυπώνοντας ένα μοντέλο που περιλαμβάνει τα κυριότερα στοιχεία του, όπως τον κύκλο ζωής, τις διαδικασίες ανάπτυξης, τους πελάτες κτλ. Στο μοντέλο αυτό, οι ερευνητές ξεχωρίζουν την ανάπτυξη λογισμικού ως δραστηριότητα ίσης σημασίας με την παράλληλη ανάπτυξη του προϊόντος και της υπηρεσίας. Επιπλέον, εξετάζοντας μεθοδολογίες για την ανάπτυξη υλικών αγαθών, υπηρεσιών, λογισμικού και ΣΠΥ, υποστηρίζουν ότι είναι απαραίτητη η ενσωμάτωση στοιχείων από όλα αυτά τα πεδία με σκοπό την κάλυψη των πολλαπλών στοιχείων, ιδιοτήτων και πλευρών ενός ΣΠΥ. (Berkovich, Leimeister, & Krcmar, Requirements Engineering for Product Service Systems, 2011).

Το 2012, οι Vasantha κ.α, ανέλυσαν την μέχρι τότε έρευνα πάνω στην ανάπτυξη Συστημάτων Προϊόντος – Υπηρεσίας, και ανέλυσαν μεθοδολογίες ανάπτυξης ΣΠΥ με υψηλό βαθμό ωριμότητας. Ο σκοπός ήταν η εξέταση της κατάστασης του πεδίου και εντοπισμός μεθοδολογικών κενών. Οι ερευνητές διαπίστωσαν ότι ο τομέας της μοντελοποίησης των ΣΠΥ δεν είναι ιδιαίτερα ώριμος, και κάνουν ειδική μνεία στην ανάγκη απεικόνισης της ροής πληροφορίας και δεδομένων σε ένα ΣΠΥ. Επιπλέον, κάνουν ειδική μνεία στη μεθοδολογία MDA (Model – Driven Architecture) ως πιθανό μέσο για την συστηματοποίηση της επικοινωνίας μεταξύ των πολλαπλών εμπλεκόμενων και την ενσωμάτωση των διαφορετικών πεδίων που εμπλέκονται κατά την ανάπτυξη ενός ΣΠΥ, καθώς και την απεικόνισή του σε περισσότερο και λιγότερο αφαιρετικά επίπεδα. (Vasantha, Roy, Lelah, & Brissaud, 2012)

Το 2013, οι Zhao και Cai αποπειρώνται να εφαρμόσουν τις τεχνικές της μοντελο-στραφούς αρχιτεκτονικής (Model - Driven Architecture) για την συνολική μοντελοποίηση ενός ΣΠΥ. Παρουσιάζουν ένα μετα-μοντέλο ενός ΣΠΥ, δηλαδή ένα μοντέλο που περιλαμβάνει όλα τα πιθανά στοιχεία, οντότητες και σχέσεις ενός μοντελοποιημένου συστήματος, π.χ. «πελάτης» που λαμβάνει «υπηρεσίες» και «προϊόντα», καθώς και τους «Πόρους ΤΠΕ». Οι ερευνητές υποστηρίζουν ότι με τον τρόπο αυτό μπορεί να μοντελοποιηθεί πλήρως ένα ΣΠΥ, αλλά δεν αναπτύσσουν περαιτέρω τα επιχειρήματά τους και δεν εμβαθύνουν περισσότερο σε εργαλεία και μεθοδολογίες για τη συνέχιση της διαδικασίας (Zhao & Cai, 2013).

Το 2014, οι Nguyen, Exner, Schnürmacher και Stark προτείνουν μία ολοκληρωμένη μεθοδολογία διαχείρισης έργου για την ανάπτυξη ενός ΣΠΥ για τη βιομηχανική αγορά (B2B). Η μεθοδολογία αποσκοπεί στο να υποστηρίξει έναν οργανισμό να διαχειριστεί τη δημιουργία ενός ΣΠΥ, ορίζοντας τα διάφορα στάδια και φάσεις ανάπτυξης, από την σύλληψη μέχρι τη διάθεση στην αγορά. Επιπλέον, κατανέμει τις διάφορες δραστηριότητες σε «διαδρόμους» (swimlanes) π.χ «ενέργειες της διοίκησης», καθώς και τις διαφορετικές πλευρές του που πρέπει να αναπτύξει, όπως η «ανάπτυξη προϊόντος», «ανάπτυξη υπηρεσίας» κτλ. Μεταξύ των στοιχείων αυτών περιλαμβάνεται και η ανάπτυξη λογισμικού για την υποστήριξη του ΣΠΥ ως ξεχωριστή δραστηριότητα. Ωστόσο, δεν γίνεται σαφής αναφορά στο πως γίνονται οι μεταβάσεις και τι είδους επικοινωνία υπάρχει ανάμεσα στα διάφορα στάδια (π.χ. από τον «προσδιορισμό στοιχείων ΣΠΥ και διεπαφών» στην «ανάπτυξη λογισμικού», «λεπτομερή σχεδιασμό προϊόντος», και «λεπτομερή σχεδιασμό υπηρεσίας»). Αναφέρεται απλώς ότι η ευθυγράμμιση λογισμικού και υπηρεσίας θα επιτυγχάνεται εφόσον υπάρχουν επαρκή σημεία επικοινωνίας ανάμεσα στις διαδικασίες ανάπτυξης των στοιχείων αυτών (Nguyen, Exner, Schnürmacher, & Rainer, 2014).

Το 2014, οι Berkovich, Leimeister, Hoffmann και Krcmar σε συνέχεια της προηγούμενης έρευνάς τους, περιγράφουν ένα μοντέλο δεδομένων για την ανάλυση απαιτήσεων ενός ΣΠΥ (PSS Requirement Data Model). Το μοντέλο αυτό περιγράφει τους διάφορους τύπους των προδιαγραφών που μπορούν να προκύψουν από την ανάλυση απαιτήσεων ενός ΣΠΥ, καθώς και τις μεταξύ τους σχέσεις. Ο απώτερος στόχος υπήρξε η επίλυση του προβλήματος της δόμησης των απαιτήσεων, της συστηματικής τους παρακολούθησης και της ανακάλυψης αντιφάσεων. Έτσι, προδιαγράφουν μία διαδικασία σχεδιασμού και ανάλυσης απαιτήσεων, η οποία ξεκινά από τις επιχειρησιακές απαιτήσεις του πελάτη και της εταιρείας («Επίπεδο Στόχων»), και προχωρά σταδιακά σε λιγότερο αφαιρετικά επίπεδα περιγραφής. Τα επίπεδα αυτά περιλαμβάνουν

- το «Επίπεδο Συστήματος» (System Level) (που περιλαμβάνει τις γενικές λειτουργίες και στόχους του ΣΠΥ χωρίς αναφορά στις πιθανές τεχνικές ή οργανωτικές λύσεις για την επίτευξή τους,). Οι απαιτήσεις είναι οργανωμένες στην οπτική γωνία του πελάτη του παρόχου, της επιχειρησιακής διαδικασίας και του εξωτερικού περιβάλλοντος
- το «Επίπεδο Χαρακτηριστικών» (Feature Level) (που περιγράφει τις απαιτήσεις για το ΣΠΥ και τα άυλα/ υλικά του μέρη, το υλικό αγαθό, τα επιθυμητά αποτελέσματα των υπηρεσιών, διαδικασίες και πόροι, τεχνικοί και ανθρωπίνους). Στα πλαίσια των πόρων περιλαμβάνονται και οι απαιτήσεις «πληροφορίας» π.χ. μορφότυπους δεδομένων, αναφορές και τεχνολογίες
- το «Επίπεδο Λειτουργίας» (Function Level) όπου τα παραπάνω στοιχεία αναλύονται ως προς τις λειτουργίες τους
- το «Επίπεδο Συστατικών Στοιχείων» (Component Level), στο οποίο δομείται το ΣΠΥ με τα συστατικά του στοιχεία σύμφωνα με τις απαιτήσεις της δομής και λειτουργίας που προδιαγράφηκαν στα προηγούμενα στάδια. Τα συστατικά πλέον που χρησιμοποιούνται εξειδικεύονται στα ειδικά πεδία εφαρμογής στα οποία ανήκουν. Στο επίπεδο αυτό περιλαμβάνονται οι αναλυτικές απαιτήσεις για το επιχειρησιακό λογισμικό που θα υποστηρίξει το ΣΠΥ, μαζί με τις αντίστοιχες απαιτήσεις για τα φυσικά προϊόντα και τις υπηρεσίες.

Το μοντέλο σταματά στο επίπεδο αυτό, χωρίς να προσδιορίζει περαιτέρω τον τρόπο με τον οποίο θα συνεχιστεί η ανάπτυξη των επιμέρους στοιχείων, και συγκεκριμένα, του λογισμικού. Οι ερευνητές έχουν αναπτύξει αναλυτικές ταξονομίες απαιτήσεων μόνο μέχρι το «Επίπεδο Χαρακτηριστικών». Παρά το γεγονός ότι περιγράφουν τους στόχους του κάθε βήματος, δεν προτείνουν μηχανισμούς ή εργαλεία για την πρακτική του υλοποίησης. Τέλος, οι ερευνητές παρατηρούν ότι το μοντέλο αντιστοιχεί στα αρχικά στάδια της ανάπτυξης ενός ΣΠΥ, και ότι υπάρχει ανάγκη για περαιτέρω έρευνα για τα επόμενα στάδια του κύκλου ζωής. Συγκεκριμένα, διαπιστώνουν ότι χρειάζεται να διερευνηθεί ο τρόπος με τον οποίο θα υλοποιηθούν οι απαιτήσεις για κάθε ένα από τα επιμέρους εμπλεκόμενα πεδία (προϊόν, λογισμικό, υπηρεσίες) μέχρι τη θέση του ΣΠΥ σε λειτουργία. Η υλοποίηση αυτή πρέπει να διασφαλίζει τη συστηματική παρακολούθηση των απαιτήσεων και κατά τη διάρκεια της υλοποίησης. Επιπλέον, το συγκεκριμένο μοντέλο πρέπει να εξειδικευθεί περαιτέρω με βάση τις πραγματικές ανάγκες της βιομηχανίας. Ωστόσο, παρά τις αδυναμίες της, η συγκεκριμένη θεώρηση θέτει τις βάσεις για την αντιμετώπιση του ΣΠΥ σαν πλήρες σύστημα στο οποίο το λογισμικό παίζει ισότιμο ρόλο με τα υλικά προϊόντα και τις υπηρεσίες του ενώ πλησιάζει τις

αρχές τις προσέγγισης που προτείνει η παρούσα έρευνα. (Berkovich M. , Leimeister, Hoffmann, & Krcmar, 2014)

Οι Stokic και Correia (2015) προτείνουν τα γενικά στοιχεία μίας αρχιτεκτονικής για ένα περιβάλλον ανάπτυξης με σκοπό την επέκταση των βιομηχανικών προϊόντων με διαδικτυακές υπηρεσίες. Η διερεύνηση και ανάπτυξή του προτείνεται ως ένας από τους στόχους ενός ευρύτερου ερευνητικού έργου πάνω στην ανάπτυξη και μηχανική υπηρεσιών (“DIVERSITY” – FP7 Factories of the Future, grant 636692). Τελικά, το περιβάλλον που αναπτύχθηκε, στα πλαίσια του έργου δεν αφορά στην ανάπτυξη λογισμικού, καθώς εστιάζει στη διαχείριση του κύκλου ζωής προϊόντων σε υψηλό και περισσότερο αφηρημένο επίπεδο (Pezzotta, et al., 2016).

Οι Correia κ.α. παρουσιάζουν μία οντολογία για την επικοινωνία εμπλεκόμενων και εργαλείων κατά την ανάπτυξη ενός ΣΠΠ. Η οντολογία, μεταξύ άλλων, περιέχει την έννοια του «λογισμικού», η οποία περιλαμβάνει την υπό-έννοια «Υπηρεσία λογισμικού» (software service) την οποία όμως δεν αναπτύσσει περισσότερο (Correia, Stokic, Siafaka, & Scholze, 2017).

Τέλος, η μέθοδος που προτείνεται από τους Metzger κ.α. περιλαμβάνει την ανάπτυξη λογισμικού, αλλά εστιάζει στο πως αυτή θα γίνει από την οπτική γωνία της διοίκησης έργου. Η μέθοδος καλύπτει τη διαδικασία συλλογής των απαιτήσεων για ένα ΣΠΠ από τους εμπλεκόμενους, με καινοτομικές μεθόδους (βίντεο, επαυξημένη πραγματικότητα κ.α.) , και την ανάλυσή τους με σκοπό τη δημιουργία απαιτήσεων για πληροφοριακά συστήματα. Ωστόσο, η μέθοδος δεν εστιάζει στη μεθοδολογία και τα εργαλεία ανάπτυξης λογισμικού που θα χρησιμοποιηθούν για να ικανοποιήσουν τις απαιτήσεις αυτές. (Metzger, Niemöller, & Thomas, 2017).

Από την παραπάνω επισκόπηση προκύπτει ότι, από την πλευρά της επιστήμης της διοίκησης και του μάρκετινγκ, δεν υπάρχουν επαρκείς μεθοδολογίες και τεχνικές για την ανάπτυξη λογισμικού για Συστήματα Προϊόντος Υπηρεσίας. Παρά το γεγονός ότι οι υποδομές και λειτουργίες ΤΠΕ αναγνωρίζονται ως κεντρικός πυλώνας των ΣΠΥ και ουσιώδες μέσο για την παραγωγή της προστιθέμενης αξίας, οι περισσότερες μεθοδολογίες αγνοούν την ανάπτυξη υποδομών ΤΠΕ, και περιορίζονται σε περιγραφές υψηλού επιπέδου επιχειρησιακών αποκλειστικά συνιστωσών. Αντιθέτως, οι λίγες που λαμβάνουν υπόψη την ανάγκη για σχεδιασμό των υποδομών ΤΠΕ τείνουν να χαρακτηρίζονται από χαλαρότητα, εστίαση στο υψηλό επίπεδο περιγραφής, αποσπασματικές περιγραφές του ΣΠΥ, απουσία συστηματικών μεθόδων και μηχανισμών και μικρή ωριμότητα. Το συγκεκριμένο μεθοδολογικό κενό αυτό πρέπει να συμπληρωθεί με την προσαρμογή υφιστάμενων εργαλείων ή την δημιουργία νέων. Εξάιρεση σε αυτό αποτελεί η μεθοδολογία MDSEA (Model – Driven Service Engineering Architecture), η οποία περιγράφεται στην επόμενη ενότητα.

3.4.4 Η μεθοδολογία ανάπτυξης Συστημάτων Υπηρεσιών MDSEA

Ο σχεδιασμός ενός Συστήματος Προϊόντος – Υπηρεσίας απαιτεί την παράλληλη επισκόπηση όλων των πλευρών του, δηλαδή των υλικών πλευρών (προϊόν, μέσα παραγωγής κ.α.), των άυλων (υπηρεσίες, οργάνωση, επιχειρησιακές διαδικασίες) και της απαραίτητης υποδομής ΤΠΕ για την υποστήριξή του (λογισμικό, δεδομένα, κ.α.).

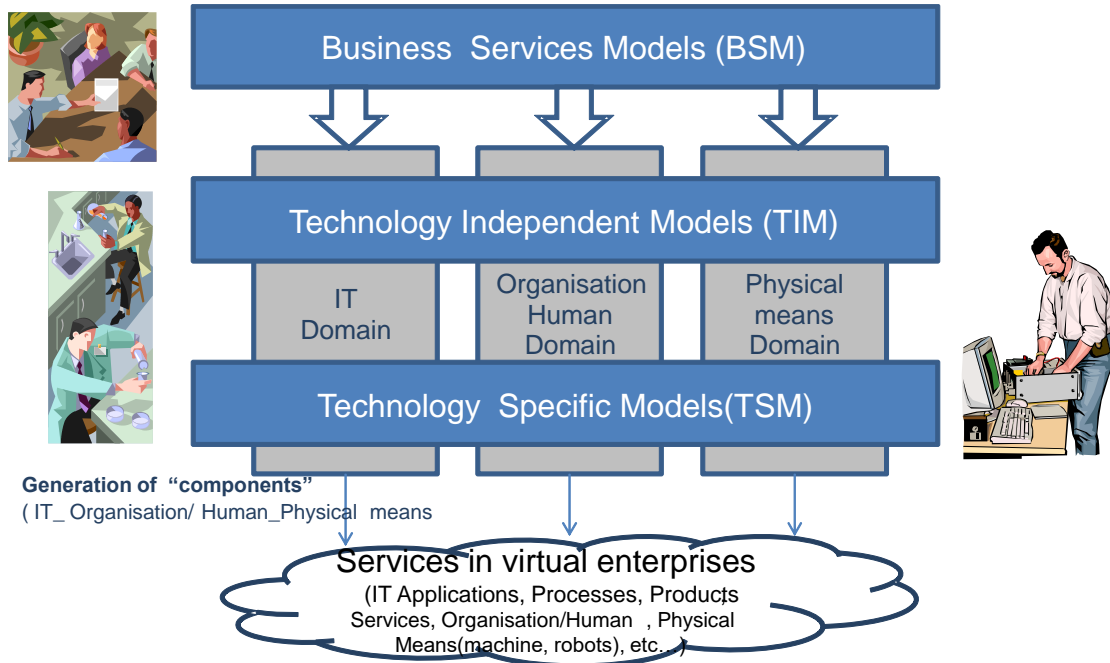
Η μεθοδολογία «Model Driven Service Engineering Architecture» (MDSEA) φιλοδοξεί να αντιμετωπίσει ολιστικά το πρόβλημα ανάπτυξης συστημάτων υπηρεσιών. Η μεθοδολογία αυτή βασίζεται στις γενικές αρχές της μεθοδολογίας MDA, δηλαδή σε διαδοχικά στάδια μοντελοποίησης και μετασχηματισμού μοντέλων. Η μεθοδολογία αποτελεί προσαρμογή της φιλοσοφίας και ροής εργασίας της MDA, αλλά έχει σχεδιαστεί με τέτοιο τρόπο ώστε να αποτυπώνει και να υποστηρίζει τη συνολική ανάπτυξη του συστήματος υπηρεσιών, ως σύνθεση υλικών και άυλων στοιχείων, υποστηριζόμενων από υποδομές ΤΠΕ, όπως λογισμικό και συστήματα επικοινωνιών.

Η μεθοδολογία ανάπτυξης λογισμικού MDA αποτελείται, συνοπτικά, από τα ακόλουθα στάδια (Touzi, Benaben, Pingaud, & Lorre, 2009):

- Προσδιορισμός ενός μοντέλου PIM (Platform Independent Model – Μοντέλο Ανεξάρτητο Πλατφόρμας)
- Προσδιορισμός ενός Platform Model (PM - Μοντέλο Πλατφόρμας) που αντιστοιχεί σε συγκεκριμένη τεχνολογική πλατφόρμα – στόχο
- Επιλογή του μοντέλου πλατφόρμας για το σύστημα υπό ανάπτυξη
- Μετασχηματισμός του PIM σε PSM (Platform – Specific Model), χρησιμοποιώντας το επιλεγμένο Platform Model
- Μετασχηματισμός του PSM σε κώδικα συστήματος

Συνεπώς, η ανάπτυξη του λογισμικού στο MSEE με βάση την MDSEA αντιστοιχεί στον μετασχηματισμό ενός μοντέλου επιπέδου PIM (Platform Independent Model) μίας βιομηχανικής υπηρεσίας σε ένα μοντέλο επιπέδου PSM (Platform Specific Model), το οποίο θα περιγράφει με τη σειρά του την υποδομή λογισμικού που θα παρέχει την υπηρεσία αυτή, όπως υλοποιείται με βάση συγκεκριμένες τεχνολογίες (π.χ. γλώσσες προγραμματισμού ή πρωτόκολλα αισθητήρων). Το μοντέλο τύπου PSM θα πρέπει στη συνέχεια να εμπλουτιστεί και να μετασχηματιστεί σε εκτελέσιμες υπηρεσίες λογισμικού (με την έννοια που αντιστοιχεί σε αρχιτεκτονικές τύπου SOA) η/και εφαρμογές.

Το δομημένο πλαίσιο της προτεινόμενης μεθοδολογίας MDSEA, διαχωρίζει τρία επίπεδα αφαίρεσης, τα οποία αναλύονται στις συνιστώσες ενός συστήματος υπηρεσίας. Τα επίπεδα αυτά αντιστοιχίζονται με τα επίπεδα TIM και TSM της MDSEA. Το πρώτο, περισσότερο αφαιρετικό, επίπεδο αφορά το σύνολο του επιχειρησιακού συστήματος. Ωστόσο, όσο τα μοντέλα γίνονται περισσότερο συγκεκριμένα, ο σχεδιασμός του συστήματος υπηρεσιών αναλύεται σε τρεις διαστάσεις: Οργάνωση/ ανθρώπινοι πόροι, φυσικά μέσα και πεδίο ΤΠΕ. Η οργάνωση αυτή φαίνεται στο παρακάτω σχήμα:



Σχήμα 3: Γενική επισκόπηση της μεθοδολογίας MDSEA του MSEE (Chen, Ducq, Doumeingts, Zachariewicz, & Alix, 2012; Ducq, Chen, & Alix, 2012; Agostinho, Bazoun, Zacharewicz, Ducq, Boye, & Jardim-Goncalves, 2013)

Αναλυτικότερα, τα τρία επίπεδα αφαίρεσης της MDSEA αντιστοιχούν στα τρία επίπεδα της MDA:

Όροι της MDA	Αντίστοιχοι όροι της MDSEA
Computation Independent Model (CIM)	Business Service Model (BSM)
Platform Independent Model (PIM)	Technology Independent Model (TIM)
Platform Specific Model (PSM)	Technology Specific Model (TSM)
Platform Model	Technology Model

Πίνακας 4: Αντιστοίχιση όρων της MDA με έννοιες της MDSEA

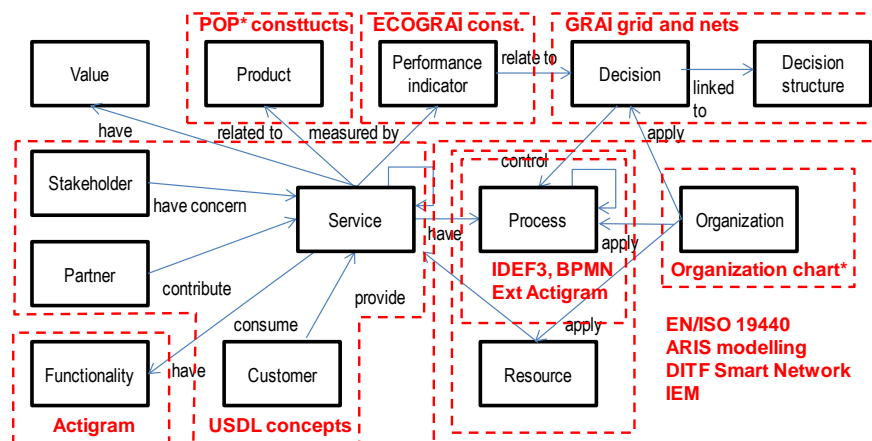
Τα τρία επίπεδα αφαίρεσης αναλύονται ως εξής:

Business Service Model (BSM): Στο επίπεδο αυτό, τα μοντέλα του ΣΠΥ αναπαριστούν την οπτική γωνία του χρήστη. Τα μοντέλα περιγράφουν το σύστημα υπηρεσιών με τρόπο όσο το δυνατόν ανεξάρτητο από την τεχνική του βάση. Τα μοντέλα στο επίπεδο BSM περιγράφουν τις επιχειρησιακές λειτουργίες, έννοιες, οντότητες και χαρακτηριστικά του συστήματος υπηρεσιών, όπως π.χ. «υπηρεσία», «προϊόν», «εμπλεκόμενοι», «οργανισμός» και «δείκτες απόδοσης». Στο επίπεδο αυτό χρησιμοποιούνται τεχνικές και έννοιες από το πεδίο της Επιχειρησιακής Μοντελοποίησης (Enterprise Modelling), όπως η προσέγγιση GRAI (Graphs with Results and Actions Inter-related) (Chen, Ducq, Doumeingts, Zachariewicz, & Alix, A Model Driven Approach for the Modeling of Services in Virtual Enterprise, 2012; Aguilar-Savén, 2004).

Η ανάλυση αυτή είναι απαραίτητη για τη γεφύρωση του κενού μεταξύ των ειδικών του πεδίου εφαρμογής και του (περισσότερο) τεχνικού προσωπικού που θα κάνει τη λεπτομερή ανάπτυξη του συστήματος. Επιπλέον, το επίπεδο αυτό διαχωρίζεται σε δύο τμήματα:

- Το «Top BSM», που αφορά στο σύστημα σε γενικές γραμμές
- Το «Bottom BSM» που αφορά σε λεπτομέρειες του επιχειρησιακού πεδίου εφαρμογής

Με βάση την αρχική ανάλυση, το σύστημα υπηρεσιών αποσυντίθεται στις συνιστώσες που το αποτελούν, δηλαδή Συστήματα ΤΠΕ, Οργάνωση/Ανθρώπινο Δυναμικό και Υλικά Μέσα.

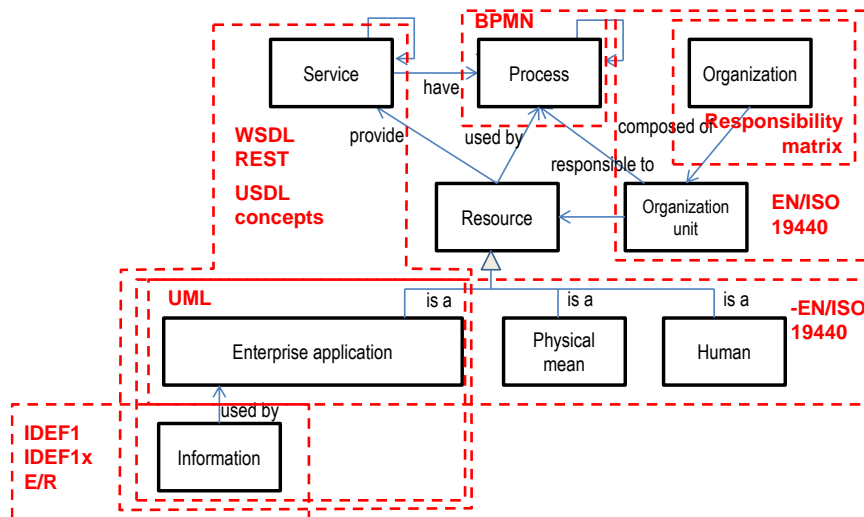


Σχήμα 4: Στοιχεία του επιπέδου BSM στη μεθοδολογία MDSEA και προτεινόμενες μέθοδοι, εργαλεία και πρότυπα μοντελοποίησης (Ducq, Chen, & Alix, 2012)

Technology Independent Model (TIM):

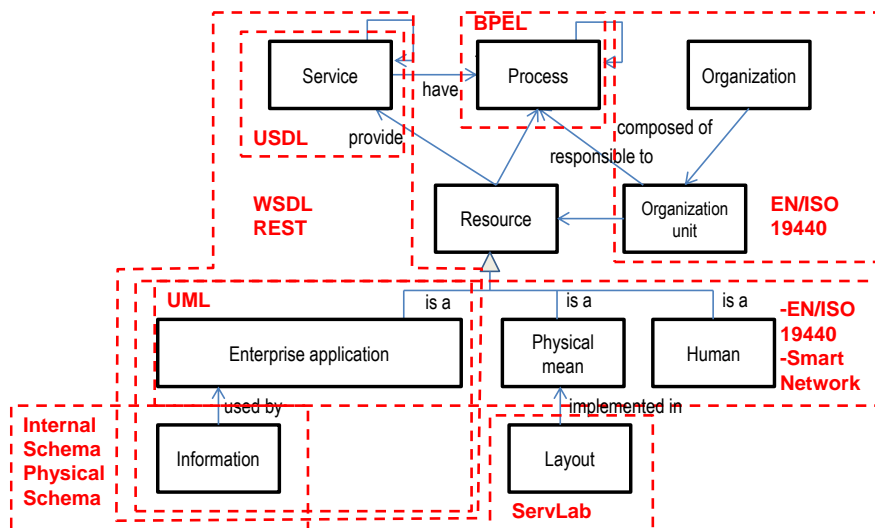
Στο επίπεδο αυτό μοντελοποιούνται τα στοιχεία του συστήματος υπηρεσιών σε υψηλό επίπεδο αφαίρεσης, δηλαδή, αλλά χωρίς να περιλαμβάνονται λεπτομέρειες για τις τεχνικές / οργανωτικές μεθόδους που με τις οποίες θα υλοποιηθούν. Σε αυτό το στάδιο, η μοντελοποίηση χωρίζεται σε τρεις διακριτές και παράλληλες συνιστώσες, η κάθε μία εκ των οποίων αναπαριστά μία πλευρά του συστήματος υπηρεσιών:

- Τεχνολογίες πληροφορικής: Τα πληροφοριακά συστήματα και διαδικασίες ΤΠΕ που υλοποιούν τις παρεχόμενες υπηρεσίες.
- Οργάνωση και ανθρώπινοι πόροι: το προσωπικό και οι επιχειρησιακές διαδικασίες που υλοποιούν το σύστημα υπηρεσιών, όπως οι ρόλοι και οι γενικές περιγραφές των απαιτούμενων εργασιών.
- Φυσικά μέσα: Οι φυσικές υποδομές που απαιτούνται, όπως τα κτίρια και ο υλικοτεχνικός εξοπλισμός.



Σχήμα 5: Στοιχεία του επιπέδου TIM στη μεθοδολογία MDSEA και προτεινόμενες μέθοδοι, εργαλεία και πρότυπα μοντελοποίησης (Ducq, Chen, & Alix, 2012)

Technology Specific Model (TSM): Στο επίπεδο αυτό τα μοντέλα προέρχονται από τα μοντέλα TIM αλλά περιλαμβάνουν πλέον πληροφορία σχετικά με τα μέσα που στηρίζουν την λειτουργία του Συστήματος Υπηρεσιών. Στο επίπεδο αυτό, όπως και στη προηγούμενη περίπτωση, τα μοντέλα μπορούν να προκύπτουν από μετασχηματισμούς και εμπλουτισμό περαιτέρω πληροφορίας σε μοντέλα του προηγούμενου επιπέδου. Τα μοντέλα στο επίπεδο TSM πρέπει να παρέχουν επαρκή πληροφορία ώστε να μπορεί πλέον να υλοποιηθεί το σύστημα, π.χ. την ανάπτυξη εφαρμογών, την πρόσληψη και την εκπαίδευση προσωπικού, τον σχεδιασμό μηχανημάτων, την προμήθεια υλικών, την σύναψη συμβάσεων κτλ. με σκοπό την παροχή της υπηρεσίας στους πελάτες. Για παράδειγμα, τα οργανωτικά μοντέλα περιέχουν συγκεκριμένες αναθέσεις ρόλων σε προσωπικό και λεπτομερείς επιχειρησιακούς κανόνες με βάση την πρακτική υλοποίηση του συστήματος. Τα μοντέλα των φυσικών μέσων περιγράφουν πραγματικές τοποθεσίες, χώρους και εξοπλισμό. Στη συνιστώσα των ΤΠΕ, το επίπεδο TSM περιλαμβάνει περιγραφές των απαιτούμενων συστημάτων προσαρμοσμένες σε συγκεκριμένες τεχνολογικές πλατφόρμες, όπως γλώσσες προγραμματισμού και υπολογιστικά συστήματα.



Σχήμα 6: Στοιχεία του επιπέδου TSM στη μεθοδολογία MDSEA και προτεινόμενες μέθοδοι, εργαλεία και πρότυπα μοντελοποίησης (Ducq, Chen, & Alix, 2012)

Η MDSEA διαθέτει αρκετά πλεονεκτήματα ως βάση για μία μεθοδολογία ανάπτυξης λογισμικού για συστήματα υπηρεσιών:

- Η μεθοδολογία, σε αντίθεση με αντίστοιχες στη τρέχουσα βιβλιογραφία, περιλαμβάνει την ανάπτυξη λογισμικού στα πλαίσια πολύπλοκων συστημάτων υπηρεσιών, και μάλιστα παράλληλα με τις οργανωσιακές και τεχνικές τους πλευρές.
- Η μεθοδολογία βασίζεται σε έννοιες από η μεθοδολογία ανάπτυξης λογισμικού MDA. Κατά συνέπεια, στην MDSEA, τα τελικά στάδια της συνιστώσας των ΤΠΕ στο πεδίο του λογισμικού αφορούν στον μετασχηματισμό μοντέλων επιπέδου TIM σε μοντέλα TSM και εκτελέσιμες εφαρμογές. Λειτουργικά, το στάδιο αυτό είναι πανομοιότυπο με τον μετασχηματισμό PIM-PSM-«εκτελέσιμος κώδικας» στα τελικά στάδια της MDA. Άρα, για την υλοποίηση του σταδίου ανάπτυξης λογισμικού του συστήματος υπηρεσιών, μπορούν να χρησιμοποιηθούν δοκιμασμένες τεχνικές και εργαλεία από την MDA.
- Η MDSEA δεν θέτει περιορισμούς σχετικά με τις μεθόδους και τα εργαλεία για την υλοποίηση των διάφορων σταδίων της. Είναι δυνατή η χρήση διάφορων εργαλείων και ροών εργασίας με μοντέλα, και η προσαρμογή τους στις απαιτήσεις συγκεκριμένων πεδίων και εφαρμογών.
- Η MDSEA ενθαρρύνει την ευθυγράμμιση των επιχειρησιακών στόχων και του συστήματος υπηρεσιών μέσω της συνολικής μοντελοποίησης του συνολικού συστήματος από τα αρχικά στάδια σχεδιασμού. Ο σχεδιασμός του συστήματος στα ύστερα στάδια της διαδικασίας προέρχεται από την ενοποιημένη εικόνα στα αρχικά στάδια (στο επίπεδο BSM), ενώ οι τρεις συνιστώσες (ΤΠΕ, οργανωσιακή και υλικοτεχνική) συσχετίζονται μεταξύ τους και ενημερώνονται κατά την διαδικασία ανάπτυξης.
- Στο πλαίσιο της MDSEA, οι τεχνικοί, επιχειρηματικοί και άλλοι εμπλεκόμενοι συμμετέχουν με τη δημιουργία και τροποποίηση μοντέλων του συστήματος υπηρεσιών, ή παρέχοντας εισόδους στη διαδικασία μοντελοποίησης. Για τον λόγο

αυτό, η MDSEA δίνει έμφαση στη χρήση κοινών γλωσσών και προτύπων μοντελοποίησης με στόχο την καλύτερη απεικόνιση και μετάδοση πληροφορίας.

Για την πρακτική της εφαρμογή, έχει προταθεί η χρήση συγκεκριμένων γλωσσών και τεχνικών μοντελοποίησης ανά επίπεδο και αντικείμενο. Για παράδειγμα, στο επίπεδο BSM, προτείνει τη χρήση της περιγραφής GRAI¹⁶, της γλώσσας USDL¹⁷, και εργαλείων όπως το ARIS¹⁸, στο επίπεδο TIM προτείνονται π.χ. η BPMN¹⁹, η UML²⁰, η WSDL²¹ κ.α., στο επίπεδο TSM οι BPEL²², WSDL κ.α.

Οι προτάσεις αυτές ακολουθούν δύο συμπληρωματικές προσεγγίσεις για το επίπεδο TIM, εστιάζοντας είτε σε «δομικές» είτε σε «συμπεριφορικές» πλευρές της υπηρεσίας και του συστήματος ΤΠΕ της. Όταν η υπηρεσία περιγράφεται με «δομικούς» όρους (δηλαδή εμπλεκόμενους, πόρους, συστατικά, δεδομένα και συσχετίσεις μεταξύ τους), αυτές μπορούν να αντιστοιχιστούν με οντότητες στο πεδίο των ΤΠΕ, όπως δομές δεδομένων και αντικειμενοστραφές λογισμικό. Από την άλλη, όταν το σύστημα υπηρεσιών μοντελοποιείται με «συμπεριφορικό» τρόπο, (δηλαδή εισόδους, εξόδους, διαδικασίες, κλάδους αποφάσεων κ.α.) είναι δυνατή η δημιουργία λογισμικού που θα υλοποιεί τη συγκεκριμένη συμπεριφορά: ακολουθίες επιχειρησιακών διαδικασιών, συνθέσεις υπηρεσιών λογισμικού (σε αρχιτεκτονικές SOA) και αυτόνομες εφαρμογές. (Agostinho, et al., 2013; Chen, Ducq, Doumeings, Zachariewicz, & Alix, A Model Driven Approach for the Modeling of Services in Virtual Enterprise, 2012; Ducq, Chen, & Alix, 2012)

Οι Ducq κ.α. (2012) προτείνουν, για το επίπεδο TIM, τη χρήση UML για την περιγραφή των επιχειρησιακών εφαρμογών του συστήματος υπηρεσιών και των συσχετισμένων τους δεδομένων. Από την άλλη, προτείνουν τη χρήση της BPMN για την περιγραφή των διαδικασιών του. Για το επίπεδο TSM, προτείνουν τη χρήση της UML για εφαρμογές και δεδομένα, και τη BPEL για τις επιχειρησιακές διαδικασίες. Η BPEL στη περίπτωση αυτή, θα επιτρέπει την εκτέλεσή τους από μία συμβατή μηχανή εκτέλεσης επιχειρησιακών διαδικασιών στην υποδομή ΤΠΕ του συστήματος υπηρεσιών.

Στη συνέχεια, οι ίδιοι ερευνητές (Ducq, Agostinho, Chen, Zacharewicz, & Jardim-Goncalves, 2014) εστιάζουν στη BPMN 2.0 ως το βασικό προϊόν στο επίπεδο TIM. Οι λόγοι που αναφέρονται για την επιλογή της BPMN είναι η συμβατότητα με πολλές πλατφόρμες εκτέλεσης επιχειρησιακών διαδικασιών, η ικανότητα να αναπαριστά ανθρώπινους και τεχνικούς πόρους και η συμμόρφωση με το αντίστοιχο μεταμοντέλο του OMG για την αναπαράσταση επιχειρησιακών διαδικασιών.

Οι καταλληλότερες γλώσσες μοντελοποίησης έχουν εξεταστεί και στη περίπτωση της ανάπτυξης ενός εργαλείου λογισμικού για την υποστήριξη της μεθοδολογίας MDSEA. Το

¹⁶ <http://www.pera.net/Methodologies/GRAI.html>

¹⁷ <http://www.w3.org/2005/Incubator/usdl/XGR-usdl-20111027/>

¹⁸ <http://www.ariscommunity.com/aris-express>

¹⁹ <http://www.bpmn.org/>

²⁰ <http://www.uml.org/>

²¹ <http://www.w3.org/TR/wsd1>

²² <http://searchsoa.techtarget.com/definition/BPEL>

εργαλείο υποστηρίζει την ανάπτυξη μοντέλων TIM με σκοπό την περαιτέρω χρήση τους για την ανάπτυξη λογισμικού. Εκεί, ως βέλτιστες λύσεις έχουν επιλεγεί και οι δύο προσεγγίσεις για το επίπεδο TIM: διαγράμματα UML για τα προϊόντα ΤΠΕ, και BPMN 2.0 για αναλυτικές επιχειρησιακές διαδικασίες. (Bazoun, Zacharewicz, Ducq, & Boyé, 2014; Boyé & Bazoun, Service Life-cycle Management Tool Box, 2014) .

3.5 Απαιτήσεις για μεθοδολογίες και εργαλεία ανάπτυξης λογισμικού για εφαρμογές υπηρεσιοποίησης στη βιομηχανία

Η μέχρι στιγμής έρευνα για τις απαιτήσεις για τις μεθοδολογίες και τα εργαλεία για την ανάπτυξη λογισμικού για συστήματα υπηρεσίας είναι πολύ περιορισμένη. Η μόνη προσέγγιση στη τρέχουσα βιβλιογραφία περιλαμβάνει κάποιες αρχικές προτάσεις: (Glushko, 2010):

- Υλοποίηση διεπαφών χρήστη και γραφικού περιβάλλοντος αλληλεπίδρασης.
- Γρήγορη δημιουργία προτύπων και δοκιμών ευχρηστίας διεπαφών χρήστη και αξιοπιστίας.
- Μοντελοποίηση και μεταφορά πολλαπλών επιχειρησιακών διαδικασιών, σύνθεση και ενορχήστρωση υπηρεσιών λογισμικού.
- Πολλαπλές τεχνολογικές πλατφόρμες, όπως αισθητήρες, αυτοματισμοί, φορητές εφαρμογές, επίγνωση πλαισίου κτλ.
- Εφαρμογή συστηματικών μεθόδων σχεδιασμού πληροφοριακών συστημάτων, λογισμικού και υπηρεσιοστραφών αρχιτεκτονικών

Ωστόσο, η βιβλιογραφία που αναλύθηκε στα προηγούμενα κεφάλαια μπορεί να προσφέρει επιπλέον δεδομένα για την ανάπτυξη ενός καλύτερα θεμελιωμένου συνόλου απαιτήσεων. Για παράδειγμα, φαίνεται ότι είναι κρίσιμη η επικοινωνία με τους εμπλεκόμενους στο ΣΠΥ, και ιδιαίτερα με τους ειδικούς του βιομηχανικού πεδίου εφαρμογής. Η διαδικασία ανάπτυξης των υποδομών ΤΠΕ πρέπει να διασφαλίζει τη μεταφορά των απαιτήσεων των εμπλεκόμενων από πολλαπλά πεδία εφαρμογής του ΣΠΥ (μηχανικούς, στελέχη, πελάτες, κτλ.), τη σύνθεση και ενσωμάτωση στο τελικό υποστηρικτικό λογισμικό. Η μεταφορά των απαιτήσεων χρήστη στη διαδικασία ανάπτυξης πρέπει να γίνεται με τρόπο που θα επιτρέπει τη συστηματική καταγραφή και τεκμηρίωση των απαιτήσεων, θα παρέχει διαδικασίες για δυναμικές μεταβολές τους, τη συμπερίληψη εμπλεκόμενων και πελατών, μηχανισμούς για την πιστοποίηση των απαιτήσεων (π.χ. πρωτότυπα) και υποστήριξη της κατάτμησης των λύσεων ΣΠΥ σε επαναχρησιμοποιούμενα τμήματα. (Berkovich, Leimeister, & Krccmar, 2011).

Επιπλέον, η ανάπτυξη ενός ΣΠΥ απαιτεί πολύ συχνά τη συνεργασία πολλών εμπλεκόμενων οργανισμών, στα πλαίσια μίας «εικονικής επιχείρησης». Οι εμπλεκόμενοι σε αυτή φέρνουν διαφορετικές δεξιότητες και οπτικές γωνίες στην ανάπτυξη της νέας υπηρεσίας και του επιχειρησιακού λογισμικού της (Ducq, Chen, & Alix, 2012). Οι διαδικασίες ανάπτυξης λογισμικού πρέπει να μπορούν να υποστηρίξουν τις συνεργασίες αυτές, αλλά και να αναγνωρίζουν τον ευμετάβλητο και προσωρινό χαρακτήρα τους. Για τον λόγο αυτό πρέπει να διασφαλίζουν τη δυναμική διαχείριση προνομίων ασφαλείας και πρόσβασης σε πνευματική ιδιοκτησία.

Χτίζοντας σε αυτή τη βάση, και με τα δεδομένα από τη βιβλιογραφική επισκόπηση των προηγούμενων ενοτήτων, είναι δυνατόν να εξαχθούν συγκεκριμένες βασικές απαιτήσεις για μια κατάλληλη ροή εργασιών και των εργαλείων για την ανάπτυξη λογισμικού στα πλαίσια υπηρεσιοποίησης βιομηχανικών προϊόντων.

3.5.1 Απαίτηση 1: Ευελιξία και γενικότητα εργαλείων και μεθοδολογίας

Η παροχή υπηρεσιών με τη χρήση τεχνολογίας λαμβάνει χώρα σε διάφορα «πλαίσια»: πρόσωπο με πρόσωπο, αυτοεξυπηρέτηση, «πολύ-καναλικές» υπηρεσίες, υπηρεσίες υποβάθρου και υπηρεσίες με επίγνωση πλαισίου και τύπου όπου παρέχονται (Glushko, 2010; Lim, Kim, Heo, & Kim, 2015). Το γεγονός αυτό επιβάλλει τη χρήση πολλών τύπων τεχνολογιών και πλατφορμών (π.χ. ενσωματωμένα συστήματα, φορητές συσκευές, ή τερματικά πωλήσεων), και εφαρμογών (π.χ. ιστότοποι, βάσεις δεδομένων, συστήματα υποστήριξης αποφάσεων και συστήματα χρηματοπιστωτικών συναλλαγών). Οι εμπλεκόμενοι στο «οικοςύστημα παροχής υπηρεσιών» θα χρησιμοποιούν διαφορετικά συστήματα και τεχνολογίες. Αυτό ισχύει τόσο για τους παρόχους των υπηρεσιών (π.χ. διαφορετικές πλατφόρμες, λειτουργικά συστήματα και λογισμικό υποδομής, αισθητήρες, ενεργοποιητές, έξυπνες συσκευές και υποδομές IOT – Internet of Things), όσο και για τους τελικούς πελάτες ή καταναλωτές (π.χ. κινητά τηλέφωνα, προσωπικοί υπολογιστές κ.α.).

Επίσης, είναι εμφανές ότι θα πρέπει να αξιοποιήσουν μία μεγάλη ποικιλία από τεχνολογίες, ανάλογα με τη φύση και τον τρόπο παροχής των τελικών υπηρεσιών. Άρα, χρειάζονται μεθοδολογίες και εργαλεία τα οποία θα είναι επαρκώς ευέλικτα ώστε να μπορούν να υποστηρίξουν την ανάπτυξη λογισμικού για πολλούς τύπους πλατφόρμας και τεχνολογίας με βάση τις αρχικές απαιτήσεις.

Συνεπώς, οι διαδικασίες ανάπτυξης λογισμικού θα πρέπει να παρέχουν ευελιξία στον τύπο του λογισμικού που θα μπορεί να αναπτυχθεί και συμβατότητα με κοινά αποδεκτά πρότυπα και τεχνολογίες.

Έτσι, μπορεί να διαμορφωθεί η παρακάτω απαίτηση:

<i>1. Η υπηρεσιοποίηση στη βιομηχανία απαιτεί την ανάπτυξη ενός μεγάλου εύρους εφαρμογών για την παροχή υπηρεσιών (λογιστική, συλλογή δεδομένων, ιστοσελίδες, κ.α.), συνεπώς, τα απαραίτητα εργαλεία δεν πρέπει να περιορίζονται σε ένα μόνο πεδίο εφαρμογής.</i>

3.5.2 Απαίτηση 2: Χρήση κατάλληλης μεθοδολογίας μοντελοποίησης

Ένα σύστημα υπηρεσιών είναι μία «δυναμική διαμόρφωση πόρων για τη συνδημιουργία αξίας», το οποίο περιέχει ανθρώπους, οργανισμούς, διαμοιρασμένη πληροφορία και τεχνολογία. Ένα σύστημα υπηρεσιών αυτό είναι συνδεδεμένο εσωτερικά και εξωτερικά με άλλα συστήματα υπηρεσιών (Maglio, Vargo, Caswell, & Spohrer, 2009; Vargo, Maglio, & Akaka, 2008). Η συνεργασία πολλών εταιρών (επιχειρήσεων, πελατών, προμηθευτών κ.α.) είναι απαραίτητη για την παροχή σύνθετων υπηρεσιών κατά τη μετάβαση της επιχείρησης προς μία λογική «παροχής υπηρεσιών» (Johnson & Mena, 2008; Gao, Yao, Zhu, Sun, & Lin, 2011). Τα Συστήματα Προϊόντος Υπηρεσίας (ΣΠΥ) που προσφέρονται από τη σύγχρονη υπηρεσιοποιημένη βιομηχανία περιλαμβάνουν ένα δίκτυο κατασκευαστών, παρόχους

υπηρεσιών, διοικητικό, εμπορικό και τεχνικό προσωπικό, συνεργαζόμενες επιχειρήσεις και επαγγελματίες, και, τέλος, τους πελάτες (Meier, Volker, & Funke, Industrial Product-Service Systems (IPS). Paradigm shift by mutually determined products and services, 2011; Bikfalvi, Lay, Maloca, & Waser, 2013; Reim, Parida, & Örtqvist, 2015). Η δημιουργία ενός νέου συστήματος υπηρεσιών απαιτεί την εμπλοκή οργανισμών και προσωπικού από διαφορετικά υπόβαθρα εξειδίκευσης, τεχνικά και μη-τεχνικά (Brax & Visintin, 2016).

Η εφαρμογή των ΤΠΕ έχει το αποτέλεσμα να μετατοπίζει σημαντικό βάρος στα αρχικά στάδια της διαδικασίας σχεδιασμού και ανάπτυξης του λογισμικού. Οι Heiskala, Hiekkänen και Korhonen παρατηρούν ότι ένα σύστημα λογισμικού (όπως και οποιοδήποτε τεχνολογικό σύστημα) λειτουργεί στα πλαίσια της λειτουργικότητας που ορίζει ο σχεδιασμός του. Συνεπώς, δεν είναι εφικτή η, κατά προαίρεση, προσαρμογή της υπηρεσίας που παρέχει στις ανάγκες του πελάτη, όπως μπορεί να γίνει με μία παραδοσιακή υπηρεσία που παρέχεται με ανθρώπινη προσπάθεια, δηλαδή η «συν δημιουργία αξίας» της υπηρεσίας (βλ. ενότητα 2.1). Για τον λόγο αυτό, πέφτει σημαντικό βάρος στις διαδικασίες σχεδιασμού και ανάπτυξης του συστήματος λογισμικού για την ενσωμάτωση των απαιτήσεων του πελάτη ή πελατών στο νέο σύστημα, αφού πρέπει να ενσωματώσει πλήρως την απαιτούμενη επιχειρησιακή λογική (Heiskala, Hiekkänen, & Korhonen, 2011). Συνεπώς, η ανάπτυξη λογισμικού για βιομηχανικές υπηρεσίες πρέπει να δίνει ιδιαίτερη έμφαση στις φάσεις σχεδιασμού και ανάπτυξης και ιδιαίτερα στην αποτύπωση και επικοινωνία των επιχειρησιακών προδιαγραφών, με σκοπό τη μετατροπή τους σε λειτουργικότητα ΤΠΕ.

Η δημιουργία του συστήματος υπηρεσιών απαιτεί την παράλληλη ανάπτυξη όλων των πλευρών, όπως τα φυσικά προϊόντα στα οποία αφορούν, οι οργανωτικές δομές, επιχειρησιακές διαδικασίες, τεχνολογίες πληροφορικής, και οι φυσικές υποδομές. Ο σχεδιασμός του θα πρέπει να λαμβάνει υπόψη τους ρόλους των ανθρώπων, της τεχνολογίας, των φυσικών χώρων, του εξοπλισμού και των διαδικασιών (Goldstein, Johnston, Duffy, & Rao, 2002).

Για παράδειγμα, η παροχή υπηρεσιών υποστήριξης για νέο βιομηχανικό εξοπλισμό απαιτεί τον καθορισμό μίας σειράς από στοιχεία. Αυτά περιλαμβάνουν τους τύπους των υπηρεσιών που θα παρασχεθούν, τις ίδιες τις διαδικασίες υποστήριξης. Επίσης περιλαμβάνει τους τεχνικούς και άλλο προσωπικού που θα τις υλοποιήσουν και το πώς οργανώνονται σε ομάδες (π.χ. γεωγραφικά ή βάσει ειδίκευσης), και που θα βρίσκονται. Ακόμα, πρέπει να καθοριστούν φυσικά στοιχεία όπως οι χώροι που θα υποστηρίζουν την παροχή της υπηρεσίας (γραφεία, αποθήκες, εργαστήρια) και άλλοι φυσικοί πόροι, όπως ανταλλακτικά και αναλώσιμα. Μαζί με όλα τα παραπάνω, απαιτείται και ο καθορισμός των τεχνολογιών ΤΠΕ (λογισμικού και υλικού) που θα υποστηρίζουν την παροχή των υπηρεσιών.

Στη περίπτωση ενός περισσότερο άυλου προϊόντος, η υλοποίηση ενός καταστήματος εφαρμογών («app store») για φορητές συσκευές απαιτεί πάλι τον καθορισμό των επιχειρησιακών διαδικασιών, του προσωπικού (διαχειριστές, ελεγκτές ποιότητας και προγραμματιστές), φυσικές υποδομές (π.χ. γραφεία) και υποδομές ΤΠΕ οι οποίες θα μπορούν να εξυπηρετήσουν ένα μεγάλο αριθμό τελικών χρηστών.

Και στις δύο περιπτώσεις, το σύστημα υπηρεσιών δεν θα λειτουργήσει σωστά (ή και καθόλου) αν οποιαδήποτε από αυτά τα στοιχεία λείπει ή είναι κακώς σχεδιασμένο και

υλοποιημένο. Στη περίπτωση των ΣΠΥ, οι λειτουργίες των υπηρεσιών και του προϊόντος πρέπει να είναι ολοκληρωμένες μεταξύ τους από τα πρώιμα στάδια σχεδιασμού. Ακόμα και τότε, οι σχεδιαστές των ΣΠΥ πρέπει να λαμβάνουν υπόψη τις πολλαπλές οντότητες που εμπλέκονται, τον κύκλο ζωής του συστήματος και τα αντίστοιχα δίκτυα εμπλεκόμενων (Cavalieri & Pezzota, 2012).

Με βάση τα παραπάνω, είναι προφανές ότι υπάρχει η ανάγκη συντονισμού και επικοινωνίας πολλαπλών εμπλεκόμενων στον σχεδιασμό και την υλοποίηση του συστήματος υπηρεσιών. Ανάμεσα σε αυτούς θα βρίσκονται οι μηχανικοί λογισμικού που θα σχεδιάσουν και υλοποιήσουν τα προγράμματα που θα υποστηρίξουν την παροχή της υπηρεσίας. Κατά τον σχεδιασμό, θα λαμβάνουν πληροφορίες, προτάσεις και ανάδραση από πολλές πηγές (π.χ. επιχειρησιακούς αναλυτές, μέλη του δικτύου συνεργατών του συστήματος υπηρεσίας, πελάτες, κ.α.).

Ένας από τους κρίσιμότερους παράγοντες επιτυχίας στη δημιουργία συστημάτων υπηρεσίας με ΤΠΕ είναι η συνεργασία μεταξύ των εμπλεκόμενων, καθώς και η καλή επικοινωνία κατά τη διαδικασία ανάπτυξης του νέου συστήματος. Αυτό οφείλεται όχι μόνο στην ανάγκη συνεργασίας των προγραμματιστών με ειδικούς στο βιομηχανικό πεδίο εφαρμογής, αλλά και επειδή τα νέα συστήματα ΤΠΕ που εισάγονται για την υποστήριξη της υπηρεσίας προκαλούν ριζικές αλλαγές στον τρόπο λειτουργίας των παραδοσιακών βιομηχανικών επιχειρήσεων (Westergren, 2011).

Συνεπώς, οι πληροφορίες για το πεδίο εφαρμογής, τις επιχειρησιακές διαδικασίες και τη συνολική λύση (τεχνολογική και οργανωτική) πρέπει να διατυπώνονται ξεκάθαρα και να μεταδίδονται αποτελεσματικά από τους ειδικούς του πεδίου εφαρμογής, ή άλλους εμπλεκόμενους, στους μηχανικούς λογισμικού. Αναμένεται ότι η ανάπτυξη λογισμικού για τις υπηρεσίες αυτές θα απαιτεί την συνεργασία μεταξύ των εμπλεκόμενων αυτών ή, κατ'ελάχιστο, την συγκέντρωση των ετερόκλητων και, ενδεχομένως, αντιφατικών απαιτήσεων τους για τη λειτουργικότητα του λογισμικού υπηρεσιών.

Η ανάγκη για συνεργατικές προσεγγίσεις έχει παρατηρηθεί σε πολλές πλευρές της ανάπτυξης λογισμικού για τη βιομηχανία και την παροχή υπηρεσιών. Για παράδειγμα, έχει διαπιστωθεί στην ανάπτυξη της ασφάλειας για λογισμικό εφοδιαστικών αλυσίδων (Bartol, 2014), την ανάπτυξη προϊόντων «εντάσεως λογισμικού» (Pernstål, Gorschek, Feldt, & Florén, 2015) και την αυτοκινητοβιομηχανία (Lim, Kim, Heo, & Kim, 2015).

Έτσι, χρειάζεται η απρόσκοπτη και συστηματική μεταφορά των επιχειρησιακών απαιτήσεων της υποστηριζόμενης υπηρεσίας και της γνώσης των ειδικών του βιομηχανικού πεδίου εφαρμογής στη διαδικασία ανάπτυξης λογισμικού. Οι μεθοδολογίες και τα εργαλεία ανάπτυξης πρέπει να μπορούν να υποστηρίξουν τη συνεργασία μεταξύ των εμπλεκόμενων π.χ. με την εφαρμογή κοινής γλώσσας ή με την παροχή εργαλείων επικοινωνίας και συνεργασίας.

Η χρήση κατάλληλων γλωσσών και εργαλείων μοντελοποίησης θα συμβάλλει σε αυτό το σκοπό. Στο πεδίο της ανάπτυξης λογισμικού, η «μηχανική με χρήση μοντέλων» (Model-Driven Engineering – MDE) στοχεύει στη μείωση της απόστασης μεταξύ του πεδίου εφαρμογής και του λογισμικού. Αυτό επιτυγχάνεται με τη χρήση μοντέλων που

αναπαριστούν το σχεδιαζόμενο σύστημα σε διάφορα επίπεδα αφαίρεσης και από διαφορετικές οπτικές γωνίες. Η MDE έχει φανεί χρήσιμη σε οργανισμούς που αναπτύσσουν λογισμικό σε κλάδους εκτός της καθαρής μηχανικής λογισμικού. Η χρήση μοντέλων διευκολύνει την επικοινωνία μεταξύ των ειδικών του πεδίου εφαρμογής (π.χ. μηχανικούς και στελέχη μάρκετινγκ) και τους προγραμματιστές που αναπτύσσουν το λογισμικό για τα προϊόντα, υπηρεσίες και δραστηριότητες της επιχείρησης (Hutchinson, Rouncefield, & Whittle, 2011). Μάλιστα, η εφαρμογή «μετά-μοντέλων», δηλαδή μοντέλα για τις οντότητες, τις σχέσεις και αλληλεπιδράσεις σε ένα πεδίο εφαρμογής ή κλάδο γενικότερα (σε αντίθεση με μία συγκεκριμένη εφαρμογή) τυποποιεί την εφαρμογή σε συγκεκριμένα πεδία.

Μία προσέγγιση ανάλυσης απαιτήσεων και ανάπτυξης βασισμένη σε μοντελοποίηση, θα διευκόλυνε την επικοινωνία μεταξύ των πολλαπλών εμπλεκόμενων. Επιπλέον, θα ήταν κατάλληλη για την υλοποίηση λογισμικού με υψηλό βαθμό αφαίρεσης, όπως επιτάσσουν οι τεχνολογίες SOA και BPM, με τις οποίες είναι ιδιαίτερα συμβατές (Ameller, et al., 2015). Μάλιστα, η χρήση μοντέλων επιτρέπει τη συνέπεια του παραγόμενου λογισμικού σε πολλαπλές τεχνολογικές πλατφόρμες. (Francese, Risi, Scanniello, & Tortora, 2015). Από τις διαθέσιμες μεθοδολογίες, η MDSEA συνδυάζει τη χρήση μοντέλων για την ανάπτυξη συστημάτων υπηρεσιών και την υποστήριξη για την ανάπτυξη λογισμικού. Έτσι, διαμορφώνεται η παρακάτω απαίτηση:

2. Καθώς η (βασισμένη στην MDA) MDSEA είναι η μόνη μεθοδολογία ανάπτυξης υπηρεσιών με μοντελοποίηση που υποστηρίζει αξιόπιστα την ανάπτυξη λογισμικού, οι υποψήφιες προσεγγίσεις θα πρέπει να είναι συμβατές με αυτή. Αυτό επιτυγχάνεται με το να υποστηρίζουν και να υλοποιούν μετασχηματισμούς μοντέλων τύπου MDA.

3.5.3 Απαίτηση 3: Υποστήριξη δομικής και συμπεριφορικής προσέγγισης

Τα συστήματα υπηρεσιών είναι μία ειδική περίπτωση επιχειρήσεων, και η μοντελοποίηση συστημάτων υπηρεσιών είναι υποσύνολο του κλάδου της Επιχειρησιακής Μοντελοποίησης (Enterprise Modelling). Η Επιχειρησιακή Μοντελοποίηση ενθαρρύνει τη διάκριση μεταξύ των «δομικών» και των «συμπεριφορικών» στοιχείων του επιχειρησιακού συστήματος (Jonkers, et al., 2005). Τα δομικά μοντέλα ενός συστήματος περιλαμβάνουν τις οντότητες, τις σχέσεις και τις έννοιες μέσα σε αυτό (π.χ. κλάσεις οντοτήτων). Από την άλλη, τα «συμπεριφορικά» μοντέλα περιγράφουν δράσεις, συμβάντα και αλλαγές στη δομή του συστήματος (Oliné, 2007).

Για παράδειγμα, το πρότυπο UML (Rumbaugh, Jacobson, & Booch, 2004) διαχωρίζει τυπικά τις μορφές μοντελοποίησης που περιλαμβάνει ως δομικές ή συμπεριφορικές. Ένα δομικό μοντέλο αποτυπώνει πληροφορία για τις οντότητες του συστήματος, η οποία μπορεί να είναι ακριβής για μια δεδομένη χρονική στιγμή. Αντίθετα, ένα συμπεριφορικό μοντέλο περιγράφει τον τρόπο με τον οποίο οι οντότητες του συστήματος αλλάζουν στη διάρκεια του χρόνου (Object Management Group, 2015).

Η MDSEA έχει μελετηθεί σε σχέση και με τους δύο τύπους μοντελοποίησης: Δομικά διαγράμματα UML και επιχειρησιακές διαδικασίες σε BPMN σε περισσότερο υψηλά επίπεδα αφαίρεσης, και UML/BPEL σε χαμηλότερα (Ducq, Chen, & Alix, 2012; Bazoun, Zacharewicz, Ducq, & Boyé, 2014).

Κατά συνέπεια, εφόσον είναι επιθυμητή η καλύτερη και πληρέστερη δυνατή περιγραφή του συστήματος υπηρεσιών, η προσέγγιση μοντελοποίησης που θα υιοθετηθεί θα πρέπει να συνδυάζει δομικά και συμπεριφορικά στοιχεία του συστήματος υπηρεσιών. Για τον λόγο αυτό, διαμορφώνεται η ακόλουθη απαίτηση:

3. Η μοντελοποίηση ενός συστήματος υπηρεσιών επωφελείται από τη χρήση των συμπληρωματικών προσεγγίσεων της δομικής και συμπεριφορικής μοντελοποίησης για την ανάπτυξη λογισμικού. Συνεπώς, και οι δύο προσεγγίσεις πρέπει να υποστηρίζονται.

3.5.4 Απαίτηση 4: Ολοκλήρωση εργαλείων ανάπτυξης

Συγκεκριμένα, οι βιομηχανικές εταιρείες που αναπτύσσουν λογισμικό πρέπει να αναπτύξουν μία σειρά από δεξιότητες για τη δημιουργία λογισμικού για μαζική πώληση, την ανάπτυξη λογισμικού ως υπηρεσία για τρίτους, και την ανάπτυξη λογισμικού σαν μέρος επιχειρησιακής λύσης. Σε όλες τις περιπτώσεις, κρίνεται απαραίτητη μία σειρά από ικανότητες, μεταξύ των οποίων περιλαμβάνεται, η τεχνογνωσία ανάπτυξης, η κατανόηση των αναγκών του τελικού χρήστη, και η διαχείριση έργων ανάπτυξης λογισμικού. Σε περιπτώσεις όπου το λογισμικό είναι τυποποιημένο, περιλαμβάνεται, μεταξύ άλλων, ο έλεγχος εκδόσεων λογισμικού και η σχετική τεκμηρίωση, ενώ στην παροχή υπηρεσιών ανάπτυξης λογισμικού για τρίτους, η ανάγκη για συστηματική καταγραφή και ανάλυση απαιτήσεων των τελικών χρηστών. Στη περίπτωση παροχής επιχειρησιακών λύσεων, απαιτείται η δυνατότητα προσαρμογής διασύνδεσης του παραγόμενου λογισμικού με πολλαπλά συστήματα και τεχνολογίες (Värgrynen, 2010).

Οι βιομηχανικές επιχειρήσεις αναπτύσσουν λογισμικό εφόσον απαιτηθεί, αλλά δεν αποτελεί κύρια δραστηριότητά τους. Η ανάπτυξη του λογισμικού αποτελεί δευτερεύουσα δραστηριότητα και υποστηρικτική στη κύρια αλυσίδα αξίας της βιομηχανικής επιχείρησης. Για τον λόγο αυτό, όταν αποφασιστεί η μετάβαση προς ένα «υπηρεσιοποιημένο» επιχειρηματικό μοντέλο, οι δυνατότητες και ικανότητες για την εσωτερική ανάπτυξη λογισμικού μπορεί να είναι περιορισμένες (Wallin, Parida, & Isaksson, 2015) ή εστιασμένες σε πεδία εκτός των επιχειρησιακών εφαρμογών για παροχή υπηρεσιών. Το προσωπικό τους μπορεί επίσης να έχει περιορισμένη πρόσβαση σε κατάλληλα εργαλεία και τεχνογνωσία. Στις περισσότερες περιπτώσεις, οι λειτουργίες της ανάπτυξης λογισμικού βρίσκονται σε δεύτερη μοίρα σε σχέση με την κύρια δραστηριότητα της επιχείρησης, δηλαδή την παραγωγή υλικών αγαθών. Για τον λόγο αυτό, τα συγκεκριμένα τμήματα ενδέχεται να λειτουργούν κάτω από πιέσεις χρόνου, κόστους και ελλείψεως ανθρώπινου δυναμικού.

Πολλές φορές, η ανάπτυξη λογισμικού ανατίθεται σε κάποιο περισσότερο εξειδικευμένο εξωτερικό συνεργάτη, κάτι που απαιτεί την ανάπτυξη αποτελεσματικών καναλιών επικοινωνίας και συνεργασίας. Ακόμα όμως και στη περίπτωση της εξωτερικής ανάθεσης, απαιτείται στενή συνεργασία και καλή επικοινωνία με την ανάδοχο εταιρεία – η οποία όμως δυσχεραίνεται ακόμα περισσότερο από τη «διοικητική» απόσταση (καθώς πρόκειται για άλλους οργανισμούς), τους διαφορετικούς κώδικες επικοινωνίας και τις διαφορετικές εξειδικεύσεις των εμπλεκόμενων στελεχών. Επιπλέον, δεδομένου ότι αναμένεται η διαχρονική εξέλιξη της μορφής και λειτουργίας του ΣΠΥ, η ανάδοχος εταιρεία θα πρέπει να συμμετέχει στην τροποποίηση του νέου λογισμικού. Αυτό σημαίνει ότι και η ίδια εντάσσεται πλέον στο «οικοσύστημα βιομηχανικών υπηρεσιών» που υλοποιεί το ΣΠΥ, άρα πρέπει να

διασφαλιστούν τα μέσα για την ομαλοποίηση της συνεργασίας, τόσο με την αναθέτουσα (η οποία μπορεί να παίζει κύριο ή δευτερεύοντα ρόλο στο συγκεκριμένο συνεργατικό δίκτυο), όσο και με τους άλλους εμπλεκόμενους.

Για τους παραπάνω λόγους, οι βιομηχανικές επιχειρήσεις πρέπει να αποκτήσουν τα κατάλληλα εργαλεία για την ανάπτυξη λογισμικού ή για τη συμμετοχή σε μία τέτοια διαδικασία (π.χ. ώστε να συνεργάζονται με εξωτερικούς υπεργολάβους και προμηθευτές εφαρμογών). Η ροή εργασιών ανάλυσης απαιτήσεων και ανάπτυξης λογισμικού πρέπει να μπορεί να εγκατασταθεί εύκολα, με στάδια ανάπτυξης και εργαλεία που θα χαρακτηρίζονται από ομαλή διαλειτουργικότητα. Ταυτόχρονα, πρέπει να παρέχει επαρκή ευελιξία για όλα τα απαιτούμενα τελικά προϊόντα (για παράδειγμα εφαρμογές υπηρεσίες λογισμικού, συνθέσεις υπηρεσιών και βάσεις δεδομένων). Αυτό υποδεικνύει την ανάγκη για την εφαρμογή των αρχών της «ολοκλήρωσης εργαλείων μηχανικής λογισμικού».

Για τη βελτίωση της παραγωγικότητας και διευκόλυνση του προγραμματιστή, έχει προταθεί η ενσωμάτωση των σχετικών εργαλείων λογισμικού σε αλυσίδες εργαλείων και ενιαία περιβάλλοντα, τα Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης (Integrated Development Environments - IDEs). Αυτά έχουν τη δυνατότητα να αυξήσουν την παραγωγικότητα του προγραμματιστή μέσω της ολοκλήρωσης διαφορετικών πλευρών της διαδικασίας ανάπτυξης (πλατφόρμας, δεδομένων, παρουσίασης, ελέγχου, δεδομένων και διαδικασίας).

Αυτή η ολοκλήρωση εργαλείων μπορεί να περιλαμβάνει «κάθεται» και «οριζόντια» εργαλεία. Τα «κάθεται» εργαλεία διαχειρίζονται συγκεκριμένα στάδια της ροής εργασιών (όπως εργαλεία μοντελοποίησης, επεξεργαστές κώδικα και μεταγλωττιστές). Από την άλλη, τα «οριζόντια» εργαλεία παρέχουν υποστήριξη για το σύνολο της διαδικασίας (όπως η διαχείριση του χώρου εργασίας, ο έλεγχος εκδόσεων και η σύνταξη εγγράφων τεκμηρίωσης) (Wasserman, 1990; Thomas & Nejmeh, 1992). Τα Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης (Integrated Development Environments – IDEs) εκπροσωπούν το υψηλότερο επίπεδο ολοκλήρωσης εργαλείων.

Με βάση μπορούμε να υποθέσουμε ότι η ανάπτυξη λογισμικού για συστήματα υπηρεσιών και την υπηρεσιοποιημένη βιομηχανία θα εξυπηρετούνταν από την χρήση ολοκληρωμένων εργαλείων, και ειδικότερα IDEs. Τα ολοκληρωμένα εργαλεία μπορούν να εγκατασταθούν εύκολα και μπορούν να προσαρμοστούν σε διάφορα πεδία εφαρμογής και τεχνολογίες. Επίσης, καθώς τα συστήματα υπηρεσιών δημιουργούνται σε συνεργατικά περιβάλλοντα, υποστήριξη της ανάπτυξης λογισμικού από πολλαπλούς συμμετέχοντες, ενδεχομένως από διαφορετικούς οργανισμούς. Οι συνεργασίες αυτές είναι δυναμικές και μεταβάλλονται κατά περίπτωση. Πρέπει να υποστηρίζεται τόσο η κοινή πρόσβαση σε πόρους όσο και η προστασία των πόρων αυτών από μη-εξουσιοδοτημένη πρόσβαση. Τα IDEs καλύπτουν και αυτή την περίπτωση, καθώς μπορούν να περιλαμβάνουν εργαλεία συνεργασίας, όπως κοινά αποθετήρια, dashboards και άμεσα μηνύματα.

Με βάση τα παραπάνω, διαμορφώνεται η ακόλουθη απαίτηση:

4. Τα Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης (IDE – Integrated Development Environments) είναι προτιμότερα από αλυσίδες διακριτών εργαλείων, καθώς αυξάνουν σημαντικά την παραγωγικότητα και είναι ευκολότερα στη διαχείριση.

3.5.5 Απαίτηση 5: Ολοκλήρωση ροής και εργαλείων ανάπτυξης με το σύστημα εκτέλεσης του λογισμικού υπηρεσιών

Ο «χρόνος διάθεσης στην αγορά» (time-to-market) των κύριων προϊόντων και υπηρεσιών είναι κρίσιμος παράγοντας για την επίτευξη ανταγωνιστικού πλεονεκτήματος. Καθώς όμως απαιτείται λογισμικό για την υποστήριξη του συστήματος υπηρεσιών, ο χρόνος διάθεσης στην αγορά ορίζει και τον διαθέσιμο χρόνο για την ανάπτυξη του υποστηρικτικού λογισμικού. Σε ένα τέτοιο περιβάλλον, είναι ζωτικής σημασίας η δυνατότητα δοκιμής προτύπων η μείωση του «χρόνου-προς-την-αγορά» («time-to-market»), δηλαδή του χρόνου που χρειάζεται για να λανσαριστεί η νέα υπηρεσία (Van Riel & Lievens, 2004; Mietinnen, Rontti, Kuure, & Lindström, 2012). Οι κύκλοι ανάπτυξης του λογισμικού παροχής υπηρεσιών εξαρτώνται από την εξέλιξη των προσφερόμενων υπηρεσιών, η οποία με τη σειρά της, ανταποκρίνεται στις πιέσεις του ανταγωνισμού. Το λογισμικό υπηρεσιών δεν μπορεί να θεωρηθεί «ολοκληρωμένο», όσο το σύστημα υπηρεσιών που υποστηρίζει συνεχίζει να εξελίσσεται (Chae, 2014)

Έτσι, πρέπει να είναι δυνατή η γρήγορη δημιουργία πρωτοτύπων εφαρμογών, οι οποίες θα χρησιμοποιούνται στη πιστοποίηση και ανάπτυξη των κύριων προϊόντων / υπηρεσιών του συστήματος. Η ανάγκη αυτή μπορεί να εξυπηρετηθεί εφόσον τα εργαλεία ανάπτυξης διαλειτουργούν με την υποδομή ΤΠΕ του συστήματος υπηρεσιών (εξυπηρετητές, μηχανές εκτέλεσης επιχειρησιακών διαδικασιών κ.α.). Έτσι, ενοποιώντας την ανάπτυξη λογισμικού με την θέση σε παραγωγική λειτουργία θα προσφέρουν μηχανισμούς για την συνεχή μετάδοση νέων απαιτήσεων για το λογισμικό και την παραγωγή νέων εκδόσεων, με ταχύ, ευέλικτο και αξιόπιστο τρόπο. Έτσι, διαμορφώνεται η παρακάτω απαίτηση:

5. Η χρησιμότητα των περιβάλλοντων μοντελοποίησης και ανάπτυξης για συστήματα υπηρεσιών αυξάνεται σημαντικά από την ολοκλήρωσή τους με το περιβάλλον εκτέλεσης εφαρμογών (τις υποδομές ΤΠΕ) του συστήματος υπηρεσιών, δίνοντας τη δυνατότητα για τη γρήγορη εκκίνηση της παραγωγικής λειτουργίας των νέων εφαρμογών.

3.5.6 Απαίτηση 6: Τεχνολογική υποδομή: SOA και BPM

Ένα σύννηθες χαρακτηριστικό των υποδομών ΤΠΕ για επιχειρήσεις είναι εφαρμογή των αρχών της υπηρεσιοστραφούς αρχιτεκτονικής (Service-Oriented Architectures - SOA), δηλαδή της σύνθεσης μονάδων λογισμικού (υπηρεσιών) για την δημιουργία περίπλοκων εφαρμογών (Rosen, Lublinsky, Smith, & Balcer, Applied SOA: Service Oriented Architecture and Design Strategies, 2012; Huhns & Singh, 2005; Papazoglou & Georgakopoulos, 2003). Οι περισσότερες πλατφόρμες που χρησιμοποιούν τις αρχές της SOA στηρίζονται σε web services και τις σχετικές τεχνολογίες όπως REST, SOAP, XML, WSDL και άλλες (Rosen, Lublinsky, Smith, & Balcer, 2012). Τα συστήματα SOA συσχετίζονται με τεχνολογίες διαχείρισης επιχειρησιακών διαδικασιών (Business-Process Management – BPM), οι οποίες αποσκοπούν στην αποτύπωση και εκτέλεση των διαδικασιών αυτών ως σειρά εργασιών, πολλές εκ των οποίων είναι υπηρεσίες τύπου SOA. Οι τεχνολογίες αυτές χρησιμοποιούν αφαιρετικές αναπαραστάσεις όπως η Business Process Model and Notation (BPMN) (White, 2008) και άλλες. Λόγω της έμφασης στην αφαίρεση, οι μεθοδολογίες ανάπτυξης λογισμικού με

μοντέλα είναι κατάλληλες για την ανάπτυξη συστημάτων τύπου SOA και BPM, πεδίο στο οποίο υπάρχει σημαντική ερευνητική δραστηριότητα (Ameller, et al., 2015).

Η SOA και η BPM προσπαθούν να βελτιώσουν τη σύνδεση των επιχειρησιακών διαδικασιών με το λογισμικό μέσω της αντιστοίχισης με τα στοιχεία του πεδίου εφαρμογής. Επίσης, η χαλαρή διασύνδεση των συστατικών των συστημάτων SOA και BPM σημαίνει ότι μπορούν να αναδιαμορφωθούν εύκολα για την παραγωγή νέας λειτουργικότητας, άρα και να ανταποκριθούν στον ανταγωνισμό και τις ανάγκες του πελάτη. Τέλος οι τεχνολογίες αυτές είναι ώριμες και ευρέως αποδεκτές μέθοδοι για τον σχεδιασμό και υλοποίηση πληροφοριακών συστημάτων, ειδικά στον τομέα των επιχειρήσεων. Έτσι, όταν εξετάζεται το μελλοντικό περιβάλλον ΤΠΕ της παροχής υπηρεσιών, είναι λογικό να αναμένεται η ευρεία χρήση των τεχνολογιών SOA και BPM.

Έτσι, μπορεί να εξαχθεί η ακόλουθη απαίτηση:

<i>6. Η τεχνολογική υποδομή της υπηρεσιοποίησης στη βιομηχανία θα στηρίζεται σε τεχνολογίες υπηρεσιοστραφών αρχιτεκτονικών (SOA) και διαχείρισης επιχειρησιακών διαδικασιών (BPM).</i>
--

3.5.7 Συμπεράσματα

Με βάση τα δεδομένα των προηγούμενων ενοτήτων, είναι δυνατόν να εξαχθούν συμπεράσματα για το επιχειρησιακό και τεχνολογικό περιβάλλον της ανάπτυξης λογισμικού για την παροχή υπηρεσιών στη βιομηχανία. Τα συμπεράσματα αυτά μπορούν να συμπυκνωθούν σε έξι βασικές απαιτήσεις για μία μεθοδολογία ανάπτυξης λογισμικού και για τα αντίστοιχα εργαλεία (Πίνακας 5).

<i>1. Η υπηρεσιοποίηση στη βιομηχανία απαιτεί την ανάπτυξη ενός μεγάλου εύρους εφαρμογών για την παροχή υπηρεσιών (λογιστική, συλλογή δεδομένων, ιστοσελίδες, κ.α.), συνεπώς, τα απαραίτητα εργαλεία δεν πρέπει να περιορίζονται σε ένα μόνο πεδίο εφαρμογής.</i>
<i>2. Καθώς η (βασισμένη στην MDA) MDSEA είναι η μόνη μεθοδολογία ανάπτυξης υπηρεσιών με μοντελοποίηση που υποστηρίζει αξιόπιστα την ανάπτυξη λογισμικού, οι υποψήφιες προσεγγίσεις θα πρέπει να είναι συμβατές με αυτή. Αυτό επιτυγχάνεται με το να υποστηρίζουν και να υλοποιούν μετασχηματισμούς μοντέλων τύπου MDA.</i>
<i>3. Η μοντελοποίηση ενός συστήματος υπηρεσιών επωφελείται από τη χρήση των συμπληρωματικών προσεγγίσεων της δομικής και συμπεριφορικής μοντελοποίησης για την ανάπτυξη λογισμικού. Συνεπώς, και οι δύο προσεγγίσεις πρέπει να υποστηρίζονται.</i>
<i>4. Τα Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης (IDE – Integrated Development Environments) είναι προτιμότερα από αλυσίδες διακριτών εργαλείων, καθώς αυξάνουν σημαντικά την παραγωγικότητα και είναι ευκολότερα στη διαχείριση.</i>
<i>5. Η χρησιμότητα των περιβάλλοντων μοντελοποίησης και ανάπτυξης για συστήματα υπηρεσιών αυξάνεται σημαντικά από την ολοκλήρωσή τους με το περιβάλλον εκτέλεσης εφαρμογών (τις υποδομές ΤΠΕ) του συστήματος υπηρεσιών, δίνοντας τη δυνατότητα για τη γρήγορη εκκίνηση της παραγωγικής λειτουργίας των νέων εφαρμογών.</i>
<i>6. Η τεχνολογική υποδομή της υπηρεσιοποίησης στη βιομηχανία θα στηρίζεται σε τεχνολογίες υπηρεσιοστραφών αρχιτεκτονικών (SOA) και διαχείρισης επιχειρησιακών διαδικασιών (BPM).</i>

Πίνακας 5: Απαιτήσεις για μία μεθοδολογία και το αντίστοιχο περιβάλλον ανάπτυξης για ενσωμάτωση υπηρεσιών σε βιομηχανικά προϊόντα.

3.6 Μεθοδολογίες και εργαλεία ανάπτυξης λογισμικού για συστήματα SOA και BPM

3.6.1 Μεθοδολογίες και εργαλεία

Όπως γίνεται φανερό από τις απαιτήσεις της προηγούμενης ενότητας, τα πεδία των MDE, SOA, BPM και Ολοκλήρωσης Εργαλείων μπορούν να συμβάλλουν στην ανάπτυξη των απαραίτητων μεθοδολογιών και εργαλείων. Για τον λόγο αυτό, πρέπει να εξεταστεί η δυνατότητα εφαρμογής προσεγγίσεων που συνδυάζουν:

- Μηχανική με χρήση μοντέλων (Model-Driven Engineering – MDE): Χρήση μοντελοποίησης για τη μετάδοση και καταγραφή των απαιτήσεων, την παραγωγή κώδικα και εκτελέσιμων.
- Υπηρεσιοστραφή Αρχιτεκτονική (Service-Oriented Architectures – SOA): Στόχευση στη παραγωγή λογισμικού για υπηρεσιοστραφείς πλατφόρμες
- Διαχείριση Επιχειρησιακών Διαδικασιών (Business Process Management – BPM): Στόχευση πλατφορμών με δυνατότητες διαχείρισης επιχειρησιακών διαδικασιών
- Ολοκλήρωση Εργαλείων Ανάπτυξης Λογισμικού: συνδυασμένα εργαλεία ή αλυσίδες εργαλείων, με έμφαση σε Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης (IDE)
- Συνδυασμό μεθοδολογίας και εργαλείων: Ερευνητικές προσεγγίσεις που συνδυάζουν ροές εργασιών (παραδοσιακές ή νέες) με τα εργαλεία που απαιτούνται για την υλοποίησή τους.

Οι προσεγγίσεις που πληρούν τα παραπάνω χαρακτηριστικά θα καλύπτουν, εξ'ορισμού, την απαίτηση 6 της προηγούμενης ενότητας, για την υποστήριξης τεχνολογιών SOA και BPM. Έτσι, θα πρέπει να αξιολογηθούν με βάση και τις υπόλοιπες πέντε βασικές απαιτήσεις.

Η ανάγκη για αξιοποίηση της συνέργειας μεταξύ της ανάπτυξης λογισμικού με μοντέλα και των υπηρεσιοστραφών αρχιτεκτονικών έχει οδηγήσει την έρευνα και την παραγωγή εργαλείων, αλυσίδων εργαλείων και ολοκληρωμένων περιβαλλόντων ανάπτυξης για τον σκοπό αυτό. Οι προσπάθειες αυτές προέρχονται από την αγορά, τις πρωτοβουλίες ανοικτού λογισμικού αλλά και από την ακαδημαϊκή έρευνα. Συνήθως, τα εργαλεία αυτά συνδέονται με συγκεκριμένες μεθοδολογίες ανάπτυξης, αλλά υπάρχουν και πλαίσια που επιτρέπουν την σύνθεση μεθόδων και διαδικασιών ανάπτυξης ανάλογα με τις απαιτήσεις του προβλήματος. Επιπλέον, υπάρχουν εργαλεία που απευθύνονται σε συγκεκριμένα τελικά πεδία εφαρμογής (π.χ. τηλεπικοινωνίας) αλλά και εργαλεία γενικού σκοπού.

Μεγάλο μέρος των τρεχόντων εργαλείων για την ανάπτυξη λογισμικού με μοντέλα σχετίζονται με το πλαίσιο ανοικτού κώδικα Eclipse²³. Το λογισμικό Eclipse αποτελεί μία ανοικτή πλατφόρμα ανάπτυξης λογισμικού, η οποία παρέχει ένα παραμετροποιήσιμο περιβάλλον ανάπτυξης. Το περιβάλλον αυτό μπορεί να τροποποιηθεί και να επεκταθεί με

²³ <https://eclipse.org/>

κατάλληλα πρόσθετα λογισμικού και εφαρμογές (plugins) ώστε να προσαρμόζεται σε διαφορετικούς τύπους και τεχνολογίες ανάπτυξης.

Έτσι, σημαντικές πρωτοβουλίες για την ανάπτυξη λογισμικού με μοντέλα στα πλαίσια του Eclipse, αποτελούν το Eclipse Modeling Project²⁴, και οι σχετιζόμενες τεχνολογίες όπως το Eclipse Modeling Framework²⁵, το Graphical Modelling Framework²⁶, τα EMF Forms²⁷, πρόσθετα μετασχηματισμού μοντέλων (model transformation plugins) όπως το Acceleo²⁸, η γλώσσα μετασχηματισμών Atlas²⁹ (Atlas Transformation Language – ATL) κ.α. Τα εργαλεία για τη μοντελοποίηση με το Eclipse δεν περιορίζονται στο EMF, και υπάρχουν και πολλές άλλες πρωτοβουλίες, όπως το AndromDA³⁰, τα πρόσθετα του Taylor MDA³¹ κ.τ.λ.

Παράλληλα, το Eclipse SOA Platform Project³² έχει σαν σκοπό την παροχή μίας λύσης ανοικτού κώδικα για τον σχεδιασμό, την ανάπτυξη και λειτουργία υπηρεσιοστραφών αρχιτεκτονικών. Στα πλαίσιά του εντάσσονται εργαλεία για γλώσσες περιγραφής διαδικασιών (BPEL Designer³³, BPMN2 Modeler Project³⁴), για την εκτέλεσή τους (eBAM³⁵, eBPM³⁶), για τη μοντελοποίηση διαδικασιών (SCA Tools³⁷, Mangrove³⁸, Winery³⁹) και συνολικές λύσεις όπως το Java Workflow Tooling⁴⁰ και το Stardust⁴¹.

Με τη χρήση των παραπάνω εργαλείων, μπορούν να συντεθούν ευέλικτες ροές εργασιών και περιβάλλοντα που να υποστηρίζουν την ανάπτυξη λογισμικού υπηρεσιών και διαδικασιών σε υπηρεσιοστραφείς αρχιτεκτονικές. Λόγω αυτής της ευελιξίας, το περιβάλλον ανάπτυξης Eclipse και τα σχετικά του πρόσθετα χρησιμοποιούνται κατά κόρον στην έρευνα και τον κλάδο του λογισμικού ως βάση για καινοτόμα εργαλεία, αλυσίδες εργαλείων και ολοκληρωμένα περιβάλλοντα ανάπτυξης.

Το λογισμικό Rational Software Architect⁴² της IBM βασίζεται στο Eclipse και την ανοικτή αρχιτεκτονική του. Είναι ένα ώριμο εμπορικό προϊόν που παρέχει ένα περιβάλλον ανάπτυξης και μία πλήρη εργαλειοθήκη για την ανάπτυξη λογισμικού με μοντέλα. Παράλληλα, υποστηρίζει διαδικασίες ανάπτυξης για υπηρεσιοστραφείς αρχιτεκτονικές, μοντελοποίηση

²⁴ <https://www.eclipse.org/modeling/>

²⁵ <https://www.eclipse.org/modeling/emf/>

²⁶ <http://www.eclipse.org/modeling/gmp/>

²⁷ <http://www.eclipse.org/ecp/emfforms/>

²⁸ <http://www.eclipse.org/acceleo/>

²⁹ <https://eclipse.org/atl/>

³⁰ <http://www.andromda.org/>

³¹ <http://taylor.sourceforge.net/index.php/Overview>

³² <https://www.eclipse.org/soa/>

³³ <https://projects.eclipse.org/projects/soa.bpel>

³⁴ <https://projects.eclipse.org/projects/soa.bpmn2-modeler>

³⁵ <https://projects.eclipse.org/projects/soa.ebam>

³⁶ <https://projects.eclipse.org/projects/soa.ebpm>

³⁷ <https://projects.eclipse.org/projects/soa.sca>

³⁸ <https://projects.eclipse.org/projects/soa.mangrove>

³⁹ <https://projects.eclipse.org/projects/soa.winery>

⁴⁰ <https://projects.eclipse.org/projects/soa.jwt>

⁴¹ <https://projects.eclipse.org/projects/soa.stardust>

⁴² <http://www.ibm.com/developerworks/downloads/r/architect/>

και περιγραφές διαδικασιών. Επιπλέον, το γεγονός ότι στηρίζεται στο Eclipse, επιτρέπει την περαιτέρω επέκταση και παραμετροποίηση του με τη χρήση προσθέτων.

Το πακέτο λογισμικού Enterprise Architect⁴³ της SPARX παρέχει λειτουργίες μοντελοποίησης για τον κύκλο ανάπτυξης λογισμικού σε μία ποικιλία εφαρμογών, από επιχειρησιακές εφαρμογές μέχρι και λογισμικό ενσωματωμένων συστημάτων (embedded systems). Μαζί με τη μοντελοποίηση, υποστηρίζει προσομοίωση της συμπεριφοράς των μοντέλων, καταγραφή και παρακολούθηση απαιτήσεων, παραγωγή πηγαίου κώδικά, σχεδιασμό βάσεων δεδομένων και άλλα. Το EA επιτρέπει τη μοντελοποίηση και ενορχήστρωση επιχειρησιακών διαδικασιών με BPMN και την παραγωγή περιγραφών BPEL και WSDL, υποστηρίζοντας τη δημιουργία υπηρεσιοστραφών συστημάτων (Rosenberg, 2010).

Πολλές ερευνητικές προσπάθειες πάνω στην ανάπτυξη λογισμικού με μοντελοποίηση έχουν οδηγήσει στη μελέτη, σχεδιασμό και δημιουργία εργαλείων για την υποστήριξη μεθοδολογιών και συγκεκριμένων πεδίων εφαρμογής.

Το ερευνητικό έργο A-MUSE περιέγραψε μία πρότυπη υπηρεσιοστραφή αρχιτεκτονική για εφαρμογές επίγνωσης πλαισίου (context awareness) για φορητές εφαρμογές (Daniele, Silva, Pires, & van Sinderen, 2009). Το έργο όρισε επίσης μία μεθοδολογία ανάπτυξης υπηρεσιών με μοντέλα, σύμφωνη με τις προδιαγραφές της MDA. Η διαδικασία ξεκινά στο «επίπεδο περιγραφής υπηρεσιών» (service description level), το οποίο αντιστοιχεί στο επίπεδο CIM της MDA και βασίζεται σε μοντέλα UML για τη δημιουργία διαγραμμάτων κλάσεων και συστήματος, καθώς και μία Γλώσσα Συγκεκριμένου Πεδίου (Domain-Specific Language). Μετά από μετασχηματισμούς, η μοντελοποίηση συνεχίζει στο «επίπεδο σχεδιασμού υπηρεσιών ανεξάρτητα πλατφόρμας» (platform – independent service design level) στο οποίο συνεχίζει η ανάπτυξη των παραπάνω μοντέλων. Στη συνέχεια, ένα ακόμα βήμα μετασχηματισμών οδηγεί στο «επίπεδο σχεδιασμού υπηρεσιών συγκεκριμένης πλατφόρμας» (platform – specific service design level), το οποίο περιλαμβάνει προδιαγραφές BPEL για ενορχήστρωση υπηρεσιών, περιγραφές WSDL διεπαφών υπηρεσιών και κώδικα Java για φορητές συσκευές.

Για τις ανάγκες της παραπάνω μεθοδολογίας, οι Almeida κ.α. εξέτασαν τη δυνατότητα σύνθεσης εργαλείων σε ένα ολοκληρωμένο περιβάλλον ανάπτυξης για την υποστήριξη της μεθοδολογίας A-MUSE και της παροχής των παραγόμενων υπηρεσιών. Συγκεκριμένα, εξέτασαν τα είδη των ολοκληρωμένων λύσεων, σε σχέση με την ολοκλήρωση δεδομένων, ελέγχου, παρουσίασης και διαδικασίας. Οι ερευνητές εξετάζουν πέντε προσεγγίσεις:

Ολοκλήρωση κατά περίπτωση (ad-hoc) ανεξάρτητων εργαλείων λογισμικού με διαλειτουργικότητα 1-προς-1: Χρησιμοποιούνται ανεξάρτητες εφαρμογές μοντελοποίησης, οι οποίες διασυνδέονται μεταξύ τους με προσαρμογείς (adapters) για την ανταλλαγή δεδομένων. Με τον τρόπο αυτό π.χ. δημιουργείται μία αλυσίδα που ακολουθεί τη διαδικασία ανάπτυξης, και η έξοδος μίας εφαρμογής αποτελεί την είσοδο άλλης. Η τροποποίηση της αλυσίδας (π.χ. με νέα εργαλεία ή αλλαγή των σχέσεων μεταξύ τους) επιβάλλει την δημιουργία νέων προσαρμογέων. Με τον τρόπο αυτό, επιτυγχάνεται

⁴³ <http://www.sparxsystems.com/products/ea/>

ολοκλήρωση δεδομένων μεταξύ των εργαλείων, αλλά μόνο σε βασικό επίπεδο. Το περιβάλλον παρέχει πολύ μικρή ευελιξία σε αλλαγές.

Ολοκλήρωση εργαλείων σε κοινό δίαυλο επικοινωνίας: Υπάρχει κοινός μηχανισμός επικοινωνίας για την ανταλλαγή δεδομένων μεταξύ των εργαλείων μοντελοποίησης. Ο μηχανισμός αυτός επιτρέπει και την ύπαρξη κοινών αποθετηρίων μοντέλων και άλλων δεδομένων. Τα εργαλεία εμφανίζονται ξεχωριστά αλλά η προσθήκη ή τροποποίηση των εργαλείων δεν απαιτεί τη αναπροσαρμογή των μηχανισμών διαλειτουργικότητας. Ωστόσο, απαιτείται κοινός μορφότυπος δεδομένων ή η ανάπτυξη μηχανισμών 1-1 μετασχηματισμού των μορφότυπων κατά την επικοινωνία μέσω του διαύλου. Επιτυγχάνεται ολοκλήρωση δεδομένων με τη χρήση κοινού μορφότυπου και ολοκλήρωση ελέγχου μέσω της χρήσης κοινών αποθετηρίων δεδομένων. Οι παραπάνω μηχανισμοί αυξάνουν ελαφρά την ευελιξία, η οποία όμως παραμένει χαμηλή.

Ενιαίος δίαυλος επικοινωνίας και μετασχηματισμοί μετα-μοντέλου: Υπάρχει ενιαίος μηχανισμός διαλειτουργικότητας αλλά δεν περιορίζεται στην μετάδοση δεδομένων. Αντιθέτως, υπάρχουν ενιαίοι μηχανισμοί για την αυτόματη μετατροπή των μορφότυπων μοντελοποίησης από τον ένα τύπο στον άλλο. Στη περίπτωση αυτή επιτυγχάνεται ολοκλήρωση δεδομένων μέσω των μετασχηματισμών, ολοκλήρωση ελέγχου με τη χρήση κοινών αποθετηρίων και ολοκλήρωση διαδικασίας λόγω των μηχανισμών μετασχηματισμού των μοντέλων από ένα εργαλείο στο άλλο, κάτι που οδηγεί σε υψηλή ευελιξία.

Ενιαίος δίαυλος επικοινωνίας και ενιαία διεπαφή χρήστη: Υπάρχει ενιαίος μηχανισμός για την επικοινωνία των εφαρμογών αλλά και μηχανισμοί για την παρουσίαση ενιαίας διεπαφής χρήστη. Τα εργαλεία εντάσσονται στο ίδιο περιβάλλον (π.χ. κοινό GUI) και χρησιμοποιούν κοινές ιεραρχίες και συμβάσεις αλληλεπίδρασης. Επιτυγχάνεται ολοκλήρωση δεδομένων και ελέγχου, αλλά και ολοκλήρωση παρουσίασης, αλλά η ευελιξία είναι σχετικά χαμηλή λόγω της έλλειψης μετασχηματισμών μοντέλων.

Συνδυασμός δίαυλου διαλειτουργικότητας, μετασχηματισμού μετα-μοντέλων και ενιαία διεπαφή χρήστη: Στη περίπτωση αυτή χρησιμοποιούνται όλοι οι παραπάνω μηχανισμοί ολοκλήρωσης, και επιτυγχάνεται ταυτόχρονα ολοκλήρωση δεδομένων, διαδικασίας, ελέγχου και παρουσίασης. Η λύση αυτή παρέχει τη μέγιστη δυνατή ευελιξία, αλλά και τέσσερεις από τους εξεταζόμενους τύπους ολοκλήρωσης για ένα ολοκληρωμένο περιβάλλον ανάπτυξης υπηρεσιών, την οποία θεωρούν καταλληλότερη για τη μεθοδολογία A-MUSE (Almeida, Iacob, Jonkers, Lankhorst, & van Leeuwen, 2007).

Το STRIDE (STructured Requirements and Interface Design Environment) είναι μία μεθοδολογία και πακέτο εργαλείων που αποσκοπεί στην υποστήριξη του σχεδιασμού υπηρεσιών, την αναχρησιμοποίηση των μοντέλων υπηρεσιών και των ενδιάμεσων προϊόντων ανάπτυξης καθώς και τη διαχείριση εκδόσεων (versioning) των παραγόμενων υπηρεσιών. Η μεθοδολογία βασίζεται σε μεταμοντέλο «περιγραφής υπηρεσίας», και τα μοντέλα υπηρεσιών που προκύπτουν από αυτό περιλαμβάνουν πολλαπλά στοιχεία όπως ταυτότητα, απαιτήσεις (λειτουργικές και μη-λειτουργικές), διεπαφές, συμπεριφορά, επιχειρησιακές λειτουργίες, μοντέλων δεδομένων κτλ. Η μεθοδολογία υποστηρίζεται από το πακέτο εργαλείων STRIDE για την ανάπτυξη υπηρεσιοστραφών εφαρμογών στο πεδίο των τηλεπικοινωνιών. Τα μοντέλα της STRIDE χρησιμοποιούνται για την παραγωγή προϊόντων

χαμηλότερου επιπέδου, όπως XML schemas, περιγραφές WSDL και, τελικά, κώδικα Java μαζί με τη σχετική τεκμηρίωση της υπηρεσίας (Bahler, Caruso, & Micallef, 2007).

Το 2007, οι Huang και Fan πρότειναν μία πλατφόρμα Επιχειρησιακής Ολοκλήρωσης (Enterprise Integration Platform), βασισμένη σε υπηρεσιοστραφείς αρχιτεκτονικές και την MDA. Η SMEIP (SOA and MDA based Enterprise Integration Platform) περιγράφεται μόνο σε υψηλό επίπεδο, και περιλαμβάνει λειτουργίες για την ενοποίησης επιχειρησιακών λειτουργιών, υποδομές για την παροχή υπηρεσιών, αποθετήρια δεδομένων κτλ. Οι ερευνητές διατυπώνουν την άποψη ότι η SMEIP οφείλει να διαθέτει ενσωματωμένο περιβάλλον ανάπτυξης λογισμικού βασισμένο στην MDA (Model-Driven Service Development Framework) (Huang & Fan, 2007).

Επίσης το 2007 οι Walderhaug, Stav and Mikalsen ανέφεραν την αλυσίδα εργαλείων με μοντέλα που αναπτύχθηκε στα πλαίσια του έργου MPOWER. Το έργο διερεύνησε τη χρήση μεθόδων ανάπτυξης με μοντέλα για την παραγωγή υπηρεσιών για εφαρμογές οικιακής περίθαλψης. Στη περίπτωση αυτή, οι απαιτήσεις χρήση μοντελοποιούνται με τη χρήση UML στο πακέτο SPARX Enterprise Architect, το οποίο παράγει περιγραφές WSDL. Η μοντελοποίηση των ανταλασσόμενων δεδομένων βασίζεται στο πρότυπο Health Level 7 (HL7) Reference Information Model (HL7, 2007). Οι περιγραφές WSDL εισάγονται στο πακέτο Java platform EE 5 with NetBeans 5.5, με το οποίο παράγονται web services τύπου SOAP, οι οποίες με τη σειρά τους τρέχουν πάνω σε Java Application Server της Sun Microsystems. (Walderhaug, Stav, & Mikalsen, 2007).

Η πλατφόρμα WebRatio 5 είναι ένα εργαλείο CASE, το οποίο εφαρμόζει μία προσέγγιση τύπου MDA για την ανάπτυξη διαδικτυακών εφαρμογών. Το λογισμικό αποτελείται από μία ομάδα πρόσθετων για το περιβάλλον ανάπτυξης Eclipse. Το WebRatio βασίζεται στην WebML⁴⁴ (Web Modeling Language) αλλά υποστηρίζει και την μοντελοποίηση επιχειρησιακών διαδικασιών με τη χρήση BPMN. Επιτρέπει στους προγραμματιστές τον άμεσο μετασχηματισμό των μοντέλων WebML σε κώδικα, με τη χρήση κατάλληλα προσαρμοσμένης γεννήτριας κώδικα (Acerbis, Bongio, Brambilla, & Butti, WebRatio 5: An Eclipse-based CASE tool for engineering Web applications, 2007) (Acerbis, et al., 2008)

Το έργο Sensoria χρηματοδοτήθηκε από την Ευρωπαϊκή Ένωση στα πλαίσια του Framework Programme 6 και διήρκεσε από το 2005 μέχρι το 2010. Είχε σαν αντικείμενο τη μελέτη των διαδικασιών ανάπτυξης πληροφοριακών συστημάτων βασισμένων σε υπηρεσιοστραφείς αρχιτεκτονικές, ενσωματώνοντας θεωρητικές βάσεις, μεθοδολογίες, τεχνικές και εργαλεία σε ένα ενιαίο πλαίσιο. Στα αποτελέσματα του έργου περιλαμβάνεται ολοκληρωμένη μεθοδολογία ανάπτυξης, η SDA (Sensoria Development Approach), η οποία βασίζεται σε μοντελοποίηση και είναι συμβατή με την προσέγγιση MDA και περιλαμβάνει μοντελοποίηση και μετασχηματισμούς μοντέλων στα επίπεδα CIM, PIM και PSM. Μαζί με τη μεθοδολογία, αναπτύχθηκε το αντίστοιχο περιβάλλον ανάπτυξης, το SDE (Sensoria Development Environment). Το επίπεδο PIM υποστηρίζεται από τη γλώσσα μοντελοποίησης UML4SOA⁴⁵ (Mayer, Koch, Schroeder, & Knapp, 2009), η οποία αναπτύχθηκε στα πλαίσια του έργου για

⁴⁴ <http://www.webml.org/webml/page1.do>

⁴⁵ <http://www.uml4soa.eu/>

την περιγραφή υπηρεσιοστραφών υπηρεσιών και βασίζεται στη UML. Στη συνέχεια, το SDE υποστηρίζει τον μετασχηματισμό PIM προς PSM, όπου παρέχεται υποστήριξη για περιγραφές BPEL / WSDL και τις γλώσσες Java και Jolie. Το SDE βασίζεται στο Eclipse, και υλοποιείται μέσω μίας σειράς προσθέτων που υλοποιημένων στα πλαίσια του έργου για μία σειρά από λειτουργικότητες όπως ανάλυση υπηρεσιών, αποτύπωση μη-λειτουργικών προδιαγραφών, μετασχηματισμούς κτλ. Με τη σειρά τους, το ίδιο το περιβάλλον ανάπτυξης και τα εργαλεία αυτά ακολουθούν τις αρχές της υπηρεσιοστραφούς αρχιτεκτονικής, όπως π.χ. ανεξαρτησία από τεχνολογίες, σύνθεση των εργαλείων για την υποστήριξη της ροής ανάπτυξης ως ενορχηστρώσεις υπηρεσιών κτλ. (Wirsing, Hölzl, Koch, Mayer, & Schroeder, 2008)

Η αρχιτεκτονική SOMA (Service Oriented Modelling and Architecture) προτάθηκε από την IBM για την υποστήριξη της ανάπτυξης υπηρεσιοστραφών συστημάτων με μοντέλα. Οι ερευνητές της IBM ανέπτυξαν μία αλυσίδα εργαλείων για την υποστήριξη της διαδικασίας SOMA: οι επιχειρησιακές διαδικασίες περιγράφονται με UML χρησιμοποιώντας το πακέτο Rational Software Architect, αντιστοιχίζονται χειροκίνητα σε μοντέλα XSD⁴⁶ (XML – Schema Definition), που με τη σειρά τους χρησιμοποιούνται για να δημιουργήσουν ένα μοντέλο βασισμένο στο Ecore, δηλαδή το μεταμοντέλο στο οποίο αντιστοιχίζονται τα μοντέλα του Eclipse Modelling Framework (EMF) . Στη συνέχεια, χρησιμοποιείται μία γεννήτρια κώδικα βασισμένη στη τεχνολογία Java Emitter Templates (JET) του EMF για την παραγωγή πηγαίου κώδικα σε Java. (Bercovici, Fournier, & Wecker, 2008)

Η πλατφόρμα ContextServ υποστηρίζει την ανάπτυξη web services με δυνατότητες επίγνωσης πλαισίου (context awareness). Η οι εφαρμογές με επίγνωση πλαισίου λαμβάνουν υπόψη το πλαίσιο μέσα στο οποίο γίνεται η χρήση τους, όπως π.χ. το φυσικό περιβάλλον και η κατάσταση στην οποία χρησιμοποιούνται, και αποκρίνονται με κατάλληλο τρόπο. Η διαδικασία ανάπτυξης στο ContextServ ξεκινά με την περιγραφή της υπηρεσίας χρησιμοποιώντας την ContextUML, μίας εξειδικευμένης γλώσσας βασισμένης στη UML (Sheng & Benatallah, Sheng, Quan Z., and Boualem Benatallah. "ContextUML: a UML-based modeling language for model-driven development of context-aware web services, 2005) σε ένα γραφικό περιβάλλον μοντελοποίησης. Στη συνέχεια, τα μοντέλα εξάγονται με τη μορφή XMI⁴⁷ (XML Metadata Interchange), και μετασχηματίζονται σε εκτελέσιμες περιγραφές υπηρεσιών WSDL, και ενορχηστρώσεις BPEL, οι οποίες δημοσιεύονται και εκτελούνται σε εξυπηρετητές web (Sheng, et al., 2009).

Οι Fabra κ.α. παρουσιάζουν μία προσέγγιση για τη μετάφραση των επιχειρησιακών προδιαγραφών σε εκτελέσιμο κώδικά, με βάση την σύνθεση αποτελεσμάτων από πρότερη τους έρευνα. Συγκεκριμένα, προτείνουν τη σύνθεση της μεθοδολογίας SOD-M (Service-Oriented Development Method) (De Castro, Marcos, & Wieringa, 2009) και την πλατφόρμα εκτέλεσης επιχειρησιακών διαδικασιών DENEb (Development and Execution of iNteroperable dynamic wEB processes) (Fabra, Álvarez, & Bañares, 2011). Ο στόχος είναι η παροχή ενός πλαισίου για την ανάλυση, σχεδιασμό και εκτέλεση επιχειρησιακών διαδικασιών. Η προτεινόμενη αλυσίδα εργαλείων γραφικής μοντελοποίησης βασίζεται στο Eclipse Modelling

⁴⁶ <http://www.w3schools.com/schema/>

⁴⁷ <http://www.omg.org/spec/XMI/>

Framework (EMF). Από την άλλη, η αντιστοίχιση των μοντέλων με εκτελέσιμες ροές εργασιών της πλατφόρμας DENEb υλοποιείται με ειδικά σχεδιασμένο πρόσθετο βασισμένο στην ATL (Atlas Transformation Language) (Fabra, De Castro, Álvarez, & Marcos, 2012).

Οι Aho, Mäki, Pakkala and Ovaska ανέπτυξαν μία αλυσίδα εργαλείων (tool chain) βασισμένη στην MDA για την ανάπτυξη web services. Η αλυσίδα υποστηρίζει δύο ροές εργασίες: την ανάπτυξη των ίδιων των web services και των λειτουργιών πρόσβασης δεδομένων. Η ανάπτυξη των webservices χρησιμοποιεί το SPARX Enterprise Architect για τη δημιουργία μοντέλων UML models των υπηρεσιών, οι οποίες χρησιμοποιούνται για την παραγωγή περιγραφών WSDL και XSD. Στη συνέχεια, οι προγραμματιστές χρησιμοποιούν το πακέτο Apache Axis 2 για να παράγουν απευθείας κώδικα Java από τις περιγραφές αυτές. Στη περίπτωση της πρόσβασης δεδομένων, το σχήμα των βάσεων μοντελοποιείται με UML στο SPARX EA, το οποίο παράγει σενάρια ενεργειών SQL για τη δημιουργία της αντίστοιχης βάσης. Στη συνέχεια, γίνεται χρήση του πακέτου FireStorm / Data Access Objects (DAO) για την παραγωγή του κατάλληλου κώδικα σε Java για την πρόσβαση στη βάση. Τέλος, τα αποτελέσματα και από τις δύο ροές εισάγονται στο Eclipse, όπου ενοποιούνται, τροποποιούνται και δοκιμάζονται. Οι ερευνητές αξιολόγησαν την αποδοτικότητα της διαδικασίας με εφαρμογή σε συγκεκριμένο πεδίο, (Resource Availability Services) και ανέφεραν ότι, στο τελικό προϊόν, πάνω από 95% ήταν αποτέλεσμα αυτόματης παραγωγής κώδικα, ενώ λιγότερο από 5% εισήχθη από τους χρήστες. (Aho, Mäki, Pakkala, & Ovaska, 2009).

Οι Haase και Nagl πρότειναν μία μεθοδολογία ανάπτυξης με μοντέλα και υποστηρικτικό εργαλείο για την ανάπτυξη υπηρεσιοστραφούς ενδιάμεσου λογισμικού διαλειτουργικότητας ετερογενών εφαρμογών. Η μεθοδολογία και τα εργαλεία χρησιμοποιήθηκαν στα πλαίσια ευρύτερου ερευνητικού έργου στο πεδίο της χημικής μηχανικής. Η μεθοδολογία υποστηρίζει τη χρησιμοποίηση «περιτυλιγμάτων» (wrappers) για την παροχή διαλειτουργικότητας μεταξύ διαφορετικών εφαρμογών. Η διαδικασία ανάπτυξης των wrappers περιλαμβάνει τη μοντελοποίηση των διεπαφών, των δομών δεδομένων και τη συμπεριφορά τους, η οποία οδηγεί στη παραγωγή κώδικα Java. Η διαδικασία υλοποιείται μέσω εξειδικευμένου περιβάλλοντος «διαμόρφωσης αρχιτεκτονικής λογισμικού» («software architecture editor»). Το περιβάλλον λογισμικού, αφορούσε αποκλειστικά το πεδίο εφαρμογής του έργου, δηλαδή την διασύνδεση συγκεκριμένων εφαρμογών (Haase & Nagl, 2009) (Haase & Nagl, 2011).

Οι Mayr κ.α. περιέγραψαν μία διαδικασία μοντελοποίησης για την ανάπτυξη υπηρεσιών πρόσβασης δεδομένων (Data Access Services – DAS) σε υπηρεσιοστραφείς αρχιτεκτονικές. Ο όρος DAS αναφέρεται στην ευρύτερη κατηγορία των υπηρεσιών που αφορούν στην εγγραφή, αναζήτηση και τροποποίηση δεδομένων, όπως π.χ. από μία βάση δεδομένων ή ένα αποθετήριο. Η μεθοδολογία έχει σαν στόχο την βελτίωση της δυνατότητας συντήρησης των DAS, την τεκμηρίωση, έλεγχο, την αναχρησιμοποίηση βέλτιστων πρακτικών και την υποστήριξη πολλαπλών οπτικών γωνιών κατά τη διαδικασία ανάπτυξης. Οι ερευνητές προτείνουν μία διαδικασία μοντελοποίησης, η οποία αποτυπώνει πολλαπλές όψεις μίας DAS, όπως τις συνδέσεις με τις βάσεις δεδομένων, τον μορφότυπο δεδομένων κτλ. Με βάση την προσέγγιση αυτή, οι ερευνητές προτείνουν μία γενική αρχιτεκτονική για τις υπηρεσίες πρόσβασης δεδομένων, σύμφωνα με τις γενικές αρχές της SOA, την VMDA (View-based Model-driven Data Access Architecture). Η VMDA μπορεί να ενταχθεί στα πλαίσια ευρύτερων

υπηρεσιοστραφών αρχιτεκτονικών, και περιλαμβάνει αυτόνομη λειτουργικότητα για την καταχώρηση, ανάπτυξη και παροχή υπηρεσιών δεδομένων. Η ανάπτυξη των υπηρεσιών υποστηρίζεται με τη χρήση εργαλείου βασισμένου στο Eclipse, το οποίο παρέχει λειτουργικότητες μοντελοποίησης με το λογισμικό Ecore Model Editor⁴⁸. Η γλώσσα Xpand, η οποία εντάσσεται στη πρωτοβουλία Eclipse M2T (Model to Text)⁴⁹ χρησιμοποιείται για την δημιουργία του τελικού πηγαίου κώδικα των υπηρεσιών (Mayr, Zdun, & Dustdar, 2011).

Το ερευνητικό έργο CHOReOS χρηματοδοτήθηκε στα πλαίσια του Framework Programme 7 και διήρκεσε από τον 2010 μέχρι το 2013. Στα πλαίσια του έργου προτάθηκε μία διαδικασία ανάπτυξης για «χορογραφίες υπηρεσιών» (service choreographies). Οι κλασικές «ενορχηστρώσεις» υπηρεσιών (service orchestrations), περιγράφουν τα διαδοχικά βήματα της αλληλεπίδρασης των συστατικών τους στοιχείων. Αντίθετα, ο όρος «χορογραφίες» περιγράφει τις αλληλεπιδράσεις των υπηρεσιών για την επίτευξη κάποιου τελικού στόχου με τη χρήση γενικών κανόνων για τους ρόλους των υπηρεσιών, αντί δομημένων σεναρίων. Το έργο έχει αναπτύξει ολοκληρωμένο περιβάλλον για την ανάπτυξη και την εκτέλεση των υπηρεσιών (CHOReOS Integrated Development and Runtime Environment⁵⁰), το οποίο περιλαμβάνει και ολοκληρωμένο περιβάλλον ανάπτυξης (CHOReOS Development Environment⁵¹) Η ανάπτυξη των επιχειρησιακών διαδικασιών στο CHOReOS ξεκινά με τη μοντελοποίηση των χορογραφιών σε BPMN2 (BPMN2 Choreography Diagrams). Τα μοντέλα αυτά μετασχηματίζονται με τη χρήση μηχανισμού βασισμένου στην ATL, σε γράφους CLTS (Choreography Labelled Transition System). Οι προγραμματιστές επεξεργάζονται τους γράφους CLTS σε ένα γραφικό περιβάλλον ανάπτυξης βασισμένο στο Graphical Modelling Framework (GMF) του Eclipse. Οι γράφοι μετασχηματίζονται ξανά με τη χρήση μηχανισμού ATL σε ένα σύνολο μοντέλων βασισμένων στο εξειδικευμένο μεταμοντέλο «Coord». Το μεταμοντέλο Coord αναπτύχθηκε στα πλαίσια του έργου για την μοντελοποίηση του συντονισμού υπηρεσιών. Τα μοντέλα αυτά μετασχηματίζονται σε κώδικα Java του λογισμικού που θα εκτελεί τον συντονισμό, με τη χρήση μίας γεννήτριας κώδικα βασισμένης στο Acceleo του Eclipse. (Autili, Di Ruscio, Di Salle, Inverardi, & Tivoli, 2013)

Η προσέγγιση MoDAR (Model-driven Development of Dynamically Adaptive Service-Oriented Systems with Aspects and Rules) είναι μία προσέγγιση για την ανάπτυξη υπηρεσιοστραφών συστημάτων βασισμένων στη BPEL. Η προσέγγιση βασίζεται σε μέρος της διαδικασίας της MDA, παρέχοντας υποστήριξη για τον καθορισμό μοντέλων επιπέδου PIM και τον μετασχηματισμό τους σε μοντέλα PSM. Τα μοντέλα επιπέδου PIM στη MoDAR περιέχουν ταυτόχρονα απεικονίσεις BPMN και επιχειρησιακού κανόνες (business rules), ενώ τα μοντέλα στο επίπεδο PSM στηρίζονται σε προδιαγραφές BPEL και επιχειρησιακού κανόνες στη γλώσσα Drools DRL. Για την υποστήριξη της διαδικασίας, έχει αναπτυχθεί μία εργαλειοθήκη ανάπτυξης, η οποία υποστηρίζει την συσχέτιση υπηρεσιών web με επιχειρησιακές διαδικασίες, καθώς και την σύνταξη και δημοσίευση κώδικα BPEL και επιχειρησιακών κανόνων. Ο κώδικας εκτελείται σε εξειδικευμένο περιβάλλον ανάπτυξης, δημιουργημένο ειδικά για την προσέγγιση MoDAR. Η εργαλειοθήκη περιλαμβάνει γραφικό εργαλείο για την

⁴⁸ <http://www.eclipse.org/ecoretools/>

⁴⁹ <https://eclipse.org/modeling/m2t/>

⁵⁰ <http://www.choreos.eu/bin/view/Documentation/WebHome>

⁵¹ http://www.choreos.eu/bin/view/Documentation/Development_Environment

επισκόπηση του επιχειρησιακού πεδίου εφαρμογής (Business Domain Explorer), το οποίο επιτρέπει στους χρήστες την προβολή εισηγμένων οντολογιών στη γλώσσα OWL⁵². Οι ερευνητές εφάρμοσαν την προσέγγιση σε πραγματική περίπτωση εφαρμογής, στον τομέα της διαχείρισης νοσοκομείων. Επιπλέον, αξιολόγησαν την απόδοσή τόσο ως προς τις επιδόσεις των παραγόμενων υπηρεσιών αλλά και ως προς την παραγωγικότητα των προγραμματιστών που τη χρησιμοποίησαν (Yu, et al., 2015).

3.6.2 Αξιολόγηση

Στις προηγούμενες ενότητες, διατυπώθηκε μία σειρά απαιτήσεων για τις μεθοδολογίες και τα εργαλεία για την ανάπτυξη λογισμικού. Με βάση τις απαιτήσεις αυτές μπορούμε να αξιολογήσουμε τις υφιστάμενες προσεγγίσεις ως προς την καταλληλότητά τους. Τα αποτελέσματα της αξιολόγησης παρουσιάζονται στον Πίνακα 6.

Πλατφόρμα	GSPA	SWS	BWS	IDE	REI	MDSEA-C
Eclipse	●	○	○	●	○	○
Rational	●	○	○	●	●	○
Sparx	●	●	●	●	○	●
A-MUSE	-	●	●	-	-	●
STRIDE	-	-	●	●	●	○
MPOWER	-	●	-	-	●	●
WebRatio 5	-	●	●	●	●	○
SDA	●	-	●	●	-	●
SOMA	●	●	-	●	-	○
ContextServ	-	-	●	●	●	●
Aho, κ.α (2009)	●	●	-	-	-	●
Haase και Nagl (2009) (2011)	-	●	-	-	-	○
SOD-M / DENEb	●	-	●	●	●	●
CHOReOS	●	-	●	●	●	○
MoDAR	●	-	●	●	●	●
● :Υποστηρίζεται, ○ : Απαιτούνται επιπλέον στοιχεία ή προσαρμογές, - : Δεν υποστηρίζεται						
GSPA	Γενικού τύπου εργαλεία ανάπτυξης					
SWS	Υποστήριξη Δομικής Μοντελοποίησης					
BWS	Υποστήριξη Συμπεριφορικής Μοντελοποίησης					
IDE	Ολοκληρωμένο Περιβάλλον Ανάπτυξης					
REI	Ενσωμάτωση στο περιβάλλον εκτέλεσης εφαρμογών					
MDSEA-C	Βασισμένη στην MDA, άρα συμβατή με την MDSEA					

Πίνακας 6: Αξιολόγηση υφιστάμενων προσεγγίσεων ως προς την καταλληλότητα για ανάπτυξη λογισμικού για βιομηχανικές υπηρεσίες

Όλες οι παραπάνω απαιτήσεις συνδυάζουν προσεγγίσεις SOA, BPM, ανάπτυξης λογισμικού με μοντέλα και έχουν αναπτύξει τα απαραίτητα εργαλεία. Όμως, είναι εμφανές ότι καμία από τις προσεγγίσεις που παρουσιάζονται δεν πληρούν όλα τα υπόλοιπα κριτήρια, είτε επειδή είναι ουσιαστικά ασύμβατες με αυτά, είτε απαιτούν επιπλέον προσαρμογές και συμπληρωματικά στοιχεία. Η υπηρεσιοποίηση απαιτεί την ανάπτυξη μεγάλου εύρους εφαρμογών, αλλά πολλές προσεγγίσεις εστιάζουν σε συγκεκριμένους τύπους λογισμικού ή πεδία εφαρμογής. Από την άλλη, αρκετές προσεγγίσεις αντιμετωπίζουν το πρόβλημα της μοντελοποίησης αποκλειστικά από την δομική ή την συμπεριφορική οπτική γωνία, και όχι και από τις δύο. Η χρήση Ολοκληρωμένων Περιβαλλόντων Ανάπτυξης (IDE) προτιμάται σε σχέση με τη χρήση αλυσίδων διακριτών εργαλείων, αλλά πολλές προσεγγίσεις, ενώ παρέχουν εργαλεία, αυτά δεν είναι επαρκώς ολοκληρωμένα. Η δοκιμή πρωτοτύπων και η

⁵² <http://www.w3.org/TR/owl-features/>

γρήγορη θέση σε παραγωγική λειτουργία είναι απαραίτητη για την απόκριση στις ανάγκες των πελατών και τις κινήσεις του ανταγωνισμού. Μία τέτοια απόκριση επιτυγχάνεται με την διασύνδεση του περιβάλλοντος ανάπτυξης με το περιβάλλον εκτέλεσης (το περιβάλλον παραγωγής) του λογισμικού υπηρεσιών, κάτι που δεν παρέχουν πολλές από τις υφιστάμενες προσεγγίσεις. Τέλος, καθώς η MDSEA είναι η μόνη μεθοδολογία ανάπτυξης υπηρεσιών με υποστήριξη για την ανάπτυξη λογισμικού, η βέλτιστη προσέγγιση θα πρέπει να είναι συμβατή μαζί της. Καθώς η MDSEA είναι ουσιαστικά ειδική περίπτωση της MDA, και η ανάπτυξη λογισμικού της MDSEA ακολουθεί τη ροή της MDA, οι υποψήφιες προσεγγίσεις πρέπει να ακολουθούν τη ροή εργασιών και τα επίπεδα της MDA. Ωστόσο, σημαντικός αριθμός δεν μπορεί να χρησιμοποιηθεί για τον σκοπό αυτό, χωρίς εκτεταμένες τροποποιήσεις και προσαρμογές.

Συνεπώς, απαιτείται η εξ'αρχής ανάπτυξη μίας μεθοδολογίας και των απαραίτητων εργαλείων ανάπτυξης, τα οποία θα καλύπτουν όλες αυτές τις απαιτήσεις.

4 Μεθοδολογία και εργαλείο ανάπτυξης λογισμικού υπηρεσιών

4.1 Μεθοδολογία

4.1.1 Γενικά

Η δραστηριότητα της ανάπτυξης λογισμικού αντιστοιχεί στα τελικά βήματα της ροής ΤΠΕ της μεθοδολογίας MDSEA.

Σύμφωνα με την MDSEA, η ανάπτυξη λογισμικού περιλαμβάνει

- τον μετασχηματισμό μοντέλων επιπέδου TIM των στοιχείων ΤΠΕ του συστήματος υπηρεσιών σε μοντέλα TSM
- την παραγωγή εκτελέσιμου κώδικα ή άλλων εκτελέσιμων στοιχείων που αντιστοιχούν στη τεχνολογική πλατφόρμα του συστήματος υπηρεσιών.

Καθώς η MDSEA προέρχεται από την (εστιασμένη στο λογισμικό) MDA, η διαδικασία αυτή αντιστοιχεί στον μετασχηματισμό PIM-PSM της τελευταίας. Αυτό σημαίνει ότι έννοιες, εργαλεία και διαδικασίες της MDA μπορούν να εφαρμοστούν στο πρόβλημα της ανάπτυξης μίας ροής εργασιών σύμφωνης με την MDSEA.

4.1.2 Μετασχηματισμός μοντέλων στην MDA

Για τον μετασχηματισμό PIM προς PSM στην MDA, μπορεί να χρησιμοποιηθεί μία προσέγγιση «αντιστοίχισης» ή «χαρτογράφησης» (mapping approach), δηλαδή η αντιστοίχιση στοιχείων του μοντέλου PIM σε στοιχεία του μοντέλου PSM. Η αντιστοίχιση αυτή μπορεί να διαχωριστεί σε δύο τρόπους συσχέτισης:

- Αντιστοίχιση Τύπων Μοντέλου (Model Type Mapping)
- Αντιστοίχιση Στιγμιότυπων Μοντέλου (Model Instance Mapping)

Η Αντιστοίχιση Τύπων Μοντέλου περιλαμβάνει τον ορισμό συγκεκριμένων τύπων στοιχείων του PIM σε τύπους στοιχείων του PSM. Για παράδειγμα, ένας απλός κανόνας αντιστοίχισης τύπου μοντέλου μπορεί να επιβάλλει ότι οι «κλάσεις» και οι «ιδιότητες κλάσεων» σε ένα διάγραμμα κλάσεων UML αντιστοιχούν σε κλάσεις της Java και σε μεταβλητές των κλάσεων αυτών.

Από την άλλη, η Αντιστοίχιση Στιγμιότυπων Μοντέλου αφορά συγκεκριμένα στοιχεία στο αρχικό μοντέλο PIM. Κατά τη διαδικασία μετασχηματισμού, αυτά επισημαίνονται με με σημασιολογικά στοιχεία («semantic marks»), τα οποία θα υποδεικνύουν πώς αυτά στοιχεία αυτά θα μεταφερθούν στο πεδίο του PSM.

Η Αντιστοίχιση Στιγμιότυπων εφαρμόζεται όταν, λόγω ελλιπούς πληροφορίας στο μοντέλο PIM δεν μπορεί να γίνει αντιστοίχιση μεταξύ τύπων στοιχείων. Η επιπλέον πληροφορία αντλείται από ένα συνδυασμό άλλων πηγών, όπως π.χ. από τις απαιτήσεις συστήματος, τις προδιαγραφές της γλώσσας ανάπτυξης κ.α. Για παράδειγμα, η ιδιότητα (attribute) μίας κλάσης UML μπορεί αντιστοιχιστεί σε μεταβλητή μίας κλάσης Java, ωστόσο μπορεί να μην υπάρχει επαρκής πληροφορία για τον τύπο πρόσβασης (π.χ. αν θα είναι public ή private) ή την αρχική τιμή της. Για τον λόγο αυτό, η συγκεκριμένη επισήμανση πρέπει να καταχωρηθεί

σε κάθε περίπτωση που εμφανίζεται το στοιχείο «ιδιότητα κλάσης UML» μέσω της προσθήκης της κατάλληλης επισήμανσης αυτόματα ή χειροκίνητα. Στη συνέχεια, ανάλογα με τα σημεία κάθε στοιχείου θα εφαρμοστούν οι κατάλληλοι κανόνες αντιστοίχισης.

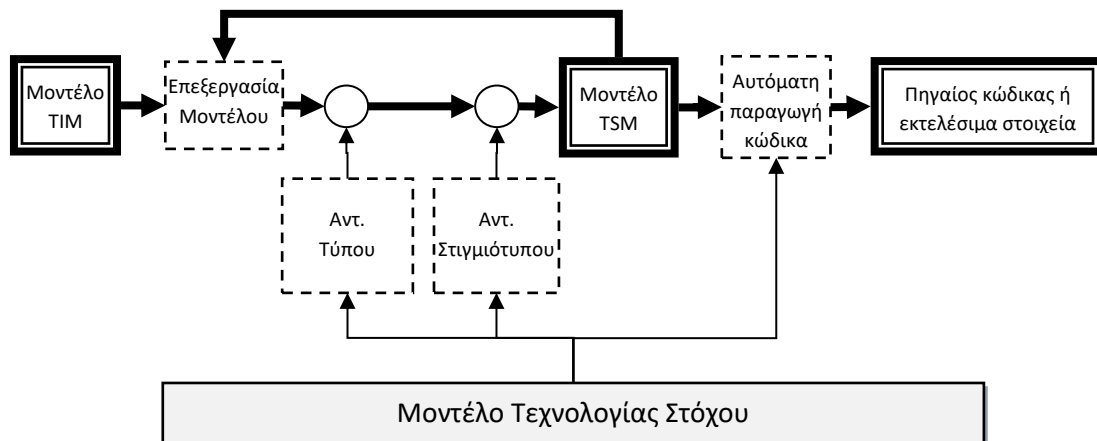
Σε γενικές γραμμές, η Αντιστοίχιση Τύπων μπορεί να εφαρμοστεί σε γλώσσες PIM και PSM όπου υπάρχει επαρκής συσχέτιση ανάμεσα στους τύπους των στοιχείων (π.χ. διαγράμματα κλάσεων UML και Java). Αντιθέτως, η Αντιστοίχιση Στιγμιότυπων καλύπτει «κατά περίπτωση» (ad-hoc) αντιστοιχίσεις, ενώ προσθέτει πληροφορία που δεν υπάρχει ή υπονοείται από το αρχικό μοντέλο.

Τα δύο είδη αντιστοίχισης μπορούν να χρησιμοποιηθούν σε συνδυασμό για να εισάγουν πληροφορίες που σχετίζονται με την τεχνολογική πλατφόρμα που θα υποστηρίξει το σύστημα, και να δημιουργήσουν μοντέλα PSM και, τελικά, κώδικα ή άλλα εκτελέσιμα στοιχεία. Τα προϊόντα της διαδικασίας αυτής μπορεί να εκτείνονται από προσχέδια του τελικού αποτελέσματος (π.χ. code templates, άδειες κλάσεις κτλ.) μέχρι και ολοκληρωμένες υλοποιήσεις). (Truyen, *The Fast Guide to Model Driven Architecture: The Basics of Model Driven Architecture (MDA)*, 2006; Sacevski & Veseli, 2007; Osis & Asnina, 2010).

Με βάση τις έννοιες αυτές, η ανάπτυξη λογισμικού με μοντέλα στην MDSEA περιλαμβάνει, στον πυρήνα της, έναν μετασχηματισμό PIM-προς-PSM, χρησιμοποιώντας ταυτόχρονα αντιστοίχιση Τύπων και Στιγμιότυπων. Ο τελικός στόχος είναι η παραγωγή εκτελέσιμων στοιχείων (κώδικα, λογισμικού, περιγραφές επιχειρησιακών διαδικασιών κτλ.) με έναν συνδυασμό αυτοματοποιημένων εργαλείων και χειροκίνητων μεθόδων.

Σε όρους της MDSEA, δηλαδή αντικαθιστώντας τους όρους PIM και PSM με τους αντίστοιχους TIM και TSM, η διαδικασία μπορεί να περιγραφεί ως εξής:

- Ο προγραμματιστής επεξεργάζεται το μοντέλο TIM
- Η χαρτογραφική αντιστοίχιση (mapping) λαμβάνει χώρα με έναν συνδυασμό αντιστοίχισης τύπων μοντέλου και στιγμιότυπων μοντέλου. Αυτή μπορεί να είναι μία χειροκίνητη, ημι-αυτοματοποιημένη ή πλήρως αυτοματοποιημένη διαδικασία.
- Το αποτέλεσμα της διαδικασίας είναι η παραγωγή ενός μοντέλου TSM βασισμένο στο εισηγμένο μοντέλο TIM.
- Με βάση το μοντέλο TSM παράγονται εκτελέσιμα στοιχεία για την τεχνολογική πλατφόρμα που θα υποστηρίξει το σύστημα (π.χ. κώδικας ή άλλα προϊόντα)
- Ο προγραμματιστής επεξεργάζεται τα εκτελέσιμα στοιχεία, και τα προετοιμάζει για εκτέλεση



Σχήμα 7: Γενική διαδικασία TIM προς εκτελέσιμα αρχεία

Η μεθοδολογία MDSEA παρέχει μοντέλα TIM για το πεδίο των ΤΠΕ του συστήματος υπηρεσίας υπό ανάπτυξη, τα οποία τροφοδοτούν την ανάπτυξη λογισμικού.

Όσο αφορά στις εισόδους της ροής εργασίας, πρέπει να συμπεριληφθούν δομικά και συμπεριφορικά μοντέλα. Έτσι, λαμβάνοντας υπόψη πρότερη έρευνα πάνω στην MDSEA, οι εισοδοί περιλαμβάνουν:

- Διαγράμματα κλάσεων UML, που αντιστοιχούν σε οντότητες του επιχειρησιακού πεδίου εφαρμογής
- Διαγράμματα επιχειρησιακών διαδικασιών του συστήματος υπηρεσιών σε BPMN

Στη πράξη, αυτά συμπληρώνονται από πρόσθετες πληροφορίες σε λιγότερο τυποποιημένες (formal) μορφές, όπως περιγραφές της συμπεριφοράς του λογισμικού, προσχέδια διεπαφών, μη-λειτουργικές προδιαγραφές κ.α.

Από την άλλη, το παραγόμενο λογισμικό θα λειτουργεί στα πλαίσια ενός υπηρεσιοστραφούς συστήματος με στοιχεία διαχείρισης επιχειρησιακών διαδικασιών (BPM). Αυτό σημαίνει ότι η ροή εργασιών θα πρέπει να υποστηρίζει την ανάπτυξη υπηρεσιών και συνθέσεων υπηρεσιών, με εστίαση σε πρωτίστως στην ανάπτυξη web services. Περαιτέρω ευελιξία μπορεί να επιτευχθεί μέσω της παροχής εργαλείων για

- Την ανάπτυξη αυτόνομων εφαρμογών (εκτός της υπηρεσιοστραφούς αρχιτεκτονικής).
- Την ανάπτυξη χωρίς είσοδο μοντέλων, με τη δημιουργία μοντέλων εντός της εφαρμογής ή την απευθείας συγγραφή κώδικα (εκτός της μεθοδολογίας MDSEA).
- Την ανάπτυξη ενδιάμεσων προϊόντων όπως μοντέλα, κώδικας κτλ. (π.χ. για επαναχρησιμοποίηση σε αργότερο χρόνο ή από άλλους χρήστες).

Η γενική ροή εργασιών που παρουσιάστηκε προηγουμένως μπορεί να εφαρμοστεί για όλες τις απαιτούμενες εισόδους και εξόδους, όπως φαίνεται στον αντίστοιχο πίνακα (Πίνακας 7). Έτσι δημιουργείται μία «δομική» και μία «συμπεριφορική» μορφή της ίδιας γενικής ροής εργασιών.

Σχεδιαστικές εισοδοί	Έξοδοι
<u>Δομικές περιγραφές:</u> Διαγράμματα κλάσεων UML	<u>Αντικειμενοστραφής κώδικας εφαρμογών:</u> <ul style="list-style-type: none"> • Αυτόνομες εφαρμογές • Web services • Συνθέσεις υπηρεσιών / web services (με υπηρεσίες αναπτυγμένες τοπικά, ή παρεχόμενες από τρίτους)
<u>Συμπεριφορικές περιγραφές:</u> Μοντέλα Διαδικασιών BPMN 2.0	<u>Εκτελέσιμα σενάρια διαδικασιών:</u> Αυτόνομες διαδικασίες Συνθέσεις εργασιών, (οι οποίες μπορούν να συμπεριλαμβάνουν εργασίες / υπηρεσίες αναπτυγμένες τοπικά, ή παρεχόμενες από τρίτους)

Πίνακας 7: Σχεδιαστικές εισοδοί και αναμενόμενες έξοδοι της ροής εργασιών

Ο συνδυασμός και των δύο προσεγγίσεων μπορεί να υποστηρίξει την ανάπτυξη πλήρων υπηρεσιοστραφών συστημάτων τύπου SOA. Η δομική προσέγγιση μπορεί να χρησιμοποιηθεί για να ορίσεις συγκεκριμένους δρώντες, εμπλεκόμενους ή οντότητες στο πεδίο εφαρμογής, και για την ανάπτυξη αυτόνομων «ατομικών» υπηρεσιών. Μόλις αυτές αναπτυχθούν, οι προγραμματιστές μπορούν να χρησιμοποιήσουν τη συμπεριφορική προσέγγιση για να αποτυπώσουν τις σύνθετες αλληλεπιδράσεις μεταξύ των υπηρεσιών αυτών και να τις αναθέσουν σε μηχανές εκτέλεσης επιχειρησιακών διαδικασιών.

Για να καλυφθούν οι απαιτήσεις αυτές, σχεδιάστηκαν δύο ροές εργασίας:

- Διαγράμματα κλάσεων UML προς κώδικα Java.
- Διαγράμματα ροών εργασίας σε BPMN 2.0 προς εκτελέσιμες επιχειρησιακές διαδικασίες και συνθέσεις υπηρεσιών.

4.1.3 Δομική ροή εργασίας: UML προς Java

Στη περίπτωση αυτή, «τεχνολογικά ανεξάρτητα» διαγράμματα κλάσεων (μοντέλα TIM) χρησιμοποιούνται για την παραγωγή διαγραμμάτων κλάσεων της Java (μοντέλο TSM). Στη συνέχεια, οι κλάσεις του μοντέλου χρησιμοποιούνται για την παραγωγή κώδικα Java (Java Beans).

Η αντικειμενοστραφής γλώσσα Java ταιριάζει φυσικά στην επίσης αντικειμενοστραφή λογική των διαγραμμάτων κλάσεων της UML, κάτι που διευκολύνει σημαντικά τον μετασχηματισμό TIM προς TSM. Καθώς UML μπορεί να επεκταθεί με τη χρήση προφίλ UML (Fuentes-Fernandez & Valecillo-Moreno, 2004) τα οποία μπορούν να χρησιμοποιηθούν για να προσθέσουν επιπλέον σημασιολογικά στοιχεία (π.χ. σχετικά με το πεδίο εφαρμογής και την τεχνολογία-στόχο) σε ένα μοντέλο.

Τα διαγράμματα κλάσεων UML περιγράφουν οντότητες (όπως έγγραφα, συσκευές, ρόλοι κτλ.) και σχέσεις στο πεδίο της επιχειρησιακής εφαρμογής. Από μία τεχνική άποψη, τα μοντέλα TIM περιέχουν πληροφορία σχετικά με κλάσεις UML, components, ιδιότητες, γενικεύσεις, συσχετίσεις και κλάσεις συσχέτισης. Παρά την έλλειψη αλγοριθμικής

πληροφορίας (Neubauer, Mayerhofer, & Gerti, 2014), περιλαμβάνουν αρκετά δεδομένα για να μοντελοποιήσουν τη βασική δομή μίας αντικειμενοστραφούς εφαρμογής.

Αυτό μπορεί να επιτευχθεί με την χαρτογραφική αντιστοίχιση (mapping) στοιχείων UML σε έννοιες της γλώσσας-στόχου. Για παράδειγμα, ένα υποσύνολο μίας τέτοιας χαρτογράφησης για τη γλώσσα Java μπορεί να είναι το ακόλουθο.

Στοιχείο UML του μοντέλου TIM	Αντιστοίχιση Java
Κλάση	Κλάση Java (μέσω επισήμανσης με το στερεότυπο «Javabean» της UML)
Ιδιότητα κλάσης (απροσδιόριστου τύπου δεδομένων)	Μεταβλητή ενός συγκεκριμένου τύπου δεδομένων της Java. Ο τύπος δεδομένων δεν παρέχεται άμεσα από το μοντέλο TIM και καθορίζεται από τον προγραμματιστή μέσω της κατάλληλης «σήμανσης».
Λειτουργία	Μέθοδος
Σχέση γενίκευσης	Γενίκευση
Σύνδεση/ Συνάθροιση	Σύνδεση/ Συνάθροιση
Κλάση συσχέτισης	Κλάση Java (μέσω επισήμανσης με το στερεότυπο «Javabean» της UML)

Πίνακας 8: Αντίστοιχιση των στοιχείων του Μοντέλου TIM σε στοιχεία της Java

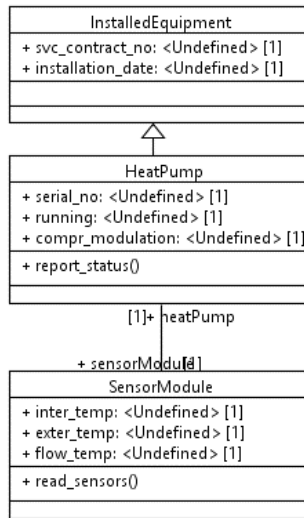
Σημειώνεται ότι στη περίπτωση αυτή, οι κλάσεις UML, οι ιδιότητες, λειτουργίες και συσχετίσεις μεταφράζονται με τη χρήση αντιστοίχισης τύπου στα κατάλληλα στοιχεία της Java (κλάσεις, μέθοδοι, μεταβλητές και συσχετίσεις). Ωστόσο, στην περίπτωση των μεταβλητών, οι τύποι δεδομένων δεν είναι καθορισμένοι, οπότε κάθε «στιγμιότυπο» μίας μεταβλητής πρέπει να επισημανθεί με τον κατάλληλο τύπο δεδομένων της Java. Η ενέργεια αυτή αποτελεί αντιστοίχιση στιγμιότυπου μοντέλου.

Η παραπάνω μεθοδολογία επιτρέπει την εισαγωγή επισημάνσεων στο μοντέλο UML μέσω ενός εξειδικευμένου προφίλ UML (UML profile). Ένα προφίλ παρέχει επιπλέον στοιχεία σημειογραφίας με σκοπό τη χρήση του μοντέλου σε συγκεκριμένα πεδία εφαρμογής. Στη συγκεκριμένη περίπτωση, μπορεί να γίνει εφαρμογή ενός κατάλληλου προφίλ το οποίο υποστηρίζει τη δημιουργία μοντέλων UML για την αναπαράσταση λογισμικού σε Java, δηλαδή μοντέλα TSM. Για παράδειγμα, πλέον, οι κλάσεις του (γενικής χρήσης) διαγράμματος κλάσεων της UML στο TIM, αντιστοιχούν σε κλάσεις Java στο TSM. Σε αυτή την περίπτωση, οι κλάσεις μπορούν να περιλαμβάνουν νέα στοιχεία που αφορούν σε μεθόδους και χαρακτηριστικά (μεταβλητές) μία δεδομένης κλάσης.

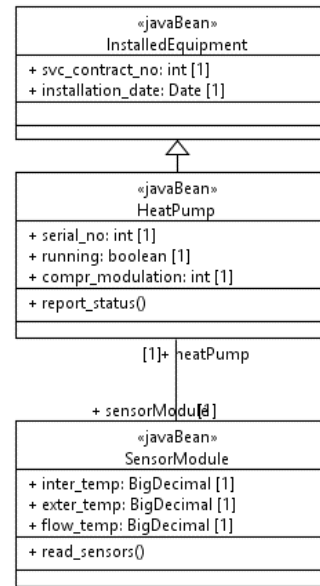
Ο προγραμματιστής χρησιμοποιεί τα σημασιολογικά στοιχεία του προφίλ για να προσθέσει επισημάνσεις στο μοντέλο TIM.

Το αποτέλεσμα είναι ένα μοντέλο κλάσεων UML, εμπλουτισμένο με κατάλληλα σημασιολογικά στοιχεία, και το οποίο αναπαριστά τη βασική δομή της υπό ανάπτυξης εφαρμογής σε γλώσσα Java, δηλαδή ένα μοντέλο «συγκεκριμένης τεχνολογίας» (Technology-Specific Model).

Μοντέλο Υπηρεσίας Επιπέδου TIM
(Technology - Independent Model)



Μοντέλο επιπέδου TSM
(Technology - Specific Model) για Java



Αντιστοίχιση
και εμπλουτισμός
μοντέλου

Σχήμα 8: Παράδειγμα μετασχηματισμού ενός μοντέλου TIM σε TSM για τη γλώσσα Java

Στη πράξη, αναμένεται ότι οι προγραμματιστές θα επεξεργάζονται το μοντέλο και πέρα από τη συγκεκριμένη διαδικασία αντιστοίχισης. Η ανάπτυξη λειτουργικού και αποτελεσματικού λογισμικού μπορεί να απαιτεί την προσαρμογή του μοντέλου σε συγκεκριμένη τεχνολογία υλοποίησης (για παράδειγμα με την προσθήκη κλάσεων, μεταβλητών και μεθόδων που δεν περιλαμβάνονται στην αρχική, ανεξάρτητη τεχνολογίας, αναπαράσταση), την προσαρμογή και αναχρησιμοποίηση υφιστάμενων στοιχείων κ.α.

Το μοντέλο TSM περιέχει πλέον αρκετή πληροφορία για την αυτόματη παραγωγή πλαισίων κώδικα Java για όλες τις κλάσεις που περιέχει, μαζί με κατάλληλες δηλώσεις μεταβλητών και μεθόδων, και λαμβάνοντας υπόψη τις γενικεύσεις και συσχετίσεις τους.


```

/* HeatPump.java */
import java.util.*;
import java.math.*;
public class HeatPump extends InstalledEquipment {

private int compr_modulation;
private boolean running;
private boolean operation;
private int serial_no;

public HeatPump() { super();

}

public void report_status() {
// Start of user code for the body of report_status
// TODO should be implemented
// End of user code
}

public int getCompr_modulation() {
return this.compr_modulation;
}

public void setCompr_modulation(int newCompr_modulation) {
this.compr_modulation = newCompr_modulation;
}

... (etc)

```

Σχήμα 9: Παράδειγμα πλαισίου κώδικα javaBean που έχει παραχθεί από μοντέλο TSM

Στη συνέχεια, οι προγραμματιστές υλοποιούν τις εφαρμογές συμπληρώνοντας τα πλαίσια κώδικα, προσθέτοντας νέες κλάσεις, μεταβλητές και μεθόδους και ακολουθώντας, εν γένει, κοινές πρακτικές ανάπτυξης λογισμικού.

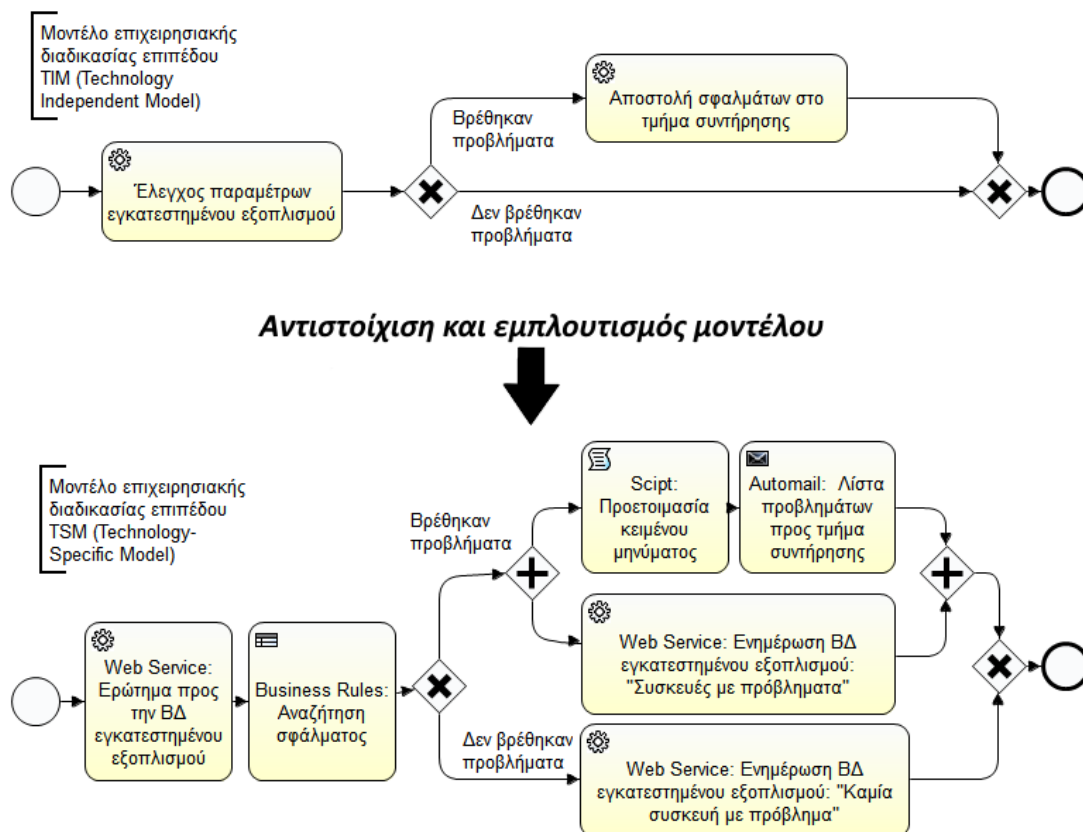
4.1.4 Συμπεριφορική ροή εργασίας: Μοντέλα BPMN 2.0 προς εκτελέσιμες επιχειρησιακές διαδικασίες και συνθέσεις υπηρεσιών

Η συμπεριφορική ροή εργασίας βασίζεται στη χρήση μοντέλων BPMN 2.0 για τη δημιουργία εκτελέσιμων διαδικασιών. Οι εκτελέσιμες διαδικασίες αφορούν συστήματα διαχείρισης επιχειρησιακών διαδικασιών (Georgakopoulos, Hornick, & Sheth, 1995), τα περισσότερα από τα οποία είναι συμβατά με τη σημειογραφία BPMN 2.0. Η χρήση της ίδιας γλώσσας ως εισόδου και εξόδου επιτρέπει την ανάπτυξη εκτελέσιμων επιχειρησιακών διαδικασιών χωρίς την ανάγκη να μεταφραστούν σε άλλες γλώσσες (όπως BPEL), διατηρώντας μία απλή και συνεπή ροή εργασιών.

Τα μοντέλα TIM BPMN 2.0 περιέχουν αναφορές σε δραστηριότητες (εργασίες, υπό-εργασίες κ.α.), συμβάντα (έναρξη, ενδιάμεσα, τελικά), πύλες και διακλαδώσεις (αποκλειστικές XOR, παράλληλες, εξαρτώμενες από γεγονότα κ.α.), συνδέσεις (π.χ. μηνύματα, συνδέσεις με άλλες εργασίες, συσχετίσεις κ.α.) και άλλα σημασιολογικά στοιχεία (π.χ. δεδομένα, έγγραφα, «διάδρομοι» και «δεξαμενές»).

Τα μοντέλα BPMN 2.0 του επιπέδου TIM περιέχουν μόνο πληροφορία για την επιχειρησιακή διαδικασία και δεν είναι προσαρμοσμένα στις απαιτήσεις και ιδιαιτερότητες οποιασδήποτε μηχανής εκτέλεσης (συνεπώς αντιστοιχούν σε Μοντέλα «Ανεξάρτητα Πλατφόρμας» - PIM της MDA). Κατά τη διάρκεια της ροής εργασίας, οι προγραμματιστές θα εργάζονται με το μοντέλο TIM ώστε να δημιουργήσουν ένα μοντέλο TSM, το οποίο θα μπορεί να εκτελεστεί στα πλαίσια της υποδομής ΤΠΕ του συστήματος υπηρεσιών. Οι εργασίες που απαιτούνται στο στάδιο αυτό θα εξαρτώνται από παράγοντες όπως οι επιχειρησιακές απαιτήσεις, η τεχνολογία στόχος, και η ωριμότητα του μοντέλου TIM (π.χ. ως προς την ανάλυση των απαιτούμενων εργασιών). Οι απαιτούμενες εργασίες μπορεί να περιλαμβάνουν έναν συνδυασμό των παρακάτω δραστηριοτήτων:

- Τροποποιήσεις του μοντέλου: υλοποίηση και ανάλυση λειτουργικότητας που περιγράφεται με αφαιρετικό τρόπο στο μοντέλο εισόδου, βελτιώσεις της ροής διαδικασιών, της ευχρηστίας και της συνέπειας, η αναχρησιμοποίηση υφιστάμενων δομών και ροών, καθώς και η εισαγωγή επιπλέον στοιχείων BPMN (π.χ. εργασίες, μηνύματα, γεγονότα και πύλες).
- Προσθήκη στοιχείων ελέγχου εκτέλεσης: αυτά μπορεί να περιλαμβάνουν μεταβλητές που μεταδίδουν δεδομένα κατά την εκτέλεση της διαδικασίας, σενάρια ελέγχου, επιχειρησιακοί κανόνες και στοιχεία που αφορούν στη τελική μηχανή εκτέλεσης επιχειρησιακών διαδικασιών, όπως εργασίες ηλεκτρονικού ταχυδρομείου και αναθέσεις εργασιών σε συγκεκριμένους υπολογιστικούς πόρους.
- Προσθήκη αναφορών σε υπηρεσίες λογισμικού και εξωτερικά στοιχεία: Κατά την εκτέλεση διαδικασιών από τις περισσότερες μηχανές εκτέλεσης, είναι εφικτή η επίκληση υπηρεσιών λογισμικού από συστήματα με υπηρεσιοστραφή αρχιτεκτονική. Στη περίπτωση αυτή, οι μηχανικοί λογισμικού προσθέτουν «εργασίες web service» (web service tasks), οι οποίες καλούν, δίνουν ορίσματα και δέχονται αποτελέσματα από διαθέσιμες υπηρεσίες λογισμικού. Η συγκεκριμένη λειτουργικότητα επιτρέπει τη δημιουργία συνθέσεων υπηρεσιών τύπου SOA μέσα από τη συμπεριφορική ροή εργασιών.
- Προσθήκη πολύπλοκης λειτουργικότητας μέσω κώδικα εφαρμογών: Πολλές σύνθετες συμπεριφορές δεν μπορούν να αναπαρασταθούν αποτελεσματικά με BPMN 2.0. Για τον λόγο αυτό, πολλές μηχανές εκτέλεσης επιχειρησιακών διαδικασιών επιτρέπουν στους προγραμματιστές την περιγραφή σύνθετων συμπεριφορών μέσω της προσθήκης κώδικα σε διάφορες γλώσσες προγραμματισμού.
- Προσθήκη στοιχείων αλληλεπίδρασης χρήστη: Οι προγραμματιστές μπορούν να ορίσουν και να σχεδιάσουν στοιχεία αλληλεπίδρασης χρήστη, όπως φόρμες και μενού συσχετισμένα με συγκεκριμένες εργασίες της επιχειρησιακής διαδικασίας. Τα στοιχεία αυτά εμφανίζονται στους τελικούς χρήστες μέσα από τη μηχανή εκτέλεσης διαδικασιών, και υποστηρίζουν την αλληλεπίδραση με τον τελικό χρήστη κατά την εκτέλεση της διαδικασίας.



Σχήμα 10: Παράδειγμα μετασχηματισμού μοντέλου BPMN 2.0 από TIM σε TSM μέσω αντιστοίχισης και εμπλουτισμού

Το αποτέλεσμα της διαδικασίας είναι ένα νέο μοντέλο BPMN 2.0 (όπως φαίνεται στο Σχήμα 10) της επιχειρησιακής διαδικασίας, το οποίο όμως περιέχει πληροφορία που επιτρέπει την εκτέλεσή του από την τελική μηχανή εκτέλεσης επιχειρησιακών διαδικασιών του συστήματος υπηρεσιών.

Τα μοντέλα TIM και TSM είναι εκφρασμένα σε BPMN 2.0 με το μοντέλο TSM να περιέχει επεκτάσεις και προσθήκες που αφορούν σε συγκεκριμένη τεχνολογική πλατφόρμα. Η Αντιστοίχιση Τύπων περιορίζεται στην τετριμμένη συσχέτιση στοιχείων της BPMN (π.χ. εργασίες και γεγονότα). Από την άλλη, η Αντιστοίχιση Στιγμιότυπων περιλαμβάνει την προσθήκη διάφορων σημασιολογικών στοιχείων που αφορούν στην πλατφόρμα εκτέλεσης της διαδικασίας (όπως συγκεκριμένες εργασίες, ροές και κώδικά), που αντιστοιχεί σε ένα μετασχηματισμό PIM-PSM της MDA.

Στο παράδειγμα στο Σχήμα 10, η επιχειρησιακή διαδικασία αφορά τον εντοπισμό σφαλμάτων σε εγκατεστημένο εξοπλισμό, κατά την παροχή υπηρεσιών απομακρυσμένης παρακολούθησης και συντήρησης. Το μοντέλο TIM περιγράφει τη διαδικασία με αφαιρετικούς όρους: «έλεγχος παραμέτρων» και, εφόσον βρεθούν σφάλματα, αποστολή λίστας σφαλμάτων στους τεχνικούς. Ο μηχανικός λογισμικού θα πρέπει να εμπλουτίσει και να ενημερώσει το μοντέλο σύμφωνα με την τεχνολογική υποδομή που θα παρέχεται από το σύστημα υπηρεσιών, δηλαδή την πλατφόρμα εκτέλεσης διαδικασιών, τις διαθέσιμες

υπηρεσίες λογισμικού κ.α. Ο μηχανικός λογισμικού αναπτύσσει τη διαδικασία και προσθέτει επιπλέον λεπτομέρεια: η αναζήτηση σφαλμάτων απαιτεί την κλήση του απαραίτητου web service για την ανάγνωση των λειτουργικών παραμέτρων των υφιστάμενων αντλιών, και την εφαρμογή ενός συνόλου από επιχειρησιακούς κανόνες ώστε να προσδιοριστεί αν υπάρχουν δυσλειτουργίες. Εάν βρεθούν δυσλειτουργίες, πρέπει να ενημερωθεί το τμήμα συντήρησης. Έτσι, πρέπει να προστεθεί κώδικας για την δημιουργία μίας λίστας βλαβών, και στη συνέχεια να κληθεί το υποσύστημα ηλεκτρονικού ταχυδρομείου ώστε να τη στείλει στο τμήμα συντήρησης. Τέλος, πρέπει να ενημερωθεί η βάση δεδομένων της υπηρεσίας με το αποτέλεσμα του ελέγχου, δηλαδή την ύπαρξη ή όχι σφαλμάτων, μέσω της κλήσης ήδη υφιστάμενου web service.

```
<?xml version="1.0" encoding="UTF-8"?>
<bpmn:definitions xmlns:bpmn="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:di="http://www.omg.org/spec/DD/20100524/DI"
xmlns:dc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" id="Definitions_1">
  <bpmn:process id="Process_1" isExecutable="true">
    <bpmn:startEvent id="StartEvent_1">
      <bpmn:outgoing>SequenceFlow_06t7scv</bpmn:outgoing>
    </bpmn:startEvent>
    <bpmn:serviceTask id="Task_1bzlr4m" name="Equipment_DB_Inquiry">
      <bpmn:incoming>SequenceFlow_06t7scv</bpmn:incoming>
      <bpmn:outgoing>SequenceFlow_183rfn4</bpmn:outgoing>
    </bpmn:serviceTask>
    <bpmn:sequenceFlow id="SequenceFlow_06t7scv" sourceRef="StartEvent_1"
targetRef="Task_1bzlr4m" />
    <bpmn:sequenceFlow id="SequenceFlow_183rfn4" sourceRef="Task_1bzlr4m"
targetRef="Task_0b0eia0" />
    <bpmn:businessRuleTask id="Task_0b0eia0" name="Check_for_faults">
      <bpmn:incoming>SequenceFlow_183rfn4</bpmn:incoming>
      <bpmn:outgoing>SequenceFlow_1xmlfz9</bpmn:outgoing>
    </bpmn:businessRuleTask>
    <bpmn:exclusiveGateway id="ExclusiveGateway_lehdavf">
      <bpmn:incoming>SequenceFlow_1xmlfz9</bpmn:incoming>
      <bpmn:outgoing>SequenceFlow_1c2f321</bpmn:outgoing>
      <bpmn:outgoing>SequenceFlow_1cbnjpp</bpmn:outgoing>
    </bpmn:exclusiveGateway>
    <bpmn:sequenceFlow id="SequenceFlow_1xmlfz9" sourceRef="Task_0b0eia0"
targetRef="ExclusiveGateway_lehdavf" />
    <bpmn:sequenceFlow id="SequenceFlow_1c2f321" name="Faults_found"
sourceRef="ExclusiveGateway_lehdavf" targetRef="ExclusiveGateway_14yyh3q" />
    <bpmn:parallelGateway id="ExclusiveGateway_14yyh3q">
      <bpmn:incoming>SequenceFlow_1c2f321</bpmn:incoming>
      <bpmn:outgoing>SequenceFlow_1nbvbs3</bpmn:outgoing>
      <bpmn:outgoing>SequenceFlow_0q3a8eq</bpmn:outgoing>
    </bpmn:parallelGateway>
    <bpmn:sequenceFlow id="SequenceFlow_1cbnjpp" name="No_faults"
sourceRef="ExclusiveGateway_lehdavf" targetRef="Task_0el4r5w" />
    <bpmn:serviceTask id="Task_0el4r5w" name="Equipment_DB_Update_No_Faults">
      <bpmn:incoming>SequenceFlow_1cbnjpp</bpmn:incoming>
      <bpmn:outgoing>SequenceFlow_076aiuc</bpmn:outgoing>
    </bpmn:serviceTask>
    <bpmn:endEvent id="EndEvent_10nc8wn">
      <bpmn:incoming>SequenceFlow_0ff7szc</bpmn:incoming>
    </bpmn:endEvent>
  </bpmn:process>
  . . . (etc)
```

Σχήμα 11: Ενδεικτικός κώδικας BPMN 2.0 σε μορφή XML

Το ολοκληρωμένο διάγραμμα BPMN αντιστοιχεί σε κώδικα XML ο οποίος θα μπορεί να αναγνωστεί και να εκτελεστεί από κατάλληλη μηχανή εκτέλεσης επιχειρησιακών διαδικασιών, όπως φαίνεται στο Σχήμα 11.

4.2 Αρχιτεκτονική της Πλατφόρμας Ανάπτυξης Λογισμικού Υπηρεσιών

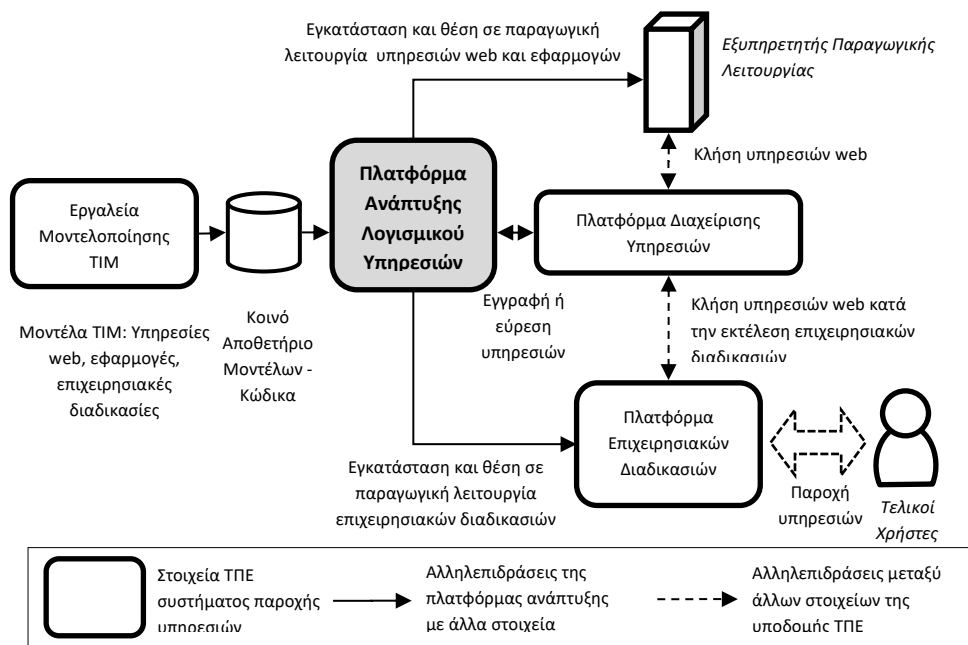
4.2.1 Στόχοι και πλαίσιο συστήματος

Η Πλατφόρμα Ανάπτυξης Λογισμικού Υπηρεσιών (από εδώ και στο εξής «Πλατφόρμα Ανάπτυξης») είναι ένα περιβάλλον ανάπτυξης βασισμένης σε μοντέλα, η οποία υποστηρίζει τη ροή εργασιών που περιγράφεται στις προηγούμενες ενότητες. Η είσοδος της πλατφόρμας ανάπτυξης είναι μοντέλα επιπέδου TIM (διαγράμματα κλάσεων UML και μοντέλα διαδικασιών BPMN 2.0) από πρότερα στάδια της μεθοδολογίας MDSEA. Οι έξοδοι της της πλατφόρμας είναι εφαρμογές, υπηρεσίες web, συνθέσεις υπηρεσιών, εκτελέσιμες επιχειρησιακές διαδικασίες και ενδιάμεσα προϊόντα όπως μοντέλα και αποσπάσματα κώδικα.

Όπως φαίνεται στο Σχήμα 12, η πλατφόρμα ανάπτυξης υφίσταται στα πλαίσια της υποδομής ΤΠΕ του συστήματος υπηρεσιών. Συγκεκριμένα:

- Έναρξη: Οι είσοδοι TIM προέρχονται από σχεδιαστές και αναλυτές που χρησιμοποιούν εργαλεία μοντελοποίησης UML και BPMN στα προηγούμενα στάδια της μεθοδολογίας MDSEA.
- Σύμφωνα με τις συνθήκες της ανάπτυξης λογισμικού για ΣΠΥ, αναμένεται ότι η ανάπτυξη των μοντέλων και λογισμικού θα λαμβάνει χώρα σε ένα συνεργατικό περιβάλλον με πολλούς χρήστες. Στη περίπτωση αυτή είναι απαραίτητη η χρήση ενός κοινού αποθετηρίου (repository) για μοντέλα, κώδικα, τεκμηρίωση και εκτελέσιμα αρχεία. Έτσι, οι σχεδιαστές των υπηρεσιών θα μεταφέρουν τα μοντέλα σε ένα κοινό «Αποθετήριο Μοντέλων», από το οποίο θα παραλαμβάνονται από τους προγραμματιστές.
- Ο μετασχηματισμός TIM-TSM-«εκτελέσιμα αρχεία», λαμβάνει χώρα στην πλατφόρμα ανάπτυξης, μέσω της μεθοδολογίας που περιγράφεται στις προηγούμενες ενότητες.
- Οι εφαρμογές και οι υπηρεσίες web εγκαθίστανται και μπαίνουν σε παραγωγική λειτουργία στους αντίστοιχους εξυπηρετητές παραγωγικής λειτουργίας (Deployment Servers).
- Όταν ένα μία υπηρεσία web εγκαθίσταται και ενεργοποιείται, εγγράφεται στη Πλατφόρμα Διαχείρισης Υπηρεσιών των υποδομών ΤΠΕ του συστήματος υπηρεσιών. Η εγγραφή περιλαμβάνει μεταδεδομένα για την υπηρεσία web, τη διεύθυνση κλήσης, και άλλες πληροφορίες που απαιτούνται για την χρήση της.
- Επιπλέον, η πλατφόρμα ανάπτυξης αναχρησιμοποιεί υπηρεσίες για την ανάπτυξη συνθέσεων υπηρεσιών. Αυτό απαιτεί την εύρεση και ανάκληση πληροφορίας όπως οι περιγραφές των υπηρεσιών (π.χ. σε μία κατάλληλη γλώσσα όπως η WSDL). Η πληροφορία αυτή ανασύρεται από την Πλατφόρμα Διαχείρισης Υπηρεσιών του συστήματος ΤΠΕ.

- Οι εκτελέσιμες επιχειρησιακές διαδικασίες και οι συνθέσεις υπηρεσιών εγκαθίστανται σε μία κατάλληλη Πλατφόρμα Εκτέλεσης Επιχειρησιακών Διαδικασιών.
- Η Πλατφόρμα Επιχειρησιακών Διαδικασιών εκτελεί τις διαδικασίες και παρέχει τις απαιτούμενες υπηρεσίες στους τελικούς χρήστες του συστήματος (παραγωγούς, καταναλωτές και συνεργάτες).
- Κατά την εκτέλεση των επιχειρησιακών διαδικασιών, η Πλατφόρμα Επιχειρησιακών Διαδικασιών καλεί υπηρεσίες web που έχουν εγγραφεί στην υποδομή Διαχείρισης Υπηρεσιών Web. Η υποδομή παρακολουθεί τη λειτουργία των υπηρεσιών και λειτουργεί ως ενδιάμεσος μεταξύ των εξυπηρετητών παραγωγικής λειτουργίας και την Μηχανή Εκτέλεσης Διαδικασιών.



Σχήμα 12: Το πλαίσιο και οι αλληλεπιδράσεις της Πλατφόρμας Ανάπτυξης Λογισμικού Υπηρεσιών

4.2.2 Προδιαγραφές Πλατφόρμας Ανάπτυξης Λογισμικού Υπηρεσιών.

4.2.2.1 Εισαγωγή

Ο σχεδιασμός του περιβάλλοντος ανάπτυξης απαιτεί τον καθορισμό των απαιτήσεων και βασικών προδιαγραφών του.

4.2.2.2 Σενάριο Χρήσης

Ένα από τα βασικά εργαλεία για την ανάλυση απαιτήσεων ενός λογισμικού είναι η δημιουργία των σεναρίων χρήσης της εφαρμογής (ή αλλιώς «περιπτώσεων χρήσης» - use cases). Το σενάριο χρήσης περιγράφει τη χρήση και τη λειτουργία του λογισμικού με αφηγηματικό τρόπο και σε βήματα. Η αποτύπωση αυτή επιτρέπει τον προσδιορισμό των

χρηστών, των λειτουργικών μερών, των δεδομένων και των διαδικασιών του τελικού λογισμικού.

Το βασικό σενάριο χρήσης της Πλατφόρμας Ανάπτυξης είναι η διαδικασία της μετατροπής ενός μοντέλου επιπέδου TIM σε μοντέλο επιπέδου TSM και εν συνεχεία η παραγωγή του σχετικού κώδικα ή άλλων εκτελέσιμων προϊόντων. Το σενάριο μπορεί να αναλυθεί σε δύο κλάδους με τα ίδια βήματα ανάλογα με το αν εκτελείται η δομική ή η συμπεριφορική ροή εργασιών. Ανάλογα όμως με το ποια ροή εργασίας εκτελείται, τα βήματα μπορεί να εκτελούνται με διαφορετικό τρόπο.

Η βασική περίπτωση χρήσης αναλύεται ως ακολούθως:

Εμπλεκόμενοι
Οι κύριοι εμπλεκόμενοι (χρήστες) είναι τεχνικά καταρτισμένοι αναλυτές υπηρεσιών (π.χ. αναλυτές λογισμικού) και μηχανικοί λογισμικού.
Είσοδοι
<u>Ένα μοντέλο επιπέδου TIM (Technology Independent Model):</u> Το μοντέλο αυτό έχει παραχθεί από αναλυτές και σχεδιαστές υπηρεσιών με τη χρήση εργαλείων μοντελοποίησης. Τα αρχεία των μοντέλων ακολουθούν κοινά αποδεκτά πρότυπα βασισμένα σε XML για τις αντίστοιχες σημειογραφίες μοντελοποίησης (UML και BPMN 2.0). Αναμένεται ότι στη κανονική ροή εργασιών, το μοντέλο TIM βρίσκεται αποθηκευμένο σε κοινό αποθετήριο μοντέλων και εισάγεται στη πλατφόρμα ανάπτυξης με κατάλληλη διεπαφή. Κατ'εξαιρεση, μπορεί να γίνει εισαγωγή μοντέλων από άλλες πηγές, π.χ. με απευθείας φόρτωση στη πλατφόρμα ανάπτυξης.
<u>Ενδιάμεσα προϊόντα και πόροι ανάπτυξης λογισμικού:</u> Κατά την ανάπτυξη λογισμικού παράγονται ή χρησιμοποιούνται και άλλα ενδιάμεσα προϊόντα και πόροι, όπως άλλα μοντέλα TIM και TSM, τμήματα πηγαίου κώδικα, εκτελέσιμα αρχεία κ.α. Αυτά μπορεί να περιλαμβάνονται στο αποθετήριο μοντέλων ή να εισάγονται απευθείας στη πλατφόρμα ανάπτυξης.
<u>Μεταδεδομένα για υπηρεσίες εγγεγραμμένες στη Πλατφόρμα Παροχής Υπηρεσιών:</u> Αφορούν υπηρεσίες λογισμικού που υπάρχουν ήδη εγγεγραμμένες στη Πλατφόρμα Παροχής Υπηρεσιών της υποδομής ΤΠΕ του συστήματος παροχής υπηρεσιών. Οι είσοδοι αυτές χρησιμοποιούνται κατά τη διαδικασία εμπλουτισμού των μοντέλων και ανάπτυξης του κώδικα με σκοπό την εισαγωγή κλήσεων προς τις αντίστοιχες υπηρεσίες.
Αποτελέσματα
<u>Νέα ή τροποποιημένη εκτελέσιμη υπηρεσία (web service):</u> Το αποτέλεσμα της δομικής ροής εργασίας. Η υπηρεσία εγκαθίσταται σε εξυπηρετητή παραγωγικής λειτουργίας και τα μεταδεδομένα της εγγράφονται στη Πλατφόρμα Παροχής Υπηρεσιών του συστήματος ΤΠΕ.

Νέα ή τροποποιημένη εκτελέσιμη επιχειρησιακή διαδικασία: Το αποτέλεσμα της συμπεριφορικής ροής εργασίας. Περιλαμβάνει την ενορχήστρωση υπηρεσιών που έχουν παραχθεί με τη δομική ροή εργασίας, αλλά και τρίτες υπηρεσίες και λογισμικό. Η επιχειρησιακή διαδικασία εγκαθίσταται στην υποδομή εκτέλεσης επιχειρησιακών διαδικασιών της υποδομής ΤΠΕ.

Άλλα τελικά ή ενδιάμεσα προϊόντα: Η πλατφόρμα ανάπτυξης υποστηρίζει την ανάπτυξη γενικού σκοπού και επιτρέπει τη δημιουργία επιπλέον προϊόντων όπως αυτόνομες εφαρμογές, μοντέλα, κώδικα, κ.α.

Περιβάλλον

Πλατφόρμα Ανάπτυξης: Περιβάλλον ανάπτυξης λογισμικού με γραφική διεπαφή, η οποία επιτρέπει λειτουργίες τύπου κατάδειξης και επιλογής (“point-and-click”). Περιλαμβάνει μηχανισμούς διαχείρισης του χώρου εργασίας του χρήστη (“workspace management”) και την ανάπτυξη λογισμικού με βάση τις ροές εργασίας που περιγράφηκαν στη προηγούμενη ενότητα.

Αποθετήριο Μοντέλων: ένα αποθετήριο που περιέχει διάφορους πόρους για χρήση κατά τη διαδικασία ανάπτυξης λογισμικού, όπως είσοδοι (μοντέλα TIM) ενδιάμεσα προϊόντα κ.α. Στο αποθετήριο έχουν κοινή πρόσβαση οι αναλυτές / σχεδιαστές των υπηρεσιών και οι προγραμματιστές.

Εξυπηρετητής παραγωγικής λειτουργίας : εξυπηρετητής ή εξυπηρετητές (εικονικοί ή φυσικοί) στους οποίους εγκαθίστανται οι υπηρεσίες web και οι εφαρμογές που αναπτύσσονται μέσω της πλατφόρμας ανάπτυξης.

Πλατφόρμα Παροχής Υπηρεσιών: πλατφόρμα που διαχειρίζεται την παροχή των υπηρεσιών web. Έχει τον ρόλο του service-oriented middleware (Issarny, et al., 2011) και λειτουργεί σαν ενδιάμεσος μεταξύ των εξυπηρετητών παραγωγικής λειτουργίας και της υποδομής εκτέλεσης επιχειρησιακών διαδικασιών. Σε αυτή εγγράφονται τα μεταδεδομένα των διαθέσιμων υπηρεσιών web (URL κλήσης, παράμετροι, τεκμηρίωση κ.α.) κατά την εγκατάσταση των υπηρεσιών web στους εξυπηρετητές. Τα δεδομένα αυτά παρέχονται στους προγραμματιστές κατά την ανάπτυξη λογισμικού και επιχειρησιακών διαδικασιών ώστε να ενσωματώσουν κλήσεις προς τις αντίστοιχες υπηρεσίες web.

Πλατφόρμα Εκτέλεσης Επιχειρησιακών Διαδικασιών: περιλαμβάνει και εκτελεί τις επιχειρησιακές διαδικασίες που αναπτύσσονται στη πλατφόρμα ανάπτυξης, με σκοπό την παροχή υπηρεσιών στους τελικούς χρήστες. Καλεί υπηρεσίες web μέσω της Πλατφόρμας Παροχής Υπηρεσιών. Περιλαμβάνει λειτουργίες διαχείρισης και παρακολούθησης της εκτέλεσης των επιχειρησιακών διαδικασιών.

Κύρια Ροή

Η κύρια ροή περιλαμβάνει τα βασικά βήματα της κεντρικής περίπτωσης χρήσης της Πλατφόρμας Ανάπτυξης:

1. Ο χρήστης ξεκινά τη διαδικασία εισαγωγής TIM μοντέλου από το Αποθετήριο της Πλατφόρμας Ανάπτυξης.

2. Ο χρήστης επιλέγει ένα μοντέλο επιπέδου TIM για εισαγωγή από το Αποθετήριο Μοντέλων στη Πλατφόρμα Ανάπτυξης. Το μοντέλο μπορεί να είναι «δομικό» (Διάγραμμα κλάσεων UML) ή «συμπεριφορικό» (επιχειρησιακή διαδικασία BPMN 2.0).
3. Ο χρήστης επιλέγει μία Τεχνολογία Πλατφόρμας (ένα Μοντέλο Τεχνολογίας), το οποίο θα καθορίσει τις συγκεκριμένες παραμέτρους της διαδικασίας μετασχηματισμού TIM προς TSM.
4. Σε οποιοδήποτε σημείο, ο χρήστης μπορεί να αναζητήσει και να προσθέσει διάφορα εξωτερικά στοιχεία:
 - a. Πόρους αποθηκευμένους στο Αποθετήριο Μοντέλων
 - b. Αναφορές σε υπηρεσίες εγγεγραμμένες στη Πλατφόρμα Παροχής Υπηρεσιών
 - c. Εξωτερικά στοιχεία όπως π.χ. μοντέλα αποθηκευμένα στο σύστημα αρχείων του σταθμού εργασίας του χρήστη κ.τ.λ.
5. Ο χρήστης επεξεργάζεται το μοντέλο TIM («επεξεργασία μοντέλου υπηρεσίας». Αυτό επιτυγχάνεται χρησιμοποιώντας τη δυνατότητες που παρέχει η Πλατφόρμα Ανάπτυξης, όπως η γραφική διεπαφή, διαδικασίες «κατάδειξης και επιλογής» (point-and-click) και καταχώρηση/επεξεργασία δομημένου/ ελεύθερου κειμένου.
 - a. Ο χρήστης μπορεί να κάνει πλήρη επεξεργασία του μοντέλου TIM, εισάγοντας και αφαιρώντας στοιχεία, αλλάζοντας τις διασυνδέσεις και τις ιδιότητες των στοιχείων, προσθέτοντας αναφορές σε εξωτερικά στοιχεία
 - b. Ο χρήστης μπορεί να εισάγει αποσπάσματα μοντέλων στη Πλατφόρμα Ανάπτυξης και να τα επαναχρησιμοποιήσει. Η προέλευσή τους μπορεί να είναι το Αποθετήριο Μοντέλων ή εξωτερικές πηγές (σύστημα αρχείων χρήστη κτλ.)
 - c. Ο χρήστης δεν χρειάζεται να χτίσει μία ολόκληρη υπηρεσία από αρχή έως τέλους (end-to-end) σε μία συνεδρία. Μπορούν να επιλέξει να επεξεργαστεί σε αυτό το βήμα (και να προωθήσει στα επόμενα) μία συλλογή στοιχείων ή ένα τμήμα μίας βιομηχανικής υπηρεσίας. Το αποτέλεσμα μπορεί να είναι τμήματα μοντέλων TSM για επαναχρησιμοποίηση σε αργότερο χρόνο ή από άλλους χρήστες.
6. Στη συνέχεια ο χρήστης εμπλέκεται στη διαδικασία «εμπλουτισμού μοντέλου». Σε ένα περιβάλλον «κατάδειξης και επιλογής», ο χρήστης «εμπλουτίζει» το (ανεξάρτητο τεχνολογίας) μοντέλο επιπέδου TIM με σημάνσεις (“annotations”) προερχόμενες από το επιλεγμένο Τεχνολογικό Μοντέλο (το Μοντέλο Πλατφόρμας). Με τον τρόπο αυτό παράγει το μοντέλο επιπέδου TSM (Technology-Specific Model).
 - a. Ανάλογα με την αρχική γλώσσα μοντελοποίησης (του επιπέδου TIM) και την τεχνολογία-στόχο, κάποια μέρη της διαδικασίας αυτής μπορούν να αυτοματοποιηθούν. Αυτό μπορεί να επιτευχθεί αν, για παράδειγμα, η γλώσσα προέλευσης και η γλώσσα στόχος χρησιμοποιούν έννοιες που μπορούν να αντιστοιχιστούν μεταξύ τους με σχέσεις «1 προς 1» (δηλαδή «αντιστοίχιση τύπων μοντέλου»).

- b. Ανάλογα με την επιλεγμένη Πλατφόρμα Τεχνολογίας, η Πλατφόρμα Ανάπτυξης μπορεί να προσφέρει λειτουργίες και στοιχεία διεπαφής, οι οποίες να αφορούν τη συγκεκριμένη τεχνολογία - στόχο.
7. Με την ολοκλήρωση του μοντέλου TSM, ο χρήστης αιτείται την παραγωγή του αντίστοιχου πηγαίου κώδικα (υπηρεσιών web ή εκτελέσιμων επιχειρησιακών διαδικασιών).
8. Στη συνέχεια, η πλατφόρμα ανάπτυξης χρησιμοποιεί το μοντέλο TSM για να δημιουργήσει πηγαίο κώδικα ή άλλα προϊόντα για τη συγκεκριμένη τεχνολογία
- a. Αναμένεται ότι κάποια από τα παραγόμενα στοιχεία θα απαιτούν περαιτέρω συμπλήρωση. Για παράδειγμα, ένα η τεχνολογία – στόχος είναι αντικειμενοστραφής (όπως στη δομική ροή εργασίας), ο πηγαίος κώδικας μπορεί να περιέχει άδειες κλάσεις, οι οποίες θα περιέχουν π.χ. ορισμούς ιδιοτήτων και άδειες μεθόδους αλλά όχι τον κώδικα για την εκτέλεσή τους. Αυτό μπορεί να συμβαίνει καθώς ενδέχεται τα στοιχεία του TSM να αντιστοιχούν σε περισσότερο «αφηρημένες» έννοιες της τεχνολογίας – στόχου (π.χ. το TSM να καθορίζει κλάσεις με ιδιότητες και μεθόδους, αλλά να μην περιγράφει το πώς θα υλοποιηθούν οι μέθοδοι αυτοί σε επίπεδο κώδικα). Το τελικό αποτέλεσμα μπορεί να είναι ένα πρότυπο ή πλαίσιο το οποίο θα πρέπει να συμπληρωθεί από τον προγραμματιστή. (Αυτό μπορεί να αποφευχθεί εάν π.χ. έχει ήδη γίνει στο παρελθόν η αντιστοίχιση των στοιχείων αυτών και είναι εφικτή η αναχρησιμοποίηση του κώδικα – στη περίπτωση αυτή θα γίνεται εισαγωγή από το Αποθετήριο Μοντέλων)
9. Ο χρήστης επεξεργάζεται, δοκιμάζει και αφαιρεί σφάλματα από τον πηγαίο κώδικα ή το πλαίσιο πηγαίου κώδικα.
- a. Από τεχνικής άποψης, το βήμα αυτό δεν διαφέρει από τις διαδικασίες ανάπτυξης λογισμικού όπως αυτές εκτελούνται σε ολοκληρωμένα περιβάλλοντα λογισμικού γενικά (IDEs) και δεν θα αναλυθούν περαιτέρω στο στάδιο αυτό. Οι δραστηριότητες αυτές περιλαμβάνουν καταχώρηση και επεξεργασία κώδικα, τη δημιουργία εκτελέσιμων αρχείων, δοκιμαστική λειτουργίας, εντοπισμό και αφαίρεση σφαλμάτων κ.α.
10. Ο χρήστης μεταγλωττίζει τον πηγαίο κώδικα σε εκτελέσιμο λογισμικό / επιχειρησιακή διαδικασία, τον εγκαθιστά και τον θέτει σε λειτουργία στην αντίστοιχη πλατφόρμα. Ο τρόπος με τον οποίο γίνεται αυτό διαφέρει ανάλογα με τη ροή εργασιών που λαμβάνει χώρα.
- a. Στη δομική ροή, ο χρήστης εγκαθιστά την υπηρεσία web στους εξυπηρετητές παραγωγικής λειτουργίας. Στη συνέχεια αιτείται την εγγραφή της νέας υπηρεσίας στη Πλατφόρμα Διαχείρισης Υπηρεσιών της υποδομής ΤΠΕ του συστήματος υπηρεσιών. Η υπηρεσία εγγράφεται στη Πλατφόρμα Παροχής Υπηρεσιών και γίνεται διαθέσιμη προς χρήση.
- b. Στη συμπεριφορική δομή, ο χρήστης εγκαθιστά την επιχειρησιακή διαδικασία στη Πλατφόρμα Εκτέλεσης Επιχειρησιακών Διαδικασιών. Η πλατφόρμα εκτελεί την επιχειρησιακή διαδικασία και παρέχει υπηρεσίες στους τελικούς χρήστες.

Επεκτάσεις

Οι επεκτάσεις είναι πιθανές αποκλίσεις από την κύρια ροή.

Στα βήματα 1-2

- Η διαδικασία ανάπτυξης λογισμικού μπορεί να ξεκινήσει χωρίς την εισαγωγή μοντέλου TIM. Στο βήμα 5, ο χρήστης δημιουργεί ένα νέο μοντέλο υπηρεσίας από την αρχή. Η διαδικασία ανάπτυξης συνεχίζεται κανονικά. Τα βήματα 1, 2 παραλείπονται.

Στα βήματα 1-5

- Η διαδικασία ανάπτυξης λογισμικού ξεκινά με την εισαγωγή συγκεκριμένων εξωτερικών στοιχείων (π.χ. μοντέλα, πηγαίος κώδικας κτλ.) , τα οποία θα τροποποιηθούν κατάλληλα για χρήση στο MSEE. Τα βήματα 6,7,8 μπορούν να παραλειφθούν αν τα εξωτερικά στοιχεία είναι πηγαίος κώδικας.

Στο βήμα 9

- Το αποτέλεσμα της αυτόματης παραγωγής κώδικα μπορεί να μην απαιτεί περαιτέρω επεξεργασία. Όλες οι απαραίτητες εισοδοί έχουν παρασχεθεί σε προηγούμενα βήματα. Το βήμα 9 παραλείπεται.

Στα βήματα 11-12

- Το παραγόμενο αποτέλεσμα είναι αυτόνομη εφαρμογή ή ενδιάμεσο προϊόν (μοντέλο, κώδικας, κ.α.) που θα αποθηκευτεί στο αποθετήριο μοντέλων για μελλοντική χρήση. Τα βήματα 11,12 παραλείπονται.

Σε οποιοδήποτε βήμα.

- Ο χρήστης μπορεί να φυλάξει όλα τα στοιχεία του έργου του (project) σε προσωπικό χώρο εργασίας ή σε κοινό χώρο εργασίας (στη περίπτωση συνεργατικής ανάπτυξης λογισμικού). Το συνολικό project μπορεί να περιλαμβάνει
 - Το μοντέλο ή μοντέλα TIM
 - Επιλογές σχετικά με την τεχνολογία – στόχο (π.χ. Πλατφόρμα Τεχνολογίας)
 - Αναφορές σε εξωτερικά στοιχεία
 - Παραγόμενο μοντέλο ή μοντέλα TSM
 - Παραγόμενος πηγαίος κώδικας
 - Κατάλληλα μεταδεδομένα (όπως εγγραφές στη Πλατφόρμα Παροχής Υπηρεσιών, εμπλεκόμενοι χρήστες κτλ.)
- Ο χρήστης μπορεί να επιλέξει να αποθηκεύσει διάφορα ενδιάμεσα προϊόντα στο Αποθετήριο μοντέλων. Η αποθήκευση μπορεί να κάνει διαθέσιμα τα προϊόντα αυτά σε άλλους χρήστες, αν και η πρόσβαση μπορεί να υπόκειται σε συγκεκριμένους κανόνες και προνόμια πρόσβασης. Αυτά τα προϊόντα μπορούν να περιλαμβάνουν:
 - Διαμορφωμένα μοντέλα TIM

- Μοντέλα TSM
- Πηγαίο κώδικα
- Άλλα στοιχεία και προϊόντα εφόσον απαιτείται (π.χ. έγγραφα, τεκμηρίωση κτλ.)

4.2.2.3 Απαιτήσεις

Η βασική χρήση της Πλατφόρμας Ανάπτυξης, όπως αναλύθηκε στις προηγούμενες ενότητες, οδηγεί σε ένα σύνολο από απαιτήσεις, λειτουργικές και μη-λειτουργικές.

Η περιγραφή των απαιτήσεων της Πλατφόρμας Ανάπτυξης στο παρόν κείμενο ακολουθεί τις κατευθυντήριες γραμμές του προτύπου IETF RFC 2119⁵³ “Key words for use in RFCs to indicate Requirement Levels”. Στο κείμενο, χρησιμοποιούνται οι λέξεις κλειδιά «Συνίσταται» (“Recommended” στο πρωτότυπο) και «Προαιρετικό» (“Optional” στο πρωτότυπο). Η λέξη-κλειδί «Απαιτείται» (“Required” στο πρωτότυπο) υπονοείται για κάθε απαίτηση, εκτός αν γίνεται χρήση μίας από τις άλλες δύο λέξεις-κλειδιά.

4.2.2.3.1.1 Λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις ορίζουν συγκεκριμένες συμπεριφορές κατά τη λειτουργία του συστήματος.

4.2.2.3.1.1.1 Αλληλεπίδραση χρήστη

- Η εφαρμογή θα παρέχει γραφική διεπαφή χρήστη
- Όλες οι ενέργειες χρήστες που αφορούν στη λειτουργία της εφαρμογής θα είναι προσβάσιμες από αυτή τη μοναδική και ενοποιημένη διεπαφή
- Η διεπαφή χρήστη θα παρέχει ενοποιημένη εποπτεία της διαδικασίας ανάπτυξης, και θα παρέχει πρόσβαση σε όλα τα στοιχεία και ενδιάμεσα προϊόντα ενός έργου ανάπτυξης λογισμικού (software development project)
- Η σύνθεση και επεξεργασία μοντέλων υπηρεσιών θα λαμβάνει χώρα σε περιβάλλον «κατάδειξης και επιλογής», εντός του ενοποιημένου γραφικού περιβάλλοντος της Πλατφόρμας Ανάπτυξης

4.2.2.3.1.1.2 Βασικές λειτουργίες

4.2.2.3.1.1.2.1 Κύρια ροή

- Η εφαρμογή θα υποστηρίζει την εισαγωγή μοντέλων TIM
 - Από το Αποθετήριο Μοντέλων
 - *Συνίσταται*: Σαν ξεχωριστά αρχεία/ομάδες αρχείων από τα συστήματα αρχείων στα οποία έχει πρόσβαση ο σταθμός εργασίας του χρήστη
 - Η εφαρμογή θα παρέχει υποστήριξη (κατ’ ελάχιστο) για τα πρότυπα UML (Universal Modelling Language), BPMN (Business Process Modelling Notation) και WSDL (Web Service Definition Language – υποστήριξη περιγραφών για web services τύπου SOAP)

⁵³IETF RFC 2193, Key words for use in RFCs to indicate Requirement Levels, διαθέσιμο στη διεύθυνση <http://www.ietf.org/rfc/rfc2119.txt>

- *Συνίσταται:* Υποστήριξη για REST-ful web services
- Η εφαρμογή θα υποστηρίζει την επιλογή ενός Μοντέλου Τεχνολογίας και την ρύθμιση κατάλληλων παραμέτρων για την πλατφόρμα – στόχο:
 - *Συνίσταται (Πρωτότυπη εφαρμογή):* Οι τεχνολογίες στόχοι στη πρωτότυπη εφαρμογή θα είναι
 - JAVA : πηγαίος κώδικας εφαρμογών
 - JAX – WS : ανάπτυξη web services (Java)
 - Εκτελέσιμα αρχεία συμβατά με BPMN 2.0 , προς εισαγωγή και εκτέλεση στην Πλατφόρμα Εκτέλεσης Επιχειρησιακών Διαδικασιών. Η μορφή των αρχείων και η τελική τεχνολογία θα εξαρτηθεί από τις τεχνολογίες υλοποίησης της υποδομής ΤΠΕ του συστήματος υπηρεσιών.
- *Προαιρετικά (Πρωτότυπη εφαρμογή):* REST-ful web services, άλλες τεχνολογίες εφόσον κριθούν απαραίτητες
- Η εφαρμογή θα υποστηρίζει αναζήτηση για επιπλέον πόρους, με σκοπό την συμπερίληψή τους στο μοντέλο TIM κατά την επεξεργασία του:
 - Από το Αποθετήριο Μοντέλων (μοντέλα επιπέδου TIM, τμήματα μοντέλων κτλ.)
 - Από την Πλατφόρμα Παροχής Υπηρεσιών (εγγεγραμμένες υπηρεσίες)
 - *Συνίσταται:* Σαν ξεχωριστά αρχεία/ομάδες αρχείων από τα συστήματα αρχείων στα οποία έχει πρόσβαση ο σταθμός εργασίας του χρήστη
- Η εφαρμογή θα εισάγει αυτά τα στοιχεία στο έργο ανάπτυξης λογισμικού και θα επιτρέπει την ενσωμάτωση στο μοντέλο
- Η εφαρμογή θα παρέχει στον χρήστη δυνατότητες επεξεργασίας της υπηρεσίας και των συστατικών της στοιχείων στο επίπεδο TIM
- Η εφαρμογή θα παρέχει στον χρήστη δυνατότητες σήμανσης του μοντέλου TIM χρησιμοποιώντας στοιχεία προερχόμενα από το επιλεγμένο Μοντέλο Τεχνολογίας, παράγοντας έτσι ένα μοντέλο TSM.
- Η εφαρμογή θα παράγει τον πηγαίο κώδικα των υπηρεσιών web και επιχειρησιακών διαδικασιών από το TSM με βάση:
 - Τη δομή του μοντέλου TSM.
 - Τις σημάνσεις του χρήστη με βάση την τεχνολογία στόχο.
 - Την επιλεγμένη τεχνολογία - στόχο (με βάση το Μοντέλο Τεχνολογίας) και άλλες σχετικές τεχνολογικές παραμέτρους που έχει θέσει ο χρήστης.
 - Άλλες επιλογές του χρήστη και ρυθμίσεις.
- Η εφαρμογή θα παρέχει στον χρήστη τα μέσα για την επεξεργασία, δοκιμή, εντοπισμό σφαλμάτων των τελικών προϊόντων
- Η εφαρμογή θα επιτρέπει την εγγραφή μίας πλήρους υπηρεσίας στη Πλατφόρμα Παροχής Υπηρεσιών.
- *Συνίσταται:* Η εφαρμογή θα παρέχει αυτοματισμού για την υποστήριξη της διαδικασίας ανάπτυξης λογισμικού και υπηρεσιών. Αυτές μπορούν να περιλαμβάνουν:
 - Αυτόματες σημάνσεις στο επίπεδο TIM, βασισμένες στο επιλεγμένο Μοντέλο Τεχνολογίας
 - Λειτουργίες αυτόματης συμπλήρωσης κειμένου (autocomplete)

- Εύκολη πρόσβαση σε αναχρησιμοποιούμενους ή κοινούς πόρους
- Προαιρετικό: Λοιπές δυνατότητες κατά περίπτωση

4.2.2.3.1.1.2.2 Επεκτάσεις

- Η εφαρμογή θα επιτρέπει την αποθήκευση της εργασίας του χρήστη σε διάφορα στάδια ολοκλήρωσης, κατά τη διάρκεια της κύριας ροής, με τη μορφή «έργων» (projects)
- Η εφαρμογή θα επιτρέπει την αποθήκευση των προϊόντων της διαδικασίας ανάπτυξης στο Αποθετήριο Μοντέλων
- Η εφαρμογή θα επιτρέπει την επανάληψη των βημάτων ή και του συνόλου της διαδικασίας ανάπτυξης π.χ. με τον χρήστη να επεξεργάζεται εκ νέου ένα μοντέλο βιομηχανικής υπηρεσίας στο επίπεδο TIM με στόχο την βελτίωση του παραγόμενου κώδικα.
- Η εφαρμογή θα επιτρέπει την διαδικασία ανάπτυξης λογισμικού χωρίς αρχικό μοντέλο τύπο TIM, δηλαδή την ανάπτυξη υπηρεσιών, εφαρμογών και εντοπισμένων υπηρεσιών εκ του μηδενός.

4.2.2.3.1.1.2.3 Ολοκλήρωση με την υποδομή πληροφορικής του συστήματος υπηρεσιών και διαλειτουργικότητα με τα στοιχεία της

Η εφαρμογή θα πρέπει να επικοινωνεί με τα ακόλουθα εξωτερικά στοιχεία:

- Εργαλεία μοντελοποίησης TIM, έμμεσα, διαμέσου του Αποθετηρίου Μοντέλων.
- Πλατφόρμα Παροχής Υπηρεσιών διαμέσου του API (Application Programming Interface) που εκθέτει.
 - Η εφαρμογή θα πρέπει να μπορεί να εγγράψει υπηρεσίες στη Πλατφόρμα Παροχής Υπηρεσιών
 - Η εφαρμογή θα πρέπει να μπορεί να αναζητήσει υπηρεσίες εγγεγραμμένες στη Πλατφόρμα Παροχής Υπηρεσιών, με σκοπό την εισαγωγή των σχετικών αναφορών κατά τη διαδικασία ανάπτυξης λογισμικού
- Πλατφόρμα Εκτέλεσης Επιχειρησιακών Διαδικασιών διαμέσου του API (Application Programming Interface) που εκθέτει. Η εφαρμογή πρέπει να αποστέλλει επιχειρησιακές διαδικασίες στη Πλατφόρμα Εκτέλεσης Επιχειρησιακών διαδικασιών.

4.2.2.3.1.1.2.4 Συνεργασία και χώροι εργασίας

Η Πλατφόρμα Ανάπτυξης θα παρέχει εργαλεία για τη συνεργατική ανάπτυξη λογισμικού. Η συνεργασία μεταξύ των χρηστών θα βασίζεται στη χρήση κοινών χώρων εργασίας στο Αποθετήριο Μοντέλων.

Το Αποθετήριο Μοντέλων θα περιλαμβάνει τα ακόλουθα στοιχεία:

- Ένα κοινό αποθετήριο πόρων και ενδιάμεσων προϊόντων, το οποίο θα υποστηρίζει τη διαδικασία ανάπτυξης υπηρεσιών και λογισμικού, από το επιχειρησιακό επίπεδο (π.χ. ανάλυσης υπηρεσίας) στο τεχνικό (π.χ. επίπεδο κώδικα)
- Δυνατότητες κλειδώματος, αποκλειστικής χρήσης και ελέγχου ακεραιότητας δεδομένων με σκοπό τη διευκόλυνση της εργασίας με κοινούς πόρους

- Δυνατότητες ελέγχου πρόσβασης και δικαιωμάτων χρήστη, ανάλογα με τον ρόλο/ομάδα / οργανισμό
- Δυνατότητες αναζήτησης στους κοινούς πόρους. Το εύρος της περιοχής αναζήτησης θα βασίζεται στα προνόμια του χρήστη.
- *Συνίσταται:* Δυνατότητες ελέγχου εκδόσεων
- *Συνίσταται:* Παρακολούθηση και καταγραφή δραστηριότητας και ενεργειών χρηστών
- *Προαιρετικό:* Άλλες δυνατότητες εφόσον κριθούν αναγκαίες.

4.2.2.3.1.2 *Μη Λειτουργικές απαιτήσεις*

4.2.2.3.1.2.1 *Ευχρηστία*

Τα Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης (Integrated Development Environments – IDE) ανήκουν σε έναν σχετικά ώριμο πλέον τομέα λογισμικού, με αποτέλεσμα την ανάλογη ωριμότητα των στοιχείων διεπαφής και ευχρηστίας που παρέχουν. Τα σύγχρονα IDE προσφέρουν πλέον πολλά χρήσιμα εργαλεία αλληλεπίδρασης χρήστη, όπως tooltips, κατάδειξη σύνταξης (syntax highlighting), διαχείριση του χώρου εργασίας του χρήστη, κτλ. Όλα τα παραπάνω επιτρέπουν συντομότερους χρόνους εκμάθησης και υψηλότερη παραγωγικότητα στην ανάπτυξη λογισμικού.

4.2.2.3.1.2.2 *Ενσωμάτωση πολλαπλών τεχνολογιών και προτύπων*

Από τις διαδικασίες και τις λειτουργικές απαιτήσεις που περιγράφηκαν προηγουμένως, γίνεται φανερό ότι η Πλατφόρμα ανάπτυξης θα πρέπει να ενσωματώνει ενδιάμεσα προϊόντα και διαδικασίες που βασίζονται σε διαφορετικά πρότυπα και τεχνολογίες, π.χ. σύνθεση υπηρεσιών με γραφικό τρόπο, πηγαίο κώδικα σε διάφορες γλώσσες προγραμματισμού, γλώσσες μοντελοποίησης, κτλ. Επιπλέον, οι τελικές εφαρμογές βιομηχανικών υπηρεσιών εμπλέκουν μία μεγάλη ποικιλία από πλατφόρμες και πλαίσια λογισμικού (π.χ. σταθμούς εργασίας, εξυπηρετητές, κινητές εφαρμογές, μικροελεγκτές, αισθητήρες κτλ.). Το περιβάλλον της Πλατφόρμας ανάπτυξης θα πρέπει να μπορεί να υποστηρίξει την ολοκληρωμένη ανάπτυξη περίπλοκων εφαρμογών βασισμένων σε ετερογενείς τεχνολογίες και πρότυπα.

4.2.2.3.1.2.3 *Ευελιξία*

Όπως σημειώθηκε προηγουμένως, η Πλατφόρμα Ανάπτυξης θα κληθεί να παρέχει μία ποικιλία από λειτουργικότητες και να είναι συμβατή με μία σειρά από ετερογενείς τεχνολογίες. Για τον λόγο αυτό, το περιβάλλον ανάπτυξης θα πρέπει να μπορεί να αναβαθμίζεται και να προσαρμόζεται στις επιχειρησιακές και τεχνολογικές ανάγκες όπως αυτές προκύπτουν από τη μεγάλη ποικιλία εφαρμογών που πρέπει να υποστηρίξει. Για παράδειγμα, μπορεί να απαιτείται η ενσωμάτωση υποστήριξης για νέες γλώσσες, τεχνολογίες, υλικό κτλ. Συνεπώς, το περιβάλλον ανάπτυξης θα πρέπει να μπορεί να αναβαθμίζεται και να προσαρμόζεται στις ανάγκες αυτές, χωρίς να απαιτεί υπέρογκη προσπάθεια, πόρους και κόστος. Άρα, προτείνεται η χρήση τεχνολογιών που να επιτρέπουν μία «αρθρωτή» (modular) αρχιτεκτονική για το περιβάλλον ανάπτυξης,

4.2.2.3.1.2.4 *Διαλειτουργικότητα*

Αναπόσπαστο στοιχείο της αποστολής της Πλατφόρμας Ανάπτυξης είναι η ένταξή της σε ένα συνεκτικό περιβάλλον παροχής υπηρεσιών. Για τον λόγο αυτό, απαιτείται η εφαρμογή βέλτιστων πρακτικών διαλειτουργικότητας, τουλάχιστον εντός του συστήματος αυτού, για

την εκτέλεση λειτουργιών της διαδικασίας ανάπτυξης λογισμικού, όπως π.χ. η παραλαβή μοντέλων υψηλού επιπέδου και η θέση σε λειτουργία των παραγόμενων εφαρμογών.

Στη τελική υλοποίηση, η ανάγκη αυτή εκφράζεται με συγκεκριμένες απαιτήσεις διαλειτουργικότητας. Συγκεκριμένα, η Πλατφόρμα Ανάπτυξης θα πρέπει να διαλειτουργεί με άλλα συστατικά της πληροφοριακής υποδομής του συστήματος υπηρεσιών, και κυρίως την Πλατφόρμα Παροχής Υπηρεσιών, άλλα εργαλεία μοντελοποίησης (μέσω του Αποθετηρίου Μοντέλων), και την Πλατφόρμα Εκτέλεσης Επιχειρησιακών Διαδικασιών. Για τον λόγο αυτό, θα πρέπει να σχεδιαστεί και υλοποιηθεί με τέτοιο τρόπο ώστε να διασφαλίζει την πλήρη διαλειτουργικότητα με τα τμήματα αυτά, καθώς και να τηρεί τα κοινά συμφωνηθέντα πρότυπα και πρωτόκολλα.

4.2.2.3.1.2.5 Ευρωστία και διαχείριση σφαλμάτων

Όπως γίνεται εμφανές από τις λειτουργικές απαιτήσεις, η Πλατφόρμα Ανάπτυξης δέχεται εισόδους από πολλές πηγές, ενώ, ταυτόχρονα, παρέχει ένα σύνθετο περιβάλλον ανάπτυξης λογισμικού. Για τον λόγο αυτό, θα πρέπει να μπορεί να διαχειριστεί επιτυχώς λειτουργικά σφάλματα στη διαδικασία ανάπτυξης, απροσδόκητες εισόδους, χειρισμούς χρήστη κτλ.

Συγκεκριμένα, κρίνονται σημαντικές οι παρακάτω απαιτήσεις:

- Έλεγχος των εισόδων για συμφωνία με πρότυπα
- Περιγραφικά μηνύματα λάθους
- Κλιμακούμενες συνθήκες σφάλματος, αποφυγή συνολική κατάρρευσης της εφαρμογής
- Τακτικά αντίγραφα ασφαλείας των εργασιών του χρήστη
- *Προτείνεται:* Δυνατότητα επαναφοράς σε κανονική λειτουργία μετά από αστοχία της εφαρμογής

4.2.2.3.1.2.6 Ασφάλεια και έλεγχος πρόσβασης

Η Πλατφόρμα Ανάπτυξης είναι ένα ισχυρό εργαλείο μέσα στη πληροφοριακή υποδομή του συστήματος υπηρεσιών, καθώς επιτρέπει την επεξεργασία και δημιουργία νέων βιομηχανικών υπηρεσιών και εφαρμογών. Παράλληλα, επιτρέπει την εγγραφή των υπηρεσιών αυτών στους μηχανισμούς διαχείρισης των υπηρεσιών της πλατφόρμας, δηλαδή στη Πλατφόρμα Παροχής Υπηρεσιών και στη Πλατφόρμα Εκτέλεσης Επιχειρησιακών Διαδικασιών. Επιπλέον, το Αποθετήριο Μοντέλων και η Πλατφόρμα Παροχής Υπηρεσιών περιέχουν εμπιστευτικά στοιχεία ή πνευματικής ιδιοκτησία του οργανισμού/ οργανισμών που χρησιμοποιούν την πληροφοριακή υποδομή και συμμετέχουν στο σύστημα υπηρεσιών. Αυτά μπορούν να περιλαμβάνουν πηγαίο κώδικα, μοντέλα βιομηχανικών υπηρεσιών, περιγραφές web services κτλ. Τέλος, καθώς η πλατφόρμα υποστηρίζει τη συνεργασία αναλυτών και προγραμματιστών, φαίνεται παρέχει σε πολλούς χρήστες λειτουργίες ανάπτυξης λογισμικού και πρόσβασης σε ευαίσθητους πόρους και δεδομένα. Αναμένεται, μάλιστα, ότι οι χρήστες αυτοί θα έχουν διαφορετικές αρμοδιότητες και δικαιώματα πρόσβασης, ενώ μπορούν να ανήκουν σε διαφορετικές οργανωτικές μονάδες και οργανισμούς.

Για τον λόγο αυτό, η Πλατφόρμα Ανάπτυξης θα πρέπει να υποστηρίζει μεθόδους και λειτουργίες για τον διαχωρισμό των δικαιωμάτων πρόσβασης σε λειτουργίες και πόρους του MSEE. Ο έλεγχος πρόσβασης οφείλει να εφαρμόζεται σε δύο πεδία:

- Στο πεδίο πόρων και λειτουργιών της Πλατφόρμας Ανάπτυξης
- Στο πεδίο λειτουργιών που σχετίζονται με την υπόλοιπη πληροφοριακή υποδομή του συστήματος υπηρεσιών

Στο πρώτο πεδίο, για παράδειγμα, η πρόσβαση σε συγκεκριμένα μοντέλα (ή ομάδες μοντέλων) στο Αποθετήριο Μοντέλων μπορεί να περιορίζεται σε συγκεκριμένους χρήστες που σχετίζονται με την ανάπτυξη μίας συγκεκριμένης υπηρεσίας, και ανήκουν σε συγκεκριμένη επιχείρηση.

Στο δεύτερο πεδίο, ο έλεγχος πρόσβασης θα πρέπει να δίνει έμφαση σε ενέργειες χρηστών που αφορούν σε κρίσιμες λειτουργίες και διαλειτουργικότητα με άλλα μέρη της πλατφόρμας του MSEE, π.χ. κατά την εγγραφή νέων υπηρεσιών στη Πλατφόρμα Παροχής Υπηρεσιών.

4.2.3 Προτεινόμενη Αρχιτεκτονική

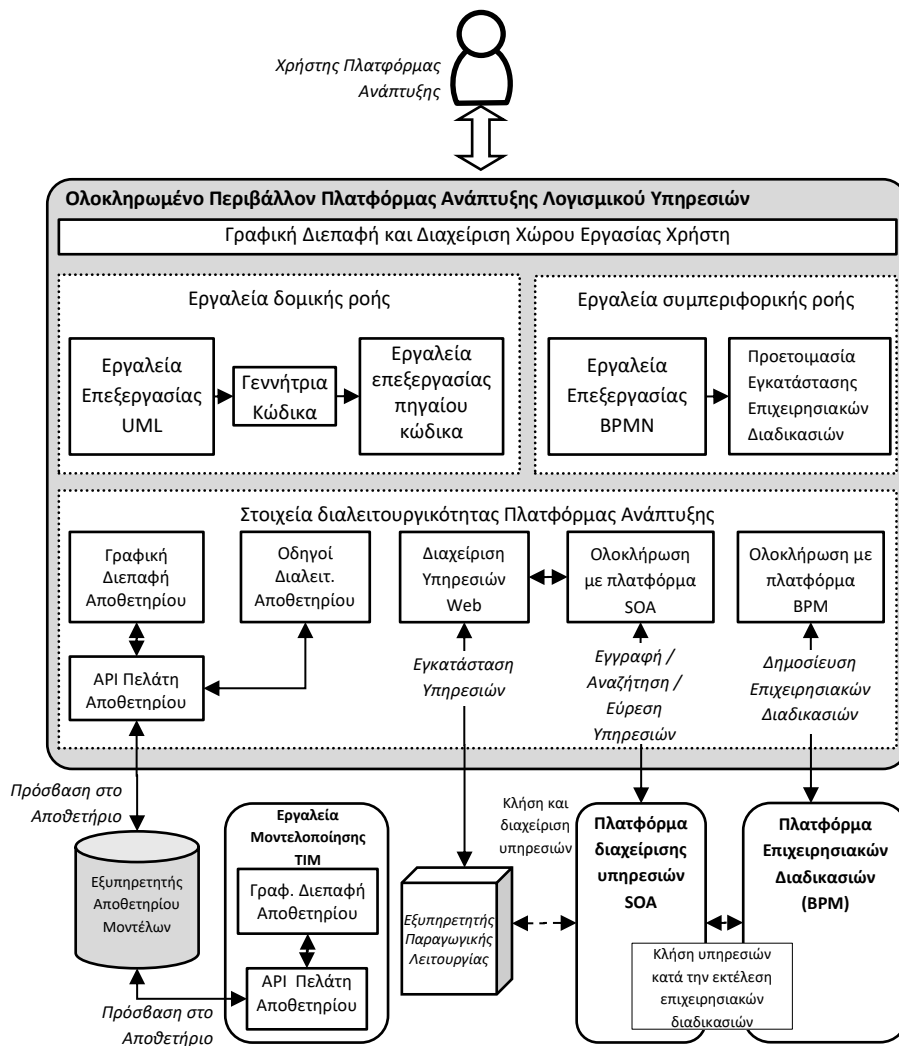
Οι Brog κ.α. (2010), μελέτησαν τις απαιτήσεις σχετικά με τα αρχιτεκτονικά στοιχεία ενός IDE, το οποίο υποστηρίζει ροή εργασίας με μοντέλα. Σύμφωνα με τους ερευνητές, ένα σημαντικό θέμα σε τέτοια περιβάλλοντα είναι ο συνδυασμός διαφόρων οπτικών μοντελοποίησης σε μία ενιαία ροή εργασίας αλλά και η παροχή επαρκούς ολοκλήρωσης των αναγκαίων εργαλείων (με σκοπό τη μείωση των χειροκίνητων μετασχηματισμών και εισαγωγής δεδομένων). Οι ερευνητές απαρίθμησαν τα στοιχεία της αρχιτεκτονικής ενός «Ολοκληρωμένου Περιβάλλοντος Μηχανικής με Μοντέλα» (“Integrated Model Engineering Environment”):

1. Ένα κοινό αποθετήριο μοντέλων
2. Εργαλεία επεξεργασίας μοντέλων
3. Εργαλεία για την ανάλυση και τη σύνθεση νέων προϊόντων, όπως μοντέλα, κώδικας, κείμενο τεκμηρίωσης κ.α.
4. Μία «μηχανή ροής εργασίας» (“workflow engine”) η οποία θα οδηγεί τους χρήστες στα βήματα της διαδικασίας ανάπτυξης.

Η παρουσία του κοινού αποθετηρίου σε ένα περιβάλλον ανάπτυξης με μοντέλα θεωρείται απαραίτητη από τους Haberl κ.α. (2010). Σε συνεργατικά περιβάλλοντα ανάπτυξης, αυτό το κεντρικό αποθετήριο μπορεί να μειώσει τις επαναλήψεις (π.χ. πολλά αντίγραφα των ίδιων στοιχείων) και τις ασυνέπειες (π.χ. σύγχυση σχετικά με τις διαφορετικές εκδόσεις του λογισμικού, των μοντέλων και του κώδικά). Επιπλέον, μπορεί να υποστηρίξει καλύτερα τη «διαχείριση αλλαγών» (π.χ. αλλαγές στις απαιτήσεις του λογισμικού) και τη «διαχείριση διαρρυθμισμού» (configuration management) (π.χ. παρακολούθηση αλλαγών στη διαρρυθμισμό των κοινών πόρων και του συνεργατικού περιβάλλοντος ανάπτυξης)

Τέλος, οι Almeida κ.α. (2007), κατά την εξέταση των εναλλακτικών αρχιτεκτονικών ενός περιβάλλοντος ανάπτυξης υπηρεσιοστραφούς λογισμικού με μοντέλα σημείωσαν την ανάγκη για διασύνδεση του με το περιβάλλον εκτέλεσης των υπηρεσιών. Αυτό, λ.χ. μπορεί να περιλαμβάνει τους μηχανισμούς εκτέλεσης των υπηρεσιών, ή το αποθετήριο μεταδεδομένων σχετικά με τις υπηρεσίες.

Η αρχιτεκτονική της πλατφόρμας ανάπτυξης βασίζεται στην επέκταση αυτής της θεωρητικής έρευνας και την αποκρυστάλλωσή της σε μία συνεκτική αρχιτεκτονική. Η προτεινόμενη αρχιτεκτονική υποστηρίζει τις δύο ροές εργασίας (δομική και συμπεριφορική) της μεθοδολογίας ανάπτυξης, στα πλαίσια ενός συνεργατικού περιβάλλοντος με πολλούς χρήστες. Επιπλέον προβλέπει την ενσωμάτωση των εργαλείων ανάπτυξης λογισμικού σε μία διευρυμένη ροή ανάπτυξης προηγμένων ψηφιακών υπηρεσιών. Αυτό επιτυγχάνεται από τη μία πλευρά μέσω διαλειτουργικότητας με εργαλεία μοντελοποίησης για υψηλότερα επίπεδα αφαίρεσης, δηλαδή με τα στάδια σχεδιασμού του συστήματος υπηρεσιών. Από την άλλη, προβλέπεται η διαλειτουργικότητα με το παραγωγικό περιβάλλον εκτέλεσης των νέων εφαρμογών, δηλαδή τις υποδομές ΤΠΕ του συστήματος υπηρεσιών.



Σχήμα 13: Αρχιτεκτονική Πλατφόρμας Ανάπτυξης

Το Ολοκληρωμένο Περιβάλλον Ανάπτυξης περιλαμβάνει το μεγαλύτερο μέρος της απαιτούμενης λειτουργικότητας. Κάθε χρήστης έχει πρόσβαση στο δικό του αντίγραφο του περιβάλλοντος ανάπτυξης. Η αρχιτεκτονική του περιλαμβάνει ένα κοινό πλαίσιο που προσφέρει ολοκλήρωση πλατφόρμας, παρουσίασης, ελέγχου, δεδομένων και διαδικασίας (Wasserman, 1990). Προσφέρει στον χρήστη μία γραφική διεπαφή και έναν «χώρο

εργασίας», με τα εργαλεία και στοιχεία (μοντέλα, κώδικας, κ.α.) που απαιτούνται για την διαδικασία ανάπτυξης. Η αρχιτεκτονική απεικονίζεται στο Σχήμα 13.

Συνοπτικά, η αρχιτεκτονική αποτελείται από τα ακόλουθα στοιχεία:

- *Κεντρικά στοιχεία*, τα οποία υποστηρίζουν τη βασική λειτουργικότητα του περιβάλλοντος ανάπτυξης, δηλαδή επεξεργασία, αντιστοίχιση, εμπλουτισμός και μετασχηματισμός μοντέλων, παραγωγή πηγαίου κώδικα και επεξεργασία του. Αυτά είναι οργανωμένα με βάση τις δύο υποστηριζόμενες ροές εργασίας: Δομική και Συμπεριφορική
- *Στοιχεία διαλειτουργικότητας*, τα οποία διασυνδέουν την πλατφόρμα ανάπτυξης με άλλα, εξωτερικά συστήματα. Αυτά είναι το αποθετήριο μοντέλων και η υποδομή ΤΠΕ του συστήματος παροχής υπηρεσιών.
- *Το Αποθετήριο Μοντέλων*, το οποίο είναι μία κοινή αποθήκη για μοντέλα και άλλα προϊόντα της ανάπτυξης λογισμικού, όπως πηγαίος κώδικας, τεκμηρίωση κ.α. Ένα μοναδικό Αποθετήριο Μοντέλων εξυπηρετεί ταυτόχρονα πολλούς χρήστες της Πλατφόρμας Ανάπτυξης. Στο Αποθετήριο Μοντέλων έχουν επίσης πρόσβαση σχεδιαστές και αναλυτές υπηρεσιών, στο οποίο αποθηκεύουν μοντέλα επιπέδου TIM ως εισόδους στις ροές εργασίας της Πλατφόρμας Ανάπτυξης.

4.2.3.1 Στοιχεία Δομικής Ροής

Τα στοιχεία της δομικής ροής περιλαμβάνουν τα ακόλουθα:

Τα στοιχεία της δομικής ροής περιλαμβάνουν τα ακόλουθα:

- *Εργαλεία Επεξεργασίας UML*: Χρησιμοποιούνται για την επεξεργασία και την αντιστοίχιση διαγράμματα κλάσεων χωρίς στοιχεία τεχνολογίας-στόχου (TIM) σε διαγράμματα κλάσεων εστιασμένα σε συγκεκριμένη τεχνολογία (TSM). Περιλαμβάνει τη δυνατότητα επεξεργασίας UML και τη δυνατότητα της εισαγωγής και εφαρμογής προφίλ UML σε μοντέλα. Τα διαγράμματα κλάσεων μπορούν να προέρχονται από το Αποθετήριο Μοντέλων ή να δημιουργούνται από την αρχή στο περιβάλλον σχεδιασμού.
- *Γεννήτρια Κώδικα*: Παράγει πλαίσια κλάσεων αντικειμενοστραφούς κώδικα από το διάγραμμα UML με στοιχεία τεχνολογίας. Χρησιμοποιεί ένα σύνολο κανόνων μετασχηματισμού εστιάζοντας τη συγκεκριμένη αντικειμενοστραφή γλώσσα.
- *Εργαλεία ανάπτυξης κώδικα*: Τυπικά εργαλεία για την ανάπτυξη λογισμικού, όπως επεξεργαστές κειμένου, μεταγλωττιστές, εργαλεία αποσφαλμάτωσης κ.α.

Τα αποτελέσματα της χρήσης των παραπάνω στοιχείων μπορεί να είναι και ενδιάμεσα προϊόντα (όπως κώδικας και μοντέλα) που μπορεί να αποθηκευτούν στο Αποθετήριο Μοντέλων, αυτόνομες εφαρμογές, και web services.

4.2.3.2 Στοιχεία Συμπεριφορικής Ροής

Τα στοιχεία της συμπεριφορικής ροής περιλαμβάνουν τα ακόλουθα:

- *Εργαλεία επεξεργασίας BPMN*: Χρησιμοποιείται για την επεξεργασία διαγραμμάτων επιχειρησιακών διαδικασιών BPMN 2.0, και για τον εμπλουτισμό τους με σημασιολογικά στοιχεία της τεχνολογίας στόχου. Ο εμπλουτισμός αφορά προσαρμογή στη μηχανή

εκτέλεσης επιχειρησιακών διαδικασιών αλλά και στη χρήση άλλων πόρων του συστήματος ΤΠΕ, όπως οι διαθέσιμες web services. Τα μοντέλα TIM μπορούν να εισάγονται από το Αποθετήριο Μοντέλων ή να δημιουργούνται από την αρχή στο συγκεκριμένο περιβάλλον. The TIM models can be imported from the Model Repository or created entirely in this environment.

- *Προετοιμασία Εγκατάστασης Επιχειρησιακών Διαδικασιών:* Πριν την εγκατάσταση στη συγκεκριμένη πλατφόρμα εκτέλεσης επιχειρησιακών διαδικασιών, μπορεί να απαιτείται περαιτέρω επεξεργασία ή μετασχηματισμός, ανάλογα με τις τεχνολογίες που χρησιμοποιούνται.

Τα ενδιάμεσα προϊόντα (π.χ. διαγράμματα επιχειρησιακών διαδικασιών, εκτελέσιμες διαδικασίες) μπορούν να εξαχθούν στο Αποθετήριο Μοντέλων, ενώ οι εκτελέσιμες διαδικασίες εγκαθίστανται στη Πλατφόρμα Εκτέλεσης – Διαχείρισης Επιχειρησιακών Διαδικασιών της υποδομής ΤΠΕ του συστήματος υπηρεσιών.

4.2.3.3 Ολοκλήρωση με την υποδομή ΤΠΕ του συστήματος βιομηχανικών υπηρεσιών

Μεγάλο μέρος της προστιθέμενης αξίας της Πλατφόρμας Ανάπτυξης προέρχεται από τις διασυνδέσεις της με τα συστήματα ΤΠΕ του συστήματος υπηρεσιών. Οι είσοδοί της προέρχονται από εργαλεία μοντελοποίησης που χρησιμοποιούν σχεδιαστές και αναλυτές υπηρεσιών, και τα αποτελέσματά της (web services, εκτελέσιμα, επιχειρησιακές διαδικασίες) εγκαθίστανται και εκτελούνται από τους εξυπηρετητές του συστήματος. Η συνεργασία μεταξύ των εμπλεκόμενων (π.χ. των επαγγελματιών του πεδίου εφαρμογής, των επιχειρησιακών αναλυτών και των προγραμματιστών) απαιτεί την πρόσβαση σε κοινούς πόρους. Ακόμη, η δημιουργία επιχειρησιακών διαδικασιών με τη χρήση συνθέσεων υπηρεσιών απαιτεί την εύρεση και κλήση των κατάλληλων διαθέσιμων web services, στην υποδομή SOA του συστήματος υπηρεσιών. Τέλος, τα ολοκληρωμένα προϊόντα της εγκαθίστανται στο περιβάλλον παραγωγικής λειτουργίας (εξυπηρετητές και συστήματα εκτέλεσης επιχειρησιακών διαδικασιών) του συστήματος υπηρεσιών. Η διαλειτουργικότητα αυτή επιτυγχάνεται μέσα από μία σειρά συστημάτων διαλειτουργικότητας που αναλύονται στις παρακάτω ενότητες.

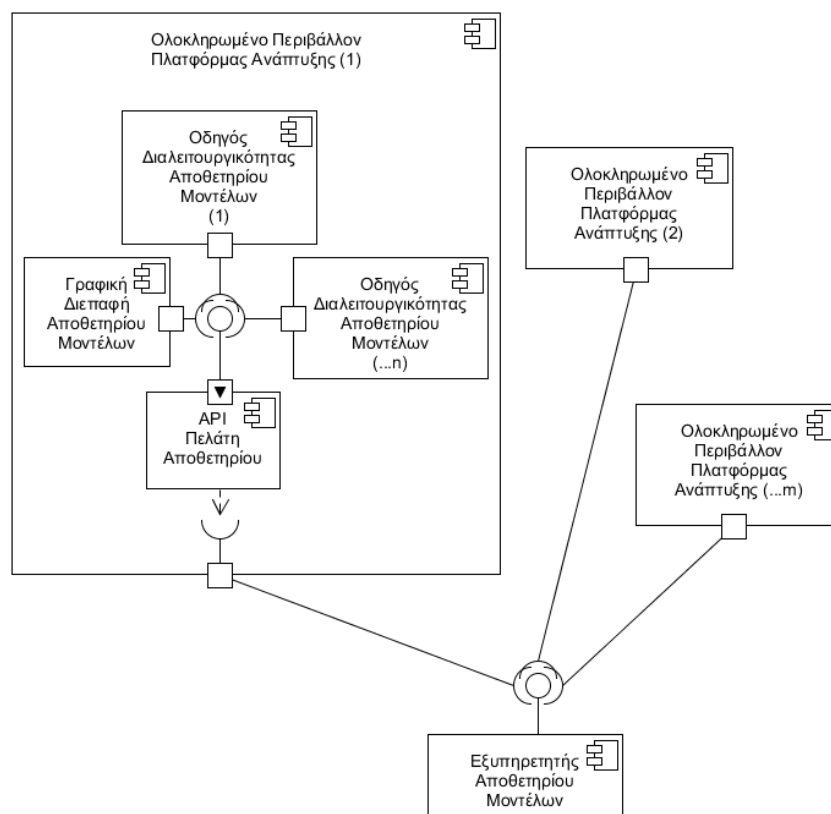
Στοιχεία πρόσβασης στο Αποθετήριο Μοντέλων

Η Πλατφόρμα Ανάπτυξης χρησιμοποιεί το Αποθετήριο Μοντέλων για την αποθήκευση, αναζήτηση και ανάκληση μοντέλων, κώδικα και άλλων πόρων που χρησιμοποιούνται κατά την ανάπτυξη λογισμικού. Πολλαπλά στιγμιότυπα της Πλατφόρμας Ανάπτυξης (που αντιστοιχούν σε διαφορετικούς χρήστες) επικοινωνούν με ένα κεντρικό εξυπηρετητή Αποθετηρίου Μοντέλων διαμέσου μίας ενσωματωμένης «στίβας» (stack) διαλειτουργικότητας, όπως φαίνεται στο Σχήμα 14.

Η λειτουργικότητα του Αποθετηρίου Μοντέλων περιέχεται στα ακόλουθα συστατικά:

- *Εξυπηρετητής Αποθετηρίου Μοντέλων:* Ενεργεί ως ένα κεντρικό «μοναδικό αποθετήριο αναφοράς» για μοντέλα και πόρους λογισμικού (π.χ. πηγαίο κώδικά, κείμενο και τεκμηρίωση).

- *API Πελάτη Αποθετηρίου*: Επικοινωνεί απευθείας με τον Εξυπηρετητή Αποθετηρίου Μοντέλων μέσω του API που αυτός εκθέτει. Ταυτόχρονα όμως, παρέχει ένα ενδιάμεσο API (Application Programming Interface) ώστε στοιχεία της Πλατφόρμας Ανάπτυξης (όπως γραφικές διεπαφές και οδηγοί διαλειτουργικότητας) να μπορούν να επικοινωνήσουν με το Αποθετήριο Μοντέλων.
- *Γραφική Διεπαφή Αποθετηρίου*: Το στοιχείο αυτό επιτρέπει στους τελικούς χρήστες της Πλατφόρμας Ανάπτυξης την αλληλεπίδραση με το Αποθετήριο Μοντέλων με γραφικό τρόπο. Επικοινωνεί εσωτερικά στη Πλατφόρμα Ανάπτυξης με τον API Πελάτη Αποθετηρίου.
- *Οδηγοί Διαλειτουργικότητας Αποθετηρίου*: Παρέχουν απλοποιημένες διεπαφές με τη μορφή οδηγών (wizards) για την εκτέλεση συγκεκριμένων ενεργειών με το Αποθετήριο Μοντέλων, όπως Εισαγωγή/Εξαγωγή Μοντέλων, Αναζήτηση Μοντέλων με σημασιολογικά κριτήρια κ.α. Ο σκοπός τους είναι η επιτάχυνση κοινών δραστηριοτήτων με σκοπό τη διευκόλυνση της διαδικασίας ανάπτυξης.



Σχήμα 14: Στοιχεία πρόσβασης και λειτουργικότητας Αποθετηρίου Μοντέλων

Ολοκλήρωση με τα συστήματα διαχείρισης υπηρεσιών SOA

Η Πλατφόρμα Ανάπτυξης χρησιμοποιείται για την δημιουργία web services για ένα πληροφοριακό σύστημα που βασίζεται στις αρχές της SOA και υποστηρίζει λειτουργίες όπως εγγραφή και αναζήτηση υπηρεσιών. Εδώ η επικοινωνία είναι δικατευθυντική με την Πλατφόρμα Διαχείρισης Υπηρεσιών SOA και συμβαίνει σε δύο περιπτώσεις:

- *Εγγραφή υπηρεσιών:* Όταν ένα web service εγκαθίσταται στον αντίστοιχο εξυπηρετητή παραγωγικής λειτουργίας, το στοιχεί αυτό αυτοματοποιεί τη διαδικασία εγγραφής του στην υποδομή διαχείρισης υπηρεσιών. Η εγγραφή περιλαμβάνει την καταχώρηση μεταδεδομένων σχετικά με το web service, τα οποία μπορεί να είναι τεχνικά (π.χ. σημείο κλήσης, λίστα παραμέτρων), σημασιολογικά δεδομένα όπως tags για τον τύπο web service και τη λειτουργικότητα που προσφέρεται (π.χ. «λογιστική») και στοιχεία τεκμηρίωσης όπως ελεύθερο κείμενο κ.α.
- *Αναζήτηση υπηρεσιών:* Επιτρέπει την αναζήτηση και ανάκληση δεδομένων σχετικά με τις διαθέσιμες υπηρεσίες κατά τη διάρκεια ανάπτυξης λογισμικού. Τα δεδομένα που ανακαλούνται περιλαμβάνουν τα στοιχεία που καταχωρούνται στην υποδομή διαχείρισης SOA κατά την εγγραφή υπηρεσιών. Οι πληροφορίες χρησιμοποιούνται για την ενσωμάτωση κλήσεων προς τις υπηρεσίες αυτές κατά την ανάπτυξη πηγαίου κώδικα ή την ενσωμάτωση κλήσεων προς αυτές σε επιχειρησιακές διαδικασίες.

Διαχείριση Υπηρεσιών Web / Ολοκλήρωση με πλατφόρμα BPM

Ο σκοπός των στοιχείων αυτών είναι η αυτοματοποίηση και απλοποίηση της εγκατάστασης νέων web services ή επιχειρησιακών διαδικασιών στις αντίστοιχες υποδομές του συστήματος υπηρεσιών. Συγκεκριμένα:

- *Διαχείριση Υπηρεσιών Web:* Χειρίζεται την εγκατάσταση του web service στον αντίστοιχο εξυπηρετητή παραγωγικής λειτουργίας.
- *Ολοκλήρωση με πλατφόρμα BPM:* Χειρίζεται την εγκατάσταση της επιχειρησιακής διαδικασίας στην υποδομή διαχείρισης και εκτέλεσης επιχειρησιακών διαδικασιών.

4.2.4 Υποστήριξη των ροών εργασίας από την προτεινόμενη αρχιτεκτονική.

Η προτεινόμενη αρχιτεκτονική της Πλατφόρμας Ανάπτυξης είναι σχεδιασμένη ώστε να εξυπηρετεί τη δομική και συμπεριφορική ροή εργασίας, και να παρέχει διαλειτουργικότητα με τα άλλα στοιχεία των υποδομών ΤΠΕ του συστήματος υπηρεσιών. Στις επόμενες ενότητες περιγράφεται ο τρόπος με τον οποίο επιτυγχάνεται αυτό.

4.2.4.1 Δομική ροή εργασίας

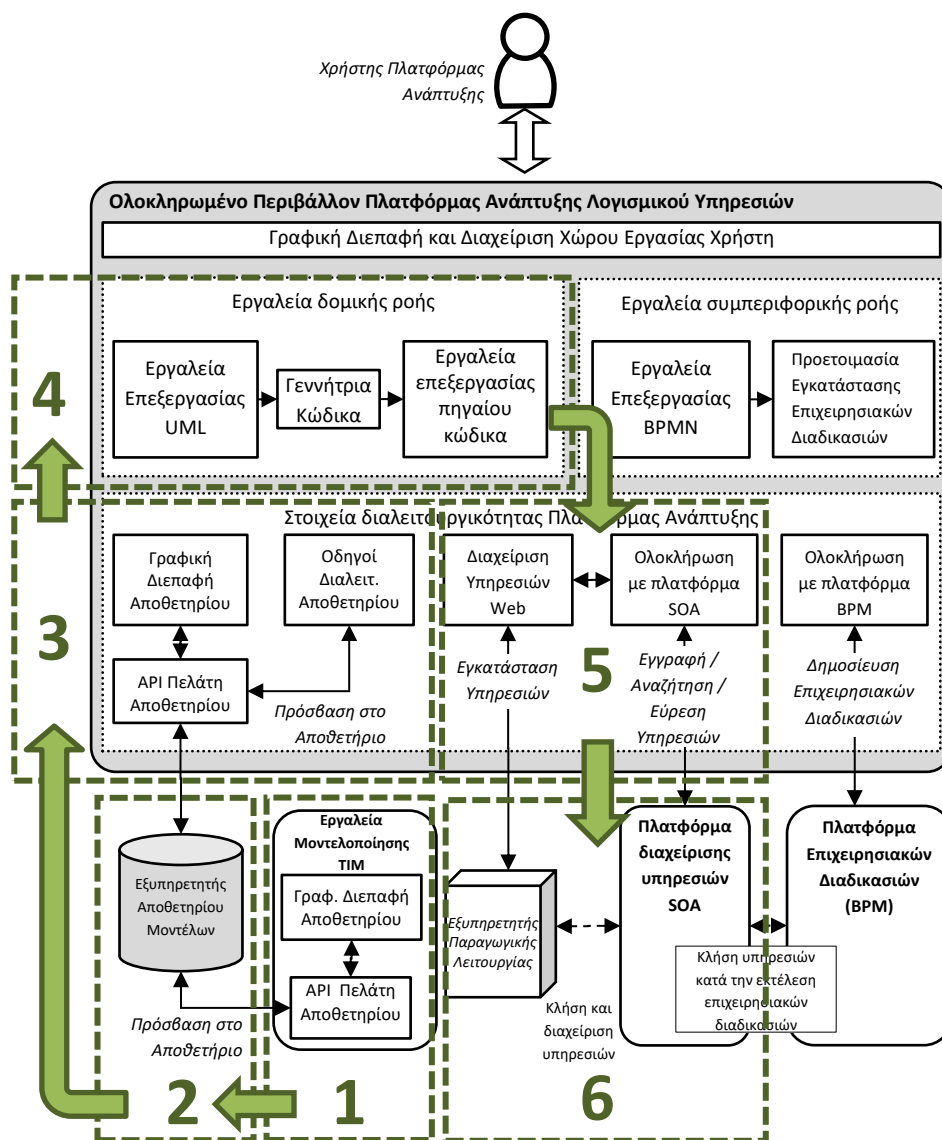
Όπως παρουσιάζεται στο Σχήμα 15, τα μέρη της αρχιτεκτονικής συνδυάζονται ώστε να διευκολύνουν ή να εκτελέσουν τα βήματα της δομικής ροής εργασίας:

Βήμα 1: Η δομική μοντελοποίηση στο επίπεδο TIM λαμβάνει χώρα εκτός της πλατφόρμας ανάπτυξης, από επιχειρησιακούς αναλυτές, σχεδιαστές υπηρεσιών και άλλους χρήστες. Στους χρήστες αυτούς παρέχονται εφαρμογές και στοιχεία διαλειτουργικότητας ανάλογα με αυτά της Πλατφόρμας Ανάπτυξης, είτε σαν αυτόνομες εφαρμογές ή σαν μέρη του περιβάλλοντος σχεδιασμού που χρησιμοποιούν. Ο σκοπός είναι η κοινή πρόσβαση στο Αποθετήριο μοντέλων από όλα τα μέλη της ομάδας ανάπτυξης υπηρεσιών (αναλυτές, σχεδιαστές, μηχανικούς λογισμικού κ.α.).

Βήμα 2: Οι σχεδιαστές αποθηκεύουν τα μοντέλα επιπέδου TIM στον Εξυπηρετητή Αποθετηρίου Μοντέλων.

Βήμα 3: Ο μηχανικός λογισμικού ανασύρει τα μοντέλα από το Αποθετήριο Μοντέλων. Δύο μέθοδοι μπορούν να χρησιμοποιηθούν: είτε μέσω της Γραφικής Διεπαφής Αποθετηρίου μοντέλων, είτε με τη χρήση των Οδηγών Διαλειτουργικότητας Αποθετηρίου.

Βήμα 4: Ο μηχανικός λογισμικού εργάζεται πάνω στον μετασχηματισμό του μοντέλου TIM σε μοντέλο TSM, και την παραγωγή του πηγαίου κώδικα. Τα Εργαλεία Επεξεργασίας UML χρησιμοποιούνται για την επεξεργασία και εμπλουτισμό του μοντέλου με σημασιολογικά στοιχεία που αφορούν την τεχνολογία – στόχο. Ο πηγαίος κώδικας παράγεται από το μοντέλο TSM μέσω της γεννήτριας κώδικας. Ο μηχανικός λογισμικού επεξεργάζεται, συμπληρώνει, αφαιρεί σφάλματα και δοκιμάζει τον πηγαίο κώδικα χρησιμοποιώντας συμβατικά εργαλεία ανάπτυξης λογισμικού ενσωματωμένα στη Πλατφόρμα Ανάπτυξης. Το τελικό προϊόν είναι η εκτελέσιμα αρχεία web services.



Σχήμα 15: Η δομική ροή εργασίας όπως υλοποιείται από τα αρχιτεκτονικά στοιχεία της Πλατφόρμας Ανάπτυξης, οργανωμένη σε βήματα

Βήμα 5: Μόλις το web service είναι έτοιμο, ο μηχανικός το εγκαθιστά στον αντίστοιχο εξυπηρετητή παραγωγικής λειτουργίας, χρησιμοποιώντας τη λειτουργικότητα της Διαχείρισης Υπηρεσιών Web. Παράλληλα εγγράφει τη νέα υπηρεσία στη Πλατφόρμα Διαχείρισης Υπηρεσιών SOA, διαμέσου του στοιχείου Ολοκλήρωσης με την Πλατφόρμα SOA.

Βήμα 6: Η υπηρεσία web μπαίνει σε παραγωγική λειτουργία, εγγράφεται στην υποδομή SOA και είναι έτοιμη προς χρήση από επιχειρησιακές διαδικασίες ή ως αυτόνομη υπηρεσία από άλλες εφαρμογές.

Σε όλη τη διάρκεια της διαδικασίας, ο μηχανικός λογισμικού μπορεί να έχει, ανά πάσα στιγμή, πρόσβαση στον Εξυπηρετητή Αποθετηρίου Μοντέλων διαμέσου των αντίστοιχων μηχανισμών διαλειτουργικότητας. Έτσι, έχει τη δυνατότητα ανάκλησης και αποθήκευσης προϊόντων της διαδικασίας ανάπτυξης λογισμικού (κώδικα, μοντέλα, κείμενα, αρχεία κ.α.) στο Αποθετήριο μοντέλων. Καθώς όλα τα μέλη της ομάδας (προγραμματιστές, σχεδιαστές κ.α.) έχουν πρόσβαση στο Αποθετήριο Μοντέλων, αυτό μπορεί να χρησιμοποιηθεί και σαν εργαλεία συνεργασίας ομάδας.

4.2.4.2 Συμπεριφορική ροή εργασίας

Το Σχήμα 16 παρουσιάζει τα αρχιτεκτονικά στοιχεία που χρησιμοποιούνται στη συμπεριφορική ροή, οργανωμένα στα αντίστοιχα βήματα της διαδικασίας.

Κατά τα βήματα της συμπεριφορικής ροής, τα μέρη της Πλατφόρμας Ανάπτυξης συμμετέχουν ως εξής:

Βήμα 1: Η μοντελοποίηση της συμπεριφοράς της επιχειρησιακής διαδικασίας λαμβάνει χώρα εκτός της Πλατφόρμας Ανάπτυξης από σχεδιαστές υπηρεσιών, επιχειρησιακούς αναλυτές και άλλους χρήστες, όπως και στη δομική ροή εργασίας.

Βήμα 2: Οι σχεδιαστές ανεβάζουν τα μοντέλα επιπέδου TIM στον κοινό Εξυπηρετητή Αποθετηρίου Μοντέλων.

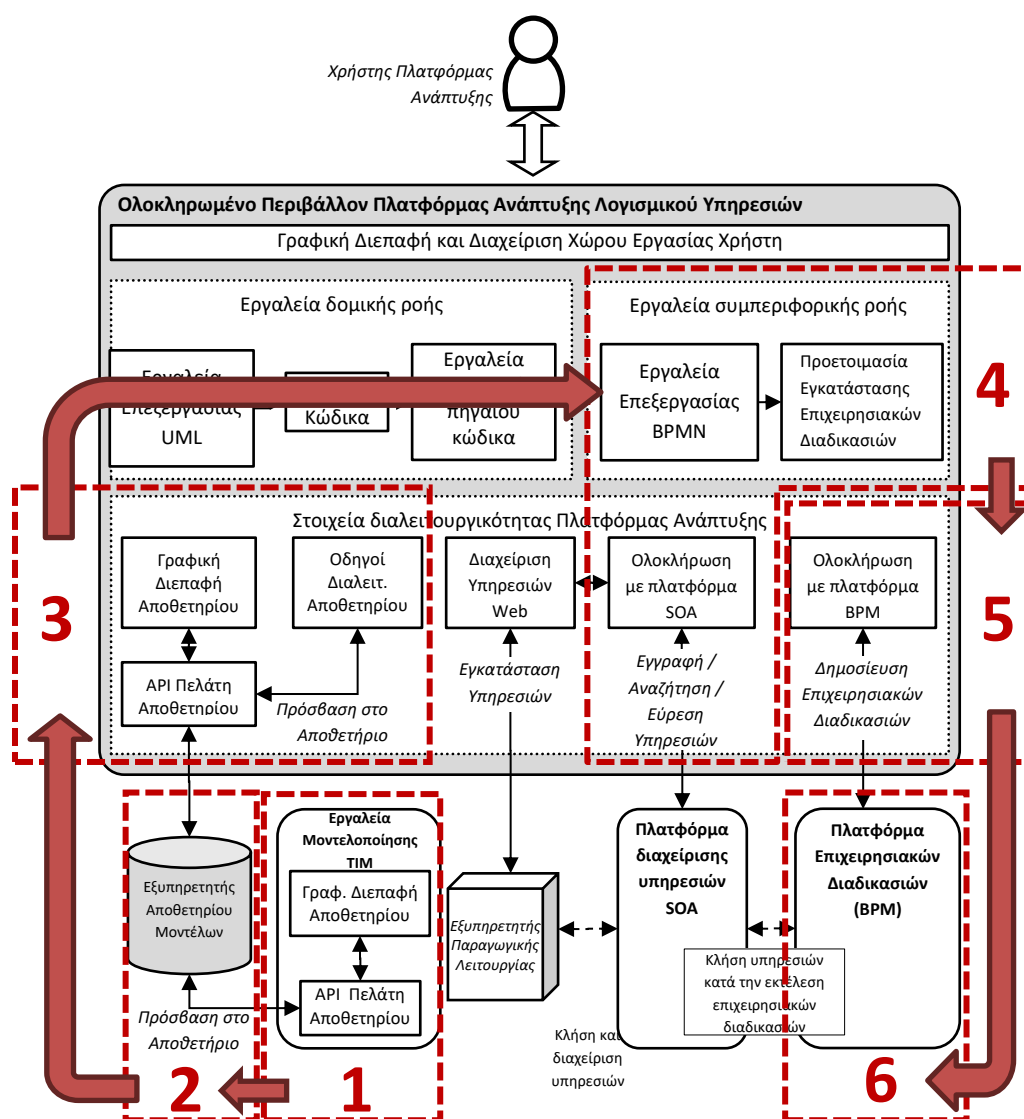
Βήμα 3: Ο μηχανικός λογισμικού ανασύρει το μοντέλο της επιχειρησιακής διαδικασίας από το Αποθετήριο Μοντέλων, όπως και στη δομική ροή εργασιών.

Βήμα 4: Ο μηχανικός εργάζεται πάνω στο μετασχηματισμό του μοντέλου TIM σε μοντέλο TSM, δηλαδή σε μία εκτελέσιμη διαδικασία BPMN 2.0. Τα Εργαλεία Επεξεργασίας BPMN χρησιμοποιούνται για την επεξεργασία του μοντέλου και τον εμπλουτισμό του με στοιχεία της τεχνολογίας – στόχου (δηλαδή της μηχανής εκτέλεσης επιχειρησιακών διαδικασιών του συστήματος υπηρεσιών). Κατά τη διάρκεια αυτής της διαδικασίας, ο μηχανικός λογισμικού μπορεί να αναζητήσει για κατάλληλες υπηρεσίες web που είναι διαθέσιμες στην υποδομή ΤΠΕ και είναι ήδη εγγεγραμμένες στη Πλατφόρμα Διαχείρισης Υπηρεσιών SOA. Η λειτουργικότητα Ολοκλήρωσης με την Πλατφόρμα SOA χρησιμοποιείται για την αναζήτηση και εισαγωγή περιγραφών υπηρεσιών από την Πλατφόρμα SOA. Στη συνέχεια, το στοιχείο Προετοιμασίας Εγκατάστασης Επιχειρησιακών Διαδικασιών χρησιμοποιείται για να ετοιμάσει την ολοκληρωμένη διαδικασία BPMN για καταχώρηση στη Πλατφόρμα Εκτέλεσης/Διαχείρισης Επιχειρησιακών Διαδικασιών (Πλατφόρμα BPM) του συστήματος βιομηχανικών υπηρεσιών.

Βήμα 5: Το στοιχείο Ολοκλήρωσης με την Πλατφόρμα BPM διαλειτουργεί με την Πλατφόρμα Επιχειρησιακών Διαδικασιών για την εγγραφή της διαδικασίας.

Βήμα 6: Η επιχειρησιακή διαδικασία έχει εγκατασταθεί και μπορεί να χρησιμοποιηθεί από τους τελικούς χρήστες του συστήματος βιομηχανικών υπηρεσιών.

Πάλι, όπως και με τη δομική ροή εργασίας, ο μηχανικός λογισμικού μπορεί να σώσει και να επαναφέρει ενδιάμεσα προϊόντα της δομικής ροής εργασιών (μοντέλα, επιχειρησιακούς κανόνες, έγγραφα και άλλους πόρους) στο Αποθετήριο Μοντέλων, καθώς και να το χρησιμοποιήσει για τη συνεργασία του με άλλους εμπλεκόμενους.



Σχήμα 16: Η συμπεριφορική ροή εργασίας όπως υλοποιείται από τα αρχιτεκτονικά στοιχεία της Πλατφόρμας Ανάπτυξης, οργανωμένη σε βήματα

5 Εφαρμογή και αξιολόγηση

5.1 Εισαγωγή

Η προτεινομένη μεθοδολογία και αρχιτεκτονική αξιολογήθηκαν μέσω της πιλοτικής τους εφαρμογής σε μια σειρά από βιομηχανικά case studies.

Στα πλαίσια του ερευνητικού έργου MSEE (Manufacturing Service Ecosystem) έγινε εφικτή η συνεργασία και ανταλλαγή τεχνογνωσίας με άλλους ακαδημαϊκούς και τεχνικούς εταίρους στο πεδίο των συστημάτων παροχής βιομηχανικών υπηρεσιών. Οι εταίροι ανέπτυξαν μεθοδολογίες για την υποστήριξη της υπηρεσιοποίησης, και σχεδίασαν τμήματα μίας συνολικής πλατφόρμας ΤΠΕ για την δημιουργία και παροχή βιομηχανικών υπηρεσιών. Η μεθοδολογία και τα εργαλεία ανάπτυξης λογισμικού της παρούσας έρευνας ενσωματώνονται σε αυτό το ευρύτερο πλαίσιο. Επίσης, το έργο έδωσε την ευκαιρία για την δοκιμή της προτεινόμενης μεθοδολογίας και εργαλείου σε πιλοτικές εφαρμογές στη βιομηχανία. Αυτό έγινε εφικτό καθώς τέσσερις βιομηχανικές επιχειρήσεις συμμετείχαν στο έργο ως εταίροι, δίνοντας πρόσβαση σε ένα αρχικό οικοσύστημα δυνητικών χρηστών.

Έτσι, αναπτύχθηκε ένα πρωτότυπο περιβάλλον ανάπτυξης βασισμένο στη πρότυπη αρχιτεκτονική που περιγράφεται στις προηγούμενες ενότητες. Το περιβάλλον ανάπτυξης χρησιμοποιήθηκε για τη δημιουργία εφαρμογών για την παροχή υπηρεσιών σε συνδυασμό με βιομηχανικά προϊόντα, και αξιολογήθηκε από τους μηχανικούς λογισμικού που το χρησιμοποίησαν. Το έργο MSEE, το πρωτότυπο, η διαδικασία και τα αποτελέσματα αξιολόγησης περιγράφονται στις ακόλουθες ενότητες.

5.2 Το ερευνητικό Έργο MSEE

5.2.1 Στόχοι και αποτελέσματα

Το MSEE (Manufacturing Service Ecosystem) υπήρξε ερευνητικό έργο χρηματοδοτούμενο από την Ευρωπαϊκή Ένωση, στα πλαίσια του Έβδομου Προγράμματος – Πλαισίου (7th Framework Programme – FP7). Η ερευνητική προσπάθεια έρχεται σαν απάντηση σε δύο μεγάλες προκλήσεις:

Η προσαρμογή της επιστήμης και μηχανικής υπηρεσιών στις εξελίξεις των σύγχρονων βιομηχανικών συστημάτων παραγωγής και των εργοστασίων του μέλλοντος: Συγκεκριμένα, η προσαρμογή μεθόδων της επιστήμης και μηχανικής υπηρεσιών ώστε να μπορούν να χρησιμοποιηθούν από παραδοσιακές βιομηχανικές επιχειρήσεις, και να δημιουργήσουν αρχιτεκτονικές και πλατφόρμες για παγκοσμιοποιημένα συστήματα παραγωγής

Ο μετασχηματισμός των σύγχρονων, ιεραρχικών, εφοδιαστικών αλυσίδων σε ανοικτά βιομηχανικά οικοσυστήματα: Ορισμός και υλοποίηση μεθόδων για συνεργατική καινοτομία στη βιομηχανία και αρχιτεκτονικές για συνεργασίες και λήψη αποφάσεων με βάση τις ΤΠΕ.

Οι ανάγκες αυτές προκύπτουν από τις εξελίξεις στο οικονομικό περιβάλλον της βιομηχανίας, όπου υπάρχει η σύγκλιση πολλών παραγόντων: του ανταγωνισμού μέσω της ενσωμάτωσης υπηρεσιών στα βιομηχανικά προϊόντα, καθώς και των νέων εξελίξεων στις ΤΠΕ και στις μεθόδους της βιομηχανικής παραγωγής. Οι συνθήκες αυτές δείχνουν ότι, για τις

βιομηχανικές επιχειρήσεις, δεν ενδείκνυται η αποκλειστική επικέντρωση στα υλικά προϊόντα και τις διαδικασίες παραγωγής και διάθεσής. Αντίθετα, μεγάλη σημασία έχει πλέον και η επένδυση στις υπηρεσίες και τις επιχειρηματικές διαδικασίες που οδηγούν στον σχεδιασμό και την παροχή τους.

Για τον λόγο αυτό, το έργο MSEE εστίασε στην καινοτομία στις υπηρεσίες και τη μηχανική υπηρεσιών στη βιομηχανία. Αποσκοπούσε στην ανάπτυξη μοντέλων, διαδικασιών και εργαλείων με απώτερο στόχο την ισότιμη διασύνδεση μεταξύ της μηχανικής υπηρεσιών και προϊόντων κατά μήκος του ενοποιημένου κύκλου ζωής προϊόντος – υπηρεσίας. Από τεχνολογική σκοπιά, το MSEE στόχευσε στην υλοποίηση μίας τεχνολογικής πλατφόρμας για καινοτομικές υπηρεσίες στη βιομηχανία, η οποία θα υποστηρίζει την αλληλεπίδραση ανθρώπων, υπηρεσιών και υλικών αντικειμένων στη διάρκεια του κύκλου ζωής τους.

Κεντρική έννοια του ερευνητικού έργου ήταν το «εκτεταμένο προϊόν» (ΕΠ - extended product). Το ΕΠ είναι ένα Σύστημα – Προϊόντος Υπηρεσίας το οποίο στηρίζεται σε ένα κεντρικό υλικό προϊόν. Η σταδιακή ενσωμάτωση υπηρεσιών σε αυτό μπορεί να γίνει αντιληπτή ως η προσθήκη «στρωμάτων υπηρεσίας» γύρω από το κεντρικό υλικό προϊόν, τα οποία το εμπλουτίζουν και το «επεκτείνουν». Στη πρώτη φάση, τα στρώματα αυτά μπορούν να περιλαμβάνουν απλώς υποστηρικτικές υπηρεσίες, π.χ. συντήρηση, οι οποίες είναι απαραίτητες για τη λειτουργία του προϊόντος. Στο επόμενο στάδιο, οι επιπλέον μπορούν να αποτελέσουν στοιχείο διαφοροποίησης. Η βιομηχανική επιχείρηση μπορεί να κινηθεί περαιτέρω προς την κατεύθυνση των υπηρεσιών, αφαιρώντας το φυσικό προϊόν από το προσφερόμενο πακέτο. Έτσι προσφέρει το προϊόν με τη μορφή υπηρεσίας, π.χ. με τη μίσθωσή του στον πελάτη ή την απλή παροχή των αποτελεσμάτων της χρήσης του. (Ducq, Chen, & Alix, 2012)

Καθώς η μετάβαση προς την ενσωμάτωση υπηρεσιών γίνεται με σταδιακό τρόπο γύρω από ένα εδραιωμένο προϊόν, η συγκεκριμένη θεώρηση είναι κατάλληλη για χρήση από βιομηχανικές επιχειρήσεις, οι οποίες ξεκινούν από έναν προσανατολισμό παραγωγής και προϊόντος.

Στο MSEE, η έννοια του ΕΠ ως κλειστού συστήματος προϊόντος και σχετικών υπηρεσιών τροποποιείται για να καλύψει επιπλέον πλευρές της ενσωμάτωσης υπηρεσιών. Έτσι, ένα ΕΠ μπορεί να διασυνδέεται με εξωτερικές υπηρεσίες (π.χ. διασύνδεση με κοινωνικά δίκτυα), μπορεί να απαιτεί τροποποίηση του προϊόντος για την παροχή υπηρεσιών (π.χ. την προσθήκη αισθητήρων), μπορεί να απαιτεί προσαρμογή των υπηρεσιών στο προϊόν, αλλά και να περιλαμβάνει παραδοσιακές υποστηρικτικές υπηρεσίες (π.χ. εγκατάσταση και συμβουλευτική υποστήριξη) (Wiesner, Westphal, Hirsch, & Thoben, 2013).

Το έργο έχει ολοκληρωθεί από τον Σεπτέμβριο του 2014, και έχει οδηγήσει σε μία σειρά από ερευνητικά αποτελέσματα. Συνοπτικά, τα αποτελέσματα αυτά μπορούν να ταξινομηθούν σε τρεις ευρείες κατηγορίες:

Γενικά αποτελέσματα: εργαλεία για την υποστήριξη των βιομηχανικών επιχειρήσεων στην ενσωμάτωση υπηρεσιών στη λειτουργία τους, όπως μοντέλα στρατηγικής, ωριμότητας, πρότυπες επιχειρησιακές αρχιτεκτονικές και εργαλεία ΤΠΕ για την υποστήριξη του κύκλου ζωής των βιομηχανικών υπηρεσιών (π.χ. σύλληψη, ανάπτυξη, παροχή κτλ.)

Οικοσυστήματα βιομηχανικών υπηρεσιών: μελέτη του τρόπου διαχείρισης ενός οικοσυστήματος βιομηχανικών υπηρεσιών, εργαλεία για τη σύλληψη ιδεών και την καινοτομία, διαχείριση άυλων και υλικών τμημάτων και πλατφόρμα ΤΠΕ για την υποστήριξή τους.

Εικονικές βιομηχανικές επιχειρήσεις: πλαίσιο για την ενσωμάτωση βιομηχανικών υπηρεσιών και διαχείριση του κύκλου ζωής τους στις εικονικές βιομηχανικές επιχειρήσεις, μεθοδολογία ανάπτυξης υπηρεσιών, εργαλεία λογισμικού για τη μοντελοποίηση και ανάπτυξη νέων υπηρεσιών.

Μεγάλο μέρος των εργαλείων που παρήγαγε το MSEE πιστοποιήθηκαν μέσα από την εφαρμογή σε τέσσερεις βιομηχανικούς τομείς, σε ισάριθμες μελέτες περίπτωσης: βιομηχανικό εξοπλισμό, οικιακές συσκευές, οικιακά ηλεκτρονικά (τηλεοράσεις) και ρουχισμό. (Wiesner, Guglielmina, Gusmeroli, & Doumeingts, 2014).

5.2.2 Αρχιτεκτονική της Πλατφόρμας Υπηρεσιών MSEE

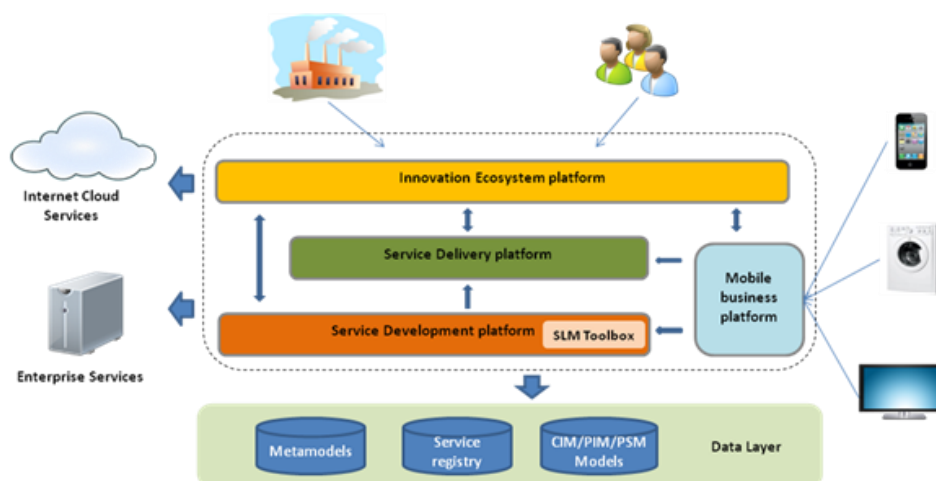
Στην έρευνα πάνω στη μηχανική υπηρεσιών έχει διατυπωθεί η άποψη ότι η απαιτούμενη ευελιξία στον σχεδιασμό, παροχή και διοίκηση υπηρεσιών μπορεί να επιτευχθεί μόνο με την εφαρμογή τεχνολογιών ΤΠΕ (Chen H. M., 2008; Böhmman, Leimeister, & Möslin, 2014; Cardoso, Voigt, & Winkler, Service engineering for the internet of services, 2009). Για τον λόγο αυτό, το έργο MSEE περιελάμβανε την διερεύνηση εργαλείων και τεχνολογιών πληροφορικής με σκοπό την υποστήριξη των βιομηχανικών υπηρεσιών σε όλο το κύκλο ζωής τους (σχεδιασμό, υλοποίηση, παροχή κ.α.). Ο απώτερος στόχος των εργαλείων αυτών είναι η διερεύνηση και η πρακτική εφαρμογή των θεωρητικών αποτελεσμάτων του έργου.

Στα πλαίσια της προσπάθειας αυτής, έγινε ο σχεδιασμός και η υλοποίηση πρωτότυπης πλατφόρμας ΤΠΕ, αποτελούμενη από εργαλεία και λειτουργικότητες για την ανάπτυξη και παροχή βιομηχανικών υπηρεσιών με τη χρήση Συστημάτων Προϊόντος – Υπηρεσίας. Τα εργαλεία αυτά βασίστηκαν σε καινοτομικές μεθοδολογίες και είχαν σαν σκοπό την αξιολόγηση των επιμέρους μεθοδολογιών, εφαρμογών αλλά και του συνολικού συστήματος ανάπτυξης και παροχής προηγμένων υπηρεσιών. Η πλατφόρμα είχε δύο ειδικούς στόχους: την υποστήριξη των βιομηχανικών χρηστών κατά την ενσωμάτωση υπηρεσιών σε βιομηχανικά προϊόντα, και την υποστήριξη της καινοτομίας σε οικοσυστήματα βιομηχανικών υπηρεσιών.

Ο πρώτος στόχος επιτυγχάνεται με την ενσωμάτωση εργαλείων για τη διοίκηση και τη διαχείριση του κύκλου ζωής των υπηρεσιών σε Συστήματα Προϊόντος - Υπηρεσίας. Έτσι, π.χ. περιλαμβάνεται λειτουργικότητα για την συνεργασία των εμπλεκόμενων σε ένα ΣΠΥ κατά τον προσδιορισμό του επιχειρηματικού μοντέλου, τον καθορισμό των δεικτών απόδοσης, τον σχεδιασμό των επιχειρησιακών διαδικασιών σε υψηλό επίπεδο, την παρακολούθηση των δεικτών αυτών, κτλ.

Ο δεύτερος στόχος επιτυγχάνεται με την ανάπτυξη μίας υποδομής για τον σχεδιασμό, την ανάπτυξη και εκτέλεση προηγμένων ψηφιακών υπηρεσιών στα Συστήματα Προϊόντος – Υπηρεσίας, στα οποία αναφέρεται ο πρώτος στόχος. Το σύστημα αυτό είναι σχεδιασμένο σύμφωνα με τις αρχές της υπηρεσιοστραφούς αρχιτεκτονικής. Οι εφαρμογές που υποστηρίζουν τις βιομηχανικές υπηρεσίες υλοποιούνται με τη χρήση «υπηρεσιών

λογισμικού» και ενορχηστρώσεών τους. Έτσι, το πληροφοριακό σύστημα του MSEE παίζει τον ρόλο ενός ενδιάμεσου λογισμικού υποδομής SOA (SOA Middleware), το οποίο παρέχει, με όσο πιο διάφανο τρόπο γίνεται, όλα τα απαραίτητα εργαλεία και υποδομές.



Σχήμα 17: Στοιχεία της πρωτότυπης πλατφόρμας ΤΠΕ του MSEE (MSEE, MSEE Service-System Functional and Modular Architecture, 2012)

Στο παραπάνω σχήμα παρουσιάζονται οι λειτουργικές μονάδες που συνθέτουν το πληροφοριακό σύστημα του MSEE.

Η **Πλατφόρμα Ανάπτυξης Υπηρεσιών (Service Development Platform)** περιλαμβάνει μία σειρά από εργαλεία για την ανάπτυξη βιομηχανικών υπηρεσιών. Περιλαμβάνει:

- Εργαλεία για τον σχεδιασμό και επιχειρησιακή μοντελοποίηση βιομηχανικών υπηρεσιών και τη διαχείριση κύκλου ζωής υπηρεσίας (Service Lifecycle Management Toolbox).
- Εργαλεία για την ανάπτυξη λογισμικού και εφαρμογών, με βάση την επιχειρησιακή μοντελοποίηση των βιομηχανικών υπηρεσιών. Τα εργαλεία αυτά αποσκοπούν πρωτίστως στη δημιουργία «υπηρεσιών λογισμικού» και ενορχηστρώσεών τους, σύμφωνα με τις αρχές της υπηρεσιοστραφούς αρχιτεκτονικής πληροφοριακών συστημάτων. Η Πλατφόρμα Ανάπτυξης Λογισμικού Υπηρεσιών που περιγράφεται στις προηγούμενες ενότητες εντάσσεται σε αυτή την κατηγορία.

Η **Πλατφόρμα Παροχής Υπηρεσιών (Service Delivery Platform)** διαχειρίζεται την εγγραφή και παροχή υπηρεσιών λογισμικού, στα πλαίσια της υπηρεσιοστραφούς αρχιτεκτονικής του συστήματος.

Η **Πλατφόρμα Κινητών Εφαρμογών (Mobile Business Platform)** επεκτείνει τη λειτουργικότητα των δύο προηγούμενων στο πεδίο των κινητών εφαρμογών, υποστηρίζοντας τις υπηρεσίες και ενορχηστρώσεις τους στο πεδίο των κινητών εφαρμογών.

Η **Πλατφόρμα Οικοσυστήματος Καινοτομίας (Innovation Ecosystem Platform)** αποτελεί το σημείο αλληλεπίδρασης των τελικών χρηστών, και διασυνδέεται με τα υπόλοιπα τμήματα του συστήματος του MSEE. Η Πλατφόρμα αυτή αναλαμβάνει την εκτέλεση των

ενορχηστρώσεων υπηρεσιών, χρησιμοποιώντας υπηρεσίες που έχουν εγγραφεί στη Πλατφόρμα Παροχής Υπηρεσιών. Οι χρήστες των ψηφιακών υπηρεσιών που υποστηρίζει η πλατφόρμα του MSEE (όπως βιομηχανικοί παραγωγοί, συνεργάτες και καταναλωτές) έχουν πρόσβαση στο λογισμικό των υπηρεσιών αυτών διαμέσου του λογισμικού IEP (Innovation Ecosystem Platform). Το IEP περιλαμβάνει μία αφοσιωμένη μηχανή εκτέλεσης επιχειρησιακών διαδικασιών (BPM Engine), η οποία βασίζεται στη πλατφόρμα Activiti BPMN⁵⁴. Το πακέτο Activiti μέσα στην IEP μπορεί να εκτελεί επιχειρησιακές διαδικασίες, παρέχει φόρμες για αλληλεπίδραση με τον χρήστη, εφαρμόζει επιχειρησιακούς κανόνες (business rules), καλεί υπηρεσίες εγγεγραμμένες στις υποδομές παροχής υπηρεσιών του MSEE αλλά και άλλα στοιχεία λογισμικού.

Στο **Επίπεδο Δεδομένων (Data Layer)** περιλαμβάνονται δεδομένα όπως μητρώα υπηρεσιών λογισμικού (service registry), μοντέλα βιομηχανικών υπηρεσιών σε διάφορα επίπεδα ανάπτυξης και τα σχετικά τους μεταμοντέλα.

Κάθε οικοσύστημα βιομηχανικών υπηρεσιών έχει διαφορετικούς εμπλεκόμενους, εφαρμογές και ανάγκες. Οι χρήστες του οικοσυστήματος χρησιμοποιούν τα εργαλεία του πληροφοριακού συστήματος του MSEE για να αναπτύξουν και να διαχειρίζονται τις βιομηχανικές υπηρεσίες που παρέχουν σε άλλους εμπλεκόμενους και στους τελικούς καταναλωτές. Έτσι, σε ένα δεδομένο οικοσύστημα βιομηχανικών υπηρεσιών θα υπάρχει μία μοναδική πλατφόρμα ΤΠΕ του MSEE, η οποία θα εξυπηρετεί τις αλληλεπιδράσεις των συμμετεχόντων εταιρειών και προσώπων.

5.2.3 Ανάπτυξη λογισμικού υπηρεσιών στα πλαίσια του MSEE

Η ανάπτυξη υπηρεσιών στα πλαίσια του MSEE βασίστηκε στην μεθοδολογία MDSEA, καθώς, όπως περιγράφεται στις προηγούμενες ενότητες, αποτελεί τη μοναδική, μέχρι στιγμής, μεθοδολογία που μπορεί να συνδυάσει την ανάπτυξη συστημάτων ΤΠΕ και λογισμικού με τον σχεδιασμό βιομηχανικών υπηρεσιών.

Έτσι, συστατικά τμήματα της Πλατφόρμας Ανάπτυξης Υπηρεσιών του MSEE στην υποστήριξη του κύκλου ζωής βιομηχανικών υπηρεσιών στα διάφορα στάδιά του, ακολουθώντας τη διαδικασία MDSEA. Σύμφωνα με τον αρχικό σχεδιασμό του συνολικού συστήματος ΤΠΕ του MSEE (MSEE, MSEE Service-System Functional and Modular Architecture, 2012), και όπως περιγράφεται στη προηγούμενη ενότητα, η «Πλατφόρμα Ανάπτυξης Υπηρεσιών» θα υποστηρίζει τα ακόλουθα στάδια του Κύκλου Ζωής Συστήματος Υπηρεσιών:

«1. Απαιτήσεις συστήματος βιομηχανικών υπηρεσιών: ταυτοποίηση, περιγραφή και μοντελοποίηση των απαιτήσεων των τελικών χρηστών για το σύστημα βιομηχανικών υπηρεσιών

2. Σχεδιασμός συστήματος βιομηχανικών υπηρεσιών: σχεδιασμός, εξειδίκευση και προσομοίωση του συστήματος που θα παρέχει τη βιομηχανική υπηρεσία

⁵⁴ <http://www.activiti.org/>

3. *Υλοποίηση συστήματος βιομηχανικών υπηρεσιών: περιγραφή του πως το υπό σχεδιασμό σύστημα βιομηχανικών υπηρεσιών θα υλοποιηθεί, ανάπτυξη όλων των απαραίτητων μηχανισμών και εξαρτημάτων*

4. *Λειτουργία συστήματος βιομηχανικών υπηρεσιών: το σύστημα υπηρεσιών είναι λειτουργικό και μπορεί να χρησιμοποιηθεί από τους πελάτες και όλους τους εμπλεκόμενους. Αυτό περιλαμβάνει την κατανάλωση των βιομηχανικών υπηρεσιών, την αλληλεπίδραση με τους πελάτες, την παρακολούθηση της λειτουργίας, την αξιολόγηση της απόδοσης και τη συντήρηση.*

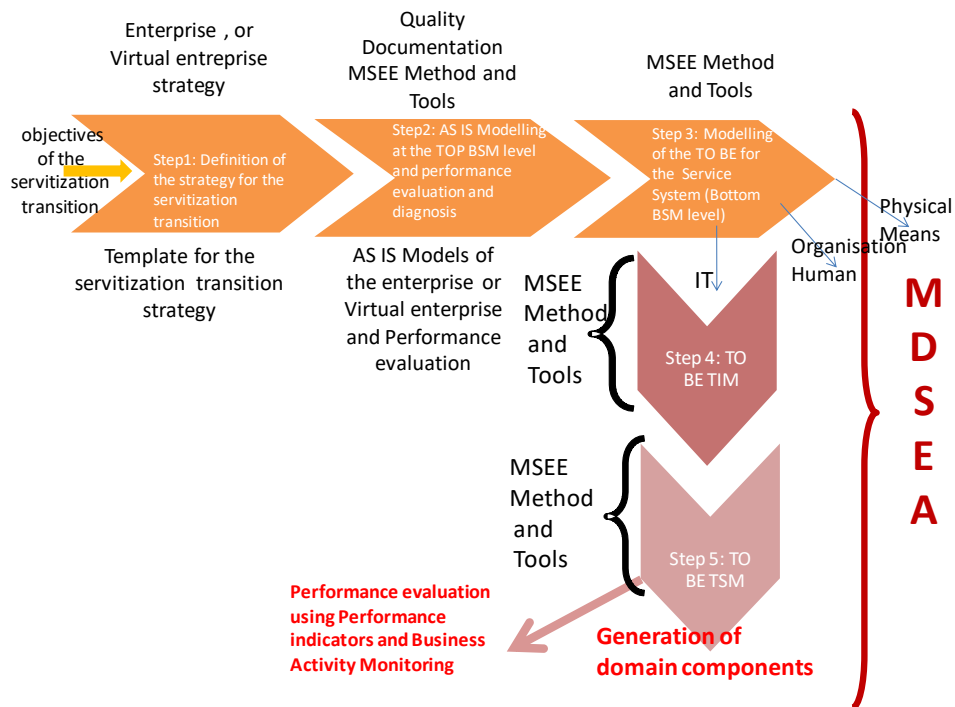
Τα στάδια 1 και 2 εξυπηρετούνται ευθέως από το πακέτο εργαλείων «Εργαλειοθήκη Διαχείρισης Κύκλου Ζωής Υπηρεσίας - Service Lifecycle Management Toolbox / SLM Toolbox (Bazoun, Zacharewicz, Ducq, & Boyé, 2014; Boyé & Bazoun, Service Life-cycle Management Tool Box, 2014) (MSEE, 2012), το οποίο επιτρέπει τη συνεργατική σχεδίαση ενός Συστήματος Προϊόντος – Υπηρεσίας μέσω της αλληλεπίδρασης πολλών εμπλεκόμενων. Το στάδιο 4 υποστηρίζεται έμμεσα, καθώς το SLM Toolbox χρησιμοποιείται για να καθορίσει στοιχεία των μηχανισμών διοίκησης της βιομηχανικής υπηρεσίας, όπως τον τρόπο παρακολούθησης και ελέγχου. Συγκεκριμένα, το SLM Toolbox παρέχει τις πρώτες φάσεις της ανάπτυξης υπηρεσιών (service engineering) όπως αυτά έχουν καθοριστεί στα πλαίσια του έργου MSEE. (MSEE, Service concepts, models and method: Model Driven Service Engineering, 2012).

Συγκεκριμένα, το SLM Toolbox, με βάση τον σχεδιασμό του είναι:

«Ένα ολοκληρωμένο εργαλείο μοντελοποίησης, αφιερωμένο στην διαχείριση του κύκλου ζωής βιομηχανικών υπηρεσιών. Θα επιτρέπει:

- *Την εκμετάλλευση μίας αρχιτεκτονικής βασισμένης σε μοντέλα: συντακτική επιβεβαίωση, μετασχηματισμό και εκτέλεση (syntactic validation, transformation, execution)*
- *Διατήρηση της συνέπειας και συνοχής σε όλη τη διαδικασία ανάπτυξης των βιομηχανικών υπηρεσιών: από τις επιχειρησιακές απαιτήσεις στην υλοποίηση του συστήματος ΤΠΕ (μοντελοποίηση)*
- *Πρόβλεψη και προσομοίωση του αποτελέσματος της υπηρεσίας (μηχανική – engineering)*
- *Σχεδιασμός της διοίκησης της υπηρεσίας (παρακολούθηση και έλεγχος)»*

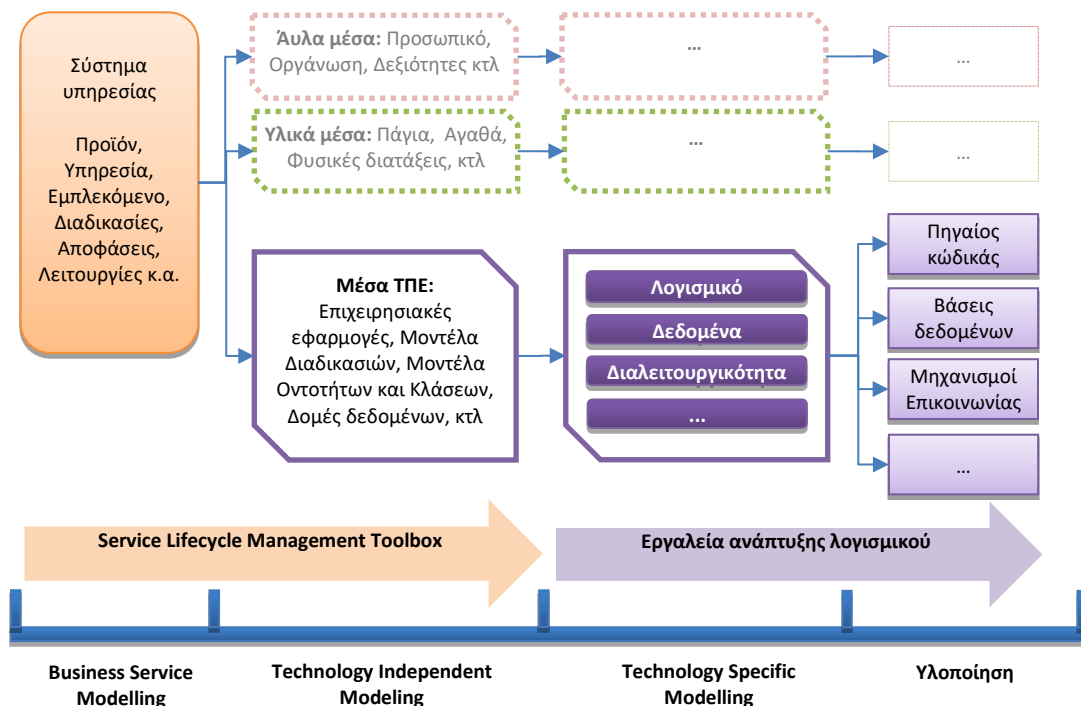
Από τις παραπάνω λειτουργίες γίνεται σαφές ότι το SLM Toolbox δεν παρέχει υποστήριξη για την υλοποίηση των μηχανισμών παροχής της υπηρεσίας, δηλαδή το στάδιο 3. Άρα, οι λειτουργίες αυτές, δηλαδή η ανάπτυξη του προϊόντος, των επιχειρησιακών διαδικασιών, ανθρώπινου δυναμικού και των υποδομών ΤΠΕ και λογισμικού πρέπει να υποστηριχθούν από άλλα εργαλεία της Πλατφόρμας Ανάπτυξης Υπηρεσιών του MSEE. (MSEE, 2012)



Σχήμα 18: Διαχωρισμός της ανάπτυξης των υποδομών ΤΠΕ στη μεθοδολογία MDSEA κατά τη διαδικασία της ενσωμάτωσης υπηρεσιών (MSEE, Service concepts, models and method: Model Driven Service Engineering, 2012)

Τα αρχικά στάδια της διαδικασίας ανάπτυξης, όπως αυτά διακρίνονται στο παραπάνω σχήμα, εξυπηρετούνται από το SLM Toolbox, και καταλήγουν στον ορισμό του μοντέλου επιπέδου Business-Service Model (BSM). Όπως περιγράφηκε στις προηγούμενες ενότητες, η μεθοδολογία συνεχίζει να εφαρμόζεται και στα επόμενα στάδια για την ανάπτυξη των υποδομών ΤΠΕ, όπως και για τις υπόλοιπες συνιστώσες του συστήματος υπηρεσίας.

Οι διαδικασίες ανάπτυξης των συνιστωσών του ΣΠΥ διαχωρίζονται κατά την εφαρμογή της μεθοδολογίας MDSEA, αμέσως μετά το στάδιο BSM. Έτσι, στα πλαίσια του SLM Toolbox, αναπτύσσονται τα μοντέλα TIM για το τμήμα ΤΠΕ της υπηρεσίας. Στη συνέχεια, αυτά παραδίδονται σε κατάλληλα εργαλεία της Πλατφόρμας Ανάπτυξης Υπηρεσιών, τα οποία αναλαμβάνουν τον μετασχηματισμό τους σε μοντέλα επιπέδου TSM και, στη συνέχεια, σε εκτελέσιμα στοιχεία όπως κώδικας, κ.α. όπως φαίνεται στο ακόλουθο σχήμα:



Σχήμα 19: Παράλληλες ροές ανάπτυξης των υποδομών του συστήματος υπηρεσιών, στα πλαίσια της μεθοδολογίας MDSEA

Η διαδικασία ανάπτυξης λογισμικού στο MSEE συνάδει με τις αρχές της μεθοδολογίας MDSEA, και την επεκτείνει κατάλληλα ώστε να οδηγεί στην παραγωγή εκτελέσιμου λογισμικού και υπηρεσιών.

Έτσι, η ροή εργασιών θα πρέπει να ξεκινά με μία αρχική σύνθεση υψηλού επιπέδου των απαραίτητων στοιχείων και των βημάτων για την παροχή της βιομηχανικής υπηρεσίας, τα οποία θα αντιστοιχούν στο μοντέλο επιπέδου TIM (Technology Independent Model). Το μοντέλο TIM θα αντιστοιχίζεται με ένα μοντέλο TSM, με βάση ένα «Μοντέλο Τεχνολογίας» (Technology Model – TM). Το Μοντέλο Τεχνολογίας θα περιέχει τους απαραίτητους κανόνες που διέπουν την αντιστοίχιση των στοιχείων του TIM σε στοιχεία TSM, τα οποία διέπονται από συγκεκριμένες τεχνικές παραμέτρους, ονοματολογία και ιδιότητες. Υπό κάποιες προϋποθέσεις (και ειδικότερα αν το επιτρέπει η συγκεκριμένη τεχνολογία και το περιβάλλον ανάπτυξης) το παραγόμενο μοντέλο TSM θα μπορούσε να είναι εκτελέσιμο ή, τουλάχιστον, θα επέτρεπε προσομοίωση και εντοπισμό σφαλμάτων. Στη συνέχεια, οι προγραμματιστές θα επεξεργάζονται το μοντέλο TSM μέσα στο περιβάλλον ανάπτυξης, και θα παράγουν εκτελέσιμο λογισμικό, το οποίο, με τη σειρά του, θα είναι έτοιμο προς εγκατάσταση και χρήση στις επιθυμητές τεχνολογικές υποδομές (όπως αυτές έχουν καθοριστεί από το Μοντέλο Τεχνολογίας για τον μετασχηματισμό TIM προς TSM).

Οι υπηρεσίες και οι εφαρμογές που υλοποιούνται με τον τρόπο αυτό μπορούν να υλοποιούνται εξ' ολοκλήρου με τα εργαλεία ανάπτυξης λογισμικού της πλατφόρμας του MSEE. Εναλλακτικά, μπορούν να επαναχρησιμοποιούν ήδη υφιστάμενα προϊόντα και λειτουργίες, διαθέσιμα στους χρήστες του συστήματος του MSEE. Αυτά μπορεί να είναι ήδη

υλοποιημένες εφαρμογές και υπηρεσίες λογισμικού, όπως π.χ. υπηρεσίες που έχουν εγγραφεί στη Πλατφόρμα Παροχής Υπηρεσιών του MSEE (Service Delivery Platform).

Επιπλέον, τα εργαλεία ανάπτυξης θα πρέπει να επιτρέπουν στους χρήστες:

- την εύκολη προσαρμογή των σχεδίων υπηρεσιών και εφαρμογών στις επιθυμητές τεχνολογικές πλατφόρμες – στόχους,
- την ανάπτυξη και σχεδιασμό στο επίπεδο μοντέλων και σε επίπεδο κώδικα
- την αξιοποίηση αυτοματισμών, όπως μετασχηματισμούς μοντέλων και παραγωγή πηγαίου κώδικα

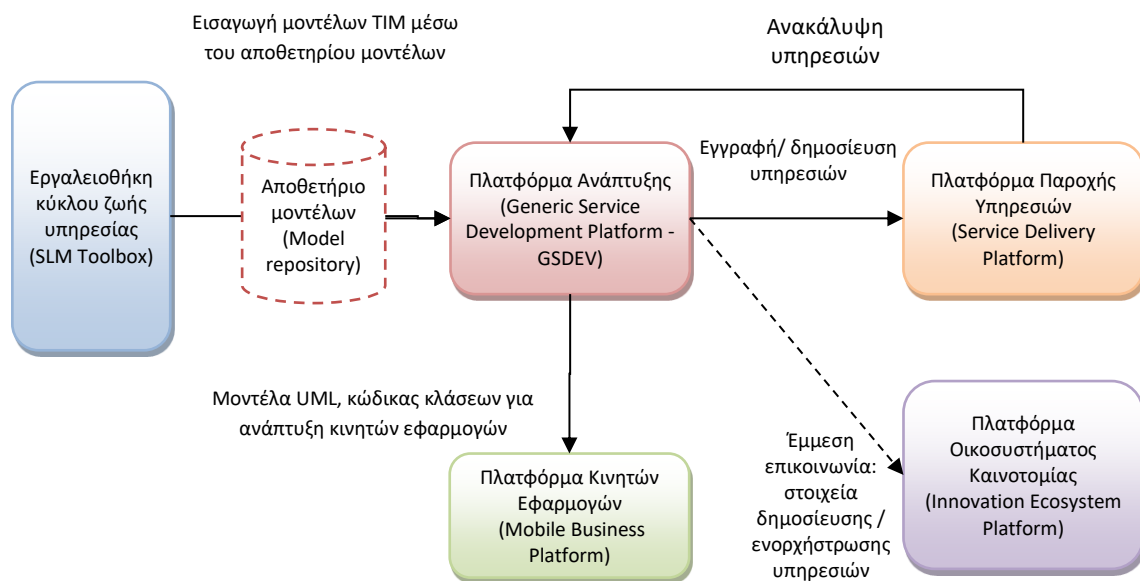
5.2.4 Πλατφόρμα Ανάπτυξης Υπηρεσιών Λογισμικού

Για την κάλυψη της παραπάνω λειτουργικότητας, απαιτείται η ανάπτυξη ενιαίου περιβάλλοντος ανάπτυξης λογισμικού, ενσωματωμένου στη πλατφόρμα ΤΠΕ του MSEE, δηλαδή μίας Πλατφόρμας Ανάπτυξης βασισμένης στη μεθοδολογία και αρχιτεκτονική που παρουσιάστηκε στις προηγούμενες ενότητες.

Για την υποδομή ΤΠΕ των συστημάτων υπηρεσιών του MSEE, η Πλατφόρμα Ανάπτυξης (με την επωνυμία Generic Service Development Platform – GSDEV) είναι ένα θεμελιώδες τμήμα. Μαζί με την Εργαλειοθήκη Διαχείρισης Κύκλου Ζωής Υπηρεσίας (SLM Toolbox) (MSEE, 2012) παρέχει τη δυνατότητα στους χρήστες της πλατφόρμας του MSEE για:

«...την ανάπτυξη περίπλοκων μετωπικών (front-end) εφαρμογών, συμπεριλαμβανομένης της εκτέλεσης σχετικά πολύπλοκων επιχειρησιακών διαδικασιών, οι οποίες βασίζονται σε υπηρεσίες υποβάθρου(back – end) με επιχειρησιακούς χρήστες. Η Πλατφόρμα Ανάπτυξης θα επιτρέπει τον σχεδιασμό των διάφορων διεπαφών με τις οποίες θα αλληλεπιδρούν οι τελικοί χρήστες. Έτσι, οι χρήστες της Πλατφόρμας θα προσδιορίζουν τις σχέσεις μεταξύ των τελικών χρηστών και τους πόρους/ υπηρεσίες που θα παρέχει το υπόβαθρο (back – end).» (MSEE, MSEE Service-System Functional and Modular Architecture, 2012)

Με βάση το παραπάνω, η Πλατφόρμα Ανάπτυξης αποτελεί ένα περιβάλλον υλοποίησης εφαρμογών ,δηλαδή υπηρεσιών στα πλαίσια μίας «υπηρεσιοστραφούς αρχιτεκτονικής» (Service- Oriented Architecture), υλοποιούμενης από τη συνολική πλατφόρμα του MSEE. Υποστηρίζει την εφαρμογή της μεθοδολογίας MDSEA για τα τμήματα ΤΠΕ του ΣΠΥ. Το περιβάλλον λειτουργίας στο MSEE αντανακλά το θεωρητικό περιβάλλον που παρουσιάστηκε στις προηγούμενες ενότητες. Έτσι, η Πλατφόρμα διασυνδέεται με το υπόλοιπο σύστημα ΤΠΕ του MSEE όπως στο Σχήμα 20.



Σχήμα 20: Διασύνδεση της Πλατφόρμας Ανάπτυξης με το υπόλοιπο σύστημα

Το πακέτο εργαλείων SLM Toolbox (Bazoun, Zacharewicz, Ducq, & Boyé, 2014; Boyé & Bazoun, Service Life-cycle Management Tool Box, 2014) χρησιμοποιείται για τη δημιουργία μοντέλων υψηλού επιπέδου (TIM – Technology Independent Models) των υπηρεσιών, τα οποία στη συνέχεια τοποθετούνται σε ένα κοινό αποθετήριο μοντέλων (το οποίο έχει υλοποιηθεί σαν μέρος της Πλατφόρμας GSDEV).

Τα μοντέλα αυτά εισάγονται στο ενιαίο περιβάλλον της Πλατφόρμας Ανάπτυξης, και με βάση τα μοντέλα αυτά, οι προγραμματιστές μπορούν να αναπτύξουν εκτελέσιμες υπηρεσίες. Οι υπηρεσίες αυτές στη συνέχεια εγγράφονται στη Πλατφόρμα Παροχής Υπηρεσιών (Toma, García, Larizgoitia, & Fensel, 2014), ώστε να μπορούν να αναζητηθούν και να χρησιμοποιηθούν στο δεδομένο σύστημα υπηρεσιών.

Επιπλέον, κατά την ανάπτυξη λογισμικού, οι χρήστες της Πλατφόρμας Ανάπτυξης θα μπορούν να αναζητήσουν και να προσθέτουν αναφορές προς υπηρεσίες ήδη καταχωρημένες στη Πλατφόρμα Παροχής Υπηρεσιών του MSEE. Τέλος, η Πλατφόρμα ανάπτυξης θα πρέπει να υποστηρίζει την Πλατφόρμα Κινητών Εφαρμογών, προσφέροντας μοντέλα UML και για την παραγωγή κώδικα για το συγκεκριμένο πεδίο.

Μία περαιτέρω έμμεση διασύνδεση υπάρχει με τη «Πλατφόρμα Οικοσυστήματος Καινοτομίας» (Innovation Ecosystem Platform – IEP), η οποία εκτελεί τις επιχειρησιακές διαδικασίες και παρέχει τις ψηφιακές υπηρεσίες στους τελικούς χρήστες, χωρίς όμως άμεση διαλειτουργικότητα μεταξύ των δύο συστημάτων. Σαν μέρος της κανονικής ροής λειτουργίας, η Πλατφόρμα Ανάπτυξης παράγει προϊόντα σχετικά με την εκτέλεση και ενορχήστρωση υπηρεσιών, όπως εκτελέσιμα αρχείου τύπου .bar (Business Archives) τα οποία μπορούν να φορτωθούν και να εκτελεστούν από τη «Πλατφόρμα Οικοσυστήματος Καινοτομίας».

5.3 Πρωτότυπο

Με βάση αυτές τις προδιαγραφές, το πρωτότυπο της Πλατφόρμας Ανάπτυξης αναπτύχθηκε ως ένα κατάλληλα προσαρμοσμένο Ολοκληρωμένο Περιβάλλον Ανάπτυξης, βασισμένο στη πλατφόρμα ανάπτυξης λογισμικού Eclipse⁵⁵. Το Eclipse παρέχει ένα πλήρες Ολοκληρωμένο Περιβάλλον Ανάπτυξης Λογισμικού, το οποίο μπορεί να τροποποιηθεί με τη χρήση «πρόσθετων» (plugins). Χρησιμοποιώντας αυτή την «αρθρωτή» (modular) αρχιτεκτονική, είναι εφικτή η προσαρμογή του σε διάφορες ροές εργασίας και πεδία εφαρμογής, χρησιμοποιώντας υφιστάμενα ή νέα πρόσθετα (off-the-shelf, custom-built).

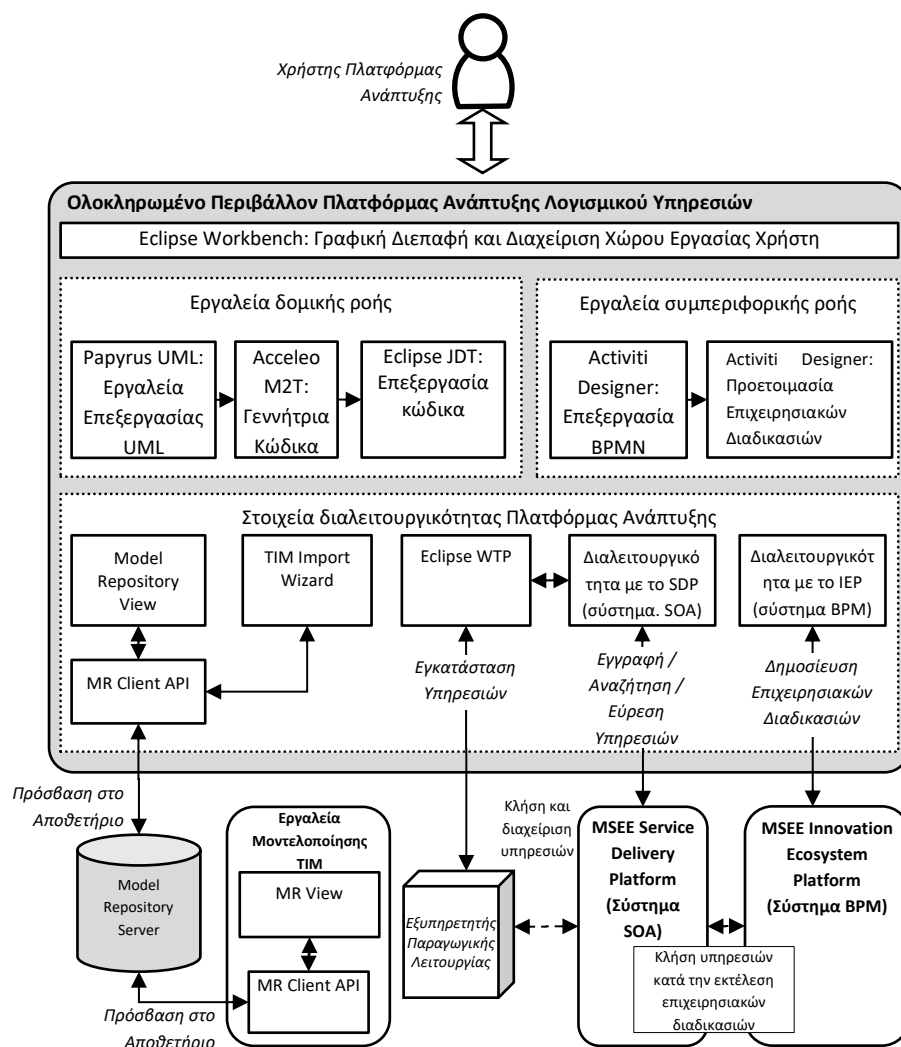
Οι λειτουργίες της ΠΑ υλοποιούνται με τον συνδυασμό των λειτουργιών της πλατφόρμας Eclipse (π.χ. εργαλεία ανάπτυξης Java, γραφική διεπαφή χρήστη, διαχείριση χώρου εργασίας κτλ.) και ενός συνόλου εξειδικευμένων plugins. Η αρθρωτή αρχιτεκτονική του Eclipse επιτρέπει την προσαρμογή της ΠΑ για συγκεκριμένες εφαρμογές και πεδία, με την προσθήκη επιπλέον πρόσθετων και άλλων τροποποιήσεων. Έτσι, η ΠΑ δεν περιορίζεται αυστηρά στο πλαίσιο της MDSEA, και μπορεί να υποστηρίξει άλλες συμβατικές ή εξειδικευμένες ροές ανάπτυξης λογισμικού, αυξάνοντας την ωφελιμότητά της για τον τελικό χρήστη.

Η αρχιτεκτονική της ΠΑ βασίζεται σε αυτή την προσέγγιση ώστε να παρέχει όλη την απαιτούμενη λειτουργικότητα ροών εργασίας και διαλειτουργικότητας. Η ΠΑ υποστηρίζεται από ένα αποθετήριο πόρων, στο οποίο έχουν πρόσβαση οι χρήστες της και οι χρήστες του SLM Toolbox.

Από την άποψη της υλοποίησης, η πρωτότυπη εφαρμογή της Πλατφόρμας Ανάπτυξης είναι ένα Ολοκληρωμένο Περιβάλλον Ανάπτυξης που βασίζεται στο Eclipse Juno SR2 (v4.2.2). Στο βασικό πλαίσιο που παρέχει το Eclipse προστέθηκε ένας αριθμός από πρόσθετα που σχεδιάστηκαν, τροποποιήθηκαν και παραμετροποιήθηκαν ώστε συνολικά να επιτρέπουν τις προβλεπόμενες ροές εργασίας. Η πρωτότυπη Πλατφόρμα Ανάπτυξης εγκαθίσταται και εκτελείται τοπικά στους υπολογιστές των χρηστών. Οι χρήστες της αναμένονται να είναι αναλυτές και μηχανικοί λογισμικού. Μπορεί να χρησιμοποιηθεί σε οποιοδήποτε λειτουργικό σύστημα με το οποίο είναι συμβατό το Eclipse, δηλαδή Microsoft Windows, MacOS X και Linux. Έχει δοκιμαστεί με επιτυχία σε MS Windows και MacOS X.

Στο Σχήμα 21 παρουσιάζεται συνοπτικά η αρχιτεκτονική του πρωτότυπου, καθώς και τα εργαλεία και τεχνολογίες που χρησιμοποιήθηκαν για την παροχή της απαιτούμενης λειτουργικότητας.

⁵⁵www.eclipse.org



Σχήμα 21: Αρχιτεκτονική και τεχνολογίες πρωτότυπης Πλατφόρμας Ανάπτυξης

Το Ολοκληρωμένο Περιβάλλον Ανάπτυξης Λογισμικού της ΠΑ αποτελείται από τα ακόλουθα στοιχεία και λειτουργίες (Σημειώνεται ότι το Eclipse περιλαμβάνει πολύ περισσότερες λειτουργίες, ωστόσο συμπεριλαμβάνονται μόνο αυτές που είναι άμεσα σχετικές με τη διαδικασία ανάπτυξης του MSEE):

- Το γραφικό περιβάλλον ανάπτυξης και ο χώρος εργασίας χρήστη του Eclipse (Eclipse Workbench)
- Τα πρόσθετα Κεντρικών Λειτουργιών της Πλατφόρμας Ανάπτυξης (Core Plugins). Αυτά περιλαμβάνουν:
 - Έναν περιβάλλον επεξεργασίας UML (Papyrus UML⁵⁶)
 - Ένα περιβάλλον επεξεργασίας BPMN, και προετοιμασίας επιχειρησιακών ροών (Activiti Designer⁵⁷ του πλαισίου Activiti Business Process Management)

⁵⁶ <http://www.papyrusuml.org>

⁵⁷ <http://www.activiti.org/>

- Μία γεννήτρια κώδικα (βασισμένη στο πρόσθετο Acceleo για το Eclipse and τη γλώσσα MOF Model-to-Text (MTL)⁵⁸, μαζί με προσαρμοσμένους κανόνες μετασχηματισμού)
- Τα εργαλεία ανάπτυξης Java του Eclipse (Java Development Tools - JDT⁵⁹)
- Τα Στοιχεία Διαλειτουργικότητας:
 - Λειτουργίες Πρόσβασης στο Αποθετήριο Μοντέλων (Model Repository Access Stack):
 - Προβολή Αποθετηρίου Μοντέλων (Model Repository View plugin), το οποίο παρέχει μία γραφική διεπαφή για πρόσβαση στο αποθετήριο μοντέλων.
 - API Πελάτη για το Αποθετήριο Μοντέλων (Model Repository Client API plugin), το οποίο παρέχει απευθείας προγραμματιστική πρόσβαση στο αποθετήριο μοντέλων
 - Ο Οδηγός Εισαγωγής TIM (TIM import wizard), ένα πρόσθετο για την διευκόλυνση της διαδικασίας εισαγωγή μοντέλων TIM από το αποθετήριο μοντέλων
 - Τα εργαλεία της πλατφόρμας Eclipse Web Tools Platform (WTP⁶⁰), η οποία συμπεριλαμβάνει εργαλεία για την ανάπτυξη διαδικτυακών υπηρεσιών (JST Web Services tools⁶¹).
 - Διαλειτουργικότητα με την Πλατφόρμα Παροχής Υπηρεσιών web services (Service Delivery Platform Integration - SDP Integration) που περιλαμβάνει:
 - Τον οδηγό ανακάλυψης υπηρεσιών (Service Discovery wizard)
 - Τον οδηγό εγγραφής υπηρεσιών (Service Registration wizard)
 - Διαλειτουργικότητα με την Πλατφόρμα Οικοσυστήματος Καινοτομίας (Innovation Ecosystem Platform Integration - IEP Integration) για την εγκατάσταση (deployment) εκτελέσιμων Επιχειρησιακών Διαδικασιών στο IEP.

Ο Εξυπηρετητής Αποθετηρίου Μοντέλων (Model Repository Server) είναι ένα διακριτό στοιχείο που διαλειτουργεί στενά με την Πλατφόρμα Ανάπτυξης. Στον εξυπηρετητή έχουν κοινή πρόσβαση οι χρήστες της Πλατφόρμας του MSEE (της Πλατφόρμας Ανάπτυξης και του SLM Toolbox), με σκοπό την αποθήκευση και ανάγνωση μοντέλων και άλλων προϊόντων της διαδικασίας ανάπτυξης. Για τους χρήστες αυτούς, το Αποθετήριο Μοντέλων (AM) είναι ένα κοινός χώρος εργασίας, ο οποίος επιτρέπει στους χρήστες του SLM Toolbox να στέλνουν μοντέλα TIM στους χρήστες της Πλατφόρμας Ανάπτυξης για περαιτέρω επεξεργασία. Σημειώνεται ότι, για τον σκοπό αυτό, τα πρόσθετα Πρόσβασης στο Αποθετήριο Μοντέλων (MR View και MR Client API) περιλαμβάνονται και στο περιβάλλον του SLM Toolbox.

Τα συστατικά της αρχιτεκτονικής της ΠΑ περιγράφονται αναλυτικά στις επόμενες ενότητες.

⁵⁸ <http://www.eclipse.org/acceleo/>

⁵⁹ <https://eclipse.org/jdt/>

⁶⁰ <https://eclipse.org/webtools/>

⁶¹ <https://eclipse.org/webtools/ws/>

5.3.1 Περιβάλλον Eclipse

5.3.1.1 Ολοκληρωμένο περιβάλλον ανάπτυξης (Eclipse IDE and Workbench)

Το Eclipse Workbench είναι το περιβάλλον ανάπτυξης λογισμικού της πλατφόρμας Eclipse και αποτελεί τη βάση της γραφικής διεπαφής χρήστη. Διαχειρίζεται τον χώρο εργασίας του χρήστη, επιτρέποντας την ταυτόχρονη διαχείριση πολλαπλών έργων, φακέλων και στοιχείων όπως κώδικας, κείμενο, και μοντέλα. Προσφέρει πρόσβαση στα πρόσθετα και τη λειτουργικότητα της πλατφόρμας Eclipse, συμπεριλαμβανομένων των πρόσθετων που συνθέτουν την Πλατφόρμα Ανάπτυξης. Επιπλέον, το Workbench δίνει πρόσβαση σε αρχεία βοήθειας, συνεργατικές λειτουργίες και λειτουργίες εντοπισμού σφαλμάτων.

5.3.1.2 Υποστηρικτικά πρόσθετα ανάπτυξης λογισμικού του Eclipse

5.3.1.2.1 Εργαλεία ανάπτυξης JAVA (Eclipse Java Development Tools -JDT)

Το JDT Project παρέχει εργαλεία με τη μορφή πρόσθετων του Eclipse, τα οποία συνθέτουν ένα περιβάλλον ανάπτυξης Java για οποιαδήποτε εφαρμογή (συμπεριλαμβανομένων και άλλων πρόσθετων). Προσθέτει την έννοια του «έργου Java» (Java Project) και σχετικούς τρόπους οργάνωσης του γραφικού περιβάλλοντος χρήστη (Java Perspective) στο Eclipse Workbench. Σε αυτό περιλαμβάνονται απεικονίσεις, οδηγοί, και εργαλεία διαχείρισης και αναδόμησης (refactoring) κώδικα. Το JDT επιτρέπει στο πλαίσιο Eclipse να είναι ένα αυτόνομο περιβάλλον ανάπτυξης για κάθε είδους εφαρμογής Java.

5.3.1.2.2 Περιβάλλον ανάπτυξης πρόσθετων (Plug-in Development Environment - PDE)

Το περιβάλλον ανάπτυξης πρόσθετων του Eclipse (Plug-in Development Environment - PDE) προσφέρει εργαλεία για τη δημιουργία, ανάπτυξη, δοκιμή, εντοπισμό σφαλμάτων, μεταγλώττιση και εγκατάσταση πρόσθετων του Eclipse, λειτουργίες, ιστοσελίδες ενημέρωσης και προϊόντα Rich Client Platform (RCP). Το PDE υποστηρίζει τις πρότυπες προδιαγραφές OSGi (Open Service Gateway initiative), για τον προγραμματισμό αρθρωτών εφαρμογών. Η αρθρωτή αρχιτεκτονική του Eclipse και η προσθήκη του PDE επιτρέπει στους τελικούς χρήστες για την ανάπτυξη εξειδικευμένων πρόσθετων για συγκεκριμένα βιομηχανικά πεδία εφαρμογής και ροές εργασίας. Με τον τρόπο αυτό, η Πλατφόρμα Ανάπτυξης μπορεί να επεκταθεί από τους χρήστες της και να εξελίσσεται σύμφωνα με τις ανάγκες τους.

5.3.2 Κύριες Λειτουργίες της Πλατφόρμας ανάπτυξης

Τα στοιχεία των κύριων λειτουργιών υποστηρίζουν τη βασική λειτουργικότητα της ΠΑ, δηλαδή την επεξεργασία, τη χαρτογραφική αντιστοίχιση (mapping), τον εμπλουτισμό μοντέλων, καθώς και την αυτόματη παραγωγή κώδικα, την επεξεργασία του και την παραγωγή εκτελέσιμων προϊόντων.

Οι λειτουργίες αυτές οργανώνονται σύμφωνα με τις δύο βασικές ροές εργασίας που υποστηρίζει η ΠΑ: Δομική Ροή (Διαγράμματα κλάσεων UML προς Java) και Συμπεριφορική Ροή (BPMN προς Εκτελέσιμες Επιχειρησιακές Διαδικασίες).

Τα βήματα των διαδικασιών αυτών υλοποιούνται από το αντίστοιχο λογισμικό και τεχνολογίες. Η αντιστοίχιση αυτή παρουσιάζεται συνοπτικά στον ακόλουθο πίνακα:

Δραστηριότητα	Εργαλείο	Τεχνολογικό υπόβαθρο
---------------	----------	----------------------

Δομική Ροή: Διαγράμματα κλάσεων UML προς Java		
Επεξεργασία και Εμπλουτισμός Μοντέλου (Κλάσεις UML)	Εργαλεία επεξεργασίας UML	Parvus UML (Lanusse, et al., 2009), και εμπλουτισμός μοντέλου με την εφαρμογή UML Java profile εντός του εργαλείου.
Αυτόματη παραγωγή κώδικα (TSM-to-code)	Γεννήτρια κώδικα	Πρόσθετο Acceleo (MOF Model-to-text Language) (Eclipse Foundation, 2006) με εξειδικευμένους κανόνες μετάσχηματισμού UML προς Javabeans (βλ. παράρτημα)
Επεξεργασία κώδικα	Εργαλεία επεξεργασίας κώδικα	Eclipse Java Development Toolkit
<i>Συμπληρωματική λειτουργία: Αυτόματη παραγωγή κώδικα πελάτη webservice για προγραμματική σύνθεση υπηρεσιών</i>	<i>Εργαλεία webservises και διαλειτουργικότητα με την υποδομή παροχής υπηρεσιών του MSEE</i>	<i>Eclipse Webtools (Eclipse WTP) (Dai, Mandel, & Ryman, 2007) – μηχανισμοί διαλειτουργικότητας ΠΑ</i>
BPMN-προς-Εκτ. Επιχειρησιακές Διαδικασίες		
Επεξεργασία και Εμπλουτισμός μοντέλου (BPMN)	Εργαλεία επεξεργασίας BPMN	Activiti Designer (Rademakers, 2012) με εξειδικευμένες τροποποιήσεις
Προετοιμασία Επιχειρησιακών Διαδικασιών για Εγκατάσταση	Εργαλεία επεξεργασίας BPMN	Activiti Designer με εξειδικευμένες τροποποιήσεις
<i>Συμπληρωματική λειτουργία: Εισαγωγή αναφορών σε webservice για τη δημιουργία συνθέσεων υπηρεσιών</i>	<i>Επεξεργαστής BPMN και διαλειτουργικότητα με την υποδομή παροχής υπηρεσιών του MSEE</i>	<i>Activiti Designer με εξειδικευμένες τροποποιήσεις</i>
Στοιχεία Διαλειτουργικότητας Πλατφόρμας Ανάπτυξης		
Πλήρης πρόσβαση στο Αποθετήριο Μοντέλων	Γραφική Διεπαφή Αποθετηρίου Μοντέλων	Model Repository View: Πρωτότυπο πρόσθετο διεπαφής χρήστη
Υποστήριξη πρόσβασης στο Αποθετήριο Μοντέλων από τα στοιχεία της Πλατφόρμας Ανάπτυξης	API Πελάτη Αποθετηρίου	Model Repository Client API: Πρωτότυπο πρόσθετο διαλειτουργικότητας
Πρόσβαση στο Αποθετήριο μοντέλων για την εκτέλεση εξειδικευμένων λειτουργιών	Οδηγοί διαλειτουργικότητας Αποθετηρίου Μοντέλων	TIM Import Wizard: Πρωτότυπο πρόσθετο διεπαφής χρήστη
Εγκατάσταση web services σε εξυπηρετητές παραγωγικής λειτουργίας	Εργαλείο διαχείρισης webservises	Eclipse Webtools (Eclipse WTP)

Εγγραφή / Αναζήτηση / Εύρεση Υπηρεσιών στην υποδομή SOA	Διαλειτουργικότητα με το Service Delivery Platform	Service Delivery Platform Integration Plugin: Πρωτότυπο πρόσθετο διεπαφής χρήστη και διαλειτουργικότητας
Εγκατάσταση εκτελέσιμων επιχειρησιακών διαδικασιών	Πρόσθετο Business Process Deployment	Πρωτότυπο πρόσθετο διαλειτουργικότητας

Πίνακας 9: Κυρίες λειτουργίες της Πλατφόρμας Ανάπτυξης, Τεχνολογίες και Μέθοδοι Υλοποίησης

5.3.2.1 Ανάπτυξη λογισμικού στη Δομική Ροή εργασίας: Κλάσεις UML προς Java

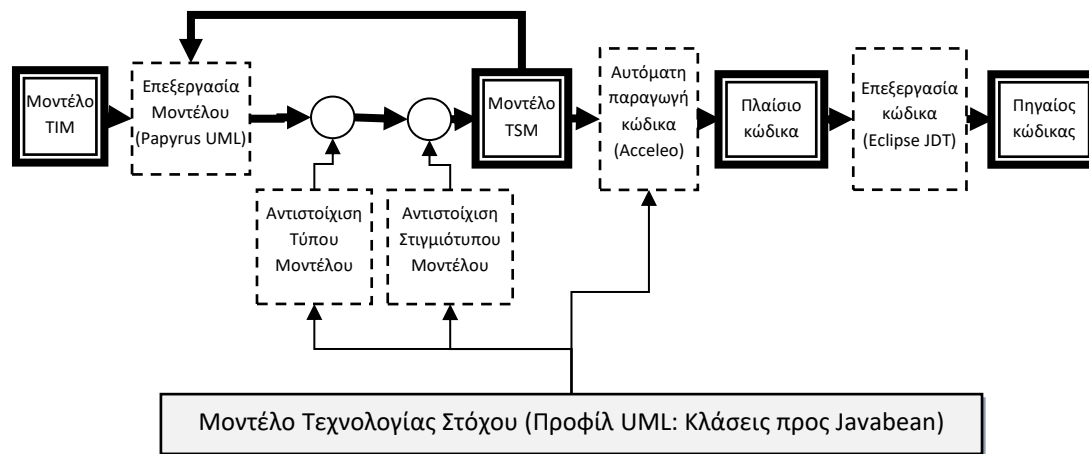
Στη ροή αυτή, μοντέλα κλάσεων UML μετασχηματίζονται σε κλάσεις Java (JavaBeans). Η ροή περιλαμβάνει την επεξεργασία και εμπλουτισμό του μοντέλου TIM (σε UML), την αυτόματη παραγωγή templates κώδικα και την περαιτέρω επεξεργασία του. Για τον σκοπό αυτό, η Πλατφόρμα Ανάπτυξης χρησιμοποιεί ένα περιβάλλον επεξεργασίας UML, μία γεννήτρια κώδικα και εργαλεία ανάπτυξης Java.

Η διαδικασία αυτή περιλαμβάνει τα ακόλουθα βήματα:

- Ο χρήστης εισάγει ένα μοντέλο TIM από το Αποθετήριο Μοντέλων, χρησιμοποιώντας τον αντίστοιχο οδηγό ή γραφική διεπαφή.
- Ο χρήστης επεξεργάζεται το μοντέλο TIM στον γραφικό επεξεργαστή UML (Papyrus UML)
- Ο χρήστης εισάγει ένα προφίλ UML το οποίο εισάγει στοιχεία της Java στο περιβάλλον μοντελοποίησης, με τα οποία μπορεί να «σημειώσει» τα αντίστοιχα στοιχεία κατά τη διαδικασία εμπλουτισμού.
- Ο χρήστης αιτείται την εφαρμογή του προφίλ UML στο μοντέλο. Με την εφαρμογή του προφίλ, ο χρήστης μπορεί να προσθέσει «επισημάνσεις» που αφορούν στη Java.
 - Ο χρήστης χρησιμοποιεί τη γραφική διεπαφή του επεξεργαστή UML για να προσθέσει επισημάνσεις Java π.χ. οι κλάσεις της UML σημειώνονται ως JavaBeans και οι ιδιότητες των κλάσεων UML αντιστοιχίζονται με τύπους δεδομένων πεδίων Java (float, integer, date etc)
- Το μοντέλο UML, μαζί με τις επιπλέον πληροφορίες τεχνολογικής πλατφόρμας αντιστοιχεί σε ένα μοντέλο TSM.
- Με την ολοκλήρωση της διαδικασίας εμπλουτισμού, ο χρήστης ζητά την παραγωγή κώδικα.
- Η ΠΑ παράγει ένα σκελετό κώδικα, βασισμένο στο εμπλουτισμένο μοντέλο (το μοντέλο TSM). Ο σκελετός περιλαμβάνει κλάσεις Java (JavaBeans), μαζί με τις ιδιότητες, τις μεθόδους τους και άλλα στοιχεία.
- Ο χρήστης συμπληρώνει τον σκελετό (συμπληρώνει τις μεθόδους, προσθέτει μεταβλητές κτλ.), δοκιμάζει και αποσφαλματώνει τον κώδικα, χρησιμοποιώντας τυπικές μεθόδους ανάπτυξης λογισμικού στο περιβάλλον του Eclipse (Eclipse Java Development Toolkit).

Η συγκεκριμένη ροή εργασίας μπορεί να δημιουργεί προϊόντα πηγαίου κώδικα (π.χ. κλάσεις κ.α.), εφαρμογές, ή web services. Η ενσωμάτωση ενός web service στο MSEE περιλαμβάνει την εγκατάσταση σε web server, την εγγραφή στην υποδομή παροχής υπηρεσιών του MSEE μέσω της παραγωγή περιγραφής WSDL και των μηχανισμών διαλειτουργικότητας της ΠΑ. Η διαδικασία αυτή περιγράφεται αναλυτικά στις επόμενες ενότητες.

Στο ακόλουθο σχήμα, παρουσιάζεται η ροή UML προς Java ως ειδική περίπτωση της γενικής διαδικασίας μετασχηματισμού μοντέλων της Πλατφόρμας Ανάπτυξης.



Σχήμα 22: Δομική ροή εργασίας στη πρωτότυπη Πλατφόρμα Ανάπτυξης

Αναλυτικότερα, τα εργαλεία που χρησιμοποιούνται για τη ροή αυτή είναι τα ακόλουθα:

5.3.2.1.1 Επεξεργαστής UML

Η διαδικασία ξεκινά με την εισαγωγή ενός μοντέλου TIM σε UML. Ο χρήστης επεξεργάζεται το μοντέλο με τη διεπαφή και τα εργαλεία του πρόσθετου Papyrus⁶² UML για το Eclipse (π.χ. σε γραφική μορφή ή μορφή XML).

Το Papyrus είναι ένα γραφικό εργαλείο επεξεργασίας για μοντέλα του Eclipse Modelling Framework, αλλά κυρίως για μοντέλα της UML 2.4.1. Δίνει έμφαση στη συμβατότητα με τις προδιαγραφές του OMG για τη UML 2⁶³ και έχει άδεια «Open Source», επιτρέποντας την τροποποίηση και προσαρμογή του στις ιδιαίτερες ανάγκες της Πλατφόρμας Ανάπτυξης.

5.3.2.1.2 Εφαρμογή προφίλ UML για την τεχνολογία-στόχο

Το Papyrus υποστηρίζει τη χρήση προφίλ UML για την τροποποίηση και εξειδίκευση του περιβάλλοντος ανάπτυξης. Ένα «προφίλ UML» (UML profile) είναι ένα σύνολο προδιαγραφών που τροποποιούν το βασικό μεταμοντέλο των διαγραμμάτων της UML. Χρησιμοποιούνται για την εξειδίκευση της UML «γενικής χρήσης» για συγκεκριμένες εφαρμογές και σκοπούς. Ένα προφίλ μπορεί να περιλαμβάνει την προσθήκη επιπλέον ορολογίας, σύνταξης, σημειογραφίας καθώς και νέους περιορισμούς και κανόνες αντιστοίχισης (mapping rules).

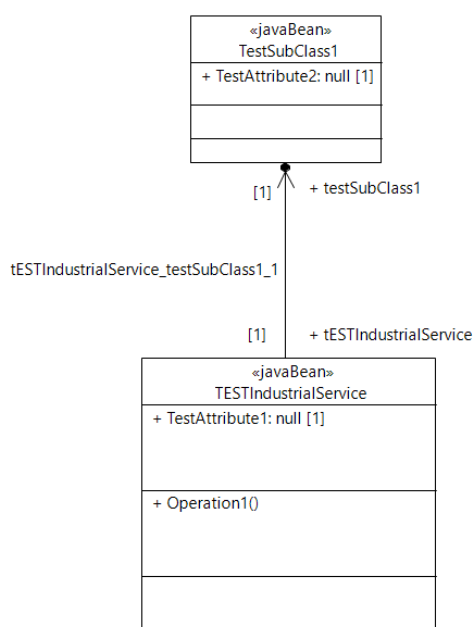
Στη περίπτωση της ΠΑ, τα προφίλ UML χρησιμοποιούνται για να εισάγουν στοιχεία της τεχνολογίας – στόχου στο περιβάλλον επεξεργασίας UML. Τα στοιχεία αυτά θα χρησιμοποιήσει ο χρήστης για να εμπλουτίσει το μοντέλο TIM και να δημιουργήσει το

⁶²<http://www.papyrusuml.org/>

⁶³ <http://www.uml.org/>

αντίστοιχο μοντέλο TSM για την επιλεγμένη τεχνολογία στόχο. Το προφίλ μπορεί να χρησιμοποιηθεί για να εισάγει στοιχεία των τελικών γλωσσών προγραμματισμού, αλλά και στοιχεία για την περιγραφή άλλων τεχνολογικών υποδομών που θα υποστηρίξουν το σύστημα υπηρεσιών, στοιχεία του βιομηχανικού πεδίου εφαρμογής, του βιομηχανικού οικοσυστήματος, ή ακόμα και συγκεκριμένης επιχείρησης. Έτσι, η χρήση των προφίλ UML ενισχύει την ευελιξία της ΠΑ, και την προσαρμοστικότητά της σε διάφορα περιβάλλοντα ανάπτυξης και βιομηχανικά πεδία.

Για τις ανάγκες της πρωτότυπης Πλατφόρμας Ανάπτυξης, αναπτύχθηκε ένα προφίλ UML, το οποίο εισάγει σημασιολογικά στοιχεία της Java (όπως το στερεότυπο «JavaBean», επιπλέον τύπους δεδομένων κ.ά.). Η εφαρμογή του προφίλ παρέχει στον χρήστη τα στοιχεία που χρειάζεται για να προσθέσει επισημάνσεις στο μοντέλο TIM και να δημιουργήσει ένα μοντέλο TSM εστιασμένο στη Java.



Σχήμα 23: Παράδειγμα επέκτασης μετα-μοντέλου UML για βιομηχανικές υπηρεσίες στη γλώσσα Java

Στο Σχήμα 23 φαίνεται ένα παράδειγμα προφίλ UML, το οποίο επεκτείνει το βασικό μεταμοντέλο των διαγραμμάτων κλάσης της UML. Στη περίπτωση αυτή εισάγει δύο καινούρια είδη κλάσεων (TestSubClass1 και TESTIndustrialService), οι οποίες συμφωνούν με το στερεότυπο «javaBean»⁶⁴ και έχουν τις ιδιότητές του. Το προφίλ εισάγει στοιχεία της γλώσσας Java, όπως, π.χ. τύπους δεδομένων της γλώσσας. Έτσι, γενικές κλάσεις UML μπορούν πλέον να μαρκαριστούν με το στερεότυπο «javaBean» (λ.χ. με αντιστοίχιση τύπου μοντέλου), ενώ οι ιδιότητές τους να μαρκαριστούν με τους κατάλληλους τύπους δεδομένων (λ.χ. με αντιστοίχιση στιγμιότυπου μοντέλου). Έτσι, μία γεννήτρια κώδικα θα μπορεί να

⁶⁴ <https://docs.oracle.com/javase/tutorial/javabeans/index.html>

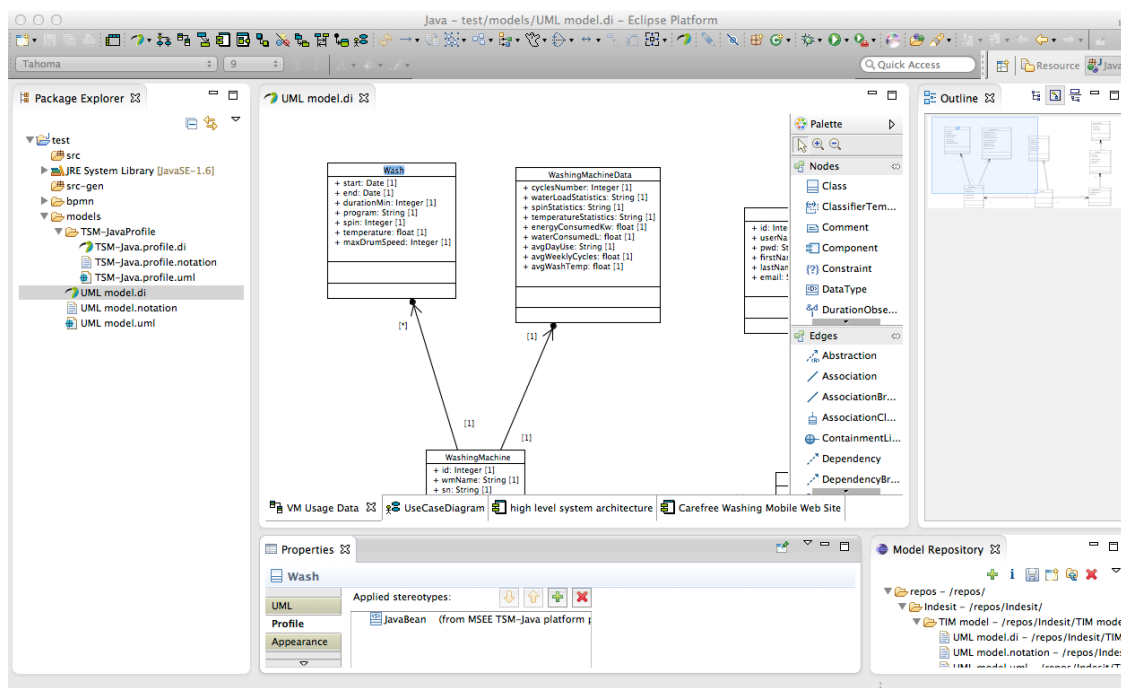
παράξει των αντίστοιχο κώδικα κλάσεων javaBean με τα απαραίτητα στοιχεία που ορίζει η γλώσσα και το πρότυπο javaBean.

Έτσι, ο χρήστης εισάγει το κατάλληλο προφίλ UML για τον εμπλουτισμό του μοντέλου. Μετά από την εφαρμογή του προφίλ στο μοντέλο, γίνεται διαθέσιμο το στερεότυπο «JavaBean» καθώς και επιπλέον τύποι δεδομένων της Java (πέρα από τους ήδη ορισμένους από το μεταμοντέλο της UML ως Primitive Types π.χ. Integer, String κτλ.).

Τα στοιχεία αυτά μπορούν να εφαρμοστούν στο μοντέλο διαμέσου της διεπαφής του Papyrus UML, και συγκεκριμένα μέσα από τις «καρτέλες ιδιοτήτων» για κάθε στοιχείο του μοντέλου. Ο χρήστης εμπλουτίζει το μοντέλο με τα νέα στοιχεία, από το προφίλ UML, δηλαδή προσθέτει το στερεότυπο JavaBean όπου αφορά, και τους τύπους δεδομένων των ιδιοτήτων της κάθε κλάσης (π.χ. float, date, Boolean κτλ.).

Τη πρωτότυπη εφαρμογή, ο εμπλουτισμός των μοντέλων περιλαμβάνει τα ακόλουθα βήματα:

- Εφαρμογή του προφίλ UML “MSEE-Java” στο εισηγμένο μοντέλο TIM, το οποίο παρέχει τύπους δεδομένων και στερεότυπα που υποστηρίζουν το τεχνολογικό μοντέλο της Java.
- Περιγραφή των ιδιοτήτων της κάθε κλάσης με βάση απλούς ή πολύπλοκους τύπους δεδομένων της Java, όπως αυτοί παρέχονται από το προφίλ “MSEE-Java”.
- Συσχέτιση της κάθε κλάσης UML με το στερεότυπο <<JavaBean>>, για την αυτόματη παραγωγή κώδικα με βάση τα περιεχόμενά του.
- Συσχέτιση της κάθε κλάσης που περιγράφει μία ξεχωριστή υπηρεσία web (web service stub) με το στερεότυπο <<Webservice>>.



Σχήμα 24: Επεξεργασία και εμπλουτισμός διαγράμματος κλάσεων UML στη Πλατφόρμα Ανάπτυξης

5.3.2.1.3 Γεννήτρια κώδικα

Μετά την ολοκλήρωση της διαδικασίας εμπλουτισμού, ο χρήστης αιτείται την παραγωγή πηγαίου κώδικα από το μοντέλο. Η ΠΑ χρησιμοποιεί το ενσωματωμένο πρόσθετο Acceleo⁶⁵ του Eclipse σε συνδυασμό με τους κανόνες μετασχηματισμού και παράγει κλάσεις Java για κάθε κλάση του μοντέλου UML, μαζί με τις αντίστοιχες ιδιότητες και μεθόδους.

Το εργαλείο Acceleo, στηρίζεται στη γλώσσα μετασχηματισμού μοντέλων MOF Model-To-Text (MOFM2T) (Object Management Group, 2008), η οποία βασίζεται στο πρότυπο Meta-Object-Facility και αποτελεί μέρος της αρχιτεκτονικής MDA όπως έχει καθοριστεί από το OMG. Το πρόσθετο Acceleo χρησιμοποιεί κανόνες εκφρασμένους στη γλώσσα MOFM2T για το μετασχηματισμό μοντέλων σε κείμενο, όπως π.χ. πηγαίο κώδικα, περιγραφές κειμένου ή τεκμηρίωση.

Για την χρήση του στην πρωτότυπη ΠΑ, αναπτύχθηκε ένα σύνολο κανόνων για τον μετασχηματισμό εμπλουτισμένων (με σημασιολογικά στοιχεία της Java) μοντέλων κλάσεων UML σε κώδικα Java (Javabeans). Οι κανόνες αυτοί αντιστοιχίζουν τα στοιχεία του μοντέλου UML με δομές και εκφράσεις της Java.

Αναλυτικότερα:

- Κλάσεις UML με στερεότυπο «Javabean» σε Javabeans.
- Ιδιότητες UML σε ιδιότητες κλάσεων Java με τον αντίστοιχο τύπο.
- Λειτουργίες κλάσεων UML (operations) σε μεθόδους Java.
- Σχέσεις γενίκευσης κλάσεων UML σε σχέσεις επέκτασης (π.χ. «public class Class2 extends Class 1 {...}»).
- Προσθήκη κειμένου και σημειώσεων εντός του κώδικα όπως:
 - Εισαγωγικές πληροφορίες: «/** This file has been generated by MSEE Generic Service Development Platform»
 - Placeholders για τα σημεία που θα αναπτυχθούν περαιτέρω από τον χρήστη, όπως:
 - /** The documentation of the method operation1.
 - // Start of user code for the body of operation1 // TODO should be implemented // End of user code
 - /* * The documentation of the getter getAttribute1.
 - /** The documentation of the setter setAttribute1.
 - Κ.α..

Οι κανόνες μετασχηματισμού απεικονίζονται αναλυτικά στο ακόλουθο σχήμα:

```
[comment encoding = UTF-8 /]
/**
 * The documentation of the module generate.
 */]
[module generate('http://www.eclipse.org/uml2/4.0.0/UML')]
/**
 * The documentation of the template generateElement:
```

⁶⁵ <https://www.eclipse.org/acceleo/>

```

* This module is being used to develop transformation rules for MSEE GSDP enriched UML-to-
Java transformations
* The module builds class templates inc. methods and attributes, and is informed by
generalisation
* @param aClass
*/]

[template public generateElement(aClass : Class)]

[comment @main/]

[file (aClass.name.concat('.java'), false)]

/**
 * This file has been generated by MSEE Generic Service Development Platform:
Transformation Rules Dev. Version
*/

import java.util.*;
import java.math.*;
//[protected ('for imports')]
//[protected]

/**
 * The documentation of the class [aClass.name/]
 *
 * @generated
 */

public class [aClass.name.toUpperFirst()/] [if (not (aClass.superClass-
>isEmpty()))]extends [aClass.superClassName/][if] {

    [for (p: Property | aClass.attribute) separator('\n')]
    private [p.type.name/] [p.name.toLowerFirst()/];
    [//for]

        public class [aClass.name.toUpperFirst()/] {
            super();
        }

    [for (o: Operation | aClass.ownedOperation) separator('\n')]
    /**
     * The documentation of the method [o.name/].
     *
     * @generated
     */
    public void [o.name/]() {
        //[protected (o.name)]
        // TODO should be implemented
        //[protected]
    }
    [//for]

    [for (p: Property | aClass.attribute) separator('\n')]
    /**
     * The documentation of the getter get[p.name.toUpperFirst()/].
     *
     * @generated
     */
    public [p.type.name/] get[p.name.toUpperFirst()/]() {
        return this.[p.name/];
    }

    /**
     * The documentation of the setter set[p.name.toUpperFirst()/].
     *
     * @generated
     */
    public void set[p.name.toUpperFirst()/]([p.type.name/]
new[p.name.toUpperFirst()/]) {
        this.[p.name/] = new[p.name.toUpperFirst()/];
    }
    [//for]

```

```
}  
[/file]  
[/template]
```

Σχήμα 25: Κανόνες μετασχηματισμού μοντέλου TSM σε Java στη Πλατφόρμα Ανάπτυξης

Καθώς το μοντέλο UML δεν περιλαμβάνει πληροφορία για τη συμπεριφορά των κλάσεων, των μεθόδων τους, και άλλων στοιχείων, ο παραγόμενος κώδικας αποτελεί ένα «πλαίσιο κώδικα» (code template) το οποίο ο χρήστης θα επεξεργαστεί και εμπλουτίσει με τη χρήση συμβατικών μεθόδων και εργαλείων ανάπτυξης λογισμικού. Ένα τέτοιο πλαίσιο απεικονίζεται στο Σχήμα 26

```
/**  
 * This file has been generated by MSEE Generic Service Development Platform  
 */  
  
// Start of user code for imports  
import java.util.*;  
import java.math.*;  
// End of user code  
  
/**  
 * The documentation of the class HeatPump  
 *  
 * @generated  
 */  
public class HeatPump extends InstalledEquipment {  
  
    private int compr_modulation;  
    private boolean running;  
    private boolean operation;  
    private int serial_no;  
  
    private int comp_modulation;  
  
    public HeatPump() {  
        super();  
    }  
  
    /**  
     * The documentation of the method report_status.  
     *  
     * @generated  
     */  
    public void report_status() {  
        // Start of user code for the body of report_status  
        // TODO should be implemented  
        // End of user code  
    }  
  
    /**  
     * The documentation of the getter getCompr_modulation.  
     *  
     * @generated  
     */  
    public int getCompr_modulation() {  
        return this.compr_modulation;  
    }  
  
    /**  
     * The documentation of the setter setCompr_modulation.  
     *  
     * @generated  
     */  
    public void setCompr_modulation(int newCompr_modulation) {  
        this.compr_modulation = newCompr_modulation;  
    }  
  
    /**  
     * The documentation of the getter getRunning.  
     *
```



```

* @generated
*/
public boolean getRunning() {
    return this.running;
}

/**
 * The documentation of the setter setRunning.
 *
 * @generated
 */
public void setRunning(boolean newRunning) {
    this.running = newRunning;
}

/**
 * The documentation of the getter getSerial_no.
 *
 * @generated
 */
public int getSerial_no() {
    return this.serial_no;
}

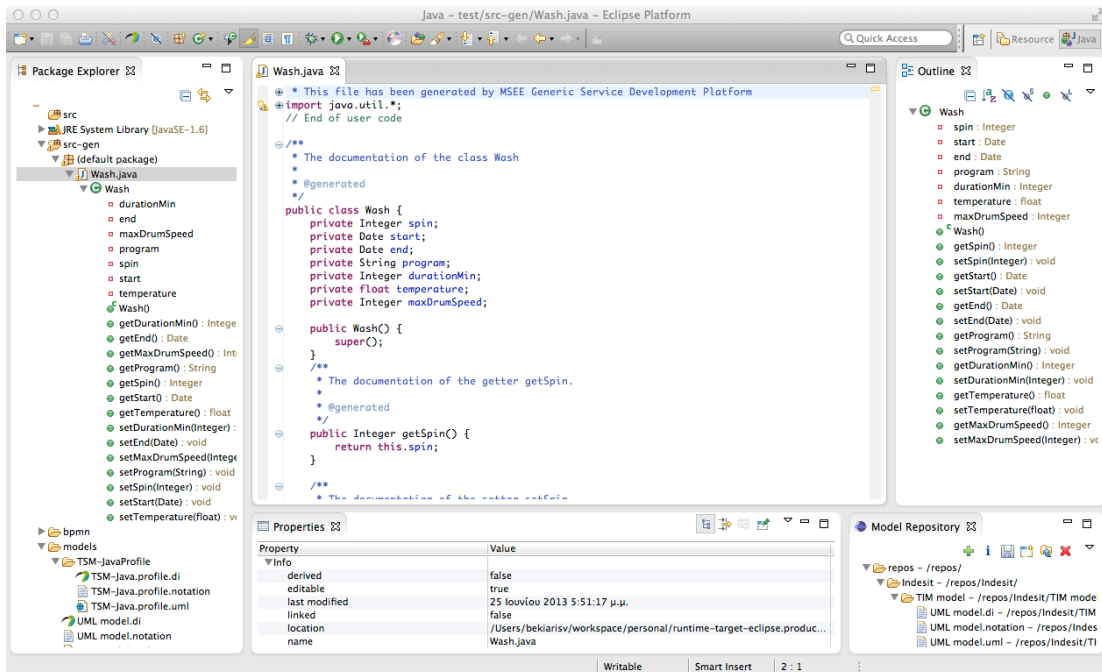
/**
 * The documentation of the setter setSerial_no.
 *
 * @generated
 */
public void setSerial_no(int newSerial_no) {
    this.serial_no = newSerial_no;
}
}

```

Σχήμα 26: Πλαίσιο κώδικα κλάσης Java που παράχθηκε αυτόματα από την Πλατφόρμα Ανάπτυξης

Ανάπτυξη κώδικα Java

Μόλις παραχθεί ο απαιτούμενος κώδικας, προστίθεται στο περιβάλλον εργασίας του χρήστη (user workspace) με τη μορφή αρχείων κλάσεων Java (.java), όπως φαίνεται στο Σχήμα 27.



Σχήμα 27: Επεξεργασία κώδικα Java που παράχθηκε από ένα μοντέλο TSM στη Πλατφόρμα Ανάπτυξης

Η πλατφόρμα του Eclipse παρέχει ένα σύνολο από εργαλεία ανάπτυξης (Java Development Tools – JDT), τα οποία επιτρέπουν την ανάπτυξη λογισμικού σε Java. Τα εργαλεία αυτά χρησιμοποιούνται στα πλαίσια της ΠΑ χωρίς τροποποιήσεις για την περαιτέρω ανάπτυξη του κώδικα και την παραγωγή των εκτελέσιμων αρχείων. Τα εργαλεία αυτά μπορεί να περιλαμβάνουν το JDT του Eclipse, αλλά και άλλα πρόσθετα που θα έχει ενσωματώσει ο χρήστης για την κάλυψη συγκεκριμένων αναγκών και εφαρμογών.

5.3.2.2 Ανάπτυξη Λογισμικού στη Συμπεριφορική ροή: BPMN προς εκτελέσιμες επιχειρησιακές διαδικασίες (.BAR)

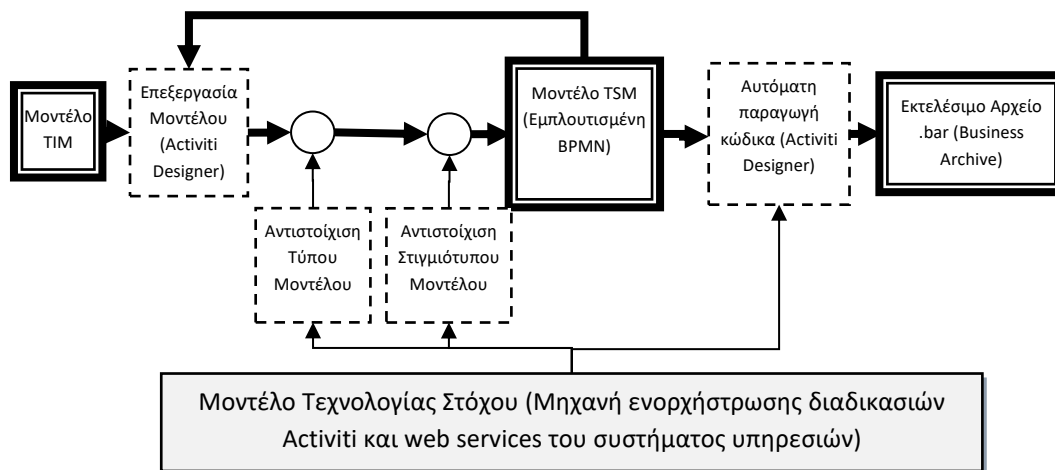
Η δεύτερη ροή εργασίας περιλαμβάνει την δημιουργία σύνθετων ενορχηστρώσεων υπηρεσιών από μοντέλα επιχειρησιακών διαδικασιών σε BPMN. Στα πλαίσια του MSEE, οι επιχειρησιακές διαδικασίες εκτελούνται από τη «μηχανή ενορχήστρωσης» (orchestration engine) της εφαρμογής Innovation Ecosystem Platform του MSEE. Η «μηχανή ενορχήστρωσης» της IEP βασίζεται με τη σειρά της στην εφαρμογή Activiti⁶⁶, και δέχεται ως είσοδο πακέτα (business archives - .bar) που περιλαμβάνουν, μαζί με τις περιγραφές των επιχειρησιακών διαδικασιών, και όλα τα επιπλέον στοιχεία (όπως φόρμες, scripts κτλ.) που απαιτούνται για την εκτέλεσή τους.

Η ροή περιλαμβάνει τα ακόλουθα βήματα:

- Ο χρήστης εισάγει ένα μοντέλο TIM BPMN από το αποθετήριο μοντέλων
- Ο χρήστης ανοίγει το μοντέλο με τον επεξεργαστή BPMN
- Ο χρήστης επεξεργάζεται και εμπλουτίζει το μοντέλο προσθέτοντας στοιχεία σχετικά με την πλατφόρμα εκτέλεσης των επιχειρησιακών διαδικασιών και τις τεχνολογικές υποδομές που τη στηρίζουν. Ο χρήστης αιτείται την παραγωγή της εκτελέσιμης περιγραφής από την Πλατφόρμα Ανάπτυξης. Η εκτελέσιμη περιγραφή περιλαμβάνει την επιχειρησιακή διαδικασία, καθώς και όλα τα στοιχεία που απαιτούνται για την εκτέλεσή της σε μορφή .BAR.

Η ροή αυτή παρουσιάζεται στο Σχήμα 28 ως ειδική περίπτωση της γενικής διαδικασίας μετασχηματισμού μοντέλων στο MSEE.

⁶⁶ <https://www.activiti.org/>



Σχήμα 28: Συμπεριφορική ροή εργασίας στη πρωτότυπη Πλατφόρμα Ανάπτυξης

Η BPMN μπορεί να χρησιμοποιηθεί για να παράγει «ενορχηστρώσεις υπηρεσιών» (service orchestrations) με τη μορφή αρχείων .BAR του Activiti, τα οποία εκτελούνται στη «μηχανή επιχειρησιακών διαδικασιών» της IEP (Innovation Ecosystem Platform).

Με βάση τις απαιτήσεις για τη συγκεκριμένη ροή εργασιών, ο επεξεργαστής BPMN βασίζεται στη πλατφόρμα διαχείρισης επιχειρησιακών διαδικασιών Activiti (Activiti Business Process Management Platform). Η Activiti βασίζεται στη διαχείριση και εκτέλεση διαδικασιών βασισμένων σε μοντέλα BPMN2, ενώ παράλληλα προσφέρει λειτουργίες όπως στατιστικά, καταγραφή, διεπαφές χρήστη κτλ.

Η πλατφόρμα Activiti περιλαμβάνει μία σειρά από εφαρμογές οργανωμένες σε τρεις ευρείες κατηγορίες:

- Μοντελοποίηση διαδικασιών, η οποία περιλαμβάνει τα Activiti Modeller, Activiti Designer και Activiti Kickstart.
- Περιβάλλον εκτέλεσης (runtime) και το Activiti Engine.
- Διαχείριση εκτέλεσης, το οποίο περιλαμβάνει και τα Activiti Explorer και Activiti REST.

Ο επεξεργαστής BPMN βασίζεται στο πρόσθετο ανοικτού κώδικα Activiti Designer. Το πρόσθετο παρέχει λειτουργικότητα για τη δημιουργία και επεξεργασία μοντέλων BPMN, καθώς και τον εμπλουτισμό τους με δεδομένα και στοιχεία απαραίτητα για την εκτέλεσή τους από την Activiti Engine. Με βάση τα μοντέλα αυτά, το Activiti Designer μπορεί να παράγει «Τεχνουργήματα Εγκατάστασης Υπηρεσιών» («Service Deployment Artefacts»), δηλαδή, εκτελέσιμα αρχεία .BAR. Το Activiti Designer μπορεί να τροποποιηθεί περαιτέρω με αλλαγές στη διεπαφή και την προσθήκη νέων στοιχείων μοντελοποίησης, κανόνων πιστοποίησης μοντέλων, κτλ. (Activiti, 2015)

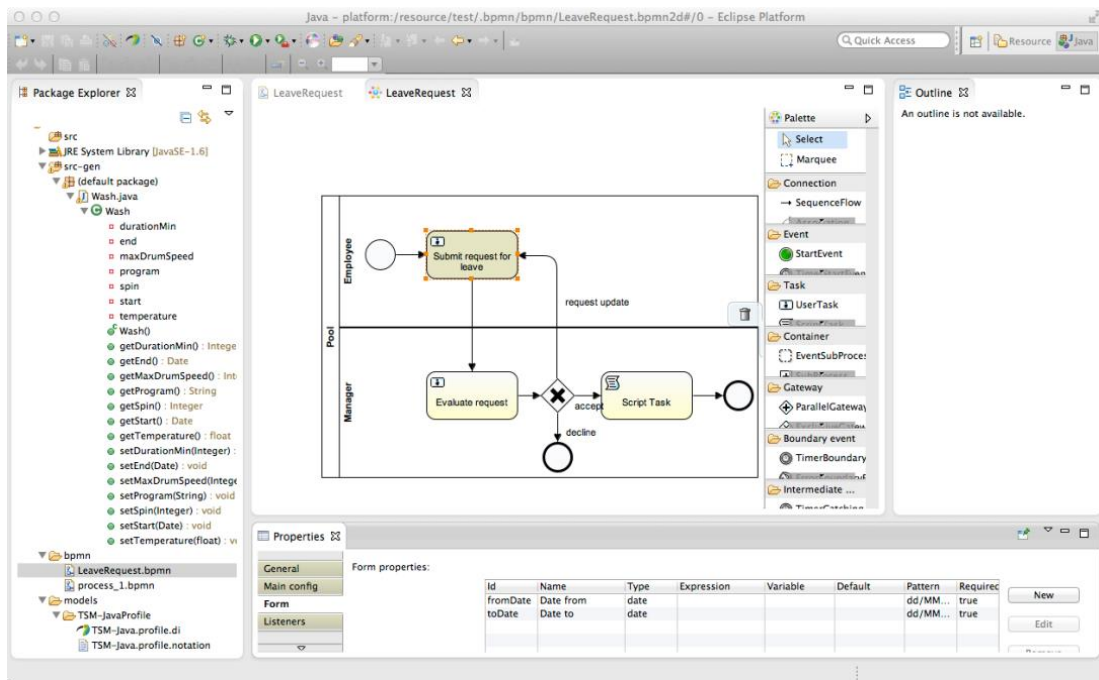
Η ροή εργασίας ξεκινά με τον χρήστη να εισάγει ένα μοντέλο TIM σε γλώσσα BPMN από το Αποθετήριο Μοντέλων, σε ένα έργο (Project) του Activiti Designer. Ο χρήστης επεξεργάζεται και εμπλουτίζει με επιπλέον στοιχεία μοντελοποίησης και σημασιολογίας (semantics) το

μοντέλο στο περιβάλλον επεξεργασίας ώστε να το προετοιμάσει για εκτέλεση από τη μηχανή επιχειρησιακών διαδικασιών της IEP. Συγκεκριμένα, ο χρήστης μπορεί να τροποποιήσει το μοντέλο, να προσθέσει αναφορές σε web services της πλατφόρμας MSEE, περιγραφές φορμών, μεταβλητές, σενάρια, επιχειρησιακούς κανόνες και στοιχεία της πλατφόρμας Activiti που δεν αποτελούν μέρος της τυπικής BPMN. Αναλυτικότερα αυτό μπορεί να περιλαμβάνει εργασίες όπως:

- Επεξεργασία μοντέλου: υλοποίηση και ανάλυση λειτουργικότητας ή οποία περιγράφεται με αφηρημένους όρους στο μοντέλο εισόδου, βελτιώσεις στη ροή εργασίας, ευχρηστία, συνέπεια, την αναχρησιμοποίηση υπαρχόντων πόρων και περαιτέρω στοιχεία BPMN (εργασίες, πύλες ελέγχου ροής, μηνύματα κ.α.)
- Κλήση υπηρεσιών από την πλατφόρμα του MSEE: Οι υπηρεσίες που είναι εγγεγραμμένες στη πλατφόρμα SOA του MSEE μπορούν να κληθούν κατά την εκτέλεση των επιχειρησιακών διαδικασιών. Στη συγκεκριμένη περίπτωση, οι προγραμματιστές προσθέτουν εξειδικευμένες εργασίες («MSEE service tasks»), οι οποίες καλούν, μεταφέρουν στοιχεία και δέχονται αποτελέσματα από υπηρεσίες της υπηρεσιοστραφούς υποδομής του MSEE. Αυτές αντικαθιστούν υφιστάμενες εργασίες του μοντέλου εισόδου ή προστίθενται εκ νέου. Αυτή η λειτουργικότητα επιτρέπει την δημιουργία «συνθέσεων υπηρεσιών» της SOA και είναι επέκταση του πλαισίου Activiti BPM που αναπτύχθηκε ειδικά για τις ανάγκες του MSEE.
- Προσθήκη κλάσεων Java: Περίπλοκη συμπεριφορά που δεν μπορεί να αποτυπωθεί αποτελεσματικά από τα στοιχεία της BPMN 2.0 μπορεί να εισαχθεί μέσω κλάσεων της Java. Αυτό επιτυγχάνεται με τη συμπερίληψη εργασιών τύπου «Activiti Java Service», οι οποίες καλούν κλάσεις Java, μεθόδους, μεταφέρουν τιμές σε πεδία, λαμβάνουν αποτελέσματα κ.α.
- Σχεδιασμός φορμών: οι προγραμματιστές μπορούν να ορίσουν φόρμες και να τις συνδέσουν με συγκεκριμένες εργασίες, επιτυγχάνοντας την αλληλεπίδραση με τον χρήστη. Οι φόρμες που ορίζονται με αυτό τον τρόπο μπορούν να παρουσιαστούν μέσω της διεπαφής web που προσφέρει το IEP στους τελικούς χρήστες.
- Προσθήκη στοιχείων εκτέλεσης της επιχειρησιακής διαδικασίας: Μεταβλητές που μεταφέρουν δεδομένα κατά την εκτέλεση της διαδικασίας, scripts μέσω εργασιών σεναρίων («Script tasks») σύμφωνα με το πρότυπο JSR 223⁶⁷, επιχειρησιακοί κανόνες μέσω εργασιών κανόνων («Rules Tasks»), εργασίες αποστολής ηλεκτρονικού ταχυδρομείου, αναθέσεις εργασιών σε χρήστες ή ομάδες χρηστών, έλεγχος ροής κ.α. Τα περισσότερα από αυτά τα στοιχεία είναι επεκτάσεις της BPMN 2.0 που χρησιμοποιεί το πλαίσιο Activiti BPM.

Μόλις ολοκληρωθεί η διαδικασία, ο χρήστης αιτείται την παραγωγή του σχετικού αρχείου .BAR. Με βάση το εμπλουτισμένο μοντέλο, η πλατφόρμα παράγει το αρχείο .BAR, το οποίο προστίθεται στο περιβάλλον εργασίας του χρήστη.

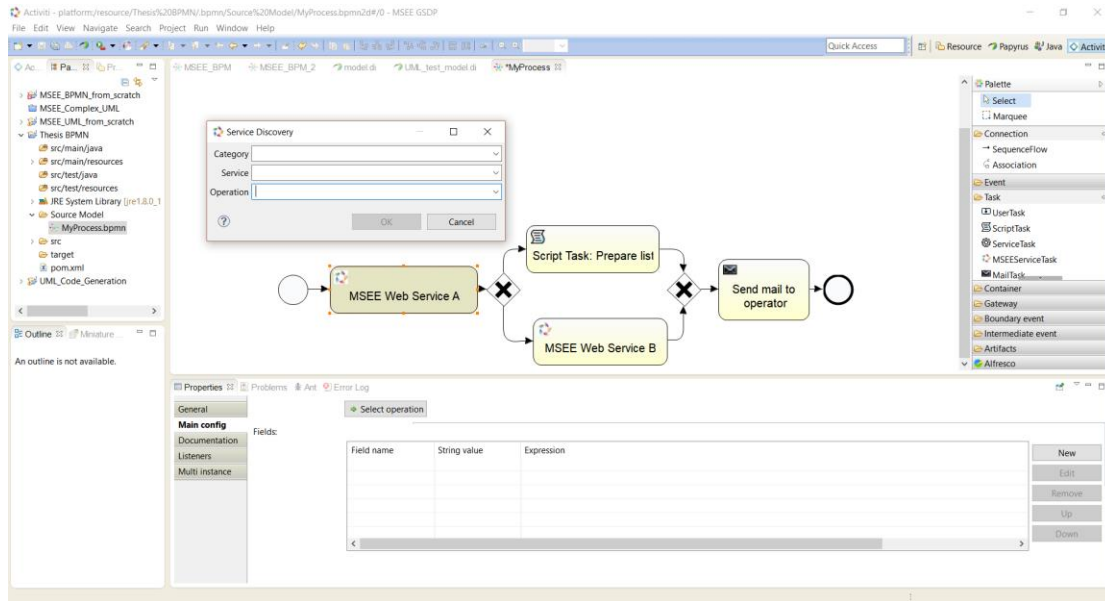
⁶⁷ <https://www.jcp.org/en/jsr/detail?id=223>



Σχήμα 29: Επεξεργασία μοντέλου BPMN στη πρωτότυπη πλατφόρμα ανάπτυξης

Το αρχείο .BAR περιέχει την περιγραφή της επιχειρησιακής διαδικασίας, καθώς και τα συμπληρωματικά στοιχεία (assets) (ή αναφορές σε αυτά) που απαιτούνται για την εκτέλεσή της στην IEP. Σημειώνεται ότι τα στοιχεία αυτά μπορούν να έχουν παραχθεί και με τη χρήση εργαλείων και δυνατοτήτων που παρέχει η ΠΑ (π.χ. λογισμικό Java από UML) ή άλλες δυνατότητες του περιβάλλοντος του Eclipse και των πρόσθετών του.

Για να αξιοποιηθούν οι υπηρεσιοστραφείς δυνατότητες της πλατφόρμας του MSEE, ο Activiti Designer τροποποιήθηκε μέσω την προσθήκης ενός νέου στοιχείου εργασίας BPMN (BPMN task) με τίτλο «MSEE Service Task» («Εργασία Υπηρεσίας MSEE»). Η συγκεκριμένη εργασία αποτελεί επέκταση του υφιστάμενου τύπου «Web Service Task» του Activiti, αλλά τροποποιημένου ώστε να διευκολύνει την αναφορά και επίκληση εγγεγραμμένων υπηρεσιών της υπηρεσιοστραφούς αρχιτεκτονικής του MSEE. Η εισαγωγή των web services απεικονίζεται στο Σχήμα 30.



Σχήμα 30: Αναζήτηση και ενσωμάτωση υφιστάμενων web services του συστήματος υπηρεσιών του MSEE στην επιχειρησιακή διαδικασία

Όπως φαίνεται στο σχήμα, η καρτέλα ιδιοτήτων («properties») του συγκεκριμένου τύπου εργασίας έχει σχεδιαστεί με τέτοιο τρόπο ώστε να επιτρέπει στους χρήστες να αναζητούν διαδικτυακές υπηρεσίες του MSEE και να προσθέτουν αναφορές σε αυτές και να τους μεταδίδουν ορίσματα και παραμέτρους. Η αναζήτηση των εγγεγραμμένων υπηρεσιών γίνεται μέσω του πρόσθετου διαλειτουργικότητας με την υποδομή παροχής υπηρεσιών του MSEE (SDI Integration Plugin). Το πρόσθετο αξιοποιεί το API ανακάλυψης υπηρεσιών της SDI για να υποστηρίξει την αναζήτηση του χρήστη.

Συγκεκριμένα, αφού οι χρήστες πατήσουν στο «Select Operation»:

- Καλείται το πρόσθετο διαλειτουργικότητας με την SDI
- Το πρόσθετο εμφανίζει κατάλογο με την κατηγορίες διαθέσιμων υπηρεσιών των υπηρεσιών και των λειτουργιών (categories, services, operations). Οι χρήστες μπορούν να αναζητήσουν την κατάλληλη υπηρεσία και λειτουργία μέσω διαδοχικών drop-down μενού. Έτσι, ο χρήστης επιλέγει πρώτα την αντίστοιχη «σημασιολογική» κατηγορία (π.χ. λογιστική, εφοδιαστική αλυσίδα, κτλ.), στη συνέχεια την υπηρεσία (δηλαδή εγγεγραμμένες υπηρεσίες που ανήκουν σε αυτή) και τέλος τη λειτουργία της υπηρεσίας αυτής που θέλει να καλείται από την μηχανή εκτέλεσης της IEP στο αντίστοιχο σημείο της μοντελοποιημένης επιχειρησιακής ροής.
- Μόλις ο χρήστης κάνει τις σχετικές επιλογές, τα μεταδεδομένα του στοιχείου συμπληρώνονται αυτόματα με τις απαραίτητες αναφορές.
- Στη συνέχεια, οι χρήστες μπορούν να προσθέσουν επιπλέον παραμέτρους π.χ. επιλογές, τιμές, μεταβλητές κτλ., που επιθυμούν να περάσουν στην υπηρεσία τη στιγμή της εκτέλεσης.

Με τον τρόπο αυτό, η ΠΑ δίνει τη δυνατότητα ανάπτυξης υπηρεσιοστραφών συνθέσεων υπηρεσιών χρησιμοποιώντας μόνο εργαλεία «εντός πλατφόρμας» (in-platform) π.χ. αναπτύσσοντας υπηρεσίες σε Java και δημιουργώντας συνθέσεις τους σε BPMN. Με την

ολοκλήρωση της διαδικασίας, ο χρήστης επιλέγει τη δημιουργία του εκτελέσιμου αρχείου από την Πλατφόρμα Ανάπτυξης («Create Deployment Artefacts»).

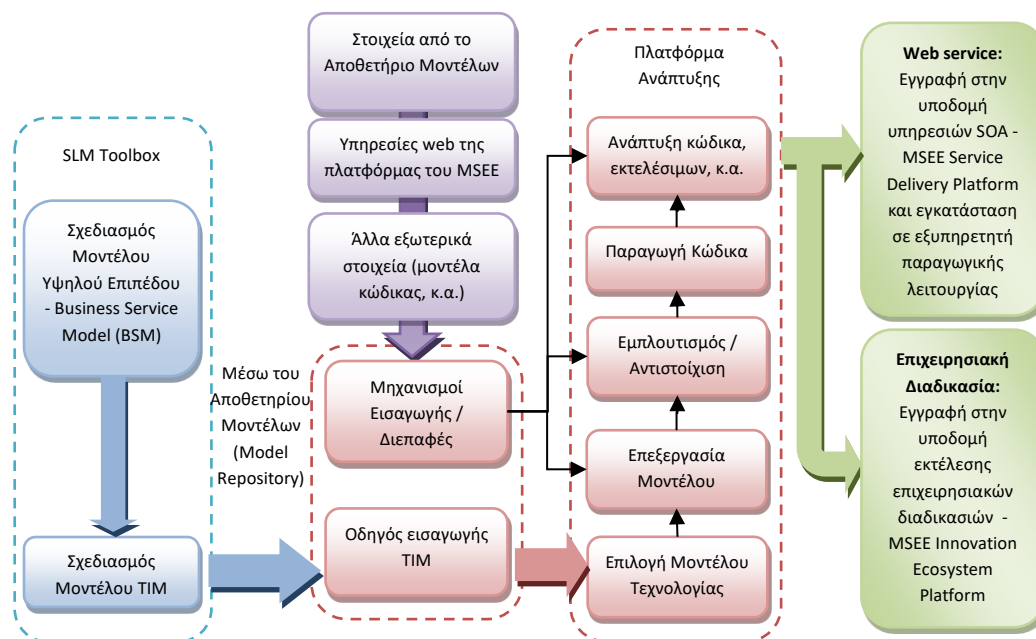
5.3.3 Διαλειτουργικότητα με άλλα στοιχεία της πλατφόρμας ΤΠΕ του MSEE

Πέρα από τις λειτουργικότητες ανάπτυξης λογισμικού, σημαντικό μέρος της προστιθέμενης αξίας που παρέχεται από την Πλατφόρμα Ανάπτυξης, προέρχεται από τη διαλειτουργικότητα της με την υποδομή ΤΠΕ του συστήματος υπηρεσιών. Οι ροές εργασίας απαιτούν διασύνδεση με τα στοιχεία σχεδιασμού υπηρεσιών του MSEE (δηλαδή το SLM Toolbox), καθώς και αυτά που σχετίζονται με τη θέση σε παραγωγική λειτουργία και εκτέλεση του λογισμικού (Service Delivery Platform, Innovation Ecosystem Platform).

Επιπλέον, η δημιουργία συνθέσεων web services που υπάρχουν ήδη στη πλατφόρμα του MSEE απαιτεί μηχανισμούς για την εύρεση τους και την προσθήκη αναφορών σε αυτές. Τέλος, η συνεργασία μεταξύ των εμπλεκόμενων (π.χ. ειδικών του πεδίου εφαρμογής, επιχειρησιακοί αναλυτές και μηχανικών λογισμικού) απαιτεί πρόσβαση σε κοινούς πόρους και κοινά αποθετήρια. Στη Πλατφόρμα Ανάπτυξης, οι διασυνδέσεις αυτές υποστηρίζονται από ένα σύνολο «Στοιχείων Διαλειτουργικότητας»

Όπως περιγράφηκε στις προηγούμενες ενότητες, η Πλατφόρμα Ανάπτυξης συμμετέχει στη διαδικασία ανάπτυξης λογισμικού web services διαλειτουργώντας με το SLM Toolbox και τη Service Delivery Platform. Η διασύνδεση της Πλατφόρμας Ανάπτυξης με το SLM Toolbox γίνεται με τη μεσολάβηση του Αποθετηρίου Μοντέλων. Το Αποθετήριο Μοντέλων επιτρέπει την αποθήκευση, αναζήτηση και ανάκληση των μοντέλων που δημιουργούνται στο SLM Toolbox. Τα μοντέλα αυτά εισάγονται στη Πλατφόρμα Ανάπτυξης και χρησιμοποιούνται για την ανάπτυξη υπηρεσιών web μέσω της διαδικασίας μετατροπής μοντέλων TIM σε TSM μέσω εμπλουτισμού με στοιχεία της τεχνολογίας στόχου, και της παραγωγής και επεξεργασίας του πηγαίου κώδικα. Οι υπηρεσίες web και οι επιχειρησιακές διαδικασίες / ενορχηστρώσεις υπηρεσιών που δημιουργούνται έτσι, εγκαθίστανται στους εξυπηρετητές παραγωγικής λειτουργίας και εγγράφονται στην υποδομή SOA του συστήματος υπηρεσιών (στη περίπτωση του MSEE στη Service Delivery Platform) ή στους μηχανισμούς εκτέλεσης επιχειρησιακών διαδικασιών (Innovation Ecosystem Platform).

Αυτή η πλήρως ολοκληρωμένη ροή εργασίας απεικονίζεται στο Σχήμα 31.



Σχήμα 31: Ροή εργασίας για την ανάπτυξη λογισμικού μέσω διαλειτουργικότητας των στοιχείων της πλατφόρμας του MSEE

Τα στοιχεία διαλειτουργικότητας παρουσιάζονται αναλυτικά στις ακόλουθες ενότητες.

5.3.3.1 Εξυπηρετητής Αποθετηρίου Μοντέλων και πρόσθετα Πρόσβαση

Η πρωτότυπη Πλατφόρμα Ανάπτυξης, βασίζεται σε ένα Αποθετήριο μοντέλων για την αποθήκευση, αναζήτηση και ανάκληση μοντέλων και άλλων πόρων. Αυτοί οι πόροι χρησιμοποιούνται για τη συνεργατική ανάπτυξη λογισμικού για υπηρεσίες στη βιομηχανία και σε αυτούς έχουν πρόσβαση οι χρήστες της Πλατφόρμας Ανάπτυξης και του λογισμικού σχεδιασμού υπηρεσιών (το SLM Toolbox).

Η συνεργατική και «δια-οργανωσιακή» φύση του συστήματος υπηρεσιών με πολλαπλούς εταίρους επιβάλλει την ανάγκη για προστασία των δεδομένων και τον έλεγχο πρόσβασης σε αυτά. Έτσι, το λογισμικό του αποθετηρίου παρέχει μηχανισμούς ελέγχου πρόσβασης και ακεραιότητας δεδομένων.

Το Αποθετήριο Μοντέλων αποθηκεύει τους πόρους σε μοναδικό κεντρικό εξυπηρετητή. Ο εξυπηρετητής παρέχει ένα κοινό σημείο πρόσβασης για ταυτόχρονη πρόσβαση από κάθε στιγμιότυπο της Πλατφόρμας Ανάπτυξης και του SLM Toolbox.

Η «στοίβα» (stack) του Αποθετηρίου Μοντέλων αποτελείται από τα παρακάτω στοιχεία:

- Model Repository Server (Εξυπηρετητής Αποθετηρίου Μοντέλων): Το κεντρικό και μοναδικό αποθετήριο
- Model Repository Client API (API Πελάτη Αποθετηρίου): Αποτελεί κατάλληλα υλοποιημένο πρόσθετο του Eclipse, ενσωματωμένο στη Πλατφόρμα Ανάπτυξης. Προσφέρει ένα API διαμέσου του οποίου άλλα πρόσθετα και στοιχεία της Πλατφόρμας Ανάπτυξης μπορούν να επικοινωνήσουν με τον κεντρικό εξυπηρετητή.

- Model Repository View (Προβολή Αποθετηρίου Μοντέλων): Αποτελεί κατάλληλα υλοποιημένο πρόσθετο του Eclipse, το οποίο παρέχει γραφική διεπαφή χρήστη για εργασία με τους πόρους του Αποθετηρίου Μοντέλων.

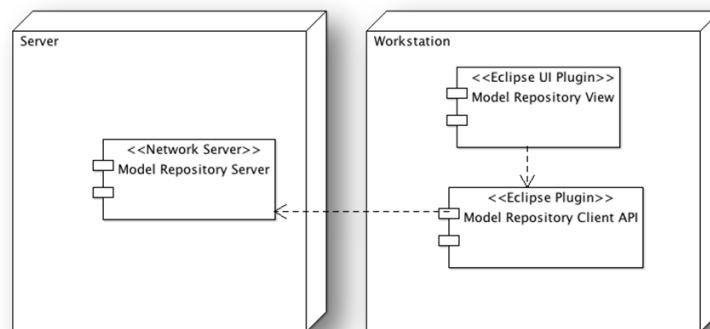


Figure 4: Model Repository deployment and access

Ο Εξυπηρετητής του Αποθετηρίου Μοντέλων υλοποιήθηκε ως ένα network process σε έναν κεντρικό εξυπηρετητή δικτύου. Η υλοποίηση αυτή επιτρέπει τη δημιουργία πολλαπλών αποθετηρίων π.χ. ένα κεντρικό αποθετήριο για ένα σύστημα υπηρεσιών ή για κάθε οργανισμό. Τα άλλα δύο στοιχεία επιτρέπουν την πρόσβαση στο αποθετήριο, και εκτελούνται τοπικά στους σταθμούς εργασίας των χρηστών (ως μέρη της Πλατφόρμας Ανάπτυξης και του SLM Toolbox).

5.3.3.1.1 Model Repository Server – Εξυπηρετητής Αποθετηρίου Μοντέλων

Ο εξυπηρετητής Αποθετηρίου περιέχει το αποθετήριο, υλοποιημένο ως ένας εξυπηρετητής WebDAV (Web-based Distributed Authoring and Versioning)⁶⁸ (Whitehead & Wiggins, 1998). Το WebDAV είναι ένα σύνολο από επεκτάσεις για το πρωτόκολλο HTTP/1.1, οι οποίες προσθέτουν νέες μεθόδους, κεφαλίδες, μορφές αιτήματος – απόκρισης κ.α. Οι επεκτάσεις αυτές παρέχουν επίσης τη δυνατότητα του κλειδώματος πόρων, τη συσχέτισή τους με μεταδεδομένα και τη διαχείριση χώρου ονομασιών (namespace management) για συλλογές πόρων. Μελλοντικές εκδόσεις θα περιλαμβάνουν διατεταγμένες συλλογές, χειρισμός εκδόσεων και ασφάλεια πρόσβασης. Έτσι, είναι ένα ισχυρό εργαλείο για την αποθήκευση, δημιουργία και πρόσβαση σε πόρους από πολλαπλούς, καταναμημένους χρήστες.

Οι επεκτάσεις WebDAV για το πρωτόκολλο HTTP/1.1 επιτρέπουν την υλοποίηση ενός ιεραρχικού αποθετηρίου πόρων, με αρχεία και φακέλους. Η υλοποίηση αυτή επιτρέπει στους χρήστες την αποθήκευση, όχι μόνο μοντέλων, αλλά οποιοδήποτε τύπου αρχείου που μπορεί να μεταδοθεί μέσω HTTP.

5.3.3.1.2 Model Repository Client API – API Πελάτη Αποθετηρίου

Το «API Πελάτη Αποθετηρίου» είναι λογισμικό που λειτουργεί σαν μεσολαβητής μεταξύ του Εξυπηρετητή Αποθετηρίου Μοντέλων και οποιουδήποτε άλλου πρόσθετου Eclipse της Πλατφόρμας Ανάπτυξης, το οποίο χρειάζεται να έχει πρόσβαση στο Αποθετήριο Μοντέλων. Το API περιλαμβάνει μία σειρά από μεθόδους για λειτουργίες CRUD (Create, Retrieve,

⁶⁸ <http://www.webdav.org/>

Update, and Delete) σε μία συλλογής στοιχείων οργανωμένων σε ιεραρχίες φακέλων. Το API χρησιμοποιείται από τη γραφική διεπαφή του Αποθετηρίου (Model Repository View), τον οδηγό εισαγωγής TIM (TIM Import Wizard), καθώς και το SLMToolbox.

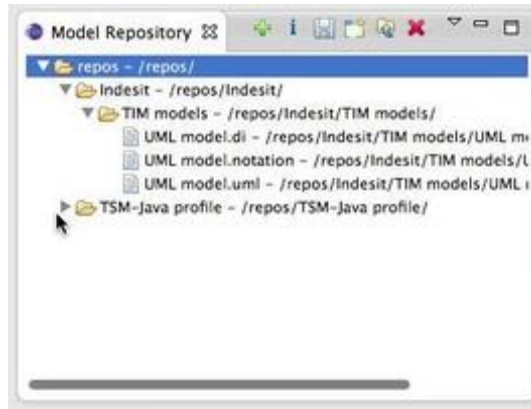
Οι βασικές λειτουργίες του API παρατίθενται στον ακόλουθο Πίνακα 10. Σημειώνεται ότι η εξαίρεση IOException είναι σφάλματα που αναφέρονται από το σύστημα αρχείων, ενώ η DavException υποδεικνύει αστοχία της υποδομής WebDAV.

Ορισμός Μεθόδου	Παράμετροι	Περιγραφή
void configure (ModelRepositoryConfiguration configuration);	Μία διαμόρφωση Εξυηρητητή Αποθετηρίου (δηλαδή διεύθυνση διακομιστή, θύρα, κ.α.)	Ενημερώνει τον πελάτη για τη διαμόρφωση του απομακρυσμένου εξυηρητητή Αποθετηρίου (διακομιστής, θύρα, διαδρομή κ.α.)
void createFolder (String path) throws DavException, IOException;	Διαδρομή προς τον φάκελο	Δημιουργεί έναν φάκελο στη συγκεκριμένη διαδρομή φακέλου
List<DavResource> listItems (String path) throws IOException, DavException;	Διαδρομή προς τον πόρο	Παραθέτει κατάλογο με τα «τέκνα» του πόρου στη διαδρομή{@code path}
void createFile (String path, File file, String contentType) throws IOException, DavException;	Διαδρομή προς τον φάκελο, Όνομα του αρχείου για ανάγνωση, τύπος περιεχομένου	Δημιουργεί ένα αρχείο ανοίγοντας μία ροή ανάγνωσης (filestream) προς των εξυηρητητή WebDAV.
void createFile (String path, byte[] binary, String contentType) throws IOException, DavException;	Διαδρομή προς το αρχείο, δυαδικά δεδομένα, τύπος δεδομένων	Δημιουργεί ένα αρχείο με τα δυαδικά δεδομένα της αντίστοιχης παραμέτρου.
void deleteResource (String path) throws IOException, DavException	Διαδρομή προς τον αποθηκευμένο πόρο	Διαγράφει έναν πόρο (φάκελο ή αρχείο).
void moveResource (String path, String newPath, boolean overwrite) throws IOException, DavException	Διαδρομή προέλευσης, διαδρομή προορισμού, παράμετρος αντικατάστασης υφιστάμενου πόρου	Μετακινεί έναν πόρο σε ένα νέο URI. Ένας πόρος υπάρχει και στο δεδομένο URI, και η παράμετρος αντικατάστασης (Overwrite) είναι "True", τότε αντικαθίσταται με το νέο. Αλλιώς αποτυγχάνει.
byte[] get (String path) throws Exception	Διαδρομή προς τον ζητούμενο πόρο στο αποθετήριο	Ανασύρει έναν πόρο από το αποθετήριο.
String getRootPath ();	Καμία	Παρέχει τη διαδρομή προς τη «ρίζα» του αποθετηρίου (π.χ. στο 'http://192.168.33.11/repos/', επιστρέφει "/repos/")

Πίνακας 10: API Πελάτη Αποθετηρίου

5.3.3.1.3 Model Repository View – Προβολή Αποθετηρίου Μοντέλων

Το Model Repository View (MR View) παρέχει την κύρια γραφική διεπαφή προς το Αποθετήριο Μοντέλων διαμέσου της Πλατφόρμας Ανάπτυξης. Η γραφική διεπαφή είναι προσβάσιμη διαμέσου της διεπαφής της Πλατφόρμας Ανάπτυξης και μπορεί να ενσωματωθεί μόνιμα στον χώρο εργασίας του χρήστη. Αφού εκκινήσει μία σύνδεση με τον εξυηρητητή του Αποθετηρίου Μοντέλων (διαμέσου του API Πελάτη Αποθετηρίου), παρουσιάζει μία όψη του Αποθετηρίου ως μία δομή «δένδρου» με ιεραρχικούς φακέλους και τα περιεχόμενά τους αρχεία. Οι χρήστες έτσι μπορούν να περιηγηθούν στη δενδροειδή δομή και να εκτελέσουν εργασίες με τους πόρους που περιέχει.



Σχήμα 32: Model Repository View - Γραφική Διεπαφή Αποθετηρίου

Συγκεκριμένα, η διεπαφή επιτρέπει τις ακόλουθες λειτουργίες, οι οποίες είναι προσβάσιμες διαμέσου των αντίστοιχων πλήκτρων, όπως φαίνεται στο Σχήμα 32 (από αριστερά προς δεξιά):

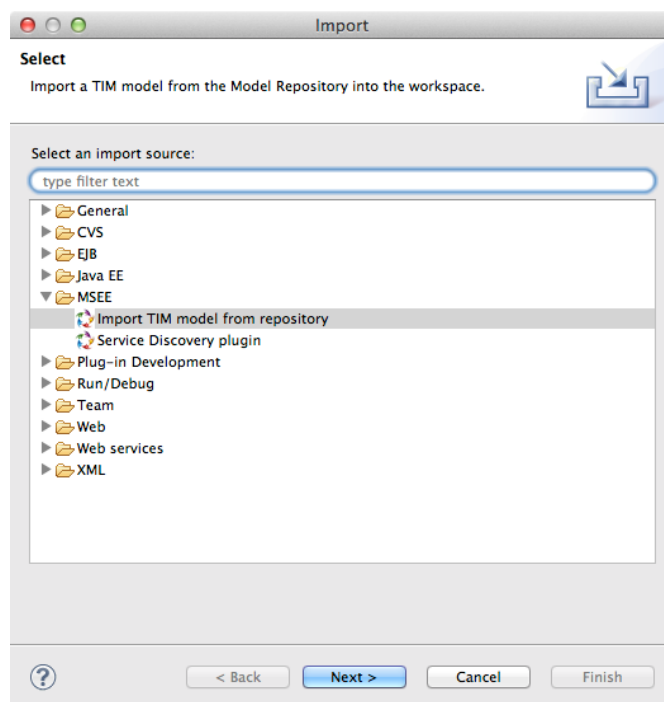
- *Σύνδεση στο Αποθετήριο Μοντέλων:* Το πρόσθετο συνδέεται με ένα Αποθετήριο Μοντέλων. Μόλις επιλεγεί, ζητείται από τον χρήστη να εισάγει τις λεπτομέρειες της σύνδεσης, όπως το όνομα του διακομιστή, τη θύρα σύνδεσης, την εσωτερική διαδρομή αρχείου στο Αποθετήριο (αν χρειάζεται) και πιστοποιητικά πρόσβασης (όνομα χρήστη και κωδικός). Μόλις γίνει η σύνδεση, το πρόσθετο παρουσιάζει τους φακέλους και τα αρχεία του Αποθετηρίου στη συγκεκριμένη διαδρομή φακέλου ως ένα διάγραμμα δένδρου (tree view).
- *Προβολή των ιδιοτήτων συγκεκριμένου πόρου ή συλλογής πόρων:* Ο χρήστης μπορεί να επιλέξει ένα αρχείο ή φάκελο και να διαβάσει τα μεταδεδομένα του.
- *Μεταφόρτωση πόρου από το Αποθετήριο:* Ο χρήστης μπορεί να επιλέξει ένα μοντέλο στην απεικόνιση δένδρου (tree view) και να το «κατεβάσει» στον χώρο εργασίας του Eclipse και το project που επεξεργάζεται εκείνη τη στιγμή.
- *Δημιουργία «συλλογής»:* Ο χρήστης μπορεί να δημιουργήσει μία «συλλογή» (δηλαδή έναν φάκελο), και να τη χρησιμοποιήσει για να ομαδοποιήσει αρχεία.
- *Ανάρτηση πόρου:* Ο χρήστης μπορεί να «ανεβάσει» έναν πόρο (μοντέλο, κώδικα ή άλλα δεδομένα) στο Αποθετήριο.
- *Διαγραφή πόρου:* Ο χρήστης μπορεί να διαγράψει έναν πόρο ή μία συλλογή από το Αποθετήριο.

5.3.3.1.4 Model Repository Import Wizard – Οδηγός Εισαγωγής Μοντέλων

Το πρόσθετο Προβολής Αποθετηρίου Μοντέλων προσφέρει μία διεπαφή γενικής χρήσης για πρόσβαση στο Αποθετήριο. Από την άλλη, ο Οδηγός Εισαγωγής Μοντέλων εστιάζει μόνο στην εισαγωγή μοντέλων TIM, με αποτελεσματικό τρόπο και με τον μικρότερο αριθμό βημάτων. Μπορεί να χρησιμοποιηθεί, για παράδειγμα, στη περίπτωση που ο χρήστης θέλει να έχει μόνο μία φορά πρόσβαση στο Αποθετήριο για να εισάγει ένα συγκεκριμένο μοντέλο, και στη συνέχεια να συνεχίσει να εργάζεται με αυτό. Ο οδηγός είναι προσβάσιμος απευθείας από το περιβάλλον χρήστη του Eclipse και χρησιμοποιεί το API Πελάτη Αποθετηρίου για να συνδεθεί με τον εξυπηρετητή Αποθετηρίου.

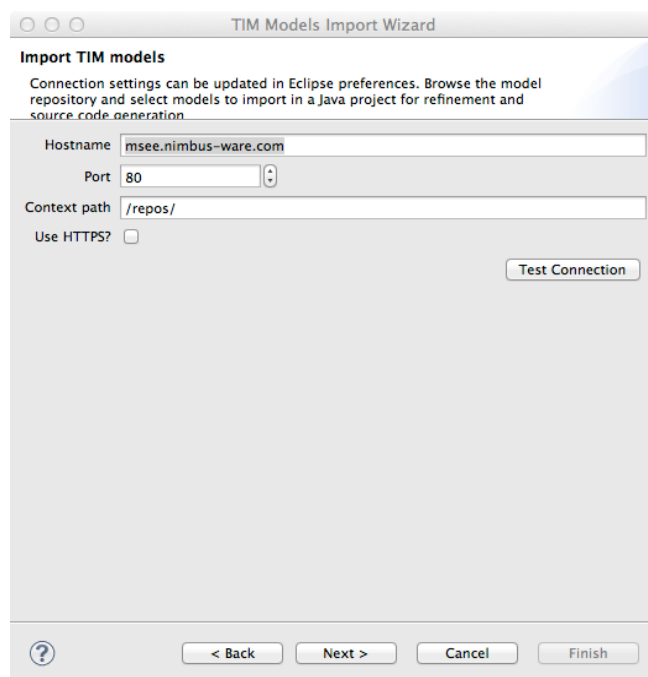
Η χρήση του Οδηγού περιλαμβάνει τα ακόλουθα τέσσερα βήματα:

Ο Οδηγός Εισαγωγής Μοντέλων επιλέγεται από το μενού Import του Eclipse (Σχήμα 33)



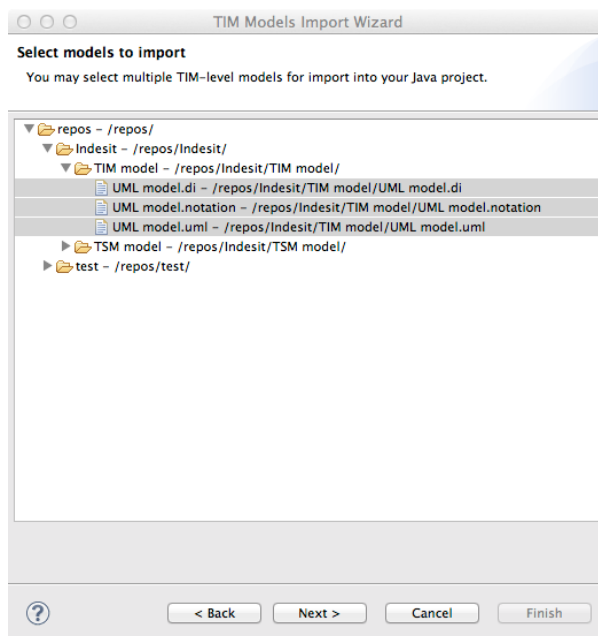
Σχήμα 33: Εκτέλεση του Οδηγού Εισαγωγής Μοντέλου

Εμφανίζεται ένα παράθυρο διαλόγου ζητώντας από τον χρήστη να εισάγει τα στοιχεία σύνδεσης, όπως φαίνεται στο Σχήμα 34. Αν τα στοιχεία δεν έχουν εισαχθεί στο παρελθόν, ο χρήστης καταχωρεί το όνομα διακομιστή του εξυπηρετητή του Αποθετηρίου, τη θύρα σύνδεσης, την εσωτερική διαδρομή αρχείου, όνομα και κωδικό χρήστη. Μόλις επιλέξει «Επόμενο», γίνεται η σύνδεση.



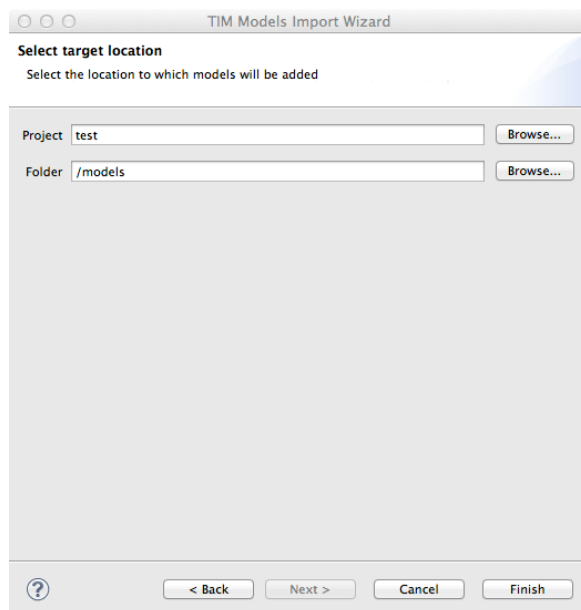
Σχήμα 34: Σύνδεση στο Αποθετήριο Μοντέλων

Ο χρήστης βλέπει μία δομή δένδρου (treeview) με τους διαθέσιμους πόρους και φακέλους (συλλογές πόρων), και επιλέγει τον πόρο ή πόρους που θέλει να εισάγει (Σχήμα 35). Στη συνέχεια επιλέγει «Επόμενο».



Σχήμα 35: Επιλογή Μοντέλων για Εισαγωγή

Ο χρήστης επιλέγει τον προορισμό εισαγωγής μέσα στον χώρο εργασίας του έργου (project), όπως φαίνεται στο Σχήμα 36. Μόλις ο χρήστης επιλέξει «Τέλος», το μοντέλο εισάγεται στον χώρο εργασίας της Πλατφόρμας Ανάπτυξης.



Σχήμα 36: Επιλογή προορισμού για την εισαγωγή μοντέλων

5.3.3.2 Ολοκλήρωση με την πλατφόρμα παροχής υπηρεσιών SOA (MSEE Service Delivery Platform)

Η Πλατφόρμα Παροχής Υπηρεσιών του MSEE (MSEE Service Delivery Platform) υποστηρίζει την εύρεση, ταξινόμηση, επιλογή, κλήση και παρακολούθηση των υπηρεσιών web στη πλατφόρμα ΤΠΕ του MSEE. Η επικοινωνία με την Πλατφόρμα Ανάπτυξης είναι αμφίδρομη και συμβαίνει σε δύο περιπτώσεις:

- Κατά την εγγραφή μίας νέας υπηρεσίας web στη Πλατφόρμα Παροχής Υπηρεσιών
- Κατά την εύρεση και εισαγωγή περιγραφών WSDL υπηρεσιών web από την Πλατφόρμα Ανάπτυξης κατά τη διαδικασία ανάπτυξης λογισμικού

Το πρόσθετο Ολοκλήρωσης με την Πλατφόρμα Παροχής Υπηρεσιών (Service Delivery Platform Plugin), που υλοποιήθηκε για τις ανάγκες της πρωτότυπης εφαρμογής, καλύπτει και τις δύο περιπτώσεις, με τη χρήση εξειδικευμένων γραφικών οδηγιών.

5.3.3.2.1 Οδηγός εγγραφής υπηρεσιών (Service registration wizard)

Οι δύο πλατφόρμες επικοινωνούν στα τελικά στάδια της διαδικασίας ανάπτυξης λογισμικού, όταν η ολοκληρωμένη υπηρεσία web εγκαθίσταται στους εξυπηρετητές παραγωγικής λειτουργίας και εγγράφεται στο μητρώο (Registry module) της Πλατφόρμας Παροχής Υπηρεσιών του MSEE.

Ο πηγαίος κώδικας που παράγεται στη Πλατφόρμα Ανάπτυξης εμπλουτίζεται και αναπτύσσεται με τη χρήση τυπικών εργαλείων και διαδικασιών ανάπτυξης λογισμικού (επεξεργασία, εντοπισμός σφαλμάτων, δημιουργία εκτελέσιμων αρχείων κ.α.). Μόλις ολοκληρωθεί η υπηρεσία web, εγκαθίσταται στον εξυπηρετητή παραγωγικής λειτουργίας με τη χρήση του πακέτου Eclipse WebTools (Eclipse WTP). Το πακέτο παράγει μία περιγραφή WSDL (Web Service Description Language) της υπηρεσίας web. Ο χρήστης μπορεί να προσθέσει επιπλέον σημασιολογικά στοιχεία, παράγοντας μία περιγραφή σύμφωνη με το πρότυπο SA-WSDL (Semantic Annotations for WSDL)⁶⁹, με σκοπό την διευκόλυνση της μελλοντικής αναζήτησης και εύρεσης της υπηρεσίας κατά τη διαδικασία ανάπτυξης (π.χ. εντάσσοντας την υπηρεσία σε συγκεκριμένες κατηγορίες).

Από το σημείο αυτό:

- Επιλέγεται ο Οδηγός Εγγραφής Υπηρεσιών από το μενού «Export» του Eclipse
- Ο οδηγός εμφανίζει ένα παράθυρο διαλόγου, το οποίο
 - Ζητά το URL της περιγραφής WSDL (με τα σημασιολογικά στοιχεία)
 - Ζητά την επιβεβαίωση του URL κλήσης της υπηρεσίας (Service Endpoint)
- Ο χρήστης εισάγει τις απαραίτητες πληροφορίες και επιλέγει «Τέλος»
- Το πρόσθετο Ολοκλήρωσης με την Πλατφόρμα Παροχής υπηρεσιών καλεί το API εγγραφής της Πλατφόρμας Παροχής Υπηρεσιών και ανεβάζει την περιγραφή της υπηρεσίας web.

⁶⁹ <http://www.w3.org/2002/ws/sawSDL/>

- Το πρόσθετο εμφανίζει μήνυμα επιτυχίας.

Η νέα υπηρεσίας έχει εγγραφεί στο μητρώο της Πλατφόρμας Παροχής Υπηρεσιών του MSEE και μπορεί πλέον να κληθεί από άλλους χρήστες ή εφαρμογές της πλατφόρμας ΤΠΕ του MSEE.

5.3.3.2.2 Οδηγός Εύρεσης Υπηρεσιών (Service discovery wizard)

Η Πλατφόρμα Ανάπτυξης παρέχει λειτουργικότητες που επιτρέπουν την αναζήτηση και εύρεση υπηρεσιών web εγγραμμένων στο μητρώο της Πλατφόρμας Παροχής Υπηρεσιών (Service Delivery Platform Registry Module). Αυτό επιτρέπει στους χρήστες την εισαγωγή κλήσεων προς τις συγκεκριμένες υπηρεσίες στον κώδικα των εφαρμογών που αναπτύσσουν.

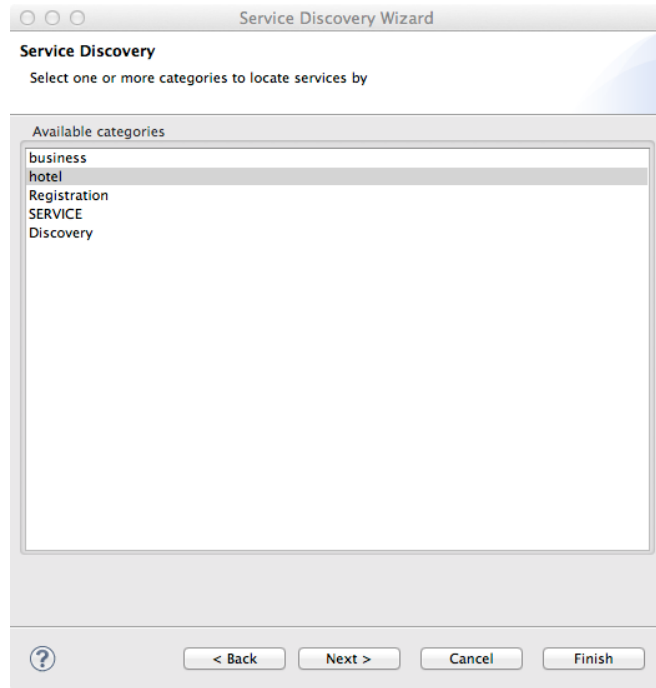
Συγκεκριμένα, οι χρήστες μπορούν να εισάγουν τις περιγραφές WSDL με σημασιολογικές προσθήκες (Semantically Annotated WSDL) των υπηρεσιών web της πλατφόρμας του MSEE. Με τη σειρά του, το εργαλείο Eclipse Web Tools μπορεί να δημιουργήσει κώδικα πελάτη σε Java, με βάση τις περιγραφές αυτές.

Για τη διευκόλυνση της διαδικασίας, το πρόσθετο ολοκλήρωσης με την Πλατφόρμα Παροχής Υπηρεσιών προσφέρει έναν γραφικό «Οδηγό Εύρεσης Υπηρεσιών» (Service Discovery Wizard), ο οποίος απλοποιεί τη διαδικασία της ανακάλυψης υφιστάμενων υπηρεσιών web στη πλατφόρμα ΤΠΕ του MSEE. Ο οδηγός εκμεταλλεύεται το API «Εύρεσης» (Discovery API) της Πλατφόρμας Παροχής Υπηρεσιών.

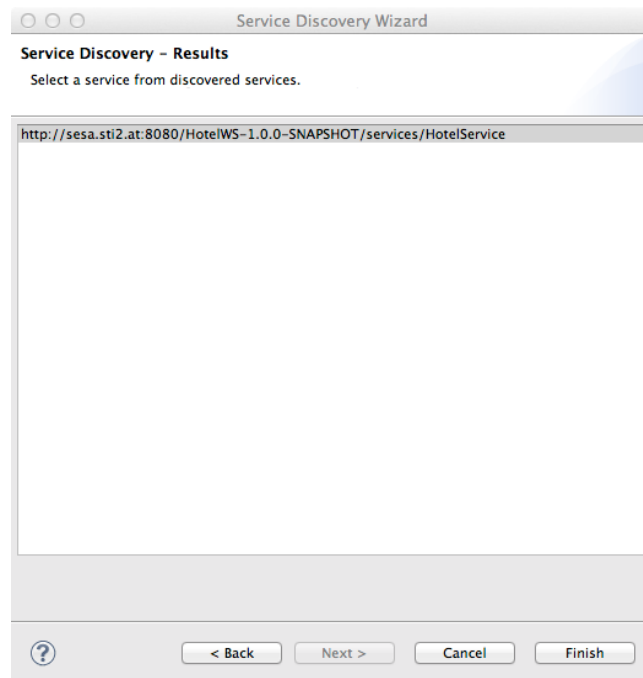
Η διαδικασία έχει τα ακόλουθα βήματα:

- Ο χρήστης επιλέγει τον Οδηγό Εύρεσης Υπηρεσιών από το μενού «Import» του Eclipse.
- Ο οδηγός καλεί το API Εύρεσης της Πλατφόρμας Παροχής Υπηρεσιών και κατεβάζει ένα κατάλογο των διαθέσιμων κατηγοριών υπηρεσιών, όπως φαίνεται στο Σχήμα 37.
- Ο οδηγός απεικονίζει τον κατάλογο στον χρήστη, και ο χρήστης επιλέγει μία από τις κατηγορίες υπηρεσιών, και πατά «Επόμενο»
- Ο οδηγός παρουσιάζει μία λίστα υπηρεσιών από την επιλεγμένη κατηγορία.
- Ο χρήστης επιλέγει μία από τις περιγραφές υπηρεσιών (με τη μορφή αρχείου WSDL) και επιλέγει «Επόμενο»
- Ο οδηγός ζητά μία τοποθεσία στον χώρο εργασίας του Eclipse για την εισαγωγή της περιγραφής.
- Μόλις εισαχθεί η πληροφορία, ο χρήστης επιλέγει «Τέλος»
- Η περιγραφή WSDL φορτώνεται στον χώρο εργασίας του χρήστη.

Μόλις η περιγραφή γίνει διαθέσιμη, οι μηχανικοί λογισμικού μπορούν να χρησιμοποιήσουν το πρόσθετο Eclipse Webtools για την αυτόματη παραγωγή κώδικα πελάτη για την υπηρεσία σε Java. Ο κώδικας αυτός μπορεί με τη σειρά του να εισαχθεί σε άλλες κλάσεις Java υπό ανάπτυξη. Έτσι, οι μηχανικοί λογισμικού μπορούν να αναχρησιμοποιήσουν υπηρεσίες web και να δημιουργήσουν συνθέσεις υπηρεσιών με προγραμματικό τρόπο, αντί να τις ορίζουν ως επιχειρησιακές διαδικασίες σε BPMN (οι οποίες πρέπει να εκτελεστούν από την αντίστοιχη μηχανή επιχειρησιακών διαδικασιών του Innovation Ecosystem Platform).



Σχήμα 37: Κατηγορίες υπηρεσιών στη Πλατφόρμα Παροχής Υπηρεσιών



Σχήμα 38: Περιγραφές υπηρεσιών συγκεκριμένης κατηγορίας

5.3.3.3 Ολοκλήρωση με το σύστημα IEP (Innovation Ecosystem Platform)

Το Innovation Ecosystem Platform παρέχει τις ψηφιακές υπηρεσίες στους τελικούς χρήστες, μέσω της εκτέλεσης των αντίστοιχων επιχειρησιακών διαδικασιών. Η διασύνδεση με το IEP αφορά στην αποστολή των εκτελέσιμων επιχειρησιακών διαδικασιών με τη μορφή αρχείων .BAR (Business Archive) και τη δημοσίευσή τους.

Για τη διευκόλυνση της διαδικασίας αυτής, η Πλατφόρμα Ανάπτυξης περιλαμβάνει έναν «Οδηγό εγκατάστασης» των επιχειρησιακών διαδικασιών:

- Αφού οι χρήστες ολοκληρώσουν την επεξεργασία και τον εμπλουτισμό του μοντέλου της επιχειρησιακής διαδικασίας σε BPMN και παραχθεί το εκτελέσιμο αρχείο .bar, μπορούν να επιλέξουν τη λειτουργία «Deploy Process to IEP» από το μενού πλαισίου του έργου.
- Ένα παράθυρο διαλόγου ζητά από τον χρήστη να επιβεβαιώσει το URL του API εγκατάστασης / δημοσίευσης επιχειρησιακών διαδικασιών του IEP. Οι χρήστες μπορούν να εισάγουν ένα εναλλακτικό URL εφόσον είναι απαραίτητο.
- Ο χρήστης επιλέγει «OK» και το πρόσθετο Ολοκλήρωσης με το IEP χρησιμοποιεί το API για να στείλει τη νέα επιχειρησιακή διαδικασία.
- Ένα παράθυρο μηνύματος επιβεβαιώνει την επιτυχία της διαδικασίας ή παρέχει σχετικά μηνύματα λάθους.
- Η επιχειρησιακή διαδικασία γίνεται διαθέσιμη στους χρήστες της πλατφόρμας MSEE διαμέσου του IEP.

5.3.3.4 Εγκατάσταση Υπηρεσίας Web και παραγωγή κώδικα πελάτη : πακέτο Eclipse WTP

Η Πλατφόρμα Ανάπτυξης χρησιμοποιεί τα εργαλεία για υπηρεσίες Web⁷⁰ από το πακέτο Eclipse Web Tools Platform ώστε να προσφέρει δύο κύριες λειτουργίες: την εγκατάσταση υπηρεσιών web στους αντίστοιχους εξυπηρετητές και την αυτόματη παραγωγή κώδικα πελάτη σε Java για την κλήση των υπηρεσιών αυτών.

5.3.3.4.1 Εγκατάσταση υπηρεσιών

Η λειτουργία περιλαμβάνει την εγκατάσταση υπηρεσιών web στους εξυπηρετητές παραγωγικής λειτουργίας και την παραγωγή των σχετικών περιγραφών «Semantically Annotated WSDL» που χρησιμοποιούνται κατά την εγγραφή και εύρεση υπηρεσιών από στην υποδομή SOA του MSEE. Αναλυτικότερα:

- Μόλις αναπτυχθούν στο MSEE, οι υπηρεσίες web εγκαθίστανται κανονικά σε έναν εξυπηρετητή παραγωγικής λειτουργίας, χρησιμοποιώντας το Eclipse WTP.
- Μία περιγραφή WSDL παράγεται αυτόματα κατά τη διαδικασία εγκατάστασης της υπηρεσίας.
- Ο μηχανικός λογισμικού επεξεργάζεται το αρχείο WSDL ώστε να προσθέσει σημασιολογικά στοιχεία όπως η «Κατηγορία» της υπηρεσίας, με σκοπό την υποστήριξη της αναζήτησης κατά το στάδιο ανάπτυξης άλλων εφαρμογών. Το αποτέλεσμα είναι μία περιγραφή SA-WSDL.

5.3.3.4.2 Παραγωγή κώδικα υπηρεσιών (client code) σε Java για την κλήση υπηρεσιών web του MSEE

Τη δημιουργία πλαισίων κώδικα πελάτη σε Java, βασισμένα στις περιγραφές SA-WSDL των υπηρεσιών, κατά τη διάρκεια ανάπτυξης άλλων εφαρμογών. Μόλις εισαχθεί μία περιγραφή SA-WSDL από την Πλατφόρμα Παροχής Υπηρεσιών του MSEE, το πακέτο Eclipse Web Service

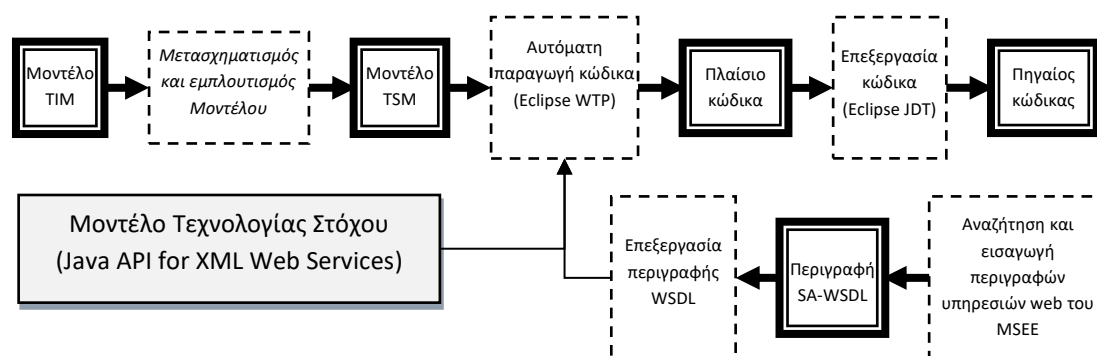
⁷⁰ <https://eclipse.org/webtools/ws/>

Tools μπορεί να δημιουργήσει κώδικα κλήσης της υπηρεσίας web. Ο κώδικας αποτελεί απλώς ένα υποτυπώδες πλαίσιο, το οποίο, μπορεί, στη συνέχεια, να συμπληρωθεί κατάλληλα από τον μηχανικό λογισμικού.

Η προσθήκη αυτών των αναφορών σε υφιστάμενες υπηρεσίες web αποτελεί μία ειδική υπό-περίπτωση της Δομικής ροής εργασίας, η οποία οδηγεί από περιγραφές WSDL σε εκτελέσιμο κώδικα. Η ροή ακολουθεί τα παρακάτω βήματα:

- Εισάγεται μία περιγραφή SA-WSDL μέσω των λειτουργιών αναζήτησης και εισαγωγής της Πλατφόρμας Ανάπτυξης (που περιγράφηκαν στις προηγούμενες ενότητες)
- Η περιγραφή SA-WSDL φορτώνεται στον επεξεργαστή WSDL (Eclipse WTP), και ο χρήστης την επεξεργάζεται εφόσον είναι απαραίτητο.
- Ο χρήστης αιτείται την παραγωγή εκτελέσιμου κώδικα πελάτη από την περιγραφή της συγκεκριμένης υπηρεσίας.
- Η Πλατφόρμα Ανάπτυξης παράγει τον κώδικα πελάτη: κώδικα Java για web services (Web Service Stub) βασισμένο στο API της Java για Web Services σε XML (Java API for XML Web Services/JAX-WS⁷¹)

Οι χρήστες μπορούν στη συνέχεια να επεξεργαστούν τον κώδικα πελάτη και να τον προσθέσουν απευθείας στον κώδικα που παράγεται από τη Δομική ροή εργασίας (κλάσεις UML προς Java). Η διαδικασία απεικονίζεται στο Σχήμα 39 ως κλάδος της γενικής ροής:



Σχήμα 39: Ροή Εργασίας SA-WSDL προς κώδικα πελάτη Java

5.4 Αξιολόγηση

5.4.1 Προσέγγιση αξιολόγησης

Πριν τη φάση της υλοποίησης, μία αρχιτεκτονική λογισμικού μπορεί να αξιολογηθεί χρησιμοποιώντας διάφορες ποιοτικές ή ποσοτικές μεθόδους, οι οποίες μετρούν συγκεκριμένες ιδιότητες. Αυτές μπορεί να περιλαμβάνουν μη-λειτουργικά χαρακτηριστικά του συστήματος, όπως αξιοπιστία, ευελιξία ή ευκολία συντήρησης. Οι ποιοτικές μέθοδοι χρησιμοποιούν «ερωτηματικές» τεχνικές, όπου οι εμπλεκόμενοι (π.χ. χρήστες, διαχειριστές κ.α.) συζητούν σχετικά με τις αρχιτεκτονικές και τα αποτελέσματα αναλύονται με βάση ένα «ημί-τυπικό» πλαίσιο (Dobrica & Niemela, 2002). Για παράδειγμα, μέθοδοι όπως η SAAM

⁷¹ <http://docs.oracle.com/javase/5/tutorial/doc/bnayl.html>

(Software Architecture Analysis Method) και η PASA (Performance Assessment of Software Architecture), εφαρμόζουν την αξιολόγηση των χαρακτηριστικών του λογισμικού από τους εμπλεκόμενους σε σχέση με και συγκεκριμένα σενάρια χρήσης (Babar & Gorton, 2004).

Από την άλλη, οι «μετρητικές» τεχνικές βασίζονται σε ποσοτικές μετρήσεις συγκεκριμένων χαρακτηριστικών, και εφαρμόζουν μία τυπική προσέγγιση με σκοπό τον υπολογισμό προκαθορισμένων δεικτών. Για παράδειγμα, οι αντικειμενοστραφείς αρχιτεκτονικές μπορούν να αξιολογηθούν με βάση δείκτες που έχουν οριστεί με τυπικό τρόπο (όπως η «σύζευξη μεταξύ κλάσεων» και η «έλλειψη συνοχής των μεθόδων») (Chidamber & Kemerer, 1994). Οι υπολογιζόμενοι δείκτες μπορούν να χρησιμοποιηθούν για να προβλέψουν χαρακτηριστικά όπως την ποιότητα του λογισμικού (Basili & Briand, 1996) ή την ασφάλειά του (Chowdhury & Zulkernine, 2011).

Ωστόσο, η έρευνα πάνω στα συστήματα λογισμικού μπορεί να αξιολογηθεί και σε σχέση με τι υποστηρίζει ότι προσφέρει παραπάνω από την υφιστάμενη κατάσταση της έρευνας (Munzner, 2006). Οι Ellis και Dix (2006) επιχειρηματολόγησαν ότι ο στόχος της αξιολόγησης πρέπει να είναι το αν και πότε το σύστημα λειτουργεί ή είναι χρήσιμο στην επίτευξη του στόχου του. Αυτό σημαίνει ότι η αξιολόγηση γίνεται σε σχέση με το πρόβλημα το οποίο δηλώνει ότι αντιμετωπίζει.

Στη περίπτωση της Πλατφόρμας Ανάπτυξης και της προτεινόμενης μεθοδολογίας, η έρευνα έχει ως έναυσμα επιχειρησιακές (και όχι τεχνικές) ανάγκες. Η επιχειρησιακή ανάγκη που αντιμετωπίζουν είναι η επιτυχής ανάπτυξη λογισμικού για την ενσωμάτωση ψηφιακών υπηρεσιών σε βιομηχανικά προϊόντα, μέσω της τεχνικής υποστήριξης της διαδικασίας αυτής. Έτσι, αποφασίστηκε ότι η αξιολόγηση του συστήματος θα πρέπει να εστιάζει στο πως τα νέα εργαλεία και τρόποι εργασίας επηρέασαν την ανάπτυξη λογισμικού σε τρεις βασικές διαστάσεις:

- Τη συνεργασία της ομάδας ανάπτυξης των υπηρεσιών
- Τη παραγωγικότητα των μηχανικών λογισμικού
- Τη ποιότητα του παραγόμενου κώδικα

Οι παραπάνω διαστάσεις επιλέχθηκαν καθώς σχετίζονται με τις επιχειρησιακές ανάγκες που περιγράφηκαν στις προηγούμενες ενότητες.

Η συνεργασία είναι απαραίτητο στοιχείο για την ανάπτυξη συστημάτων υπηρεσιών, καθώς απαιτείται ο συντονισμός και επικοινωνία πολλαπλών εμπλεκόμενων (μηχανικών λογισμικού, αναλυτών, ειδικών του πεδίου εφαρμογής κ.α.). Η παραγωγικότητα των μηχανικών λογισμικού σχετίζεται με τον χρόνο που χρειάζεται για την υλοποίηση της εφαρμογής με βάση τις προδιαγραφές και μοντέλα που έχουν παραλάβει από προηγούμενα στάδια της διαδικασίας ανάπτυξης. Τέλος, η ποιότητα του κώδικα σχετίζεται με δύο παραμέτρους: τον χρόνο που απαιτείται για την διόρθωση σφαλμάτων ώστε η εφαρμογή να περάσει σε παραγωγική λειτουργία, καθώς και με την ποιότητα της τελικής υπηρεσίας που παρέχει.

Οι πιλοτικές εφαρμογές του ερευνητικού έργου MSEE παρείχαν μία ευκαιρία για την εκτίμηση της συνεισφοράς της προτεινόμενης μεθοδολογίας και των εργαλείων σε αυτές τις

διαστάσεις. Σε γενικές γραμμές, οι πιλότοι εστίαζαν στις στρατηγικές, επιχειρησιακές και τεχνικές πλευρές της σύλληψης, ανάπτυξης και υλοποίησης καινοτόμων υπηρεσιών σε Συστήματα Προϊόντος – Υπηρεσίας στη βιομηχανία. Με σκοπό την υποστήριξη των πιλότων, οι εταίροι του έργου σχεδίασαν και υλοποίησαν τα βασικά στοιχεία μίας πλατφόρμας ΤΠΕ για την παροχή υπηρεσιών. Η πλατφόρμα του MSEE βασίστηκε στις αρχές της υπηρεσιοστραφούς αρχιτεκτονικής (SOA) και της διαχείρισης επιχειρησιακών διαδικασιών (BPM). Η πλατφόρμα περιλάμβανε εργαλεία για τον σχεδιασμό, ανάπτυξη και παροχή υπηρεσιών σε συνεργάτες και πελάτες υπηρεσιοποιημένων βιομηχανικών εταιρειών. Τα εργαλεία αυτά περιλάμβαναν την πρωτότυπη Πλατφόρμα Ανάπτυξης και τα στοιχεία του συστήματος με τα οποία διαλειτουργεί.

Ανάλογα με τα ειδικά χαρακτηριστικά της ανάπτυξης του Συστήματος Προϊόντος – Υπηρεσίες σε κάθε πιλότο, η Πλατφόρμα Ανάπτυξης χρησιμοποιήθηκε από μηχανικούς λογισμικού των συμμετεχόντων βιομηχανικών επιχειρήσεων ή άλλους εταίρους του έργου. Συνεπώς, οι συγκεκριμένοι εμπλεκόμενοι θα μπορούσαν να αξιολογήσουν τη συνεισφορά της προτεινόμενης μεθοδολογίας και των εργαλείων με βάση τις εμπειρίες από τη χρήση τους και τη γνώση τους σχετικά με το βιομηχανικό πεδίο εφαρμογής ή την ανάπτυξη λογισμικού.

Αποφασίστηκε η συλλογή της εξαγωγή της αξιολόγησης μέσω ημι-δομημένων συνεντεύξεων. Η προσέγγιση αυτή επιλέχθηκε καθώς υπάρχουν ενδογενείς δυσκολίες στην αξιολόγηση της Ολοκληρωμένων Περιβάλλοντων Ανάπτυξης. Υπάρχει σημαντική πολυπλοκότητα στις αλληλεπιδράσεις μεταξύ των χαρακτηριστικών του χρήστη, των συστημάτων και των πεδίων εφαρμογής (Budgen & Thomson, 2003). Τα τυποποιημένα ερωτηματολόγια, σε αυτό το περιβάλλον, δεν θα ήταν ικανά να αποτυπώσουν ολοκληρωμένα αυτές τις περίπλοκες αλληλεπιδράσεις, πέρα από τις συνηθισμένες προκλήσεις που θέτει η αξιολόγηση Περιβαλλόντων Ανάπτυξης (Kline & Seffah, 2005). Η κατάσταση περιπλέκεται από το ότι η Πλατφόρμα Ανάπτυξης εισάγει νέες μεθόδους και αλληλεπιδράσεις με τα άλλα στάδια της ροής ανάπτυξης υπηρεσιών. Τα αποτελέσματα του φαινομένου αυτού μπορεί να είναι απρόβλεπτα και να μην λαμβάνονται υπόψη από τυποποιημένες μεθόδους. Για να κατανοηθεί ο αντίκτυπος ενός πληροφοριακού συστήματος σε ένα οργανισμό και τους εμπλεκόμενούς του, πρέπει να λάβουμε υπόψη την πολυπλοκότητα των αλληλεπιδράσεων σε περιβάλλοντα του πραγματικού κόσμου. Σε αυτές τις περιπτώσεις, οι ερμηνευτικές και ποιοτικές μέθοδοι έρευνας μπορούν να προσφέρουν νέες οπτικές γωνίες, αναδεικνύοντας θέματα και μεταβλητές που δεν περίμεναν οι ερευνητές (Galliers, 1990). Στη περίπτωση της Πλατφόρμας Ανάπτυξης, μία ημι-δομημένη προσέγγιση θα μπορούσε να αντλήσει γνώση από τη γνώση των συμμετεχόντων στα βιομηχανικά πεδία εφαρμογής ή στην ανάπτυξη λογισμικού.

5.4.2 Διαδικασία αξιολόγησης

Οι τεχνικοί πόροι που αναπτύχθηκαν ως αποτέλεσμα της έρευνας στο MSEE εφαρμόστηκαν στις τέσσερις πιλοτικές εφαρμογές Συστήματος Προϊόντος – Υπηρεσίας του έργου. Η πρωτότυπη Πλατφόρμα Ανάπτυξης χρησιμοποιήθηκε σε τρεις από αυτές. Οι πιλοτικές εφαρμογές της Πλατφόρμας Ανάπτυξης αφορούσαν:

- Λογισμικό για τηλεοπτικούς δέκτες Smart TV
- Κατασκευή προσωποποιημένων ενδυμάτων

- Υπηρεσία απομακρυσμένης παρακολούθησης και συντήρησης πλυντηρίων ρούχων.

Στις δύο πρώτες περιπτώσεις, η Πλατφόρμα Ανάπτυξης δοκιμάστηκε παράλληλα με παραδοσιακές μεθόδους για την ανάπτυξη λογισμικού υπηρεσιών. Η Πλατφόρμα Ανάπτυξης εφαρμόστηκε στην υλοποίηση υπηρεσιών web, στη σύνθεση υπηρεσιών web και στοιχείων διεπαφής χρήστη.

Στη τρίτη περίπτωση, η Πλατφόρμα Ανάπτυξης χρησιμοποιήθηκε ως μέρος μίας πρωτότυπης «Πλατφόρμας Ανάπτυξης Κινητών Εφαρμογών» (Mobile Development Platform). Στη περίπτωση αυτή, η Πλατφόρμα Ανάπτυξης συνέβαλλε στη δημιουργία του φορητού τμήματος της υπηρεσίας παρακολούθησης και συντήρησης.

Στη συνέχεια, ζητήθηκε από εννέα μηχανικούς λογισμικού από τους εταίρους του έργου να αξιολογήσουν τη συμβολή της Πλατφόρμας Ανάπτυξης στα πλαίσια ημι-δομημένων συνεντεύξεων. Άτομα που συνέβαλλαν στον σχεδιασμό ή την υλοποίηση της Πλατφόρμας Ανάπτυξης αποκλείστηκαν από τη μελέτη. Οι συμμετέχοντες ερωτήθηκαν σχετικά με τις εμπειρίες τους σχετικά με την Πλατφόρμα μέσω τηλεφώνου ή διαδικτυακής τηλεδιάσκεψης. Οι συνεντεύξεις ήταν ένας-προς-έναν και διήρκεσαν περίπου 20-30 λεπτά η κάθε μία.

Στην αρχή της συζήτησης, οι συμμετέχοντες ενημερώθηκαν για το γενικό σκοπό και εύρος της μελέτης (δηλαδή «Η αξιολόγηση της Πλατφόρμας Ανάπτυξης ως εργαλείο για την υλοποίηση συστημάτων υπηρεσιών στη βιομηχανία»). Στη συνέχεια, ερωτήθηκαν τρεις, ανοικτές, ερωτήσεις:

- E1: «Πως επηρέασε η Πλατφόρμα Ανάπτυξης τη συνεργασία κατά την ανάπτυξη υπηρεσιών;»
- E2: «Πως επηρέασε η Πλατφόρμα Ανάπτυξης την παραγωγικότητα του μηχανικού λογισμικού και τον χρόνο ανάπτυξης του λογισμικού;»
- E3: «Πως επηρέασε η Πλατφόρμα Ανάπτυξης την ποιότητα του παραγόμενου κώδικα;»

Ο ερευνητής παρείχε ελάχιστη καθοδήγηση κατά τη συνέντευξη (π.χ. διευκρινίσεις ή επαναφορά στο θέμα της συζήτησης) και οι συμμετέχοντες μπορούσαν να διατυπώσουν τις απόψεις τους ελεύθερα.

5.4.3 Αποτελέσματα

Όσο αφορά στην E1, οι συμμετέχοντες ανέφεραν ότι η ροή εργασιών με μοντέλα της Πλατφόρμας Ανάπτυξης βελτίωσε την επικοινωνία με τους σχεδιαστές της υπηρεσίας, καθώς και με άλλο προσωπικό που εμπλέκεται στην υλοποίηση του συστήματος υπηρεσιών. Αυτό αποδόθηκε στους παρακάτω παράγοντες:

- Διάφορα αρχιτεκτονικά στοιχεία υποστήριζαν την καλύτερη συνεργασία και επικοινωνία. Το Αποθετήριο Μοντέλων προσέφερε έναν κοινό χώρο εργασίας με τους σχεδιαστές των υπηρεσιών. Η διασύνδεση με το περιβάλλον εκτέλεσης των εφαρμογών (τους εξυπηρετητές ή τη μηχανή εκτέλεσης των επιχειρησιακών διαδικασιών) εξομάλυνε τον βρόχο ανάδρασης που αποτελείται από την ανάπτυξη λογισμικού, την αξιολόγησή του από τα επιχειρησιακά στελέχη, τις βελτιώσεις στο

σχεδιασμό της υπηρεσίας, τις τροποποιήσεις των εφαρμογών και την εγκατάσταση των νέων εκδόσεών τους στο περιβάλλον λειτουργίας.

- Η διαδικασία της MDSEA τυποποιεί τη διαδικασία αποτύπωσης απαιτήσεων και της μετάδοσής τους για τους πολλαπλούς συμμετέχοντες που εμπλέκονται. Ταυτόχρονα επιβάλλει ένα πλαίσιο συνεργασίας μέσω της χρήσης κοινών συμβάσεων (μοντέλων, γλώσσας και ορολογίας).
- Τα μοντέλα TIM και TSM εκφράζονται στην ίδια γλώσσα (UML, BPMN) σε όλα τα επίπεδα. Οι σχεδιαστές μοντέλων στο επίπεδο TIM μπορούν να καταλάβουν τα προτεινόμενα μοντέλα TSM. Από την πλευρά τους, οι μηχανικοί λογισμικού μπορούν να υποδείξουν αδυναμίες στις έννοιες και ιδέες που εκφράζουν τα μοντέλα TIM. Έτσι και οι δύο κατηγορίες εμπλεκόμενων μπορούν να συνεισφέρουν ουσιαστικά στο τελικό σχεδιασμό του λογισμικού υπηρεσιών.
- Αποτυπώνονται τα δομικά και συμπεριφορικά στοιχεία του συστήματος υπηρεσιών μέσα στην ίδια διαδικασία και σύνολο εργαλείων. Οι σχεδιαστές των υπηρεσιών και οι μηχανικοί λογισμικού μπορούν έτσι να δουλεύουν πάνω σε μία πληρέστερη εικόνα του συστήματος υπηρεσιών.
- Τα προηγούμενα στάδια της μεθοδολογίας MDSEA καλύπτουν πολλαπλές πλευρές του συστήματος υπηρεσιών, στις οποίες περιλαμβάνονται ζητήματα στρατηγικής, λειτουργίας και διοίκησης των τελικών υπηρεσιών. Παρά την εστίαση στα τμήματα ΤΠΕ του συστήματος υπηρεσιών, οι σχεδιαστικές «είσοδοι» για τους χρήστες της Πλατφόρμας Ανάπτυξης είναι το αποτέλεσμα αυτής της «ολιστικής» προσέγγισης. Αυτό συμβάλλει στη καλύτερη ευθυγράμμιση μεταξύ επιχειρησιακών στόχων και των προϊόντων λογισμικού που τους εξυπηρετούν.
- Τέλος, η χρήση τυπικής μοντελοποίησης υποστηρίζει την τεκμηρίωση και «ανιχνευσιμότητα» (traceability) της διαδικασίας ανάπτυξης. Τα μοντέλα TIM και TSM είναι μία αφηρημένη απεικόνιση του υλοποιούμενου λογισμικού, στην οποία είναι εύκολο να γίνουν αναφορές ή να παρουσιαστεί σε άλλα μέλη της ομάδας του συστήματος υπηρεσιών.

Σχετικά με τις E2 και E3, οι συμμετέχοντες παρατήρησαν ότι δεν υπήρχαν αξιοσημείωτες επιδράσεις στη παραγωγικότητα του μηχανικού λογισμικού και την ποιότητα του παραγόμενου κώδικα. Αυτό συνέβη παρά την βελτιωμένη επικοινωνία και την προσθήκη εργαλείων για την υποβοήθηση της εργασίας, όπως η γεννήτρια κώδικα, η εύρεση υπηρεσιών και οι μηχανισμοί για την κλήση τους, η εύκολη εγκατάσταση των υπηρεσιών web ή των διαδικασιών στη πλατφόρμα εκτέλεσης κ.α. Έτσι, διαπιστώθηκε ότι η βελτιωμένη επικοινωνία σχετικά με τις απαιτήσεις δεν επηρέασε σημαντικά τον αριθμό των σφαλμάτων. Επίσης, ο χρόνος υλοποίησης δεν ήταν μικρότερος σε σχέση με τις παραδοσιακές μεθόδους. Ακόμα και τα εργαλεία παραγωγικότητας που περιλαμβάνει η Πλατφόρμα Ανάπτυξης, μεγάλο μέρος των εργασιών ακόμα στηρίζονται σε προσωπική προσπάθεια του χρήστη. Για παράδειγμα, ο μετασχηματισμός «TIM-TSM-εκτελέσιμο αρχείο» δεν μπορεί εύκολα να αυτοματοποιηθεί πλήρως. Από τη μία πλευρά, κάθε βήμα απαιτεί την προσθήκη επιπλέον πληροφορίας και λεπτομερειών στα μοντέλα, ενώ από την άλλη, τα παραγόμενα πλαίσια κώδικα πρέπει να συμπληρωθούν χειροκίνητα.

Αναλυτικότερα, το φαινόμενο αυτό αποδόθηκε στους εξής παράγοντες:

- Ο αντικειμενοστραφής κώδικας που παράγεται από τη γεννήτρια περιλαμβάνει άδεια πλαίσια κώδικα για κλάσεις, πεδία και μεθόδους από το μοντέλο TSM. Από το σημείο αυτό, οι μηχανικοί λογισμικού εργάζονται χρησιμοποιούν παραδοσιακές μεθόδους ανάπτυξης λογισμικού. Η συνεισφορά της αυτόματης παραγωγής κώδικα δεν αρκούσε για την ουσιαστική μείωση του χρόνου ανάπτυξης.
- Η MDSEA δεν καλύπτει ένα πλήρες εύρος από λειτουργικές και μη-λειτουργικές απαιτήσεις. Για παράδειγμα, στη δομική ροή εργασίας δεν υπάρχει συγκεκριμένη υποστήριξη για περιγραφές της συμπεριφοράς των δομικών στοιχείων. Οι απαιτήσεις αυτές έπρεπε να μεταδοθούν, να συζητηθούν και να αποτυπωθούν έξω της διαδικασίας της MDSEA.
- Όλα τα στοιχεία για τις συγκεκριμένες εφαρμογές (συμπεριλαμβανομένων μοντέλα, έγγραφα σχεδιασμού, κώδικας, υπηρεσίες web κ.α.) δημιουργήθηκαν εξ' αρχής για τις πιλοτικές εφαρμογές του έργου. Ωστόσο, μεγάλο μέρος της αναμενόμενης προστιθέμενης αξίας της Πλατφόρμας Ανάπτυξης βρίσκεται στην επαναχρησιμοποίηση πόρων – όπως για παράδειγμα στη βελτίωση συστημάτων ή εφαρμογών υπηρεσιών. Ωστόσο, αυτό θα μπορούσε να αξιολογηθεί μόνο στα πλαίσια μακροπρόθεσμων πιλοτικών εφαρμογών με πολλούς κύκλους αναθεώρησης και βελτίωσης του συστήματος υπηρεσιών.
- Ο μικρός χρόνος των πιλότων είχε ως αποτέλεσμα να αναλώνεται μόνο μικρό μέρος του χρόνου ανάπτυξης στην εισαγωγή, αναζήτηση και εγκατάσταση στοιχείων λογισμικού. Για παράδειγμα, η σημασιολογική αναζήτηση υπηρεσιών web είναι χρήσιμη όταν υπάρχουν πολλές διαθέσιμες υπηρεσίες και κατηγορίες υπηρεσιών. Ωστόσο, οι υπηρεσίες που παρήγαγε ο κάθε πιλότος ήταν πολύ λίγες ώστε να υπάρξουν σημαντικές ωφέλειες από την αντίστοιχη λειτουργικότητα της Πλατφόρμας Ανάπτυξης

6 Συμπεράσματα και Προοπτικές

6.1 Συμπεράσματα

Στη τρέχουσα βιβλιογραφία, οι Τεχνολογίες Πληροφορικής και Επικοινωνιών έχουν αναγνωριστεί ως κεντρικοί πυλώνες των σύγχρονων Συστημάτων Προϊόντος – Υπηρεσίας, και ένα σημαντικό στοιχείο της παροχής προηγμένων υπηρεσιών. Ωστόσο, έχουν υπάρξει πολύ λίγες προσπάθειες για τη διερεύνηση του τρόπου σχεδίασης και ανάπτυξης των απαραίτητων υποδομών ΤΠΕ γενικότερα και του λογισμικού τους ειδικότερα. Τα υφιστάμενα ερευνητικά αποτελέσματα περιορίζονται σε απλές αναφορές στην ύπαρξη ή την ανάγκη υλοποίησης υποδομών ΤΠΕ σε γενικά πλαίσια για την ανάπτυξη Συστημάτων Προϊόντος – Υπηρεσίας.

Το πλαίσιο της ανάπτυξης λογισμικού σε αυτές τις συνθήκες δεν έχει διερευνηθεί και δεν υπάρχουν εξειδικευμένες μεθοδολογίες ή υποστηρικτικά εργαλεία για την ανάπτυξη λογισμικού στην υπηρεσιοποιημένη βιομηχανία. Ως συνέπεια, τέτοια εργαλεία δεν έχουν ποτέ αξιολογηθεί σε πραγματικές ή, έστω, σε πειραματικές συνθήκες. Συνεπώς, το πεδίο αυτό είναι μία τελείως ανεξερεύνητη περιοχή του πεδίου της ανάπτυξης λογισμικού, αλλά και του διεπιστημονικού πεδίου των σύγχρονων Συστημάτων Προϊόντος – Υπηρεσίας.

Ως απάντηση στο συγκεκριμένο ερευνητικό κενό, η διατριβή αυτή διερευνά το πλαίσιο υλοποίησης λογισμικού για την ενσωμάτωση προηγμένων ψηφιακών υπηρεσιών σε βιομηχανικά προϊόντα. Με βάση τα συμπεράσματα από αυτή τη μελέτη, διατυπώνει τις απαιτήσεις για μεθοδολογίες και τα εργαλεία που θα τις υποστηρίζουν. Στη συνέχεια, προτείνει μία κατάλληλη μεθοδολογία ανάπτυξης λογισμικού με μοντέλα, βασισμένη στη προσέγγιση MDSEA για την ανάπτυξη συστημάτων υπηρεσιών. Η MDSEA προέρχεται από την μεθοδολογία λογισμικού MDA, άρα η προτεινόμενη μεθοδολογία βασίζεται σε έννοιες και εργασίες της MDA. Με σκοπό την υποστήριξη της προτεινόμενης μεθόδου, αναπτύχθηκε η αρχιτεκτονική ενός κατάλληλου Ολοκληρωμένου Περιβάλλοντος Ανάπτυξης λογισμικού, η Πλατφόρμα Ανάπτυξης. Η Πλατφόρμα Ανάπτυξης λειτουργεί στα πλαίσια μίας ευρύτερης υποδομής ΤΠΕ για την παροχή υπηρεσιών ενσωματωμένων σε βιομηχανικά προϊόντα. Για τον λόγο αυτό, πέρα από τις λειτουργίες μετασχηματισμού μοντέλων και ανάπτυξης κώδικα, η Πλατφόρμα Ανάπτυξης περιλαμβάνει μηχανισμούς διαλειτουργικότητας με άλλα μέρη του συστήματος ΤΠΕ. Αυτοί παρέχουν πρόσβαση σε κοινούς πόρους, λειτουργίες εγκατάστασης, εγγραφής, εύρεσης και σύνθεσης υπηρεσιών SOA, ενεργοποίηση επιχειρησιακών διαδικασιών κ.α.

Η μεθοδολογία και η προτεινόμενη Πλατφόρμα Ανάπτυξης αξιολογήθηκαν στα πλαίσια των πιλότων του ερευνητικού έργου MSEE, σε συνεργασία με βιομηχανικούς εταίρους του έργου. Συλλέχθηκαν παρατηρήσεις από τους εμπλεκόμενους μηχανικούς λογισμικού, οι οποίες παρείχαν δύο κύρια ευρήματα. Κατά πρώτον, η μεθοδολογία και τα εργαλεία βελτίωσαν την επικοινωνία μεταξύ των σχεδιαστών υπηρεσιών και των μηχανικών λογισμικού. Η προστιθέμενη αξία προερχόταν από την επιβολή κοινής γλώσσας επικοινωνίας για όλους τους εμπλεκόμενους, κάτι που βοήθησε την ευθυγράμμιση των επιχειρησιακών στόχων με τις απαιτήσεις του λογισμικού, επέτρεψε την αμφίδρομη επικοινωνία μεταξύ σχεδιαστών και προγραμματιστών, ενώ βελτίωσε την τεκμηρίωση του λογισμικού και την ανιχνευσιμότητα (traceability) κατά τη διαδικασία ανάπτυξης.

Από την άλλη, ο χρόνος που απαιτήθηκε για την ανάπτυξη του νέου λογισμικού και η ποιότητα του παραγόμενου κώδικα ήταν αμετάβλητα σε σχέση με τις συμβατικές μεθόδους και εργαλεία. Εντοπίστηκαν ορισμένοι παράγοντες που συνέβαλλαν, σε μεγαλύτερο ή μικρότερο βαθμό, σε αυτό το αποτέλεσμα. Σε αυτούς περιλαμβάνονται η μικρή ποσότητα του αυτόματα παραγόμενου κώδικα, η σύντομη διάρκεια των πιλοτικών εφαρμογών, και το μικρό εύρος των απαιτήσεων που καλύπτει η μεθοδολογία.

Τα αποτελέσματα αυτά υποδεικνύουν ότι η προτεινόμενη μεθοδολογία και τα εργαλεία θα μπορούσαν να προσφέρουν ωφέλειες εφόσον εφαρμοστούν στο «διεπιστημονικό» περιβάλλον της ανάπτυξης λογισμικού για βιομηχανικά Συστήματα Προϊόντος – Υπηρεσίας. Η αποδεδειγμένη βελτίωση της ποιότητας της συνεργασίας και της επικοινωνίας των εμπλεκόμενων διευκολύνει την ανάπτυξη λογισμικού των ψηφιακών υπηρεσιών – χωρίς όμως απαραίτητα να μειώνει δραματικά τον χρόνο ανάπτυξης (όπως φάνηκε από την αξιολόγηση).

Η προστιθέμενη αξία εντοπίζεται σε πολλά σημεία. Οι ομάδες ανάπτυξης συστημάτων προϊόντος – υπηρεσίας στη βιομηχανία προέρχονται από διαφορετικά πεδία. Ειδικά για τον σχεδιασμό και την ανάπτυξη λογισμικού, εμπλέκονται τρεις γενικές κατηγορίες στελεχών: διοικητικά – εμπορικά στελέχη (για τον σχεδιασμό του συστήματος υπηρεσίας), ειδικοί του πεδίου εφαρμογής (π.χ. μηχανικοί παραγωγής, στελέχη γραμμής που θα διοικούν ή θα υλοποιούν υπηρεσίες) και μηχανικοί λογισμικού. Οι εμπλεκόμενοι αυτοί χρησιμοποιούν διαφορετικές ορολογίες και αντιλαμβάνονται το σύστημα υπηρεσιών από διαφορετικές σκοπιές. Το σύστημα τυποποιεί τη διαδικασία συλλογής και επεξεργασίας των απαιτήσεων λογισμικού, δίνει εργαλεία για την αποτύπωση δομικών και συμπεριφορικών πλευρών του συστήματος, παρέχει κοινή πρόσβαση σε μοντέλα για όλους τους εμπλεκόμενους, και επιβάλλει τη χρήση μίας «κοινής γλώσσας» μοντελοποίησης. Άρα:

- Μειώνει την ανάγκη για διευκρινίσεις από τους σχεδιαστές του συστήματος υπηρεσιών, και για την «ερμηνεία» των προδιαγραφών του λογισμικού από τους μηχανικούς λογισμικού.
- Τυποποιεί τη μεθοδολογία συλλογής και ανάλυσης απαιτήσεων, προδιαγράφοντας σαφή στάδια και «παραδοτέα» για τους εμπλεκόμενους (π.χ. μοντέλα διαφορετικών επιπέδων αφαίρεσης, κώδικά, εκτελέσιμα αρχεία κ.α.)
- Επιτρέπει στους εμπλεκόμενους να ανατρέξουν ανά πάσα στιγμή σε κοινώς αποδεκτά και διαθέσιμα σχεδιαστικά έγγραφα του συστήματος υπηρεσιών (μοντέλα), αλλά και να τα αναθεωρήσουν εφόσον απαιτείται.
- Βελτιώνει την τελική τεκμηρίωση του συστήματος, διευκολύνοντας τη συντήρηση και πιθανές μελλοντικές τροποποιήσεις.
- Όλα τα παραπάνω έχουν σαν αποτέλεσμα τη μείωση του πνευματικού φόρτου των εμπλεκόμενων. Τα τεχνικά σφάλματα του κώδικα (bugs) μπορεί να μην επηρεάζονται (όπως φάνηκε από την αξιολόγηση). Ωστόσο άλλα σφάλματα αποτρέπονται, όπως π.χ. προβλήματα συνεννόησης μεταξύ σχεδιαστών και μηχανικών λογισμικού, αναντιστοιχία με την επιθυμητή λειτουργία των υπηρεσιών ή ελλείψεις στη τεκμηρίωση του συστήματος.

Πέρα από την διευκόλυνση της ανάπτυξης λογισμικού, η εφαρμογή των προτεινόμενων μεθόδων και εργαλείων μπορεί να έχει και στρατηγικές ωφέλειες. Οι σχεδιαστές του συστήματος υπηρεσιών βρίσκονται περισσότερο κοντά στην επιχειρησιακή λογική του συστήματος, καθώς και την γενικότερη στρατηγική της επιχείρησης που υλοποιεί. Ως αποτέλεσμα, η αρχική αφαιρετική αποτύπωση του επιθυμητού συστήματος υπηρεσιών είναι, σε μεγάλο βαθμό, σύμφωνη με τον επιχειρησιακό σχεδιασμό, π.χ. με τη στρατηγική μάρκετινγκ, τους δείκτες και στόχους απόδοσης, τη συνολική στρατηγική αλλά και με τις επιμέρους λειτουργίες που εμπλέκονται στην υλοποίηση του συστήματος υπηρεσιών. Ωστόσο, κατά τον μετασχηματισμό της αφαιρετικής αποτύπωσης σε αναλυτικές τεχνικές προδιαγραφές, πολλά από τα παραπάνω στοιχεία μπορεί να χαθούν ή να αλλοιωθούν, ως αποτέλεσμα της «ερμηνείας» που θα δώσουν οι αναλυτές και μηχανικοί λογισμικού. Η προτεινόμενη μεθοδολογία κάνει χρήση κοινής γλώσσας και μοντέλων που διατηρούνται και εμπλουτίζονται στα διάφορα βήματα, συμβάλλοντας στην λογική συνέχεια μεταξύ των λιγότερο και περισσότερο αφαιρετικών απεικονίσεων. Με τον τρόπο αυτό, συμβάλλει στην σωστή υλοποίηση των επιχειρησιακών λειτουργιών από το λογισμικό και την ευθυγράμμιση του με την γενικότερη επιχειρησιακή στρατηγική.

Για παράδειγμα, έστω μία βιομηχανική επιχείρηση που παράγει πλυντήρια ρούχων. Η εταιρεία επιθυμεί να προσφέρει ένα Σύστημα Προϊόντος – Υπηρεσίας. Συγκεκριμένα, ο πελάτης αγοράζει το πλυντήριο, και έχει την επιλογή να εγγραφεί σε μία συνδρομητική υπηρεσία προληπτικής συντήρησης. Η υπηρεσία περιλαμβάνει αφορά στην απομακρυσμένη παρακολούθηση του πλυντηρίου μέσω αισθητήρων, τη συλλογή δεδομένων για την λειτουργική του κατάσταση και την παροχή δωρεάν προληπτικής συντήρησης / επισκευών εφόσον η χρήση βρίσκεται στα πλαίσια της αναμενόμενης οικιακής χρήσης. Γίνεται άμεσα επικοινωνία με τον πελάτη, και κλείνεται ραντεβού με τεχνικό από τρίτη εταιρεία που εξυπηρετεί την περιοχή.

Ο επιχειρησιακός σχεδιασμός του Συστήματος Προϊόντος – Υπηρεσίας ξεκινά από τα διοικητικά στελέχη της βιομηχανικής εταιρείας. Στη συνέχεια, στελέχη άλλων τμημάτων και εξειδικεύσεων συμμετέχουν στην ανάπτυξη των πλευρών του συστήματος υπηρεσιών. Ενδεικτικά, οι μηχανικοί προϊόντος και παραγωγής πρέπει να εξετάσουν τον σχεδιασμό και παραγωγή του προϊόντος ώστε να υποστηρίξει την υπηρεσία (π.χ. αισθητήρες και τρόπος επικοινωνίας). Στελέχη πωλήσεων και προμηθειών θα οργανώσουν τον τρόπο με τον οποίο θα παρέχεται η υπηρεσία και τις συνεργασίες με τρίτους όπως εξωτερικά συνεργεία τεχνικών, τους προμηθευτές αναλωσίμων, και άλλους εμπλεκόμενους. Τα στελέχη της μηχανοργάνωσης θα πρέπει να εξετάσουν τις υποδομές ΤΠΕ που απαιτούνται για την υλοποίηση της υπηρεσίας. Ο συντονισμός και η επικοινωνία για τον σχεδιασμό του συστήματος υπηρεσιών περιλαμβάνει και εξωτερικούς εμπλεκόμενους. Χρειάζεται, για παράδειγμα, να συζητηθεί με τους εξωτερικούς τεχνικούς συνεργάτες ο τρόπος με τον οποίο θα κλείνονται τα ραντεβού συντήρησης στην περιοχή ευθύνης τους, και ποια δεδομένα θα χρειάζονται ή θα καταχωρούν οι τεχνικοί κατά την επισκευή. Ο σχεδιασμός αυτός αντιστοιχεί στις προδιαγραφές και απαιτήσεις για το λογισμικό που θα υποστηρίξει το σύστημα υπηρεσιών. Αυτό συνεπάγεται ότι οι εμπλεκόμενοι στο σύστημα υπηρεσιών θα συμμετέχουν στον καθορισμό των απαιτήσεων του λογισμικού άμεσα ή έμμεσα. Σε κάποιες περιπτώσεις θα διατυπώσουν οι ίδιοι τις προδιαγραφές (π.χ. οι μηχανικοί που θα περιγράψουν το πρωτόκολλο επικοινωνίας με τους αισθητήρες του πλυντηρίου), ενώ σε άλλες περιπτώσεις

θα δώσουν τις απαραίτητες πληροφορίες για τη διαμόρφωση συγκεκριμένων απαιτήσεων (π.χ. τα στελέχη πωλήσεων, εξωτερικοί συνεργάτες, πελάτες και άλλοι χρήστες). Οι τελικές προδιαγραφές και ο σχεδιασμός του λογισμικού μπορεί να διαμορφωθεί εσωτερικά στην εταιρεία, αν το λογισμικό πρόκειται να παραχθεί εσωτερικά. Πολλές, φορές όμως, το λογισμικό υπηρεσιών υλοποιείται από μία άλλη εταιρεία, με εξειδίκευση στα επιχειρησιακά συστήματα, έτσι η ανάλυση απαιτήσεων, ο σχεδιασμός και η ανάπτυξη του συστήματος γίνονται από τρίτους.

Τυπικά, η συλλογή των προδιαγραφών γίνεται μέσα από τις συζητήσεις των εμπλεκόμενων μερών με τους υπεύθυνους για τον σχεδιασμό του συστήματος λογισμικού. Οι εμπλεκόμενοι βλέπουν συχνά το σύστημα από την ιδιαίτερή τους οπτική γωνία, με αποτέλεσμα να χρειάζεται η ερμηνεία μετάφραση των αναγκών τους σε συνεκτικές προδιαγραφές. Τις περισσότερες φορές, οι προδιαγραφές αυτές είναι, τελικά, λεκτικές περιγραφές της επιθυμητής λειτουργικότητας και των χαρακτηριστικών του συστήματος. Οι αναλυτές και οι μηχανικοί λογισμικού πρέπει να βασιστούν σε αυτές για να προχωρήσουν στον σχεδιασμό και ανάπτυξη του λογισμικού υπηρεσιών. Αν υποθέσουμε ότι αυτοί ανήκουν σε μία άλλη εταιρεία με ειδίκευση στο λογισμικό, είναι πολύ πιθανό να μην γνωρίζουν τις λεπτομέρειες και τα ιδιαίτερα χαρακτηριστικά του πεδίου εφαρμογής (δηλαδή τον κλάδο των πλυντηρίων), ή τις συνθήκες και τη λειτουργία της βιομηχανικής επιχείρησης – πελάτη.

Η εφαρμογή της προτεινόμενης μεθοδολογίας και του εργαλείου αλλάζει τον τρόπο που διεξάγεται η ανάλυση απαιτήσεων και ο σχεδιασμός του συστήματος. Αφού καθοριστεί το επιχειρηματικό μοντέλο του συστήματος υπηρεσιών (εμπλεκόμενοι, δραστηριότητες, κανάλια, ροές εσόδων, δομή κόστους, πόροι κ.α.), αναπτύσσονται αρχικά δομικά και συμπεριφορικά μοντέλα του συστήματος υπηρεσιών: «ποιος εμπλέκεται» (συστήματα λογισμικού, μηχανήματα, άνθρωποι κ.α.) και «τι συμβαίνει» (διαδικασία παροχής υπηρεσίας, διοίκησης συστήματος κ.α.). Η αρχική μοντελοποίηση έχει σαν στόχο την αποτύπωση της επιχειρησιακής λογικής του συστήματος υπηρεσιών σε μία σειρά από κεντρικά σχεδιαστικά έγγραφα. Τα μοντέλα περιγράφουν π.χ. από πού συλλέγονται τα δεδομένα, τους βασικούς επιχειρησιακούς κανόνες για τον εντοπισμό βλαβών, τον τρόπο τιμολόγησης και εξόφλησης, τα εμπλεκόμενα μέρη και τους ρόλους τους.

Τα αφαιρετικά αυτά μοντέλα χρησιμοποιούνται κατά τις συζητήσεις με τους εμπλεκόμενους ως η βάση για τον σχεδιασμό του συστήματος. Οι εμπλεκόμενοι στη διαδικασία σχεδιασμού (εσωτερικοί και εξωτερικοί) αναμένεται να κατανοούν τουλάχιστον τα μέρη των αφαιρετικών μοντέλων που τους αφορούν, και θα έχουν την ευκαιρία να εκφράσουν τη γνώμη τους ή να προτείνουν αλλαγές σε αυτά. Έτσι, οι μηχανικοί παραγωγής θα μπορούν να εκφράσουν την άποψή τους για το πότε και πως συλλέγονται τα δεδομένα συντήρησης, και με ποια διαδικασία διαπιστώνεται η βλάβη, οι τεχνικοί για τον τρόπο με τον οποίο πρέπει να κλειστεί ένα ραντεβού συντήρησης κ.α. Η συλλογή των δεδομένων μπορεί να περιλαμβάνει συναντήσεις και επικοινωνία με ανεξάρτητους εμπλεκόμενους, ή τη διεξαγωγή συζητήσεων «στρογγυλής τράπεζας» ή παρουσιάσεων με εκπρόσωπους των εμπλεκόμενων στο σύστημα. Σε όλες τις περιπτώσεις, τα σχέδια των μοντέλων του συστήματος υπηρεσιών χρησιμοποιούνται ως βάση για τη συζήτηση και επικοινωνία. Το τελικό μοντέλο (ή μοντέλα) σε αυτό το στάδιο αντιστοιχεί στο Business Service Model (BSM) της μεθοδολογίας MDSEA.

Στη συνέχεια, το μοντέλο αυτό μετασχηματίζεται και εξειδικεύεται ώστε να αναπαριστά τις υποδομές ΤΠΕ του συστήματος υπηρεσιών από τους αναλυτές συστήματος της εταιρείας λογισμικού. Και πάλι, η συζήτηση για το «τι περιλαμβάνει» και το «τι κάνει» το σύστημα υπηρεσιών συλλέγει δεδομένα από όλους τους εμπλεκόμενους για την περαιτέρω ανάπτυξη των προδιαγραφών του συστήματος από την οπτική γωνία των υποδομών πληροφορικής (λογισμικού και υλικού). Έτσι, δημιουργούνται αναλυτικότερα δομικά και συμπεριφορικά μοντέλα, τα οποία αντιστοιχούν στο επίπεδο Technology-Independent Model (TIM) της μεθοδολογίας MDSEA, τα οποία διατηρούν τη βασική επιχειρησιακή λογική του συστήματος υπηρεσιών αλλά ενσωματώνουν πληροφορίες και την οπτική γωνία των ετερογενών εμπλεκόμενων του συστήματος. Τα μοντέλα πλέον περιγράφουν τι θα κάνει το λογισμικό (συλλογή δεδομένων, εφαρμογή επιχειρησιακών κανόνων, ενημέρωση πελάτη κ.α.) και την εμπλοκή εξωτερικών οντοτήτων (π.χ. την καταχώρηση δεδομένων από το προσωπικό της εταιρείας και τους πελάτες, τη διασύνδεση με άλλα συστήματα λογισμικού, κ.α.), και τις σχέσεις μεταξύ τους.

Τα μοντέλα TIM παραλαμβάνονται από τους μηχανικούς λογισμικού, οι οποίοι ξεκινούν τον μετασχηματισμό τους σε μοντέλα τύπου Technology Specific Model (TSM). Έτσι, εφαρμόζεται η διαδικασία εμπλουτισμού όπως περιεγράφηκε στις προηγούμενες ενότητες: τα μοντέλα TIM ενημερώνονται με τεχνικά στοιχεία και τροποποιούνται ώστε να αντιστοιχούν στις τεχνολογίες που θα χρησιμοποιηθούν για την υλοποίηση του συστήματος.

Στις παραδοσιακές μεθοδολογίες ανάπτυξης, οι μηχανικοί λογισμικού θα έπρεπε να υλοποιήσουν τον κώδικα απευθείας από τις λεκτικές απαιτήσεις και προδιαγραφές των προηγούμενων σταδίων, ουσιαστικά δημιουργώντας την δική τους «ερμηνεία» για τις απαιτήσεις του συστήματος. Ωστόσο, στη προτεινόμενη μεθοδολογία, οι μηχανικοί εργάζονται, μέσω του εργαλείου ανάπτυξης, πάνω σε μοντέλα που έχουν ήδη συμφωνηθεί και γίνει αποδεκτά από τους εμπλεκόμενους στο σχεδιασμό του συστήματος υπηρεσιών. Κατά συνέπεια, η σύνδεση με τα προηγούμενα στάδια σχεδιασμού (και με την γενική επιχειρησιακή λογική) είναι καλύτερη.

Επιπλέον, η χρήση κοινών μοντέλων και γλώσσας με τους άλλους εμπλεκόμενους διευκολύνει την λύση προβλημάτων και την ανταλλαγή πληροφορίας, καθώς η επικοινωνία συνεχίζεται και κατά τη φάση ανάπτυξης. Οι χρήστες και άλλοι εμπλεκόμενοι του συστήματος (π.χ. τελικοί χρήστες, εξωτερικοί συνεργάτες, μηχανικοί παραγωγής, και στελέχη πωλήσεων) θα κληθούν να δώσουν διευκρινίσεις ή να τροποποιήσουν μέρη του επιχειρησιακού σχεδιασμού. Μπορεί, για παράδειγμα, να χρειάζονται επιπλέον συζήτηση με τους μηχανικούς παραγωγής πάνω στον αλγόριθμο εντοπισμού βλαβών, αν, π.χ. δεν είναι επαρκώς σαφείς οι κανόνες ή δεν είναι εφικτό να γίνει με τον αναμενόμενο τρόπο. Στη περίπτωση αυτή, η χρήση κοινής γλώσσας και μοντέλων συμβάλλει στην συνεννόηση πλευρών με διαφορετικό υπόβαθρο.

Στο τέλος της ανάπτυξης, οι εξωτερικοί εμπλεκόμενοι θα αξιολογήσουν τις πρωτότυπες εκδόσεις του λογισμικού και θα πιστοποιήσουν την επιχειρησιακή του ετοιμότητα προτείνοντας ταυτόχρονα διορθώσεις και βελτιώσεις. Σε αυτή την περίπτωση, οι αλλαγές στα επιχειρησιακά μοντέλα (BSM/TSM) οδηγούν σε αντίστοιχες τροποποιήσεις των τεχνικών

μοντέλων (TSM) και του παραγόμενου κώδικα διατηρώντας την συνέπεια μεταξύ των σχεδιαστικών εγγράφων του συστήματος λογισμικού.

Επίσης, λόγω της τήρησης συνεπών μοντέλων BSM, TIM και TSM η μεθοδολογία ενθαρρύνει τη συστηματική τεκμηρίωση του σχεδιασμού του λογισμικού στα επιχειρησιακά και τεχνικά επίπεδα. Η τεκμηρίωση είναι απαραίτητη, καθώς, κατά την παραγωγική λειτουργία, το σύστημα ενημερώνεται και τροποποιείται, οδηγώντας στην επανάληψη του κύκλου συλλογής απαιτήσεων, σχεδιασμού, υλοποίησης, πιστοποίησης και παραγωγικής λειτουργίας. Τα μοντέλα χρησιμοποιούνται εκ νέου ως βάση για την επικοινωνία των εμπλεκόμενων, για τον σχεδιασμό των τροποποιήσεων και για την υλοποίησή τους. Έτσι, για παράδειγμα, η επιχείρηση αποφασίζει την προσθήκη μίας επιπλέον υπηρεσίας για την παράδοση αναλωσίμων (όπως απορρυπαντικά) με βάση τα δεδομένα του συστήματος επιτήρησης στα πλαίσια του υφιστάμενου συστήματος υπηρεσιών. Εφόσον υπάρχει αναλυτική τεκμηρίωση, μπορούν τα υφιστάμενα μοντέλα να ενημερωθούν ώστε να περιλαμβάνουν τη νέα υπηρεσία (με νέους εμπλεκόμενους και διαδικασίες), και να παραδοθούν στους μηχανικούς λογισμικού για την ανάπτυξη των απαραίτητων υποδομών ΤΠΕ. Μάλιστα, μέσω της χρήσης τεχνολογιών SOA και BPM, η νέα υπηρεσία μπορεί να αναχρησιμοποιήσει υπηρεσίες λογισμικού και λειτουργίες της υφιστάμενης υπηρεσίας (π.χ. το web service της τιμολόγησης, ή τη διεπαφή web για την επικοινωνία με τον πελάτη και τη διαχείριση των συνδρομών του). Τέλος, σε όλη τη διαδικασία ανάπτυξης, η ύπαρξη των σαφών και κοινά αποδεκτών οροσήμων (μοντέλων BSM, TIM, TSM και εκτελέσιμων προϊόντων) συμβάλλει στον συντονισμό των εμπλεκόμενων στη διαδικασία ανάπτυξης του λογισμικού.

6.2 Προοπτικές

Από επιστημονικής άποψης, εντοπίζεται ένας αριθμός από κατευθύνσεις για μελλοντική έρευνα στο πεδίο της ανάπτυξης λογισμικού για ΣΠΥ στο πεδίο των βιομηχανικών προϊόντων με βάση την προτεινόμενη μεθοδολογία και αρχιτεκτονική εργαλείων.

Μία πρώτη γραμμή έρευνας θα περιλάμβανε τη διερεύνηση μεθόδων για την περίληψη ενός μεγαλύτερου εύρους λειτουργικών απαιτήσεων, καθώς και μη-λειτουργικών απαιτήσεων. Καθώς οι βιομηχανικοί οργανισμοί δραστηριοποιούνται συνήθως σε καλά ορισμένα πεδία εφαρμογής, μπορεί να υπάρχει η δυνατότητα μείωσης του φόρτου εργασίας κατά το χειροκίνητο εμπλουτισμό μοντέλων TIM. Αυτό μπορεί να επιτευχθεί, ενδεχομένως, μέσω της αναχρησιμοποίησης πόρων επιπέδου TSM (π.χ. προ-εμπλουτισμένες κλάσεις), της χρήσης εξειδικευμένων προφίλ UML ή εξειδικευμένων μεταμοντέλων για συγκεκριμένα μεταμοντέλα για συγκεκριμένα βιομηχανικά πεδία. Για παράδειγμα, το μεταμοντέλο υπηρεσιών που προτείνει η USDL (Unified Service Description Language) (Cardoso, Barros, May, & Kylaou, 2010) θα μπορούσε να εξεταστεί ως πλαίσιο για μοντέλα επιπέδου TSM. Όσο αφορά σε συγκεκριμένα πεδία εφαρμογής, χαρακτηριστικό παράδειγμα αποτελεί το μεταμοντέλο HL7 (Health Level Seven International, 2017), για τον χώρο της υγείας.

Επιπλέον, οι δομικές περιγραφές ενός πεδίου εφαρμογής ή ενός συστήματος υπηρεσίας μπορούν να βοηθήσουν στην ανάπτυξη σχημάτων βάσεων δεδομένων. Έτσι θα μπορούσαν να γίνουν προσαρμογές στη συγκεκριμένη ροή εργασίας ή την δημιουργία ενός τρίτου κλάδου (πέρα από τη δομική και συμπεριφορική ροή), η οποία θα εστίαζε στα δεδομένα του

συστήματος υπηρεσιών. Ένας τέτοιος κλάδος θα μπορούσε, βάση του μοντέλου των δεδομένων να παράγεται τον απαραίτητο κώδικα (π.χ. σε SQL στη περίπτωση των Βάσεων Δεδομένων) για τη δημιουργία των κατάλληλων δομών δεδομένων.

Μία άλλη προσέγγιση θα μπορούσε να περιλαμβάνει την αναθεώρηση της μεθοδολογίας ώστε να απομακρυνθεί από τα γραμμικά μοντέλα ανάπτυξης λογισμικού τύπου «καταρράκτη», καθώς η MDSEA και η MDA προϋποθέτουν την εφαρμογή τέτοιων γραμμικών μεθόδων ανάπτυξης. Έτσι, θα μπορούσε να διερευνηθεί η προσαρμογή της μεθοδολογίας ώστε να υποστηρίζει επαναληπτικές ή άλλες ευέλικτες μεθόδους όπως η Agile. Κάτι τέτοιο θα απαιτούσε την ενσωμάτωση τεχνικών για τη διαχείριση αλλαγών και τη διάδοσή τους αντίστροφα στην αλυσίδα μοντέλων (δηλαδή από λιγότερο προς περισσότερο αφηρημένα μοντέλα). Σημειώνεται ότι τα προβλήματα του «ground-trip engineering» και της συνέπειας των μοντέλων αποτελούν τη στιγμή αυτή σημαντικά ερευνητικά ζητήματα στο πεδίο της ανάπτυξης λογισμικού με μοντέλα.

Ακόμη, η προσέγγιση που περιγράφεται στη παρούσα διατριβή βασίζεται στη χρήση αντικειμενοστραφούς προγραμματισμού και μεθόδων μοντελοποίησης και διαχείρισης επιχειρησιακών διαδικασιών. Η διεύρυνση των τεχνολογιών – στόχου (π.χ. λειτουργικών γλωσσών προγραμματισμού) θα απαιτούσε τη διερεύνηση διαφορετικών συμβάσεων μοντελοποίησης και τρόπων μετασχηματισμού.

Τέλος, μία περισσότερο μακροχρόνια αξιολόγηση της μεθόδου και των εργαλείων θα απέδιδε περισσότερο χρήσιμα συμπεράσματα. Θα μείωνε την επίδραση του αρχικού χρόνου προσαρμογής των χρηστών. Η ανάγκη για συνεχή βελτίωση του συστήματος υπηρεσιών και του σχετικού λογισμικού θα απαιτούσε την αναχρησιμοποίηση και ενημέρωση πόρων, άρα και την χρήση των αντίστοιχων εργαλείων διαλειτουργικότητας της Πλατφόρμας Ανάπτυξης.

Όσο αφορά στην εφαρμογή των προτεινόμενων μεθοδολογιών, υπάρχουν ενδείξεις ότι η υπηρεσιοποίηση στη βιομηχανία θα γίνεται περισσότερο έντονη, οδηγώντας στη προσφορά Συστημάτων Προϊόντων – Υπηρεσίας με βιομηχανικά προϊόντα συνδυασμένα με ψηφιακές υπηρεσίες.

Πολλές βιομηχανικές επιχειρήσεις ήδη προσφέρουν πρόσθετες υπηρεσίες: Για παράδειγμα, στην Ευρωπαϊκή Ένωση, έχει ήδη καταγραφεί αύξηση μεταξύ του 2000 και του 2014 στο μερίδιο των υπηρεσιών που προσφέρονται από βιομηχανικές επιχειρήσεις. Σε ένα δείγμα 10 χωρών στην ΕΕ, το ποσοστό των υπηρεσιοποιημένων βιομηχανικών Μικρομεσαίων Επιχειρήσεων (ΜΜΕ) κυμαίνεται μεταξύ 30%-70%. Στην ίδια μελέτη, εκτιμάται ότι το 2018 περίπου 10% των εργαζόμενων στις βιομηχανικές ΜΜΕ στην ΕΕ απασχολούνται στην παροχή Συστημάτων Προϊόντος – Υπηρεσίας. Η υπηρεσιοποίηση αύξησε το τζίρο κατά 1%-10%, για τις βιομηχανικές ΜΜΕ, με αποτέλεσμα περίπου το 12% του τζίρου τους να προέρχεται από υπηρεσίες. Τέλος προβλέπεται αύξηση του τζίρου που προέρχεται από την υπηρεσιοποίηση κατά 5% από το 2018 έως το 2025 (Study on the potential of servitisation and other forms of product-service provision for EU SMEs, 2018).

Ταυτόχρονα, φαίνεται ότι οι επιχειρήσεις συνειδητοποιούν τη σημασία των υποδομών ΤΠΕ για την εφαρμογή υπηρεσιοποίησης και την προσφορά συστημάτων προϊόντος – υπηρεσίας. Από αυτές που παρέχουν ήδη συνδυασμούς προϊόντος - υπηρεσίας, η έλλειψη κατάλληλων

υποδομών ΤΠΕ θεωρείται «σημαντικός» (39%) ή «πολύ σημαντικός» (31%) φραγμός στην υπηρεσιοποίηση. Σε αυτές που δεν παρέχουν, τα ποσοστά είναι 28% και 35%. (Study on the potential of servitisation and other forms of product-service provision for EU SMEs, 2018). Ταυτόχρονα, φαίνεται ότι οι επιχειρήσεις, στα πλαίσια της ψηφιακής υπηρεσιοποίησης, εγείρουν ζητήματα συνεργασίας των εμπλεκόμενων, κοινής χρήσης εργαλείων ΤΠΕ και εξασφάλισης των απαραίτητων δεξιοτήτων ΤΠΕ. (Sklyar, Kowalkowski, Tronvoll, & Sörhammar, 2019).

Οι δυνατότητες των ΤΠΕ για την υπηρεσιοποίηση της βιομηχανίας μπορούν να χρησιμοποιηθούν σε δύο διαστάσεις, «εσωτερική/ back-end» και «εξωτερική/ front-end» ψηφιοποίηση. Στην πρώτη, αναπτύσσονται εσωτερικές δυνατότητες (π.χ. ευέλικτη γραμμή παραγωγής) για την παροχή προηγμένων βιομηχανικών υπηρεσιών (π.χ. παραγωγή προσωποποιημένων προϊόντων). Στη δεύτερη αναπτύσσονται εξωτερικές δυνατότητες (π.χ. νέοι τρόποι επικοινωνίας, αλληλεπίδρασης και εμπορίας) για τη δημιουργία νέων εμπορικών υπηρεσιών (π.χ. μίσθωση προϊόντων, ηλεκτρονικές πληρωμές, απομακρυσμένη παρακολούθηση προϊόντων με αισθητήρες κ.α.). Η ενσωμάτωση ψηφιακών (βιομηχανικών και εμπορικών) υπηρεσιών στα προϊόντα μπορεί να γίνεται κατά προτεραιότητα στη μία ή στην άλλη διάσταση, ή ακόμα και συνδυασμένα. Στη περίπτωση αυτή βελτιώνονται και συνδυάζονται δυνατότητες back-end και front-end παρέχοντας ολοκληρωμένα καινοτόμα Συστήματα Προϊόντος – Υπηρεσίας (π.χ. μία διεπαφή σχεδιασμού προσωποποιημένων προϊόντων που τροφοδοτεί μία προηγμένη ευέλικτη αλυσίδα παραγωγής) (Coreynen, Matthyssens, & Van Bockhaven, 2017).

Έτσι, στο κοντινό μέλλον θα συνυπάρχουν τρεις συνθήκες:

- Αύξηση της διείσδυσης της υπηρεσιοποίησης στη βιομηχανία.
- Συνειδητοποίηση από τις επιχειρήσεις ότι οι υποδομές ΤΠΕ είναι αναγκαίες για την παροχή προηγμένων Συστημάτων Προϊόντος – Υπηρεσίας.
- Προσδιορισμός νέων τρόπων παραγωγής και παροχής αξίας μέσω της ενσωμάτωσης ψηφιακών υπηρεσιών (υπηρεσιών back-end, front-end και συνδυασμών) στα βιομηχανικά προϊόντα.

Οι συνθήκες αυτές δημιουργούν ένα περιβάλλον στο οποίο οι βιομηχανικές επιχειρήσεις, στην προσπάθειά τους να προσφέρουν προηγμένες υπηρεσίες και επιπλέον προστιθέμενη αξία με ψηφιακά μέσα, θα αναζητούν καλύτερες μεθόδους και εργαλεία για τη δημιουργία των αναγκαίων υποδομών ΤΠΕ, υλικού και λογισμικού.

Στο τομέα του λογισμικού, αναμένεται ότι θα αναζητήσουν μεθόδους και εργαλεία που θα καλύπτουν τις απαιτήσεις που περιγράφηκαν στις προηγούμενες ενότητες:

- Ευελιξία και γενικότητα εργαλείων και μεθοδολογίας για την ανάπτυξη λογισμικού για πολυδιάστατα συστήματα που θα συνδυάζουν υπηρεσίες front-end και back-end.
- Χρήση κατάλληλης μεθοδολογίας μοντελοποίησης για τη συνεργασία ετερογενών εμπλεκόμενων και την συνολική ανάπτυξη του συστήματος υπηρεσιών.
- Υποστήριξη δομικών και συμπεριφορικών περιγραφών του συστήματος υπηρεσιών και του λογισμικού του.

- Ολοκληρωμένα και εύχρηστα εργαλεία ανάπτυξης που υποστηρίζουν σύγχρονες μεθοδολογίες ανάπτυξης λογισμικού.
- Διασύνδεση των εργαλείων ανάπτυξης με το σύστημα και εκτέλεσης του λογισμικού υπηρεσιών.
- Υποστήριξη βασικών αρχών Υπηρεσιοστραφούς Αρχιτεκτονικής (Service-Oriented Architecture – SOA) και Διαχείρισης Επιχειρησιακών Διαδικασιών (Business Process Management – BPM).

Η προτεινόμενη μεθοδολογία και το εργαλείο ανάπτυξης καλύπτουν τις παραπάνω απαιτήσεις, και μάλιστα έχουν αποδεδειγμένες ωφέλειες, τουλάχιστον στη συνεργασία και επικοινωνία των εμπλεκόμενων πλευρών.

Έτσι, ένα ενδεικτικό παράδειγμα μελλοντικής εμπορικής εφαρμογής θα μπορούσε να περιλαμβάνει την παροχή υπηρεσιών ανάπτυξης και διαχείρισης βιομηχανικών υπηρεσιών σε μικρές ή μεσαίες βιομηχανικές επιχειρήσεις (ΜΜΕ). Στη περίπτωση αυτή, μία εταιρεία λογισμικού θα μπορούσε προσφέρει στις βιομηχανικές ΜΜΕ με χαμηλό επίπεδο «υπηρεσιοποίησης» ένα ολοκληρωμένο «πακέτο ψηφιακών υπηρεσιών», το οποίο θα περιλάμβανε συμβουλευτική υποστήριξη στον σχεδιασμό τους, την ανάπτυξη λογισμικού, αλλά και τη φιλοξενία και διαχείριση των νέων βιομηχανικών υπηρεσιών. Το πακέτο αυτό θα μπορούσε να βασίζεται σε μία ενιαία πλατφόρμα με λειτουργίες σχεδιασμού και μοντελοποίησης, ανάπτυξης λογισμικού, περιβάλλοντος εκτέλεσης, και εργαλείων διαχείρισης υπηρεσιών. Οι επιθυμητές υπηρεσίες σχεδιάζονται σε συνεργασία με τον πελάτη, και τα δύο μέρη έχουν πρόσβαση σε κοινό περιβάλλον σχεδιασμού/μοντελοποίησης. Στη συνέχεια τα μοντέλα υπηρεσιών μεταφέρονται στο ενσωματωμένο περιβάλλον ανάπτυξης (όπως αυτό που περιγράφηκε στη παρούσα διατριβή), το λογισμικό υλοποιείται από τους μηχανικούς λογισμικού (με μεθοδολογία αντίστοιχη με την προτεινόμενη). Τέλος, οι εφαρμογές εγκαθίστανται και τίθενται τελικά σε λειτουργία σε υποδομές νέφους υπό την διαχείριση της εταιρείας λογισμικού. Οι οργανωσιακές αλλαγές στην βιομηχανική επιχείρηση σχεδιάζονται και υποστηρίζονται από τους συμβούλους της εταιρείας. Η διαχείριση των υπηρεσιών γίνεται μέσα από την ενιαία πλατφόρμα, στις διεπαφές της οποίας έχει πρόσβαση ο πελάτης για την παρακολούθηση της καθημερινής λειτουργίας τους. Το όλο πακέτο παρέχεται με ένα αρχικό κόστος για την υλοποίηση, και στη συνέχεια με συνδρομή, (ως ένα είδος «service-as-a-service») η οποία θα περιλαμβάνει συντήρηση, φιλοξενία, διαχείριση και νέες εκδόσεις των υπηρεσιών ανάλογα με τις νέες επιχειρησιακές ανάγκες.

Με την αύξηση του ενδιαφέροντος της βιομηχανικής αγοράς, αναμένεται ότι θα υπάρξουν πολλές άλλες ιδέες και προϊόντα στον τομέα αυτό, προσφέροντας νέες προσεγγίσεις και προοπτικές (επιστημονικές, τεχνικές, επιχειρησιακές και εμπορικές) για την ενσωμάτωση ψηφιακών υπηρεσιών σε βιομηχανικά προϊόντα. Στα πλαίσια αυτά, τα ερευνητικά αποτελέσματα της παρούσας διατριβής αποτελούν την πρώτη, μέχρι στιγμής, συστηματική προσέγγιση για την αντιμετώπιση του ιδιαίτερου προβλήματος της ανάπτυξης αυτών των υπηρεσιών, θέτοντας τις βάσεις για μελλοντική έρευνα και πρακτικές εφαρμογές.

Βιβλιογραφία

- Acerbis, R., Bongio, A., Brambilla, M., & Butti, S. (2007). WebRatio 5: An Eclipse-based CASE tool for engineering Web applications. *7th International Conference, ICWE 2007 Como, Italy, July 16-20, 2007 Proceedings* (σσ. 501-505). Springer Berlin Heidelberg.
- Acerbis, R., Bongio, A., Brambilla, M., Butti, S., Ceri, S., & Fraternali, P. (2008). Web applications design and development with WebML and Webratio 5.0. Στο *Objects, components, models and patterns* (σσ. 392-411). Springer Berlin Heidelberg, 2008.
- Activiti. (2015). *Activiti User Guide*. Ανάκτηση από <http://www.activiti.org/userguide>
- Agostinho, C., Bazoun, H., Zacharewicz, G., Ducq, Y., Boye, H., & Jardim-Goncalves, R. (2013). Model Transformation Principles for Servitization and Service Systems. *International IFIP Working Conference on Enterprise Interoperability*. Enschede.
- Aguilar-Savén, R. S. (2004). Business process modelling: Review and framework. *International Journal of Production Economics*, 90(2), 129-149.
- Aho, P., Mäki, M., Pakkala, D., & Ovaska, E. (2009). MDA-based tool chain for web services development. *Proceedings of the 4th Workshop on Emerging Web Services Technology* (σσ. 11-18). ACM.
- Almeida, J. P., Iacob, M.-E., Jonkers, H., Lankhorst, M., & van Leeuwen, D. (2007). An integrated model-driven service engineering environment. Στο *Enterprise Interoperability* (σσ. 79-89). Springer London.
- Ameller, D., Burgués, X., Collell, O., Costal, D., Franch, X., & Papazoglou, M. P. (2015). Development of service-oriented architectures using model-driven development: A mapping study. *Information and Software Technology*, 62, σσ. 42-66.
- Ameller, D., Franch, X., & Cabot, J. (2010). Dealing with non-functional requirements in model-driven development. *Requirements Engineering Conference (RE), 2010 18th IEEE International* (σσ. 189-198). IEEE.
- Asplund, F., & Törngren, M. (2015). The Discourse on Tool Integration Beyond Technology, A Literature Survey. *Journal of Systems and Software*, 106, σσ. 117-131.
- Aurich, J. C., Fuchs, C., & Wagenknecht, C. (2006). Life cycle oriented design of technical Product-Service Systems. *Journal of Cleaner Production*, 14(17), σσ. 1480-1494.
- Autili, M., Di Ruscio, D., Di Salle, A., Inverardi, P., & Tivoli, M. (2013). A Model-Based Synthesis Process for Choreography Realizability Enforcement. *Fundamental Approaches to Software Engineering, 16th International Conference, FASE 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings*, (σσ. 37-52).
- Babar, M. A., & Gorton, I. (2004). Comparison of scenario-based software architecture evaluation methods. *Software Engineering Conference, 2004. 11th Asia-Pacific* (σσ. 600-607). IEEE.

- Bahler, L., Caruso, F., & Micallef, J. (2007). A Practical Method and Tool for Systems Engineering of Service-Oriented Applications. *8th International Conference on Web Information Systems Engineering Nancy, France, December 3-7, 2007 Proceedings* (σσ. 472-483). Springer Berlin Heidelberg.
- Baines, T., & Lightfoot, H. W. (2013). Servitization of the manufacturing firm: Exploring the Operations Practices and Technologies that deliver advanced services. *International Journal of Operations & Production Management*, *34*(1).
- Baines, T., Lightfoot, H., Peppard, J., Johnson, M., Tiwari, A., Shehab, E., & Swink, M. (2009). Towards an operations strategy for product centric servitization. *International Journal of Operations and Production Management*, *29*(5), σσ. 494-519.
- Barnett, N. J., Parry, G., Saad, M., Newnes, L. B., & Goh, Y. (2013). Servitization: Is a Paradigm Shift in the Business Model and Service Enterprise Required? *Strategic Change*, *22*(3-4), σσ. 145-156.
- Bartol, N. (2014). Cyber supply chain security practices DNA – Filling in the puzzle using a diverse set of disciplines. *Technovation*, *34*(7), 354-361.
- Basili, V. R., & Briand, L. C. (1996). A validation of object-oriented design metrics as quality indicators. *IEEE Transactions on software engineering*, *22*(10), 751-761.
- Bastl, M., Johnson, M., Lightfoot, H., & Evans, S. (2012). Buyer-supplier relationships in a servitized environment: an examination with Cannon and Perreault's framework. *International Journal of Operations & Production Management*, *32*(6), σσ. 650-675.
- Bazoun, H., Zacharewicz, G., Ducq, Y., & Boyé, H. (2014). SLMToolBox: An Implementation of MDSEA for Servitisation and Enterprise Interoperability. Στο *Enterprise Interoperability VI* (σσ. 101-111). Springer International Publishing.
- Becker, J., Beverungen, D. F., & Knackstedt, R. (2010). The challenge of conceptual modeling for product–service systems: status-quo and perspectives for reference models and modeling languages. *Information Systems and e-Business Management*, *8*(1), σσ. 33-66.
- Beisiegel, M., Blohm, H., Booz, D., Dubray, J.-J., Colyer, A., Edwards, M., . . . Pavlik, G. (2007). *Service component architecture: Building systems using a Service Oriented Architecture*. BEA, IBM, Interface21, IONA, Oracle, SAP, Siebel, Sybase.
- Ben Hamida, A., Kon, F., Oliva, G. A., Dos Santos, C. E., Lorre, J. P., Autili, M., . . . Bertolino, A. (2012). An Integrated Development and Runtime Environment for the Future Internet. Στο *The Future Internet* (σσ. 81-92). Springer Berlin Heidelberg.
- Bercovici, A., Fournier, F., & Wecker, A. J. (Επιμ.). (2008). From Business Architecture to SOA Realization Using MDD. *Model Driven Architecture – Foundations and Applications, 4th European Conference, ECMDA-FA 2008, Berlin, Germany, June 9-13, 2008. Proceedings*, (σσ. 381-392).

- Berkovich, M., Leimeister, J. M., & Krcmar, H. (2009). Suitability of product development methods for hybrid products as bundles of classic products, software and service elements. *ASME 2009 - International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE*. San Diego, USA.
- Berkovich, M., Leimeister, J. M., Hoffmann, A., & Krcmar, H. (2014). A requirements data model for product service systems. *Requirements Engineering*, 19(2), σσ. 161-186.
- Berkovich, M., Leimeister, J., & Krcmar, H. (2011). Requirements Engineering for Product Service Systems. *Business & Information Systems Engineering*, 3(6), σσ. 369-380.
- Bikfalvi, A., Lay, G., Maloca, S., & Waser, B. R. (2013). Servitization and networking: large-scale survey findings on product-related services. *Service Business*, 7(1), σσ. 61-82.
- Böhmman, T., Leimeister, J. M., & Möslin, K. (2014). Service-Systems-Engineering. *Business & Information Systems Engineering*, 6(2), σσ. 73-79.
- Boyé, H., & Bazoun, H. (2014). Service Life-cycle Management Tool Box. Στο S. Wiesner, C. Guglielmina, S. Gusmeroli, & G. Doumeingts (Επιμ.), *Manufacturing Service Ecosystem: Achievements of the European 7th Framework Programme FoF-ICT Project MSEE: Manufacturing Service Ecosystem (Grant No. 284860)* (σσ. 60-66).
- Boyé, H., Bazoun, H., & Belkhelladi, K. (2014). SLMToolBox: A tool set for service engineering. *Model-Driven Engineering and Software Development (MODELSWARD), 2014 2nd International Conference on* (σσ. 666-672). IEEE.
- Brax, S., & Visintin, F. (2016). Meta-model of servitization: The integrative profiling approach. *Industrial Marketing Management*. doi:10.1016/j.indmarman.2016.04.014
- Broy, M., Feilkas, M., Herrmannsdoerfer, M., Merenda, S., & Ratiu, D. (2010). Seamless model-based development: From isolated tools to integrated model engineering environments. *Proceedings of the IEEE*, 98(4), σσ. 526-545.
- Budgen, D., & Thomson, M. (2003). CASE tool evaluation: experiences from an empirical study. *Journal of Systems and Software*, 67(2), 55-75.
- Bullinger, H.-J., Fähnrich, K.-P., & Meiren, T. (2003). Service engineering—methodical development of new service products. *International Journal of Production Economics*, 85(3), σσ. 275-287.
- Cannon, J. P., & Perreault, W. D. (1999, November). Buyer-Seller Relationships in Business Markets. *Journal of Marketing Research*, 36(4), σσ. 439-460.
- Cardoso, J., Barros, A., May, N., & Kylau, U. (2010). Towards a unified service description language for the internet of services: Requirements and first developments. *Services Computing (SCC), 2010 IEEE International Conference on* (σσ. 602-609). IEEE.
- Cardoso, J., Voigt, K., & Winkler, M. (2009). Service engineering for the internet of services. Στο *Enterprise Information Systems* (σσ. 15-27).

- Cavaliere, S., & Pezzota, G. (2012). Product-Service Systems Engineering: State of the art and research challenges. *Computers in Industry*, 63(4), σσ. 278-288.
- Chae, B. K. (2014). A complexity theory approach to IT-enabled services (IESs) and service innovation: Business analytics as an illustration of IES. *Decision Support Systems*, 57.
- Chae, B. K. (2014). A complexity theory approach to IT-enabled services (IESs) and service innovation: Business analytics as an illustration of IES. *Decision Support Systems*, 57, σσ. 1-10.
- Channabasavaiah, K., Holley, K., & Tuggle, E. (2003). Migrating to a service-oriented architecture. *IBM DeveloperWorks*, 16.
- Chappell, D. (2007). *Introducing SCA*. Chappell & Associates.
- Chen, D., Ducq, Y., Doumeingts, G., Zachariewicz, G., & Alix, T. (2012). A Model Driven Approach for the Modeling of Services in Virtual Enterprise. Στο M. Zelm, R. Sanchis, R. Poler, & G. Doumeingts (Επιμ.), *Enterprise Interoperability: I-ESA'12 Proceedings*.
- Chen, D., Ducq, Y., Doumeingts, G., Zachariewicz, G., & Alix, T. (2012). A Model Driven Approach for the Modeling of Services in Virtual Enterprise. Στο M. Zelm, R. Sanchis, R. Poler, & G. Doumeingts (Επιμ.), *Enterprise Interoperability: I-ESA'12 Proceedings*. 181-187.
- Chen, H. M. (2008). Towards Service Engineering: Service Orientation and Business-IT Alignment. *Proceedings of the 41st Hawaii International Conference on System Sciences*.
- Chidamber, S. R., & Kemerer, C. F. (1994). A metrics suite for object oriented design. *IEEE Transactions on software engineering*, 20(6), 476-493.
- Chowdhury, I., & Zulkernine, M. (2011). Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities. *Journal of Systems Architecture*, 57(3), 294-313.
- Constantin, J. A., & Lusch, R. F. (1994). *Understanding resource management*. Oxford, OH: Planning Forum.
- Coreynen, W., Matthyssens, P., & Van Bockhaven, W. (2017). Boosting servitization through digitization: Pathways and dynamic resource configurations for manufacturers. *Industrial Marketing Management*, 42-53.
- Correia, A., Stokic, D., Sifaka, R., & Scholze, S. (2017). Ontology for Collaborative Development of Product Service Systems Based on Basic Formal Ontology. 2017 *International Conference on Engineering, Technology and Innovation (ICE/ITMC)* (σσ. 1214-1221). IEEE.
- Dai, N., Mandel, L., & Ryman, A. (2007). *Eclipse Web tools platform: developing Java Web applications*. Pearson Education.

- Daniele, L. M., Silva, E., Pires, L. F., & van Sinderen, M. (2009). A SOA-based platform-specific framework for context-aware mobile applications. Στο *Enterprise Interoperability* (σσ. 25-37).
- De Castro, V., Marcos, E., & Vara, J. M. (2011). Applying CIM-to-PIM model transformations for the service-oriented development of information systems. *Information and Software Technology*, 53(1), σσ. 87-105.
- De Castro, V., Marcos, E., & Wieringa, R. (2009). Towards a service-oriented MDA-based approach to the alignment of business processes with IT systems: From the business model to a web service composition model. *International Journal of Cooperative Information Systems*, 18(02), σσ. 225-260.
- Dialogic; Executive Agency for Small and Medium-sized Enterprises (European Commission); Technopolis Group; University of Cambridge;. (2018). *Study on the potential of servitisation and other forms of product-service provision for EU SMEs*. European Commission, Executive Agency for Small and Medium-sized Enterprises. European Commission.
- Dobrica, L., & Niemela, E. (2002). A survey on software architecture analysis methods. *IEEE Transactions on software Engineering*, 28(7), 638-653.
- Ducq, Y., Agostinho, C., Chen, D., Zacharewicz, G., & Jardim-Goncalves, R. (2014). Generic Methodology for Service Engineering based on Service Modelling and Model Transformation. Στο S. Wiesner, C. Guglielmina, S. Gusmeroli, & G. Doumeingts (Επιμ.), *Manufacturing Service Ecosystem: Achievements of the European 7th Framework Programme FoF-ICT Project MSEE: Manufacturing Service Ecosystem (Grant No. 284860)* (σσ. 41-49).
- Ducq, Y., Chen, D., & Alix, T. (2012). Principles of Servitization and Definition of an Architecture for Model Driven Service System Engineering. Στο *Enterprise Interoperability* (σσ. 117-128).
- Duffy, R., & Fearne, A. (2004). The impact of supply chain partnership on supplier performance. *The International Journal of Logistics Management*, 15(1), σσ. 57-71.
- Dumas, M. M., O'Sullivan, J. J., Heravizadeh, M., David, E., & ter Hofstede, A. H. (χ.χ.). Towards a semantic framework for service description. Στο R. Meersman, K. Aberer, & T. S. Dillon (Επιμ.), *Proceedings Data Semantics 9: Semantic Issues in E-Commerce*.
- Eclipse Foundation. (2006). *Acceleo*. Ανάκτηση January 2017, από <https://www.eclipse.org/acceleo/>
- Ellis, G., & Dix, A. (2006). An Explorative Analysis of User Evaluation Studies in Information Visualisation. *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization* (σσ. 1-7). ACM.

- Esposito, E., & Evangelista, P. (2014). Investigating virtual enterprise models: literature review and empirical findings. *International Journal of Production Economics*, 148, σσ. 145-157.
- Fabra, J., Álvarez, P., & Bañares, J. A. (2011). DENEb: a platform for the development and execution of interoperable dynamic Web processes. *Concurrency and Computation: Practice and Experience*, 18, σσ. 2421-2451.
- Fabra, J., De Castro, V., Álvarez, P., & Marcos, E. (2012). Automatic execution of business process models: Exploiting the benefits of Model-driven Engineering approaches. *Journal of Systems and Software*, 85(3), σσ. 607-625.
- Ferrario, R., & Guarino, N. (2009). Towards an ontological foundation for services science. Στο J. Domingue, & P. Traverso (Επιμ.), *Future Internet – FIS 2008* (σσ. 152-169). Springer Berlin Heidelberg.
- France, R., & Rumpe, B. (2007). Model-driven development of complex software: A research roadmap. *Proceedings of Future of Software Engineering 2007* (σσ. 37-54). IEEE Computer Society.
- Francese, R., Risi, M., Scanniello, G., & Tortora, G. (2015). Model-Driven Development for Multi-platform Mobile Applications. *Product-Focused Software Process Improvement , 16th International Conference, PROFES 2015 Proceedings* (σσ. 61-67). Springer International Publishing Switzerland.
- Froehle, C. M., & Roth, A. V. (2004). New measurement scales for evaluating perceptions of the technology-mediated customer service experience. *Journal of Operations Management*, 22(1), σσ. 1-21.
- Fuentes-Fernandez, L., & Valecillo-Moreno, A. (2004, April). An Introduction to UML Profiles. *UPGRADE: The European Journal for the Informatics Professional*, V(2).
- Galliers, R. D. (1990). Choosing appropriate information systems research approaches: a revised taxonomy. *Proceedings of the IFIP TC8 WG8*, 2.
- Gao, J., Yao, Y., Zhu, V. C., Sun, L., & Lin, L. (2011). Service-oriented manufacturing: a new product pattern and manufacturing paradigm. *Journal of Intelligent Manufacturing*, 22(3), 435-446. doi:10.1007/s10845-009-0301-y
- Gebauer, H., Fleisch, E., & Friedli, T. (2005). Overcoming the service paradox in manufacturing companies. *European Management Journal*, 23(1), σσ. 14-26.
- Georgakopoulos, D., Hornick, M., & Sheth, A. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3(2), σσ. 119-153.
- Gilmore, S., Gönczy, L., Koch, N., Mayer, P., Tribastone, M., & Varró, D. (2011). Non-functional properties in the model-driven development of service-oriented systems. *Software & Systems Modeling*, 10(3), σσ. 287-311.

- Glushko, R. J. (2010). Seven contexts for service system design. Στο *Handbook of service science* (σσ. 219-249). Springer US.
- Goldstein, S. M., Johnston, R., Duffy, J., & Rao, J. (2002). The service concept: the missing link in service design research? *Journal of Operations management*, 20(2), 121-134.
- Grace, A., Finnegan, P., & Butler, T. (2008). "Service Co-Creation with the Customer: the Role of Information Systems. *ECIS 2008 Proceedings*. Galway, Ireland.
- Haase, T., & Nagl, M. (2009). Service-oriented architectures and tool integration. *Proceedings of the 8th world congress of chemical engineering*. Montreal, Canada.
- Haase, T., & Nagl, M. (2011). Application integration within an integrated design environment. *Computers & Chemical Engineering*, 35(4), σσ. 736-747.
- Haberl, W., Herrmannsdoerfer, M., Kugele, S., Tautschnig, M., & Wechs, M. (2010). Seamless Model-Driven Development Put into Practice. *4th International Symposium on Leveraging Applications, ISoLA 2010, Heraklion, Crete, Greece, October 18-21, 2010* (σσ. 18-32). Springer-Verlag Berlin Heidelberg.
- Health Level Seven International. (2017, December 24). *Introduction to HL7 Standards*. Ανάκτηση από <http://www.hl7.org/implement/standards/>
- Heiskala, M., Hiekkanen, K., & Korhonen, J. J. (2011). The Impact of Information Technology Enabled Services on Value Co-Creation. *Naples Forum on Service*, (σσ. 14-17). Villa Orlandi, Capri, Italy.
- HL7. (2007). *HL7 Reference Information Model 2.16*. Ανάκτηση από <http://www.hl7.org/implement/standards/rim.cfm>
- Hockerts, K., & Weaver, N. (2002). Towards a theory of sustainable product service systems. *Ecological and Economic Basics*.
- Huang, S., & Fan, Y. (2007). Model Driven and Service Oriented Enterprise Integration---The Method, Framework and Platform. *Advanced Language Processing and Web Information Technology, 2007. ALPIT 2007. Sixth International Conference on* (σσ. 504-509). IEEE.
- Huhns, M. N., & Singh, M. P. (2005). Service-Oriented Computing: Key Concepts and Principles. *IEEE Internet Computing*.
- Hutchinson, J., Rouncefield, M., & Whittle, J. (2011). Model-driven engineering practices in industry. *Software Engineering (ICSE), 2011 33rd International Conference on* (σσ. 633-642). IEEE.
- International Labour Organization. (2015). *World Employment and Social Outlook: Trends 2015*.

- Issarny, V., Georgantas, N., Hachem, S., Zarras, A., Vassiliadis, P., Autili, M., . . . Ben Hamida, A. (2011). Service-oriented middleware for the future internet: state of the art and research directions. *Journal of Internet Services and Applications*, 2(1), σσ. 23-45.
- Jilani, A. A., Usman, M., & Halim, Z. (2010). Model transformations in model driven architecture. *Universal Journal of Computer Science and Engineering Technology*, 1(1), σσ. 50-54.
- Johnson, M., & Mena, C. (2008). Supply chain management for servitised products: A multi-industry case study. *International Journal of Production Economics*, 114(1), 27-39.
- Jonkers, H., Groenewegen, L., Bonsangue, M., van Buuren, R., Quartel, D. A., Lankhorst, M. M., & Aldea, A. (2005). A Language for Enterprise Modelling. Στο M. M. Lankhorst, *Enterprise Architecture at Work* (σσ. 83-113).
- Kardoš, M., & Drozdová, M. (2010). Analytical method of CIM to PIM transformation in Model Driven Architecture (MDA). *Journal of Information and Organizational Sciences*, 34(1), σσ. 89-99.
- Kirchner, M. (2017). *High Performance Through Business Process Management*. Springer Nature.
- Kline, B., & Seffah, A. (2005). Evaluation of integrated software development environments: Challenges and results from three empirical studies. *International Journal of Human-Computer Studies*, 63, 607-027.
- Kokkinakos, P., Markaki, O., Panopoulos, D., Kousouris, S., & Askounis, D. (2013). Dynamic Manufacturing Networks Monitoring and Governance. *Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services*, σσ. 446-453.
- Kowalkowski, C., Kindstrom, D., & Gebauer, H. (2013). ICT as a catalyst for service business orientation. *Journal of Business & Industrial Marketing*, 28(6), σσ. 506-513.
- Lanusse, A., Tanguy, Y., Espinoza, H., Mraidha, C., Gerard, S., Tessier, P., . . . Terrier, F. (2009). Papyrus UML: an open source toolset for MDA. *Proc. of the Fifth European Conference on Model-Driven Architecture Foundations and Applications (ECMDA-FA 2009)*, (σσ. 1-4).
- Lim, C. H., Kim, M. J., Heo, J. Y., & Kim, K. J. (2015). Design of informatics-based services in manufacturing industries: case studies using large vehicle-related databases. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-015-1123-8
- Maglio, P. P., & Spohrer, J. (2008). Fundamentals of service science. *Journal of the Academy of Marketing Science*(36), σσ. 18-20.
- Maglio, P. P., Srinivasan, S., Kreulen, J. T., & Spohrer, J. (2006). Service systems, service scientists, SSME, and innovation. *Communications of the ACM*, 49(7), σσ. 81-85.

- Maglio, P. P., Vargo, S. L., Caswell, N., & Spohrer, J. (2009). The service system is the basic abstraction of service science. *Information Systems and e-business Management*, 7(4), σσ. 395-406.
- Martinez, M. T., Fouletier, P., Park, K. H., & Favrel, J. (2001). Virtual Enterprise - Organization, evolution and control. *International Journal of Production Economics*, 74(1), σσ. 225-238.
- Martinez, V., Bastl, M., Kingston, J., & Evans, S. (2010). Challenges in transforming manufacturing organisations into product-service providers. *Journal of Manufacturing Technology Management*, 21(4), σσ. 449-469.
- Maussang, N., Sakao, T., Zwolinski, P., & Brissaud, D. (2007). A model for designing product-service systems using functional analysis and agent based model. *INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07. PARIS, FRANCE*.
- Mayer, P., Koch, N., Schroeder, A., & Knapp, A. (2009). *The UML4SOA profile*. Ludwig-Maximilians-Universitaet Muenchen.
- Mayr, C., Zdun, U., & Dustdar, S. (2011). View-based model-driven architecture for enhancing maintainability of data access services. *Data & Knowledge Engineering*, 70(9).
- Meier, H., Volker, O., & Funke, B. (2011). Industrial Product-Service Systems (IPS). Paradigm shift by mutually determined products and services. *International Journal of Advanced Manufacturing Technology*, 52, σσ. 1175-1191.
- Meier, H., Volker, O., & Funke, B. (2011). Industrial Product-Service Systems (IPS). Paradigm shift by mutually determined products and services. *International Journal of Advanced Manufacturing Technology*, 52, σσ. 1175-1191.
- Metzger, D., Niemöller, C., & Thomas, O. (2017). Design and demonstration of an engineering method for service support systems. *Information Systems and e-Business Management*, 15(4), 789–823.
- Mietinnen, S., Rontti, S., Kuure, E., & Lindström, A. (2012). Realizing design thinking through a service design process and an innovative prototyping laboratory—Introducing Service Innovation Corner (SINCO). *Proceedings of the conference on design research society (DRS 2012)*.
- Morelli, N. (2002). Designing Product/Service Systems: A Methodological Exploration. *Design Issues*, 18(3), σσ. 3-17.
- MSEE. (2012). *MSEE Service-System Functional and Modular Architecture*. Ανάκτηση από <http://interop-vlab.eu/wp-content/uploads/securepdfs/2017/04/MSEE-D41.1-MSEE-Service-System-Functional-and-Modular-Architecture.pdf>
- MSEE. (2012). *Service concepts, models and method: Model Driven Service Engineering*. Ανάκτηση από

- <https://cordis.europa.eu/docs/projects/cnect/0/284860/080/deliverables/001-MSEED112ServiceconceptmodelsmethodatCIMPIMPSMlevelV2.pdf>
- MSEE. (2012). *Specifications and Design of Service Lifecycle Management ToolBox*. Ανάκτηση από <https://cordis.europa.eu/docs/projects/cnect/0/284860/080/deliverables/001-MSEED153SpecificationsandDesignofServiceLifecycleManagementPlatformM18.pdf>
- MSEE. (2013). *FI Utility Service specifications and architecture - M21 Issue*. Ανάκτηση από <https://cordis.europa.eu/docs/projects/cnect/0/284860/080/deliverables/001-D332FIUtilityServicesSpecificationsandArchitectureM21.pdf>
- MSEE. (2013). *Service Delivery Infrastructure Specifications and Architecture*. Ανάκτηση από <https://cordis.europa.eu/docs/projects/cnect/0/284860/080/deliverables/001-D432ServiceDeliveryInfrastructureSpecificationsandArchitectureM21.pdf>
- Mukerji, J., & Miller, J. (2003). *MDA Guide Version 1.0.1*. The Object Management Group (OMG).
- Müller, P., & Stark, R. (2008). Detecting and Structuring Requirments for the development of product-service systems. *19th Symposium "Design for X"*. Neukirchen.
- Munzner, T. (2006). A nested model for visualization design and validation. *IEEE transactions on visualization and computer graphics*, 15(6).
- Neely, A. (2008). Exploring the financial consequences of the servitization of manufacturing. *Operations Management Research*, 1(2), σσ. 103-118.
- Neely, A., Benedettini, O., & Visnjic, I. (2011). The servitization of manufacturing: Further evidence. *Proceedings of the 18th European Operations Management Association Conference*. Cambridge.
- Neff, A. A., Hamel, F., Herz, T. P., Uebernickel, F., & Brenner, W. (2013). Fostering Efficiency in Information Systems Support for Product-Service Systems in the Manufacturing Industry.
- Neubauer, P., Mayerhofer, T., & Gerti, K. (2014). Towards Integrating Modeling and Programming Languages: The Case of UML and Java*. *GEMOC 2014*, (σ. 23).
- Ng, I., Parry, G., Wild, P., McFarlane, D., & Tasker, P. (Επιμ.). (2011). *Complex Engineering Service Systems: Concepts and Research*. London: Springer-Verlag.
- Nguyen, H. N., Exner, K., Schnürmacher, C., & Rainer, S. (2014). Operationalizing IPS² development process: A method for realizing IPS² developments based on Process-based project planning. *Procedia CIRP* 16, (σσ. 217-222).
- Object Management Group. (2008). *MOF Model to Text Transformation Language*. Ανάκτηση από OMG Website: <http://www.omg.org/spec/MOFM2T/1.0/>

- Object Management Group. (2008). *MOF Model to Text Transformation Language Specification*. Ανάκτηση January 2017, από <http://www.omg.org/spec/MOFM2T/About-MOFM2T/>
- Object Management Group. (2011). *Business Process Model And Notation (BPMN) - Normative documents*. Ανάκτηση από <http://www.omg.org/spec/BPMN/2.0/>
- Object Management Group. (2015). *Unified Modeling Language, Version 2.5*. Ανάκτηση από <http://www.omg.org/spec/UML/2.5>
- Object Management Group. (2015, 3 17). *XML Metadata Interchange (XMI)*. Ανάκτηση από Object Management Group Website: <http://www.omg.org/spec/XMI/>
- Oliva, R., & Kallenberg, R. (2003). Managing the transition from products to services. *International Journal of Service Industry Management*, 14(2), σσ. 160-172.
- Olivé, A. (2007). *Conceptual Modeling of Information Systems*. Springer Berlin Heidelberg. doi:10.1007/978-3-540-39390-0
- Osis, J., & Asnina, E. (Επιμ.). (2010). *Model-Driven Domain Analysis and Software Development: Architectures and Functions*. IGI Global.
- Papazoglou, M. P., & Georgakopoulos, D. (2003, October). Service Oriented Computing. *Communications of the ACM*, 46(10), σσ. 25-28.
- Pernstål, J., Gorschek, T., Feldt, R., & Florén, D. (2015). Requirements communication and balancing in large-scale software-intensive product development. *Information and Software Technology*, 67, 44-64.
- Pezzotta, G., Sala, R., Pirola, F., Campos, A. R., Margarito, A., Correia, A. T., . . . Mourtzis, D. (2016). Definition of a PSS engineering environment: From the theoretical methodology to the platform implementation. *XXI Summer School Francesco Turco 2016-SMART MANUFACTURING: NEW PARADIGMS FOR A SMARTER WORLD*. 13, σσ. 97-101. Naples: AIDI-Italian Association of Industrial Operations Professors.
- Purchase, V., Parry, G., Valerdi, R., Nightingale, D., & Mills, J. (2011). Enterprise transformation: Why are we interested, what is it, and what are the challenges? *Journal of Enterprise Transformation*, 1(1), σσ. 14-33.
- Rademakers, T. (2012). *Activiti in Action: Executable business processes in BPMN 2.0*. Manning Publications Co.
- Radhakrishnan, R., & Wookey, M. (2004). *Model driven architecture enabling service oriented architectures*. Whitepaper, Sun Microsystems.
- Reim, W., Parida, V., & Örtqvist, D. (2015). Product–Service Systems (PSS) business models and tactics—a systematic literature review. *Journal of Cleaner Production*(97), σσ. 61-75.

- Romero, D., Rabelo, R., & Molina, A. (2012). On the Management of Virtual Enterprise's Inheritance between Virtual Manufacturing & Service Enterprises: Supporting "Dynamic" Product-Service Business Ecosystems. *18th International ICE-Conference on Engineering, Technology and Innovation*, (σσ. 1-11).
- Rosen, M., Lublinsky, B., Smith, K. T., & Balcer, M. J. (2012). *Applied SOA: Service Oriented Architecture and Design Strategies*. Wiley Publishing Inc.
- Rosen, M., Lublinsky, B., Smith, K. T., & Balcer, M. J. (2012). *Applied SOA: Service Oriented Architecture and Design Strategies*. Wiley Publishing Inc.
- Rosenberg, D. (2010). *Modeling Service-Oriented Architectures: An Illustrated Example using Sparx Systems*. Sparx Systems Pty Ltd and ICONIX.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *Unified Modeling Language Reference Manual, The*. Pearson Higher Education.
- Saccani, N., Visintin, F., & Rapaccini, M. (2014). Investigating the linkages between service types and supplier relationships in servitized environments. *International Journal of Production Economics*, σσ. 226-238.
- Sacevski, I., & Veseli, J. (2007). Introduction to Model Driven Architecture (MDA). *Seminar Paper*. University of Salzburg.
- Sampson, S. E., & Froehle, C. M. (2006). Foundations and implications of a proposed unified services theory. *Production and operations management*, σσ. 329-343.
- Scheidgen, M. (2006). Model patterns for model transformations in model driven development. *Model-Based Development of Computer-Based Systems and Model-Based Methodologies for Pervasive and Embedded Software, 2006. MBD/MOMPES 2006. Fourth and Third International Workshop on*. IEEE.
- Schmenner, R. W. (2009). Manufacturing, service, and their integration: some history and theory. *International Journal of Operations & Production Management*, 29(5), σσ. 431-443.
- Schneider, B. (1995). *Winning the Service Game*. Boston, MA: Harvard Business School Press.
- Schot, N. (2012). *Model-Driven SOA*. Report for research topics, University of Twente, EEMCS - Software Engineering Group.
- Schumann, J. H., Wunderlich, N. V., & Wangenheim, F. (2012). Technology mediation in service delivery: A new typology and an agenda for managers and academics. *Technovation*, 32(2), σσ. 133-143.
- Sheng, Q. Z., & Benatallah, B. (2005). Sheng, Quan Z., and Boualem Benatallah. "ContextUML: a UML-based modeling language for model-driven development of context-aware web services. *Mobile Business, 2005. ICMB 2005. International Conference on*, (σσ. 206-212).

- Sheng, Q. Z., Pohlenz, S., Yu, J., Wong, H. S., Ngu, A. H., & Maamar, Z. (2009). ContextServ: A Platform for Rapid and Flexible Development of Context-Aware Web Services. *Proceedings of the 31st International Conference on Software Engineering* (σσ. 619-622). IEEE Computer Society.
- Sklyar, A., Kowalkowski, C., Tronvoll, B., & Sörhammar, D. (2019). Organizing for digital servitization: A service ecosystem perspective. *Journal of Business Research*. doi:10.1016/j.jbusres.2019.02.012
- Skouradaki, M., Roller, D. H., Leymann, F., Ferme, V., & Pautasso, C. (2015). On the road to benchmarking BPMN 2.0 workflow engines. *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering* (σσ. 301-304). ACM.
- Spohrer, J., & Kwan, S. K. (2008). Service Science, Management, Engineering, and Design (SSMED): An Emerging Discipline. *he future of services: Trends and perspectives, 1*, σ. 232.
- Stokic, D., & Correia, A. T. (2015). Context sensitive Web service engineering environment for product extensions in manufacturing industry. *7th International Conf. on Advanced Service Computing*. Nice.
- Sullivan, A., & Sheffrin, S. M. (2003). *Economics: Principles in action*. Upper Saddle river, New Jersey 07458: Pearson Prentice Hall.
- The Open Group. (2011). *SOA Reference Architecture Technical Standard*. The Open Group.
- Thomas, I., & Nejme, B. A. (1992). Definitions of tool integration for environments. *Software*, 9(2), σσ. 29-35.
- Toma, I., García, J. M., Larizgoitia, I., & Fensel, D. (2014). A Semantically Enabled Service Delivery Platform: An Architectural Overview. Στο R. Ramanathan, & K. Raja (Επιμ.), *Handbook of Research on Architectural Trends in Service-Driven Computing* (σσ. 181-186). IGI Global.
- Touzi, J., Benaben, F., Pingaud, H., & Lorre, J. P. (2009). A model-driven approach for collaborative service oriented architecture design. *International Journal of Production Economics*, 121(1), σσ. 5-20.
- Truyen, F. (2006). *The Fast Guide to Model Driven Architecture The Basics of Model Driven Architecture*. Cephas Consulting Corp.
- Truyen, F. (2006). *The Fast Guide to Model Driven Architecture: The Basics of Model Driven Architecture (MDA)*. Cephas Consulting Corp.
- Tukker, A. (2004). Eight types of Product-Service System: Eight ways to sustainability? Experiences from Suspronet. *Business Strategy and the Environment*, 13, σσ. 246-260.

- Tukker, A. (2004). Eight types of Product-Service System: Eight ways to sustainability? Experiences from SUSPRONET. *Business Strategy and the Environment*, 13, σσ. 246-260.
- Van Riel, A. C., & Lievens, A. (2004). New service development in high tech sectors: A decision-making perspective. *International Journal of Service Industry Management*, 15(1), 72-101.
- Vandermerwe, S., & Rada, J. (1988). Servitization of business: adding value by adding services. *European Management Journal*, 6(4), σσ. 314-324.
- Vandermerwe, S., & Rada, J. (1988). Servitization of business; Adding value by adding services. *European Management Journal*, 6(4), σσ. 314-324.
- Vargo, S. (2012). RMIT International Visiting Fellow Presentation, Karlstad University. Melbourne, Australia.
- Vargo, S. L., & Lusch, R. F. (2004). "Evolving to a new dominant logic for marketing.". *Journal of marketing*, 68(1), σσ. 1-17.
- Vargo, S. L., & Lusch, R. F. (2007). From goods to service(s): Divergences and convergence of logics. *Industrial Marketing Management*, 37, σσ. 254-259.
- Vargo, S. L., Maglio, P. P., & Akaka, M. A. (2008). On value and value co-creation: A service systems and service logic perspective. *European management journal*, 26(3), σσ. 145-162.
- Vasantha, G., Roy, R., Lelah, A., & Brissaud, D. (2012). A review of product-service systems design methodologies. *Journal of Engineering Design*, 23(9), σσ. 635-659.
- Väyrynen, K. (2010). Software business and industrial companies: Identifying Capabilities for three types of software business. *ICIS 2010 Proceedings*.
- Walderhaug, S., Stav, E., & Mikalsen, M. (2007). The MPOWER Tool Chain-Enabling Rapid Development of Standards-based and Interoperable Homecare Applications. *Proceedings of Norsk Informatikk Konferanse (NIK 2007)*, (σσ. 103-107).
- Wallin, J., Parida, V., & Isaksson, O. (2015). Understanding product-service system innovation capabilities development for manufacturing companies. *Journal of Manufacturing Technology Management*, 26(5), 763-787.
- Wasserman, A. I. (1990). Tool integration in software engineering environments. Στο *Software Engineering Environments* (σσ. 137-149). Springer Berlin Heidelberg.
- Weske, M. (2007). *Business Process Management*. Berlin Heidelberg: Springer-Verlag .
- Westergren, U. H. (2011). Opening up innovation: the impact of contextual factors on the co-creation of IT-enabled value adding services within the manufacturing industry. *Information Systems and e-Business Management*, 9(2), σσ. 223-245.

- White, S. A. (2008). *BPMN modeling and reference guide: understanding and using BPMN*. Future Strategies Inc.
- Whitehead, E. J., & Wiggins, M. (1998). WebDAV: IETF standard for collaborative authoring on the Web. *IEEE Internet Computing*, 2(5), 34-40.
- Wiesner, S., Guglielmina, C., Gusmeroli, S., & Doumeingts, G. (Επιμ.). (2014). *Manufacturing Service Ecosystem - Achievements of the European 7th Framework Programme FoF-ICT Project MSEE: Manufacturing Service Ecosystem (Grant No. 284860)*.
- Wiesner, S., Westphal, I., Hirsch, M., & Thoben, K.-D. (2013). Manufacturing Service Ecosystems. Στο *Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services* (σσ. 305-312). Springer Berlin Heidelberg.
- Wirsing, M., Hölzl, M., Koch, N., Mayer, P., & Schroeder, A. (2008). Service Engineering: The Sensoria Model Driven Approach. *Proceedings of Software Engineering Research, Management and Applications (SERA 2008)*, (σσ. 20-22).
- World Bank. (2015, 3 31). *Services, etc., value added (% of GDP)*. Ανάκτηση από World Bank national accounts data, and OECD National Accounts data files.: <http://data.worldbank.org/indicator/NV.SRV.TETC.ZS/countries/1W?display=default>
- Yang, Y., & Han, J. (1996). Classification of and experimentation on tool interfacing in software development environments. *Software Engineering Conference, 1996. Proceedings., 1996 Asia-Pacific*. IEEE.
- Yu, J., Sheng, Q. Z., Swee, J. K., Han, J., Liu, C., & Noor, T. H. (2015). Model-driven development of adaptive web service processes with aspects and rules. *Journal of Computer and System Sciences*, 81(3), σσ. 533-552.
- Zhang, W.-J., & Li, Q. (1999). Information modelling for made-to-order virtual enterprise manufacturing systems. *Computer-Aided Design*, 31(10), σσ. 611-619.
- Zhao, Z.-z., & Cai, X. (2013). Research on Modeling Framework of Product Service System Based on Model Driven Architecture. *The 19th International Conference on Industrial Engineering and Engineering Management* (σσ. 1283-1290). Springer-Verlag Berlin Heidelberg.

Παράρτημα

Ενδεικτικοί Κανόνες μετασχηματισμού Acceleo (MOFM2T) UML προς Java

```
[comment encoding = UTF-8 /]
/**
 * The documentation of the module generate.
 */]
[module generate('http://www.eclipse.org/uml2/4.0.0/UML')]

/**
 * The documentation of the template generateElement:
 * This module is being used to develop transformation rules for MSEE GSDP
 enriched UML-to-Java transformations
 * The module builds class templates inc. methods and attributes, and is
 informed by generalisation
 * @param aClass
 */]

[template public generateElement(aClass : Class)]

[comment @main/]

[file (aClass.name.concat('.java'), false)]

/**
 * This file has been generated by MSEE Generic Service Development
 Platform: Transformation Rules Dev. Version
 */

import java.util.*;
import java.math.*;
//[protected ('for imports')]
//[protected]

/**
 * The documentation of the class [aClass.name/]
 *
 * @generated
 */

public class [aClass.name.toUpperFirst()] [if (not
(aClass.superClass->isEmpty()))]extends [aClass.superClass.name/] [if] {

[for (p: Property | aClass.attribute) separator('\n')]
private [p.type.name/] [p.name.toLowerFirst()];
[/for]

    public class [aClass.name.toUpperFirst()] {
        super();
    }

[for (o: Operation | aClass.ownedOperation) separator('\n')]
/**
 * The documentation of the method [o.name/].
```

```

    *
    * @generated
    */
    public void [o.name/]() {
        //[protected (o.name)]
        // TODO should be implemented
        //[protected]
    }
}
[/for]

[/for (p: Property | aClass.attribute) separator('\n')]
/**
 * The documentation of the getter get[p.name.toUpperFirst()/].
 *
 * @generated
 */
public [p.type.name/] get[p.name.toUpperFirst()/]() {
    return this.[p.name/];
}

/**
 * The documentation of the setter set[p.name.toUpperFirst()/].
 *
 * @generated
 */
public void set[p.name.toUpperFirst()/]([p.type.name/]
new[p.name.toUpperFirst()/]) {
    this.[p.name/] = new[p.name.toUpperFirst()/];
}
[/for]

}
[/file]
[/template]

```

