



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**Περιβάλλον Πλαίσιο για την Εξομοίωση Εταιρικών
Διαδικασιών με Παραμετροποιήσιμα Μοντέλα Κόστους**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΒΑΣΙΛΕΙΟΥ ΘΕΟΔΩΡΟΥ

Επιβλέπων : Κώστας Κοντογιάννης,
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2010



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Περιβάλλον Πλαίσιο για την Εξομοίωση Εταιρικών Διαδικασιών με Παραμετροποιήσιμα Μοντέλα Κόστους

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΒΑΣΙΛΕΙΟΥ ΘΕΟΔΩΡΟΥ

Επιβλέπων : Κώστας Κοντογιάννης
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 29^η Οκτωβρίου 2010.

.....
Κώστας Κοντογιάννης
Αν. Καθηγητής Ε.Μ.Π.

.....
Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

.....
Τιμολέον Σελλής
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2010

.....

ΒΑΣΙΛΕΙΟΣ ΘΕΟΔΩΡΟΥ

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

© 2010 – All rights reserved

Περίληψη

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η ανάπτυξη ενός περιβάλλοντος πλαισίου που καθιστά εφικτή την προσομοίωση επιχειρηματικών διαδικασιών με σκοπό την αξιολόγηση της συνολικής τους επίδοσης, του κόστους, και της ρυθμαπόδοσης για τις εναλλακτικές διαμορφώσεις που ενδέχεται να έχει μία συγκεκριμένη επιχειρηματική διαδικασία. Πιο συγκεκριμένα, οι δραστηριότητες των επιχειρηματικών διαδικασιών επισημαίνονται με ποσοτικοποιήσιμα χαρακτηριστικά που συμμορφώνονται σε κατάλληλα μετα-μοντέλα που επίσης ορίζονται στην παρούσα εργασία. Το προτεινόμενο πλαίσιο χρησιμοποιείται έπειτα σε ένα περιβάλλον που καλείται Διαχείριση Επιχειρηματικών Διαδικασιών (BPM - Business Process Management), που υποστηρίζει το σχεδιασμό, την εκτέλεση, τον έλεγχο και την ανάλυσή των διαδικασιών.

Η βασική λειτουργικότητα του συστήματος είναι η αποτίμηση και η αξιολόγηση των επιχειρηματικών διαδικασιών βασισμένη σε συγκεκριμένες καθοριζόμενες από το χρήστη επιλογές, όπως, συναρτήσεις χρησιμότητας, στρατηγικές αποτίμησης και, κριτήρια κατάταξης. Σε μια τυπική προσομοίωση και σενάριο αποτίμησης, ο χρήστης έχει τη δυνατότητα επιλογής των χαρακτηριστικών που τον ενδιαφέρουν για την αποτίμηση της διαδικασίας. Το σύστημα, με βάση τη δομή και τη ροή των διαδικασιών, αποτιμά διακριτά και εναλλακτικά μονοπάτια εκτέλεσης και διαμορφώσεις και παρουσιάζει τα συνολικά αποτελέσματα στο χρήστη, σύμφωνα με τις καθορισμένες στρατηγικές υπολογισμού και αποτίμησης που έχουν επιλεγεί.

Το σύστημα δομείται χρησιμοποιώντας το στυλ της διαστρωματωμένης (Layered) Αρχιτεκτονικής, με αρκετά στοιχεία Αντικειμενοστρέφειας (Object Orientation), όπου κάθε στιβάδα προσφέρει συγκεκριμένες υπηρεσίες χρησιμοποιώντας ένα σύνολο διακριτών και αλληλεπιδρώντων μονάδων (agents) που είναι καταναμημένες στα διάφορα επίπεδα (levels) και είναι υπεύθυνες για συγκεκριμένες λειτουργίες.

Τα δεδομένα εισόδου στο σύστημα αναπαρίστανται με τη μορφή μοντέλων επιχειρηματικών διαδικασιών, καθοριζόμενα από το χρήστη κριτήρια αξιολόγησης, και καθοριζόμενες από το χρήστη παραμέτρους διαμόρφωσης. Η προδιαγραφή και η μοντελοποίηση τέτοιων κριτηρίων γβασίζεται σε μεταμοντέλα που συμμορφώνονται το πρότυπο MOF. Από την άποψη αυτή, οι επιχειρηματικές διαδικασίες που προδιαγράφονται με διάφορα εργαλεία και περιβάλλοντα (π.χ. BPMN, Visio, BPEL) μπορούν να αντιστοιχιστούν στο κοινό μεταμοντέλο MOF, να επισημειωθούν με καθοριζόμενα από το χρήστη κριτήρια αξιολόγησης και, να προσομοιωθούν χρησιμοποιώντας το προτεινόμενο πλαίσιο.

Το βασικό πλεονέκτημα του προτεινόμενου περιβάλλοντος πλαισίου είναι α) η ευελιξία μοντελοποίησης που προσφέρει και η δυνατότητα υποστήριξης διαφορετικών πεδίων εφαρμογής μέσω της ομοιόμορφης προδιαγραφής μοντέλων επιχειρηματικών διαδικασιών· β) η προσαρμόσιμη δυνατότητα προσομοίωσης και αποτίμησης μέσω των καθοριζόμενων από το χρήστη στρατηγικών αξιολόγησης και κριτηρίων κατάταξης, και· γ) η επεκτασιμότητα που προσφέρει καθώς βασίζεται σε ανοικτά πρότυπα και τεχνολογίες θέσπισης επιχειρηματικών διαδικασιών, όπως MOF, BPEL και ODE, που επιτρέπουν τη μοντελοποίηση και την ανάλυση μιας ευρείας γκάμας επιχειρηματικών διαδικασιών.

Λέξεις Κλειδιά: Διαχείριση Επιχειρηματικών Διαδικασιών, Μοντελοποίηση Επιχειρηματικών Διαδικασιών, BPM, Αποτίμηση Επιχειρηματικών Διαδικασιών, KPIs, MOF, EMF, BPMN, BPEL

Abstract

The scope of this thesis is the design and development of a framework that allows for the simulation of business processes for the purpose of evaluating the overall performance, cost and, throughput of different alternative configurations a particular business process may be allowed to have. More specifically, business process tasks are annotated with quantifiable features that conform to appropriate meta-models that are also defined in this thesis. The proposed framework is then used within an environment called Business Process Management (BPM), that supports process design, execution, control and, analysis.

The main functionality of the proposed framework is the evaluation and assessment of business processes based on certain user-defined options such as, utility functions, evaluation strategies and, ranking criteria. In a typical simulation and evaluation scenario, the user is given the capability of choosing the features (KPIs) in which he/she is interested on evaluating the process. The system based on the structure and the flow of the processes, evaluates discrete and alternative execution paths and configurations and presents the overall results to the user, according to the specified estimation and evaluation strategies that have been chosen.

The system is structured utilizing the Layered Architecture style, including multiple features of Object Orientation, where each layer provides certain services using a set of discrete and interacting units (agents), which are distributed over the various levels and hold responsibility for certain operations.

The system's input data are represented by business process models, user defined evaluation criteria, and user defined configuration parameters. The specification and modeling of such criteria is based on metamodels conforming to the Meta-Object Facility standard (MOF). In this respect, business processes specified in various tools and environments (e.g. BPMN, Visio, BPEL) can be mapped to the common MOF meta-model, annotated with user defined evaluation criteria and, simulated by utilizing the proposed framework.

The main advantages of the proposed framework are a) the modeling flexibility it offers and the capability of supporting different application domains through the uniform specification of business process models; b) the customizable simulation and evaluation capability through the user defined evaluation strategies and ranking criteria, and; c) the extensibility it offers as it is based on open standards and business process enactment technologies such as MOF, BPEL and ODE that allow for the modeling and analysis of a wide range of business processes.

Keywords: Business Process Management, Business Process Modeling, BPM, Business Process Evaluation, KPIs, MOF, EMF, BPMN, BPEL

Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1	Περιγραφή Πλαισίου του Προβλήματος.....	1
1.2	Αντικείμενο διπλωματικής.....	3
1.2.1	Συνεισφορά.....	4
1.3	Οργάνωση κειμένου.....	4
2	Σχετικές Εργασίες.....	5
2.1	Μοντελοδηγούμενη Μηχανίκευση (Model Driven Engineering).....	5
2.1.1	<i>MOF - Meta-Object Facility</i>	8
2.1.2	<i>EMF - Eclipse Modelling Framework</i>	11
2.2	Δείκτες-Κλειδιά για την Επίδοση (Key Performance Indicators).....	15
2.3	Αποτίμηση Επιχειρηματικών Διαδικασιών (Business Process Evaluation)	17
2.3.1	<i>Business Process Modeling Notation (BPMN)</i>	19
2.3.2	<i>Business Process Execution Language (BPEL)</i>	23
3	Αρχιτεκτονική Συστήματος.....	27
3.1	Αρχιτεκτονική Αναφοράς/ Σχηματικό Διάγραμμα	27
3.2	Συστατικά μέρη (Components).....	31
3.2.1	<i>Περιγραφές</i>	31
3.2.2	<i>Διεπαφές (Interfaces) και Υπηρεσίες (Services)</i>	34
3.3	Περιγραφή Διαδικασίας(Process Outline)	37
4	Προδιαγραφή Μοντέλων.....	43
4.1	Σκιαγράφηση των Μοντέλων (Model Profiling)	44
4.2	Συγκεκριμένα Μοντέλα (Concrete Models)	48
4.2.1	<i>Μοντέλο Κόστους (Cost Model)</i>	48
4.2.2	<i>Μοντέλο Εσόδων (Revenue Model)</i>	50
4.2.3	<i>Μοντέλο Αξιοπιστίας (Dependability Model)</i>	52
4.2.4	<i>Μοντέλο Ρυθμαπόδοσης (Throughput Model)</i>	54
4.2.5	<i>Μοντέλο Χρησιμοποίησης Πόρων (Resource Utilization Model)</i>	56

4.2.6	Μοντέλο Προσπάθειας (<i>Effort Model</i>).....	58
4.2.7	Μοντέλο Διαρκείας βασισμένο στην Προσπάθεια (<i>Effort-Based Duration Model</i>) 61	
4.2.8	Μοντέλο Ποιότητας (<i>Quality Model</i>).....	63
5	Αποτίμηση Επιχειρηματικών Διαδικασιών (<i>Business Process Evaluation</i>)	67
5.1	Ανάλυση διαδρομών (<i>Path Analysis</i>)	68
5.1.1	Εισαγωγικό Μέρος.....	68
5.1.2	Αλγόριθμος.....	72
5.1.3	Ψευδοκώδικας Αλγορίθμου	73
5.2	Στρατηγική Αποτίμησης (<i>Evaluation Strategy</i>)	75
5.2.1	Εισαγωγικό Μέρος.....	76
5.2.2	Αλγόριθμος.....	78
5.2.3	Ψευδοκώδικας Αλγορίθμου	78
5.3	Σύνθεση της Στρατηγικής Αποτίμησης (<i>Composition of Evaluation Strategy</i>)	80
5.3.1	Εισαγωγικό Μέρος.....	80
5.3.2	Αλγόριθμος.....	82
5.3.3	Ψευδοκώδικας Αλγορίθμου	83
6	Μελέτες Περιπτώσεων (<i>Case Studies</i>)	85
6.1	Παράδειγμα (Α) : Υπολογισμός Κόστους.....	86
6.2	Λήψη Αποφάσεων	89
6.2.1	Παράδειγμα (Β) : Αξιολόγηση Υπηρεσιών.....	89
6.2.2	Παράδειγμα (Γ) : Επενδυτική Τακτική	92
6.3	Παράδειγμα (Δ) : Εύρεση Χωρητικότητας (<i>Capacity</i>) και Στενωπής (<i>Bottleneck</i>) Συστήματος.....	95
6.4	Παράδειγμα (Ε) : Εξέταση Επίδοσης του Συστήματος.....	98
7	Επίλογος	101
7.1	Σύνοψη και συμπεράσματα.....	101
7.2	Μελλοντικές επεκτάσεις	103
8	Βιβλιογραφία	105

1

Εισαγωγή

1.1 Περιγραφή Πλαισίου του Προβλήματος

Από τις αρχές του 1990 μέχρι σήμερα οι επιχειρηματικές διαδικασίες (business processes) και ο σχεδιασμός τους έχουν κεντρίσει το ενδιαφέρον σχεδόν κάθε επιχείρησης. Η ικανότητα συντονισμού των επιχειρηματικών διαδικασιών με έναν όσο το δυνατόν πιο αποδοτικό και ευέλικτο τρόπο έχει αναχθεί στον κρίσιμο παράγοντα για την επιτυχία των σύγχρονων εταιριών. Ο προσανατολισμός προς τις επιχειρηματικές διαδικασίες και η ανάγκη ενασχόλησης με αυτές έχει προκαλέσει μια ολοένα και αυξανόμενη ανάγκη για κατάλληλες τεχνικές και εργαλεία για την αναγνώριση (identification), ανάλυση (analysis) και προσομοίωσή τους (simulation). Η βάση για όλες αυτές τις λειτουργίες είναι τα μοντέλα των επιχειρηματικών διαδικασιών (business process models).

Η προσομοίωση διαδικασιών και γενικότερα πεδίων εφαρμογής με τη χρήση μοντέλων αποτελεί μέρος της σύγχρονης τάσης που προωθεί τη μοντελοποίηση ως τη βασική δραστηριότητα της ανάπτυξης λογισμικού και επιδιώκει τη μετατόπιση από την κλασική κωδικο-κεντρική σε μια μοντελο-κεντρική προσέγγιση. Συγκεκριμένα, τα μοντέλα των επιχειρηματικών διαδικασιών διαδραματίζουν σημαντικό ρόλο σε όλα τα διαφορετικά στάδια του επιχειρησιακού σχεδιασμού (design) ανεξάρτητα από την τεχνολογία που χρησιμοποιείται. Βασικός σκοπός της μοντελοποίησης των διαδικασιών είναι ο ορισμός και η επικοινωνία των στόχων και των στρατηγικών των εταιριών και μόλις αυτά πραγματοποιηθούν, αρχίζει η εργασία για την επίτευξη βελτίωσης. Δυστυχώς, αν και γίνονται πολλές προσπάθειες προς αυτή την κατεύθυνση, είναι γενικά αποδεκτό ότι η εφαρμογή του αποτελεσματικού ανασχεδιασμού των επιχειρηματικών διαδικασιών βρίσκεται ακόμα σε πρώιμο στάδιο. Έχουν προταθεί διάφορες αρχές και οδηγίες αλλά είναι έκδηλη η ανάγκη περαιτέρω ορισμού και προτυποποίησης μεθόδων για μια περισσότερο συστηματική προσέγγιση και για την αποφυγή απρόβλεπτων αποτελεσμάτων.

Όποια προσέγγιση και να ακολουθηθεί, απαραίτητο συστατικό για τον αποτελεσματικό σχεδιασμό των επιχειρηματικών διαδικασιών αποτελεί η ανάπτυξη μοντέλων που παρέχουν κατάλληλες απεικονίσεις της γνώσης που χρειάζεται για την κατανόηση και τον συλλογισμό (reasoning) τους. Είναι εξαιρετικά σημαντικός και δύσκολος ο ορισμός τέτοιων μοντέλων, που συχνά καλούνται μοντέλα κόστους (cost models), με γνώμονα την εστίαση στις παραμέτρους εκείνες των διαδικασιών που παρουσιάζουν το μεγαλύτερο ενδιαφέρον και προσφέρουν στις διαδικασίες τις περισσότερες δυνατότητες βελτίωσης. Οι άνθρωποι που κατέχουν τη γνώση, την εμπειρία και τις δεξιότητες που σχετίζονται με τον εντοπισμό των κρίσιμων μεγεθών που αφορούν στις επιχειρηματικές διαδικασίες και με τη λήψη σημαντικών αποφάσεων που σχετίζονται με τη διεξαγωγή τους είναι συνήθως οι αναλυτές (business analysts) και οι διευθυντές των επιχειρήσεων (business managers).

Μία από τις δυσκολίες που παρουσιάζονται στο σύγχρονο κόσμο των επιχειρήσεων είναι η επικοινωνία όλων των ανθρώπων του, συμπεριλαμβανομένων τόσο των αναλυτών και των διευθυντών επιχειρήσεων όσο και των τεχνικών προγραμματιστών (technical developers), με βάση συγκεκριμένα πρότυπα και τεχνολογίες. Τα εμπόδια για μια τέτοιου είδους συνεννόηση οφείλονται κατά κύριο λόγο στον καταμερισμό γνώσης και δεξιοτήτων που επικρατεί στον επιχειρηματικό κόσμο. Έτσι, ενώ οι αναλυτές και οι διευθυντές των επιχειρήσεων είναι σε θέση να προσεγγίσουν τις διαδικασίες από τη σωστή σκοπιά και να εστιάσουν στα κρίσιμα χαρακτηριστικά τους, συνήθως στερούνται γνώσεων και ικανοτήτων για τον αποδοτικό σχεδιασμό συστημάτων για τη μοντελοποίηση και την προσομοίωση των επιχειρηματικών διαδικασιών, τις οποίες κατέχουν οι τεχνικοί προγραμματιστές.

Για το σκοπό αυτό έχουν προτυποποιηθεί διάφορα εργαλεία σχεδιασμού και μοντελοποίησης, από τα οποία άλλα επικεντρώνουν στη γραφική απεικόνιση και άλλα στην εκτελεσιμότητα των επιχειρηματικών διαδικασιών. Αν και αυτά τα εργαλεία προσφέρουν πλήθος δυνατοτήτων και λόγω της απλότητάς τους έχουν συμβάλει αποφασιστικά στην επικοινωνία και τη διαλειτουργικότητα, εντούτοις υπάρχουν ακόμη πολλές εκκρεμότητες, όπως η προτυποποιημένη συνεργασία μεταξύ των διαφόρων εργαλείων και ο καλύτερος συμβιβασμός του επιπέδου περιγραφικότητας με την εκτελεσιμότητα των διαδικασιών, αν θέλουμε κάποια μέρα να μιλάμε για πλήρη αυτοματισμό στις διαδικασίες της ανάλυσης και της βελτίωσης των επιχειρηματικών διαδικασιών

Μία επίσης δυσκολία που παρουσιάζεται κατά τη μοντελοποίηση των επιχειρηματικών διαδικασιών είναι η έλλειψη εργαλείων που να προσφέρουν τη δυνατότητα αντιστοίχισης, στις διαδικασίες, ευέλικτων χαρακτηριστικών με ειδικότητα πεδίου. Αν και υπάρχει πληθώρα εργαλείων για την αναπαράσταση των χαρακτηριστικών των επιχειρηματικών διαδικασιών, συνήθως, είτε αδυνατούν να επικεντρωθούν στον προσανατολισμό προς τις επιχειρηματικές διαδικασίες, είτε προσφέρουν πολύ περιορισμένο βαθμό εκφραστικότητας και δυνατότητας

εξειδίκευσης πεδίου. Επιπλέον, τα διάφορα χαρακτηριστικά προσδίδονται σε συγκεκριμένο αφαιρετικό επίπεδο των επιχειρηματικών διαδικασιών, και έτσι καθίσταται αδύνατη η διασύνδεση παραμέτρων μεταξύ διαφορετικών επιπέδων αφαίρεσης.

1.2 Αντικείμενο διπλωματικής

Στην προηγούμενη ενότητα αναφέρθηκαν ορισμένες από τις δυσκολίες και τις προκλήσεις που εμφανίζονται στον κόσμο της ανάλυσης επιχειρηματικών διαδικασιών. Αντικείμενο της παρούσας διπλωματικής είναι η μελέτη υπαρχόντων προτύπων και τεχνολογιών για την προσέγγιση αυτών των θεμάτων και ο σχεδιασμός ενός περιβάλλοντος πλαισίου για την προσομοίωση επιχειρηματικών διαδικασιών σε συνδυασμό με τον εμπλουτισμό τους με ευέλικτα ποιοτικά χαρακτηριστικά.

Συγκεκριμένα, πραγματοποιείται η ανάλυση ενός συστήματος προσομοίωσης που δέχεται ως είσοδο πληροφορίες για τις επιχειρηματικές διαδικασίες και τα χαρακτηριστικά αυτών. Το ζητούμενο είναι η ενοποιημένη αναπαράσταση των επιχειρηματικών διαδικασιών σε μια συγκεκριμένη, προκαθορισμένη και επεξεργάσιμη από το σύστημα μορφή, με σκοπό την προσομοίωσή τους και την εξαγωγή συμπερασμάτων. Η αναπαράσταση αυτή πρέπει να περιέχει τόσο πληροφορίες για τη λειτουργική δομή και τη ροή των διαδικασιών, όσο και για τις διάφορες παραμέτρους που αναφέρονται στα ποιοτικά χαρακτηριστικά των στοιχείων αυτών των διαδικασιών. Για τους λόγους αυτούς, στο σχεδιαζόμενο σύστημα, οι πληροφορίες για τις διαδικασίες αυτές και τα χαρακτηριστικά τους αναπαρίστανται με τη μορφή μοντέλων. Η μοντελοποίηση των ποιοτικών χαρακτηριστικών γίνεται με γνώμονα συγκεκριμένα πρότυπα που καθορίζουν τα ουσιώδη στοιχεία, τη σύνταξη και τη δομή των μεταμοντέλων που χρησιμοποιούνται για την κατασκευή αντικειμενοστρεφών μοντέλων (MOF). Παρέχεται λοιπόν η δυνατότητα ορισμού ευέλικτων, λεπτομερών και με ειδικότητα πεδίου (domain-specific) μοντέλων για τα χαρακτηριστικά των διαδικασιών, αντιμετωπίζοντας έτσι το πρόβλημα της ελλιπούς εκφραστικότητας των υπαρχόντων συστημάτων.

Καθορίζεται λοιπόν, με βάση τις λειτουργικές του απαιτήσεις, η αρχιτεκτονική ενός τέτοιου συστήματος σε διάφορα επίπεδα. Οι βασικές λειτουργίες του συστήματος είναι το «διάβασμα» των μοντέλων εισόδου, η μετατροπή τους σε μια κοινώς επεξεργάσιμη μορφή και με βάση συγκεκριμένες επιλογές, όπως στρατηγικές και κριτήρια, η αξιολόγηση των επιχειρηματικών διαδικασιών που αντιπροσωπεύουν. Αφού καθοριστούν τα συστατικά μέρη και οι διεπαφές αυτού του περιβάλλοντος πλαισίου και αναπτυχθούν αλγόριθμοι για τις λειτουργίες του, επιχειρείται μία υλοποίηση των βασικών του συστατικών με τη χρήση σύγχρονων τεχνολογιών, όπως του πλαισίου EMF. Υλοποιείται λοιπόν ένα πλαίσιο, δίνοντας μεγάλη βαρύτητα στη χρήση σχεδιαστικών προτύπων και προγραμματιστικών αρχών από την

Τεχνολογία Λογισμικού, όπως οι αρχιτεκτονικές τεχνοτροπίες και τα σχεδιαστικά μορφήματα. Αυτό συντελεί όχι μόνο στην αναγνωσιμότητα και συντηρησιμότητα του παραγόμενου κώδικα, αλλά μεταξύ άλλων και στην ευελιξία και επεκτασιμότητα του συστήματος.

1.2.1 Συνεισφορά

Η συνεισφορά της παρούσας διπλωματικής συνοψίζεται στα εξής σημεία :

1. Μελέτη υπαρχόντων προτύπων και τεχνολογιών για τη μοντελοποίηση συστημάτων και την ανάλυση επιχειρηματικών διαδικασιών.
2. Σχεδιασμός και ανάπτυξη, χρησιμοποιώντας το στυλ της διαστρωματωμένης (Layered) Αρχιτεκτονικής, περιβάλλοντος πλαισίου προσομοίωσης και αξιολόγησης επιχειρηματικών διαδικασιών: α) σχεδιασμός αρχιτεκτονικής συστήματος, όπου κάθε στιβάδα προσφέρει συγκεκριμένες υπηρεσίες για την προσομοίωση, ανάλυση και αξιολόγηση, και β) ανάλυση συστατικών μερών και διεπαφών του.
3. Καθορισμός και σκιαγράφηση (profiling) ευέλικτων, περιγραφικών και με ειδικότητα πεδίου μοντέλων κρίσιμων χαρακτηριστικών για την επισημείωση επιχειρηματικών διαδικασιών με ποσοτικοποίησιμα ποιοτικά στοιχεία.
4. Καθορισμός μορφής επιχειρηματικών διαδικασιών για την ανάλυση και αποτίμησή τους, βασισμένης στο σχεδιαστικό μοτίβο Composite.
5. Ανάπτυξη αλγορίθμων για την ανάλυση, αποτίμηση και αξιολόγηση επιχειρηματικών διαδικασιών

1.3 Οργάνωση κειμένου

Η παρούσα εργασία απαρτίζεται από 8 Κεφάλαια. Το Κεφάλαιο 1 αποτελεί μια εισαγωγή στο αντικείμενο της διπλωματικής. Το Κεφάλαιο 2 παρουσιάζει στον αναγνώστη το θεωρητικό υπόβαθρο και τις σχετικές τεχνολογίες για το αντικείμενο της παρούσας εργασίας. Το Κεφάλαιο 3 περιλαμβάνει το σχεδιασμό της Αρχιτεκτονικής του Συστήματος και την περιγραφή των συστατικών της. Στο Κεφάλαιο 4 προδιαγράφονται τα μοντέλα που χρησιμοποιούνται από το σύστημα. Στο Κεφάλαιο 5 αναπτύσσονται (και με τη μορφή ψευδοκώδικα) οι αλγόριθμοι αποτίμησης των επιχειρηματικών διαδικασιών. Το Κεφάλαιο 6 περιλαμβάνει μελέτες περιπτώσεων για την επίδειξη πιθανών εφαρμογών του συστήματος. Στο Κεφάλαιο 7 συνοψίζονται τα αποτελέσματα της παρούσας εργασίας και προτείνονται κάποιες μελλοντικές επεκτάσεις για το σύστημα. Τέλος, στο Κεφάλαιο 8 συγκεντρώνεται η βιβλιογραφία που χρησιμοποιήθηκε για τη συγγραφή της εργασίας.

2

Σχετικές Εργασίες

Σε αυτό το κεφάλαιο γίνεται μία σύντομη εισαγωγή στις θεματικές περιοχές και τις τεχνολογίες που άπτονται του αντικείμενου της παρούσας εργασίας, η κατανόηση των οποίων κρίνεται απαραίτητη για τον αναγνώστη. Στην ενότητα 2.1 γίνεται μία εισαγωγή στην έννοια της Μοντελοδηγούμενης Μηχανίκευσης (Model Driven Engineering) και σε παραγράφους αυτής παρουσιάζονται το πρότυπο Meta-Object Facility (MOF) και η τεχνολογία Eclipse Modelling Framework (EMF), που είναι άμεσα συνδεδεμένες με αυτήν την έννοια. Στην επόμενη ενότητα (2.2) γίνεται παρουσίαση των Δεικτών – Κλειδιών για την Επίδοση (KPIs), όπου αναφέρεται ο ορισμός τους και τα βασικά τους χαρακτηριστικά. Τέλος, στην ενότητα 2.3 γίνεται μία εισαγωγή στην Αποτίμηση των Επιχειρηματικών Διαδικασιών (Business Process Evaluation), όπου πραγματοποιείται μία σύντομη ιστορική αναδρομή στην εξέλιξη των μεθόδων ανάλυσης και αξιολόγησης των επιχειρηματικών διαδικασιών και η αναφορά σύγχρονων τέτοιων μεθόδων. Συγκεκριμένα, σε παραγράφους αυτής της ενότητας περιγράφονται η Business Process Modelling Notation (BPMN) και η Business Process Execution Language (BPEL), που αποτελούν σήμερα τα επικρατέστερα εργαλεία μοντελοποίησης επιχειρηματικών διαδικασιών.

2.1 Μοντελοδηγούμενη Μηχανίκευση (Model Driven Engineering)

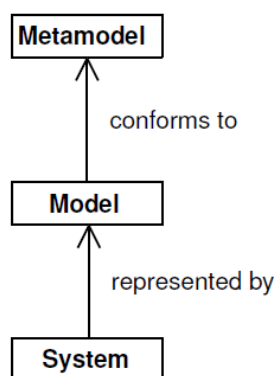
Ανέκαθεν, από τότε που οι άνθρωποι ξεκίνησαν να χρησιμοποιούν υπολογιστές, διεξάγονται έρευνες για την ανύψωση του επιπέδου αφάιρεσης (abstraction) στο οποίο οι μηχανικοί λογισμικού γράφουν προγράμματα [1]. Έτσι, έχουν πραγματοποιηθεί διάφορες μετατοπίσεις στον τρόπο προγραμματισμού, όπως για παράδειγμα από τον κώδικα μηχανής στον προγραμματισμό με συμβολομεταφραστή (assembler programming) και από τον προστακτικό (imperative) στον αντικειμενοστρεφή (object-oriented) προγραμματισμό.

Η σύγχρονη τάση προς αυτή την κατεύθυνση είναι η χρησιμοποίηση μοντέλων και γλωσσών μοντελοποίησης στις διαδικασίες της ανάπτυξης λογισμικού. Η υπόσχεση αυτής της τάσης είναι η μετατόπιση της εστίασης από την υλοποίηση στο σχεδιασμό (design).

Τα μοντέλα εξυπηρετούν ως μηχανισμοί για την καλύτερη κατανόηση, για παράδειγμα για τη σκιαγράφηση (sketching) του αρχιτεκτονικού σχεδιασμού ή την αρχειοθέτηση μίας υπάρχουσας υλοποίησης. Πέρα όμως από αυτή τη χρήση, μπορούν επίσης να αποτελέσουν είσοδο (input) για την παραγωγή κώδικα (code generation) . Έτσι, η νεότερη τάση στη τεχνολογία λογισμικού (software engineering) θεωρεί τα μοντέλα πρώτης τάξεως οντότητες για τη διαδικασία ανάπτυξης, οι οποίες χρησιμοποιούνται για την αυτόματη παραγωγή (μέρους των) συστημάτων λογισμικού μέσω των μετασχηματισμών από-μοντέλο-σε-μοντέλο (model-to-model transformations) και από-μοντέλο-σε-κώδικα (model-to-code transformations). Αυτή η τάση έχει οδηγήσει σε έναν κλάδο της τεχνολογίας λογισμικού (software engineering) που προωθεί τη μοντελοποίηση ως τη βασική δραστηριότητα της ανάπτυξης λογισμικού και επιδιώκει τη μετατόπιση από την κλασική κωδικο-κεντρική σε μια μοντελο-κεντρική προσέγγιση. Στη βιβλιογραφία, αυτός ο κλάδος αναφέρεται ως Μοντελοδηγούμενη Μηχανίευση (Model Driven Engineering), Μοντελοδηγούμενη Ανάπτυξη (Model Driven Development) και Μοντελοδηγούμενη Ανάπτυξη Λογισμικού (Model Driven Software Development) [2] .

Κεντρική θέση στην Μοντελοδηγούμενη Μηχανίευση (MDE) κατέχει η έννοια του μοντέλου (model). Σύμφωνα με τα Διαδικτυακά Λεξικά του Κέμπριτζ (Cambridge Dictionaries Online) [3], ένας από τους ορισμούς της λέξης «μοντέλο» είναι «μία αναπαράσταση κάποιου αντικειμένου, είτε ως ενός φυσικού αντικειμένου το οποίο είναι συνήθως μικρότερο από το πραγματικό αντικείμενο, είτε ως μίας απλής περιγραφής του αντικειμένου που μπορεί να χρησιμοποιηθεί σε υπολογισμούς». Για την τεχνολογία λογισμικού (software engineering), το μοντέλο είναι μία αφάιρηση, με την έννοια ότι αναπαριστά όσες από τις πτυχές και τις ιδιότητες ενός πραγματικού συστήματος είναι σχετικές στο δοθέν πλαίσιο (context) [4]. Τα μοντέλα χρησιμοποιούνται για την αντιμετώπιση της πολυπλοκότητας του λογισμικού, δίνοντας τη δυνατότητα στους προγραμματιστές να πραγματεύονται και να ασχολούνται με ένα σύστημα λογισμικού σε ένα υψηλότερο επίπεδο αφάιρησης. Στην τεχνολογία λογισμικού (software engineering) τα μοντέλα είναι συνήθως διαγραμματικά και αναπαρίστανται με τη βοήθεια γλωσσών μοντελοποίησης, όπως η Ενοποιημένης Γλώσσας Μοντελοποίησης (UML – Unified Modeling Language) [5].

Επίσης σημαντική θέση στην Μοντελοδηγούμενη Μηχανίκευση (MDE) κατέχει η έννοια του μετα-μοντέλου (meta-model) . Ο όρος «μετα-μοντελοποίηση» μπορεί να οριστεί ως «η πράξη και επιστήμη της μηχανίκευσης μετα-μοντέλων» [6]. Ωστόσο, ο ακριβής ορισμός του όρου «μετα-μοντέλο» αποτελεί πηγή πληθώρας συζητήσεων και διαφωνιών [4]. Εννοιολογικά, το πρόθεμα «μετα» υπονοεί πως η μοντελοποίηση έχει συμβεί δυο φορές, το οποίο αντανακλάται στον ορισμό : «το μετα-μοντέλο είναι ένα μοντέλο μοντέλων» [7]. Συνεπώς, ένα μετα-μοντέλο περιορίζει το σύνολο των υποστάσεων του (instances) με τον ίδιο τρόπο που ένα μοντέλο περιορίζει το σύνολο των υποστάσεων του (instances) [6]. Τεχνικά, ένα μετα-μοντέλο προσδιορίζει την αφηρημένη σύνταξη μιας γλώσσας μοντελοποίησης, η οποία ορίζει ένα σύνολο από έννοιες μοντελοποίησης, τα χαρακτηριστικά τους (attributes) και τις μεταξύ τους σχέσεις, καθώς επίσης και κανόνες για το συνδυασμό αυτών των μοντέλων για τον καθορισμό έγκυρων μοντέλων [5]. Στο Σχήμα 2.1 απεικονίζονται οι θεμελιώδεις σχέσεις μεταξύ συστήματος, μοντέλου και μεταμοντέλου.



Σχήμα 2.1: Θεμελιώδεις σχέσεις μεταξύ συστήματος (system), μοντέλου (model) και μεταμοντέλου (metamodel) [4]

Το βασικό πλεονέκτημα της Μοντελοδηγούμενης Μηχανίκευσης είναι η ενίσχυση της παραγωγικότητας και της ποιότητας, αυτοματοποιώντας επαναλαμβανόμενα, επιρρεπή σε λάθη και χρονοβόρα έργα (tasks). Επιπλέον, εκμεταλλευόμενη την αφαίρεση (abstraction) και την ειδικότητα-πεδίου (domain-specificity) βελτιώνει την επικοινωνία και διευκολύνει το διαχωρισμό της επιχειρηματικής λογικής (business logic) από τις τεχνολογίες εφαρμογής (application technologies). Οι υποδομές μηχανοργάνωσης (Computing infrastructures) συνεχώς επεκτείνονται προς κάθε κατεύθυνση σε απάντηση των επιχειρηματικών αναγκών και της τεχνολογικής ανάπτυξης. Με την υιοθέτηση της Μοντελοδηγούμενης Μηχανίκευσης, η επιχειρηματική λογική (business logic) και οι τεχνολογίες εφαρμογής (application technologies) μπορούν να εξελιχθούν ανεξάρτητα μεταξύ τους και οι οργανισμοί είναι σε θέση να ενσωματώσουν υπάρχοντα συστήματα με αυτά που οικοδομούνται μελλοντικά [8].

Αυτή τη στιγμή, η βιομηχανική εφαρμογή που προσεγγίζει τη Μοντελοδηγούμενη Μηχανίκευση (MDE) είναι η *Μοντελοδηγούμενη Αρχιτεκτονική* (MDA - Model Driven Architecture), η οποία εισήχθη από το OMG (Object Management Group) [8] [9] [7]. Στη Μοντελοδηγούμενη Αρχιτεκτονική (MDA) , ανεξαρτήτου-πλατφόρμας μοντέλα (PIMs - *platform-independent models*) εκφράζονται αρχικά σε μία ανεξαρτήτου-πλατφόρμας γλώσσα μοντελοποίησης (*platform-independent modeling language*) . Το μοντέλο αυτό, ακολούθως, μεταφράζεται σε ένα συγκεκριμένης-πλατφόρμας μοντέλο (PSM - *platform-specific model*), αντιστοιχίζοντας το ανεξαρτήτου-πλατφόρμας μοντέλο (PIM) σε κάποια συγκεκριμένη γλώσσα ή πλατφόρμα υλοποίησης, χρησιμοποιώντας τυπικούς κανόνες.

Οι βασικές ιδέες της Μοντελοδηγούμενης Αρχιτεκτονικής (MDA) είναι στενά συνδεδεμένες [4] με τον γεννητικό (*generative*) προγραμματισμό, εργοστάσια λογισμικού (*software factories*) [7], γλώσσες μοντελοποίησης με ειδικότητα-πεδίου (*domain-specific modelling languages*) [7] κ.α..

Η Μοντελοδηγούμενη Αρχιτεκτονική (MDA) βασίζεται σε ένα πλήθος προτύπων (*standards*), συμπεριλαμβανομένων της Ενοποιημένης Γλώσσας Μοντελοποίησης (UML - *Unified Modeling Language*) [5], του Meta-Object Facility (MOF) [10], στο οποίο θα αναφερθούμε στη συνέχεια, του XML Metadata Interchange (XMI) [11] και του Common Warehouse Metamodel (CWM) [12].

2.1.1 MOF - Meta-Object Facility

Το Meta Object Facility (MOF) είναι ένα πρότυπο του OMG που ορίζει μία κοινή, αφηρημένη γλώσσα για την προδιαγραφή μεταμοντέλων [10]. Το MOF είναι ένα παράδειγμα ενός μεταμοντέλου, ή μοντέλου του μεταμοντέλου (που κάποιες φορές αποκαλείται οντολογία). Το MOF είναι σαφώς αντικειμενοστρεφές από τη φύση του. Καθορίζει τα ουσιώδη στοιχεία, τη σύνταξη και τη δομή των μεταμοντέλων που χρησιμοποιούνται για την κατασκευή αντικειμενοστρεφών μοντέλων διακριτών συστημάτων.

Το MOF εξυπηρετεί ως το κοινό μοντέλο τόσο του CWM [12] όσο και των μεταμοντέλων σε UML [5].

Συγκεκριμένα, η προδιαγραφή MOF περιλαμβάνει:

- Ένα αφηρημένο μοντέλο των γενικών (*generic*) αντικειμένων MOF και των συσχετίσεών τους
- Ένα σύνολο κανόνων για τη αντιστοίχιση οποιουδήποτε βασισμένου-σε-MOF μεταμοντέλου σε ανεξαρτήτου-γλώσσας διεπαφές (*language-independent interfaces*). Μια υλοποίηση αυτών των διεπαφών για ένα δοθέν μεταμοντέλο θα μπορούσε να

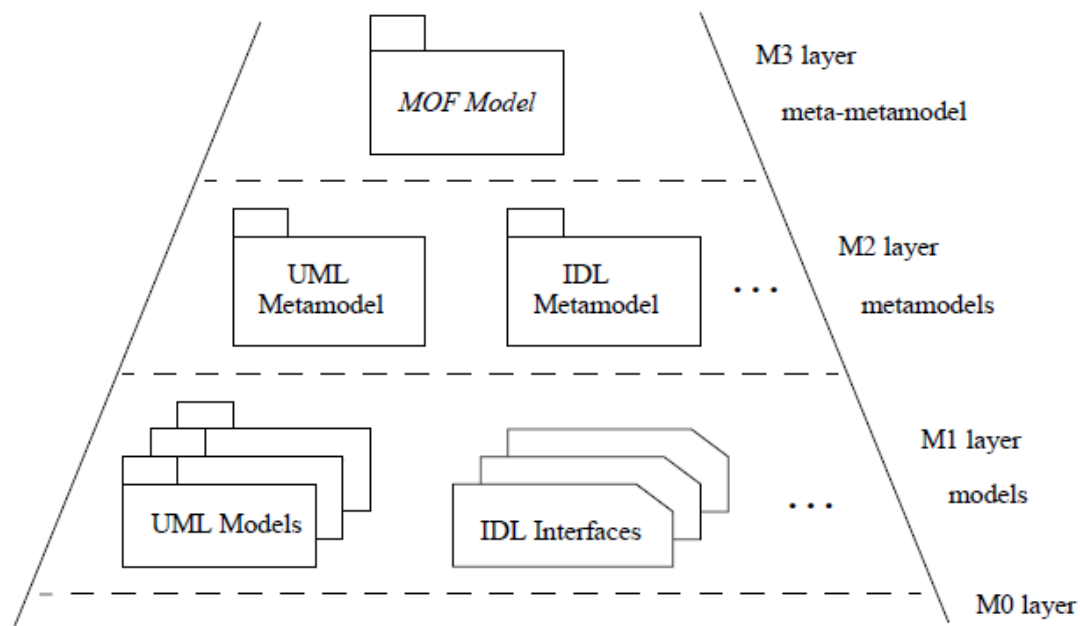
χρησιμοποιηθεί για την πρόσβαση σε και την τροποποίηση οποιουδήποτε μοντέλου βασισμένου σε εκείνο το μεταμοντέλο.

- Κανόνες που καθορίζουν τη διάρκεια του κύκλου ζωής, τη σύνθεση και σημασιολογίες τερματισμού στοιχείων των βασισμένων-σε-MOF μεταμοντέλων.
- Μια ιεραρχία από ανακλαστικές (reflective) διεπαφές. Αυτές ορίζουν γενικές λειτουργίες (generic operations) για την ανακάλυψη και το χειρισμό μοντέλων βασισμένων σε συμμορφούμενα-σε-MOF μεταμοντέλα, αλλά των οποίων οι καθορισμένες διεπαφές είναι άγνωστες.

Σύμφωνα με το όραμα του OMG για τη Μοντελοδηγούμενη Μηχανίκευση (MDE), τα μοντέλα, οι γλώσσες μοντελοποίησης και οι γλώσσες μεταμοντελοποίησης οργανώνονται σε τέσσερα επίπεδα M0-M3 στη λεγόμενη ιεραρχία τεσσάρων στιβάδων του OMG [13]. Η περισσότερο αποδεκτή ερμηνεία της ιεραρχίας αυτής είναι η εξής:

- Το επίπεδο M0 περιέχει πρωτότυπα (originals) , π.χ. τη «Λάση» (Lassie) του πραγματικού κόσμου
- Το επίπεδο M1 περιέχει μοντέλα, για παράδειγμα ένα UML διάγραμμα κλάσεων και μια επεξήγηση των στιγμιotypών του.
- Το επίπεδο M2 περιέχει μεταμοντέλα, για παράδειγμα το μεταμοντέλο της UML
- Το επίπεδο M3 περιέχει το μετα-μεταμοντέλο MOF.

Τα παραπάνω φαίνονται και στο Σχήμα 2.2.

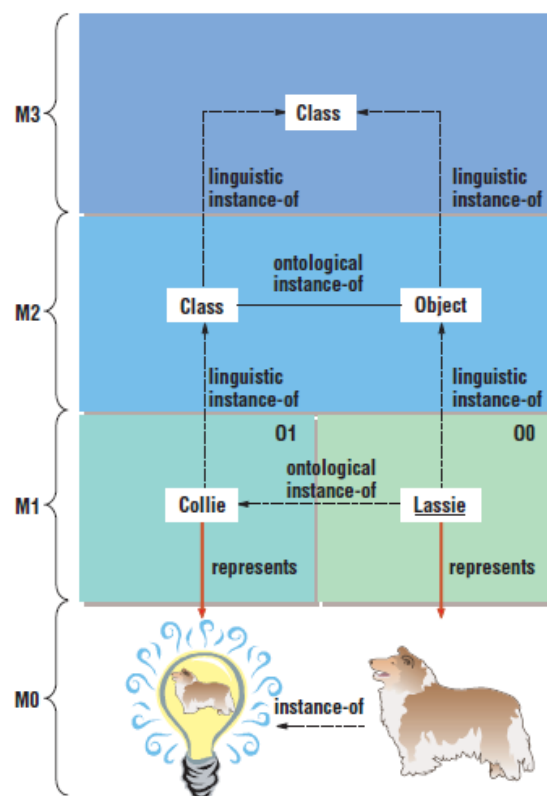


Σχήμα 2.2: Η Αρχιτεκτονική Μεταδεδομένων του MOF [13]

Έχουμε τις εξής σχέσεις μεταξύ των επιπέδων [13]:

- Ένα πρωτότυπο στο επίπεδο M0 αντιπροσωπεύεται από ένα μοντέλο στο επίπεδο M1
- Ένα μοντέλο στο επίπεδο M1 συμμορφώνεται με ένα μεταμοντέλο στο επίπεδο M2
- Ένα μεταμοντέλο στο επίπεδο M2 συμμορφώνεται με ένα μετα-μεταμοντέλο MOF στο επίπεδο M3
- Το MOF συμμορφώνεται στον εαυτό του, δηλαδή είναι αυτο-οριζόμενο (self-defined) ή αλλιώς αντανακλαστικό (reflexive).

Στην ιεραρχία τεσσάρων στιβάδων του OMG, τα UML διαγράμματα αντικειμένων και τα UML διαγράμματα κλάσεων βρίσκονται στο ίδιο επίπεδο, το M1, παρόλο που τα πρώτα μπορούν να θεωρηθούν μοντέλα που συμμορφώνονται στα δεύτερα. Εφόσον λοιπόν τα UML διαγράμματα αντικειμένων βρίσκονται στο επίπεδο M, θεωρούνται μοντέλα που συμμορφώνονται στο μεταμοντέλο. Αυτή η ερμηνεία της ιεραρχίας τεσσάρων στιβάδων που υιοθετεί το MOF θεωρεί ότι οι γλωσσικές σχέσεις στιγμιότυπο-του (instance-of) διαπερνούν (και αποτελούν τη βάση για) γλωσσικά μεταεπίπεδα (linguistic metalevels), ενώ οι αντίστοιχες οντολογικές (ontological) συσχετίζουν οντότητες εντός του ίδιου επιπέδου.



Σχήμα 2.3: Η γλωσσική σκοπιά της μεταμοντελοποίησης [1].

Με άλλα λόγια, κάθε επίπεδο από το M1 και πάνω θεωρείται ένα μοντέλο που εκφράζεται σε μία γλώσσα που ορίζεται στο αμέσως παραπάνω επίπεδο από αυτό[1]. Αυτή είναι η γλωσσική σκοπιά της μεταμοντελοποίησης, που έρχεται σε σύγκρουση με την οντολογική σκοπιά της μεταμοντελοποίησης.

Η δύναμη του MOF είναι ότι καθιστά εφικτή τη χρήση ανόμοιων μοντέλων (που αναπαριστούν διαφορετικά πεδία) με έναν διαλειτουργικό και ομοιογενή τρόπο. Οι αντιλαμβανόμενες-το-MOF εφαρμογές ενδέχεται να μην έχουν καμία γνώση των ειδικών-πεδίου (domain-specific) διεπαφών κάποιου στιγμιότυπου ενός μοντέλου, αλλά και πάλι μπορούν να διαβάσουν και να ενημερώσουν (update) αυτό το μοντέλο χρησιμοποιώντας τις γενικές λειτουργίες των ανακλαστικών διεπαφών.

Η σημασιολογία του MOF γενικά ορίζει υπηρεσίες αποθηκών μεταμοντέλων που υποστηρίζουν την κατασκευή, την ανακάλυψη, τη διάσχιση και την ενημέρωση μοντέλων, όπου τα μοντέλα γίνονται αντιληπτά ως στιγμιότυπα ενός συγκεκριμένου μεταμοντέλου. Συγκεκριμένα, η υποστήριξη του MOF για σημασιολογία της διάρκειας του κύκλου ζωής σημαίνει ότι μια υλοποίηση MOF παρέχει ένα αποτελεσματικό εργαλείο συγγραφής και έκδοσης μεταδεδομένων, όταν συνδυαστεί με υποστήριξη για οπτική μοντελοποίηση (visual modeling). Για παράδειγμα, νέα δημιουργημένα μοντέλα μπορούν να εξακολουθούν να υφίστανται (be persisted) στην αποθήκη του MOF και να συνδυαστούν με υπάρχοντα μεταμοντέλα, σύμφωνα με τη διάρκεια κύκλου ζωής του MOF και τη σημασιολογία σύνθεσης – κληρονομικότητα (inheritance), ομαδοποίηση (clustering), ένθεση (nesting), κ.τ.λ.. Μπορούν να δημιουργηθούν και να γίνουν διαθέσιμες στο περιβάλλον διεπαφές μοντέλων και προεπιλεγμένες υλοποιήσεις. Οι προεπιλεγμένες υλοποιήσεις ενισχύονται επιπλέον με τη συμπερίληψη προστιθέμενης προγραμματικής λογικής, είτε γραμμένη «στο χέρι» είτε δημιουργημένη από εργαλεία (για παράδειγμα υλοποίηση των περιορισμών σε OCL) [12]. Μία πλήρως συμμορφούμενη-σε-MOF αποθήκη παρέχει ένα σημαντικό πλήθος υπηρεσιών μεταδεδομένων που υπερβαίνουν κατά πολύ την κατασκευή και την προμήθευση μεταδεδομένων, όπως για παράδειγμα διατήρηση (persistence), δημιουργία εκδόσεων (versioning), υπηρεσίες καταλόγου (directory services) κ.α..

Ως ένα βιομηχανικό standard, το MOF χρησιμοποιείται ως η βάση για διάφορες άλλες τεχνολογίες και πλαίσια (frameworks), όπως το Eclipse Modeling Framework (EMF) [14] για το οποίο γίνεται λόγος στο επόμενο κεφάλαιο.

2.1.2 EMF - Eclipse Modelling Framework

Το Eclipse Modelling Framework1 (EMF) [14], που αρχικά αναπτύχθηκε από την IBM και αυτή τη στιγμή υποστηρίζεται ως μέρος του Eclipse Project [15], είναι ένα ανοιχτού κώδικα

πλαίσιο (framework) Java για μοντελοποίηση, ολοκλήρωση δεδομένων (data integration) και παραγωγή κώδικα (codegeneration).

Το μοντέλο είναι ένα από τα βασικά στοιχεία στα οποία βασίζεται το EMF. Ένα μοντέλο ορίζεται ως μία αφαιρετική αναπαράσταση των δεδομένων που χρησιμοποιούνται από μία εφαρμογή. Στο EMF ένα μοντέλο μπορεί να εκφραστεί σε μία από τις ακόλουθες τρεις τεχνολογίες: annotated Java interfaces [16], XML [17] και UML [5]. Επιτρέποντας την μετατροπή από μία από αυτές τις μορφές σε κάποια άλλη εξ αυτών, το EMF πράγματι ενοποιεί αυτές τις τεχνολογίες.

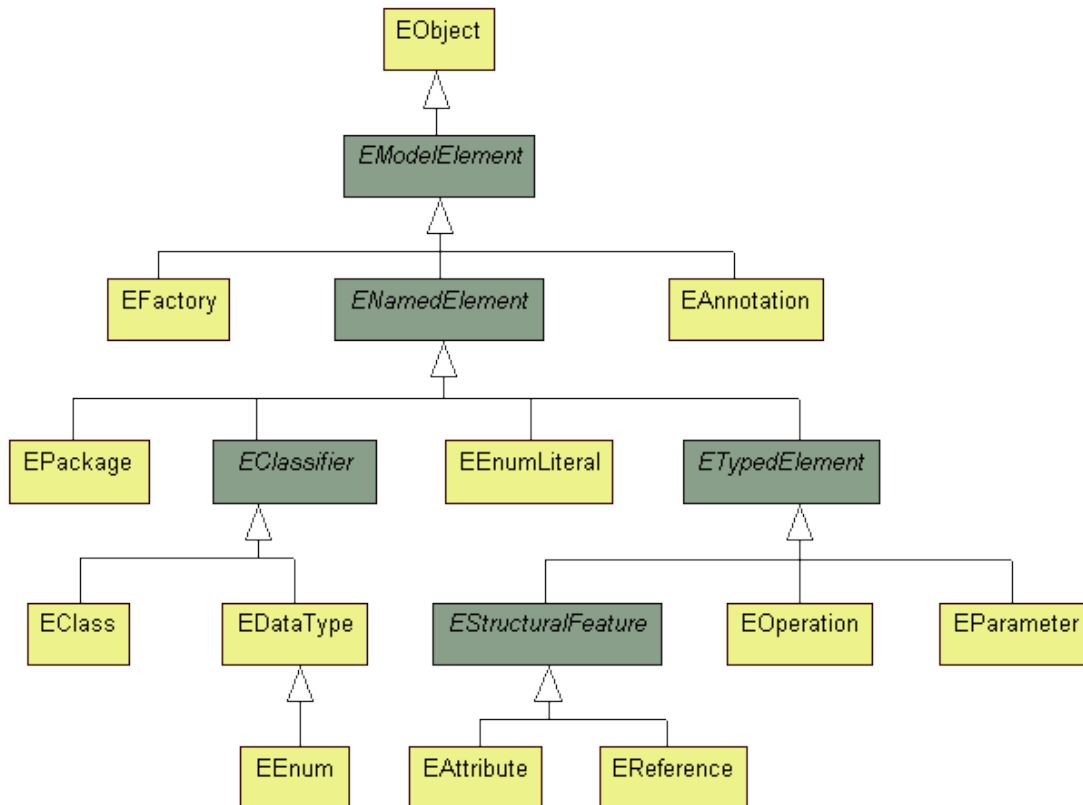
Η βασική ιδέα αυτού του πλαισίου (framework) είναι η παροχή ενός κοινού εδάφους για δύο διαφορετικές ιδέες : του ακραίου προγραμματισμού (programming extreme) και της ακραίας μοντελοποίησης (modeling extreme).

Ξεκινώντας από ένα μοντέλο είτε εκφρασμένο σε XSD [18] είτε σε UML [5], οι προγραμματιστές μπορούν να χρησιμοποιήσουν το EMF για να παράγουν Java κλάσεις και έπειτα να συνεχίσουν τη δουλειά τους με αυτές τις κλάσεις.

Όλα τα μοντέλα στο EMF αναπαρίστανται σε Ecore - ένα μεταμοντέλο για τα μοντέλα στο EMF. Υπάρχει μία ένα-προς-ένα αντιστοιχία μεταξύ του Ecore και του Essential MOF (EMOF), που είναι ένα υποσύνολο του MOF 2.0.

Η βασική διαφοροποίηση του EMF από το πρότυπο MOF προέρχεται από το σκοπό για τον οποίο δημιουργήθηκαν. Το EMF είναι ανοιχτού κώδικα πρόγραμμα (project) για δημόσια χρήση, ικανό για ταχεία εξέλιξη, όμως η προδιαγραφή (specification) ίσως έρχεται σε δεύτερη μοίρα σε σχέση με την υλοποίηση. Το EMF επιδιώκει τα πλεονεκτήματα του τυπικού προγραμματισμού (formal modeling) και της παραγωγής κώδικα Java, οπότε δεν είναι ούτε ανεξαρτήτου-γλώσσας (language-independent), ούτε κατανεμημένο (distributed). Έτσι, το EMF έχει μία προσέγγιση μάλλον από-κάτω-προς-τα-πάνω (bottom-up) σε αντίθεση με την από-πάνω-προς-τα-κάτω προσέγγιση του προτύπου MOF.

Όλη η ιεραρχία του μοντέλου Ecore φαίνεται στο Σχήμα 2.4.

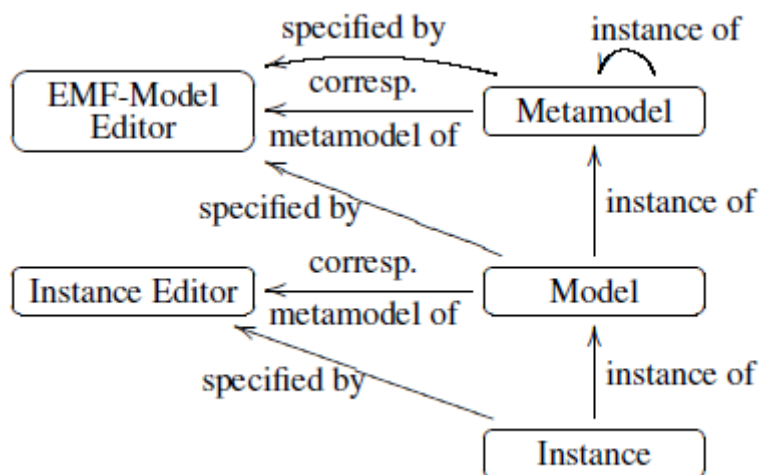


Σχήμα 2.4: Η ιεραρχία κλάσεων του μοντέλου Ecore (τα σκιασμένα κουτιά είναι αφηρημένες κλάσεις) [14]

Το μοντέλο Ecore χρησιμοποιεί EClasses για να μοντελοποιήσει κλάσεις. Μια μοντελοποιημένη κλάση μπορεί να αναφέρεται σε διάφορες άλλες κλάσεις ως supertypes αυτής. Μπορεί επίσης να έχει διάφορα datamembers, τα οποία κατατάσσονται σε δύο κατηγορίες: τα χαρακτηριστικά (attributes) και τις αναφορές (references). Τα EAttributes μοντελοποιούν τα χαρακτηριστικά και τα EReferences τις αναφορές. Τα χαρακτηριστικά και οι αναφορές έχουν δύο συστατικά: το όνομα (name) και τον τύπο (type) τους. Ο τύπος ενός χαρακτηριστικού μοντελοποιείται με ένα EDataType που αναφέρεται είτε σε έναν πρωτογενή τύπο (primitive type), όπως για παράδειγμα int, double, boolean κ.τ.λ., είτε σε ένα Αντικείμενο (Object). Ο τύπος μιας αναφοράς μοντελοποιείται με ένα EDataType το οποίο αναφέρεται σε ένα EClass. Οι συμπεριφορές μιας μοντελοποιημένης κλάσης αντικατοπτρίζονται στα EOperations. Μία λειτουργία (operation) μπορεί να έχει μηδέν ή περισσότερες παραμέτρους και έναν επιστρεφόμενο τύπο (return type), τα οποία όλα μοντελοποιούνται με EParameters.

Σύμφωνα με το EMF, η ιεραρχία μεταμοντελοποίησης οργανώνεται σε μια ιεραρχία τριών στιβάδων. Αυτές οι στιβάδες, ξεκινώντας από πάνω προς τα κάτω, καλούνται μεταμοντέλο (metamodel), μοντέλο (model) και στιγμιότυπο (instance), όπως φαίνεται και στο Σχήμα 2.5. Παρατηρούμε ότι το Ecore είναι αντανακλαστικό (reflexive), εννοώντας ότι προδιαγράφει

την ίδια του τη δομή. Παρατηρούμε επίσης ότι το EMF έχει μόνο τρεις στιβάδες, σε αντίθεση με την ιεραρχία του OMG, καθώς δεν προορίζεται για μεταμοντέλο μιας συλλογής άλλων γλωσσών, όπως συμβαίνει με το MOF. Μια ακόμη διαφορά είναι ότι το επίπεδο στιγμιότυπων διαχωρίζεται ξεκάθαρα από το επίπεδο μοντέλων. Στο EMF, που είναι εμπνευσμένο από την αντικειμενοστρεφή μοντελοποίηση, και κυρίως από τη Java, οι σχέσεις μεταξύ των μοντέλων στην ιεραρχία ονομάζονται στιγμιότυπο-του (instance-of).



Σχήμα 2.5: Η ιεραρχία μεταμοντελοποίησης του EMF [2]

Δοθέντος ενός μοντέλου ως εισόδου, είτε σε XSD, είτε σε UML ή και σε annotated Java interfaces, το EMF θα παράγει ως έξοδο ένα στιγμιότυπο Ecore για αυτό το μοντέλο. Ο EMF generator παράγει μία αντίστοιχη διεπαφή (interface) και μία κλάση (class) για κάθε μοντελοποιημένη κλάση. Ο διαχωρισμός της κλάσης από τη διεπαφή πραγματοποιείται για τη δυνατότητα υποστήριξης πολλαπλής κληρονομικότητας στη Java. Κάθε παραγόμενη διεπαφή επεκτείνει (extends) τη διεπαφή EObject, η οποία, σε αντιστοιχία με το Java Object, παρέχει κοινή συμπεριφορά για όλες τις μοντελοποιημένες κλάσεις. Εκτός από τις διεπαφές και τις κλάσεις για το μοντέλο, ο EMF generator παράγει δύο επιπλέον πακέτα: το ένα έχει την κατάληξη .Edit και το άλλο .Editor και περιέχουν γενικές (generic) και επαναχρησιμοποιήσιμες κλάσεις για την κατασκευή συντακτών (editors) για τα μοντέλα EMF. Οι συντάκτες αυτοί παρέχουν ένα βολικό εργαλείο στο χρήστη για την κατασκευή στιγμιότυπων του μοντέλου [19].

Το EMF είναι ένα ευρέως χρησιμοποιούμενο πλαίσιο μοντελοποίησης γιατί θεωρείται μία χαμηλού κόστους εφαρμογή της Μοντελοδηγούμενης Μηχανίκευσης (MDE). Επιπλέον, έχει ξεκάθαρη σημασιολογία λόγω της απλότητας του μεταμοντέλου του και επίσης υπάρχουν πολλά εργαλεία που χτίζονται γύρω από αυτό και γίνονται διαθέσιμα ως Eclipse plug-ins.

Το EMF έπαιξε σημαντικό ρόλο στην υλοποίηση του συστήματος αναφοράς της παρούσας εργασίας, καθώς αποτέλεσε το εργαλείο για τη μοντελοποίηση των διαφόρων χαρακτηριστικών των επιχειρηματικών διαδικασιών (business processes) και τη σύνδεσή τους με αυτές.

2.2 Δείκτες-Κλειδιά για την Επίδοση (Key Performance Indicators)

Οι Δείκτες-Κλειδιά για την Επίδοση (Key Performance Indicators), γνωστοί και ως KPIs ή και Δείκτες-Κλειδιά για την Επιτυχία (KSIs- Key Success Indicators) είναι προκαθορισμένες ποσοτικές μετρήσεις που αντανακλούν τους κρίσιμους παράγοντες επιτυχίας ενός οργανισμού. Είναι ο βιομηχανικός όρος που αντιστοιχεί στη Μέτρηση της Επίδοσης (Measure of Performance) και συνήθως επιλέγονται και υλοποιούνται με την εφαρμογή διαχειριστικών πλαισίων (management frameworks), όπως τα Balanced Scorecards των Kaplan και Norton. Το Balanced Scorecard βλέπει την εταιρική απόδοση μέσα από τέσσερις συνιστώσες : τα χρηματοοικονομικά μεγέθη, τον πελάτη, τις εσωτερικές επιχειρηματικές διαδικασίες και τη μάθηση και ανάπτυξη. Με λίγα λόγια αποκαλύπτει τις επιχειρησιακές και οργανωτικές λειτουργίες, δραστηριότητες και στόχους που κρύβονται πίσω από τη χρηματοοικονομική απόδοση [20].

Οι KPIs βοηθούν έναν οργανισμό να ορίσει και να μετρήσει την πρόοδό του σε σχέση με τους επιχειρησιακούς του στόχους. Συνήθως αποτελούν και το κίνητρο για βελτιώσεις και αποτελούν ενδείξεις προς τη σωστή κατεύθυνση που πρέπει να ακολουθήσει ο οργανισμός.

Οι KPIs σχετίζονται πολλές φορές με μία εξαιρετικά δημοφιλή και αποτελεσματική πρακτική που ασκούν οι επιχειρήσεις : το benchmarking (στα ελληνικά θα μπορούσε να αποδοθεί με τον όρο συγκριτική προτυποποίηση). Το benchmarking είναι η συνεχής και συστηματική σύγκριση μιας ή περισσότερων επιχειρησιακών στρατηγικών, λειτουργιών, διαδικασιών ή πρακτικών με αυτές των καλύτερων ανταγωνιστών ή αυτών των επιχειρήσεων που θεωρούνται «άριστες» στο συγκεκριμένο αντικείμενο [20]. Έχει ως στόχο την ποσοτικοποίηση των υπάρχουσών διαφορών απόδοσης, την τεκμηρίωση του λόγου ύπαρξής τους και την αναγνώριση των «βημάτων» προς τη σωστή κατεύθυνση, που συνήθως επιτυγχάνεται με την πολύτιμη βοήθεια των KPIs.

Ανάλογα με τον οργανισμό και το σκοπό τους, οι δείκτες αυτοί διαφέρουν και έτσι παραδείγματα τέτοιων δεικτών μπορεί να αποτελέσουν :

- για μια επιχείρηση το ποσοστό εσόδων προς τα έξοδα σε διάστημα ενός έτους

- για μία υπηρεσία το ποσοστό αιτημάτων που εξυπηρετούνται προς τα συνολικά αιτήματα που καταφτάνουν μέσα σε συγκεκριμένο χρονικό διάστημα
- για ένα σχολείο ο μέσος όρος της βαθμολογίας των μαθητών ενός έτους κ.τ.λ..

Όποιοι KPIs και αν επιλεγθούν, θα πρέπει να αντανakλούν τους στόχους του οργανισμού, να αποτελούν κλειδί για την επιτυχία του και να είναι μετρήσιμοι (ποσοτικοί). Λέγεται συχνά ότι το κλειδί για το σωστό ορισμό των KPIs και των στόχων τους είναι αυτοί οι δείκτες να είναι «SMART», που αποτελεί ακρωνύμιο των χαρακτηριστικών τα οποία οφείλουν να τους διακρίνουν, δηλαδή να είναι Συγκεκριμένοι (Specific), Μετρήσιμοι (Measurable), Εφικτοί (Attainable), Συναφείς (Relevant), και Χρονικά Φραγμένοι (Time-Bound). Προσθέτοντας τα χαρακτηριστικά Εξηγήσιμοι (Explainable), για την ευκολία κατανόησής τους και επικοινωνίας όλων των εμπλεκόμενων φορέων του οργανισμού και Σχετικοί (Relative), ώστε να μην επηρεάζονται από ενδεχόμενες μεταβολές της τάξης μεγέθους των χαρακτηριστικών που εξετάζονται, προκύπτει το ακρωνύμιο SMARTER [21].

Το βασικό χαρακτηριστικό για να έχουν κάποια αξία οι KPIs είναι να είναι ποσοτικοποιήσιμοι, δηλαδή να υπάρχει ένας τρόπος ακριβούς ορισμού και μέτρησής τους. Πρέπει να αποφεύγονται δείκτες που είναι αδύνατον να μετρηθούν, όπως επίσης και δείκτες με προβλήματα αμφισημίας.

Αν και οι KPIs χρησιμοποιούνται για την περιοδική αξιολόγηση της επίδοσης οργανισμών σε έναν μακροπρόθεσμο ορίζοντα, εντούτοις ο ορισμός τους και ο τρόπος με τον οποίο μετρώνται ενδέχεται να μεταβάλλεται καθώς μεταβάλλονται οι επιχειρησιακοί στόχοι ή καθώς ο οργανισμός πλησιάζει στην εκπλήρωση των στόχων του. Είναι όμως σημαντικό ο ορισμός των KPIs να είναι προσεχτικός ώστε να μη χρειαστεί να υποστεί δραστικές μεταβολές στο μέλλον. Οι KPIs δείχνουν την πορεία ενός οργανισμού σε σχέση με τους στόχους που έχει θέσει, κάτι που απαιτεί τον όσο το δυνατόν σταθερό και ακέραιο ορισμό τους στο πέρασμα του χρόνου. Για παράδειγμα, πρέπει να πραγματοποιηθεί εξαρχής η προσεχτική επιλογή κάποιου δείκτη που μετράει τις Πωλήσεις (Sales) και να μην αλλάξει ο ορισμός του στην πορεία, καθώς είναι δύσκολο να συγκρίνει κανείς π.χ. μονάδες που πουλήθηκαν, με τη συνολική αξία του κερδους σε κάποια νομισματική μονάδα.

Είναι επίσης σημαντικό να τίθενται συγκεκριμένοι στόχοι που να συνδέονται με κάθε KPI. Οι Δείκτες αυτοί πρέπει να εξυπηρετούν όχι μόνο στην προβολή της συνολικής εικόνας του οργανισμού σε κάποια συγκεκριμένη χρονική στιγμή, αλλά και στην αξιολόγηση της πορείας του όσον αφορά συγκεκριμένους, σαφείς και προκαθορισμένους στόχους, με μετρήσιμο και αναμφισβήτητο τρόπο. Για να γίνει αυτό οι KPIs δεν πρέπει να ποσοτικοποιούν όλους τους παράγοντες που μπορούν να μετρηθούν, αλλά μόνον εκείνους που ενδιαφέρουν και θα μπορούσαν να συντελέσουν στην επίτευξη συγκεκριμένων στόχων. Καλό είναι δε το πλήθος

των δεικτών να παραμένει περιορισμένο για να αποφεύγεται η σύγχυση και να επιτυγχάνεται ο συντονισμός όλων των συντελεστών του οργανισμού σε κοινές επιδιώξεις.

Επιπροσθέτως, για την αποτελεσματικότητα ενός οργανισμού, βοηθά αν τα KPIs βρίσκονται σε μία ενιαία ιεραρχία, η οποία να αντανακλά την εσωτερική οργάνωση του οργανισμού. Είναι σκόπιμο να επιτυγχάνεται ο συντονισμός των διαφόρων τμημάτων και υποτμημάτων ενός οργανισμού σε μία κοινή γραμμή που καθορίζεται από έναν γενικό στόχο υπό τη μορφή τιμών κάποιου γενικού δείκτη, που αναλύεται σε επιμέρους στόχους υπό τη μορφή τιμών αντίστοιχων δεικτών για τα επιμέρους τμήματα του οργανισμού.

Από τη στιγμή που έχουν ορθώς οριστεί, με βάση όσα προαναφέρθηκαν, συγκεκριμένοι KPIs, που αντανακλούν τους στόχους ενός οργανισμού, είναι ευθύνη της διεύθυνσής του να προωθήσει τους στόχους σε όλη τη διαστρωμάτωση αυτού του οργανισμού και να φροντίσει ώστε να γίνει κοινός στόχος όλων των φορέων η ικανοποίηση και αν είναι δυνατόν η υπέρβασή τους.

Στην παρούσα εργασία έγινε μία προσπάθεια προσδιορισμού κάποιων γενικών και αντιπροσωπευτικών KPIs, δηλαδή χαρακτηριστικών των επιχειρηματικών διαδικασιών, με έμφαση στον τομέα της Τεχνολογίας Λογισμικού (Software Engineering), όπως φαίνεται και στην ενότητα 4. Οι δείκτες αυτοί είναι γενικοί και η παρακολούθησή τους δίνει μία εποπτική εικόνα της επίδοσης και απόδοσης μιας ευρείας γκάμας επιχειρηματικών διαδικασιών. Για μια περισσότερο λεπτομερή ανάλυση κατά περίπτωση (case specific) κρίνεται σκόπιμος ο ορισμός περαιτέρω δεικτών, πιο ειδικών και σχετικών με το πεδίο (domain) της εκάστοτε επιχειρηματικής διαδικασίας.

2.3 Αποτίμηση Επιχειρηματικών Διαδικασιών (Business Process Evaluation)

Η αναζήτηση για την τυποποίηση και την αποδοτικότητα στις επιχειρηματικές διαδικασίες έχει μακρά ιστορία, που χαρακτηρίζεται από εναλλαγές άκρατου ενθουσιασμού για την άφιξη νέων, καινοτόμων ιδεών, ακολουθούμενου από βαθιά απογοήτευση όταν η μοντέρνα ιδέα εγκαταλείπεται, για να δώσει τη θέση της στη νέα γενιά μεθοδολογιών.

Από την εποχή που ο Adam Smith πρότεινε τον καταμερισμό της εργασίας στον τομέα της μεταποίησης το 1776 και έναν αιώνα αργότερα ο Frederick Winslow Taylor εισήγαγε την έννοια της επιστημονικής θεώρησης στη διαχείριση επιχειρήσεων [42], πολλά έχουν αλλάξει στον τομέα της διαχείρισης επιχειρήσεων.

Από την αρχή του 20ού αιώνα προτάθηκαν αρκετές τεχνικές για την ανάλυση και τη μοντελοποίηση επιχειρηματικών διαδικασιών (business processes), όπως τα διαγράμματα

ροής ελέγχου (control flow diagrams), τα διαγράμματα PERT, τα γραφήματα Gantt κ.α.. Η ολοένα και αυξανόμενη αντικατάσταση των χειρόγραφα συμπληρωμένων εντύπων με τα ηλεκτρονικά τους υποκατάστατα σταδιακά εξελίχθηκε σε αυτό που σήμερα είναι γνωστό ως Διαχείριση Επιχειρηματικών Διαδικασιών (Business Process Management). Η Διαχείριση Επιχειρηματικών Διαδικασιών ορίζεται ως «η υποστήριξη επιχειρηματικών διαδικασιών με τη χρήση μεθόδων, τεχνικών και λογισμικού για το σχεδιασμό, την εκτέλεση, τον έλεγχο και την ανάλυση λειτουργικών διαδικασιών που περιλαμβάνουν ανθρώπους, οργανισμούς, εφαρμογές, αρχεία και άλλες πηγές πληροφορίας» [22]. Εργαλεία λογισμικού που υποστηρίζουν τη διαχείριση τέτοιων λειτουργικών διαδικασιών έγιναν γνωστά ως συστήματα διαχείρισης επιχειρηματικών διαδικασιών (BPMS - Business Process Management Systems).

Αν και η Διαχείριση Επιχειρηματικών Διαδικασιών έχει τις ρίζες της στα διαγράμματα ροής, η βασική τους διαφορά είναι ότι η Διαχείριση Επιχειρηματικών Διαδικασιών εστιάζει τόσο σε ανθρώπινες διαδικασίες όσο και σε διαδικασίες συστημάτων σε αντίθεση με τα συστήματα διαγραμμάτων ροής που εστιάζουν σε εμπειριστατωμένες διαδικασίες που εκτελούνται από ανθρώπους.

Θεωρίες όπως η Total Quality Management (TQM), που υποστηρίζει τη συνεχή βελτίωση των διαδικασιών ως προς το κόστος, την ποιότητα, την εξυπηρέτηση και την ταχύτητα, χρησιμοποιήθηκαν για τη συνεχή βελτίωση των παρατηρηθέντων διαδικασιών.

Σήμερα, με την εντατικοποιημένη παγκοσμιοποίηση, η αποτελεσματική διαχείριση των επιχειρηματικών διαδικασιών έχει γίνει ακόμα σημαντικότερη [23]. Η συνεχής επέκταση των υποδομών μηχανοργάνωσης (Computing infrastructures) προς κάθε κατεύθυνση σε απάντηση των επιχειρηματικών αναγκών και της τεχνολογικής ανάπτυξης, έχουν οδηγήσει στην εστίαση - πέρα από την υιοθέτηση διαδικασιο-στρεφών αρχών που διέπουν την Διαχείριση Επιχειρηματικών Διαδικασιών - σε τεχνολογίες που περιλαμβάνουν δραστηριότητες αναπαράστασης διαδικασιών μιας επιχείρησης, ώστε οι τρέχουσες διαδικασίες να αναλύονται και να βελτιώνονται σε μια διαδικασία που αποκαλείται Business Process Reengineering (BPR) . Η μοντελοποίηση αυτή των επιχειρηματικών διαδικασιών καλείται Μοντελοποίηση Επιχειρηματικών Διαδικασιών (Business Process Modeling) και λαμβάνει χώρα στη μηχανίκευση συστημάτων (systems engineering) και στην τεχνολογία λογισμικού (software engineering).

Η Μοντελοποίηση Επιχειρηματικών Διαδικασιών πραγματοποιείται από αναλυτές επιχειρήσεων και διευθυντές που επιδιώκουν την βελτίωση της αποτελεσματικότητας και της ποιότητας των διαδικασιών και συνήθως απαιτεί τη χρήση Τεχνολογίας των Πληροφοριών (Informations Technology). Στην καρδιά της Μοντελοποίησης Επιχειρηματικών Διαδικασιών βρίσκεται το επιχειρηματικό μοντέλο (business model), που αναπαριστά μία ή πολλές επιχειρηματική/ές διαδικασία/ες (business process/es) και καθορίζει τον τρόπο με τον οποίο

οι διαδικασίες διεξάγονται για να επιτύχουν τους στόχους ενός οργανισμού. Μία επιχειρηματική διαδικασία (business process) είναι μια συλλογή συσχετισμένων, δομημένων δραστηριοτήτων ή αλλιώς έργων (tasks), που παράγουν ένα συγκεκριμένο προϊόν ή μια συγκεκριμένη υπηρεσία για έναν/πολλούς συγκεκριμένο/ους πελάτη/ες. Οι διαδικασίες αυτές είναι είτε διοικητικές, είτε λειτουργικές ή και βοηθητικές. Μια διαδικασία μπορεί να υποδιαιρείται σε μία ή περισσότερες υποδιαδικασία/ες, οι οποίες όλες συντονίζονται για την επίτευξη του σκοπού της.

Υπάρχουν δύο λειτουργίες που μπορούν να πραγματοποιηθούν με τη βοήθεια της Μοντελοποίησης Επιχειρηματικών Διαδικασιών. Η πρώτη είναι η μοντελοποίηση για τον τυπικό ορισμό, την απεικόνιση και καλύτερη κατανόηση των επιχειρηματικών διαδικασιών, για παράδειγμα μέσω της σκιαγράφησης (sketching) του μοντέλου τους. Η δεύτερη είναι η μοντελοποίηση διαδικασιών με σκοπό την εκτέλεσή τους, με τη χρήση κατάλληλης πλατφόρμας που καλείται Μηχανή Μοντελοποίησης Επιχειρηματικών Διαδικασιών (Business Process Modeling Engine). Η πρώτη πραγματοποιείται με τη χρήση διαγραμμάτων, όπως της UML [5] και της Business Process Modelling Notation (BPMN) [24], που θα αναλυθεί στη συνέχεια, ενώ η δεύτερη με τη χρήση γλωσσών προγραμματισμού, όπως XPDL και BPEL [25] [26], που επίσης θα αναλυθεί παρακάτω. Πάντως το όραμα της Μοντελοποίησης Επιχειρηματικών Διαδικασιών είναι τα μοντέλα να γίνουν πλήρως εκτελέσιμα και κατάλληλα για διαδραστικές προσομοιώσεις, κάτι που καθίσταται ολοένα και περισσότερο εφικτό χάρη στις ραγδαίες τεχνολογικές εξελίξεις.

2.3.1 Business Process Modeling Notation (BPMN)

Η Business Process Modeling Notation (BPMN) [24] [27] [28] είναι ένα πρότυπο που ανέπτυξε και ανακοίνωσε το 2004 ο Business Process Management Initiative (BPMI) [29] και σήμερα διατηρείται από το Object Management Group (OMG) [9] από τότε που οι δύο οργανισμοί συγχωνεύτηκαν το 2005.

Σκοπός της προσπάθειας αυτής ήταν η προτυποποίηση μιας σημειογραφίας (notation) κατανοητής από όλους τους ανθρώπους των επιχειρήσεων (business people) : τους αναλυτές επιχειρήσεων (business analysts), τους τεχνικούς προγραμματιστές (technical developers) και τους διευθυντές επιχειρήσεων (business managers). Μακροπρόθεσμη επιδίωξη της προτυποποίησης αυτής ήταν (και είναι) η συμβολή της στο όραμα της ενοποίησης των διαδικασιών της (γραφικής) μοντελοποίησης, της εκτέλεσης και του ελέγχου των επιχειρηματικών διαδικασιών (business processes) .

Η BPMN ορίζει ένα διάγραμμα, το Business Process Diagram (BPD), το οποίο βασίζεται στην τεχνική διαγραμμάτων ροής για την παραγωγή του γραφικού μοντέλου επιχειρηματικών

διαδικασιών, δηλαδή ένα δίκτυο γραφικών αντικειμένων που αναπαριστούν δραστηριότητες και ελέγχους ροής για τη σειρά εκτέλεσής τους [30].

Το Business Process Diagram (BPD) περιέχει στοιχεία που προέρχονται από ευρέως χρησιμοποιούμενα προγενέστερα εργαλεία και τεχνολογίες, όπως τα Activity-Decision Flow (ADF) Diagrams, τα Event-Process Chains (EPCs), τα UML Activity Diagrams κ.α. Με την εμπειρία και τη γνώση του παρελθόντος, τα στοιχεία αυτά για το γραφικό σχεδιασμό επιχειρηματικών μοντέλων είναι οικεία στους επιχειρηματικούς αναλυτές (business analysts). Επιπροσθέτως, κρατώντας το πλήθος των βασικών στοιχείων περιορισμένο και τα στοιχεία διακριτά μεταξύ τους, συμβιβάζοντας πάντα την απλότητα με την όσο το δυνατόν μεγαλύτερη εκφραστικότητα, η BPMN αποτελεί ένα κατανοητό και ευανάγνωστο πρότυπο με δυνατότητες επαρκούς απεικόνισης των επιχειρηματικών διαδικασιών από τους σχεδιαστές (modellers). Επιπλέον, παρέχει, πέραν των βασικών στοιχείων, τη δυνατότητα χρησιμοποίησης επιπρόσθετων παραλλαγών και πληροφοριών, για να υποστηρίζεται η μεγαλύτερη πολυπλοκότητα των διαγραμμάτων, χωρίς δραστικές αλλαγές στη βασική τους εικόνα (look-and-feel).

Τα βασικά στοιχεία της BPMN ομαδοποιούνται σε τέσσερις κατηγορίες: Αντικείμενα Ροής (Flow Objects), Συνδετικά Αντικείμενα (Connecting Objects), Διάδρομοι Κολύμβησης (Swimlanes) και Εκθέματα (Artifacts). Ακολουθεί μια σύντομη περιγραφή [27] [30] τους και στη συνέχεια η γραφική τους αναπαράσταση στο Σχήμα 2.6 :

Αντικείμενα Ροής (Flow Objects)

- Event : κάτι που συμβαίνει κατά τη διάρκεια μιας δραστηριότητας. Υπάρχουν τρία είδη γεγονότων (Events) ανάλογα με το τι επηρεάζουν : Start, Intermediate, και End, η αντίστοιχη γραφική αναπαράσταση των οποίων παρουσιάζεται στο Σχήμα 2.6
- Activity : μια δραστηριότητα – ένας γενικός όρος που περιγράφει το έργο που παράγει μια εταιρία - και χωρίζεται στις κατηγορίες *Task* και *Sub-Process*
- Gateway : κομβικό σημείο απόκλισης ή σύγκλισης της ακολουθιακής ροής, που αναπαριστά το κλασικό σημείο απόφασης και την διάσπαση (forking), συνένωση (joining) και συγχώνευση (merging) μονοπατιών εκτέλεσης.

Συνδετικά Αντικείμενα (Connecting Objects)

- Sequence Flow : δείχνει τη σειρά (sequence) με την οποία εκτελούνται οι διάφορες δραστηριότητες σε μια διαδικασία

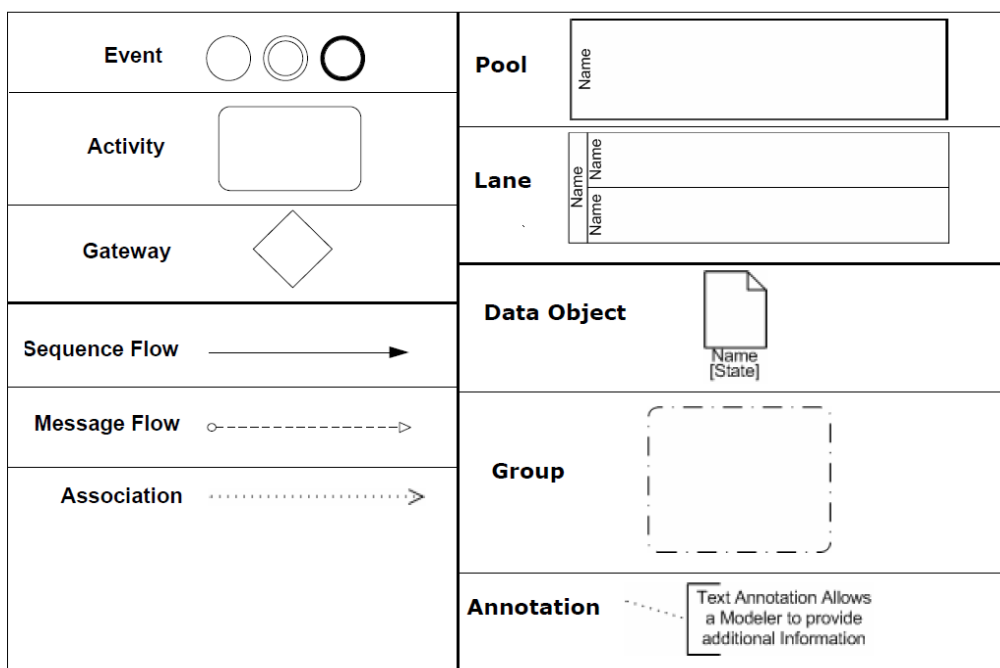
- Message Flow : δείχνει τη ροή μηνυμάτων μεταξύ των διαφορετικών συμμετεχόντων στη διαδικασία, οι οποίοι συνήθως αναπαρίστανται με ξεχωριστά Pools (βλ. παρακάτω)
- Association : συσχετίζει δεδομένα, κείμενο και άλλα εκθέματα (artifacts) με αντικείμενα ροής. Χρησιμοποιείται για να αναπαραστήσει τις εισόδους (inputs) και τις εξόδους (outputs) των δραστηριοτήτων.

Διάδρομοι Κολύμβησης (Swimlanes)

- Pool : διαχωρίζει ένα σύνολο δραστηριοτήτων από τις υπόλοιπες. Συνήθως χρησιμοποιείται για την αναπαράσταση ενός συμμετέχοντα σε μια διαδικασία σε περιπτώσεις B2B
- Lane : μία περαιτέρω υποδιαίρεση του Pool για την οργάνωση και την κατηγοριοποίηση δραστηριοτήτων.

Εκθέματα (Artifacts)

- Data Object : ένας μηχανισμός για την αναπαράσταση απαιτούμενων ή παραγόμενων δεδομένων από τις δραστηριότητες. Συνδέονται με τις δραστηριότητες μέσω συσχετίσεων (Associations)
- Group : χρησιμοποιείται για σκοπούς αρχειοθέτησης ή ανάλυσης αλλά δεν επηρεάζει τη ροή εκτέλεσης
- Annotation : ένας μηχανισμός για την παροχή από το σχεδιαστή απιρόσθετων πληροφοριών υπό μορφή κειμένου για τον αναγνώστη ενός διαγράμματος BPMN.



Σχήμα 2.6: Η γραφική αναπαράσταση των βασικών στοιχείων της BPMN

Η BPMN είναι σχεδιασμένη για να καλύψει ένα μεγάλο εύρος διαφορετικών τύπων μοντελοποίησης, και σε διαφορετικά επίπεδα εστίασης. Εντούτοις, οι διαδικασίες που μοντελοποιεί μπορούν να χωριστούν σε δύο κατηγορίες: τις συνεργατικές B2B διαδικασίες και τις εσωτερικές (ιδιωτικές) επιχειρηματικές διαδικασίες. Οι πρώτες αναφέρονται στις αλληλεπιδράσεις δύο ή περισσότερων επιχειρηματικών οντοτήτων και τις αναπαριστούν ως δραστηριότητες και μηνύματα που ανταλλάσσουν μεταξύ τους από μία αφαιρετική και χωρίς πολλές λεπτομέρειες σκοπιά. Αντίθετα, οι εσωτερικές διαδικασίες εστιάζουν στις ιδιωτικές δραστηριότητες ενός επιχειρηματικού οργανισμού, αν και ενδέχεται να εμπεριέχουν και κάποιες αλληλεπιδράσεις του με το περιβάλλον του.

Το αφαιρετικό επίπεδο στο οποίο αναφέρεται ένα BPD ποικίλει από μοντέλο σε μοντέλο, ανάλογα με το σκοπό της ανάλυσης και το στάδιο στο οποίο βρίσκεται. Συνήθως στο αρχικό στάδιο απεικονίζονται υψηλού επιπέδου δραστηριότητες που στη συνέχεια αναλύονται περαιτέρω στα συστατικά τους είτε στο ίδιο διάγραμμα είτε σε νέα διαγράμματα.

Αν και σχεδιάστηκε με σκοπό την κάλυψη ενός μεγάλου πεδίου εφαρμογών, η BPMN αναφέρεται αποκλειστικά στον τομέα της μοντελοποίησης επιχειρηματικών διαδικασιών (Business Process Modelling) και έτσι δεν υποστηρίζει και ούτε προτίθεται να υποστηρίξει στο μέλλον τη μοντελοποίηση μη-επιχειρηματικών τομέων, όπως οργανωτικές δομές και μοντέλα δεδομένων.

Η προδιαγραφή της BPMN [27] προτείνει με έναν όχι ιδιαίτερα επίσημο τρόπο κάποια αντιστοίχιση μεταξύ των στοιχείων της BPMN και της BPEL. Εντούτοις, κάποιες ουσιαστικές διαφορές αυτών των προδιαγραφών καθιστούν τη δημιουργία πιστού κώδικα σε BPEL βασισμένου σε κάποιο μοντέλο σε BPMN, αν όχι αδύνατη, τουλάχιστον δυσανάγνωστη. Η αντίστροφη διαδικασία είναι ακόμα δυσκολότερη και απαιτεί συνήθως πολλές περισσότερες πληροφορίες από όσες μπορεί να απεικονίσει ένα μοντέλο BPMN με τα βασικά του στοιχεία.

2.3.2 Business Process Execution Language (BPEL)

Η γλώσσα Business Process Execution Language (BPEL) [25] [26] είναι μία εκτελέσιμη διάλεκτος της XML [17] που καθιστά δυνατή τη μοντελοποίηση αλληλεπιδράσεων μεταξύ Υπηρεσιών του διαδικτύου σε ένα περιβάλλον «cloud computing» [44] [45]. Η μοντελοποίηση αυτού του είδους είναι πολύτιμη για την επιτυχημένη Διαχείριση Επιχειρηματικών Διαδικασιών (Business Process Management) και για την υλοποίηση της υπηρεσιο-στρεφούς αρχιτεκτονικής (SOA – Service Oriented Architecture [45] [46]). Η BPEL τυποποιήθηκε από τον OASIS [31] το 2004, κατόπιν από κοινού προσπαθειών δημιουργίας της γλώσσας από εταιρίες όπως η IBM, η Microsoft κ.α..

Μία από τις βασικές χρήσεις της BPEL είναι η μοντελοποίηση αλληλεπιδράσεων των υπηρεσιών σε ένα καταναμημένο σύστημα, επιτρέποντας τις σύνθετες ενορχηστρώσεις (orchestrations) πολλαπλών εφαρμογών υπηρεσιών μέσω μιας υπηρεσίας ελεγκτή.

Η BPEL, λοιπόν, είναι μία γλώσσα ενορχήστρωσης (orchestration) και όχι χορογραφίας (choreography). Η βασική διαφορά αυτών των δύο τεχνικών έγκειται στην εκτελεσιμότητα και τον έλεγχο. Στην ενορχήστρωση ο έλεγχος είναι κεντρικός και σαφώς ορισμένος και η αλληλεπίδραση των διαφόρων μονάδων του συστήματος πραγματοποιείται με ανταλλαγή μηνυμάτων, η συχνότητα και το είδος των οποίων ελέγχεται από έναν κεντρικό ελεγκτή. Αντίθετα, στην χορογραφία ορίζεται ένα πρωτόκολλο για τις αλληλεπιδράσεις των (ομοτίμων) μονάδων, που επιδέχεται πλήθος διαφορετικών υλοποιήσεων.

Η ιστορία της εξέλιξης της BPEL και η εστίασή της στις σύγχρονες επιχειρηματικές διαδικασίες (business processes) την οδήγησαν στην υιοθέτηση των υπηρεσιών διαδικτύου (web services) ως τον εξωτερικό της μηχανισμό επικοινωνίας. Έτσι ο μηχανισμός ανταλλαγής μηνυμάτων της BPEL στηρίζεται στη χρήση της WSDL [32]. Μία διαδικασία BPEL (BPEL process) είναι υπεύθυνη για την ορθή και έγκαιρη ανταλλαγή μηνυμάτων (με τη χρήση της τεχνολογίας SOAP [33]) μεταξύ των διαφόρων υπηρεσιών που συμμετέχουν σε αυτήν, ενδεχομένως βασισμένη στις διεπαφές WSDL που έχουν οριστεί για αυτές τις υπηρεσίες, ενώ και η ίδια η διαδικασία BPEL μπορεί να εκθέσει μια διεπαφή WSDL για τον εαυτό της. Όταν μία διαδικασία BPEL (BPEL process) περιγράφεται στα πλαίσια της

WSDL, η διαδικασία αυτή μπορεί να θεωρηθεί ως μία υπηρεσία που περιγράφει τον εαυτό της, με την περιγραφή αυτή να βρίσκεται μέσα στην ίδια την υπηρεσία.

Η γλώσσα BPEL είναι μια στατικού τύπου (statically typed) και πλήρης κατά Turing (Turing complete) γλώσσα περιγραφής σεναρίων (scripting language) [34]. Έχει προτυποποιημένη σύνταξη που ορίζεται σε XML [17] και μπορεί να διατηρεί προσωρινές δομές δεδομένων σε μεταβλητές, τις οποίες ζητούν (query) και διαχειρίζονται οι διαδικασίες BPEL (BPEL processes) για το πέρασμα παραμέτρων από τη μία υπηρεσία στην άλλη. Χρησιμοποιεί XML Schemas [18] για να ορίσει τον τύπο των μεταβλητών και των παραμέτρων εισόδου και εξόδου των διαφόρων υπηρεσιών. Η ορατότητα των δηλώσεων καθορίζεται από τις εμβέλειες (scopes) όπως και σε άλλες γλώσσες προγραμματισμού. Οι εμβέλειες μπορούν να περιέχουν ένα πλήθος δηλώσεων μεταβλητών καθώς επίσης και ένα σύνολο βασικών δραστηριοτήτων (activities) και στοιχείων ελέγχου ροής (control flow elements). Οι βασικές δραστηριότητες περιλαμβάνουν την επίκληση (invocation) μιας απομακρυσμένης υπηρεσίας (remote service), ντετερμινιστική και μη παραλαβή μηνυμάτων, ανάθεση δομών δεδομένων XML σε μεταβλητές και ένα πλήθος επιπλέον δραστηριοτήτων, όπως χειρισμό και δημιουργία εξαιρέσεων, λειτουργίες αναμονής (wait operations) κ.α.. Τα στοιχεία ελέγχου ροής, που χρησιμοποιούνται για το συνδυασμό των προαναφερθέντων δραστηριοτήτων, υποστηρίζουν υπό συνθήκη εκτελέσεις και τον ορισμό διαφόρων βρόχων, όπως είναι γνωστά και από άλλες δομημένες γλώσσες προγραμματισμού [34].

Ακολουθεί μία σύντομη περιγραφή [19] [35] των βασικών δομικών στοιχείων της BPEL :

Βασικές Δραστηριότητες (Basic Activities)

- invoke : επίκληση μιας ορισθείσας λειτουργίας (operation) μιας συγκεκριμένης Διαδικτυακής Υπηρεσίας (Web Service)
- receive : παραλαβή μηνύματος από συγκεκριμένη θύρα (portType)
- reply : αποστολή απαντητικού μηνύματος σε συγκεκριμένη θύρα (portType)
- assign : αντιγραφή δεδομένων μεταξύ μεταβλητών και άλλων δομών
- throw : παραγωγή λάθους (fault generation) για την κατάλληλη διαχείρησή του
- wait : αναμονή για καθορισμένο χρονικό διάστημα
- empty : καμία ενέργεια

Επιπλέον, στην BPEL 2.0 [36] ορίζονται οι εξής δραστηριότητες : extensionActivity, exit, rethrow, compensate, compensateScope και validate.

Περιεκτικές Δραστηριότητες (Container Activities)

- sequence : εκτέλεση των περιεχομένων δραστηριοτήτων με τη σειρά που εμφανίζονται
- flow : παράλληλη εκτέλεση όλων των περιεχομένων δραστηριοτήτων μέχρι να τελειώσουν όλες την εκτέλεσή τους
- if : εκτέλεση ακριβώς μίας εκ των περιεχομένων δραστηριοτήτων, ανάλογα με το αποτέλεσμα μιας συγκεκριμένης συνθήκης
- while : εκτέλεση της περιεχόμενης δραστηριότητας επανειλημμένα όσο επαληθεύεται μια συγκεκριμένη συνθήκη
- repeatUntil : εκτέλεση της περιεχόμενης δραστηριότητας επανειλημμένα μέχρι να πάψει να ισχύει συγκεκριμένη συνθήκη
- pick : αναμονή μέχρι την άφιξη ενός μηνύματος, το οποίο καθορίζει ποια από τις περιεχόμενες δραστηριότητες θα εκτελεστεί
- forEach : εκτέλεση της περιεχόμενης εμβέλειας (scope) με προκαθορισμένο αριθμό επαναλήψεων
- scope : καθορισμός εμβέλειας σε μια διαδικασία BPEL (BPEL process)

Λόγω των ευκατανόητων εντολών που χρησιμοποιεί για την εκτέλεση σύνθετων λειτουργιών και της αναγνωσιμότητάς της, η BPEL αποτελεί μία γέφυρα συνεννόησης μεταξύ των ανθρώπων των επιχειρήσεων (business people) και των προγραμματιστών, αν και πολλές φορές έχει επικριθεί για την σύνθετη δομή της.

Ένα από τα κρίσιμα χαρακτηριστικά της BPEL είναι η διαλειτουργικότητα, εφόσον χρησιμοποιεί και συντονίζει πολλαπλές εφαρμογές, γραμμένες ενδεχομένως σε διαφορετικές γλώσσες και τοποθετημένες σε καταναμημένα συστήματα, οπότε είναι σημαντικός ο από-άκρη-σε-άκρη έλεγχος.

Είναι σύνηθες να συσχετίζεται η BPEL με τη Business Process Management Notation (BPMN) [24], όπως αυτή περιγράφηκε στο προηγούμενο κεφάλαιο. Όπως όμως έχει αναφερθεί, σε αντίθεση με τη BPEL, η BPMN δεν είναι εκτελέσιμη και χρησιμοποιείται για το σχεδιασμό και τη γραφική αναπαράσταση των επιχειρηματικών διαδικασιών, κάτι που προς το παρόν την καθιστά ένα αποτελεσματικό μέσο, ώστε σε συνδυασμο με τη μοντελοποίηση σε BPEL, να συντελεί στην καλύτερη κατανόηση των διαδικασιών από τους ανθρώπους των επιχειρήσεων (business people) , που ενδεχομένως να στερούνται γνώσεων προγραμματισμού. Για το σκοπό αυτό έχουν προταθεί διάφορες αντιστοιχίσεις μεταξύ των δύο αυτών γλωσσών [27] [37], αν και επίσημα, από τους κατασκευαστές της BPEL δεν έχει κριθεί σκόπιμη η γραφική αναπαράστασή της με προδιαγεγραμμένο τρόπο.

3

Αρχιτεκτονική

Συστήματος

Στο κεφάλαιο αυτό γίνεται η περιγραφή της αρχιτεκτονικής του συστήματος και παρατίθεται το αντίστοιχο σχηματικό διάγραμμα. Ακολούθως, πραγματοποιείται η περιγραφή των διαφόρων συστατικών μερών του συστήματος (components), των διεπαφών (interfaces) και των διαφόρων υπηρεσιών (services). Επίσης, γίνεται η ανάλυση των απαιτήσεων για τις λειτουργίες του συστήματος υπό μορφή ακολουθιακών διαγραμμάτων UML.

3.1 Αρχιτεκτονική Αναφοράς/ Σχηματικό Διάγραμμα

Η Αρχιτεκτονική Τεχνοτροπία που χρησιμοποιήθηκε για το σύστημα είναι η, Ιεραρχική, Τεχνοτροπία της Στιβάδας (Layered) με αρκετά στοιχεία Αντικειμενοστρέφειας (Object Orientation) και τεχνοτροπίας Ροής Δεδομένων (Data Flow) και συγκεκριμένα Αυλού/Φίλτρου(Pipe and Filter).

Για το σχεδιασμό του συστήματος κρίθηκε σκόπιμη η διάσπαση των διαφόρων διαδικασιών, που αφορούν στην ανάλυση παραμετροποιημένων μοντέλων επιχειρηματικών διαδικασιών, σε επιμέρους επίπεδα (layers), καθένα εκ των οποίων :

- προσφέρει συγκεκριμένες υπηρεσίες στα υποσυστήματα που ανήκουν στο παραπάνω επίπεδο
- συμπεριφέρεται σαν πελάτης στα υποσυστήματα που βρίσκονται στο παρακάτω επίπεδο (βλ. ορισμό από τους Garlan and Shaw)
- προσφέρει υπηρεσίες που αντιστοιχούν σε κάποιο συγκεκριμένο αφαιρετικό μοντέλο.

Τα παραπάνω καθίστανται δυνατά με τη χρήση μιας Ιεραρχικής τεχνοτροπίας, και συγκεκριμένα της Τεχνοτροπίας της Στιβάδας (Layered). Με χρήση αυτής επιτρέπεται η εύκολη συντήρηση του συστήματος (η αλλαγή σε ένα επίπεδο επηρεάζει μόνο τα γειτονικά επίπεδα) και η επαναχρησιμοποίηση κώδικα (διαφορετικές υλοποιήσεις που προσφέρουν τις ίδιες υπηρεσίες σε κάποιο επίπεδο μπορούν να αντικαταστήσουν η μια την άλλη) .

Από την άλλη, η τεχνοτροπία αυτή συνεπάγεται ένα αυξημένο κόστος μηνυμάτων ανάμεσα σε υποσυστήματα σε διαφορετικά επίπεδα, κάτι όμως που δεν αποτελεί τροχοπέδη για το συγκεκριμένο σύστημα.

Έχουμε λοιπόν τα εξής επίπεδα, δίπλα σε καθένα εκ των οποίων αναγράφονται οι υπηρεσίες που προσφέρει :

- **MODEL READING:** «Διάβασμα» των διαφόρων μοντέλων- μοντέλων χρησιμότητας (utility models) και μοντέλων επιχειρηματικών διαδικασιών (business process models) - και έλεγχος της εγκυρότητας των (validation).
- **MODEL VERIFICATION:** Επαλήθευση των έγκυρων μοντέλων - έλεγχος για το συμβιβασμό τους στις διάφορες προδιαγραφές και περιορισμούς στους οποίους υπόκεινται.
- **BUSINESS PROCESS MODEL NORMALIZING & UTILITY MODEL SELECTION:** Ομαλοποίηση των επιχειρηματικών διαδικασιών και εξασφάλιση ότι βρίσκονται σε μορφή κατάλληλη προς επεξεργασία από το σύστημα . Επιλογή του μοντέλου χρησιμότητας, με βάση το οποίο πραγματοποιείται η ανάλυση από το σύστημα.
- **MODEL LOADING & ANNOTATION:** «Φόρτωση» του προς επεξεργασία ενοποιημένου μοντέλου στο σύστημα.
- **SIMULATION:** Προσομοίωση των παραμετροποιημένων επιχειρηματικών διαδικασιών και δημιουργία όλων των εναλλακτικών πορειών που αυτές δύνανται να ακολουθήσουν.
- **EVALUATION:** Πραγματοποίηση όλων των υπολογισμών των διαφόρων παραμέτρων του μοντέλου και δημιουργία προσβάσιμων αποτελεσμάτων.
- **ASSESSMENT:** Επεξεργασία αποτελεσμάτων όπως αυτά προκύπτουν από το επίπεδο εκτίμησης (evaluation layer) για την αξιολόγηση συνολικών μεγεθών και εξαγωγή συμπερασμάτων για την όλη επιχειρηματική διαδικασία.
- **RESULT PRESENTATION & STORAGE:** Παρουσίαση, με τρόπο που επιλέγεται από το χρήστη, των αποτελεσμάτων, όπως αυτά προκύπτουν από το επίπεδο αξιολόγησης (assessment layer), και αποθήκευση αυτών.

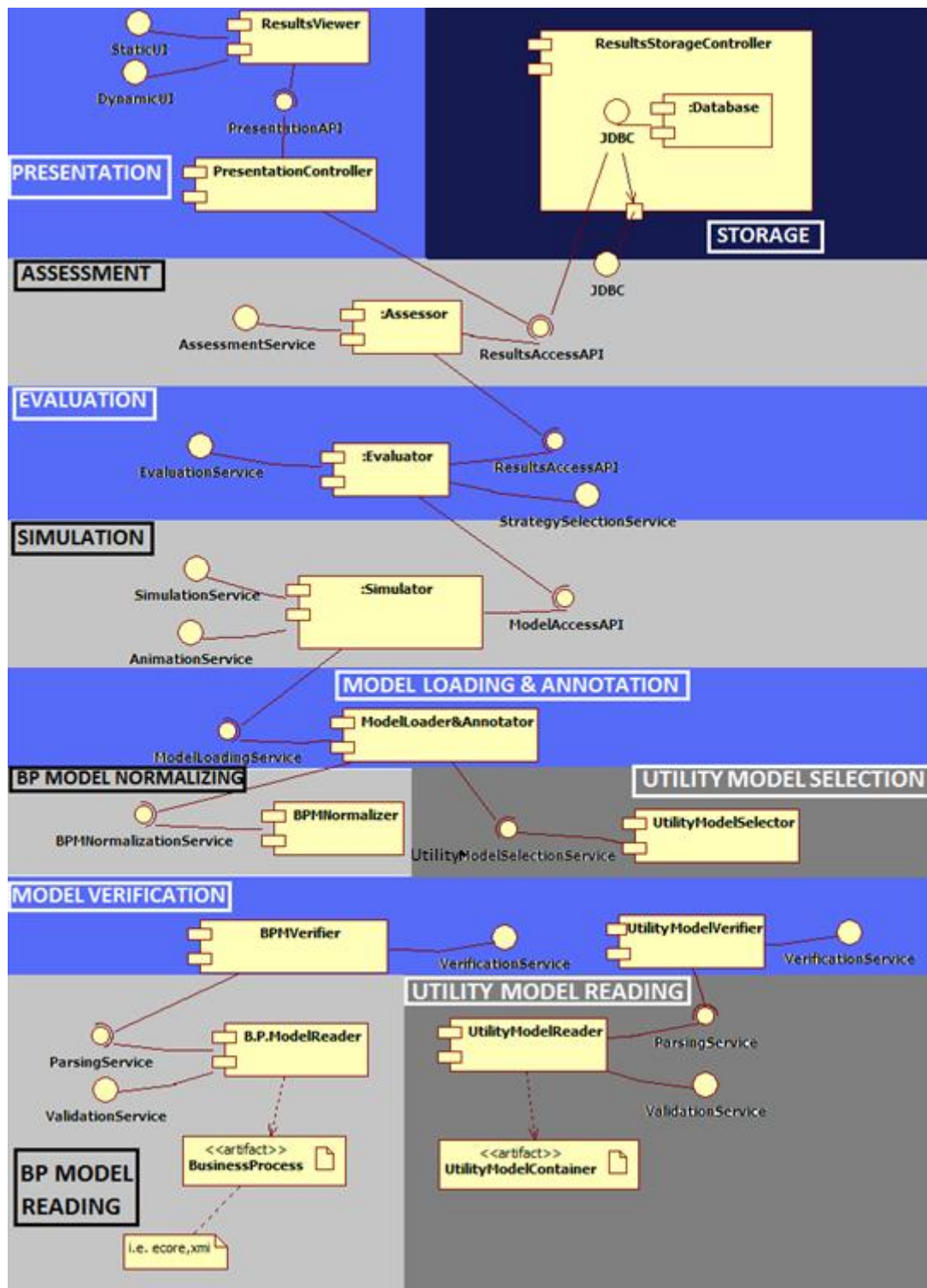
Σχεδιαστικός στόχος υπήρξε ο υψηλός βαθμός συντηρησιμότητας και δομικής ευελιξίας (structural portability) του συστήματος, οπότε το σύστημα υλοποιήθηκε με γνώμονα την κλειστή αρχιτεκτονική, χωρίς όμως να απαγορεύονται ενδεχόμενες προσβάσεις υποσυστημάτων σε προσφερόμενες υπηρεσίες υποσυστημάτων που δε βρίσκονται στο αμέσως παρακάτω επίπεδο, κάτι που μπορεί να συμβεί, για παράδειγμα, με την πρόσβαση σε υπολογιστικά αποτελέσματα, μέσω της κατάλληλης διεπαφής (application interface).

Βασικό στοιχείο του σχεδιασμού υπήρξε και η αντικειμενοστρέφεια. Η χρήση της επέτρεψε τη δόμηση του συστήματος σαν ένα σύνολο διακριτών και αλληλεπιδρώντων μονάδων (agents) που είναι κατανομημένες στα διάφορα επίπεδα (levels). Καθεμία από αυτές τις μονάδες είναι υπεύθυνη για συγκεκριμένες λειτουργίες, κάτι που επίσης συντελεί στη μεγάλη ευελιξία και επεκτασιμότητα του συστήματος και επιτρέπει την αλλαγή της υλοποίησης κάθε μονάδας, χωρίς να χρειάζεται να αλλάξουν οι μονάδες πελάτες τους, θεωρώντας ότι οι διαπροσωπείες παραμένουν οι ίδιες και εφαρμόζονται (get implemented) καταλλήλως.

Ο τρόπος επικοινωνίας αυτών των μονάδων μεταξύ τους είναι η μεθοδική ροή των δεδομένων από τη μια μονάδα στην άλλη (Τεχνοτροπία Ροής Δεδομένων).

Τα δεδομένα μεταφέρονται από τη μία μονάδα στην άλλη μόλις είναι διαθέσιμα, κατόπιν απαιτήσεως της μονάδας πελάτη, και όχι όταν όλη η λειτουργία της μονάδας ολοκληρωθεί. Επίσης, η κάθε μονάδα δεν γνωρίζει την ύπαρξη ή την κατάσταση των υπολοίπων (no shared state). Τα παραπάνω αποτελούν χαρακτηριστικά της τεχνοτροπίας Αυλού/Φίλτρου (Pipe and Filter) που συντελεί στην ευκολία συντήρησης και ανάλυσης της συμπεριφοράς του συστήματος (formal analysis - throughput and deadlock detection), ενώ ταυτόχρονα καθιστά εφικτή την πιθανότητα υλοποίησης παράλληλων διαδικασιών. Στον αντίποδα, κάθε φίλτρο αποτελεί σημαντικό κρίκο στην αλυσίδα της αλληλεπίδρασης των μονάδων και έτσι η διαχείριση συντριβής ενός φίλτρου είναι μια δύσκολη υπόθεση. Επιπρόσθετα, γίνεται ορατή η ανάγκη για κοινό παρονομαστή στην μορφολογία και τον τύπο των δεδομένων που διαχειρίζονται οι μονάδες του συστήματος.

Ακολουθεί το σχηματικό διάγραμμα του συστήματος (Σχήμα 3.1):



Σχήμα 3.1: Σχηματικό Διάγραμμα (Block Diagram) του συστήματος.

3.2 Συστατικά μέρη (Components)

3.2.1 Περιγραφές

Ακολουθεί η από-κάτω-προς-τα-πάνω (bottom-up) περιγραφή των διαφόρων συστατικών μερών του συστήματος:

- **MODEL READING:** Αυτό το επίπεδο χωρίζεται σε δύο μέρη :
 - UTILITY MODEL READING :** Το «Διάβασμα» μοντέλων χρησιμότητας (utility models), στο οποίο πραγματοποιείται ένα πρώτο πέρασμα (parsing) των μοντέλων και έλεγχος της εγκυρότητάς των (validation), με βάση προκαθορισμένες δομές που έχουν οριστεί στο σύστημα (model scemas).
 - BP MODEL READING :** Το «Διάβασμα» μοντέλων επιχειρηματικών διαδικασιών (business process models), στο οποίο, όπως και στο παραπάνω, πραγματοποιείται ένα πρώτο πέρασμα (parsing) των μοντέλων και έλεγχος της εγκυρότητάς των (validation).

Σε καθένα από τα παραπάνω μέρη του επιπέδου αυτού υπάρχει μία μονάδα (agent) που έχει το ρόλο του Αναγνώστη Μοντέλων (Model Reader) και προσφέρει τις υπηρεσίες της ανάγνωσης (Parsing Service) και ελέγχου εγκυρότητας των μοντέλων (Validation Service).
- **MODEL VERIFICATION:** Στο επίπεδο αυτό πραγματοποιείται η επαλήθευση εγκυρότητας (verification) των μοντέλων, τόσο των μοντέλων επιχειρηματικών διαδικασιών (business process models) όσο και των μοντέλων χρησιμότητας (utility models). Καθένα από αυτά τα μοντέλα υπόκειται σε ρητούς περιορισμούς και προδιαγραφές. Οι περιορισμοί μπορεί να είναι εκφρασμένοι ως ένα σύνολο λογικών εκφράσεων (logical expressions) ενώ οι προδιαγραφές μπορεί να είναι εγγενείς των μοντέλων, εκφρασμένες ως κάποια χαρακτηριστικά των ορισμένων δομών τους (model scemas). Είναι ευθύνη μιας μονάδας σε αυτό το επίπεδο, του Επαληθευτή Μοντέλων (Model Verifier), ο έλεγχος συμμόρφωσης των μοντέλων στους περιορισμούς και τις προδιαγραφές, προσφέροντας την υπηρεσία επαλήθευσης εγκυρότητας (Verification Service).
- **BUSINESS PROCESS MODEL NORMALIZING & UTILITY MODEL SELECTION:** Αυτό το επίπεδο χωρίζεται στα εξής δύο μέρη :
 - BUSINESS PROCESS MODEL NORMALIZING :** Ομαλοποίηση των επιχειρηματικών διαδικασιών και εξασφάλιση ότι βρίσκονται σε μορφή κατάλληλη προς επεξεργασία από το σύστημα . Για τη διαδικασία αυτή είναι υπεύθυνη μία

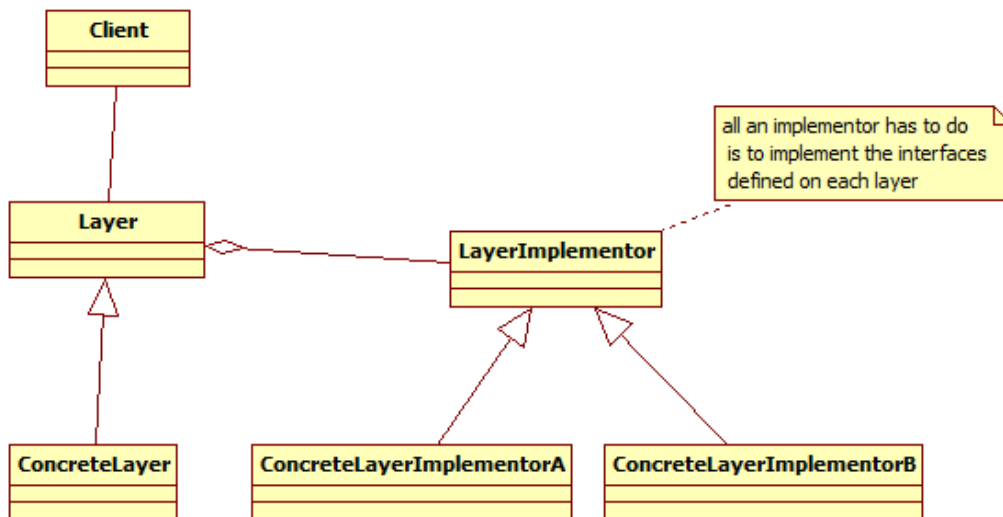
μονάδα, ο Κανονικοποιητής Μοντέλων (Model Normalizer) , ο οποίος προσφέρει την υπηρεσία κανονικοποίησης μοντέλων (Model Normalization Service).

→UTILITY MODEL SELECTION : Επιλογή του μοντέλου / των μοντέλων χρησιμότητας, με βάση το οποίο / τα οποία πραγματοποιείται η ανάλυση από το σύστημα. Για τη διαδικασία αυτή είναι υπεύθυνη μία μονάδα, ο Επιλογέας Μοντέλων (Model Selector) , ο οποίος προσφέρει την υπηρεσία επιλογής μοντέλων (Model Selection Service).

- **MODEL LOADING & ANNOTATION:** Στο επίπεδο αυτό πραγματοποιείται η «φόρτωση» (loading) του προς επεξεργασία ενοποιημένου μοντέλου στο σύστημα. Τα επιλεχθέντα μοντέλα, αφού έχουν ήδη διαβαστεί, επικυρωθεί, επαληθευτεί και κανονικοποιηθεί είναι πλέον έτοιμα για την εισαγωγή τους στο σύστημα ως ενοποιημένων και ολοκληρωμένων μονάδων, δηλαδή μοντέλων επιχειρηματικών διαδικασιών (business process models) ενοποιημένων με τα αντίστοιχα μοντέλα χρησιμότητας (utility models). Για τη διαδικασία αυτή είναι υπεύθυνη μία μονάδα, ο Φορτωτής και Επισημειωτής Μοντέλων (Model Loader & Annotator) , ο οποίος προσφέρει την υπηρεσία φόρτωσης μοντέλων (Model Loading Service).
- **SIMULATION:** Στο επίπεδο αυτό πραγματοποιείται η προσομοίωση των παραμετροποιημένων επιχειρηματικών διαδικασιών. Για τη διαδικασία αυτή είναι υπεύθυνη μία μονάδα, ο Προσομοιωτής (Simulator) , ο οποίος προσφέρει την υπηρεσία προσομοίωσης (Simulation Service). Με αυτή την υπηρεσία παράγονται όλες οι δυνατές εναλλακτικές πορείες που δύνανται να ακολουθήσουν οι διάφορες επιχειρηματικές διαδικασίες, κάτι που είναι απαραίτητο για την ανάλυση που πραγματοποιεί το σύστημα. Καλό είναι σε αυτό το επίπεδο, της προσομοίωσης, να παρέχεται και μία υπηρεσία γραφικής αναπαράστασης των διαδικασιών (Animation Service) καθώς επίσης και μια διεπαφή για την πρόσβαση στα μοντέλα (Model Access API).
- **EVALUATION:** Σε αυτό το επίπεδο πραγματοποιούνται όλοι οι υπολογισμοί των διαφόρων παραμέτρων του μοντέλου και η δημιουργία προσβάσιμων αποτελεσμάτων. Για τη διαδικασία αυτή είναι υπεύθυνη μία μονάδα, ο Εκτιμητής (Evaluator) , ο οποίος προσφέρει την υπηρεσία εκτίμησης (Evaluation Service). Με αυτή την υπηρεσία υπολογίζονται οι διάφορες παράμετροι των επιχειρηματικών διαδικασιών, με βάση κάποια στρατηγική, που επιλέγεται μέσω μίας ακόμη υπηρεσίας που προσφέρεται σε αυτό το επίπεδο, της υπηρεσίας επιλογής Στρατηγικής (Strategy Selection Service). Καλό είναι σε αυτό το επίπεδο, της αξιολόγησης, να παρέχεται και μια διεπαφή για την πρόσβαση στα αποτελέσματα (Results Access API).

- **ASSESSMENT:** Στο επίπεδο αυτό πραγματοποιείται η επεξεργασία αποτελεσμάτων όπως αυτά προκύπτουν από το επίπεδο εκτίμησης (evaluation layer) για την εκτίμηση συνολικών μεγεθών και εξαγωγή συμπερασμάτων για την όλη επιχειρηματική διαδικασία. Για τη διαδικασία αυτή είναι υπεύθυνη μία μονάδα, ο Αξιολογητής (Assessor) , ο οποίος προσφέρει την υπηρεσία αξιολόγησης (Assessment Service). Με αυτή την υπηρεσία εκτιμώνται σημαντικά συνολικά μεγέθη και, με βάση συγκεκριμένα κριτήρια, πραγματοποιείται η αξιολόγηση του συστήματος και εντοπίζονται οι βέλτιστες επιλογές. Καλό είναι και σε αυτό το επίπεδο, της αξιολόγησης, να παρέχεται και μια διεπαφή για την πρόσβαση στα αποτελέσματα (Results Access API).
- **RESULT PRESENTATION & STORAGE:** Σε αυτό το επίπεδο γίνεται η παρουσίαση, με τρόπο που επιλέγεται από το χρήστη, των αποτελεσμάτων, όπως αυτά προκύπτουν από το επίπεδο αξιολόγησης (assessment layer), και η αποθήκευση αυτών. Οι μονάδες που είναι υπεύθυνες για αυτές τις λειτουργίες, ο Ρυθμιστής Παρουσίασης (Presentation Controller) και ο Ρυθμιστής Αποθήκευσης (Storage Controller) προσφέρουν αντίστοιχα τις διεπαφές για την παρουσίαση και αποθήκευση των αποτελεσμάτων, ενώ ο πρώτος είναι αυτός που αλληλεπιδρά με τους χρήστες μέσω στατικών ή δυναμικών διεπαφών (Static UI & Dynamic UI), εκμαιεύοντας από αυτούς τις διάφορες επιλογές για την ανάλυση που πραγματοποιείται από το σύστημα.

Να σημειωθεί ότι η ιεραρχική δόμηση των αφαιρετικών μοντέλων του συστήματος σε στιβάδες (layers) και παράλληλα η αντίστοιχη ιεραρχική δόμηση υλοποιήσεων αυτών των μοντέλων, επιτρέπει να επιλέξουμε την υλοποίηση που θέλουμε να χρησιμοποιήσουμε από το ίδιο μας τον κώδικα κατά την εκτέλεση του. Οι κλάσεις των αφαιρετικών μοντέλων και των υλοποιήσεων τους, μπορούν να υλοποιηθούν και να συνδυαστούν ανεξάρτητα. Έχουμε δηλαδή τη δυνατότητα χρησιμοποίησης του σχεδιαστικού μοτίβου bridge, όπως φαίνεται στο παρακάτω διάγραμμα (Σχήμα 3.2):



Σχήμα 3.2: Διάγραμμα κλάσεων του σχεδιαστικού μοτίβου Bridge.

3.2.2 Διεπαφές (Interfaces) και Υπηρεσίες (Services)

Οι υπηρεσίες που προσφέρονται από τα διάφορα συστατικά του συστήματος, όπως αναφέρθηκαν στο προηγούμενο κεφάλαιο, υλοποιούνται με συγκεκριμένες μεθόδους, που ορίζονται ρητά μέσα σε συγκεκριμένες διεπαφές (interfaces).

Στον ακόλουθο πίνακα (Πίνακας 3.1) δίνονται οι διεπαφές του συστήματος και για καθεμία από αυτές οι μέθοδοι που περιέχει, με τα ορίσματα, τον τύπο επιστροφής και μια σύντομη περιγραφή τους:

<u>Interface Name</u>	<u>Method</u>	<u>Arguments</u>	<u>Return Type</u>	<u>Description</u>
ValidationService	checkIfValid	String ModelFile, String ModelType	Boolean	returns true only if model is valid(conforms to its metamodel)
ModelEditingService	loadModel	String ModelFile	Model	loads a model given its file path
	saveModel	String ModelFile	void	saves a model at given file path
	editModel	String ModelFile	void	edit a model
	deleteModel	String ModelFile	void	deletes a model at

				given file path
VerificationService	checkIfVerified	String ModelFile, String ModelType, Characteristics c	boolean	returns true only if model is verified, according to constraints
BusinessProcess NormalizationService				
SimulationService	findAllPaths		Container <List<Business ProcessTask>>	returns all paths through a Business Process
ModelAccessAPI				
AnimationService				
AtomicBusinessProcess TaskEvaluationService	calculateValue	Estimation Strategy, AtomicBusiness ProcessTask abpt, ConcreteUtility Model cum	ValueType	returns the estimated total value of an atomicBPTask
				based on an estimation Strategy
	getName	AtomicBusiness ProcessTask abpt	String	returns the name of an atomicBPTask
	calculateAtomic UtilityElementC ontribution	AtomicUtility Element aue	ValueType	Returns the contribution of an atomicUtility Element to the ConcreteUtility Model
	getAtomicUtilit yElementsByAs	AtomicBusinessP rocessTask abpt,	LinkedList <AtomicUtility	Returns the atomicUtility

	ending Contribution	ConcreteUtility Model cum	Element>	Elements of an atomicBPTask
				ordered by ascending contribution
	getAtomic UtilityElements ByDescending Contribution	AtomicBusiness ProcessTask abpt, ConcreteUtility Model cum	LinkedList <AtomicUtility Element>	Returns the atomicUtility Elements of an atomicBPTask
				ordered by descending contribution
BusinessProcessTask EvaluationService	calculateValue	Estimation Strategy es, BusinessProcess Task bpt, ConcreteUtility Model cum	ValueType	returns the estimated total value of a BPTask based on an estimation Strategy
ResultsAccessAPI				
AssessmentService	chosePath	Assessment Strategy as, Assessment Criteria ac	BusinessProcess Task	returns the prefered business process path based on certain criteria
	getBusiness ProcessTasksBy Ascending Preferability	Assessment Strategy as, Assessment Criteria ac	LinkedList <BusinessProcess Task>	returns the paths of a Business Process ordered by ascending preferability, according to criteria
	getBusiness ProcessTasksBy	AssessmentStrate gy as, Assessment	LinkedList<Busin essProcessTask>	returns the paths of a Business Process ordered

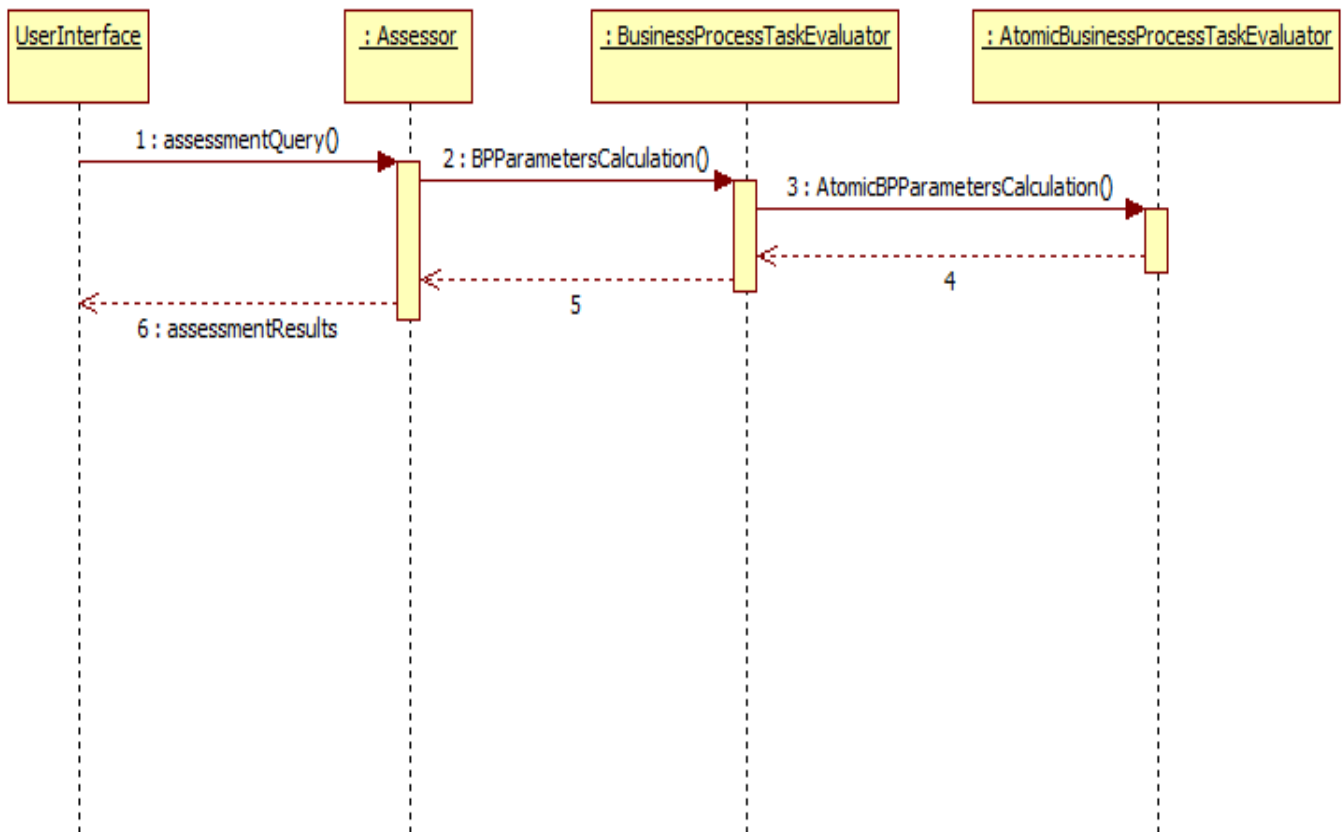
	Descending Preferability	Criteria ac		by descending preferability, according to criteria
	getExpected BusinessProcess Value	Assessment Strategy as, UtilityModel um	ValueType	returns the business processe's expected value of a specific utility, based on assessment strategy
PresentationAPI				
StaticUI				
DynamicUI				
JDBC				

Πίνακας 3.1: Διεπαφές (Interfaces) του συστήματος

3.3 Περιγραφή Διαδικασίας(Process Outline)

Στην ενότητα αυτή πραγματοποιείται η ανάλυση των απαιτήσεων για τις λειτουργίες του συστήματος υπό μορφή ακολουθιακών διαγραμμάτων (sequence diagrams).

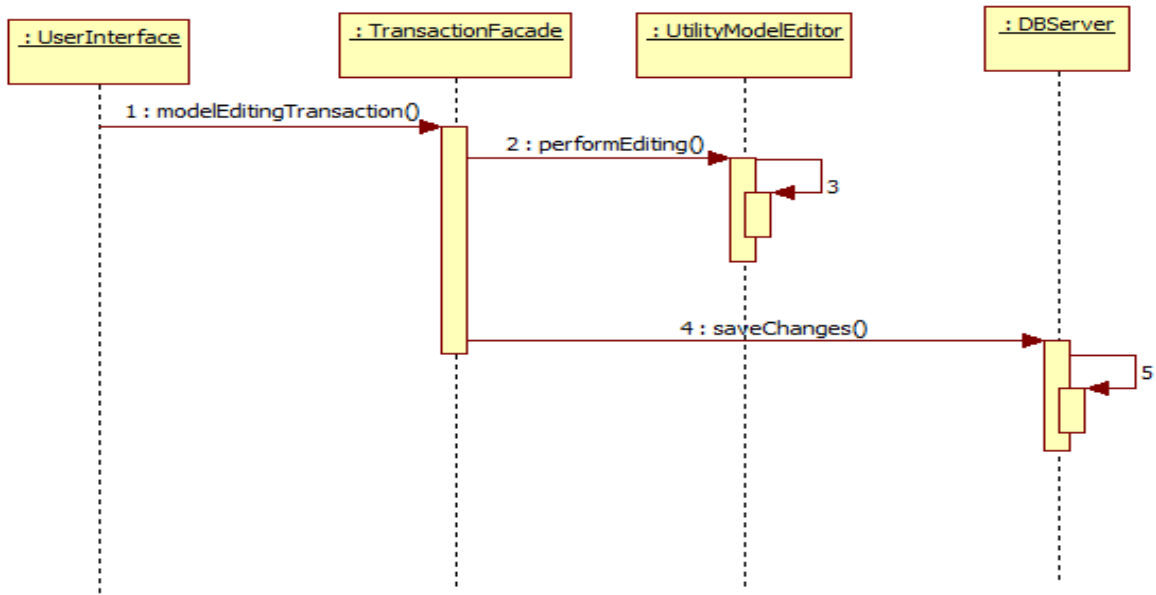
Το παρακάτω ακολουθιακό διάγραμμα (Σχήμα 3.3) αναφέρεται, σε ένα υψηλό αφαιρετικό επίπεδο, στη συνεργασία των διαφόρων βασικών μονάδων του συστήματος για την παραγωγή ζητηθέντων αποτελεσμάτων/συμπερασμάτων για μια επιχειρηματική διαδικασία:



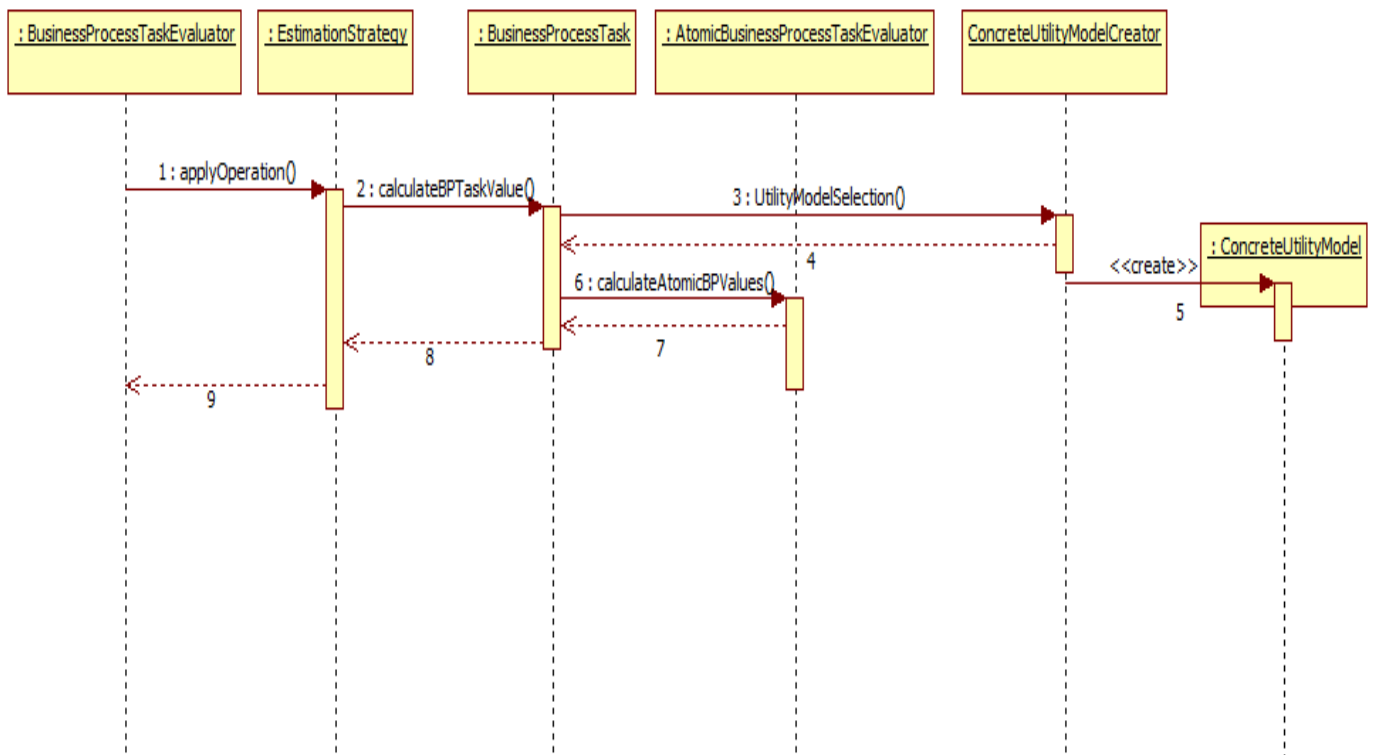
Σχήμα 3.3: Ακολουθιακό Διάγραμμα σε υψηλό αφαιρετικό επίπεδο για την παραγωγή ζητηθέντων αποτελεσμάτων/συμπερασμάτων

Όπως φαίνεται και από το διάγραμμα, έχει επιλεγεί για τη λειτουργία του συστήματος ο **κατανεμημένος έλεγχος (distributed control)**, με βάση τον οποίο η επεξεργασία κατανέμεται μεταξύ των συμμετοχόντων, με καθέναν από αυτούς να εκτελεί ένα τμήμα του αλγορίθμου. Αυτή η επιλογή συνεπάγεται όλα τα οφέλη του αντικειμενοστρεφούς σχεδιασμού.

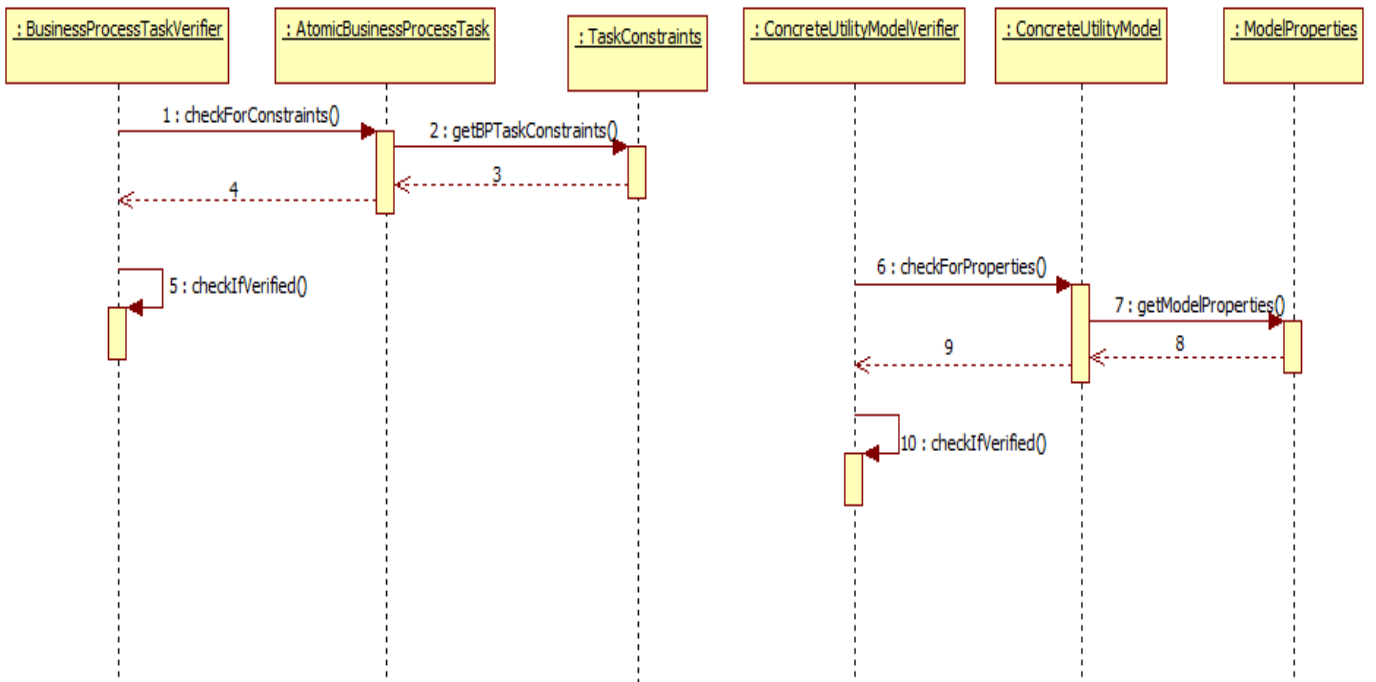
Ακολουθεί μια «προσεκτικότερη ματιά» σε συγκεκριμένες λειτουργίες του συστήματος, και συγκεκριμένα την επεξεργασία μοντέλων (model editing), την αποτίμηση (evaluation), την επαλήθευση (verification), την αξιολόγηση (assessment) και την παρουσίαση και αποθήκευση των αποτελεσμάτων (presentation & storage), στα Σχήματα 3.4, 3.5, 3.6, 3.7 και 3.8 αντίστοιχα:



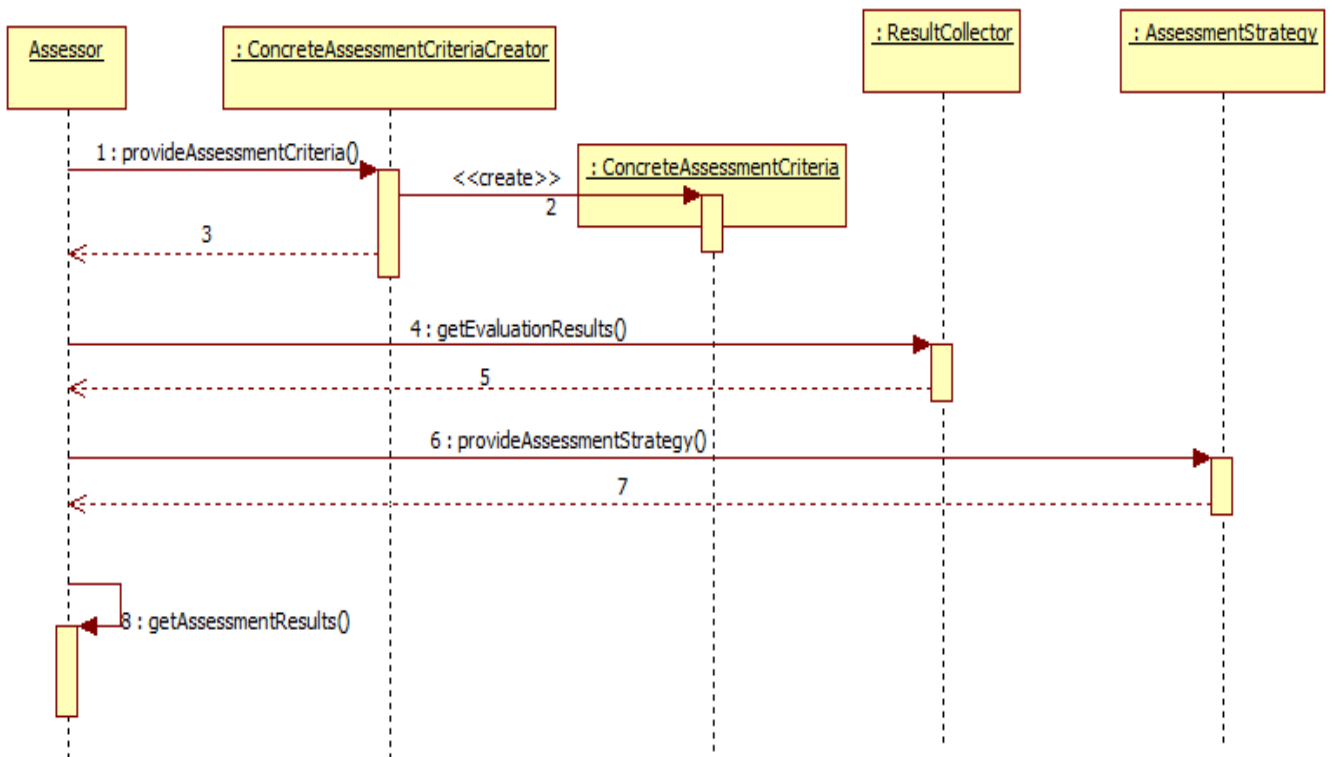
Σχήμα 3.4: Ακολουθιακό Διάγραμμα για την επεξεργασία μοντέλων (model editing)



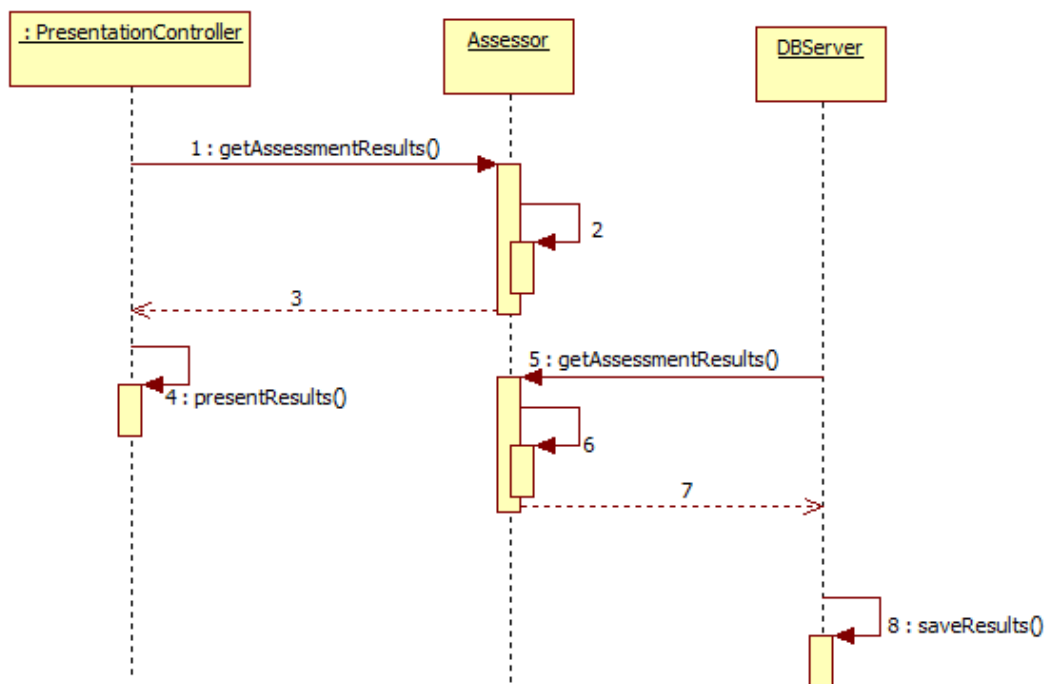
Σχήμα 3.5: Ακολουθιακό Διάγραμμα για την αποτίμηση (evaluation)



Σχήμα 3.6: Ακολουθιακό Διάγραμμα για την επαλήθευση (verification)



Σχήμα 3.7: Ακολουθιακό Διάγραμμα για την αξιολόγηση (assessment)



Σχήμα 3.8: Ακολουθιακό Διάγραμμα για την αποθήκευση των αποτελεσμάτων (presentation & storage)

4

Προδιαγραφή Μοντέλων

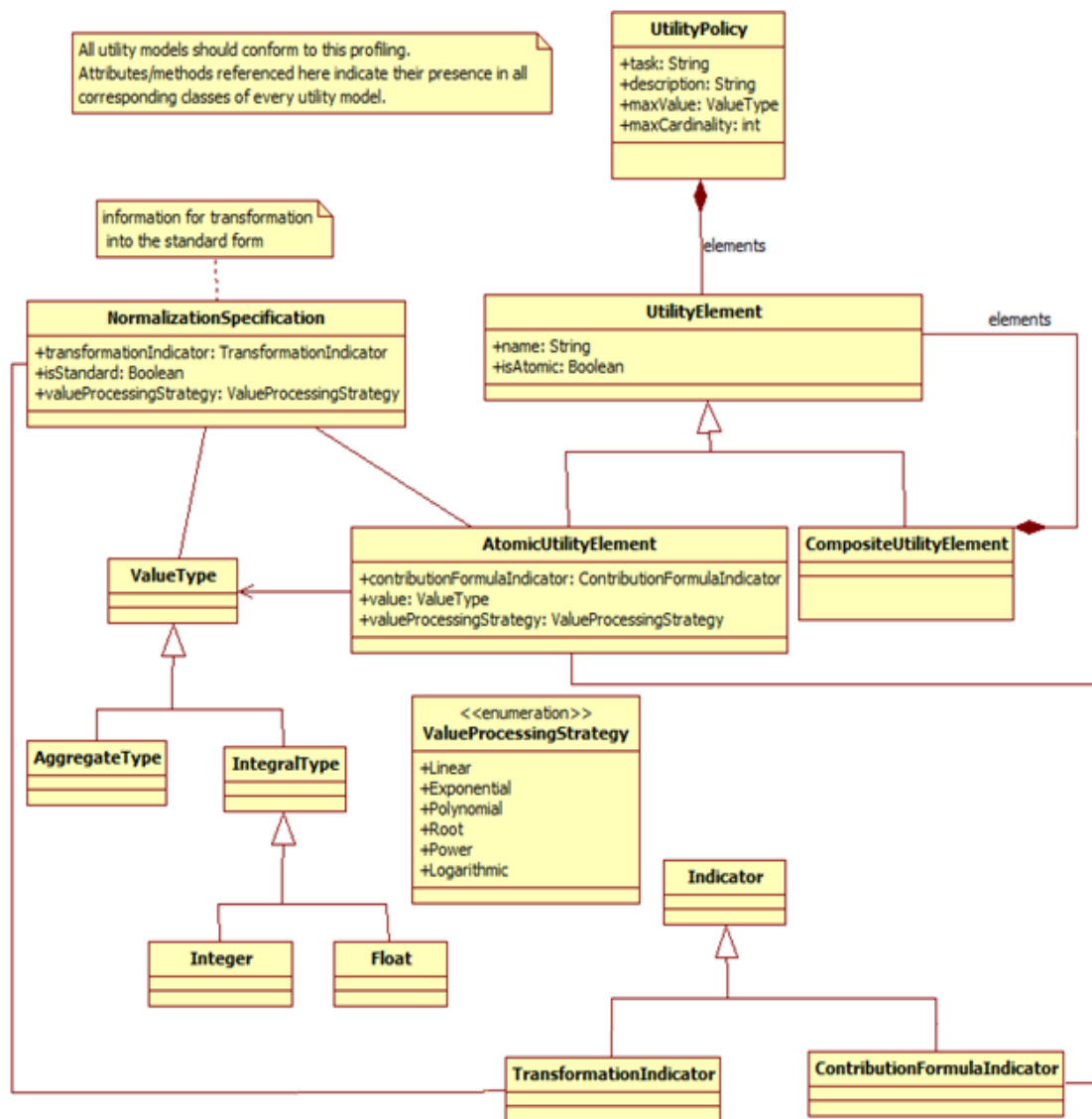
Σε αυτό το κεφάλαιο παρουσιάζονται τα μοντέλα που χρησιμοποιούνται από το σύστημα και μια σύντομη περιγραφή τους. Τα μοντέλα αυτά αποτελούν σημαντικό στοιχείο του συστήματος, καθώς προσδιορίζουν τα κρίσιμα χαρακτηριστικά των επιχειρηματικών διαδικασιών (business processes), με βάση τα οποία πραγματοποιεί το σύστημα την ανάλυσή του.

Τα κρίσιμα αυτά χαρακτηριστικά αποτελούν πρωτογενή, μετρήσιμα μεγέθη, όπως το κόστος (cost) και η ρυθμαπόδοση (throughput). Με βάση αυτά, μπορεί να προκύψει ένα πλήθος παραγόμενων δεικτών επίδοσης (KPIs), όπως η Απόδοση των Επενδύσεων (Return On Investment) και ο λόγος του Κέρδους προς την Προσπάθεια (Profit to Effort ratio), ενώ με κατάλληλο συνδυασμό αυτών μπορούν να προκύψουν άλλοι σημαντικοί δείκτες κ.ο.κ.. Έχει γίνει προσπάθεια τα μοντέλα αυτά να καλύπτουν ένα ουσιαστικό και όσο το δυνατόν ευρύτερο φάσμα χαρακτηριστικών, παρέχοντας τη δυνατότητα περιγραφής και ανάλυσης επιχειρηματικών διαδικασιών ως προς μια ικανοποιητικά πλούσια γκάμα ιδιοτήτων. Εντούτοις, χάρη στη συντηρησιμότητα και δομική ευελιξία του συστήματος, καθίσταται δυνατή και εύκολη η προσθήκη νέων μοντέλων και η επεξεργασία ή διαγραφή των ήδη υπάρχοντων. Οι επιχειρηματικές διαδικασίες στις οποίες αναφέρονται αρκετά από αυτά τα μοντέλα είναι από το χώρο της Τεχνολογίας Λογισμικού (Software Engineering) και για το λόγο αυτό έχουν το πρόθεμα “SE” στο όνομα της βασικής κλάσης τους, π.χ. SEQualityModel.

Ακολουθεί μία περιγραφή της σκιαγράφησης (profiling) αυτών των μοντέλων και στη συνέχεια, η παρουσίαση και επεξήγηση καθενός από αυτά.

4.1 Σκιαγράφηση των Μοντέλων (Model Profiling)

Το παρακάτω διάγραμμα (Σχήμα 3.9) δίνει μία εικόνα της μορφής ενός μοντέλου πολιτικής χρησιμότητας (Utility Policy) :



Σχήμα 3.9: Σκιαγράφηση των Μοντέλων (Model Profiling)

Ακολουθεί μια σύντομη περιγραφή για κάθε σημαντικό στοιχείο του παραπάνω διαγράμματος:

- **UtilityPolicy** : Είναι η πολιτική χρησιμότητας με βάση την οποία πραγματοποιείται η ανάλυση μιας επιχειρηματικής διαδικασίας. Παραδείγματα τέτοιων πολιτικών είναι το κόστος, η ρυθμολόγηση κ.α.. Κάθε «στιγμιότυπο» μιας πολιτικής χρησιμότητας

αντιστοιχίζεται σε ένα Έργο (Task) ή αλλιώς σε μία απλή υποδιαδικασία (βλ. Atomic Task, Ενότητα 5).

→task : Ένα όνομα ή κάποιο αναγνωριστικό (id) της απλής υποδιαδικασίας στην οποία αντιστοιχίζεται το συγκεκριμένο «στιγμιότυπο» της πολιτικής χρησιμότητας.

→description : Μία περιγραφή της συγκεκριμένης πολιτικής χρησιμότητας και τυχόν απαραίτητα σχόλια.

→maxValue : Ένας αριθμός που υποδηλώνει τη μέγιστη τιμή που μπορεί να λάβει η (κανονικοποιημένη) τιμή (value , βλ. παρακάτω) ενός στοιχείου αυτής της πολιτικής χρησιμότητας.

→maxCardinality : Ένας αριθμός που δηλώνει τη μέγιστη πληθικότητα των στοιχείων (elements , βλ. παρακάτω) της συγκεκριμένης πολιτικής χρησιμότητας.

- UtilityElement : Είναι ένα στοιχείο (element) της συγκεκριμένης πολιτικής χρησιμότητας. Το «στιγμιότυπο» μιας πολιτικής χρησιμότητας αντιστοιχίζεται σε ένα Έργο (Task) ή αλλιώς σε μία απλή υποδιαδικασία (βλ. Atomic Task, Ενότητα 5), δηλαδή σε μια υποδιαδικασία που δε διασπάται περαιτέρω σε απλούστερες υποδιαδικασίες. Εντούτοις, το χαρακτηριστικό (utility) ως προς το οποίο μελετάται μία απλή υποδιαδικασία ενδέχεται να διασπάται σε περισσότερα του ενός τμήματα για την ίδια υποδιαδικασία. Για παράδειγμα, η απλή υποδιαδικασία Task_A ενδέχεται να εμπεριέχει π.χ. κόστη (costs) διαφορετικού μεταξύ τους τύπου και για διαφορετικά εμπλεκόμενα τμήματα. Καθένα από αυτά θα περιγράφεται από ένα διαφορετικό utility element και μπορεί να υπάρχει συγκεκριμένη και ενδεχομένως ιεραρχική ομαδοποίησή τους.

→name : Ένα όνομα ή κάποιο αναγνωριστικό (id) του συγκεκριμένου στοιχείου.

→isAtomic : Μία λογική τιμή (boolean) που υποδηλώνει την ατομικότητα ή μη του συγκεκριμένου στοιχείου, δηλαδή για τα AtomicUtilityElements αυτή η μεταβλητή παίρνει την τιμή αληθής (true) ενώ για τα CompositeUtilityElements την τιμή ψευδής (false).

- CompositeUtilityElement : Είναι ένα στοιχείο της συγκεκριμένης πολιτικής χρησιμότητας το οποίο είναι σύνθετο, δηλαδή αποτελείται από άλλα στοιχεία, είτε απλά (AtomicUtilityElements) είτε σύνθετα (CompositeUtilityElements). Για παράδειγμα μπορεί να αντιπροσωπεύει κάποιο συγκεκριμένο είδος στοιχείων στη συγκεκριμένη πολιτική χρησιμότητας.
- AtomicUtilityElement : Είναι ένα στοιχείο της συγκεκριμένης πολιτικής χρησιμότητας το οποίο είναι απλό, δηλαδή δεν αποτελείται από άλλα στοιχεία. Αυτά

τα στοιχεία αποτελούν δομικά συστατικά της συγκεκριμένης πολιτικής χρησιμότητας, καθώς περιέχουν όλη την πληροφορία για τα ποσοτικοποιημένα χαρακτηριστικά της υποδιαδικασίας που αντιστοιχίζεται με τη συγκεκριμένη πολιτικής χρησιμότητας.

→value: Είναι η τιμή του συγκεκριμένου απλού στοιχείου μιας συγκεκριμένης πολιτικής χρησιμότητας, δηλαδή ένας αριθμός που προσδιορίζει ποσοτικά το μέγεθος στο οποίο αναφέρεται η συγκεκριμένη πολιτική χρησιμότητας. Όπως θα φανεί και από την παρουσίαση των συγκεκριμένων μοντέλων (concrete models) η τιμή αυτή ενδέχεται να χρειάζεται υπολογισμό με βάση συγκεκριμένους δείκτες, ο οποίος διαφέρει ανάλογα με το μοντέλο. Να σημειωθεί ότι είτε δίνεται, είτε προκύπτει κατόπιν υπολογισμού, αυτή η τιμή είναι «ακατέργαστη», χρίζει δηλαδή επεξεργασίας προτού χρησιμοποιηθεί για τη συμβολή της στον υπολογισμό του συνολικού μεγέθους που προκύπτει από όλα τα στοιχεία της συγκεκριμένης πολιτικής χρησιμότητας. Το είδος αυτής της επεξεργασίας προσδιορίζεται από τα επόμενα πεδία:

→valueProcessingStrategy : Είναι η στρατηγική με βάση την οποία εφαρμόζεται μία συνάρτηση στην κανονικοποιημένη τιμή (normalized value) και προκύπτει η συμβολή του συγκεκριμένου στοιχείου στον υπολογισμό της συνολικής ζητούμενης τιμής.

→contributionFormulaIndicator : Μία δομή που καταδεικνύει τον τρόπο με τον οποίο εφαρμόζεται η συνάρτηση της στρατηγικής μεταποίησης της κανονικοποιημένης τιμής (valueProcessingStrategy). Παρακάτω, στο σημείο όπου περιγράφονται οι διάφορες στρατηγικές μεταποίησης (valueProcessingStrategies) ακολουθεί μια πιο συγκεκριμένη περιγραφή, όπου φαίνεται και η χρήση αυτών των πεδίων.

- Indicator : Είναι μία δομή, και συγκεκριμένα μία ταξινομημένη συλλογή αριθμών (π.χ. LinkedList<Number>) που περιέχει κάποιους συντελεστές για την εφαρμογή των διαφόρων στρατηγικών μεταποίησης (ValueProcessingStrategies). Στα μοντέλα του συστήματος εμφανίζονται οι εξής συντελεστές:

→ ContributionFormulaIndicator , όπως έχει προαναφερθεί και

→ TransformationIndicator : δείκτης που προσδιορίζει τον τρόπο με τον οποίο εφαρμόζεται η συνάρτηση μιας στρατηγικής μεταποίησης της τιμής (value) για την κανονικοποίησή της, ότι κι αν αυτό σημαίνει π.χ. μετατροπή της τιμής σε μία κοινώς επεξεργάσιμη μορφή, όπως σε μια πρότυπη μονάδα μέτρησης.

- ValueProcessingStrategy : Είναι μία δομή απαρίθμησης (enumeration) που απαριθμεί τις παρακάτω στρατηγικές μεταποίησης:

→Linear : Η στρατηγική που υποδεικνύει την εφαρμογή ενός γραμμικού τελεστή σε μια τιμή, δηλαδή την επιστροφή της τιμής:

$a1 * x$, όπου:

- a1 : πρώτο στοιχείο ενός αντικειμένου τύπου Indicator
- x : η τιμή στην οποία εφαρμόζεται ο τελεστής

→Exponential : Η στρατηγική που υποδεικνύει την εφαρμογή ενός εκθετικού τελεστή σε μια τιμή, δηλαδή την επιστροφή της τιμής:

$a0*(e^{(a1*x)})$, όπου:

- $a0$, $a1$: πρώτο και δεύτερο στοιχείο ενός αντικειμένου τύπου Indicator
- x : η τιμή στην οποία εφαρμόζεται ο τελεστής

→Polynomial : Η στρατηγική που υποδεικνύει την εφαρμογή ενός πολυωνυμικού τελεστή σε μια τιμή, δηλαδή την επιστροφή της τιμής:

$a0+a1*x+a2*x^2+a3*x^3+...$, όπου:

- $a0$, $a1$, $a2$, $a3$, ... : πρώτο, δεύτερο, τρίτο,... στοιχείο ενός αντικειμένου τύπου Indicator
- x : η τιμή στην οποία εφαρμόζεται ο τελεστής

→Root : Η στρατηγική που υποδεικνύει την εφαρμογή ενός ριζικού τελεστή σε μια τιμή, δηλαδή την επιστροφή της τιμής:

$a0*\sqrt{x}$, όπου:

- $a0$: πρώτο στοιχείο ενός αντικειμένου τύπου Indicator
- x : η τιμή στην οποία εφαρμόζεται ο τελεστής

→Power : Η στρατηγική που υποδεικνύει την εφαρμογή ενός τελεστή ύψωσης σε δύναμη σε μια τιμή, δηλαδή την επιστροφή της τιμής:

$a0*(x^{a1})$, όπου:

- $a0$, $a1$: πρώτο και δεύτερο στοιχείο ενός αντικειμένου τύπου Indicator
- x : η τιμή στην οποία εφαρμόζεται ο τελεστής

→Logarithmic : Η στρατηγική που υποδεικνύει την εφαρμογή ενός λογαριθμικού τελεστή σε μια τιμή, δηλαδή την επιστροφή της τιμής:

$a0*\log(a1*x)$, όπου:

- $a0$, $a1$: πρώτο και δεύτερο στοιχείο ενός αντικειμένου τύπου Indicator
- x : η τιμή στην οποία εφαρμόζεται ο τελεστής

- NormalizationSpecification : Κάθε απλό στοιχείο (AtomicUtilityElement) σχετίζεται με μία προδιαγραφή κανονικοποίησης (NormalizationSpecification), για την κανονικοποίηση της τιμής του (value) σε μια πρότυπη μορφή.

→ transformationIndicator : δείκτης που προσδιορίζει τον τρόπο με τον οποίο εφαρμόζεται η συνάρτηση μιας στρατηγικής μεταποίησης της τιμής (value) για την κανονικοποίησή της .

→ valueProcessingStrategy : Είναι η στρατηγική με βάση την οποία εφαρμόζεται μία συνάρτηση στην τιμή (value) για την κανονικοποίησή της, με βάση και τον δείκτη (transformationIndicator).

→ isStandard : Μία λογική τιμή (boolean) που υποδηλώνει αν η τιμή value είναι σε πρότυπη μορφή ή όχι, δηλαδή για τα AtomicUtilityElements με κανονικοποιημένες τιμές αυτή η μεταβλητή παίρνει την τιμή αληθής (true) και υποδηλώνει πως δε χρειάζεται η διαδικασία κανονικοποίησής τους, ενώ για αυτά με μη κανονικοποιημένες τιμές την τιμή ψευδής (false).

- ValueType : Είναι ο τύπος των διαφόρων τιμών (values) των μοντέλων. Όπως φαίνεται και από το διάγραμμα, ο τύπος αυτός μπορεί να είναι είτε πρωτογενής και αναπόσπαστος (integral) είτε παραγόμενος (aggregate). Για να συμπεριλάβουμε σε αυτούς τους τύπους τόσο απλές δομές, όπως απλές αριθμητικές τιμές, όσο και πιο σύνθετες δομές, όπως τιμές δεικτών (indicators), στο σύστημα αυτοί οι τύποι ορίζονται ως συλλογές πρωτογενών τύπων, για παράδειγμα ο τύπος Integer υλοποιείται ως μία λίστα ακεραίων.

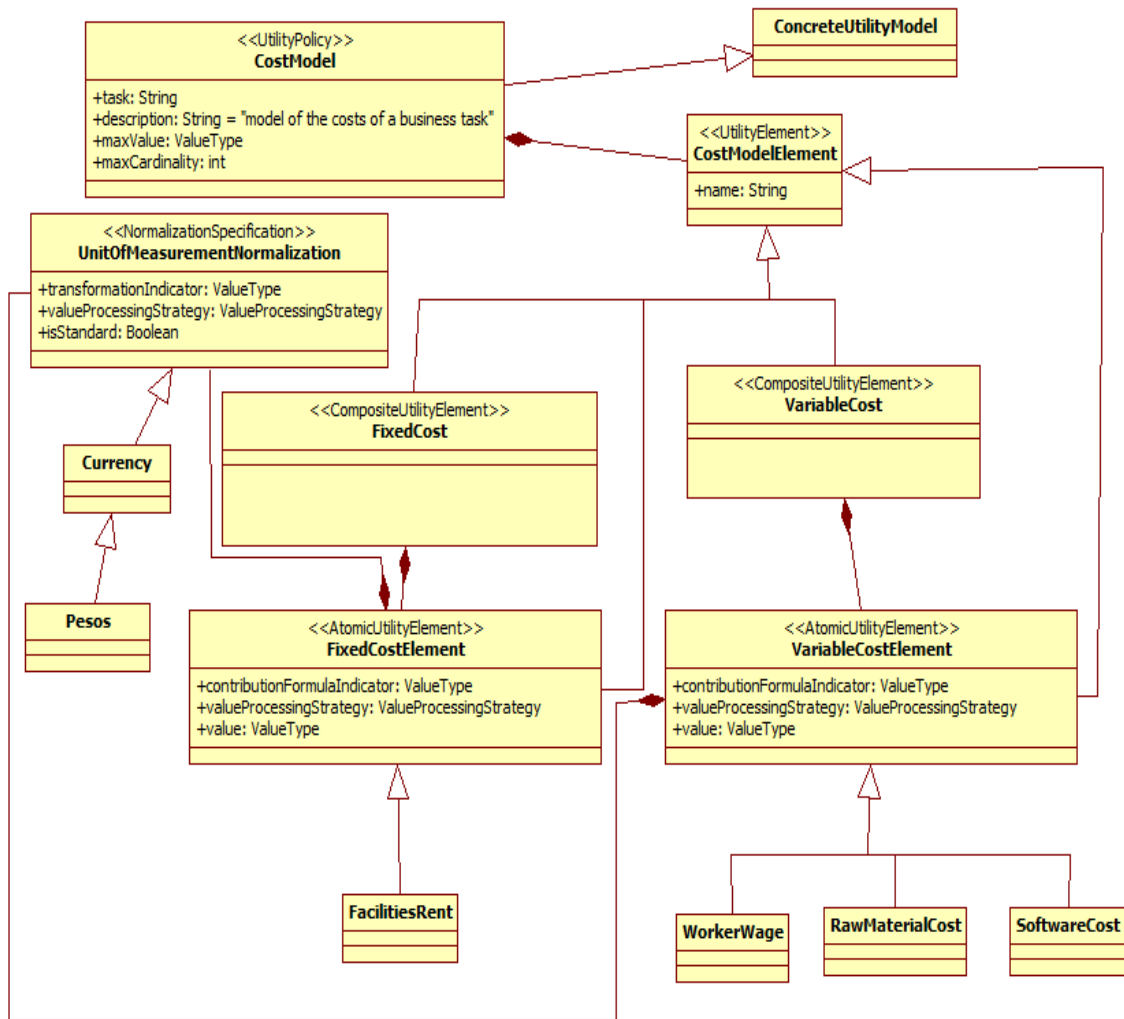
4.2 Συγκεκριμένα Μοντέλα (Concrete Models)

Στην ενότητα αυτή παρουσιάζονται τα συγκεκριμένα μοντέλα (concrete models) που χρησιμοποιούνται από το σύστημα. Επίσης γίνεται μια σύντομη επεξήγηση όσων στοιχείων αυτών των μοντέλων δεν έχουν ήδη καλυφθεί στην προηγούμενη ενότητα.

Τα μοντέλα που θα παρουσιαστούν συμμορφώνονται στη σκιαγράφιση (profiling) που παρουσιάστηκε στην προηγούμενη ενότητα, ανήκουν στο ίδιο αφαιρετικό επίπεδο και αποτελούν όλα υποκλάσεις της κλάσης ConcreteUtilityModel.

4.2.1 Μοντέλο Κόστους (Cost Model)

Το παρακάτω διάγραμμα (Σχήμα 3.10) δίνει μία εικόνα της μορφής ενός μοντέλου κόστους (CostModel), που περιγράφει τους οικονομικούς πόρους που απαιτείται να σπαταληθούν για τη διεκπεραίωση ενός Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task) :



Σχήμα 3.10: Μοντέλο Κόστους (CostModel)

Ακολουθεί μια σύντομη περιγραφή για κάθε σημαντικό στοιχείο του παραπάνω διαγράμματος:

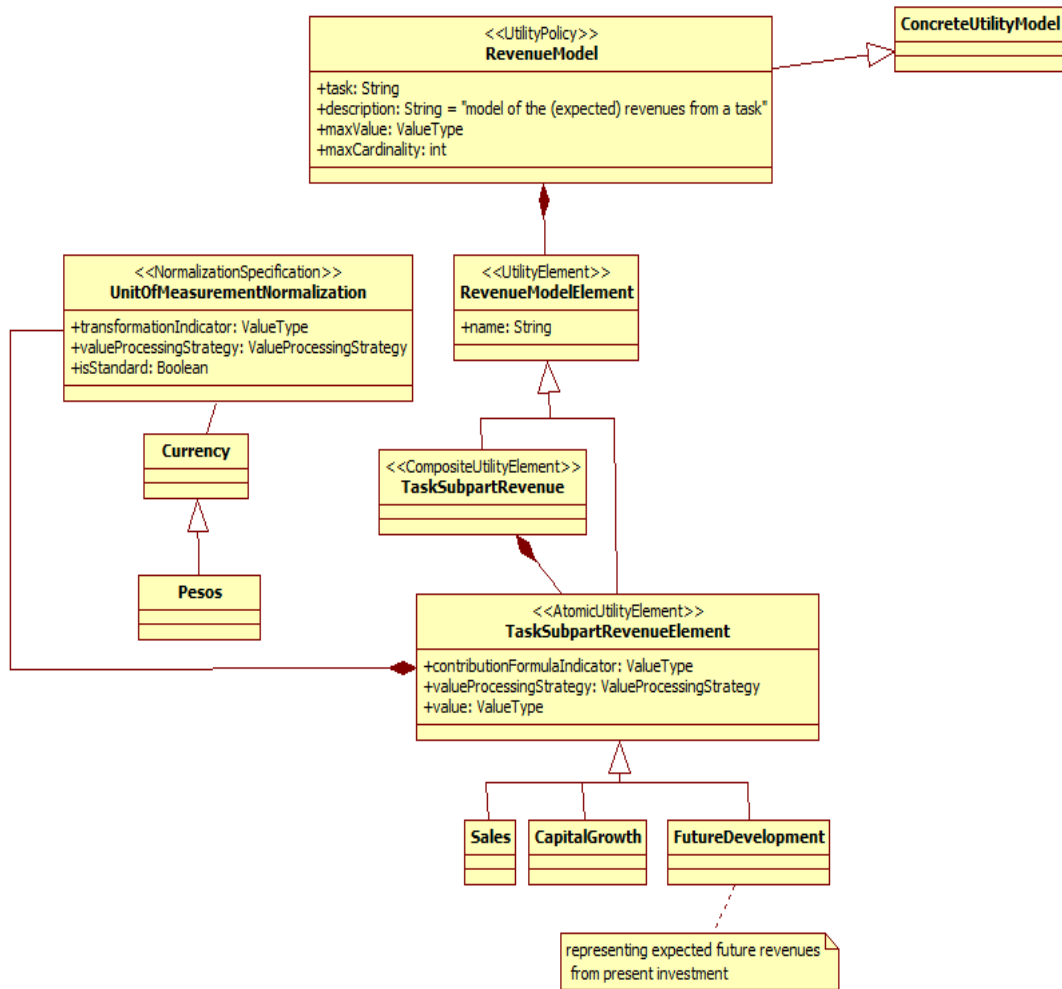
- CostModelElement : Είναι τύπου UtilityElement και αναφέρεται σε όλα τα στοιχεία του συγκεκριμένου μοντέλου, απλά (atomic elements) και σύνθετα (composite elements).
- FixedCost : Είναι τύπου CompositeUtilityElement και αναφέρεται στα πάγια έξοδα που εμπλέκονται σε μία επιχειρηματική διαδικασία. Ενδεικτικά ως FixedCostElement, δηλαδή απλό στοιχείο (AtomicUtilityElement) που απαρτίζει

αυτό το σύνθετο στοιχείο, έχει επιλεγθεί η ενοικίαση εγκαταστάσεων (FacilitiesRent).

- VariableCost : Είναι τύπου CompositeUtilityElement και αναφέρεται στα μεταβλητά έξοδα που εμπλέκονται σε μία επιχειρηματική διαδικασία. Ενδεικτικά ως VariableCostElements, δηλαδή απλά στοιχεία (AtomicUtilityElements) που απαρτίζουν αυτό το σύνθετο στοιχείο, έχουν επιλεγθεί ο μισθός εργαζομένου/ων (WorkerWage), το κόστος πρώτων υλών (RawMaterialCost), και το κόστος για τη χρήση λογισμικού (SoftwareCost).
- UnitOfMeasurementNormalization : Είναι τύπου NormalizationSpecification και παρέχει τα απαραίτητα δεδομένα για τη μετατροπή ενός κόστους στο κατάλληλο (πρότυπο) συνάλλαγμα. Για παράδειγμα, έχει επιλεγθεί ως νόμισμα (Currency) το Pesos. Αυτό σημαίνει ότι αν ένα απλό στοιχείο (AtomicUtilityElement) συνδέεται με το Pesos, τότε στο στοιχείο Pesos που είναι τύπου UnitOfMeasurementNormalization, περιέχονται οι πληροφορίες για τη μετατροπή της τιμής (value) του απλού στοιχείου από Pesos στο πρότυπο συνάλλαγμα.

4.2.2 Μοντέλο Εσόδων (Revenue Model)

Το παρακάτω διάγραμμα (Σχήμα 3.11) δίνει μία εικόνα της μορφής ενός μοντέλου εσόδων (Revenue Model), που περιγράφει τα (αναμενόμενα) οικονομικά κέρδη που προκύπτουν από τη διεκπεραίωση ενός Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task) :



Σχήμα 3.11: Μοντέλο Εσόδων (Revenue Model)

Ακολουθεί μια σύντομη περιγραφή για κάθε σημαντικό στοιχείο του παραπάνω διαγράμματος:

- RevenueModelElement : Είναι τύπου UtilityElement και αναφέρεται σε όλα τα στοιχεία του συγκεκριμένου μοντέλου, απλά (atomic elements) και σύνθετα (composite elements).
- TaskSubpartRevenue : Είναι τύπου CompositeUtilityElement και αναφέρεται στα κέρδη που προκύπτουν από ένα τμήμα του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task). Κάθε απλό στοιχείο που ανήκει σε αυτό το σύνθετο στοιχείο είναι τύπου AtomicUtilityElement και ενδεικτικά έχουν επιλεγθεί τα εξής στοιχεία για την αναπαράσταση του κέρδους :

→ Sales: Αναφέρεται στα αναμενόμενα κέρδη από πωλήσεις οι οποίες «πυροδοτούνται» από κάποιο τμήμα της διεξαγωγής του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task), είτε βραχυπρόθεσμα είτε μακροπρόθεσμα.

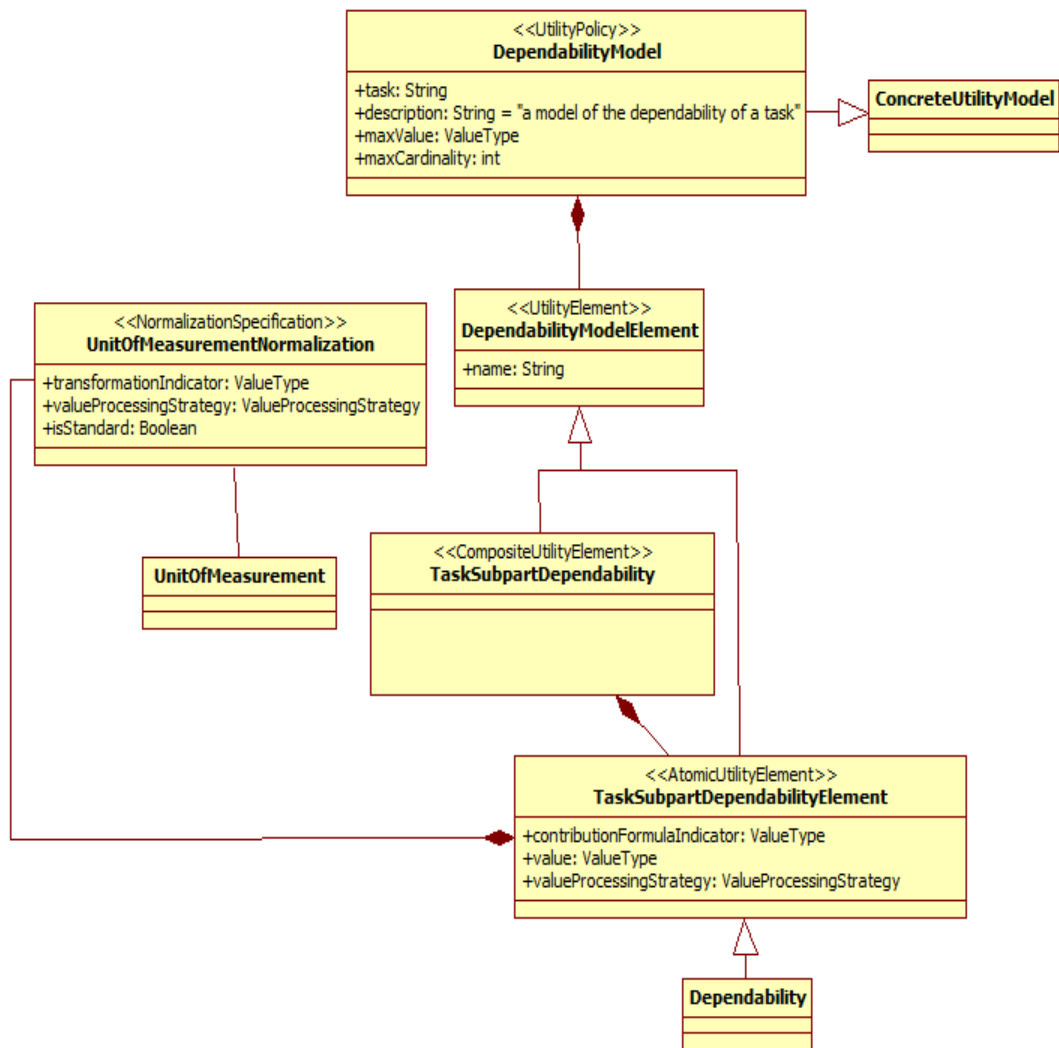
→ CapitalGrowth: Αντιπροσωπεύει την αναμενόμενη αύξηση κεφαλαίου που προκύπτει από κάποιο τμήμα της διεξαγωγής του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task), είτε βραχυπρόθεσμα είτε μακροπρόθεσμα.

→ FutureDevelopment: Αναφέρεται στα αναμενόμενα κέρδη που οφείλονται σε παύσα επενδυτική δραστηριότητα, θεωρώντας ως μία τέτοια ένα τμήμα της διεξαγωγής του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task).

- UnitOfMeasurementNormalization : Είναι τύπου NormalizationSpecification και παρέχει τα απαραίτητα δεδομένα για τη μετατροπή ενός κέρδους στο κατάλληλο (πρότυπο) συνάλλαγμα. Για παράδειγμα, έχει επιλεγθεί ως νόμισμα (Currency) το Pesos. Αυτό σημαίνει ότι αν ένα απλό στοιχείο (AtomicUtilityElement) συνδέεται με το Pesos, τότε στο στοιχείο Pesos που είναι τύπου UnitOfMeasurementNormalization, περιέχονται οι πληροφορίες για τη μετατροπή της τιμής (value) του απλού στοιχείου από Pesos στο πρότυπο συνάλλαγμα.

4.2.3 Μοντέλο Αξιοπιστίας (Dependability Model)

Το παρακάτω διάγραμμα (Σχήμα 3.12) δίνει μία εικόνα της μορφής ενός μοντέλου αξιοπιστίας (DependabilityModel), που περιγράφει την αξιοπιστία των εμπλεκόμενων τμημάτων στη διεξαγωγή ενός Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task):



Σχήμα 3.12: Μοντέλο Αξιοπιστίας (Dependability Model)

Ακολουθεί μια σύντομη περιγραφή για κάθε σημαντικό στοιχείο του παραπάνω διαγράμματος:

- DependabilityModelElement : Είναι τύπου UtilityElement και αναφέρεται σε όλα τα στοιχεία του συγκεκριμένου μοντέλου, απλά (atomic elements) και σύνθετα (composite elements).
- TaskSubpartDependability : Είναι τύπου CompositeUtilityElement και αναφέρεται στην αξιοπιστία ενός τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task). Κάθε απλό στοιχείο που ανήκει σε

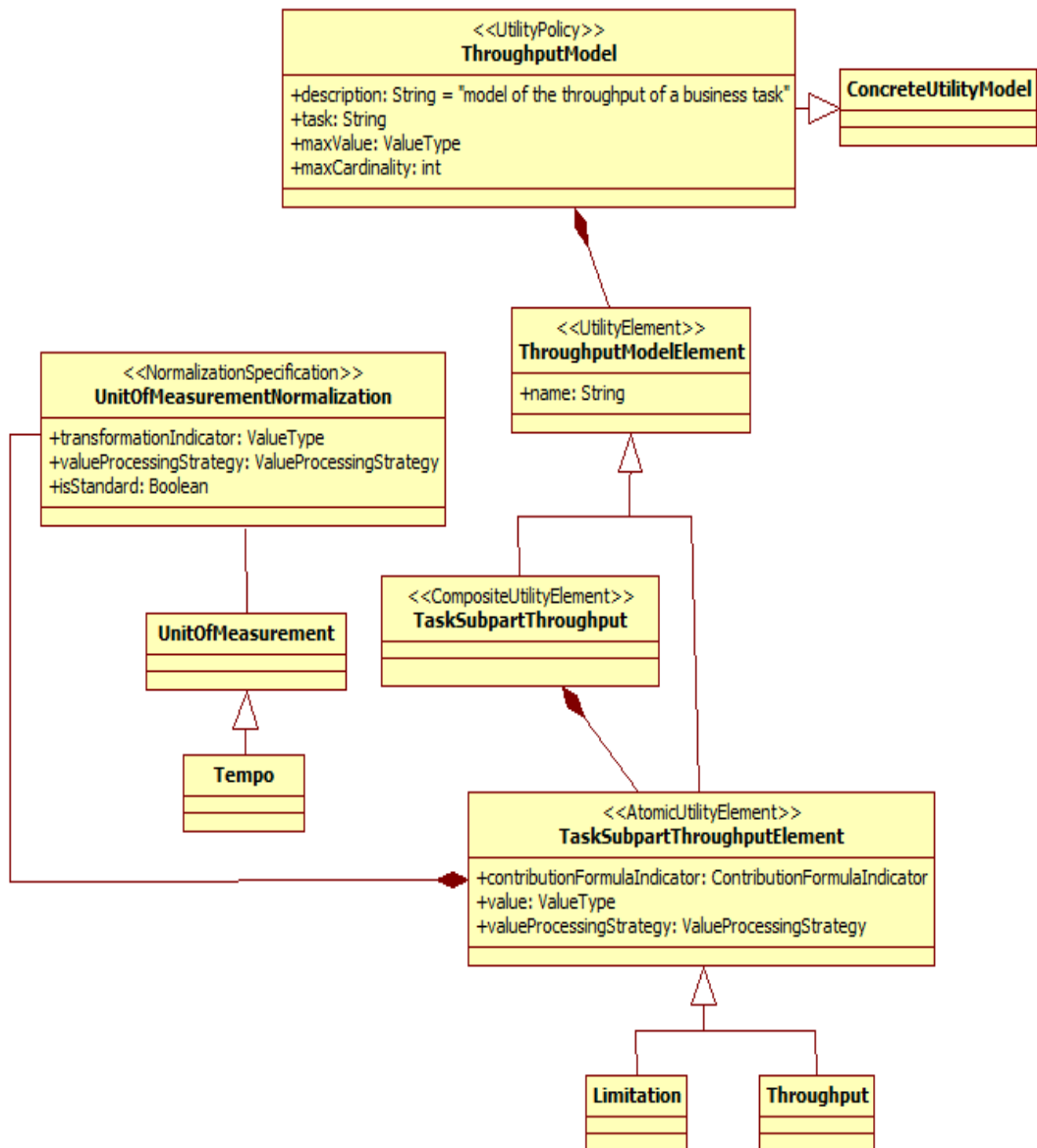
αυτό το σύνθετο στοιχείο είναι τύπου AtomicUtilityElement και είναι του εξής τύπου:

→ Dependability: Αναφέρεται στην αξιοπιστία ενός συγκεκριμένου τμήματος, η οποία γενικά ορίζεται ως η πιθανότητα ενός συστήματος να λειτουργήσει όπως σχεδιάστηκε στις προδιαγραφόμενες συνθήκες και για μια ορισμένη χρονική περίοδο λειτουργίας. Για το σύστημα, είναι μία πιθανότητα ή ένας δείκτης, που επηρεάζεται από ανθρώπινους και άλλους παράγοντες, και αντιπροσωπεύει την εμπιστοσύνη που δίδεται σε ένα τμήμα του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task) ότι θα ανταπεξέλθει στις εκάστοτε προσδοκίες.

- UnitOfMeasurementNormalization : Είναι τύπου NormalizationSpecification και παρέχει τα απαραίτητα δεδομένα για τη μετατροπή μιας αξιοπιστίας στην κατάλληλη (πρότυπη) μονάδα μέτρησης, που μπορεί να είναι κάποιος ορισμένος δείκτης.

4.2.4 Μοντέλο Ρυθμαπόδοσης (Throughput Model)

Το παρακάτω διάγραμμα (Σχήμα 3.13) δίνει μία εικόνα της μορφής ενός μοντέλου ρυθμαπόδοσης (ThroughputModel), που περιγράφει τη διεκπεραιωτική ικανότητα των τμημάτων ενός Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task). Γενικά, με τον όρο ρυθμαπόδοση στις επιχειρηματικές διαδικασίες εννοούμε το ρυθμό παραγωγής μιάς συγκεκριμένης διαδικασίας μετρούμενο στη βάση ορισμένου χρονικού διαστήματος. Η ρυθμαπόδοση μπορεί να μετρηθεί είτε με οικονομικούς όρους είτε όχι :



Σχήμα 3.13: Μοντέλο Ρυθμαπόδοσης (Throughput Model)

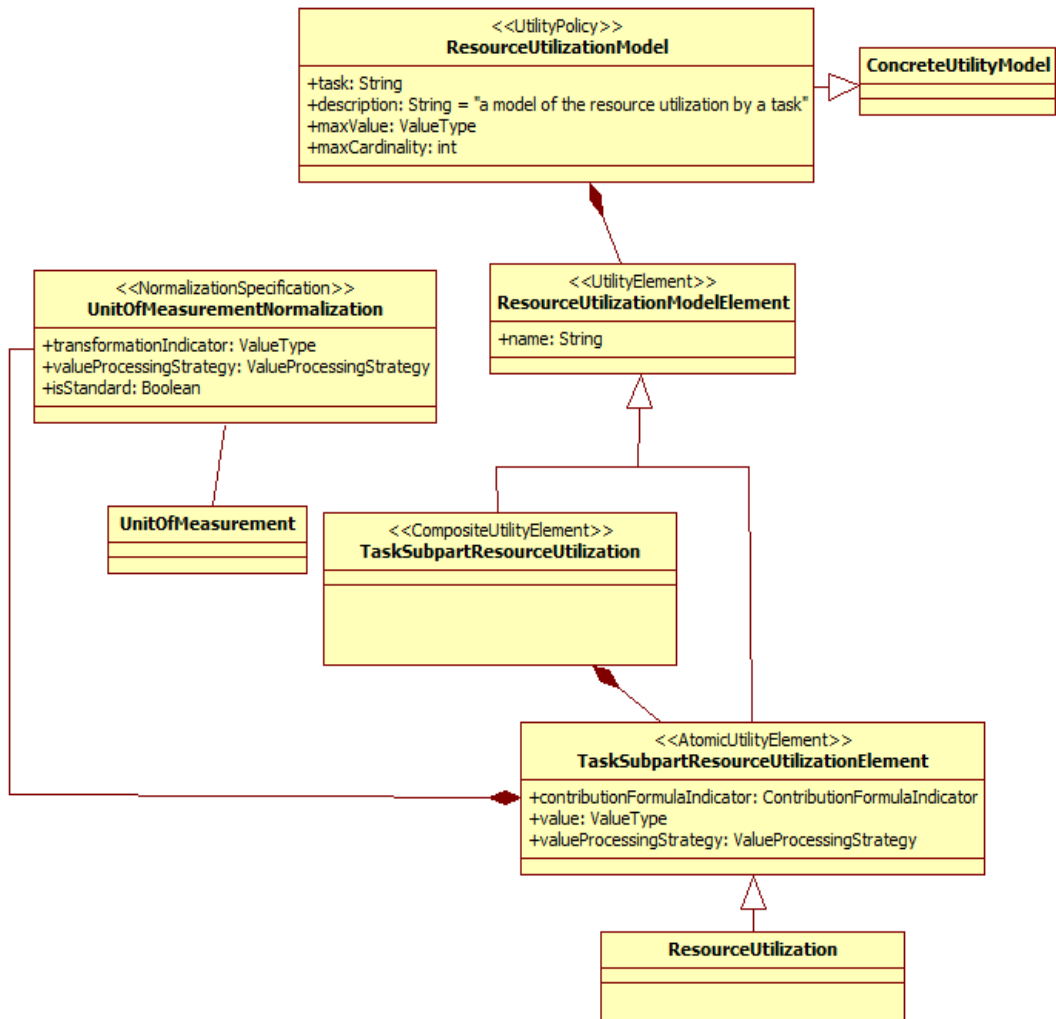
Ακολουθεί μια σύντομη περιγραφή για κάθε σημαντικό στοιχείο του παραπάνω διαγράμματος:

- ThroughputModelElement : Είναι τύπου UtilityElement και αναφέρεται σε όλα τα στοιχεία του συγκεκριμένου μοντέλου, απλά (atomic elements) και σύνθετα (composite elements).

- TaskSubpartThroughput : Είναι τύπου CompositeUtilityElement και αναφέρεται στη διεκπεραιωτική ικανότητα ενός τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task). Κάθε απλό στοιχείο που ανήκει σε αυτό το σύνθετο στοιχείο είναι τύπου AtomicUtilityElement και ενδεικτικά έχουν επιλεγθεί τα εξής στοιχεία για την αναπαράσταση του κέρδους :
 - Limitation: Αναφέρεται στους διάφορους περιορισμούς εγγενείς και μη που διέπουν κάποιο τμήμα της διεξαγωγής του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task), περιορίζοντας τη διεκπεραιωτική του ικανότητα.
 - Throughput: Αντιπροσωπεύει τη ρυθμαπόδοση κάποιου τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task), δηλαδή τη διεκπεραιωτική του ικανότητα, που ανάλογα με τη φύση της εκάστοτε επιχειρηματικής διαδικασίας (Business Process), αναφέρεται σε διαφορετικά μετρούμενα μεγέθη.
- UnitOfMeasurementNormalization : Είναι τύπου NormalizationSpecification και παρέχει τα απαραίτητα δεδομένα για τη μετατροπή μιας ρυθμαπόδοσης στην κατάλληλη (πρότυπη) μονάδα μέτρησης, που μπορεί να είναι κάποιος ορισμένος δείκτης ή μία μονάδα μέτρησης πλήθους διεξαχθέντων διεργασιών ανά μονάδα χρόνου. Για παράδειγμα, αν η επιχειρηματική διαδικασία αφορά σε services, το throughput μπορεί να μετράται σε (complete_requests / unit_time).

4.2.5 Μοντέλο Χρησιμοποίησης Πόρων (Resource Utilization Model)

Το παρακάτω διάγραμμα (Σχήμα 3.14) δίνει μία εικόνα της μορφής ενός μοντέλου χρησιμοποίησης πόρων (ResourceUtilizationModel), που περιγράφει τους πόρους που απαιτείται να χρησιμοποιηθούν/σπαταληθούν για τη διεκπεραίωση ενός Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task) :



Σχήμα 3.14: Μοντέλο Χρησιμοποίησης Πόρων (Resource Utilization Model)

Ακολουθεί μια σύντομη περιγραφή για κάθε σημαντικό στοιχείο του παραπάνω διαγράμματος:

- ResourceUtilizationModelElement : Είναι τύπου UtilityElement και αναφέρεται σε όλα τα στοιχεία του συγκεκριμένου μοντέλου, απλά (atomic elements) και σύνθετα (composite elements).
- TaskSubpartResourceUtilization : Είναι τύπου CompositeUtilityElement και αναφέρεται στην χρησιμοποίηση πόρων ενός τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task). Κάθε απλό στοιχείο που ανήκει σε αυτό το σύνθετο στοιχείο είναι τύπου AtomicUtilityElement.

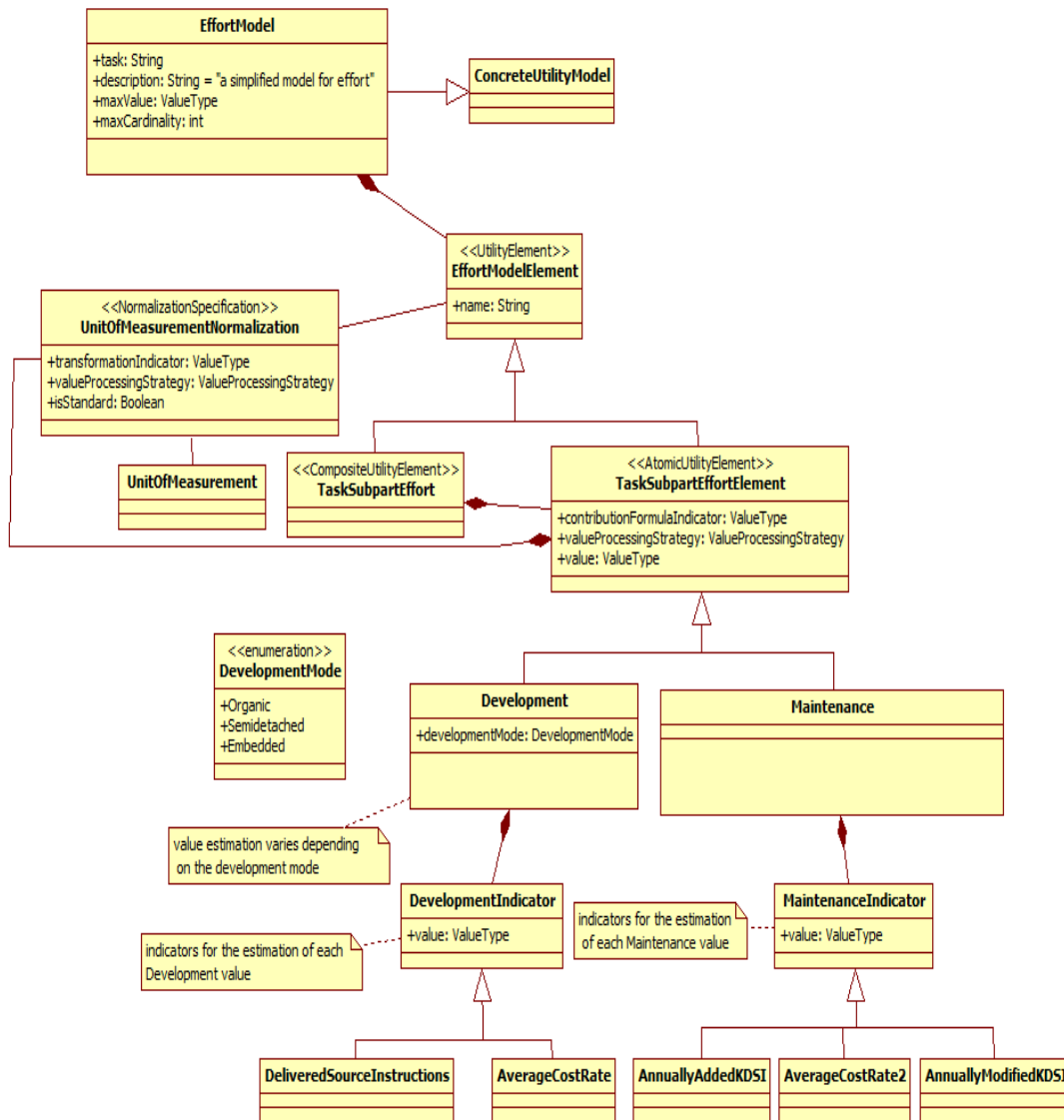
Να σημειωθεί ότι η χρησιμοποίηση πόρων αναφέρεται σε ένα υψηλό αφαιρετικό επίπεδο για τη χρησιμοποίηση πόρων γενικά και όχι για κάθε πόρο ξεχωριστά. Κάθε απλό λοιπόν στοιχείο θα έχει μία τιμή που θα αντιπροσωπεύει ένα δείκτη χρησιμοποίησης πόρων από ένα συγκεκριμένο τμήμα του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας.

- UnitOfMeasurementNormalization : Είναι τύπου NormalizationSpecification και παρέχει τα απαραίτητα δεδομένα για τη μετατροπή μιας χρησιμοποίησης πόρων στην κατάλληλη (πρότυπη) μονάδα μέτρησης, που μπορεί να είναι κάποιος ορισμένος δείκτης.

4.2.6 Μοντέλο Προσπάθειας (Effort Model)

Μέχρι στιγμής παρουσιάστηκαν μοντέλα στα οποία, με κάποιο τρόπο, η τιμή value για κάθε atomic utility element θεωρείται υπολογισμένη. Τα υπόλοιπα μοντέλα, συμπεριλαμβανομένου του παρόντος, περιέχουν ως στοιχεία τους και κάποιους δείκτες, με βάση τους οποίους πραγματοποιείται κάθε φορά ο υπολογισμός της τιμής value για καθένα εκ των atomic utility elements, ανάλογα με το εκάστοτε μοντέλο και την εκάστοτε στρατηγική υπολογισμού.

Το παρακάτω διάγραμμα (Σχήμα 3.15) δίνει μία εικόνα της μορφής ενός μοντέλου προσπάθειας (EffortModel), που περιγράφει την προσπάθεια που καταβάλλεται για τη διεκπεραίωση ενός Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task). Οι επιχειρηματικές διαδικασίες στις οποίες αναφέρεται το παρακάτω μοντέλο είναι από το χώρο της Τεχνολογίας Λογισμικού και το μοντέλο είναι βασισμένο στο Basic COCOMO [38], προορισμένο για μία bottom-up εκτίμηση .



Σχήμα 3.15: Μοντέλο Προσπάθειας (Effort Model)

Ακολουθεί μια σύντομη περιγραφή για κάθε σημαντικό στοιχείο του παραπάνω διαγράμματος:

- EffortModelElement : Είναι τύπου UtilityElement και αναφέρεται σε όλα τα στοιχεία του συγκεκριμένου μοντέλου, απλά (atomic elements) και σύνθετα (composite elements).

- TaskSubpartEffort : Είναι τύπου CompositeUtilityElement και αναφέρεται στην προσπάθεια που απαιτείται από ένα τμήμα του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task). Κάθε απλό στοιχείο που ανήκει σε αυτό το σύνθετο στοιχείο είναι τύπου AtomicUtilityElement και ενδεικτικά έχουν επιλεγθεί τα εξής στοιχεία για την αναπαράσταση της προσπάθειας:
 - Development: Αναφέρεται στην προσπάθεια που καταβάλλεται κατά το αναπτυξιακό κομμάτι κάποιου τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task).
 - Maintenance: Αναφέρεται στην προσπάθεια που καταβάλλεται κατά το κομμάτι της συντήρησης κάποιου τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task).

Ο υπολογισμός της προσπάθειας διαφέρει ανάλογα με τη λειτουργία ανάπτυξης (DevelopmentMode), η οποία αναπαρίσταται ως μία απαρίθμηση (enumeration) στο μοντέλο και απαρτίζεται από τα εξής στοιχεία:

- Organic (Οργανική κατηγορία): Μικρές ομάδες, γνωστό αντικείμενο, απλές απαιτήσεις
- Semi-detached (Ημι-χωρισμένη κατηγορία): Η ομάδα και ο οργανισμός που αναλαμβάνει το έργο έχουν σχετική εμπειρία αλλά όχι μεγάλη εμπειρία στο πεδίο εφαρμογής και στην υλοποίηση των προδιαγραφών του συστήματος
- Embed (Ενσωματωμένα): Η ομάδα έχει περιορισμένη εμπειρία, στις συγκεκριμένες εφαρμογές-συστήματα, και οι προδιαγραφές είναι πολύπλοκες

Στο μοντέλο αυτό υπάρχει επίσης η απαραίτητη πληροφορία υπό τη μορφή δεικτών (indicators) για τον υπολογισμό των τιμών value σε καθένα από τα στοιχεία Development και Maintenance.

Για τα στοιχεία τύπου Development έχουμε:

- DevelopmentIndicator : Είναι τύπου Indicator και υποκλάσεις του αποτελούν οι:
 - DeliveredSourceInstructions: Αναφέρεται στο πλήθος των πηγαίων εντολών που απαιτείται να παραδοθούν.
 - AverageCostRate: Αναφέρεται στο μέσο κόστος ανά χρονική μονάδα -συνήθως εργατομήνες (person months).

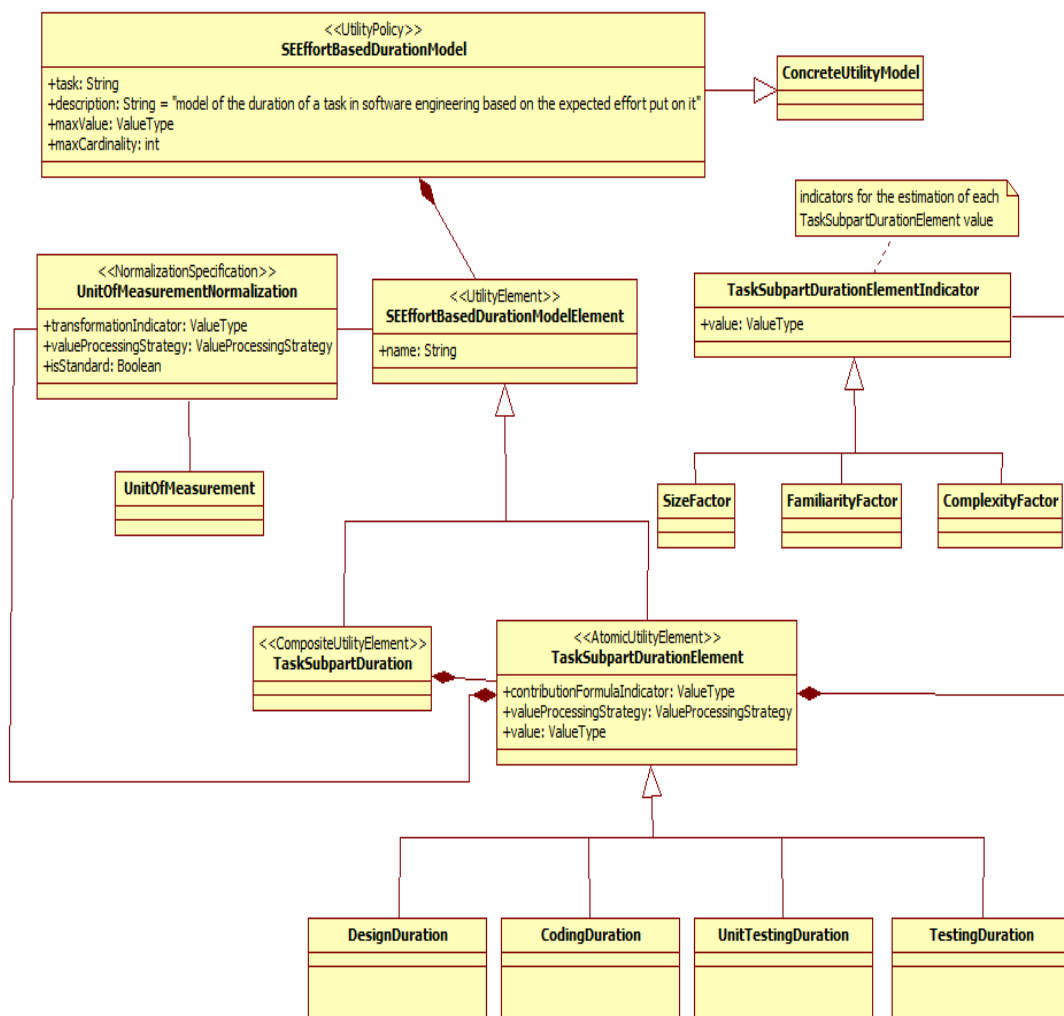
Για τα στοιχεία τύπου Maintenance έχουμε:

- MaintenanceIndicator : Είναι τύπου Indicator και υποκλάσεις του αποτελούν οι:
 - AnnuallyAddedKDSI: Αναφέρεται στο πλήθος των πηγαίων εντολών σε χιλιάδες, που παραδίδονται ετησίως.
 - AverageCostRate: Αναφέρεται στο μέσο κόστος ανά χρονική μονάδα -συνήθως εργατομήνες (person months).
 - AnnuallyModifiedKDSI: Αναφέρεται στο πλήθος των πηγαίων εντολών σε χιλιάδες, που τροποποιούνται ετησίως.

- UnitOfMeasurementNormalization : Είναι τύπου NormalizationSpecification και παρέχει τα απαραίτητα δεδομένα για τη μετατροπή μιας προσπάθειας στην κατάλληλη (πρότυπη) μονάδα μέτρησης, που μπορεί να είναι κάποιος ορισμένος δείκτης.

4.2.7 Μοντέλο Διαρκείας βασισμένο στην Προσπάθεια (Effort-Based Duration Model)

Το παρακάτω διάγραμμα (Σχήμα 3.16) δίνει μία εικόνα της μορφής ενός μοντέλου διαρκείας βασισμένου στην προσπάθεια (EffortBasedDurationModel), που περιγράφει την αναμενόμενη διάρκεια ενός Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task)). Οι επιχειρηματικές διαδικασίες στις οποίες αναφέρεται το παρακάτω μοντέλο είναι από το χώρο της Τεχνολογίας Λογισμικού (Software Engineering).



Σχήμα 3.16: Μοντέλο Διαρκείας βασισμένο στην Προσπάθεια (Effort-Based Duration Model)

Ακολουθεί μια σύντομη περιγραφή για κάθε σημαντικό στοιχείο του παραπάνω διαγράμματος:

- SEffortBasedDurationModelElement : Είναι τύπου `UtilityElement` και αναφέρεται σε όλα τα στοιχεία του συγκεκριμένου μοντέλου, απλά (atomic elements) και σύνθετα (composite elements).
- TaskSubpartDuration : Είναι τύπου `CompositeUtilityElement` και αναφέρεται στην προσπάθεια που απαιτείται από ένα τμήμα του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task). Κάθε απλό στοιχείο που ανήκει σε αυτό το σύνθετο στοιχείο είναι τύπου `AtomicUtilityElement` και ενδεικτικά έχουν επιλεχθεί τα εξής στοιχεία για την αναπαράσταση της διάρκειας:
 - `DesignDuration`: Αναφέρεται στη διάρκεια της σχεδίασης κάποιου τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task).
 - `CodingDuration`: Αναφέρεται στη διάρκεια συγγραφής κώδικα κάποιου τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task).
 - `UnitTestingDuration`: Αναφέρεται στη διάρκεια δοκιμών των δομικών μονάδων (unit testing) κάποιου τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task).
 - `TestingDuration`: Αναφέρεται στη διάρκεια δοκιμών κάποιου (συνολικού) τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task).

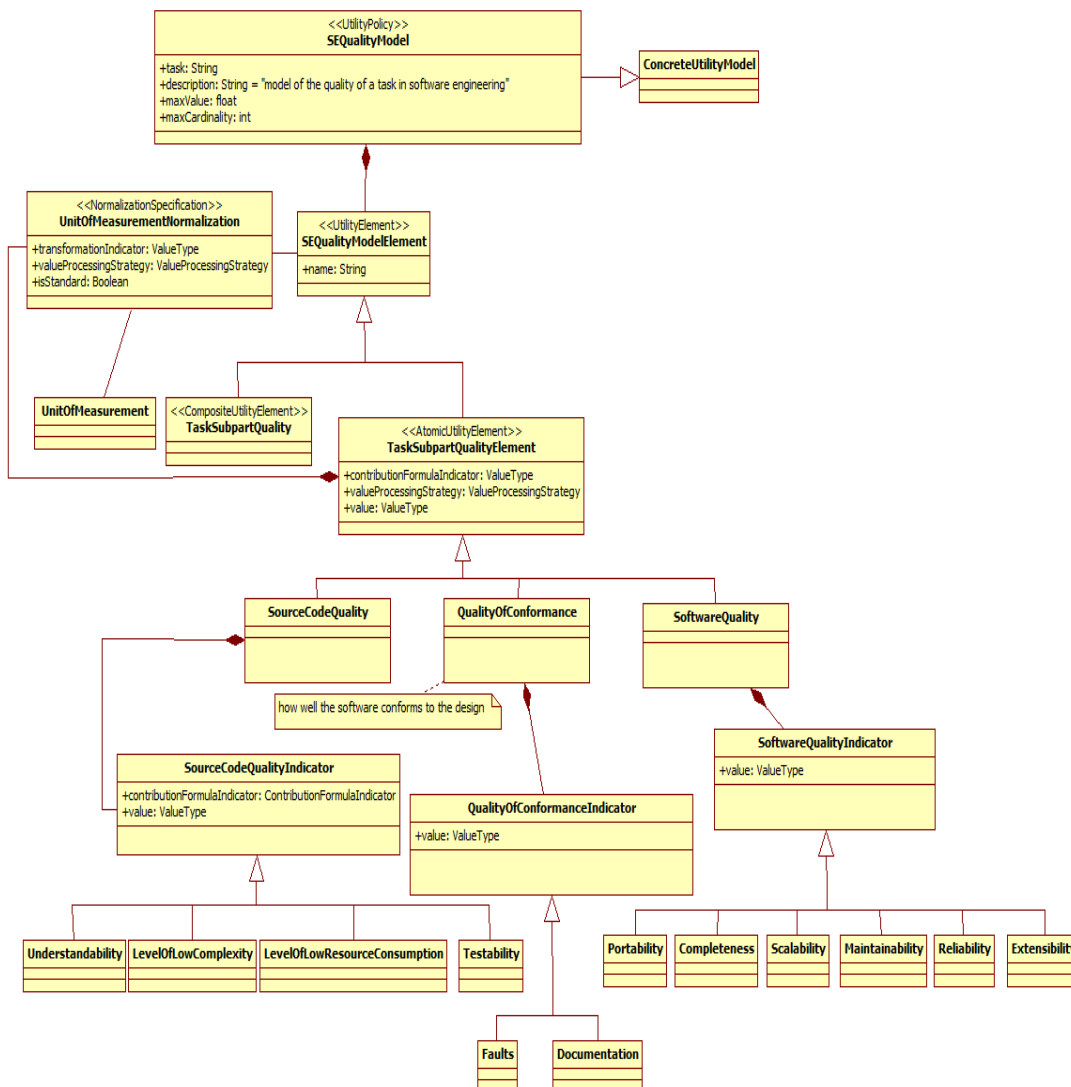
Στο μοντέλο αυτό υπάρχει επίσης η απαραίτητη πληροφορία υπό τη μορφή δεικτών (indicators) για τον υπολογισμό των τιμών value σε καθένα από τα παραπάνω στοιχεία. Σε καθένα από αυτά υπάρχουν συσχετίσεις με στοιχεία τύπου `Indicator`.

- TaskSubpartDurationElementIndicator : Είναι τύπου `Indicator` και υποκλάσεις του αποτελούν οι:
 - `SizeFactor`: Παράγοντας που αναφέρεται στο μέγεθος κάποιου τμήματος.
 - `FamiliarityFactor`: Παράγοντας που αναφέρεται στην εξοικείωση (προφανώς της τεχνικής ομάδας που ασχολείται με τη διαδικασία) με το συγκεκριμένο τμήμα του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task).
 - `ComplexityFactor`: Παράγοντας που αναφέρεται στην πολυπλοκότητα κάποιου τμήματος.

- **UnitOfMeasurementNormalization** : Είναι τύπου NormalizationSpecification και παρέχει τα απαραίτητα δεδομένα για τη μετατροπή μιας διάρκειας στην κατάλληλη (πρότυπη) μονάδα μέτρησης, π.χ ώρες, μέρες, εβδομάδες ή και μήνες.

4.2.8 Μοντέλο Ποιότητας (Quality Model)

Το παρακάτω διάγραμμα (Σχήμα 3.17) δίνει μία εικόνα της μορφής ενός μοντέλου ποιότητας (QualityModel), που περιγράφει την ποιότητα ενός Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task) . Οι επιχειρηματικές διαδικασίες στις οποίες αναφέρεται το παρακάτω μοντέλο είναι από το χώρο της Τεχνολογίας Λογισμικού (Software Engineering).



Σχήμα 3.17: Μοντέλο Ποιότητας (Quality Model)

Ακολουθεί μια σύντομη περιγραφή για κάθε σημαντικό στοιχείο του παραπάνω διαγράμματος:

- SEQualityModelElement : Είναι τύπου UtilityElement και αναφέρεται σε όλα τα στοιχεία του συγκεκριμένου μοντέλου, απλά (atomic elements) και σύνθετα (composite elements).
- TaskSubpartQuality : Είναι τύπου CompositeUtilityElement και αναφέρεται στην ποιότητα ενός τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task). Κάθε απλό στοιχείο που ανήκει σε αυτό το σύνθετο στοιχείο είναι τύπου AtomicUtilityElement και ενδεικτικά έχουν επιλεγθεί τα εξής στοιχεία για την αναπαράσταση της ποιότητας:
 - SourceCodeQuality: Αναφέρεται στην ποιότητα του πηγαίου κώδικα του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task).
 - QualityOfConformance: Αναφέρεται στην ποιότητα της συμμόρφωσης κάποιου τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task) στο σχεδιασμό.
 - SoftwareQuality: Αναφέρεται στην ποιότητα του λογισμικού κάποιου τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task).

Στο μοντέλο αυτό υπάρχει επίσης η απαραίτητη πληροφορία υπό τη μορφή δεικτών (indicators) για τον υπολογισμό των τιμών value σε καθένα από τα παραπάνω στοιχεία.

Για τα στοιχεία τύπου SourceCodeQuality έχουμε:

- SourceCodeQualityIndicator : Είναι τύπου Indicator και υποκλάσεις του αποτελούν οι:
 - Understandability: Η ευκολία κατανόησης του πηγαίου κώδικα.
 - LevelOfLowComplexity: Το επίπεδο χαμηλής πολυπλοκότητας του πηγαίου κώδικα.
 - LevelOfLowResourceConsumption: Το επίπεδο χαμηλής κατανάλωσης πόρων.
 - .
 - Testability: Η ευκολία διεξαγωγής δοκιμών (δοκιμαστικότητα) του πηγαίου κώδικα.

Για τα στοιχεία τύπου QualityOfConformance έχουμε:

- QualityOfConformanceIndicator : Είναι τύπου Indicator και υποκλάσεις του αποτελούν οι:
 - Faults: Τα σφάλματα (αστοχίες) κάποιου τμήματος του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task).

→ Documentation: Αναφέρεται στην ύπαρξη επαρκούς τεκμηρίωσης (documentation) του συγκεκριμένου Απλού Έργου Επιχειρηματικής Διαδικασίας (Atomic Business Process Task).

Για τα στοιχεία τύπου SoftwareQuality έχουμε:

- SoftwareQualityIndicator : Είναι τύπου Indicator και υποκλάσεις του αποτελούν οι:
 - Portability: Η φορητότητα του συγκεκριμένου λογισμικού.
 - Completeness: Η πληρότητα του συγκεκριμένου λογισμικού.
 - Scalability: Η δυνατότητα κλιμάκωσης του συγκεκριμένου λογισμικού.
 - Maintainability: Η δυνατότητα συντήρησης (συντηρησιμότητα) του συγκεκριμένου λογισμικού.
 - Reliability: Η αξιοπιστία του συγκεκριμένου λογισμικού.
 - Extensibility: Η επεκτασιμότητα του συγκεκριμένου λογισμικού.
- UnitOfMeasurementNormalization : Είναι τύπου NormalizationSpecification και παρέχει τα απαραίτητα δεδομένα για τη μετατροπή μιας ποιότητας στην κατάλληλη (πρότυπη) μονάδα μέτρησης, που μπορεί να είναι κάποιος ορισμένος δείκτης.

5

Αποτίμηση

Επιχειρηματικών

Διαδικασιών (Business

Process Evaluation)

Στο κεφάλαιο αυτό γίνεται η περιγραφή του τρόπου αποτίμησης των επιχειρηματικών διαδικασιών από το σύστημα. Αναλύονται (και σε μορφή ψευδοκώδικα) οι βασικοί αλγόριθμοι που χρησιμοποιούνται για τις εξής διαδικασίες :

- **Ανάλυση διαδρομών (Path Analysis)** : Εύρεση όλων των δυνατών διαδρομών που μπορούν να ακολουθηθούν για τη διεκπεραίωση μιας συγκεκριμένης επιχειρηματικής διαδικασίας.
- **Στρατηγική Αποτίμησης (Evaluation Strategy)** : Επιλογή στρατηγικής και χρησιμοποίησή της για την (ποσοτική) αποτίμηση τόσο της καθεμιάς εναλλακτικής διαδρομής, όπως αυτές έχουν προκύψει από εφαρμογή του παραπάνω αλγορίθμου, όσο και των συστατικών στοιχείων αυτών των διαδρομών.
- **Σύνθεση της Στρατηγικής Αποτίμησης (Composition of Evaluation Strategy)** : Χρήση των αποτελεσμάτων από την εφαρμογή του αμέσως παραπάνω αλγορίθμου, για την εξαγωγή συνολικών αποτελεσμάτων για την όλη επιχειρηματική διαδικασία, βασισμένων σε συγκεκριμένες επιλεχθείσες στρατηγικές και κριτήρια.

5.1 Ανάλυση διαδρομών (Path Analysis)

Στο παρελθόν έχουν χρησιμοποιηθεί πλήθος τεχνικών για την αναπαράσταση των επιχειρηματικών διαδικασιών (business processes), όπως διαγράμματα ροής, διαγράμματα ροής ελέγχου, γραφήματα Gantt κ.α.. Κάποιες από τις σύγχρονες μεθόδους μοντελοποίησης των επιχειρηματικών διαδικασιών αποτελούν οι γραφικές αναπαραστάσεις, όπως η BPMN [24] του OMG, τα διαγράμματα δραστηριότητας (activity diagrams) της UML [5] και το Microsoft VISIO, και οι γλώσσες προγραμματισμού, όπως η BPEL [25] [26] του OASIS , ARIS κ.α..

Το ενδιαφέρον της παρούσης εργασίας εστιάζεται σε διαδικασίες που αποτελούνται από (παράλληλες ή ακολουθιακές) υποδιαδικασίες (subprocesses), οι οποίες ενδεχομένως αποτελούνται από άλλες υποδιαδικασίες κ.ο.κ., παρέχοντας επιπλέον τη δυνατότητα επιλογής διαφορετικών εναλλακτικών διαδρομών που μπορούν να ακολουθηθούν για τη διεκπεραίωσή τους. Τέτοιες διαδικασίες μπορούν να αναπαρασταθούν με τη μορφή ενός ειδικού Ιεραρχικού Κατευθυνόμενου Ακυκλικού Γράφου (Hierarchical Directed Acyclic Graph) [43], ο οποίος στην περίπτωση μας δεν είναι παρά ένας κατευθυνόμενος ακυκλικός γράφος, του οποίου οι κόμβοι αντιπροσωπεύουν σύνθετες διαδικασίες. Κάθε κόμβος, δηλαδή, αυτού του γράφου είναι είτε μία απλή υποδιαδικασία (Atomic Task) είτε ένας εκ νέου σύνθετος κατευθυνόμενος ακυκλικός γράφος. Η διαδικασία της ανάλυσης διαδρομών, λοιπόν, έγκειται στο μετασχηματισμό της εκάστοτε διαδικασίας, από τη μορφή με την οποία περιγράφεται (π.χ. BPEL ή BPMN) στην παραπάνω μορφή (HDAG) και στη συνέχεια η αποσύνθεση του σχηματισθέντος γράφου σε ντετερμινιστικά ακολουθούμενες διαδρομές (Paths).

5.1.1 Εισαγωγικό Μέρος

Έστω T_0, T_1, \dots, T_N ένα σύνολο από διαδικασίες (Tasks) χωρίς κάποια ιδιαίτερη ταξινόμηση. Υποθέτουμε μια συλλογή από στοιχεία του παραπάνω συνόλου

$C = (T_{s(0)}, T_{s(1)}, T_{s(n)})$, όπου το $T_{s(i)}$ αντιπροσωπεύει την i -στή διαδικασία. Μπορούμε να υποδηλώσουμε την i -στή διαδικασία με το $C[i] = T_{s(i)}$.

ΟΡΙΣΜΟΙ

- CT : Composite Task = C_i είναι μία σύνθετη υποδιαδικασία, δηλαδή μια συλλογή από διαδικασίες. Μπορούμε να γράψουμε:

$$CT[j] = C_i[j].$$

Μία CT μπορεί να αποτελείται είτε από παράλληλες είτε από ακολουθιακές διαδικασίες, οπότε και $type(CT) = flow$ ή $type(CT) = sequence$, αντίστοιχα. Η προκαθορισμένη (default) τιμή για κάθε νέα διαδικασία είναι $type(CT) = sequence$.

- AT : Atomic Task είναι μία απλή υποδιαδικασία, δηλαδή μία αυτοτελής υποδιαδικασία που δεν μπορεί να διασπαστεί σε επιμέρους υποδιαδικασίες.
- T : Task είναι μία διαδικασία. Ισχύουν οι εξής σχέσεις:

Έστω $T_u = \{T_0, T_1, \dots, T_N\}$ το σύνολο όλων των διαδικασιών (Tasks). Έστω επίσης $C_u = \{CT_0, CT_1, \dots, CT_N\}$ το σύνολο όλων των σύνθετων διαδικασιών (Composite Tasks) και $A_u = \{AT_0, AT_1, \dots, AT_N\}$ το σύνολο όλων των απλών διαδικασιών (Atomic Tasks).

Τότε :

$$T_u = C_u \cup A_u .$$

Με άλλα λόγια, μία διαδικασία είναι είτε μία απλή διαδικασία είτε μία σύνθετη διαδικασία.

Από τα παραπάνω προκύπτει και η επεξήγηση του αναδρομικού ορισμού της σύνθετης υποδιαδικασίας (Composite Task), ο οποίος αναφέρθηκε παραπάνω.

- sp : starting point μιας σύνθετης υποδιαδικασίας (Composite Task) είναι μία διαδικασία (T) με την ιδιότητα καμία διαδικασία (Task) να μην προηγείται αυτής σε όποια σύνθετη διαδικασία και αν παίρνει μέρος, δηλαδή :
αν $sp_n = CT_n[i]$, τότε $\forall T_{s(j)}$, όπου $T_{s(j)} = CT_n[j]$, ισχύει $i < j$.
- psp : path starting point είναι μία διαδικασία (T) που αποτελεί starting point (sp) για μια υποδιαδικασία και επιπρόσθετα, η υποδιαδικασία αυτή στην οποία παίρνει μέρος είναι και αυτή path starting point (psp) , δηλαδή :
αν $psp_n = sp_n = CT_n[i]$ και $CT_n = CT_m[j]$, τότε $\forall T_{s(k)}$, όπου $T_{s(k)} = CT_m[k]$, ισχύει $j < k$ και $CT_n = psp_m$
- fp : finishing point μιας σύνθετης υποδιαδικασίας (Composite Task) είναι μία διαδικασία (T) με την ιδιότητα καμία διαδικασία να μην έπεται αυτής στη συγκεκριμένη σύνθετη διαδικασία. Μπορούμε να γράψουμε $fp_n = T_j = fp(CT_n)$. Τότε :
αν $fp_n = CT_n[i]$, τότε $\forall T_{s(j)}$, όπου $T_{s(j)} = CT_n[j]$, ισχύει $i > j$.
- pfp : path finishing point είναι μία απλή υποδιαδικασία (AT) που αποτελεί starting point (sp) για μια υποδιαδικασία και επιπρόσθετα, η υποδιαδικασία αυτή στην οποία παίρνει μέρος είναι και αυτή path finishing point (pfp) , δηλαδή :
αν $pfp_n = fp_n = CT_n[i]$ και $CT_n = CT_m[j]$, τότε $\forall T_{s(k)}$, όπου $T_{s(k)} = CT_m[k]$, ισχύει $j < k$ και $CT_n = pfp_m$
- P : Path είναι ένα μονοπάτι. Ένα μονοπάτι είναι μία σύνθετη υποδιαδικασία (CT), η οποία περιέχει τουλάχιστον ένα path starting point (psp) και τουλάχιστον ένα path finishing point (pfp).
- BP: Business Process είναι μία συλλογή από μονοπάτια (P).

Είναι λοιπόν φανερό ότι το BP (Business Process) όπως ορίστηκε παραπάνω είναι η επιθυμητή μορφή στην οποία θέλουμε να μετασχηματίσουμε τον εκάστοτε Ιεραρχικό

Κατευθυνόμενο Ακυκλικό Γράφο (Hierarchical Directed Acyclic Graph) και ένας αλγόριθμος για τη μετατροπή αυτή θα παρουσιαστεί στη συνέχεια. Πρώτα όμως θα παρουσιαστεί ένας γενικός αλγόριθμος για το μετασχηματισμό μιας διαδικασίας, από τη (γραφική ή μη) μορφή με την οποία περιγράφεται (π.χ. BPEL ή BPMN) στη μορφή HDAG.

Υποθέτουμε ότι στη μορφή με την οποία περιγράφεται η επιχειρηματική διαδικασία, είτε αυτή είναι γραφική, είτε προγραμματιστική:

- είναι πεπερασμένη (αποτελείται από πεπερασμένο πλήθος στοιχείων),
- υπάρχει ακριβώς ένα ολικό σημείο εκκίνησης για όλες τις ακολουθίες διαδικασιών (ξεκινώντας από ένα σημείο εκκίνησης μπορούμε να «επισκεφτούμε» όλες τις εμπλεκόμενες υποδιαδικασίες),
- υπάρχει μία συγκεκριμένη κατεύθυνση (ροή) από κάθε εμπλεκόμενη υποδιαδικασία σε κάποια άλλη και
- για χάρη απλότητας υποθέτουμε επιπροσθέτως ότι δεν περιέχει κύκλους (μετάβαση κατά την εκτέλεσή της διαδικασίας σε υποδιαδικασία που έχει ήδη ενεργοποιηθεί).
- επιπλέον, θεωρούμε ότι οι διαδικασίες δεν περιέχουν βρόχους (loops), αν και με κάποιες μετατροπές θα μπορούσε ο αλγόριθμος αυτός να εφαρμοστεί και σε διαδικασίες με βρόχους, τουλάχιστον τέτοιους ώστε το πλήθος επαναλήψεων να είναι εξ' αρχής προκαθορισμένο.

Έστω BPMF (Business Process Modeling Form) μία τέτοια μορφή αναπαράστασης επιχειρηματικών διαδικασιών και BPMF(bp_i) η αναπαράσταση σε αυτή τη μορφή μιας συγκεκριμένης επιχειρηματικής διαδικασίας bp_i.

Ένας Ιεραρχικός Κατευθυνόμενος Ακυκλικός Γράφος (Hierarchical Directed Acyclic Graph) είναι ένας κατευθυνόμενο ακυκλικό γράφο DAG : $G = (V,E)$ - ένας γράφος που σχηματίζεται από μια συλλογή από κόμβους V και κατευθυνόμενες ακμές E, με κάθε ακμή να ενώνει ένα κόμβο με κάποιον άλλο, έτσι ώστε να μην υπάρχει τρόπος εκκινώντας από κάποιον κόμβο v και ακολουθώντας μιά ακολουθία από ακμές να επιστρέψουμε πάλι πίσω στο v - με την επιπλέον γενίκευση ότι κάθε κόμβος v_i αποτελεί και ο ίδιος έναν HDAG. Επιπρόσθετα, κάθε κόμβος ενδέχεται να χαρακτηρίζεται από τον τύπο του (type) ως flowSource(id) ή flowTermination(id) για να συμβολίσουμε ότι η αντίστοιχη υποδιαδικασία συνδέεται με παράλληλες υποδιαδικασίες ή ότι μετά από αυτήν ακολουθεί ένδειξη τερματισμού παράλληλων διαδικασιών , αντίστοιχα.

Ακολουθεί σε μορφή ψευδοκώδικα ο αλγόριθμος μετασχηματισμού μίας μορφής αναπαράστασης επιχειρηματικών διαδικασιών στη μορφή ενός HDAG (Αλγόριθμος 5.1.1):

Αλγόριθμος 5.1.1 : Μετασχηματισμός Σε HDAG

Είσοδος: BPMF(bp_i) : η αναπαράσταση στη μορφή BPMF της επιχειρηματικής διαδικασίας bp_i

Έξοδος: HDAG(bp_i) : η αναπαράσταση στη μορφή HDAG της επιχειρηματικής διαδικασίας bp_i .

Ψευδοκώδικας:

ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ (BPMF(bp_i))

1. **αν** η bp_i είναι απλή υποδιαδικασία
2. **τότε** Δημιουργία νέου κόμβου abr ¹
3. ΕΠΕΣΤΡΕΨΕ abr ²
4. **αλλιώς**
5. Δημιουργία νέου γράφου HDAG
6. $Q \leftarrow \emptyset$
7. Εύρεση του σημείου εκκίνησης s του BPMF(bp_i)
8. ΠΡΟΣΘΗΚΗ(Q, s)
9. **ενώσω** $Q \neq \emptyset$
10. $p \leftarrow \text{ΑΦΑΙΡΕΣΗ}(Q)$
11. Δημιουργία νέου κόμβου p' ¹
12. **αν** οι διαδικασίες³ που συνδέονται⁵ στην p είναι παράλληλες
13. **τότε** $\text{type}(p') \leftarrow \text{flowSource}(\text{id})$
14. **αν** μετά την p ακολουθεί ένδειξη τερματισμού παράλληλων διαδικασιών⁴
15. **τότε** $\text{type}(p') \leftarrow \text{flowTermination}(\text{id})$
16. αύξηση του id
17. **για** κάθε διαδικασία³ t που συνδέεται⁵ στο BPMF(bp_i) με την p

¹ Κάθε φορά που δημιουργούμε ένα νέο κόμβο, υπονοείται ότι του περνάμε όποιες πληροφορίες θέλουμε για τη διαδικασία την οποία αντιπροσωπεύει.
πληροφορίες θέλουμε για τη διαδικασία την οποία αντιπροσωπεύει.

² Να τονιστεί και πάλι ότι ένας κόμβος θεωρείται HDAG, οπότε η επιστρεφόμενη τιμή του παραπάνω αλγορίθμου μπορεί να είναι ένας απλός κόμβος.

³ Με τον όρο διαδικασία εννοούμε τη γενικευμένη διαδικασία T (Task) με την έννοια που της αποδόθηκε στην παρούσα ενότητα. Για κάθε μορφή αναπαράστασης επιχειρηματικών διαδικασιών, υπάρχουν διαφορετικά στοιχεία τα οποία μπορούν να αποτελέσουν διαδικασίες, ο καθορισμός των οποίων πραγματοποιείται με βάση την κατά περίπτωση επιθυμητή ανάλυση.

⁴ Θεωρούμε ότι μετά το πέρας παράλληλα εκτελούμενων διαδικασιών υπάρχει πάντα κάποια ένδειξη τερματισμού των.

⁵ Μία διαδικασία συνδέεται με κάποια άλλη όταν οι 2 διαδικασίες είναι διαδοχικές κατά την εκτέλεση κάποιας πιθανής υποακολουθίας της συνολικής διαδικασίας. Παραδείγματα συνδέσμων αποτελούν η de-facto εκτέλεση μιας διαδικασίας αμέσως μετά από κάποια άλλη,

18. **αν** έχει_επισκεφτεί[t] = true
19. **τότε** ΠΡΟΣΘΗΚΗ_ΓΕΙΤΝΙΑΣΗΣ(t' , p')
20. **αλλιώς** έχει_επισκεφτεί[t] ← true
21. Δημιουργία νέου κόμβου t'^1
22. $t' \leftarrow$ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ (BPMF(t))
23. ΠΡΟΣΘΗΚΗ(HDAG, t')
24. ΠΡΟΣΘΗΚΗ_ΓΕΙΤΝΙΑΣΗΣ(t' , p')
25. ΠΡΟΣΘΗΚΗ(Q, t)
26. ΕΠΕΣΤΡΕΨΕ HDAG

Ο παραπάνω αλγόριθμος χρειάζεται τον επιπλέον χώρο μιας ουράς μήκους n , και ένα πίνακα επίσης μήκους n , όπου n το πλήθος των εμπλεκόμενων υποδιαδικασιών στη συνολική διαδικασία, άρα ο χώρος που χρειάζεται είναι της τάξης $O(n)$. Ο χρόνος που χρειάζεται για να διατρέξουμε όλες τις υποδιαδικασίες, θεωρώντας ότι η αναπαράσταση του γράφου αυτού γίνεται με τη βοήθεια λιστών γειτνίασης, είναι $O(n+e)$, όπου e το πλήθος των συνδέσμων μεταξύ των διαδικασιών.

Οι σειρές 8-11 του παραπάνω κώδικα χρειάζονται για τον προσδιορισμό του τύπου των διαδικασιών και έχουν νόημα σε μορφές που υποστηρίζουν εγγενώς την παράλληλη εκτέλεση διαδικασιών, όπως οι BPMN και BPEL. Εν γένει, όπως θα φανεί κα στον αλγόριθμο εύρεσης των διαδρομών, η σύνδεση ενός κόμβου με- παραπάνω από έναν- άλλους συμβολίζει την εναλλακτική επιλογή μεταξύ διαδικασιών (OR-branching), εκτός αν ο κόμβος είναι τύπου flowSource, οπότε οι διαδικασίες που συνδέονται με αυτόν είναι παράλληλες, αποτελούν δηλαδή συστατικά μέρη μιας υποδιαδικασίας τύπου flow.

Έχοντας φέρει κάποια business process στη μορφή ενός HDAG, που αποτελεί μία μορφή αναφοράς, μπορούμε να τη μετασχηματίσουμε στην μορφή Business Process (BP), όπως αυτή περιγράφηκε παραπάνω, κάτι που επιτυγχάνεται με τον παρακάτω αλγόριθμο.

5.1.2 Αλγόριθμος

Ο εν λόγω αλγόριθμος δέχεται ως είσοδο έναν Ιεραρχικό Κατευθυνόμενο Ακυκλικό Γράφο και παράγει όλες τις πιθανές διαδρομές, από τον αφετηριακό κόμβο μέχρι κάποιο τερματικό σημείο, της μορφής P (Path). Κάθε παραγόμενη διαδρομή αναπαριστά μία πιθανή ακολουθία εκτέλεσης της συνολικής διαδικασίας.

Ουσιαστικά ο αλγόριθμος διατρέχει το γράφο κατά πλάτος, μέχρι να συναντήσει τερματικό σημείο, διατηρώντας για κάθε κόμβο μία λίστα με τους προκατόχους του. Να σημειωθεί ότι είναι δυνατή η επίσκεψη κάποιου κόμβου παραπάνω από μία φορές.

η αντίστοιχη εκτέλεση υπό συνθήκη, η συμμετοχή μιας διαδικασίας σε συλλογή παράλληλα εκτελούμενων διαδικασιών που έπονται μίας άλλης διαδικασίας κ.τ.λ..

Στη συνέχεια, ξεκινώντας από ένα τερματικό σημείο και πηγαίνοντας προς τα πίσω με όλους τους δυνατούς συνδυασμούς, βάσει των λιστών προκατόχων του κάθε κόμβου, δημιουργούνται (ανεστραμμένες) όλες οι διαδρομές που μπορούν να οδηγήσουν σε αυτό το τερματικό σημείο. Αντιστρέφοντας λοιπόν όλες αυτές τις διαδρομές για όλα τα τερματικά σημεία του γράφου παράγονται όλες οι δυνατές διαδρομές. Ακριβώς η ίδια διαδικασία ακολουθείται και για κάθε κόμβο του γράφου.

5.1.3 Ψευδοκώδικας Αλγορίθμου

Ο παρακάτω αλγόριθμος (Αλγόριθμος 5.1.2) αποτελεί μια προπαρασκευαστική διαδικασία για την εφαρμογή του Αλγορίθμου 5.1.3, κατά την οποία ευρίσκονται και αποθηκεύονται όλοι οι προκατόχοι για κάθε κόμβο του Γράφου.

Αλγόριθμος 5.1.2 : Εύρεση Προκατόχων Κάθε Κόμβου Του HDAG

Είσοδος: HDAG(bp_i) : η αναπαράσταση στη μορφή HDAG της επιχειρηματικής διαδικασίας bp_i .

Έξοδος: void

Ψευδοκώδικας:

EYΡΕΣΗ_ΠΡΟΚΑΤΟΧΩΝ (HDAG (bp_i))

1. αν ο HDAG (bp_i) είναι απλός κόμβος
2. τότε καμία ενέργεια
1. αλλιώς
2. $Q \leftarrow \emptyset$
3. Εύρεση του σημείου εκκίνησης¹ s του HDAG(bp_i)
4. ΠΡΟΣΘΗΚΗ(Q, s)
5. ενόσω $Q \neq \emptyset$
6. $u \leftarrow \text{ΑΦΑΙΡΕΣΗ}(Q)$
7. αν ο u είναι τερματικό σημείο²
8. τότε ΠΡΟΣΘΗΚΗ(ΤΕΡΜΑΤΙΚΑ_ΣΗΜΕΙΑ, u)
9. για κάθε κόμβο v που συνδέεται με τον u
10. ΠΡΟΣΘΗΚΗ(ΠΡΟΚΑΤΟΧΟΙ(v), u)
11. EYΡΕΣΗ_ΠΡΟΚΑΤΟΧΩΝ(v)
12. ΠΡΟΣΘΗΚΗ(Q, v)

¹ Προφανώς το σημείο εκκίνησης είναι το starting point (sp) για τη διαδικασία bp_i .

² Τερματικό σημείο είναι κάποιο finishing point (fp) για τη διαδικασία.

Αλγόριθμος 5.1.3 : Μετασχηματισμός HDAG Σε BP

Είσοδος: HDAG(bp_i) : η αναπαράσταση στη μορφή HDAG της επιχειρηματικής διαδικασίας bp_i .

Έξοδος: BP(bp_i) : η αναπαράσταση στη μορφή BP (Business Process) της επιχειρηματικής διαδικασίας bp_i , όπως αυτή έχει οριστεί στο παρόν κεφάλαιο.

Ψευδοκώδικας:

ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ (HDAG (bp_i))

1. **αν** ο HDAG (bp_i) είναι απλός κόμβος
2. **τότε** ΕΠΕΣΤΡΕΨΕ HDAG (bp_i)
3. **αλλιώς**
4. **για** κάθε τερματικό σημείο $pfpr$ του HDAG(bp_i)
5. Δημιουργία νέας συλλογής BP
6. Δημιουργία νέας συλλογής Path
7. **$pfpr' \leftarrow$** ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ($pfpr$)
8. ΠΡΟΣΘΗΚΗ(Path , $pfpr'$)
9. ΠΡΟΣΘΗΚΗ(Q , Path)
10. **ενόσω** $Q \neq \emptyset$
11. $lu \leftarrow$ ΑΦΑΙΡΕΣΗ(Q)
12. $u \leftarrow$ ΠΡΩΤΟ_ΣΤΟΙΧΕΙΟ(lu)
13. **για** κάθε διαδικασία w , όπου $w \in$ ΠΡΟΚΑΤΟΧΟΙ (u)
14. Δημιουργία νέας συλλογής Path2
15. $w' \leftarrow$ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ(w)
16. ΠΡΟΣΘΗΚΗ(Path2 , w')
17. Path2 \leftarrow ΣΥΝΕΝΩΣΗ(Path2 , lu)
18. ΠΡΟΣΘΗΚΗ(Q , Path2)
19. **αν** η w είναι σημείο εκκίνησης
20. **τότε** ΑΝΤΙΣΤΡΟΦΗ(Path2)
21. ΠΡΟΣΘΗΚΗ(BP , Path2)
22. ΕΠΕΣΤΡΕΨΕ BP

- Θεωρούμε ότι η προτεραιότητα όλων των υποδιαδικασιών που συμμετέχουν σε μία σύνθετη υποδιαδικασία παράλληλης εκτέλεσης (flow) είναι η ίδια.
- Θεωρούμε επίσης ότι μία υποδιαδικασία που συνδέεται με παράλληλες υποδιαδικασίες δεν είναι δυνατόν να συνδέεται με άλλες υποδιαδικασίες πλην αυτών των παράλληλων διαδικασιών.

Ο παρακάτω αλγόριθμος (Αλγόριθμος 5.1.4) χρησιμεύει και είναι απαραίτητος για την περίπτωση ύπαρξης παράλληλα εκτελούμενων διαδικασιών.

Αλγόριθμος 5.1.4 : Διόρθωση Παράλληλων Διαδικασιών

Είσοδος: $BP(bp_i)$: η αναπαράσταση στη μορφή BP (Business Process) της επιχειρηματικής διαδικασίας bp_i , με διαχωρισμένες όμως σε διαφορετικά μονοπάτια τις τυχόν υπάρχουσες παράλληλα εκτελούμενες υποδιαδικασίες.

Έξοδος: $BP'(bp_i)$: η αναπαράσταση στη μορφή BP (Business Process) της επιχειρηματικής διαδικασίας bp_i , με τις παράλληλα εκτελούμενες υποδιαδικασίες να ανήκουν στην ίδια παράλληλη υποδιαδικασία τύπου flow.

Ψευδοκώδικας:

ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ ($BP(bp_i)$)

1. Δημιουργία νέου Business Process $BP'(bp_i)$
2. για κάθε μονοπάτι Path που ανήκει στην $BP(bp_i)$
3. Εύρεση υποακολουθιών που ξεκινούν με στοιχείο του ίδιου τύπου $flowSource(id)$ και τελειώνουν σε στοιχείο του ίδιου τύπου $flowTermination(id)$ συγκρίνοντας με όλες τις υπόλοιπες διαδρομές στο $BP(bp_i)$
4. Για κάθε τέτοια υποακολουθία s , αν περιέχει δημιουργία νέου Flow¹
5. Δημιουργία νέου Sequence $sqnc$
6. Προσθήκη στο $sqnc$ όλων των στοιχείων της s
7. Προσθήκη της s στο Flow
8. Για κάθε υποακολουθία που βρίσκεται στο Flow, αντικατάστασή της στο Path που βρίσκεται με το Flow
9. Απαλοιφή κοινών υποακολουθιών στο $BP(bp_i)$
10. Προσθήκη Path στο $BP'(bp_i)$
11. Διαγραφή Path από το $BP(bp_i)$
12. Επέστρεψε $BP'(bp_i)$

5.2 Στρατηγική Αποτίμησης (Evaluation Strategy)

Κάθε επιχειρηματική διαδικασία αποτελείται από συστατικά μέρη, ο καθορισμός και το είδος των οποίων εξαρτάται από την κατά περίπτωση ανάλυση της συγκεκριμένης διαδικασίας και ως προς τα μεγέθη τα οποία κάθε φορά ενδιαφέρουν.

Το στοιχείο κλειδί για την ανάλυση των επιχειρηματικών διαδικασιών είναι ο καθορισμός στοιχείων τέτοιων, ώστε η περεταίρω διαίρεσή τους να στερείται σημασίας ή και νοήματος

¹ Προτεραιότητα σε αυτή τη διαδικασία έχουν οι υποακολουθίες που βρίσκονται στο μεγαλύτερο «βάθος», για την περίπτωση που έχουμε φωλιασμένες κοινές υποακολουθίες.

ως προς τα αναλυόμενα μεγέθη. Τέτοια στοιχεία αποτελούν οι ατομικές διαδικασίες ή Atomic Tasks (AT) όπως τις έχουμε παρουσιάσει στο παρόν κεφάλαιο, δηλαδή αυτοτελείς υποδιαδικασίες που δεν μπορούν (ή δεν κρίνεται σκόπιμο) να διασπαστούν σε επιμέρους υποδιαδικασίες. Να σημειωθεί πως, όπως έχει προαναφερθεί (βλ. Ενότητα 4), οι ατομικές αυτές διαδικασίες ενδέχεται μεν να αποτελούνται και αυτές από συστατικά μέρη ως προς το υπό ανάλυση μέγεθος, η αποτίμηση όμως ενός μεγέθους για μία ατομική διαδικασία πραγματοποιείται σε διαφορετικό επίπεδο απ' ότι η αποτίμηση σε επίπεδο διαδικασιών.

Υπό αυτή την οπτική γωνία, μπορούμε να θεωρήσουμε μία επιχειρηματική διαδικασία ως μία σύνθεση από ατομικές διαδικασίες, οι οποίες συμμετέχουν σε διάφορες επιμέρους υποδιαδικασίες. Με άλλα λόγια, μια επιχειρηματική διαδικασία αποτελείται από διάφορες άλλες επιχειρηματικές διαδικασίες κ.ο.κ. των οποίων δομικά συστατικά αποτελούν οι ατομικές διαδικασίες.

Είναι λοιπόν φυσικό και σκόπιμο οι ατομικές αυτές διαδικασίες να αντιστοιχίζονται με συγκεκριμένα χαρακτηριστικά, οπότε και η αποτίμηση συνολικών μεγεθών σε μία επιχειρηματική διαδικασία καθορίζεται από τις ατομικές διαδικασίες που την απαρτίζουν και από τον τρόπο με τον οποίο αυτές σχετίζονται μεταξύ τους και με τις επιμέρους υποδιαδικασίες.

Σημαντικό ρόλο διαδραματίζει και η επιλογή συγκεκριμένης στρατηγικής για την αποτίμηση συνολικών μεγεθών, η οποία καθορίζει τον τρόπο με τον οποίο «συνεισφέρει» η κάθε υποδιαδικασία στον υπολογισμό του συνολικού μεγέθους και τον τρόπο με τον οποίο υπολογίζεται ένα συνολικό μέγεθος όταν είναι γνωστή η συνεισφορά των επιμέρους υποδιαδικασιών.

Αξίζει να παρατηρήσουμε ότι, λόγω της δομής των επιχειρηματικών διαδικασιών, η αποτίμηση ενός μεγέθους είναι δυνατόν να πραγματοποιηθεί σε όποιο επίπεδο (βάθος) είναι επιθυμητό. Για παράδειγμα μπορούμε να αποτιμήσουμε ένα μέγεθος τόσο σε επίπεδο της συνολικής επιχειρηματικής διαδικασίας, όσο και σε επίπεδο κάποιας συγκεκριμένης υποδιαδικασίας αυτής της διαδικασίας, ή ακόμη και σε επίπεδο συγκεκριμένης ατομικής διαδικασίας.

5.2.1 Εισαγωγικό Μέρος

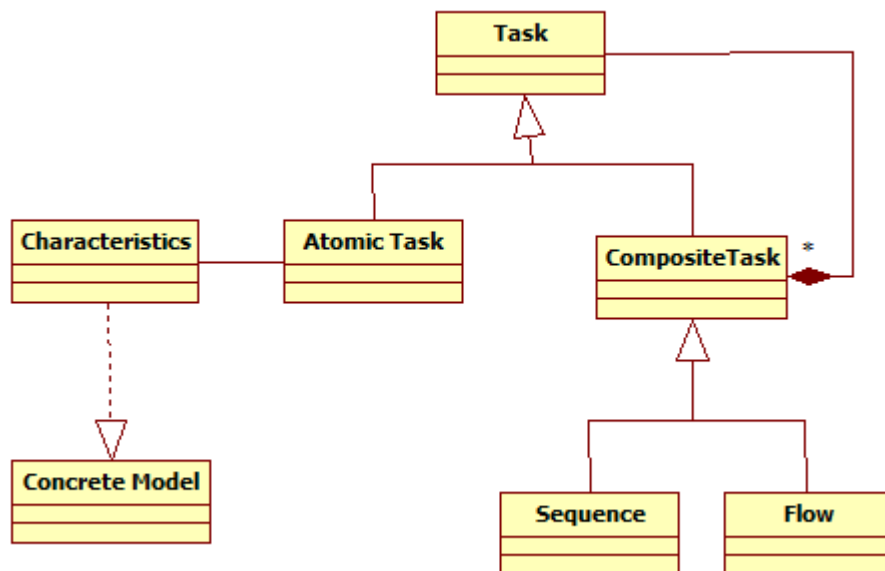
Το σύστημα αναλύει διαδικασίες αφού πρώτα τις φέρει στη μορφή BP (Business Process), όπως αυτή έχει οριστεί στο παρόν κεφάλαιο, δηλαδή μία συλλογή από μονοπάτια, καθένα εκ των οποίων αποτελεί επίσης ένα BP (Business Process) κ.ο.κ..

Με βάση τους ορισμούς στο παρόν κεφάλαιο (βλ.5.1.1), προκύπτει ότι ένα BP είναι και το ίδιο ένα Task (T), το οποίο είναι μία συλλογή από άλλα Tasks, καθένα εκ των οποίων είναι είτε ένα Composite Task (CT) είτε ένα Atomic Task (AT). Ένα Composite Task

χαρακτηρίζεται είτε ως Sequence είτε ως Flow, ανάλογα με το αν οι διαδικασίες που συμμετέχουν σε αυτό εκτελούνται ακολουθιακά ή παράλληλα. Όπου δεν αναφέρεται διαφορετικά, το Composite Task θεωρείται τύπου Sequence.

Κάθε Atomic Task αντιστοιχίζεται με συγκεκριμένα χαρακτηριστικά. Τα χαρακτηριστικά αυτά περιέχουν πληροφορίες σε μια δομή που καθορίζεται από ένα συγκεκριμένο μοντέλο, δηλαδή για κάθε χαρακτηριστικό που εξετάζουμε έχουμε ένα συγκεκριμένο μοντέλο και σε κάθε Atomic Task αντιστοιχίζουμε πληροφορίες σε μορφή συμβατή με αυτό το μοντέλο .

Οι σχέσεις των στοιχείων αυτών, όπως περιγράφηκαν, φαίνονται στο παρακάτω διάγραμμα (Σχήμα 5.1):



Σχήμα 5.1: Το μοντέλο της Διαδικασίας (Task)

Κάθε Atomic Task αντιστοιχίζεται με μία συγκεκριμένη στρατηγική, που υποδεικνύει τον τρόπο με τον οποίο θα υπολογίζεται το μέγεθος για αυτό το Task, όταν είναι γνωστή η συνεισφορά των επιμέρους τμημάτων αυτού. Επιπλέον, κάθε Task αντιστοιχίζεται με μία συγκεκριμένη στρατηγική, που υποδεικνύει τον τρόπο με τον οποίο θα υπολογίζεται το συνολικό μέγεθος, όταν είναι γνωστή (δηλαδή έχει υπολογιστεί ήδη) η συνεισφορά των επιμέρους υποδιαδικασιών που συμμετέχουν σε αυτήν. Ας σημειωθεί ότι σε κάθε στρατηγική αντιστοιχίζεται μία στρατηγική που εφαρμόζεται κάθε φορά που αυτή η στρατηγική χρησιμοποιείται για την αποτίμηση παράλληλων διαδικασιών, η οποία μπορεί να διαφέρει ή

όχι από την αρχική. Επίσης, σε κάθε στρατηγική αντιστοιχίζεται μία στρατηγική που εφαρμόζεται κάθε φορά που αυτή η στρατηγική χρησιμοποιείται για την αποτίμηση απλών διαδικασιών, η οποία μπορεί να διαφέρει ή όχι από την αρχική. Να αναφέρουμε ακόμη ότι για κάθε στρατηγική υπάρχει ένα αρχικό σημείο (starting point) που είναι ίσο με το ουδέτερο στοιχείο του τελεστή τον οποίο εφαρμόζει αυτή η στρατηγική.

5.2.2 Αλγόριθμος

Σε αυτή την παράγραφο παρουσιάζεται ο αλγόριθμος για την αποτίμηση ενός συγκεκριμένου Task ως προς ένα συγκεκριμένο χαρακτηριστικό.

Ο αλγόριθμος δέχεται ως είσοδο ένα Task στη μορφή BP (Business Process) , όπως αυτή έχει οριστεί στο παρόν κεφάλαιο. Για κάθε στοιχείο αυτού του Task εφαρμόζεται μία μέθοδος που την καλούμε calculateValue. Αυτή η μέθοδος παρουσιάζει διαφορετική συμπεριφορά όταν εφαρμόζεται σε Composite Tasks και διαφορετική σε Atomic Tasks. Στα πρώτα προκαλεί την εφαρμογή της calculateValue για καθένα εκ των στοιχείων που συμμετέχουν σε αυτά , ενώ στα δεύτερα προκαλεί την επιστροφή της τιμής τους, που προκύπτει από την εφαρμογή μιας άλλης μεθόδου, της estimateValue. Η estimateValue παίρνει ως είσοδο τις πληροφορίες για το Atomic Task ως προς το συγκεκριμένο χαρακτηριστικό και υπολογίζει μία συγκεκριμένη τιμή, την οποία και επιστρέφει. Μεταξύ των τιμών που έχουν υπολογιστεί για καθένα εκ των στοιχείων του κάθε Task, εφαρμόζεται ένας τελεστής που καθορίζεται από την αντίστοιχη στρατηγική του κάθε Task. Το ίδιο συμβαίνει και για τον υπολογισμό που παραγματοποιεί η estimateValue, δηλαδή μεταξύ των επιμέρους στοιχείων ενός Atomic Task, αφού υπολογίσει την συνεισφορά τους, εφαρμόζει ένας τελεστής που καθορίζεται από την αντίστοιχη στρατηγική του Atomic Task.

Τελικά προκύπτει μία συγκεκριμένη τιμή για το αρχικό Task, που αποτελεί και την υπολογισθείσα τιμή για το προς εξέταση χαρακτηριστικό.

5.2.3 Ψευδοκώδικας Αλγορίθμου

Αλγόριθμος 5.1.1 : Υπολογισμός Τιμής

Είσοδος: Task : μία συγκεκριμένη διαδικασία στη μορφή BP (Business Process),

Utility : ένα συγκεκριμένο χαρακτηριστικό ως προς το οποίο υπολογίζεται η έξοδος

Έξοδος: Value : μία συγκεκριμένη τιμή που αντιστοιχεί στο συγκεκριμένο Task

Ψευδοκώδικας:

calculateValue (Task , Utility)

1. **αν** το Task είναι Composite Task
2. **τότε αν** το Composite Task είναι Sequence

3. **τότε** Εύρεση Στρατηγικής es του Task
4. value←αρχικό σημείο της es
5. **για** κάθε στοιχείο t του Task
6. val←εφαρμογήΤελεστή_es(val , calculateValue (t , Utility))
7. Επέστρεψε val
- 8.
9. **αλλιώς αν** το Composite Task είναι Flow
10. **τότε** Εύρεση Στρατηγικής es του Task
11. Εύρεση αντίστοιχης Παράλληλης Στρατηγικής es'
12. value←αρχικό σημείο της es'
13. **για** κάθε στοιχείο t του Task
14. val← εφαρμογήΤελεστή_es'(val , calculateValue (t , Utility))
15. Επέστρεψε val
16. **αλλιώς αν** το Task είναι Atomic Task
17. **τότε** Εύρεση Χαρακτηριστικών chars του Atomic Task ως προς το Utility
18. Εύρεση Στρατηγικής es2 του Atomic Task
1. val← estimateValue(chars , es2)¹
19. Επέστρεψε val

Ο παραπάνω αλγόριθμος μπορεί στην πράξη να χρησιμοποιηθεί σε διάφορες ελαφρώς παραλλαγμένες μορφές. Για παράδειγμα, είναι δυνατόν κάθε φορά που γίνεται αποτίμηση μιας συγκεκριμένης διαδικασίας (Task) να τίθεται δυναμικά η αντίστοιχη στρατηγική που θα χρησιμοποιηθεί για τον υπολογισμό. Επίσης είναι δυνατόν (αλλά όχι και τόσο ουσιαστικό) να αλλάζει, κάθε φορά που γίνεται υπολογισμός μιας συγκεκριμένης διαδικασίας (Task), το προς εξέταση χαρακτηριστικό.

¹ Δεν υπεισερχόμαστε σε λεπτομέρειες που αφορούν στην μέθοδο estimateValue, καθώς αυτή εξαρτάται από τα διάφορα μοντέλα που χρησιμοποιούνται για την αναπαράσταση των διαφόρων χαρακτηριστικών. Να αναφέρουμε μόνο ότι είναι υπεύθυνη για τη μετατροπή των συστατικών μερών, ως προς το υπό ανάλυση μέγεθος, του Atomic Task, σε κανονική μορφή, δηλαδή τα φέρνει σε μία κοινώς επεξεργάσιμη μορφή, όπως για παράδειγμα στις ίδιες μονάδες μέτρησης. Επίσης είναι υπεύθυνη για τον υπολογισμό της συνεισφοράς καθενός από αυτά τα συστατικά μέρη και στη συνέχεια εφαρμογή κατάλληλου τελεστή στις επιμέρους συνεισφορές, με βάση τη στρατηγική του Atomic Task, με μια διαδικασία παρόμοια με αυτή που περιγράφηκε από τον παραπάνω αλγόριθμο της μεθόδου calculateValue.

5.3 Σύνθεση της Στρατηγικής Αποτίμησης (Composition of Evaluation Strategy)

Στην παρούσα ενότητα αναλύεται η Σύνθεση της Στρατηγικής Αποτίμησης , που αποτελεί το ανώτερο αφαιρετικό επίπεδο της ανάλυσης των διαφόρων επιχειρηματικών διαδικασιών.

Όπως έχει αναφερθεί στο παρόν κεφάλαιο, ενδιαφέρει η ανάλυση επιχειρηματικών διαδικασιών με δυνατότητα επιλογής ανάμεσα σε διάφορους δυνατούς τρόπους διεξαγωγής των, δηλαδή διαφορετικά μονοπάτια εκτέλεσης. Ο υπολογισμός των μεμονωμένων χαρακτηριστικών για τον καθένα από αυτούς τους τρόπους, δηλαδή για κάθε διαφορετικό μονοπάτι, πραγματοποιείται με βάση όσα έχουν αναφερθεί στην προηγούμενη ενότητα. Η σημασία της παρούσας ενότητας έγκειται στην παρουσίαση του τρόπου εκλογής κριτηρίων και στρατηγικών ώστε να προκύψουν συγκεκριμένα αποτελέσματα, όπως η επιλογή του κατάλληλου μονοπατιού ή η ταξινόμηση των διαφόρων μονοπατιών με βάση την ικανοποίηση των διαφόρων κριτηρίων.

5.3.1 Εισαγωγικό Μέρος

Όπως έχει προαναφερθεί, το σύστημα αναλύει διαδικασίες αφού πρώτα τις φέρει στη μορφή BP (Business Process) , όπως αυτή έχει οριστεί στο παρόν κεφάλαιο, δηλαδή μία συλλογή από μονοπάτια, καθένα εκ των οποίων αποτελεί επίσης ένα BP (Business Process) κ.ο.κ..

Για κάθε μονοπάτι, κάθε επιχειρηματικής διαδικασίας (στη μορφή BP), μπορεί να υπολογιστεί μία συγκεκριμένη τιμή, δηλαδή ένας μοναδικός αριθμός που αντιστοιχίζει ποσοτικά αυτό το μονοπάτι σε κάποιο χαρακτηριστικό. Μία τέτοια τιμή, που, ανάλογα με το χαρακτηριστικό που προσδιορίζει, μπορεί να αποτελεί τιμή μετρούμενη σε χρηματικές μονάδες, σε μονάδες χρόνου ή απλούστερα έναν αδιάστατο δείκτη , μπορεί να υπολογιστεί για κάθε διαφορετικό χαρακτηριστικό και για κάθε διαφορετική στρατηγική υπολογισμού.

Προκύπτει λοιπόν ένα μεγάλο σύνολο αποτελεσμάτων που μπορούν να εξαχθούν, ακόμη και για ένα μοναδικό μονοπάτι. Θέτοντας συγκεκριμένα κριτήρια συγκεκριμενοποιούμε και περιορίζουμε το σύνολο αυτό. Τα κριτήρια περιλαμβάνουν, προφανώς, τα συγκεκριμένα χαρακτηριστικά που ενδιαφέρουν για την εκάστοτε ανάλυση. Μαζί με κάθε χαρακτηριστικό πρέπει να δίνεται και μία στρατηγική με βάση την οποία να πραγματοποιείται ο υπολογισμός της αντίστοιχης τιμής (value calculation). Κάθε χαρακτηριστικό που συμπεριλαμβάνεται στα κριτήρια ενδέχεται να ενδιαφέρει σε διαφορετικό βαθμό από τα υπόλοιπα χαρακτηριστικά. Είναι λοιπόν απαραίτητο πέραν των χαρακτηριστικών να δίδεται και κάποια πληροφορία για την σημαντικότητα του καθενός στη συγκεκριμένη ανάλυση. Επιπλέον, για κάθε χαρακτηριστικό θα πρέπει να προσδιορίζεται αν είναι επιθυμητή η μεγιστοποίησή ή η

ελαχιστοποίησή του, ανάλογα με τη φύση αυτού του χαρακτηριστικού και την επιθυμητή ανάλυση.

Έχοντας καθορίσει τα κριτήρια της ανάλυσης, κρίνεται σκόπιμη η εκλογή κάποιας συγκεκριμένης στρατηγικής για τον υπολογισμό αποτελεσμάτων. Αυτή η στρατηγική ενδέχεται να απαιτεί τη σύγκριση μεταξύ τιμών για διαφορετικά χαρακτηριστικά. Για να καταστεί δυνατή μία τέτοια σύγκριση είναι απαραίτητη η κανονικοποίηση όλων των τιμών. Ένας αλγόριθμος για μια τέτοια κανονικοποίηση είναι ο εξής (Αλγόριθμος 5.3.1):

Αλγόριθμος 5.3.1 : Κανονικοποίηση Τιμών Όταν Είναι Επιθυμητή Η Μειστοποίησή Τους

Είσοδος: **Values** : μία συλλογή τιμών

Έξοδος: **NormalizedValues** : η συλλογή με τις τιμές κανονικοποιημένες

Ψευδοκώδικας:

normalizeValues (Values)

1. $sum \leftarrow 0$;
2. **για** κάθε στοιχείο v της Values
3. $sum \leftarrow sum + v$
4. **για** κάθε στοιχείο v της Values
5. $v \leftarrow v / sum$
6. Επέστρεψε Values

Ο παραπάνω αλγόριθμος αντικαθιστά κάθε τιμή της συλλογής με μία σχετική τιμή, σε σχέση με τις υπόλοιπες τιμές της συλλογής, ώστε το συνολικό άθροισμα των τιμών της συλλογής να είναι ίσο με τη μονάδα .

Μία γενικευμένη μορφή του παραπάνω αλγορίθμου είναι η εξής (Αλγόριθμος 5.3.2) :

Αλγόριθμος 5.3.2 : Κανονικοποίηση Τιμών

Είσοδος: **Values** : μία συλλογή τιμών

Έξοδος: **NormalizedValues** : η συλλογή με τις τιμές κανονικοποιημένες

Ψευδοκώδικας:

normalizeValues (Values)

1. $sum \leftarrow 0$
2. $N \leftarrow 0$
3. **για** κάθε στοιχείο v της Values
4. $sum \leftarrow sum + v$

5. $N \leftarrow N+1$
6. **για** κάθε στοιχείο v της Values
7. **αν** είναι επιθυμητή μεγιστοποίηση
8. **τότε** $v \leftarrow v / \text{sum}$
9. **αλλιώς αν** είναι επιθυμητή ελαχιστοποίηση
10. **τότε** $v \leftarrow (1 - (v / \text{sum})) / (N-1)$
11. Επέστρεψε Values

Ο παραπάνω αλγόριθμος ελέγχει επιπλέον αν ενδιαφέρει η μεγιστοποίηση ή ελαχιστοποίηση του χαρακτηριστικού και εκτελεί ανάλογη κανονικοποίηση των τιμών, ώστε πάντα να θεωρείται «καλύτερη» η μεγαλύτερη τιμή και το άθροισμα όλων των τιμών να είναι ίσο με τη μονάδα.

Να σημειωθεί ότι για να έχει νόημα ο παραπάνω αλγόριθμος θα πρέπει όλες οι τιμές στη συλλογή να είναι θετικές. Επίσης, για να έχει το επιθυμητό αποτέλεσμα η κανονικοποίηση, απαραίτητη προϋπόθεση είναι η φύση των χαρακτηριστικών να είναι τέτοια ώστε η καταλληλότητα κάποιου στοιχείου ως προς κάποιο χαρακτηριστικό να αυξάνεται γραμμικά όσο αυξάνεται η τιμή που του αντιστοιχίζεται ως προς αυτό το χαρακτηριστικό.

5.3.2 Αλγόριθμος

Σε αυτό το κεφάλαιο παρουσιάζεται ο αλγόριθμος για την αξιολόγηση μίας επιχειρηματικής διαδικασίας με βάση συγκεκριμένα κριτήρια και στρατηγική.

Ο αλγόριθμος δέχεται ως είσοδο μία επιχειρηματική διαδικασία στη μορφή BP (Business Process) , όπως αυτή έχει οριστεί στο παρόν κεφάλαιο. Αφού επιλεγθούν τα κατάλληλα κριτήρια και το είδος τους, για κάθε μονοπάτι εκτέλεσης αυτής της διαδικασίας υπολογίζονται όλες οι τιμές, μία για κάθε διαφορετικό χαρακτηριστικό που συμπεριλαμβάνεται στα κριτήρια, και στη συνέχεια εφαρμόζεται μία επιλεγθείσα στρατηγική για την εξαγωγή αποτελεσμάτων. Ως παράδειγμα, ο συγκεκριμένος αλγόριθμος επιστρέφει ως αποτέλεσμα ένα συγκεκριμένο μονοπάτι εκτέλεσης και συγκεκριμένα το καταλληλότερο, με βάση τα κριτήρια και τη στρατηγική, εκ των εναλλακτικών μονοπατιών της συγκεκριμένης επιχειρηματικής διαδικασίας. Για την εφαρμογή της στρατηγικής ενδεχομένως να χρειαστεί και κάποια κανονικοποίηση των δεδομένων, όπως έχει προαναφερθεί. Επιπρόσθετα, ανάλογα με τον τύπο των κριτηρίων, ενδέχεται να απαιτείται περαιτέρω κανονικοποίηση των αποτελεσμάτων, αντανακλώντας τη σημαντικότητα του κάθε χαρακτηριστικού.

5.3.3 Ψευδοκώδικας Αλγορίθμου

Αλγόριθμος 5.3.3 : Επιλογή Διαδρομής

Είσοδος: BP : μία συγκεκριμένη επιχειρηματική διαδικασία στη μορφή BP (Business Process)

Έξοδος: Path : μία συγκεκριμένη διαδρομή στη μορφή P (Path), όπως έχει οριστεί στο παρόν κεφάλαιο

Ψευδοκώδικας:

choosePath (BP)

1. Επιλογή Τύπου Κριτηρίων CT
2. Δημιουργία συγκεκριμένων Κριτηρίων criteria , τύπου CT
3. Επιλογή Στρατηγικής Αξιολόγησης as
4. Δημιουργία νέας Συλλογής U
5. για κάθε χαρακτηριστικό u που υπάρχει στα criteria
6. ΠΡΟΣΘΗΚΗ (U , u)
7. Δημιουργία Δομής Σημαντικότητας Χαρακτηριστικών importanceVector με βάση τον CT
8. Αποτελέσματα←Πίνακας αποτελεσμάτων από calculateValue για όλα τα u της U και για όλα τα μονοπάτια της BP
9. αν το απαιτεί η στρατηγική as
10. τότε Κανονικοποίηση αποτελεσμάτων
11. Εφαρμογή importanceVector στα Αποτελέσματα
12. Επιλογή μονοπατιού Path με βάση τη στρατηγική as
13. Επέστρεψε Path

Με κάποιες τροποποιήσεις, ο παραπάνω αλγόριθμος μπορεί να επιστρέφει αντί για ένα μονοπάτι όλα τα μονοπάτια σε σειρά προτεραιότητας με βάση την καταλληλότητά τους που καθορίζεται κάθε φορά από τα κριτήρια και τη στρατηγική αξιολόγησης. Επίσης θα μπορούσε να επιστρέφει συγκεκριμένες τιμές, όπως το μέσο όρο για όλα τα μονοπάτια υπολογισμένων τιμών για συγκεκριμένο χαρακτηριστικό (αναμενόμενη τιμή για αυτό το χαρακτηριστικό) κ.τ.λ..

Αυτό που αξίζει να παρατηρήσουμε είναι η πολυμορφικότητα και η ευελιξία του συστήματος, καθώς είναι δυνατή η επιλογή όχι μόνο χαρακτηριστικών αλλά και τύπου κριτηρίων, στρατηγικών υπολογισμού των διαφόρων τιμών και στρατηγικής αξιολόγησης , με βάση τα οποία πραγματοποιείται και παίρνει διαφορετική μορφή κάθε φορά η ζητούμενη ανάλυση των επιχειρηματικών διαδικασιών.

6

Μελέτες Περιπτώσεων *(Case Studies)*

Η ανάλυση μιας μελέτης περίπτωσης (case study) αποτελεί αναπόσπαστο κομμάτι της στρατηγικής των επιχειρήσεων. Μια μελέτη περίπτωσης παραθέτει, με τεκμηριωμένο τρόπο, τα γεγονότα (πραγματικά ή φανταστικά) που περιλαμβάνει μια συγκεκριμένη διοικητική κατάσταση, τοποθετεί τον αναγνώστη σε θέση ανάληψης δράσης και τον εξοικειώνει με ανάλογες περιπτώσεις, ενώ ενδέχεται να θέσει και κάποια ερωτήματα προς περαιτέρω διερεύνηση [39]. Ειδικότερα στην Τεχνολογία Λογισμικού (Software Engineering), η τεχνική αυτή χρησιμοποιείται για την αποτύπωση και επαλήθευση της απαιτούμενης συμπεριφοράς ενός συστήματος με είσοδο (input) εμπειριστατωμένα παραδείγματα βασισμένα στις λειτουργικές απαιτήσεις.

Γενικά, ανάλογα με το επίπεδο ανάλυσης, υπάρχουν τρεις τύποι μελετών περιπτώσεων με προσανατολισμό τη στρατηγική στις επιχειρήσεις:

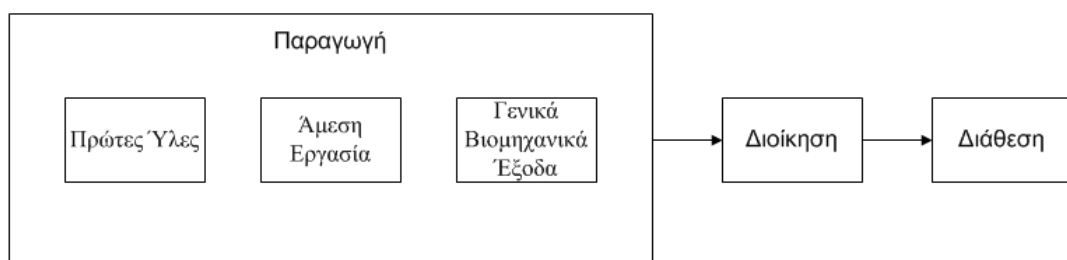
- Τύπος 1: η ανώτατη διοίκηση έρχεται αντιμέτωπη με μια σειρά καταστάσεων που απαιτούν τη λήψη στρατηγικών αποφάσεων. Το προς επίλυση πρόβλημα μπορεί είτε να είναι εμφανές είτε να χρειάζεται ο προσδιορισμός του, πριν την αναζήτηση λύσεων, η οποία πραγματοποιείται με μια σειρά γνωματεύσεων ως προς τις δράσεις που πρέπει να αναληφθούν.
- Τύποι 2 και 3 : αποτελούν ένα ανώτερο επίπεδο ανάλυσης και παρουσιάζουν γενικές εταιρικές πληροφορίες (Τύπος 2) ή πληροφορίες σχετικά με το στυλ ηγεσίας (Τύπος 3). Αυτές οι μελέτες περιπτώσεων δεν περιλαμβάνουν ένα συγκεκριμένο προς επίλυση πρόβλημα, αλλά παρέχουν μια ολοκληρωμένη εικόνα της λειτουργίας μιας επιχείρησης.

Σε αυτό το κεφάλαιο επιχειρείται μια μελέτη περιπτώσεων Τύπου 1, για την ανάδειξη των δυνατοτήτων του συστήματος κατά τη χρησιμοποίησή του στην επίλυση προβλημάτων. Τα προβλήματα αυτά είναι φανταστικά και επιλέχθηκαν με βάση τη γενικότητά και την αντιπροσωπευτικότητά τους. Να σημειωθεί ότι είναι εξαιρετικά δύσκολη έως και αδύνατη η εξεύρεση παραδειγμάτων τέτοιων μελετών περιπτώσεων σε πόρους όπως αυτούς του διαδικτύου (και μάλιστα άνευ κάποιου αντιτίμου), κυρίως για τον προφανή λόγο ότι τέτοιας φύσης προβλήματα συνήθως αποτελούν μέρος των απόρρητων πληροφοριών των επιχειρήσεων.

6.1 Παράδειγμα (Α) : Υπολογισμός Κόστους

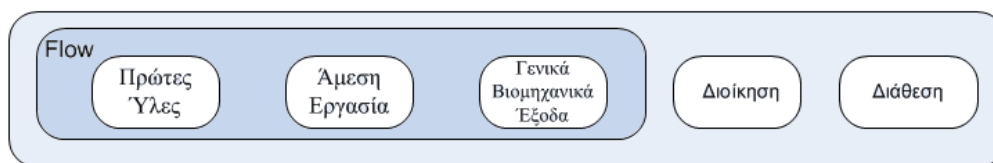
Μία απλή χρήση του συστήματος είναι ο υπολογισμός συνολικών μεγεθών για μία διαδικασία. Η συνεισφορά του συστήματος σε τέτοιου είδους υπολογισμούς έγκειται στην δυνατότητα προσομοίωσης σύνθετων, ιεραρχικών δομών και στην ευκολία ανάλυσής τους σε διαφορετικά σενάρια, η οποία αποκαλείται what-if analysis.

Στην παρούσα ενότητα περιγράφεται η διαδικασία υπολογισμού του συνολικού ετήσιου κόστους λειτουργίας της επιχείρησης «Monkey AE» [40] . Από μία υψηλού επιπέδου αφαιρετική σκοπιά η λειτουργία της επιχείρησης περιλαμβάνει τις διαδικασίες της Παραγωγής, της Διοίκησης και της Διάθεσης. Η Παραγωγή περιλαμβάνει διαδικασίες που έχουν σχέση με τις Πρώτες Ύλες, την Άμεση Εργασία και τα Γενικά Βιομηχανικά Έξοδα, που χάριν απλότητας τις αναφέρουμε ως Πρώτες Ύλες, Άμεση Εργασία και Γενικά Βιομηχανικά Έξοδα . Η λειτουργία λοιπόν αυτή φαίνεται στο Σχήμα 6.1 που υλοποιήθηκε με τη βοήθεια του προγράμματος Microsoft Visio:



Σχήμα 6.1: Η λειτουργία της επιχείρησης από μία υψηλού επιπέδου αφαιρετική σκοπιά

Στο Σχήμα 6.2 φαίνεται η αναπαράσταση της παραπάνω διαδικασίας σε μορφή επεξεργάσιμη από το σύστημα, όπως έχει παρουσιαστεί στο κεφάλαιο 5.



Σχήμα 6.2: Η αναπαράσταση της λειτουργίας της επιχείρησης στο σύστημα

Όπως φαίνεται η διαδικασία της Παραγωγής αναπαρίσταται με ένα Flow, εφόσον οι διαδικασίες που περιλαμβάνει εκτελούνται παράλληλα.

Για κάθε atomic task δίνονται συγκεκριμένα κόστη, που φαίνονται στους παρακάτω πίνακες:

Προϋπολογισμός αγοράς Πρώτων Υλών	Πρώτη Ύλη	
	Y1	Y2
Επιθυμητό τελικό απόθεμα πρώτων υλών (σε μονάδες)	6.000	1.000
Πλέον: Μονάδες που απαιτούνται για την παραγωγή ^(α)	84.000	44.000
Συνολικές ανάγκες	90.000	45.000
Μείον: Αρχικό απόθεμα πρώτων υλών (σε μονάδες)	5.000	5.000
Μονάδες που πρέπει να αγοραστούν	85.000	40.000
Τιμή αγοράς (σε ευρώ)	1,2	2,6
<i>(α) Από τον προϋπολογισμό χρήσης Άμεσων Υλικών</i>		

Πίνακας 6.1: Προϋπολογισμός αγοράς Πρώτων Υλών

Προϋπολογισμός Άμεσης Εργασίας	Μονάδες προϊόντος	Ώρες άμεσης εργασίας ανά μονάδα	Σύνολο ωρών	Κόστος Άμεσης Εργασίας (σε ευρώ)
Προϊόν Π1	6000 ^(α)	14	84000	2.05
Προϊόν Π2	1000 ^(α)	20	20000	2.05
<i>(α) Από τον προϋπολογισμό παραγωγής</i>				

Πίνακας 6.2: Προϋπολογισμός Άμεσης Εργασίας

Προϋπολογισμός Γενικών Βιομηχανικών Εξόδων (πόσα σε ευρώ)	
Μεταβλητά Έξοδα	
Διάφορες προμήθειες	30000
Έμμεση εργασία	70000
Παρεπόμενα έξοδα μισθοδοσίας	25000
Ενέργεια (μεταβλητό μέρος)	8000
Συντήρηση (μεταβλητό μέρος)	20000
Σταθερά Έξοδα	
Αποσβέσεις	25000
Φόροι περιουσίας	4000
Ασφάλειες	500
Επίβλεψη	20000
Ενέργεια (σταθερό μέρος)	1000
Συντήρηση (σταθερό μέρος)	4500

Πίνακας 6.3: Προϋπολογισμός Γενικών Βιομηχανικών Εξόδων

Προϋπολογισμός εξόδων διοικητικής λειτουργίας και λειτουργίας διάθεσης (ποσά σε ευρώ)	
Έξοδα Λειτουργίας Διάθεσης	
Προμήθειες πωλήσεων	20000
Διαφήμιση	3000
Μισθοί προσωπικού πωλήσεων	10000
Ταξίδια	5000
Έξοδα Διοικητικής Λειτουργίας	
Μισθοί υπαλλήλων	10000
Διάφορες προμήθειες	1000
Μισθοί ανώτερων στελεχών	21000
Διάφορα	5000

Πίνακας 6.4: Προϋπολογισμός εξόδων διοικητικής λειτουργίας και λειτουργίας διάθεσης

Προκύπτει λοιπόν, με εξομοίωση στο σύστημα, ότι το ετήσιο κόστος για τις βασικές λειτουργίες της επιχείρησης «Monkey AE» ανέρχεται στα 702200 Ευρώ . Μεταβάλλοντας οποιαδήποτε από τις τιμές στους παραπάνω πίνακες, δηλαδή τις αντίστοιχες τιμές στα μοντέλα του συστήματος, προκύπτει κάθε φορά το ζητούμενο συνολικό κόστος.

6.2 Λήψη Αποφάσεων

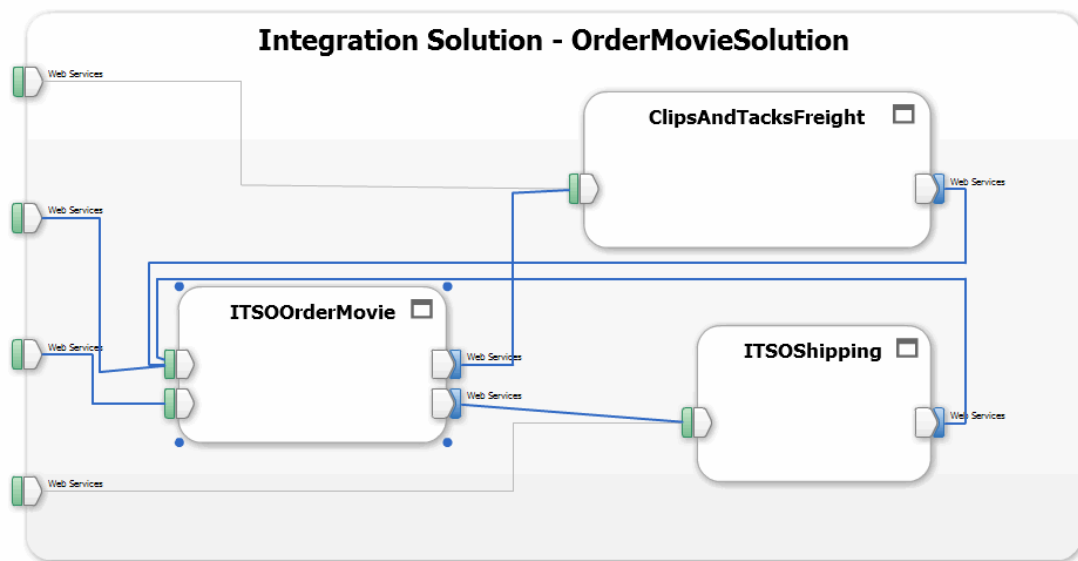
Το σύστημα προσφέρει μεταξύ άλλων τη δυνατότητα εύρεσης των βέλτιστων επιλογών με βάση συγκεκριμένα κριτήρια και συγκεκριμένες στρατηγικές. Τα συστήματα που προσφέρουν τέτοιες δυνατότητες αποτελούν χρήσιμα εργαλεία για τη λήψη κρίσιμων στρατηγικών και άλλων αποφάσεων.

Όπως έχει προαναφερθεί, ενδιαφέρει η ανάλυση επιχειρηματικών διαδικασιών με δυνατότητα επιλογής ανάμεσα σε διάφορους δυνατούς τρόπους διεξαγωγής των, δηλαδή διαφορετικά μονοπάτια εκτέλεσης. Μία βασική λειτουργία του συστήματος είναι η επιλογή της βέλτιστης διαδρομής ανάμεσα στις δυνατές, για την επίδειξη της οποίας παρουσιάζονται τα παραδείγματα που ακολουθούν.

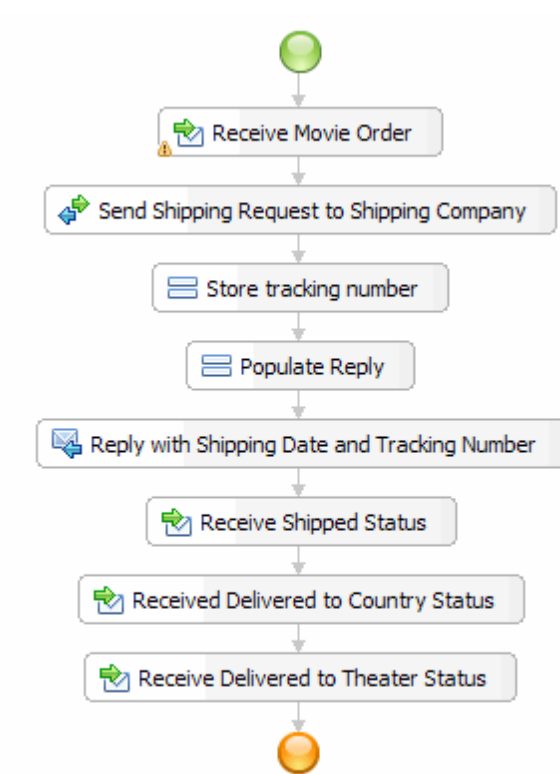
6.2.1 Παράδειγμα (B) : Αξιολόγηση Υπηρεσιών

Λόγω της φύσης και της δομής των διεργασιών που επεξεργάζεται το σύστημα, ένα χαρακτηριστικό παράδειγμα τέτοιων διεργασιών μπορούν να αποτελέσουν οι Υπηρεσίες (Services). Ο συντονισμός αλληλοεξαρτούμενων υπηρεσιών είναι ένας τομέας που συνεχώς εξελίσσεται, κυρίως για την υλοποίηση συστημάτων με βάση την υπηρεσιο-στρεφή αρχιτεκτονική (SOA – Service Oriented Architecture), με πρότυπα όπως η γλώσσα BPEL [36] να έχουν ήδη επικρατήσει. Κρίνεται λοιπόν εξαιρετικής σημασίας η αξιολόγηση τέτοιων υπηρεσιών και η επιλογή μεταξύ αυτών των κατάλληλων υπηρεσιών που ανταποκρίνονται κάθε φορά σε συγκεκριμένες πολιτικές και κριτήρια.

Για το παράδειγμα που ακολουθεί δανειζόμαστε το σενάριο της ITSO Movie [41] . Η ITSO Movie είναι ένας διανομέας ταινιών σε κινηματογράφους σε όλο τον κόσμο. Μία από τις βασικές της επιχειρηματικές δραστηριότητες είναι η λήψη παραγγελιών για ταινίες από τους πελάτες της και η αποστολή των ταινιών μέσω των λογιστικών της συνεργατών, των ITSOShipping και ClipsAndTacksFreight. Και οι δύο συνεργάτες παρέχουν υπηρεσίες διανομής χρησιμοποιώντας παρόμοιες διεπαφές, οπότε είναι εύκολο για την ITSO Movie να εναλλάσσει τις παραγγελίες μεταξύ των δύο αυτών συνεργατών. Οι λειτουργίες που παρουσιάστηκαν παραπάνω φαίνονται στα Σχήματα 6.3 και 6.4.



Σχήμα 6.3: Η αναπαράσταση της λειτουργίας με solution diagram



Σχήμα 6.4: Η αναπαράσταση της λειτουργίας σε BPEL

Ας υποθέσουμε ότι ως μέρος αναδιαρθρωτικών διοικητικών χειρισμών στην επιχείρηση, κρίνεται σκόπιμη η αξιολόγηση των υπηρεσιών που προσφέρουν οι δύο συνεργάτες και με

βάση συγκεκριμένη πολιτική και κριτήρια η ανάδειξη της καλύτερης υπηρεσίας εξ αυτών. Κάτι τέτοιο απαιτεί την αντιστοίχιση των υπηρεσιών αυτών με συγκεκριμένα χαρακτηριστικά και ο καθορισμός των επιθυμητών κριτηρίων με βάση τα οποία να πραγματοποιηθεί η αξιολόγησή τους.

Ο τομέας Έρευνας και Ανάπτυξης (R&D) της επιχείρησης καλείται να διεξάγει έρευνα για την μέτρηση ποιοτικών χαρακτηριστικών των προσφερόμενων υπηρεσιών και με βάση και αυτήν προκύπτουν τα στοιχεία που φαίνονται στον Πίνακα 6.5:

Service	Utility		Cost		Dependability		Throughput (requests served/day)
ITSOShipping	Operational (K€/year)	3.4	Availability	0.96	1564		
	Facilities (K€/year)	1.2					
	Capital Expenses (K€/year)	2.5	Reliability	0.75			
ClipsAndTacksFreight	Rental (K\$/year)	19.3	Availability	0.98	2523		
			Reliability	0.63			

Πίνακας 6.5: Χαρακτηριστικά Υπηρεσιών

Παρατηρούμε ότι μπορεί να παρουσιάζεται ετερογένεια στη μορφή και στην πληθικότητα τόσο μεταξύ των διαφορετικών χαρακτηριστικών, όσο και για το ίδιο χαρακτηριστικό στις δύο διαφορετικές υπηρεσίες.

Με τη χρήση λοιπόν του συστήματος, μπορούμε να επιλέξουμε ως κριτήριο αξιολόγησης ένα μείγμα από χαρακτηριστικά που μας ενδιαφέρουν και το βαθμό (%) στον οποίο μας ενδιαφέρει το καθένα εξ αυτών.

Επιλέγουμε λοιπόν διαδοχικά τρία διαφορετικά κριτήρια αξιολόγησης, όπως φαίνονται στον παρακάτω πίνακα (Πίνακας 6.6):

Rate of interest for:	Cost	Dependability	Throughput
AssessmentCriteria1	50 %	75 %	100 %
AssessmentCriteria2	75 %	75 %	75 %
AssessmentCriteria3	100 %	75 %	50 %

Πίνακας 6.6: Κριτήρια Αξιολόγησης

Για να πραγματοποιηθεί η αξιολόγηση από το σύστημα, επιλέγεται ως στρατηγική η (μέση) βελτιστοποίηση των χαρακτηριστικών, που για το κόστος σημαίνει ελαχιστοποίηση ενώ για τα υπόλοιπα δύο χαρακτηριστικά μεγιστοποίηση. Επίσης, τα χαρακτηριστικά αυτά κανονικοποιούνται για να είναι δυνατή η μεταξύ τους σύγκριση και εφαρμόζεται σε αυτά ένας τελεστής ανάλογα με το ενδιαφέρον μας για καθένα εξ αυτών.

Έτσι, για κάθε κριτήριο προκύπτουν τα παρακάτω αποτελέσματα (Πίνακας 6.7) :

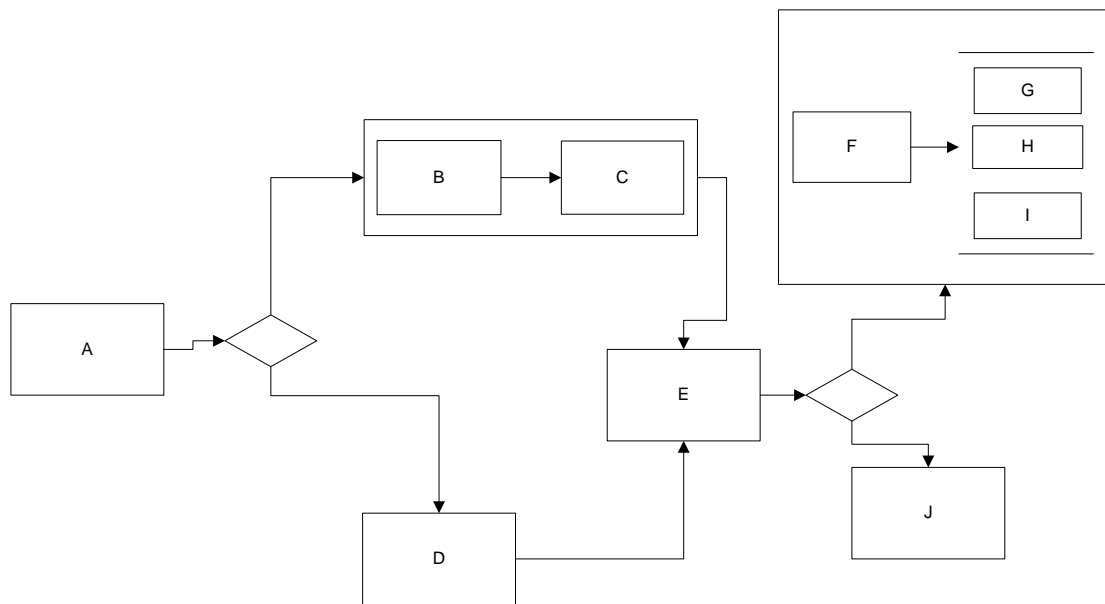
Αποτελέσματα	
AssessmentCriteria1	Επιλέγεται η ITSOShipping
AssessmentCriteria2	Επιλέγεται η ClipsAndTacksFreight
AssessmentCriteria3	Επιλέγεται η ClipsAndTacksFreight

Πίνακας 6.7: Αποτελέσματα προσομοίωσης

Αυτό το παράδειγμα ήταν χρήσιμο για την επίδειξη επιλογής διαφορετικών κριτηρίων στη λήψη αποφάσεων σε μία πολύ απλή περίπτωση. Το επόμενο παράδειγμα αφορά σε μία περισσότερο πολύπλοκη κατάσταση, λόγω της δομής του προβλήματος.

6.2.2 Παράδειγμα (Γ) : Επενδυτική Τακτική

Στο παράδειγμα αυτό υποθέτουμε ότι η εταιρία ITSO Monie [41] προγραμματίζει τα επενδυτικά της βήματα για την επόμενη τετραετία. Για το σκοπό αυτό συνεκτιμήθηκαν διάφοροι παράγοντες, όπως για παράδειγμα η στρατηγική και η ρευστότητα της εταιρίας, και η διεύθυνση της εταιρίας έχει καταλήξει σε διάφορα δυνατά σενάρια που αναπαρίστανται στο Σχήμα 6.5 υπό τη μορφή ενός Διαγράμματος Ροής που σχεδιάστηκε με τη βοήθεια του προγράμματος Microsoft Visio:



Σχήμα 6.5: Δυνατές Επενδυτικές Τακτικές της εταιρίας

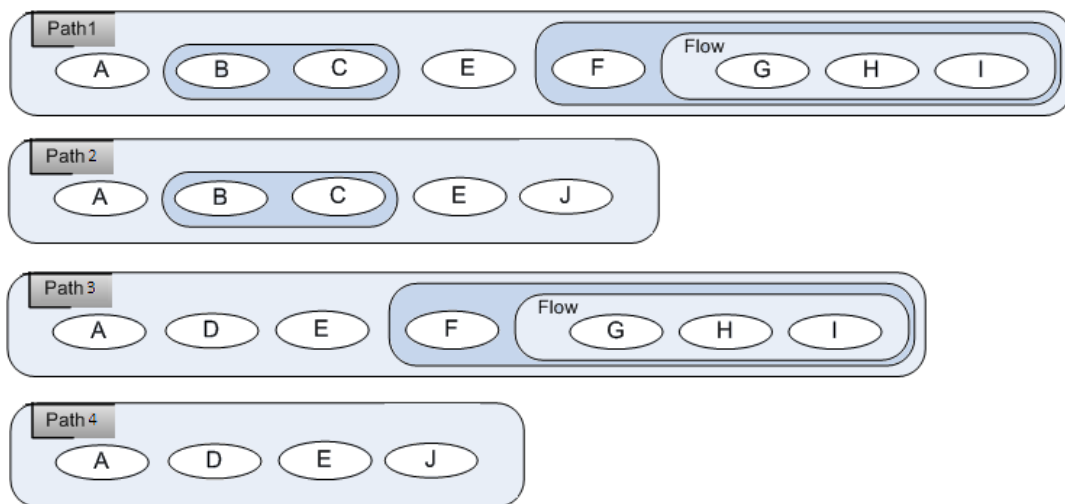
Κάθε διαδικασία (κουτάκι) στο παραπάνω διάγραμμα αναπαριστά και ένα συγκεκριμένο «επενδυτικό βήμα», το οποίο ενδεχομένως να αναπαριστά ένα πλήθος και μείγμα από εφαρμόσιμες επενδυτικές τακτικές. Παραδείγματα επενδυτικών τακτικών αποτελούν οι γεωγραφικές επεκτάσεις σε νέες περιοχές, οι επεκτάσεις της παραγωγής σε νέα προϊόντα, οι αγορές (πακέτων) μετοχών και ομολόγων κ.τ.λ.. Όπως φαίνεται και στο διάγραμμα υπάρχουν κομβικά σημεία εναλλακτικών επιλογών (ρόμβοι), καθώς επίσης και σύνθετες διαδικασίες που περιέχουν άλλες. Ακόμη, υπάρχουν διαδικασίες, μετά τις οποίες ακολουθούν παράλληλες διαδικασίες (διαδικασίες που εγκλείονται ανάμεσα σε οριζόντιες γραμμές). Το τμήμα επενδύσεων της εταιρίας προέβη σε έρευνες και εκτίμησε το κόστος και το αναμενόμενο κέρδος (απόδοση) για καθεμία από τις επενδυτικές τακτικές σε οριζόντια τετραετία, όπως φαίνεται στον παρακάτω πίνακα (Πίνακας 6.8):

Πίνακας κόστους - απόδοσης	Cost (Κ€)	Revenue (Κ€)
A	15	21
B	23	30
C	24	28
D	32	35
E	12	16

F	49	53
G	26	31
H	43	50
I	47	51
J	32	37

Πίνακας 6.8: Πίνακας κόστους – απόδοσης επενδυτικών τακτικών

Στο παρακάτω σχήμα (Σχήμα 6.6) φαίνεται η αναπαράσταση της λειτουργίας του παραδείγματος στο σύστημα. Όπως φαίνεται, προκύπτουν τέσσερις δυνατές εναλλακτικές επενδυτικές πορείες.



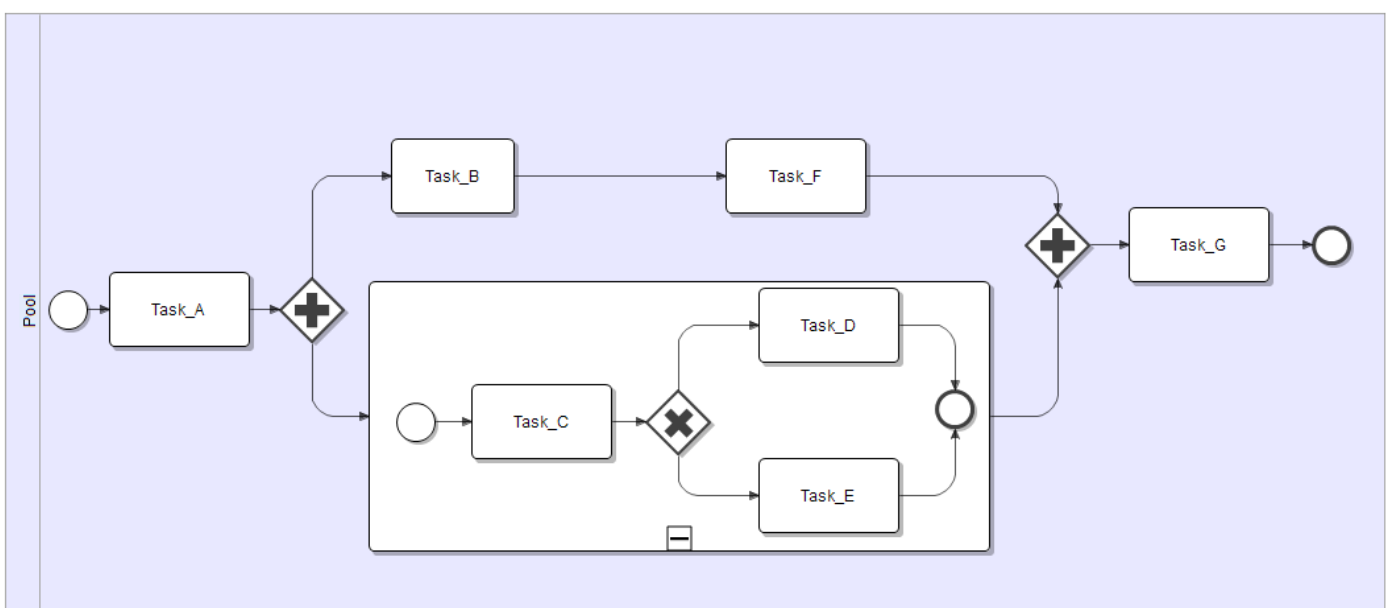
Σχήμα 6.6: Η αναπαράσταση της λειτουργίας του παραδείγματος στο σύστημα

Αποτελέσματα με βάση το ROI (έξοδα - έσοδα) : Το σύστημα υποδεικνύει την εκτέλεση των βημάτων $A \rightarrow B \rightarrow C \rightarrow E \rightarrow F \rightarrow G \rightarrow H \rightarrow I$, δηλαδή το πρώτο μονοπάτι.

6.3 Παράδειγμα (Δ) : Εύρεση Χωρητικότητας (Capacity) και Στενωπής (Bottleneck) Συστήματος

Η στενωπή (bottleneck) είναι ένα φαινόμενο κατά το οποίο η χωρητικότητα (capacity) ενός ολόκληρου συστήματος περιορίζεται από ένα ή από περιορισμένο αριθμό συστατικών/ών ή πόρου/ων του συστήματος αυτού. Για παράδειγμα, η στενωπή στην διαχείριση έργων (project management) είναι μία διαδικασία σε μια αλυσίδα διαδικασιών, τέτοια ώστε η περιορισμένη της χωρητικότητα (capacity) μειώνει την χωρητικότητα της συνολικής αλυσίδας. Η εύρεση λοιπόν τέτοιων στοιχείων είναι εξαιρετικού ενδιαφέροντος καθώς αποτελούν βασικά χαρακτηριστικά του συστήματος. Επιπλέον, ενδεχόμενη αντικατάστασή τους (αν είναι δυνατή) ή στοχευμένη αλλαγή στη δομή του συστήματος με σκοπό την καλύτερη κατανομή τους μπορεί να συντελέσει στη δραματική βελτίωση της απόδοτικότητας του συστήματος.

Σε αυτό το παράδειγμα επιδιώκεται η εύρεση της στενωπής ενός συστήματος, που στην περίπτωση μας είναι μία διαδικασία. Η διαδικασία δίνεται σε μορφή BPMN, που φαίνεται στο Σχήμα 6.7 και το χαρακτηριστικό που εξετάζεται είναι η ρυθμαπόδοση (throughput). Αν και τέτοιου είδους ανάλυση παραπέμπει στην ανάλυση των δικτύων (π.χ. στις Τηλεπικοινωνίες), εντούτοις έχει νόημα και στο χώρο των επιχειρηματικών διαδικασιών. Σε ένα υψηλό αφαιρετικό επίπεδο, οι επιχειρηματικές διαδικασίες μπορούν να θεωρηθούν συνθέσεις από δομοστοιχεία (modules). Κρίνεται λοιπόν σκόπιμη η εύρεση εκείνου ή εκείνων του/των δομοστοιχείου/ων που προκαλεί/ούν κωλύματα στην αποδοτική λειτουργία της.



Σχήμα 6.7: Σύστημα του Παραδείγματος (Δ)

Η ρυθμαπόδοση συνήθως εξαρτάται από :

- την αποδοτικότητα του βιομηχανικού κύκλου (Manufacturing Cycle Efficiency), που είναι ο οφέλιμος χρόνος παραγωγής προς το συνολικό χρόνο,
- την παραγωγικότητα της διαδικασίας (Process Productivity), δηλαδή το συνολικό αριθμό ή τη συνολική ποσότητα παραγόμενων μονάδων/αγαθών προς το συνολικό οφέλιμο χρόνο παραγωγής και
- την ποιότητα απόδοσης της διαδικασίας (Process Quality Yield) , που ορίζεται ως ο αριθμός ή η ποσότητα «σωστών» παραγόμενων μονάδων/αγαθών προς το συνολικό αριθμό ή τη συνολική ποσότητα παραγόμενων μονάδων/αγαθών.

Μία σχέση που μπορεί να χρησιμοποιηθεί για τον υπολογισμό της ρυθμαπόδοσης είναι η εξής:

$$\text{Throughput} = \text{Manufacturing Cycle Efficiency} \times \text{Process Productivity} \times \text{Process Quality Yield}$$

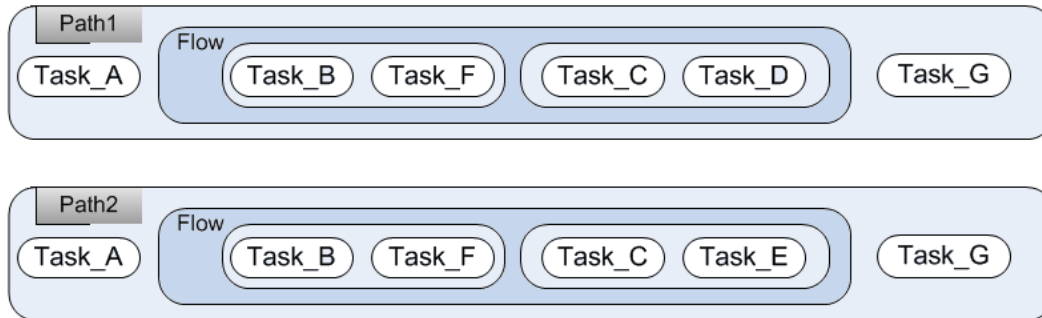
ροφανώς, οι τιμές καθενός εκ των Manufacturing Cycle Efficiency, Process Productivity και Process Quality Yield κυμαίνονται μεταξύ μηδέν και ένα (0,1) .

Για κάθε ατομική διαδικασία (Task), λοιπόν, η ρυθμαπόδοση προκύπτει ως το γινόμενο αυτών των χαρακτηριστικών της, τα οποία για τις ατομικές διαδικασίες του παραδείγματος παρουσιάζονται στον Πίνακα 6.9.

Tasks	Manufacturing Cycle Efficiency	Process Productivity	Process Quality Yield
Task_A	0.83	0.93	0.83
Task_B	0.85	0.92	0.78
Task_C	0.84	0.87	0.84
Task_D	0.77	0.92	0.83
Task_E	0.81	0.75	0.81
Task_F	0.88	0.78	0.79
Task_G	0.85	0.92	0.84

Πίνακας 6.9: Πίνακας Ρυθμαπόδοσης των ατομικών διαδικασιών

Εφαρμόζοντας τον αλγόριθμο που παρουσιάστηκε στο Κεφάλαιο 5 για την ανάλυση διαδρομών, προκύπτουν δύο εναλλακτικά μονοπάτια για τη συγκεκριμένη διαδικασία, όπως φαίνονται στο Σχήμα 6.8 , όπου τα μονοπάτια είναι στη μορφή Paths όπως παρουσιάστηκε στο Κεφάλαιο 5.



Σχήμα 6.8: Μονοπάτια Παραδείγματος (Δ)

Για κάθε διαφορετικό μονοπάτι θα βρούμε τη χωρητικότητα και τη στενωπή.

Όταν ένα σύστημα (στην περίπτωσή μας μια συνολική διαδικασία) αποτελείται από διαδικασίες που η μία έπεται της άλλης σειριακά, η συνολική ρυθμαπόδοση υπολογίζεται από το γινόμενο των επιμέρους ρυθμαποδόσεων των διαδικασιών. Για παράδειγμα, η συνολική ρυθμαπόδοση του συστήματος που φαίνεται στο Σχήμα 6.9 είναι ίση με:

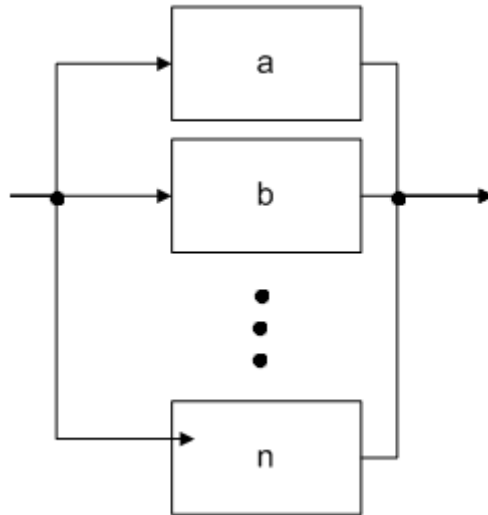
$$\text{Total_Throughput} = \text{Throughput}_a \times \text{Throughput}_b \times \dots \times \text{Throughput}_n .$$

Αντίθετα, όταν ένα σύστημα αποτελείται από διαδικασίες που εκτελούνται παράλληλα, η ρυθμαπόδοση του συστήματος ισούται με την ελάχιστη ρυθμαπόδοση εκ των διαδικασιών αυτών. Για παράδειγμα, η συνολική ρυθμαπόδοση του συστήματος που φαίνεται στο Σχήμα 6.10 είναι ίση με:

$$\text{Total_Throughput} = \min (\text{Throughput}_a, \text{Throughput}_b, \dots , \text{Throughput}_n) .$$



Σχήμα 6.9: Ακολουθία διαδικασιών εκτελούμενων σε σειρά (sequential processes)



Σχήμα 6.10: Ακολουθία διαδικασιών εκτελούμενων παράλληλα (parallel processes)

Με βάση λοιπόν αυτούς τους κανόνες εκλέγονται οι κατάλληλες στρατηγικές εκτίμησης από το σύστημα και προκύπτουν τα εξής αποτελέσματα (Πίνακας 6.10) :

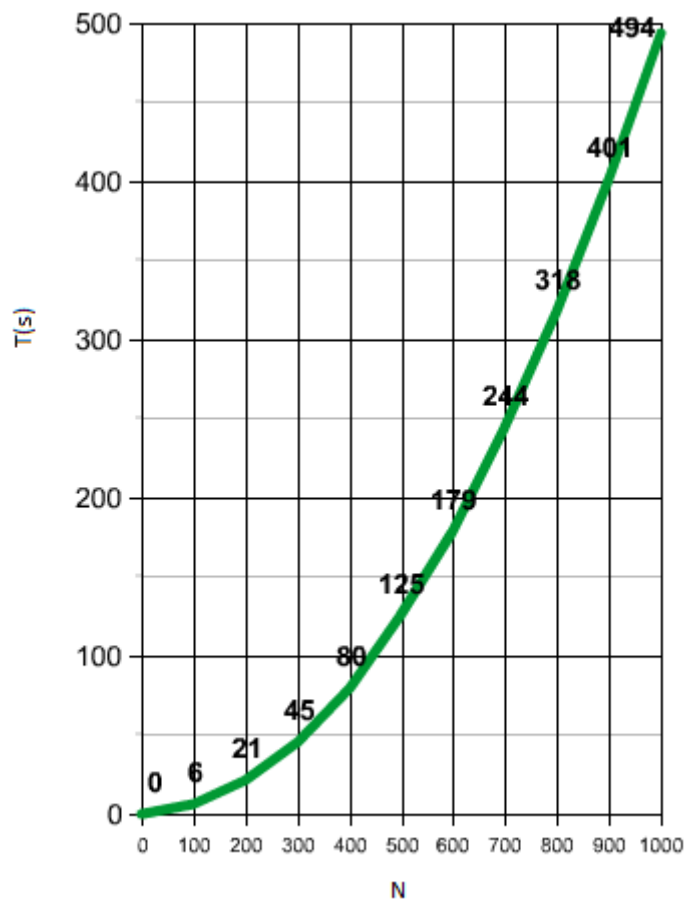
	Capacity	Bottleneck
Path1	0.139197	Task_F
Path2	0.127125	Task_E

Πίνακας 6.10: Αποτελέσματα για τη ρυθμιστική της διαδικασίας

6.4 Παράδειγμα (E) : Εξέταση Επίδοσης του Συστήματος

Στο παράδειγμα αυτό εξετάζεται η ταχύτητα απόκρισης του συστήματος σε συνάρτηση με το μέγεθος της υπό ανάλυση επιχειρηματικής διαδικασίας. Ως κριτήριο του μεγέθους αυτού θεωρούμε την πληθικότητα της διαδικασίας σε ατομικές διαδικασίες, δηλαδή διαδικασίες που επιδέχονται αντιστοίχισης με χαρακτηριστικά. Θεωρούμε λοιπόν μία τέτοια ατομική διαδικασία, το AtomicTask_A και μετρούμε το χρόνο απόκρισης του συστήματος όταν ως

είσοδο έχουμε μία σύνθετη διαδικασία αποτελούμενη από μεταβλητό πλήθος από συστατικά τύπου AtomicTask_A. Ενδεικτικά ζητούμε από το σύστημα τον υπολογισμό του συνολικού κόστους, έχοντας αποδόσει συγκεκριμένο κόστος στο AtomicTask_A.



Σχήμα 6.11: Διάγραμμα του χρόνου εκτέλεσης ως προς το μέγεθος της διαδικασίας.

Όπως φαίνεται και από το διάγραμμα (Σχήμα 6.11), όπου :

N :πληθικότητα της διαδικασίας σε ατομικές διαδικασίες,

T : χρόνος απόκρισης του συστήματος σε sec,

φαίνεται ότι ο χρόνος απόκρισης φαίνεται να αυξάνεται εκθετικά με το μέγεθος της εισόδου.

Να σημειωθεί ότι λόγω της δομής του μοντέλου των επιχειρηματικών διαδικασιών στο σύστημα, παρατηρήθηκε ότι αυτή η συμπεριφορά δεν επηρεάζεται σημαντικά από τη μορφή των διαδικασιών, παρά μόνο από το μέγεθός τους.

7

Επίλογος

Στο κεφάλαιο αυτό συνοψίζεται η παρουσίαση της παρούσας διπλωματικής εργασίας και προτείνονται ορισμένες μελλοντικές επεκτάσεις και πιθανές χρήσεις του συστήματος που αναπτύχθηκε, η υλοποίηση των οποίων κρίθηκε ότι θα είχε ενδιαφέρον.

7.1 Σύνοψη και συμπεράσματα

Η παρούσα διπλωματική εργασία είχε ως αντικείμενο την ανάπτυξη ενός συστήματος προσομοίωσης επιχειρηματικών διαδικασιών εμπλουτισμένων με ευέλικτα ποιοτικά χαρακτηριστικά και την προδιαγραφή μοντέλων για την αναπαράσταση τέτοιων χαρακτηριστικών. Με τη χρήση περιγραφικών, ευέλικτων και με ειδικότητα πεδίου (domain-specific) μοντέλων για την περιγραφή των χαρακτηριστικών των επιχειρηματικών διαδικασιών επιτυγχάνεται όντως η αποδοτικότερη και διεισδυτικότερη ανάλυση των επιχειρηματικών διαδικασιών, ενώ παράλληλα προωθείται η γεφύρωση του χάσματος μεταξύ των ανθρώπων των επιχειρήσεων, αφού τους παρέχεται η δυνατότητα έκφρασης του καθενός στη γλώσσα του. Οι αναλυτές και οι διευθυντές επιχειρήσεων μπορούν να απελευθερωθούν από τις περιορισμένες περιγραφές των χαρακτηριστικών των διαδικασιών που παρέχουν προς το παρόν τα περισσότερα συστήματα προσομοίωσης και ανάλυσης και εκτός από απλά πρωτογενή και τυποποιημένα χαρακτηριστικά όπως το κόστος και η χρονική διάρκεια, να ορίσουν όποια μετρήσιμα χαρακτηριστικά και δείκτες (KPIs) θεωρούν σημαντικά για κάθε ανάλυση που πραγματοποιούν. Παράλληλα, οι τεχνικοί προγραμματιστές μπορούν να κάνουν απευθείας απεικόνιση των μετρούμενων χαρακτηριστικών σε προσομοιώσιμη μορφή και αποδοτικό κώδικα χωρίς την απαίτηση ιδιαίτερων γνώσεων για τη φύση της κάθε διαδικασίας.

Επιπλέον, η επικέντρωση του συστήματος στην ευελιξία και την ποικιλομορφία ως προς την παραγωγή αποτελεσμάτων, παρέχοντας στους χρήστες τη δυνατότητα επιλογής μεταξύ μιας

ποικιλίας στρατηγικών και κριτηρίων αξιολόγησης, που μάλιστα μπορούν να τροποποιηθούν και να αντικατασταθούν κατά περίπτωση, προσφέρει ένα αποτελεσματικό εργαλείο για χρήση σε ένα μεγάλο εύρος περιπτώσεων ανάλυσης επιχειρηματικών διαδικασιών. Σε αυτό συντελεί και ο ορισμός ενός κοινού μοντέλου για την κανονικοποίηση σε αυτό όλων των επιχειρηματικών διαδικασιών, όποια κι αν είναι η αρχική μορφή αναπαράστασής τους. Η μορφή μάλιστα αυτού του μοντέλου επιτρέπει την πολυεπίπεδη ανάλυση των διαδικασιών και την αντιστοίχιση σε αυτές χαρακτηριστικών στο επιθυμητό κάθε φορά επίπεδο αφαίρεσης.

Εντούτοις δεν ήταν όλες οι διαπιστώσεις θετικές κατά την εκπόνηση της παρούσας εργασίας. Κατά την αναζήτηση κατάλληλου μοντέλου για την απεικόνιση των επιχειρηματικών διαδικασιών ανακαλύφθηκε ότι προς το παρόν υπάρχει ένα μεγάλο πλήθος διαφορετικών τρόπων και τυποποιήσεων για τη μοντελοποίηση διαδικασιών. Αν και πραγματοποιούνται αξιόλογες προσπάθειες από οργανισμούς, όπως ο OMG, για την επιβολή προτυποποίησης στις διάφορες λειτουργίες ανάλυσης των επιχειρηματικών διαδικασιών, σίγουρα θα περάσει αρκετός καιρός μέχρι την καθολική αποδοχή και χρήση τους, τη στιγμή μάλιστα που υπάρχουν και επικριτές τους που υποστηρίζουν τη χρησιμοποίηση διαφορετικών εργαλείων και προτύπων και σε πολλές περιπτώσεις ακόμη και τον ορισμό νέων, για την κατά περίπτωση ανάλυση διαδικασιών. Δυστυχώς, όπως διαπιστώθηκε κατά την υλοποίηση του συστήματος, ακόμη και αν τελικά επικρατήσουν καθολικά πρότυπα, η προσομοίωση όλων των διαδικασιών σε μια κοινή μορφή απαιτεί τα πρότυπα αυτά να είναι εξαιρετικά πολύπλοκα για να μπορούν να συμπεριλάβουν όλες τις δυνατότητες προσομοίωσης και εκφραστικότητας των σύγχρονων μέσων αναπαράστασης. Το γενικό μοντέλο για την αναπαράσταση διαδικασιών που ορίσαμε κατά την υλοποίηση του συστήματος είναι πολύ απλό και σίγουρα αδυνατεί να προσομοιάσει όλες τις εκφραστικές πτυχές των επιχειρηματικών διαδικασιών και της δομής τους. Παρόλα αυτά, παρατηρήθηκε ότι η αύξηση της εκφραστικότητας αυτού του μοντέλου προκαλεί εκθετική αύξηση της πολυπλοκότητας του συστήματος και απαιτεί πολλές τροποποιήσεις στη θεώρηση του συστήματος και στη χρήση των μοντέλων που αυτό επεξεργάζεται. Για το λόγο αυτό, αρκεστήκαμε στην υλοποίηση συστήματος με ενδεχόμενη πρακτική εφαρμογή σε διαδικασίες πολύ υψηλού επιπέδου και σχετικά απλής δομής. Τέλος, παρατηρήθηκε ότι απαιτείται πολλή προσπάθεια και προσοχή για τον ορισμό μοντέλων πλήρως συμβατών με πρότυπα όπως το MOF και για το λόγο αυτό περιοριστήκαμε στην εφαρμογή ενός υποσυνόλου των υποδείξεών του, με τη χρήση του πλαισίου EMF.

Αν και υπάρχουν ακόμα πολλά άλυτα ζητήματα στον τομέα της Διαχείρισης Επιχειρηματικών Διαδικασιών και στη μοντελοποίησή τους, τα πλεονεκτήματα αυτών των μεθόδων και οι πιέσεις λόγω των αυξημένων απαιτήσεων φαίνεται πως τελικά θα προκαλέσουν τη συνεχή εξέλιξη και προτυποποίησή τους με σκοπό την ακόμη μεγαλύτερη αποδοτικότητα και αυτοματοποίηση των επιχειρηματικών διαδικασιών. Βέβαια, απαραίτητη προϋπόθεση για την

πορεία προς αυτή την κατεύθυνση είναι να λυθούν θέματα όπως η συμβατότητα μεταξύ των διαφόρων τεχνολογιών, η λήψη αποφάσεων σχετικά με το επιθυμητό επίπεδο εκφραστικότητας εις βάρος της εκτελεσιμότητας και η αλλαγή στάσης των διαφόρων εταιριών, που προς το παρόν κρατούν μια στάση μυστικοπάθειας ως προς τα εργαλεία και τις τεχνολογίες που χρησιμοποιούν.

7.2 Μελλοντικές επεκτάσεις

Πιθανές και ίσως απαραίτητες μελλοντικές επεκτάσεις του συστήματος που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής είναι οι εξής:

- Επαύξηση του μοντέλου που χρησιμοποιεί το σύστημα για την αναπαράσταση επιχειρηματικών διαδικασιών, με σκοπό την αύξηση της εκφραστικότητάς του και τη δυνατότητα αναπαράστασης περισσότερων ειδών διαδικασιών και σε περισσότερα αφαιρετικά επίπεδα. Αν και το εγχείρημα αυτό ακούγεται απλό, εντούτοις, όπως προαναφέρθηκε στην παρούσα ενότητα, προκαλεί εκθετική αύξηση στην πολυπλοκότητα του συστήματος και απαιτήσεις για πλήθος αλλαγών όχι μόνο στην υλοποίηση αλλά ενδεχομένως και στο σχεδιασμό του συστήματος.
- Υλοποίηση πλήθους αναλυτικών αλγοριθμικών διαδικασιών για την αυτόματη μετατροπή από υπάρχοντα μοντέλα σημερινών τεχνολογιών, όπως BPMN και Control Flow Diagrams σε μορφή επεξεργάσιμη από το σύστημα. Για να έχουν νόημα τέτοιοι αλγόριθμοι, οι γλώσσες των μοντέλων αυτών θα πρέπει να ορίζονται τυπικά, σαφώς και μονοσήμαντα. Τέτοιου είδους προσπάθεια έχει γίνει για γλώσσες, όπως η BPMN, αλλά για πολλές από τις υπόλοιπες θα χρειαστεί κάτι παραπάνω από την απλή παρουσίαση ενός βασικού υποσυνόλου των γραφικών τους συμβόλων και προτάσεις για τη χρήση τους.
- Βελτίωση του ορισμού των μοντέλων του συστήματος, κάτι που μπορεί να επιτευχθεί με την αυστηρότερη συμμόρφωση των μοντέλων του συστήματος με το πρότυπο MOF. Για να γίνει αυτό θα πρέπει, ανεξάρτητα από τις ίσως περιορισμένες δυνατότητες που προσφέρει το πλαίσιο EMF, να οριστούν περαιτέρω συσχετίσεις μεταξύ των μοντέλων του συστήματος, ανεξαρτήτου γλώσσας διεπαφής, κανόνες για τη διάρκεια του κύκλου ζωής των μοντέλων, μια ιεραρχία από ανακλαστικές (reflective) διεπαφές κ.τ.λ..
- Επέκταση του αρχειακού συστήματος του συστήματος για τις ανάγκες αποθήκευσης και κατηγοριοποίησης μεγάλου όγκου δεδομένων, κάτι που είναι απαραίτητο στις σημερινές επιχειρήσεις με τη συνεχή επέκταση των υποδομών μηχανοργάνωσης. Θα μπορούσε το σύστημα να εξελιχθεί και να περιλαμβάνει και σχεσιακή λογική για

τα αρχεία του, κάτι που μπορεί να υλοποιηθεί με τις σύγχρονες τεχνολογίες βάσεων δεδομένων που υποστηρίζουν την αποθήκευση σειριοποιημένων δεδομένων.

- Δημιουργία ολοκληρωμένης διεπαφής χρήστη (user interface) βασισμένη στις ανάγκες για την εκάστοτε χρήση του συστήματος. Λόγω της ευελιξίας του συστήματος, μια τέτοια διεπαφή θα μπορούσε να παρέχει ένα μεγάλο πλήθος επιλογών στο χρήστη και με προσεχτική δουλειά θα μπορούσε να είναι αλληλεπιδραστική και φιλική προς το χρήστη.
- Μία πιθανή μελλοντική χρήση του συστήματος είναι η συμβολή στο Cloud Optimization όπως περιγράφεται και στο [44]. Με κατάλληλες προσαρμογές μπορεί να αξιολογεί τις διάφορες υπηρεσίες που εμπλέκονται και μάλιστα σε οποιοδήποτε από τα optimization layers, υποδεικνύοντας τους «αδύναμους κρίκους» του συστήματος και ελέγχοντας αν πληρούνται οι προδιαγραφές και οι περιορισμοί, όπως για το QoS.

8

Βιβλιογραφία

- [1] Colin Atkinson and Thomas Kühne, “Model-Driven Development: A Metamodeling Foundation.”
- [2] Adrian Rutle, “Diagram Predicate Framework A Formal Approach to MDE,” University of Bergen, Norway.
- [3] “Cambridge Dictionary Online: Free English Dictionary and Thesaurus - Cambridge University Press - Cambridge Dictionaries Online - Cambridge University Press”, available at: <http://dictionary.cambridge.org/>.
- [4] Jean Bézivin, “On the Unification Power of Models”, (INRIA & LINA) University of Nantes, available at: <http://atlanmod.emn.fr/www/papers/OnTheUnificationPowerOfModels.pdf>.
- [5] “Object Management Group - UML”, available at: <http://www.uml.org/>.
- [6] Brian Henderson-Sellers and Cesar Gonzalez-Perez, “The Rationale of Powertype-based Metamodelling to Underpin Software Development Methodologies”, University of Technology, Sydney.
- [7] “OMG Document -- omg/03-06-01 (MDA Guide V1.0.1)”, available at: <http://www.omg.org/cgi-bin/doc?omg/03-06-01>.
- [8] “MDA”, available at: <http://www.omg.org/mda/>.
- [9] “Object Management Group”, available at: <http://www.omg.org/>.
- [10] “OMG's MetaObject Facility (MOF) Home Page”, available at: <http://www.omg.org/mof/>.
- [11] “CORBA, XML and XMI Resource Page”, available at:

- <http://www.omg.org/technology/xml/>.
- [12] “OMG Modeling and Metadata Specifications”, available at:
http://www.omg.org/technology/documents/modeling_spec_catalog.htm.
- [13] “MetaObjectFacility(MOF) Specification,” Apr. 2002.
- [14] “Eclipse Modeling - EMF - Home”, available at: <http://www.eclipse.org/modeling/emf/>.
- [15] “Eclipse.org home”, available at: <http://www.eclipse.org/>.
- [16] “Interfaces”, available at:
http://java.sun.com/docs/books/jls/third_edition/html/interfaces.html.
- [17] “Extensible Markup Language (XML)”, available at: <http://www.w3.org/XML/>.
- [18] “W3C XML Schema”, available at: <http://www.w3.org/XML/Schema>.
- [19] Kien Huynh, “ANALYSIS THROUGH REFLECTION: WALKING THE EMF MODEL OF BPEL4WS”, York University, 2005.
- [20] Βασίλης Μ. Παπαδάκης, *Στρατηγική των Επιχειρήσεων Ελληνική και Διεθνής εμπειρία*, Εκδόσεις Ε. Μπένου, 2007.
- [21] “Home | KPI Library”, available at: <http://kpilibrary.com/home>.
- [22] W.M.P. van der Aalst, A.H.M. ter Hofstede, and M. Weske, “Business Process Management: A Survey.”
- [23] Ryan K.L. Ko, Stephen S.G. Lee, and Eng Wah Lee, “Business process management (BPM) standards: a survey”, Nanyang Technological University, Singapore.
- [24] “BPMN Information Home”, available at: <http://www.bpmn.org/>.
- [25] Francisco Curbera, Yaron Golan, Johannes Klein, Frank Leymann, Dieter Roller, Satish Thatte, and Sanjiva Weerawarana, “Business Process Execution Language for Web Services, Version 1.0”, Jul. 2002, available at: <http://www.ibm.com/developerworks/library/ws-bpel/>.
- [26] Tony Andrews, Francisco Curbera, Hitesh Dholakia, Yaron Golan, Johannes Klein, Frank Leymann, Kevin Liu, Dieter Roller, Doug Smith, Satish Thatte, Ivana Trickovic, and Sanjiva Weerawarana, “Business Process Execution Language for Web Services Version 1.1”, May. 2003, available at:
<http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>.
- [27] “Business Process Modeling Notation, V1.1”, Jan. 2008, available at:

- <http://www.omg.org/spec/BPMN/1.1/PDF>.
- [28] “Business Process Model and Notation (BPMN), V2.0”, Aug. 2009, available at:
<http://www.omg.org/spec/BPMN/2.0>.
- [29] “Business Process Management Initiative”, available at: <http://www.bpmi.org/>.
- [30] Stephen A. White, “Introduction to BPMN.”
- [31] “OASIS: Advancing open standards for the global information society”, available at:
<http://www.oasis-open.org/home/index.php>.
- [32] “Web Service Definition Language (WSDL)”, available at:
<http://www.w3.org/TR/wsdl>.
- [33] “SOAP Specifications”, available at: <http://www.w3.org/TR/soap/>.
- [34] Liang Chen, Bruno Wassermann, Wolfgang Emmerich, and Howard Foster, “Web Service Orchestration with BPEL”, Shanghai, China: ICSE’06, 2006.
- [35] Sebastian Breier, “Extended Data-flow Analysis on BPEL Processes”, University of Stuttgart, 2008.
- [36] “OASIS Web Services Business Process Execution Language (WS-BPEL) 2.0”, available at: <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-specification-draft.html>.
- [37] Stephen A. White, “Using BPMN to Model a BPEL Process”, Oct. 2010.
- [38] “COCOMO Software Cost Model”, available at:
<http://cost.jsc.nasa.gov/COCOMO.html>.
- [39] Βασίλης Μ. Παπαδάκης, *Στρατηγική των Επιχειρήσεων Ελληνική και Διεθνής εμπειρία*, Εκδόσεις Ε. Μπένου, 2002.
- [40] Αυγουστίνος Ι. Δημητράς και Απόστολος Α. Μπάλλας, “Παράδειγμα κατάρτισης γενικού προϋπολογισμού”, *Διοικητική Λογιστική για Προγραμματισμό και Έλεγχο*, 2009.
- [41] Martin Keen, Bryan Brown, Andy Garratt, Benjamin Käckenmeister, Ahmed Khairy, Kevin O’Mahony, and Lei Yu, “Building IBM Business Process Management Solutions Using WebSphere V7 and Business Space.”
- [42] Frederick Winslow Taylor, *The Principles of Scientific Management*, 1911.
- [43] Alexandra Poulouvassilis and Mark Levene, “A Nested-Graph Model for the Representation and Manipulation of Complex Objects”, King’s College.
- [44] Marin Litoiu, Murray Woodside, Johnny Wong, and Joanna Ng, Gabriel Iszlai, “A Business Driven Cloud Optimization Architecture,” York University, Canada.
- [45] Serge Mankovski, Kostas Kontogiannis, Hausi Müller, and Ken Wong, “Root Cause

Analysis and Diagnosis in SOA and Cloud Environments”.

- [46] Dennis Smith, Kostas Kontogiannis, and Marin Litoiu, “Impact of Multi-Organizational SOA Implementation on the Software Life Cycle”.