

Ανατροφοδότηση στη Συντήρηση των Υπηρεσιών Ιστού

Ιωάννης Ντίνος

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΕΞΕΙΔΙΚΕΥΣΗΣ



Ιωάννινα, Φεβρουάριος 2014

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA



ΒΙΒΛΙΟΘΗΚΗ
ΠΑΝΕΠΙΣΤΗΜΟΥ ΙΩΑΝΝΙΝΩΝ



026000336442



ΑΝΑΤΡΟΦΟΔΟΤΗΣΗ ΣΤΗ ΣΥΝΤΗΡΗΣΗ ΤΩΝ ΥΠΗΡΕΣΙΩΝ ΙΣΤΟΥ

Η
ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΕΞΕΙΔΙΚΕΥΣΗΣ

Υποβάλλεται στην

ορισθείσα από την Γενική Συνέλευση Ειδικής Σύνθεσης
του Τμήματος Μηχανικών Η/Υ και Πληροφορικής
Εξεταστική Επιτροπή

από τον

Ιωάννη Ντίνο

ως μέρος των Υποχρεώσεων

για τη λήψη

του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ

ΜΕ ΕΞΕΙΔΙΚΕΥΣΗ ΣΤΟ ΛΟΓΙΣΜΙΚΟ

Φεβρουάριος 2014



ΑΦΙΕΡΩΣΗ

Αφιερώνω αυτή την εργασία στην οικογένειά μου για την υποστήριξη τους όλα αυτά τα χρόνια που διήρκεσαν οι σπουδές μου.



ΕΥΧΑΡΙΣΤΙΕΣ

Χωρίς την παρουσία, την υποστήριξη και την ανεκτικότητα κάποιων ανθρώπων δε θα ήταν δυνατή η υλοποίηση της μεταπτυχιακής διατριβής.

Με το πέρας αυτής της διατριβής, θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα, κ. Απόστολο Ζάρρα, Επίκουρο Καθηγητή του Τμήματος Μηχανικών Η/Υ και Πληροφορικής του Πανεπιστημίου Ιωαννίνων, για την εποικοδομητική και αρμονική συνεργασία, καθώς και για το ενδιαφέρον που υπέδειξε, καθ' όλη την διάρκεια της εκπόνησής της.

Επίσης θα ήθελα να ευχαριστήσω τον κ. Παπαπέτρου Ευάγγελο και τον κ. Βασιλειάδη Παναγιώτη, Επίκουρους Καθηγητές του Τμήματος Πληροφορικής του Πανεπιστημίου Ιωαννίνων και μέλη της εξεταστικής επιτροπής.

Επίσης θέλω να ευχαριστήσω τους συναδέλφους μου από το χώρο εργασίας, που μου εξασφάλισαν τον απαραίτητο χρόνο για να διεκπεραιώσω τις υποχρεώσεις μου καθόλη την διάρκεια των μεταπτυχιακών σπουδών μου.

Ως πράξη φιλίας, θα ήθελα να ευχαριστήσω τον Κώστα Βόγλη για την υποστήριξη που μου πρόσφερε στην αρχή της μεταπτυχιακής εκπαιδευτικής μου διαδρομής.

Τέλος δεν θα μπορούσα να παραλείψω τους φίλους Κατερίνα και Γιώργο για την στήριξη και υπομονή που επέδειξαν, και παρέχοντας μου την ανιδιοτελή και ανυπολόγιστη υποστήριξή τους κατά τη διάρκεια δύσκολων στιγμών.



ΠΕΡΙΕΧΟΜΕΝΑ

	Σελ
ΑΦΙΕΡΩΣΗ	i
ΕΥΧΑΡΙΣΤΙΕΣ	ii
ΠΕΡΙΕΧΟΜΕΝΑ	iii
ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ	v
ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ	vi
ΠΕΡΙΛΗΨΗ	viii
ABSTRACT	x
ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ	1
ΚΕΦΑΛΑΙΟ 2. WEB SERVICES ΚΑΙ ΣΧΕΤΙΚΕΣ ΕΡΕΥΝΗΤΙΚΕΣ ΕΡΓΑΣΙΕΣ	5
2.1. Βασικές έννοιες των Web Services.	5
2.2. Πλεονεκτήματα των Web Services	6
2.3. Αρχιτεκτονική των Web Service	6
2.4. Τα πρότυπα που χρησιμοποιούν τα Web Services	8
2.4.1. Το πρωτόκολλο SOAP	8
2.4.2. Το UDDI	9
2.4.3. Περιγραφή των Web Services: WSDL	10
2.5. Σχετικές εργασίες	12
ΚΕΦΑΛΑΙΟ 3. ΕΡΓΑΛΕΙΟ ΑΝΑΛΥΣΗΣ WSDL ΑΡΧΕΙΩΝ	15
3.1. Περιγραφή Αρχιτεκτονικής	15
3.2. Υλοποίηση και Περιγραφή λειτουργιών	17
3.2.1. Πλατφόρμα Υλοποίησης	18
3.2.2. Λογισμικό Ανοιχτού Κώδικα Membrane Soa Model	18
3.2.3. Περιγραφή Λειτουργίας	19
3.2.4. Παρουσίαση εργαλείου	20
ΚΕΦΑΛΑΙΟ 4. ΠΕΡΙΓΡΑΦΗ ΥΠΗΡΕΣΙΩΝ AMAZON ΚΑΙ ΔΕΔΟΜΕΝΩΝ	27
4.1. Ορολογία, ορισμοί και συμβολισμοί	27
4.2. Περιγραφή υπηρεσιών και δεδομένων	30
4.2.1. Amazon Elastic Compute Cloud (Amazon EC2)	31
4.2.2. Amazon Elastic Load Balancing	33
4.2.3. Amazon AutoScalling	34
4.2.4. Amazon Simple Queue Service	35
4.2.5. Amazon Mechanical Turk Service	36
4.2.6. Amazon Relational Database Service	38
ΚΕΦΑΛΑΙΟ 5. ΑΞΙΟΛΟΓΗΣΗ ΤΩΝ ΝΟΜΩΝ ΤΟΥ LEHMAN	
ΣΤΗΝ ΕΞΕΛΙΞΗ ΤΩΝ ΥΠΗΡΕΣΙΩΝ ΙΣΤΟΥ	39
5.1. Ταξινόμηση των συστημάτων λογισμικού	40



5.2. Ορισμοί των Νόμων του Lehman	42
5.3. Αποτελέσματα μετρήσεων	57
5.3.1. Γραφήματα αλλαγής του συνολικού μεγέθους των σχημάτων ανά εξελικτικό βήμα.	57
5.3.2. Μετρήσεις των αλλαγών μεταξύ διαδοχικών εκδόσεων	59
5.3.3. Γραφήματα αριθμού αλλαγών ανά έκδοση σε σχέση με τα operation	62
5.3.4. Γραφήματα αριθμού αλλαγών ανά έκδοση σε σχέση με τα types (complex and simple types)	64
5.3.5. Μετρήσεις των αλλαγών του μεγέθους των εκδόσεων βάση των operations	66
5.3.6. Μετρήσεις των αλλαγών του μεγέθους των εκδόσεων βάση των types (Complex, Simple)	67
5.3.7. Μετρήσεις, portTypes, bindings, ports	68
5.3.8. Μετρήσεις ανάπτυξης (growth) των διαδοχικών εκδόσεων σε σχέση με τα operations	69
5.3.9. Μετρήσεις ανάπτυξης (growth) των διαδοχικών εκδόσεων σε σχέση με types	70
5.3.10. Μετρήσεις ρυθμού εργασίας (work rate)	72
5.3.11. Μετρήσεις Πολυπλοκότητας με βάση τις operations	74
5.3.12. Regression analysis at operations	75
5.3.13. Regression analysis at types	77
5.4. Εκτίμηση της ισχύος των νόμων του Lehman βάση των μετρήσεων για τα Amazon Web Services.	80
ΚΕΦΑΛΑΙΟ 6. ΣΥΜΠΕΡΑΣΜΑΤΑ	91
ΑΝΑΦΟΡΕΣ	92
ΠΑΡΑΡΤΗΜΑ	95



ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ

Πίνακας 5.1: Οι νόμοι της εξέλιξης λογισμικού	42
Πίνακας Π.6.1: Το σύνολο των εκδόσεων για την υπηρεσία EC2.	95
Πίνακας Π.6.2: Το σύνολο των εκδόσεων για την υπηρεσία Elastic Load Balancing	98
Πίνακας Π.6.3: Το σύνολο των εκδόσεων για την υπηρεσία Auto Scaling.	99
Πίνακας Π.6.4: Το σύνολο των εκδόσεων για την υπηρεσία Simple Queue Service	100
Πίνακας Π.6.5: Το σύνολο των εκδόσεων για την υπηρεσία Amazon Mechanical Turk	101
Πίνακας Π.6.6: Το σύνολο των εκδόσεων για την υπηρεσία Amazon Relational Database	103



ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

Σχήμα	Σελ
Σχήμα 2.1: Το μοντέλο των web service	7
Σχήμα 2.2: Δομή του SOAP	9
Σχήμα 2.3: Αρχείο περιγραφής WSDL	11
Σχήμα 2.4: Η δομή ενός WSDL	12
Σχήμα 3.1: Τα υποσυστήματα του εργαλείου	16
Σχήμα 3.2: Το παραγόμενο αρχείο Διαφορών (*.txt)	19
Σχήμα 3.3: Διεπαφή Start	21
Σχήμα 3.4: Διεπαφή Επιλογή Αρχείων	21
Σχήμα 3.5: Διεπαφή Information of selected WSDL files	22
Σχήμα 3.6: Διεπαφή Evolution in Version History	22
Σχήμα 3.7: Διεπαφή List of Evolution Step	23
Σχήμα 3.8: Διεπαφή Analysis of Differences	23
Σχήμα 3.9: Καρτέλα Αρχείο Διαφορών (Difference File)	24
Σχήμα 3.10: Καρτέλα Διαφορές των Λειτουργιών (Diffs of Operations)	24
Σχήμα 3.11: Καρτέλα Διαφορές των σύνθετων τύπων (Diffs of Complex Types)	25
Σχήμα 3.12: Καρτέλα Διαφορές των απλών τύπων (Diffs of Simple Elements)	25
Σχήμα 3.13: Καρτέλα Διαφορές των στοιχείων (Diffs of Schema Elements)	25
Σχήμα 3.14: Καρτέλα Αναφορά Διαφορών (Total Differences Report)	26
Σχήμα 3.15: Παράδειγμα Excel αρχείου	26
Σχήμα 4.1: Η υπολογιστική πλατφόρμα υπηρεσιών της Amazon.	30
Σχήμα 4.2 Total num changes for EC2 related on time	32
Σχήμα 4.3: Total num changes for elastic load balancing related on time	33
Σχήμα 4.4: Total num changes for AutoScaling related on time	34
Σχήμα 4.5: Total num changes for simple queue related at time	36
Σχήμα 4.6: Total num changes for Mechanical Turk	37
Σχήμα 4.7: Total num changes for Relational Database over time	38
Σχήμα 5.1: Συγκριτική παρουσίαση της εξέλιξης του συνολικού μεγέθους των υπηρεσιών ανά έκδοση	59
Σχήμα 5.2: Συγκριτική παρουσίαση της εξέλιξης των αλλαγών των υπηρεσιών ανά έκδοση	61
Σχήμα 5.3: Συγκριτική παρουσίαση της εξέλιξης των αλλαγών των υπηρεσιών σε σχέση με τα operations	63
Σχήμα 5.4: Συγκριτική παρουσίαση της εξέλιξης των αλλαγών των υπηρεσιών σε σχέση με τα types	65
Σχήμα 5.5: Συγκριτική παρουσίαση των αλλαγών του μεγέθους των υπηρεσιών σε σχέση με τα operations	66
Σχήμα 5.6: Συγκριτική παρουσίαση των αλλαγών του μεγέθους των υπηρεσιών σε σχέση με τα types	67



Σχήμα 5.7: Συγκριτική παρουσίαση των αλλαγών του μεγέθους των υπηρεσιών σε σχέση με τα portTypes,ports,services	68
Σχήμα 5.8: Μετρήσεις ανάπτυξης (growth) των διαδοχικών εκδόσεων σε σχέση με τα operation	70
Σχήμα 5.9: Μετρήσεις ανάπτυξης (growth) των διαδοχικών εκδόσεων σε σχέση με τα types	72
Σχήμα 5.10: Αλλαγές ανά ημέρα (work rate)	73
Σχήμα 5.11: Μετρήσεις Πολυπλοκότητας σε σχέση με τα operation	75
Σχήμα 5.12: Μετρήσεις Regression analysis σε σχέση με τα operation	77
Σχήμα 5.13: Μετρήσεις Regression analysis σε σχέση με τα types	79
Σχήμα 5.14: Διαφορετικά εμφανιζόμενα μοτίβα κατά την ανάπτυξη των υπηρεσιών	83
Σχήμα 5.15: Μοτίβα με υψηλές τιμές που ακολουθούν χαμηλές τιμές.	86



ΠΕΡΙΛΗΨΗ

Ιωάννης Ντίνος.

MSc. Τμήμα Μηχανικών Η/Υ και Πληροφορικής.

Πανεπιστήμιο Ιωαννίνων.

Φεβρουάριος 2014

Ανατροφοδότηση στη Συντήρηση των Υπηρεσιών Ιστού

Επιβλέπωντας: Απόστολος Ζάρρας

Η διαχείριση των μοναδικών και ρηξικέλευθων αλλαγών τόσο στην τεχνολογία όσο και στις επιχειρήσεις κατά την τελευταία δεκαετία έχει δημιουργήσει μια συνεχιζόμενη πρόκληση των υποδομών πληροφορικής με τις Υπηρεσίες Ιστού (Web Services) να αποτελούν την τελευταία επιτυχημένη αλλαγή στο τρόπο που αντιμετωπίζονται πλέον τα κατανεμημένα συστήματα και όπως όλα τα συστήματα λογισμικού υπόκεινται και αυτά σε συνεχές αλλαγές και αναβαθμίσεις.

Σύμφωνα με τα όσα γνωρίσουμε έως τώρα, η επιτυχής συντήρηση ενός λογισμικού πρέπει να βασίζεται σε μια συνεχή διαδικασία ανατροφοδότησης (Νόμοι του Lehman). Στη διαδικασία αυτή υπάρχουν μηχανισμοί θετικής ανατροφοδότησης, που οδηγούν στην αύξηση της λειτουργικότητας του λογισμικού και μηχανισμοί αρνητικής ανατροφοδότησης, που οδηγούν στη συντήρηση του λογισμικού και γενικότερα στον περιορισμό της ανεξέλεγκτης ανάπτυξης. Οι εν λόγω μηχανισμοί τροφοδοτούνται από διάφορες πηγές, όπως η χρήση του λογισμικού, το μάρκετινγκ, κ.α. Ο στόχος της παρούσης εργασίας μας είναι να εκτιμήσουμε το κατά πόσον τα παραπάνω ισχύουν στην περίπτωση των Υπηρεσιών Ιστού. Συγκεκριμένα, μελετάμε την ισχύ των Νόμων του Lehman στις Υπηρεσίες Ιστού που προσφέρονται από την Amazon.

Έχουμε διεξάγει ενδελεχή μελέτη σχετικά με την εξέλιξη των διαφόρων ιδιοτήτων συνολικά έξι υπηρεσιών χρησιμοποιώντας ένα εργαλείο λογισμικού που δημιουργήσαμε για αυτόν τον σκοπό. Παρουσιάζουμε μετρήσεις, για τα μεγέθη των υπηρεσιών και αναλυτικότερα σε σχέση με τις λειτουργίες, τους τύπους δεδομένων, και την ανάπτυξη αυτών ανά εξελικτικό βήμα. Αναλύουμε τη συμπεριφορά των υπηρεσιών βασισμένοι στις παρατηρήσεις μας στο κυρίαρχο χαρακτηριστικό που παρέχουν οι υπηρεσίες αυτές, τις λειτουργίες, και στις αλλαγές αυτών (προσθήκες, αφαιρέσεις, τροποποιήσεις), καθώς και στους τύπους δεδομένων που σχετίζονται με



τις λειτουργίες και αναφέρουμε τα αποτελέσματα σχετικά με την εγκυρότητα του κάθε νόμου με βάση τις εν λόγω παρατηρήσεις.



ABSTRACT

Ioannis Ntinios.

MSc. Department of Computer Engineering and Informatics.

University of Ioannina.

February 2014

Feedback on Maintenance of Web Services

Supervisor: Apostolos Zarras

The management of unique and innovative changes in both technology and business in the last decade has created an ongoing challenge of IT infrastructure with Web Services to represent the last successful change in the manner of distributed systems are treated. As all software systems, Web Services are also subjected to constant changes and upgrades.

According to what we know so far, successful maintenance of software should be based on a continuous feedback process (Laws of Lehman). In this process there are a positive feedback mechanism, leading to increased software functionality and a negative feedback mechanism, leading to maintenance of software, and generally to limit the uncontrolled growth. These mechanisms are fed from various sources, such as the use of software, marketing, etc. The objective of this study is to assess whether the same holds true in the case of Web Services. Specifically, we study the effect of Lehman's Laws on Web services offered by Amazon.

We have conducted a thorough study on the evolution of various properties, on six services using a software tool we created for this purpose. We present measurements on the size of services, and in more detail, in relation to the functions, data types, and the development of those per evolutionary step. We analyze the behavior of services based on our observations in the dominant characteristic such as functions, and there changes (additions, deletions, modifications), the types of data related to the functions and report the results on the validity of each law based on these observations.



ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ

Η εξέλιξη (evolution) σαν ευρύτερη έννοια ταυτίζεται με την αλλαγή και είναι διάχυτη σχεδόν σε όλες τις εκφάνσεις του πραγματικού κόσμου. Αποτελεί δε μια σταδιακή διαδικασία κατά την οποία κάτι αλλάζει σε κάτι διαφορετικό και συνήθως πιο σύνθετο ή σε καλύτερη μορφή. Οι πρώτες καταγεγραμμένες χρήσεις του όρου "Evolution" χρησιμοποιήθηκαν στα πλαίσια της βιολογικής επιστήμης, από τους ολλανδούς εντομολόγους Jan Schwammerdam το 1669 και Charles Bonnet στο 1740s, κατά την ανάπτυξη του εμβρύου (Richards 1992a, 1992b, Mayr 1982). [1]

Βασική συμπεριφορά των προγραμμάτων λογισμικού αποτελεί η εξέλιξη τους. Για την εξέλιξη λογισμικού (software evolution) ως όρος στερείται συγκεκριμένης ερμηνείας και για αρκετούς ερευνητές του χώρου ο όρος αποτελεί συνώνυμο της «συντήρησης λογισμικού (software maintance). Ένας ορισμός του όρου έχει αποδοθεί από τον οργανισμό IEEE (Institute of Electrical and Electronics Engineers) ως *«Modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment»*[2] (μτφ) «η τροποποίηση ενός προϊόντος λογισμικού μετά την παράδοση του για να διορθωθούν τα σφάλματα του, να βελτιωθούν οι επιδόσεις του ή άλλα χαρακτηριστικά του, ή να προσαρμοστεί το προϊόν σε ένα τροποποιημένο περιβάλλον».

Το 1969 ο Meir M. Lehman έκανε μια εμπειρική μελέτη για την IBM, με σκοπό την βελτίωση της αποτελεσματικότητας του προγραμματισμού της εταιρείας. Η μελέτη έλαβε λίγη προσοχή στην εταιρεία και δεν είχε καμία επίδραση στις πρακτικές ανάπτυξης της. Η μελέτη αυτή, ωστόσο, ήταν η απαρχή για έναν νέο και παραγωγικό τομέα της έρευνας στην «Εξέλιξη του λογισμικού» που ασχολείται με τη διαδικασία με την οποία τα προγράμματα και οι τροποποιήσεις τους προσαρμόζονται στο μεταβαλλόμενο περιβάλλον τους. Με βάση τα εμπειρικά αποτελέσματα από την



μελέτη του. και με στόχο την εξεύρεση αμετάβλητων ιδιοτήτων που πρέπει να τηρούνται για ολόκληρες κατηγορίες έργων ανάπτυξης λογισμικού κατέληξε στην διατύπωση των οκτώ νόμων που είναι γνωστοί στη βιβλιογραφία ως «Εξελικτικοί Νόμοι του Lehman » (Software Evolution Laws of Lehman) που περιγράφηκαν για πρώτη φορά από τον ίδιο στις αρχές της δεκαετίας του 70 [3][4].

Η διαχείριση των μοναδικών και ρηξικέλευθων αλλαγών τόσο στην τεχνολογία όσο και στις επιχειρήσεις κατά την τελευταία δεκαετία έχει δημιουργήσει μια συνεχιζόμενη πρόκληση των υποδομών πληροφορικής. Τα τελευταία χρόνια, η χαρακτηριστική αρχιτεκτονική των εφαρμογών των επιχειρήσεων έχει εξελιχθεί από μια κεντρική εγκατάσταση με έναν επιτραπέζιο υπολογιστή, σε λύσεις client/server, και τώρα σε χαλαρά συνδεδεμένες υπηρεσίες ιστού (Web Services). Υπάρχουν πολλοί ορισμοί για το τι είναι Web Services, περίπου όσοι και οι εταιρίες πληροφορικής που αναπτύσσουν εργαλεία για τα web services. Ένας πολύ καλός ορισμός έρχεται από την IBM: Τα Web Services είναι μια τεχνολογία που επιτρέπει στις εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως πλατφόρμας και γλώσσας προγραμματισμού. Το web service είναι μια διεπαφή λογισμικού (software interface) που περιγράφει μια συλλογή από λειτουργίες οι οποίες μπορούν να προσεγγιστούν από το δίκτυο μέσω πρότυπων μηνυμάτων XML. Χρησιμοποιεί πρότυπα βασισμένα στη γλώσσα XML για να περιγράψει μία λειτουργία (operation) προς εκτέλεση και τα δεδομένα προς ανταλλαγή με κάποια άλλη εφαρμογή[5].

Οι Web Services τεχνολογίες αποτελούν την τελευταία επιτυχημένη αλλαγή στο τρόπο που αντιμετωπίζονται πλέον τα κατανεμημένα συστήματα και όπως όλα τα συστήματα λογισμικού υπόκεινται και αυτά σε συνεχή αλλαγές και αναβαθμίσεις. Με τον όρο «Web Service Evolution» εννοούμε μια πειθαρχημένη προσέγγιση για τη διαχείριση των αλλαγών στις Web Services. Η εξέλιξη ενός Web Service εκφράζεται μέσω της δημιουργίας, της αναβάθμισης, ή της κατάργησης διαφορετικών παραλλαγών της που ονομάζονται εκδόσεις (versions) κατά την διάρκεια της ζωής του. Αυτές οι εκδόσεις πρέπει να ευθυγραμμίζονται μεταξύ τους με τέτοιο τρόπο ώστε να επιτρέπουν στους χρηστές της υπηρεσίας αυτής να παρακολουθούν τις διάφορες τροποποιήσεις που εισάγονται με το χρόνο και τα αποτελέσματά τους σχετικά με την αρχική υπηρεσία καθώς και να γνωρίζουν γιατί η αλλαγή έγινε, τι επιπτώσεις έχει. και αν η νέα έκδοση του service είναι συνεπής και δεν το καθιστά πλέον ακατάλληλο για την χρήση που είχε προεπιλεγεί[6].



Μέχρι το 2005, η Amazon είχε περάσει πάνω από μια δεκαετία επενδύοντας εκατομμύρια δολάρια στην οικοδόμηση και διαχείριση μεγάλης κλίμακας, αξιόπιστων και αποτελεσματικών πληροφοριακών και τεχνολογικών υποδομών που υποστήριζαν μια από τις μεγαλύτερες online πλατφόρμες λιανικής πώλησης στον κόσμο. Η Amazon ξεκίνησε τα Amazon Web Services (AWS), έτσι ώστε και άλλες οργανώσεις ή επιχειρήσεις να μπορούν να επωφεληθούν από την εμπειρία και τις επενδύσεις της στην λειτουργία μιας μεγάλης κλίμακας κατανεμημένης υποδομής. Τα AWS λειτουργούν από το 2006, και σήμερα εξυπηρετούν εκατοντάδες χιλιάδες πελάτες σε όλο τον κόσμο. Σήμερα η Amazon.com τρέχει μια παγκόσμια πλατφόρμα που εξυπηρετεί εκατομμύρια πελατών και διαχειρίζεται εμπορικές συναλλαγές αξίας δισεκατομμυρίων δολαρίων.

Ο στόχος της παρούσης εργασίας μας είναι να εκτιμήσουμε τη δυνατότητα εφαρμογής των νόμων του Lehman στα προσφερόμενα Web Services της Amazon. Έχουμε διεξάγει ενδελεχή μελέτη σχετικά με την εξέλιξη των διαφόρων ιδιοτήτων συνολικά έξι υπηρεσιών χρησιμοποιώντας ένα εργαλείο λογισμικού που δημιουργήσαμε για αυτόν τον σκοπό. Αυτές οι ιδιότητες περιλαμβάνουν το μέγεθος, την ανάπτυξη, καθώς εντοπίζουμε και τις αλλαγές ανά έκδοση, τόσο όσον αφορά τις λειτουργίες (operations) όσο και τους τύπους δεδομένων τους.

Με την εργασία αυτή στοχεύουμε να παρουσιάσουμε

- Την πρώτη ουσιαστική μελέτη για τις υπηρεσίες μιας από τις μεγαλύτερες εταιρίες του Διαδικτύου με σκοπό την εξαγωγή συμπερασμάτων χρήσιμων προς την επιστημονική κοινότητα και τους χρηστές αυτών των υπηρεσιών
- Δημιουργήσαμε ένα εργαλείο Λογισμικού το οποίο εξάγει αναλυτικές πληροφορίες για την δομή των WSDL αρχείων των εξεταζόμενων υπηρεσιών
- Έχουμε ολοκληρώσει τη συλλογή, κατηγοριοποίηση και την επεξεργασία των αρχείων WSDL που περιγράφουν τις νέες εκδόσεις των διαφορετικών services για τις έξι περιπτώσεις που έχουμε μελετήσει και έχουμε καταλήξει με τα αντίστοιχα σύνολα δεδομένων που χρησιμοποιούνται στην μελέτη μας.
- Αναλύουμε τη συμπεριφορά των υπηρεσιών βασισμένοι στις παρατηρήσεις μας στο κυρίαρχο χαρακτηριστικό που παρέχουν οι υπηρεσίες αυτές, δηλαδή τις



λειτουργίες (operations), και στις αλλαγές αυτών (προσθήκες, αφαιρέσεις, τροποποιήσεις), καθώς και στους τύπους δεδομένων που σχετίζονται με τις λειτουργίες.

- Παρουσιάζουμε μετρήσεις, για τα μεγέθη των υπηρεσιών και αναλυτικότερα σε σχέση με τις λειτουργίες και τους τύπους δεδομένων, και την ανάπτυξη αυτών ανά εξελικτικό βήμα.
- Αναλύουμε τους νόμους του Lehman σε βάθος και χρησιμοποιούμε συγκεκριμένες μετρικές για να επαληθεύουμε την ύπαρξη τους στην εξελικτική διαδικασία των υπηρεσιών.

Δομή της Διατριβής. Στο 2^ο κεφάλαιο παρουσιάζουμε α) μια σύντομη περιγραφή της λειτουργίας των Web Services και β) σχετικές εργασίες που έχουν δημοσιευθεί στην ανάπτυξη λογισμικού σε Web Services. Στο 3^ο κεφάλαιο παρουσιάζουμε το εργαλείο λογισμικού που δημιουργήσαμε για την εξαγωγή των απαραίτητων μετρήσεων. Στο 4^ο κεφάλαιο παρουσιάζουμε τις υπηρεσίες της Amazon που εξετάζουμε μαζί με το σύνολο δεδομένων που συλλέξαμε και των μετρήσεων που πρόέκυψαν. Στο 5^ο κεφάλαιο αναλύουμε τους νόμους του Lehman, τις μετρικές που σχετίζονται με αυτούς, και ανιχνεύουμε μέσα στις μετρήσεις μας για τις απαραίτητες ενδείξεις που θα μας οδηγήσουν σε ασφαλή συμπεράσματα. Στο 6^ο κεφάλαιο καταγράφουμε τα τελικά συμπεράσματα μας.



ΚΕΦΑΛΑΙΟ 2. WEB SERVICES ΚΑΙ ΣΧΕΤΙΚΕΣ ΕΡΕΥΝΗΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

-
- 2.1 Βασικές έννοιες των Web Services
 - 2.2 Πλεονεκτήματα των Web Services
 - 2.3 Αρχιτεκτονική των Web Service
 - 2.4 Τα πρότυπα που χρησιμοποιούν οι Web Services
 - 2.5 Σχετικές ερευνητικές εργασίες
-

2.1. Βασικές έννοιες των Web Services

Τα Web Services είναι εφαρμογές client / server που επικοινωνούν μέσω του Παγκοσμίου Ιστού και του πρωτόκολλου μεταφοράς HTTP (Hypertext transfer protocol). Όπως περιγράφεται από την Κοινοπραξία του Παγκόσμιου Ιστού (W3C), οι υπηρεσίες αυτές παρέχουν ένα πρότυπο τρόπο διαλειτουργίας των εφαρμογών λογισμικού ανεξάρτητα από πλατφόρμες και πλαίσια στα οποία αυτές έχουν δημιουργηθεί. Όπως κάθε εφαρμογή έτσι και αυτές που βασίζονται σε Web Service μπορούν να εκτελέσουν πλήθος λειτουργιών. Οι υπηρεσίες αυτές μπορεί να συνδυαστούν για να επιτευχθούν πολύπλοκες λειτουργίες καθώς επίσης μπορούν και να αλληλεπιδρούν μεταξύ τους για την παροχή εξελιγμένων υπηρεσιών. Ενώ οι browser-based εφαρμογές του διαδικτύου λειτουργούν με βάση την αναπαράσταση των δεδομένων στους τελικούς χρηστές τα Web Services εκτός από την παρουσίαση των δεδομένων επιτρέπουν να χρησιμοποιούν οι χρήστες και προγραμματιστικά το δίκτυο για να προσπελάσουν τις υπηρεσίες αυτές και να χρησιμοποιήσουν την λειτουργικότητα τους προς όφελος τους. Ίσως η βασικότερη αιτία της παγκόσμιας υιοθέτησης τους να αποτελεί το γεγονός ότι η διαλειτουργικότητα που παρουσιάζουν, καθώς επίσης και το ότι οι υπηρεσίες αυτές μειώνουν δραστικά το λειτουργικό



κόστος των επιχειρήσεων αφού αυτές είναι σε θέση να επεκτείνουν ή να επαναχρησιμοποιήσουν την υπάρχουσα λειτουργικότητα των συστημάτων τους.

2.2. Πλεονεκτήματα των Web Services

Βασικά χαρακτηριστικά των Web Services είναι:

- Η διαλειτουργικότητα μεταξύ ετερογενών περιβαλλόντων που επιτρέπει σε διαφορετικά κατανεμημένα συστήματα και υπηρεσίες να τρέχουν σε διαφορετικές αρχιτεκτονικές και λειτουργικά συστήματα και επίσης να επιτρέπει την υλοποίηση αυτών των συστημάτων σε διαφορετικές γλώσσες προγραμματισμού [5].
- Περιγράφουν τις διεπαφές (interfaces) τους με αρκετή λεπτομέρεια ώστε να έχει ο χρήστης τις απαραίτητες πληροφορίες να χτίσει μια εφαρμογή η οποία να επικοινωνήσει μαζί τους. Η περιγραφή συνήθως παρέχεται σε ένα έγγραφο XML το οποίο ονομάζεται έγγραφο WSDL (Web Service Description Language) [5].
- Απλότητα υποδομής: Τα Web Services λειτουργούν με πρότυπες γλώσσες και πρωτόκολλα όπως η XML, το HTTP και το TCP/IP. Η πλειονότητα των εταιριών έχουν ήδη την δικτυακή υποδομή και τους ανθρώπους με γνώσεις και εμπειρία που τη συντηρούν. Έτσι το κόστος για την εφαρμογή των Web Services είναι σημαντικά μικρότερο από αυτό των προηγούμενων τεχνολογιών [8].
- Τα Web Services χρησιμοποιούν ως πρωτόκολλο επικοινωνίας το SOAP. Το πρωτόκολλο αυτό είναι πολύ πιο απλό από πρωτόκολλα παλαιότερων τεχνολογιών όπως αυτά που χρησιμοποιούνταν από τα κατανεμημένα περιβάλλοντα CORBA, DCOM, RPC [8].

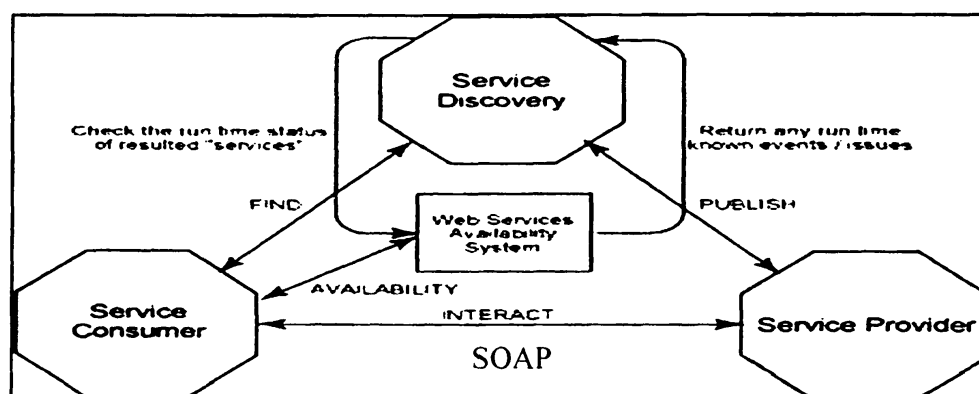
2.3. Αρχιτεκτονική των Web Services

Σήμερα τα προγραμματιστικά περιβάλλοντα, επιχειρησιακά και μη, μπορεί να είναι ιδιαίτερα πολύπλοκα λόγω της χρήσης ποικιλίας λογισμικού και υλικού, της διαδικτυακής κατανεμημένης πληροφορίας, ή της ανάγκης συνδυασμένης λειτουργίας των εφαρμογών. Η αρχιτεκτονική των Web Services βασίζεται πάνω σε ένα πρότυπο



αρχιτεκτονικής (Service-Oriented Architecture) το οποίο τις θεωρεί σαν επαναχρησιμοποιούμενες οντότητες. Όταν μιλάμε λοιπόν για μία αρχιτεκτονική προσανατολισμένη στις υπηρεσίες προκύπτουν ορισμένα ζητήματα. Η εφαρμογή που παρέχει την υπηρεσία και η εφαρμογή-πελάτης η οποία χρησιμοποιεί την υπηρεσία πρέπει να μιλάνε μεταξύ τους σε μια κοινή γλώσσα. Έπειτα οι δύο εφαρμογές χρειάζονται ένα τρόπο να εντοπίζουν η μία την άλλη πριν ξεκινήσουν να μιλούν μεταξύ τους. Ως εκ τούτου, μπορούμε να πούμε ότι μια βασική αρχιτεκτονική για Web Services πρέπει να παρέχει

- Ένα μηχανισμό που επιτρέπει στους χρήστες-πελάτες να προσπελάσουν μια υπηρεσία και έναν κατάλογο
- Ένα μηχανισμό που επιτρέπει σε διαφορετικές υπηρεσίες να καταγράψουν την ύπαρξη τους σε ένα κατάλογο και από την άλλη οι πελάτες να μπορούν να αναζητούν από τον κατάλογο αυτό τις διαθέσιμες υπηρεσίες προς αξιοποίηση.
- Ένα μηχανισμό μέσω του οποίου οι υπηρεσίες να μπορούν να εκθέτουν τις διεπαφές και οι πελάτες να είναι σε θέση να τις προσπελαίνουν.



Σχήμα 2.1: Το μοντέλο των web service (πηγή <http://www.ibm.com/developerworks/library/>)

Το μοντέλο των Web Services ακολουθεί το παράδειγμα *δημοσίευση (publish)*, *εύρεση (find)* και *σύνδεση (bind)*. Στο πρώτο βήμα, ο προμηθευτής της υπηρεσίας δημοσιεύει την υπηρεσία σε ένα κατάλογο υπηρεσιών. Στο δεύτερο βήμα, ο πελάτης ο οποίος ψάχνει για μία υπηρεσία η οποία να καλύπτει τις απαιτήσεις του, την αναζητεί στον κατάλογο. Αφού επιτυχημένα βρει πολλαπλές υπηρεσίες επιλέγει μία βάσει των προτιμήσεών του. Τότε μεταφορτώνει την περιγραφή της υπηρεσίας και συνδέεται με αυτήν ώστε να μπορέσει να καλέσει και να εκτελέσει την υπηρεσία. [9]



2.4. Τα πρότυπα που χρησιμοποιούν τα Web Services

Τα Web Services χρησιμοποιούν διαφορές τεχνολογίες που κυρίως βασίζονται στην XML για να μεταφέρουν ή να μετατρέψουν πληροφορία κατά την διάρκεια που αλληλεπιδρούν με διάφορα υπολογιστικά περιβάλλοντα.

2.4.1. Το πρωτόκολλο SOAP

Το πρωτόκολλο αυτό αρχικά σχεδιάστηκε από μηχανικούς της Microsoft το 1998 και ο οργανισμός W3C μετά από πρόταση εταιριών λογισμικού όπως οι IBM, HP, Microsoft κ.α. έχει συντάξει μία σύσταση (recommendation) η οποία είναι ότι πιο κοντινό υπάρχει στην προδιαγραφή (specification) του Simple Object Access Protocol (SOAP). Μέσα στη σύσταση αυτή βρίσκεται ο παρακάτω ορισμός: SOAP είναι ένα ελαφρύ πρωτόκολλο που προορίζεται για την ανταλλαγή δομημένης πληροφορίας (μηνυμάτων) σε ένα αποκεντρωμένο, κατανεμημένο περιβάλλον. Χρησιμοποιεί τεχνολογίες XML για να καθορίσει ένα επεκτάσιμο πλαίσιο μηνυμάτων και παρέχει δομημένα μηνύματα που μπορούν να ανταλλάσσονται σε μια ποικιλία υποκείμενων πρωτοκόλλων. Το πρωτόκολλο αυτό έχει σχεδιαστεί ώστε να είναι ανεξάρτητο από συγκεκριμένα μοντέλα προγραμματισμού και διαφορετικές υλοποιήσεις υποδομών δικτύου [10].

Το SOAP είναι ουσιαστικά ένας μηχανισμός μέσω του οποίου τα Web Services μπορούν να διασυνδεθούν αφού βασισμένο στην διαδεδομένη πλέον περιγραφική γλώσσα XML παρέχει την ικανότητα σε εφαρμογές που τρέχουν σε διαφορετικά λειτουργικά συστήματα με διαφορετικές γλώσσες προγραμματισμού και διαφορετικές τεχνολογίες να επικοινωνήσουν. Χρησιμοποιεί τα πλέον δικτυακά διαδεδομένα πρωτόκολλα HTTP και SMTP για την μεταφορά των μηνυμάτων, εξαλείφοντας προβλήματα συμβατότητας και ασφάλειας που υπήρχαν με προηγούμενα μοντέλα (RBA, DCOM).

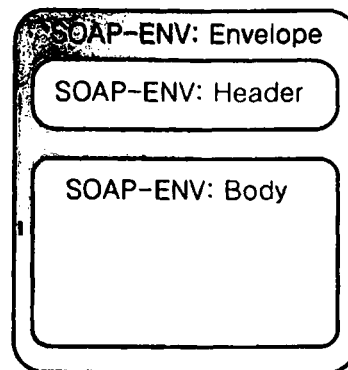


Η δομή του ακολουθεί:

Φάκελος (Envelope) – προσδιορίζει τι είναι μέσα στο μήνυμα και πως πρέπει να το διαχειριστεί κάποιος.

Επικεφαλίδα (Header) – προσδιορίζει πληροφορίες που αφορούν τη συγκεκριμένη υπηρεσία.

Σώμα μηνύματος (Body) – περιέχει το κυρίως μήνυμα προς επεξεργασία.



Σχήμα 2.2: Δομή του SOAP

2.4.2. Το UDDI

Ένα από τα βασικά συστατικά μιας service-oriented αρχιτεκτονικής είναι ένας μηχανισμός για την περιγραφή των Web Services ώστε να μπορούν να ανακαλυφθούν από πιθανούς πελάτες. Για να υλοποιηθεί το συστατικό αυτό, χρειάζεται ένας κεντρικός κατάλογος για να φιλοξενήσει τις περιγραφές των υπηρεσιών. Ο μηχανισμός UDDI αποτελεί ένα κατάλογο με καταχωρήσεις που συνδυάζουν πληροφορίες α) για τις επιχειρήσεις που το χρησιμοποιούν β) τις υπηρεσίες που προσφέρουν οι επιχειρήσεις, γ) και τα πρότυπα επικοινωνίας και τις διεπαφές που προσφέρουν οι υπηρεσίες.

Υπάρχουν 2 βασικοί τύποι UDDI καταλογών: οι δημόσιοι και οι ιδιωτικοί. Ο δημόσιος αναφέρεται και σαν UDDI business Registry στον οποίο οι υπηρεσίες της οποιασδήποτε εταιρείας μπορούν να καταχωρηθούν. Κατά την αρχική δημιουργία του, αυτός έτυχε ευρείας αποδοχής στην πορεία όμως δεν κατέληξε να χρησιμοποιείται ευρέως καθώς δεν παρείχε τα κατάλληλα επίπεδα ασφάλειας. Αποτέλεσμα αυτής της τάσης ήταν το ενδιαφέρον να επικεντρωθεί στους ιδιωτικούς καταλόγους που φιλοξενούνται στο διαδίκτυο ή σε ένα εσωτερικό δίκτυο (π.χ. μια εταιρεία μπορεί να διατηρεί ένα κατάλογο με υπηρεσίες που χρησιμοποιεί εσωτερικά για δικούς της συγκεκριμένους σκοπούς και να τον δημοσιοποιήσει για οικονομικό όφελος).

Η μορφοποίηση των καταχωρήσεων του UDDI καταλόγου αποτελείται από τρία τμήματα:

- **White pages:** Παρέχουν πληροφορίες σχετικά με την επιχείρηση που παρέχει την υπηρεσία. Αυτό περιλαμβάνει το όνομα της επιχείρησης και την περιγραφή της επιχείρησης - ενδεχομένως σε πολλές γλώσσες.
- **Yellow pages:** περιγράφουν με περισσότερη λεπτομέρεια τις επιχειρήσεις και μπορούν να παρέχουν μια ταξινόμηση των υπηρεσιών ή των επιχειρήσεων, με βάση πρότυπα ταξινομήσεων (π.χ. Standard Industrial Classification-SIC), ή γεωγραφικών κριτηρίων.
- **Green pages:** περιγράφουν πληροφορίες για το πώς να αποκτήσει κανείς πρόσβαση σε μια υπηρεσία Web, όπως η διεύθυνση της υπηρεσίας και των παραμέτρων, καθώς και αναφορές στις προδιαγραφές των διεπαφών. Επειδή μια υπηρεσία Web μπορεί να έχει πολλές συνδέσεις (όπως ορίζεται από την WSDL περιγραφή του), μια υπηρεσία μπορεί να έχει πολλαπλές Green pages, ώστε κάθε σύνδεση (binding) που αντιστοιχεί σε καθεμία από αυτές να προσεγγιστεί με διαφορετικό τρόπο.

Ο κατάλογος UDDI δεν εμπίπτει στα πλαίσια της εργασίας αυτής και ο ενδιαφερόμενος αναγνώστης παραπέμπεται για περεταίρω ανάλυση στην διεθνή βιβλιογραφία.

2.4.3. Περιγραφή των Web Services: WSDL

Η Web Services Description Language (WSDL) αποτελεί κομβικό κομμάτι του μηχανισμού των Web Services. Η WSDL δημιουργήθηκε το 2000 από την συνεργασία των Microsoft, IBM, ARIBA και καταχωρήθηκε στο W3C το 2001 (WSDL 1.0) και αποτελεί πρότυπο βασισμένο στην XML προκειμένου να περιγράψει τα Web Services. Η τελευταία του έκδοση είναι η WSDL 2.0 με καταχώρηση στο W3C το 2007 και αρκετές αλλαγές από την αρχική έκδοση.

Η WSDL χρησιμοποιείται για να περιγράψει *τί* μπορεί να κάνει ένα Web Service, *πού* βρίσκεται και *πώς* να το καλέσει κανείς και παρέχει ένα τρόπο στους παρόχους υπηρεσιών να περιγράψουν τη βασική μορφή των αιτήσεων και απαντήσεων των υπηρεσιών πάνω από διαφορετικά πρωτόκολλα και κωδικοποιήσεις.



```

<definitions targetNamespace="http://ec2.amazonaws.com/doc/2006-06-26/">
  <types>
    <xs:schema targetNamespace="http://ec2.amazonaws.com/doc/2006-06-26/"
      elementFormDefault="qualified">
      <xs:element name="RegisterImage" type="tns:RegisterImageType"/>
      <xs:complexType name="RegisterImageType">
      <xs:sequence>
        <xs:element name="imageLocation" type="xs:string"/>
      </xs:sequence>
      </xs:complexType>
    </types>
    ..... message definitions
    <message name="RegisterImageRequestMsg">
      <part name="RegisterImageRequestMsgReq" element="tns:RegisterImage"/>
    </message>
    </message>
    <portType name="AmazonEC2PortType">
      <operation name="RegisterImage">
        <input message="tns:RegisterImageRequestMsg"/>
        <output message="tns:RegisterImageResponseMsg"/>
      </operation>
    </portType>
    <binding name="AmazonEC2Binding" type="tns:AmazonEC2PortType">
      <soap:binding style="document">
      .....
      </binding>
    <service name="AmazonEC2">
      <port name="AmazonEC2Port" binding="tns:AmazonEC2Binding">
        <soap:address location="https://ec2.amazonaws.com"/>
      </port>
    </service>
  </definitions>

```

Σχήμα 2.3: Αρχείο περιγραφής WSDL

Ένα WSDL αρχείο (Σχήμα 2.3) χαρακτηρίζεται από τα εξής στοιχεία:

Definitions - είναι το πρώτο στοιχείο ενός WSDL αρχείου. Ορίζει το όνομα του service και δηλώνονται όλα τα απαραίτητα namespaces που χρησιμοποιούνται στο αρχείο. Αποτελεί το εξωτερικό περίβλημα και όλα τα υπόλοιπα στοιχεία που περιλαμβάνονται μέσα σε αυτό.

Types - το στοιχείο αυτό περιγράφει όλους τους τύπους και τις δομές δεδομένων που ανταλλάσσονται μεταξύ πελάτη και Web Service

Message - Εδώ περιγράφονται τα μηνύματα, είτε είναι μηνύματα αίτησης είτε απόκρισης. Για κάθε μήνυμα ορίζονται τα πεδία με το όνομα τους, και οι παράμετροι που αυτό μπορεί να δέχεται ή τις τιμές που επιστρέφει.

Operation - το στοιχείο αυτό περιγράφει την δομή μίας λειτουργίας που υποστηρίζεται από μία υπηρεσία. Κάθε operation αποτελείται από στοιχεία message που αντιπροσωπεύουν τα μηνύματα αίτησης και απάντησης.

PortType - Αποτελεί το περίβλημα του συνόλου των λειτουργιών του service

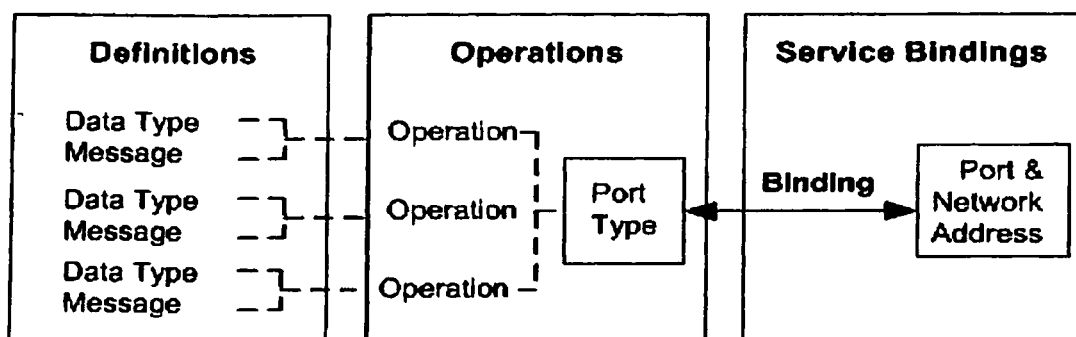
Binding - το στοιχείο αυτό περιγράφει το πρωτόκολλο και την μορφοποίηση των δεδομένων που πρόκειται να μεταδοθούν. Αν πρόκειται για το πρωτόκολλο SOAP η μετάδοση των μηνυμάτων γίνεται πάνω από το HTTP ή το SMTP.



Port – περιγράφεται το τελικό σημείο που ορίζεται σαν συνδυασμός μίας σύνδεσης (binding) και μιας διεύθυνσης δικτύου

Service – το στοιχείο περιέχει το όνομα του service και το σύνολο των τελικών σημείων (ports).

WSDL Basic Parts Relationship Diagram



Σχήμα 2.4: Η δομή ενός WSDL(πηγή: http://docs.oracle.com/cd/e18727_01/doc.121)

Οι πληροφορίες που περιγράφονται σε αρχείο WSDL μπορούμε να θεωρήσουμε ότι διακρίνονται σε δυο βασικές κατηγορίες. Η πρώτη περιλαμβάνει ποιο γενικούς ορισμούς της υπηρεσίας και στην δεύτερη περιγράφονται ποιο ειδικές και συγκεκριμένες πληροφορίες. Ποιο συγκεκριμένα τα στοιχεία εκείνα των οποίων οι περιγραφές δεν συμπεριλαμβάνουν πληροφορίες για το πώς υλοποιείται η διεπαφή της υπηρεσίας όπως types, messages, operations και portTypes συγκροτούν την ομάδα των γενικών ορισμών. Το τμήμα του WSDL που περιέχει πληροφορίες για τα στοιχεία binding, port, και service αποτελούν την δεύτερη ομάδα των ειδικών ορισμών. Τα αρχεία WSDL και η ειδικότερα η κατηγορία των γενικών ορισμών αποτελούν κρίσιμο κομμάτι της έρευνας μας και βασιζόμαστε στις πληροφορίες που αντλούμε από αυτά για να παράγουμε αποτελέσματα καθώς οι εντοπιζόμενες διαφορές των διαδοχικών εκδόσεων αυτών αποτελούν τη βάση της μελέτης εξέλιξης των συστημάτων λογισμικού που εξετάζουμε.

2.5. Σχετικές εργασίες

Γνωστή σχετική μελέτη για την εξέλιξη των Web Services είναι από τους Andrikopoulos et.al.[6]. Στην εργασία τους χρησιμοποιούν το εργαλείο srmmod



(<http://srmod.wordpress.com>) το οποίο αποτελεί επέκταση του IDE Eclipse και πιο συγκεκριμένα χρησιμοποιείται το σύνολο των εργαλείων του Epsilon. Στην μελέτη αυτή περιγράφονται οι υπηρεσίες με την χρήση ενός αφαιρετικού μεταμοντέλου που ονομάζεται Abstract Service Description (ASD) και διαχωρίζει την περιγραφή σε τρία διαφορετικά επίπεδα (non functional, Behavioral, Structural). Στην εργασία αναλύεται το Structural Layer από την οποία προκύπτουν και οι διαφορές. Προτείνεται ένα θεωρητικό πλαίσιο με σκοπό να βοηθήσει τους προγραμματιστές υπηρεσιών την προσπάθειά τους να αναπτύξουν εξελισσόμενες υπηρεσίες, με τρόπο τέτοιο ώστε να ελαχιστοποιούνται οι επιπτώσεις των αλλαγών, και να μην εκτείνονται πέρα από το πεδίο της εξέλιξης της υπηρεσίας. Η εργασία μέσω του θεωρητικού πλαισίου εντοπίζει την συμβατότητα μεταξύ δυο εκδόσεων δηλ., αν είναι προς τα εμπρός (Forward Compatibility) ή προς τα πίσω συμβατά (Backwards Compatibility) και εξετάζεται σε βάθος ποιες αλλαγές και πως, επηρεάζουν την διασύνδεση της υπηρεσίας με τους χρήστες.

Εργασία παρεμφερής με την μελέτη μας στην όποια έχει γίνει εμπειρική αξιολόγηση της εξέλιξης των υπηρεσιών διαδικτύου έχει γίνει από τους Stroulia et al. [13]. Στην μελέτη αυτή εξετάζονται πέντε διαφορετικές προσφερόμενες υπηρεσίες του διαδικτύου από διαφορετικές εταιρείες. Για την σύγκριση των αρχείων δημιούργησαν ένα εργαλείο με το όνομα V-Tracker βασισμένο σε μια παραλλαγή του αλγορίθμου tree edit distance που παρουσιάστηκε από τους Zhang – Shasha. Ο αλγόριθμος δημιουργεί δυο δέντρα τα οποία αναλύονται και ανάλογα τις διαφορές που εντοπίζονται (insertions, deletions, changes) προστίθεται και από ένα κόστος για κάθε μια από αυτές. Μετά και χρησιμοποιώντας την μέθοδο της ομαδοποίησης (clustering) ταξινομούνται οι ομάδες των διαφορών. Πρέπει να επισημάνουμε ότι τα αρχεία WSDL που χρησιμοποιούνται για σύγκριση μετατρέπονται σε μια ενδιάμεση μορφή XML κατάλληλη για την μελέτη τους. Τα αποτελέσματα τους επικεντρώνονται στην εμπροσθοβαρή συμβατότητα των εκδόσεων (Forward Compatibility) αν δηλαδή η νέα έκδοση δεν διακόπτει την ομαλή λειτουργία της διαπροσωπείας μεταξύ χρήστη και παρεχομένης υπηρεσίας.

Μια ακόμη σχετική έρευνα έχουν δημοσιεύσει οι Guowu Xie Jianbo Chen Iulian Neamtii [22]. Οι συγγραφείς αναλύουν επτά μεγάλες εφαρμογές ανοικτού κώδικα



(Samba, Bind 9, OpenSSH, SQLite, Vsftpd, Sendmail and Quagga) όλες γραμμένες στην διαδικασία γλώσσα C. Στην εργασία τους χρησιμοποιούν δυο εργαλεία, το ASTdiff για να συλλέξουν μετρήσεις που αφορούν αλλαγές και το RSM (Resource Standard Metrics) το οποίο ένα εμπορικό εργαλείο που χρησιμοποιείται για τον υπολογισμό της κυκλωματικής πολυπλοκότητα (Cyclomatic complexity). Η εργασία στοχεύει στο να εξακριβώσει κατά πόσον τα υφιστάμενα μοντέλα εξέλιξης του λογισμικού που ισχύουν για τα προγράμματα αυτά επιβεβαιώνουν τους οκτώ νόμους του Lehman. Στα αποτελέσματα επιβεβαιώνουν ότι η συνεχιζόμενη αλλαγή, η αύξηση της πολυπλοκότητα, η αυτορρύθμιση, και η συνεχής ανάπτυξη, εξακολουθούν να ισχύουν, δεν επικυρώνουν τη διατήρηση της Οργανωτικής Σταθερότητα, τη Διατήρηση εξοικείωση με το λογισμικό, την υποχώρηση της Ποιότητα του συστήματος, και την ύπαρξη ανατροφοδοτούμενου συστήματος.

Η διαφοροποίηση μας στην εργασία που παρουσιάζουμε είναι ότι δεν μένουμε στο πεδίο εύρεσης των αλλαγών μεταξύ των διαφορετικών εκδόσεων αλλά την επεκτείνουμε, μελετώντας την εξέλιξη των υπηρεσιών βασιζόμενοι σε ένα καλά εδραιωμένο σύνολο παρατηρήσεων που έχει γίνει από τον M.Lehman και την ομάδα του (αποτέλεσμα μελέτης 40 χρόνων) σχετικά με το πώς εξελίσσονται τα τυπικά συστήματα λογισμικού.



ΚΕΦΑΛΑΙΟ 3. ΕΡΓΑΛΕΙΟ ΑΝΑΛΥΣΗΣ WSDL ΑΡΧΕΙΩΝ

3.1 Περιγραφή Αρχιτεκτονικής

3.2 Υλοποίηση και Περιγραφή λειτουργιών

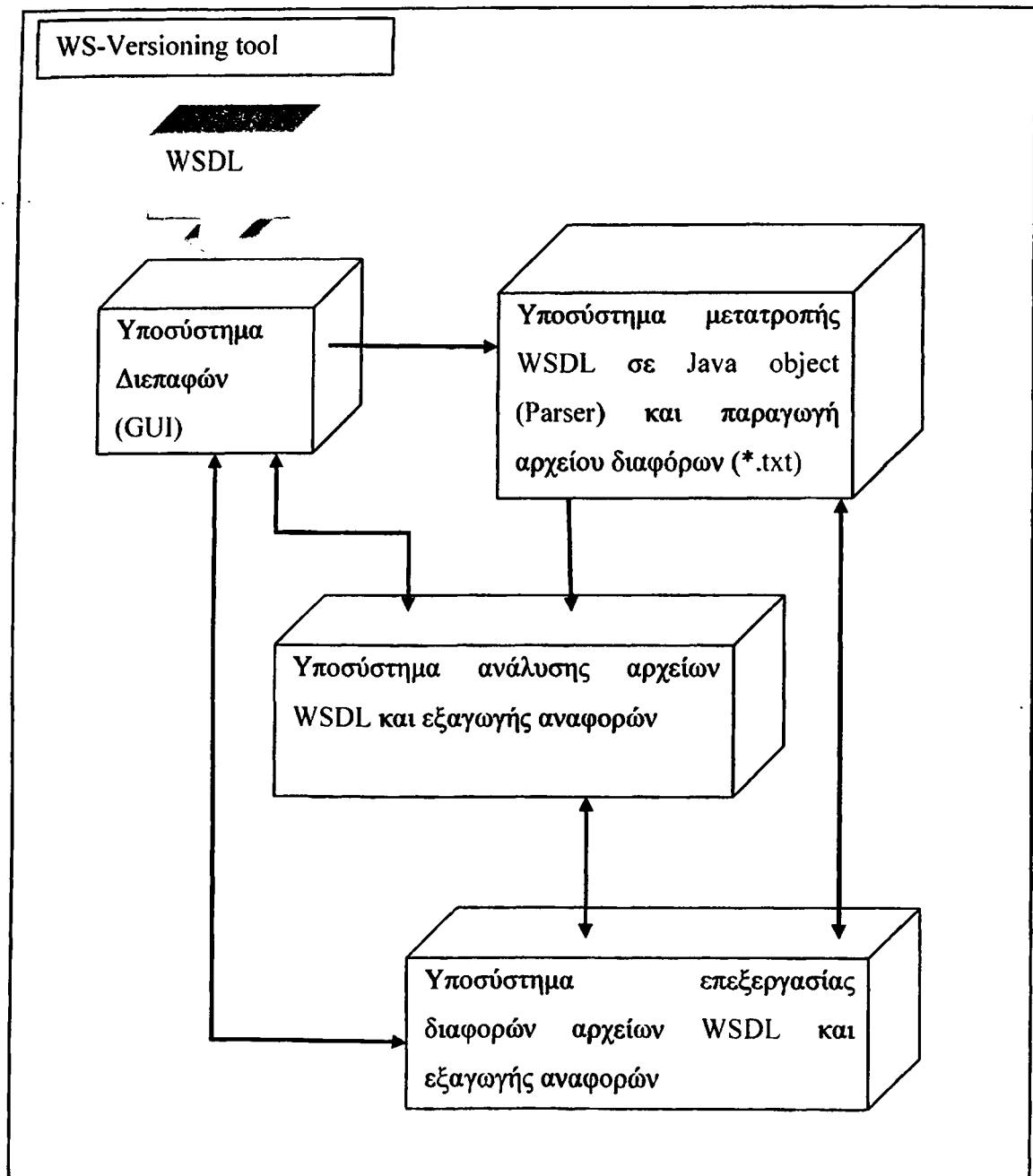
Ένας από τους στόχους μας ήταν η υλοποίηση ενός γραφικού εργαλείου που έχει την δυνατότητα να αξιοποιεί την «Εκφραστική Δύναμη» ενός WSDL αρχείου να αναπαριστά αλλά και να οπτικοποιεί, με τρόπο σαφή και κατανοητό, όλη την πληροφορία που ορίζεται σε αυτό, καθώς και να οπτικοποιηθούν οι διαφορές μεταξύ δυο εκδόσεων για να υποστηρίξει την μελέτη μας μέσω εξαγωγής των απαραίτητων αποτελεσμάτων. Το εργαλείο προσφέρει στους χρήστες πολλαπλές λειτουργίες, όπως τη ανάλυση ενός ή πολλών ταυτόχρονα WSDL αρχείων, η οπτικοποίηση της πληροφορίας που περιέχεται σε ένα αρχείο, η εξαγωγή αναφορών για χαρακτηριστικά – ιδιότητες που χρησιμοποιούμε στις μετρήσεις μας, η ανάλυση, η οπτικοποίηση και εξαγωγή των διαφορών που εντοπίζονται μεταξύ δυο εκδόσεων WSDL αρχείων και η δημιουργία αναφορών με αποτελέσματα των μετρήσεων που προκύπτουν.

3.1. Περιγραφή Αρχιτεκτονικής

Το εργαλείο που υλοποιήσαμε αποτελεί μία standalone εφαρμογή, δηλαδή δεν παρουσιάζει καμία εξωτερική διασύνδεση με άλλα συστήματα όπως το διαδίκτυο, μία βάση δεδομένων, ή ένας Web Server. Το εργαλείο αποτελείται από 4 τμήματα που αλληλεπιδρούν μεταξύ τους. Στη συνέχεια παρουσιάζονται σχηματικά τα τέσσερα



βασικά υποσυστήματα που συνιστούν το εργαλείο μας, καθώς και ο τρόπος διασύνδεσής τους.



Σχήμα 3.1: Τα υποσυστήματα του εργαλείου

- Το **υποσύστημα διεπαφών** αναλαμβάνει την παρουσίαση όλου του όγκου της πληροφορίας που συλλέγουμε καθώς και τις αρχικές επιλογές για την είσοδο των αρχείων WSDL. Το εργαλείο περιέχει πολλαπλές διαπροσωπείες (multi interfaces system), και ανάλογα με την δουλειά την οποία επιτελεί ο χρήστης έρχεται σε επαφή με διαφορετικές φόρμες, μενού και οπτικά συστατικά (visual components).
- Το **υποσύστημα μετατροπής WSDL σε Java αντικείμενα (Parser) και παραγωγής αρχείου διαφορών** αποτελεί το λογισμικό ανοικτού κώδικα που χρησιμοποιούμε και αναλαμβάνει την λεκτική ανάλυση ενός αρχείου (WSDL) την μετατροπή τους σε δομές αντικειμενοστραφούς γλώσσας Java, και μπορεί μεταξύ άλλων να παράγει σε απλή μορφή αρχείου τις εντοπιζόμενες διαφορές μεταξύ διαδοχικών εκδόσεων WSDL αρχείων.
- Το **υποσύστημα ανάλυσης αρχείων WSDL και εξαγωγής αναφορών** εξάγει την χρήσιμη πληροφορία από τα αρχεία διαφορών και την ταξινομεί βάση του πρότυπου WSDL που περιγράψαμε στο κεφάλαιο 2. Το υποσύστημα παίζει κεντρικό ρόλο και καλείται από όλα τα υποσυστήματα πλην του υποσυστήματος μετατροπής WSDL σε Java αντικείμενα.
- Το **υποσύστημα των διαφορών** όπως προκύπτει και από τον τίτλο ασχολείται αποκλειστικά με τις διαφορές των αρχείων, επιτελεί ελέγχους για τα στοιχεία που έχουν διαφοροποιηθεί, καταγράφει, αποθηκεύει και ταξινομεί τις πληροφορίες για κάθε εξελικτικό βήμα που αναλύεται, συνεργάζεται με όλα τα υποσυστήματα αμφίδρομα, και εξάγει αναφορά για το σύνολο της πληροφορίας που αφορά τις εντοπιζόμενες διαφορές.

3.2. Υλοποίηση και Περιγραφή λειτουργιών

Ακολουθεί μία σειρά θεμάτων που έχουν να κάνουν με τις τεχνολογίες που χρησιμοποιήθηκαν κατά την υλοποίηση του εργαλείου ενώ θα παρουσιαστούν και οι λειτουργίες του, μέσω παραδείγματος.



3.2.1. Πλατφόρμα Υλοποίησης

Η πλατφόρμα υλοποίησης επιλέχθηκε να είναι αντικειμενοστραφής και, συγκεκριμένα η υλοποίηση του εργαλείου συστήματος έγινε χρησιμοποιώντας την έκδοση Java Platform, Standard Edition 7, με την χρήση του Java Development Kit 1.7.0_07.

Για την ανάπτυξη του κώδικα του εργαλείου χρησιμοποιήθηκε ως περιβάλλον προγραμματισμού το Netbeans και η έκδοση 7.1.2. Το Netbeans έγινε λογισμικό ανοιχτού κώδικα (open source) από την Sun Microsystems το 2000, η οποία παρέμεινε ως χορηγός του έργου μέχρι τον Ιανουάριο του 2010, όταν η Sun Microsystems έγινε θυγατρική της Oracle.

3.2.2. Λογισμικό Ανοιχτού Κώδικα *Membrane Soa Model*

Το Membrane SOA Model δημιουργήθηκε αρχικά από την εταιρεία predic8 με έδρα την Βόννη που παρέχει συμβουλευτικές υπηρεσίες πάνω σε θέματα Soa και Web Services. Στην σελίδα <http://www.membrane-soa.org/> παρουσιάζονται διάφορα εργαλεία λογισμικού που αφορούν SOA αρχιτεκτονικές όπως τα Membrane Service Proxy, Membrane Monitor, κ.α. Εμείς χρησιμοποιούμε το Membrane SOA Model. Το λογισμικό αυτό έγινε ανοιχτό προς χρήση το Μάρτιο του 2012 υπό την Apache Software Foundation με την άδεια χρήσης ASF 2.0 Open Source License.

Το εργαλείο αυτό προσφέρει πλήθος χρησίων λειτουργιών όπως ο λεκτικός αναλυτής (WSDLParser) που μας επιτρέπει την μετατροπή του WSDL αρχείου σε σύνολο αντικειμένων Java και μεθόδους να συγκρίνουν και να αναλύουν WSDL αρχεία.

Ο λεκτικός αναλυτής είναι υλοποιημένος σε Groovy και Java και η διεπαφή του βρίσκεται στην σημαντικότερη από τις παρεχόμενες βιβλιοθήκες soa-model-core-1.4.1.4.jar και στο πακέτο com.predic8.WSDL και η υλοποιημένη κλάση είναι η WSDLParser. Η κλάση αυτή είναι επέκταση της κλάσης AbstractParser της βιβλιοθήκης com.predict8.soamodel.



Μια άλλη λειτουργία που χρησιμοποιούμε είναι την εύρεση διαφορών μεταξύ δυο WSDL αρχείων. Η διεπαφή υλοποιείται μέσα από την κλάση CompareWSDL και της μεθόδου compare() και εντοπίζεται στο πακέτο κλάσεων main.java.samples.WSDL. Μέσω της μεθόδου compare() γίνονται κλήσεις προς τον λεκτικό αναλυτή ο οποίος και μετατρέπει τα αρχεία σε Java αντικείμενα και στην συνέχεια χρησιμοποιείται η κλάση WSDLDiffGenerator – γεννήτρια διαφορών και μέσω αυτής της κλάσης παράγεται ένα αρχείο κειμένου (*.txt) με τις εντοπιζόμενες διαφορές (Σχήμα 3.2).

```

*****
Types has changed:
-- Schema http://ec2.amazonaws.com/doc/2009-11-30/ has changed:
--- Element CreateImage removed.
--- Element CreateImageResponse removed.
-----
--- Element DescribePlacementGroups added.
-----
--- ComplexType CreateImageType removed.
--- ComplexType InstanceLicenseRequestType added.
-----
--- ComplexType RunInstancesType has changed:
--- Sequence has changed:
--- Element license added.
--- Element privateIpAddress added.
-----
- PortType AmazonEC2PortType has changed:
-- Operation CreateImage removed.
-- Operation ActivateLicense added.
-----
-- Operation RunInstances has changed:
--- Input has changed:
--- Message RunInstancesRequestMsg has changed:
--- Part RunInstancesRequestMsgReq has changed:
--- Element RunInstances has changed:
--- ComplexType RunInstancesType has changed:
-----
--- Output has changed:
--- Message RunInstancesResponseMsg has changed:
--- Part RunInstancesResponseMsgResp has changed:
--- Element RunInstancesResponse has changed:
--- ComplexType RunInstancesResponseType has changed:
--- Sequence has changed:

```

Σχήμα 3.2: Το παραγόμενο αρχείο Διαφορών (*.txt)

3.2.3. Περιγραφή Λειτουργίας

Το εργαλείο εκτελεί δυο βασικές λειτουργίες. Στη μια αναλύει ομάδες αρχείων WSDL που περιγράφουν μια υπηρεσία. Καθένα από αυτά αναλύονται στα επιμέρους συστατικά τους και παίρνουμε μετρήσεις που αφορούν στο πλήθος των συστατικών. Η λειτουργία αυτή δημιουργεί και τελική αναφορά με όλες τις επιμέρους μετρήσεις.

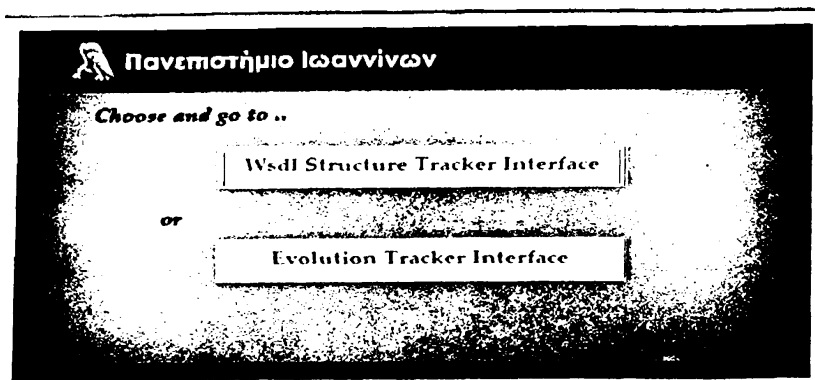


Στην δεύτερη λειτουργία μπορούμε να επιλέξουμε μια ομάδα αρχείων ή όλα από μια συγκεκριμένη υπηρεσία, και να δημιουργήσουμε ένα «ιστορικό» της διαδρομής εξέλιξης της υπηρεσία μέσω των διαδοχικών βημάτων που παρουσιάζονται χρονολογικά τα WSDL αρχεία. Για κάθε ζεύγος αρχείων (διαδοχική ακολουθία εκδόσεων) παράγουμε το αρχείο διαφορών και έπειτα αναλύουμε τις διαφορές τις ταξινομούμε και στο τέλος παίρνουμε τις απαραίτητες μετρήσεις για κάθε μια από αυτές. Οι κατηγορίες διαφορών οι οποίες εξετάζουμε είναι πρώτα σε ένα ποιο αφηρημένο επίπεδο και εξετάζουμε συνολικά αλλαγές όπως α) τις προσθήκες (additions) β) τις αφαιρέσεις (removes or deletions) γ) τις τροποποιήσεις (updates) που εντοπίζονται. Στην συνέχεια εξετάζουμε τα επιμέρους συστατικά του αρχείου και ποιες αλλαγές έγιναν σε αυτά. Τα συστατικά ενός WSDL αρχείου που εξετάζουμε είναι α) οι λειτουργίες (operations) και εντοπίζουμε τις προσθήκες, αφαιρέσεις και τις τροποποιήσεις σε αυτές β) τους τύπους δεδομένων που δηλώνονται στο αρχείο για κάθε operation (complex elements, simple elements, elements), γ) τα στοιχεία που αφορούν το επίπεδο διασύνδεσης και μεταφοράς (ports, services, bindings). Πρέπει να επισημάνουμε ότι η μελέτη μας δεν επικεντρώνεται στην εξέλιξη του τρόπου διασύνδεσης και μεταφοράς και τα στοιχεία που συγκεντρώνονται είναι για την πληρότητα των δεδομένων και όχι για περαιτέρω ανάλυση.

3.2.4. Παρουσίαση εργαλείου

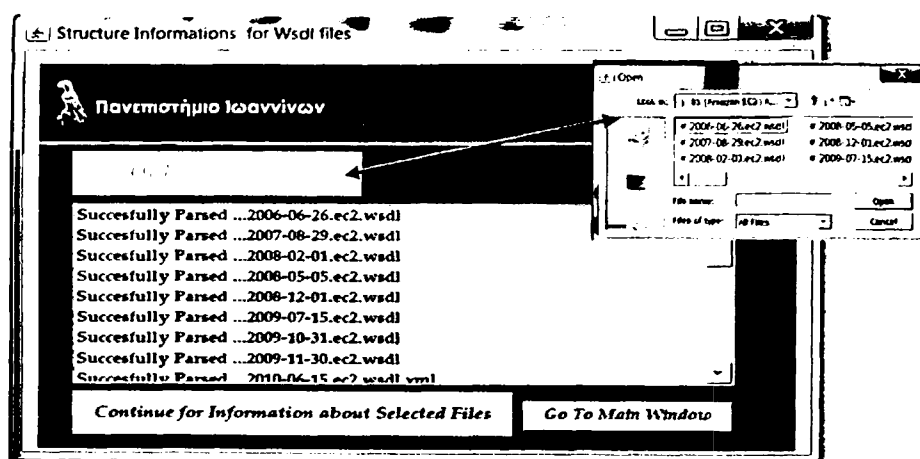
Το εργαλείο ξεκίνα με μια διεπαφή για να επιλέξουμε πια από τις προσφερόμενες λειτουργίες θέλουμε να εκτελέσουμε. Η επιλογή *WSDL Structure Tracker Interface* μας οδηγεί στην πρώτη λειτουργία της ανάλυσης της δομής των αρχείων που θα επιλέξουμε (Σχήμα 3.3: Start).





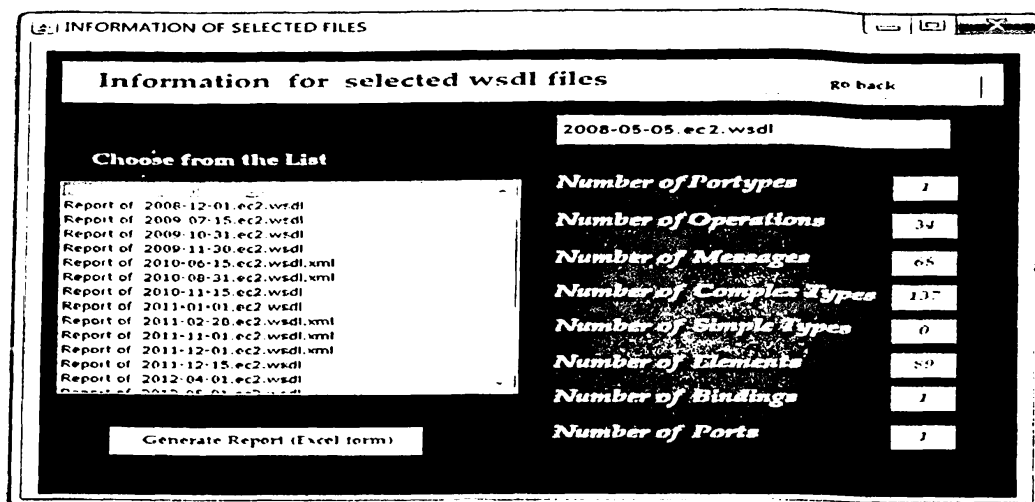
Σχήμα 3.3: Διεπαφή Start

Επιλέγοντας *WSDL Structure Tracker Interface* εισερχόμαστε στην λειτουργία ανάλυσης ομάδας αρχείων και εμφανίζεται το επόμενο παράθυρο με το οποίο επιλεγούμε τα αρχεία.



Σχήμα 3.4: Διεπαφή Επιλογή Αρχείων

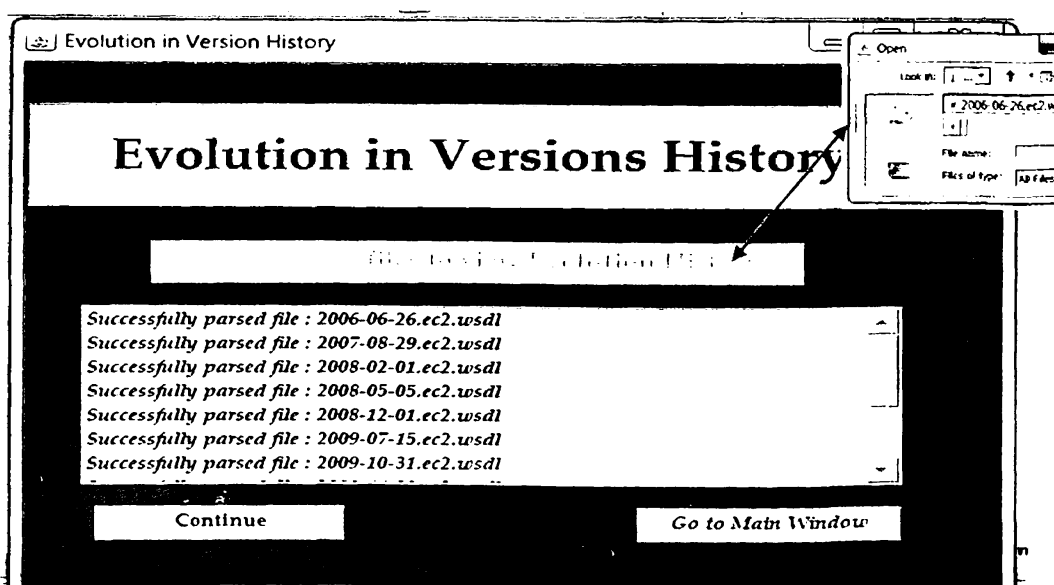
Αφού επιλέξουμε τα αρχεία εμφανίζονται στο πλαίσιο κειμένου τα ονόματα των αρχείων και στην συνέχεια επιλέγοντας «Continue for Information about Selected Files» μεταβαίνουμε στην διεπαφή ανάλυσης των αρχείων (Σχήμα 3.5). Με την επιλογή Go To Main Window επιστρέφουμε στο προηγούμενο παράθυρο για να επανακαθορίσουμε την επιλογή μας (Σχήμα 3.3).



Σχήμα 3.5: Διεπαφή Information of selected WSDL files

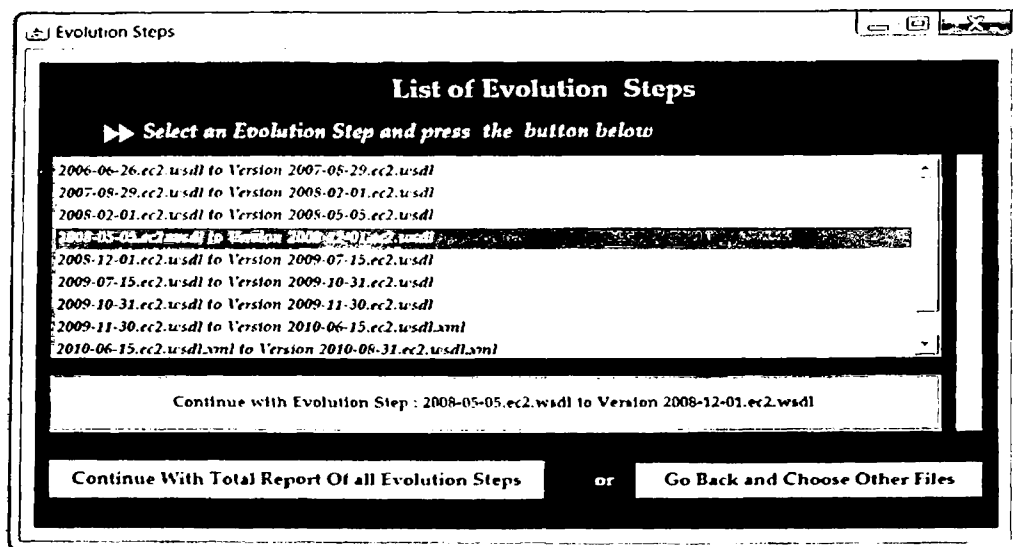
Στο παράθυρο Information of selected WSDL files επιλέγοντας ένα αρχείο από τη λίστα βλέπουμε τον αριθμό από το κάθε στοιχείο που περιέχονται σε αυτό. Η επιλογή Generate Report διαβάζει τα δεδομένα του παραθύρου και τα τοποθετεί σε αρχείο Excel.

Από το αρχικό παράθυρο (Σχήμα 3.3 Start) και επιλέγοντας Evolution Tracker Interface εκτελούμε τη δεύτερη λειτουργία που παρέχεται και αφορά την ανάλυση των διαφορών των αρχείων WSDL. Η επιλογή των αρχείων που επιθυμούμε γίνεται μέσω του παρακάτω παραθύρου (Σχήμα 3.6)



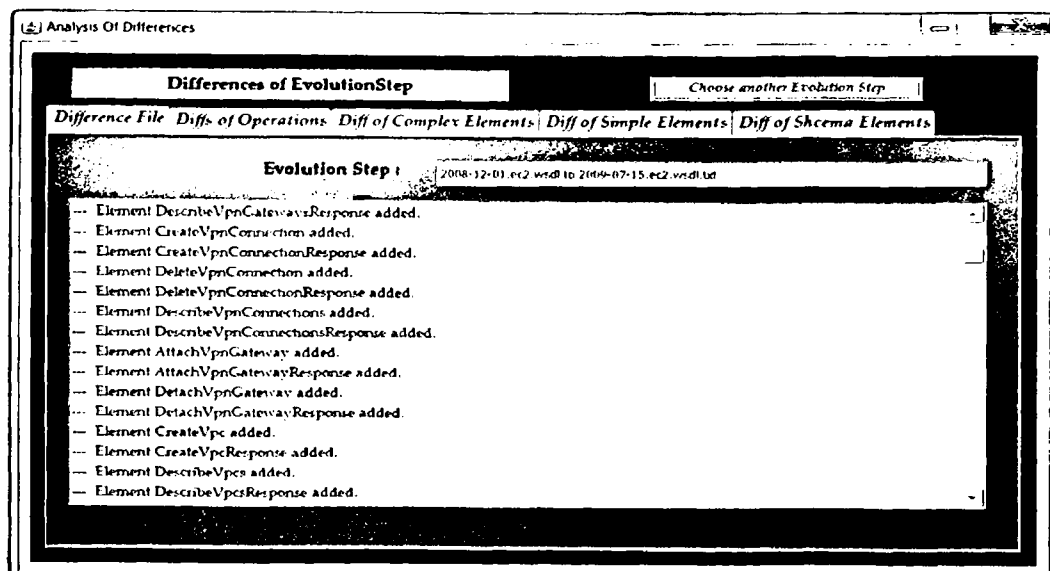
Σχήμα 3.6: Διεπαφή Evolution in Version History

Με την επιλογή Go To Main Window επιστρέφουμε στο προηγούμενο παράθυρο (Σχήμα 3.3) για να επανακαθορίσουμε την επιλογή μας. Η επιλογή continue μα οδηγεί στο παράθυρο με την λίστα των αρχείων που συγκρίναμε ανά δυο (Σχήμα 3.7)



Σχήμα 3.7: Διεπαφή List of Evolution Step

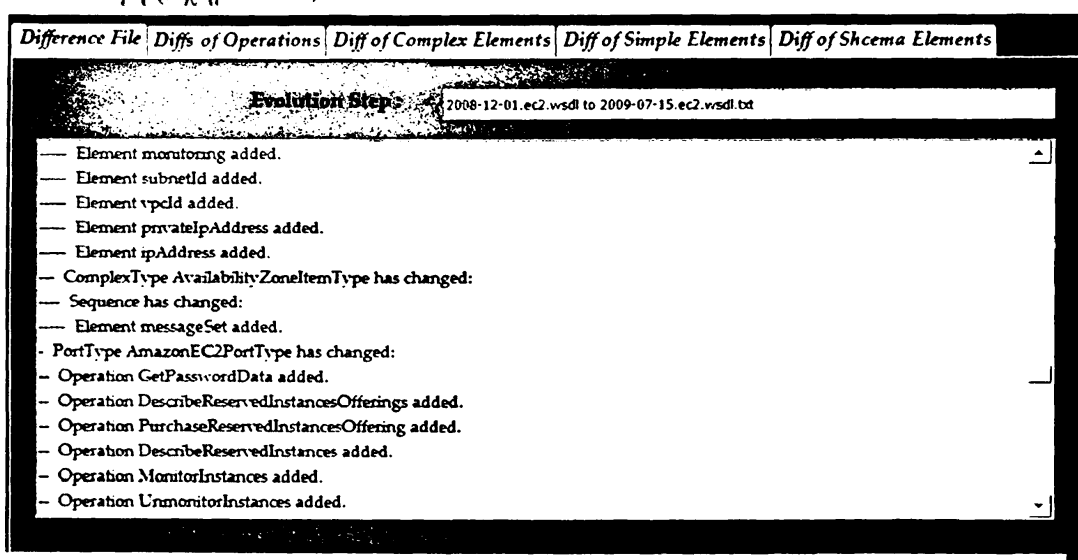
Στο πεδίο κειμένου περιέχονται οι περιγραφές με τα βήματα (βήμα είναι η σύγκριση δυο διαδοχικών αρχείων) που έχουν προκύψει κατά την εξέλιξη της υπηρεσίας ταξινομημένα βάση της χρονολογικής εμφάνισής τους. Επιλέγοντας ένα από αυτά η επιλογή « Continue with Evolution Step: (name of step)» ενεργοποιείται και επιλέγοντας την εισερχόμαστε στην διεπαφή «Analysis of Differences» στην οποία αναλύονται οι διαφορές που έχουν εντοπιστεί.



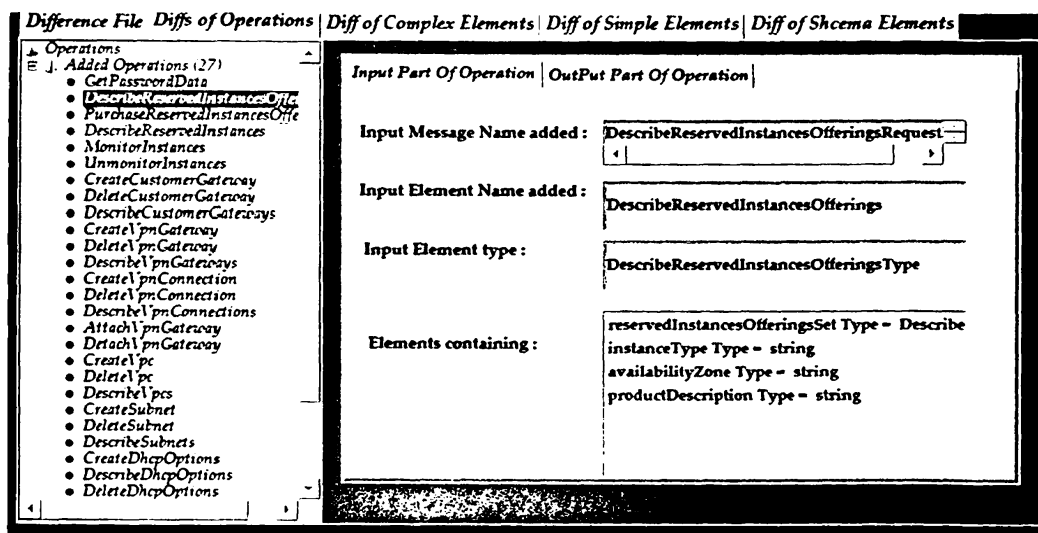
Σχήμα 3.8: Διεπαφή Analysis of Differences

Η διεπαφή αυτή περιέχει πέντε καρτέλες:

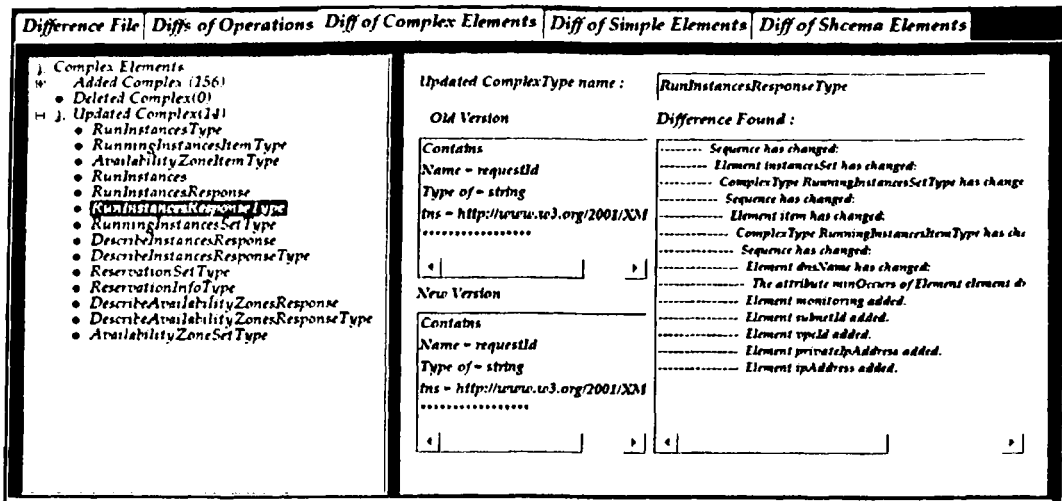
- Καρτέλα που παρουσιάζεται το αρχείο των διαφορών (Σχήμα 3.9)
- Καρτέλα με τις διαφορές που εντοπιστήκαν στις operations και πληροφορίες για την κάθε αλλαγή (προσθήκες, διαγραφές, τροποποιήσεις) (Σχήμα 3.10)
- Καρτέλα για τους σύνθετους τύπους (complex types) που άλλαξαν και πληροφορίες για την κάθε αλλαγή (Σχήμα 3.11).
- Καρτέλα για τους απλούς τύπους (simple types) που άλλαξαν και πληροφορίες για την κάθε αλλαγή (Σχήμα 3.12).
- Καρτέλα για τα στοιχεία (elements) που άλλαξαν και πληροφορίες για την κάθε αλλαγή (Σχήμα 3.13).



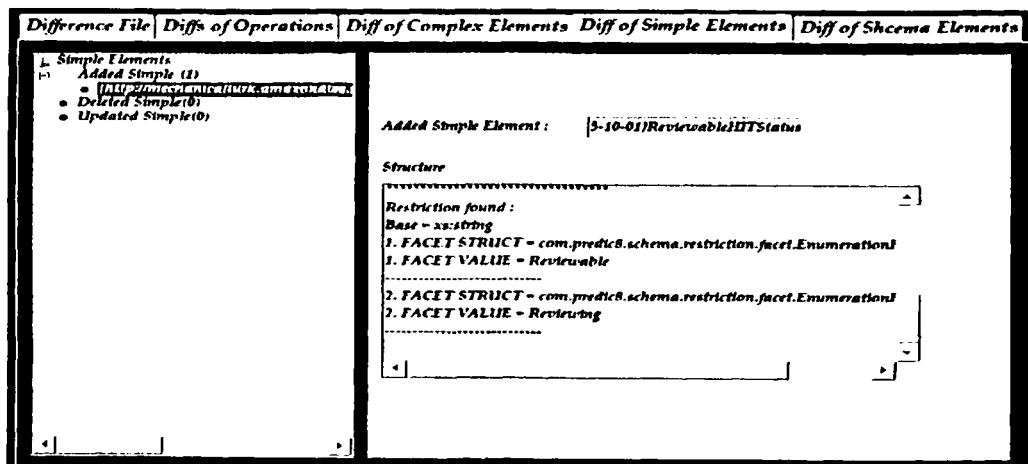
Σχήμα 3.9: Καρτέλα Αρχείο Διαφορών (Difference File)



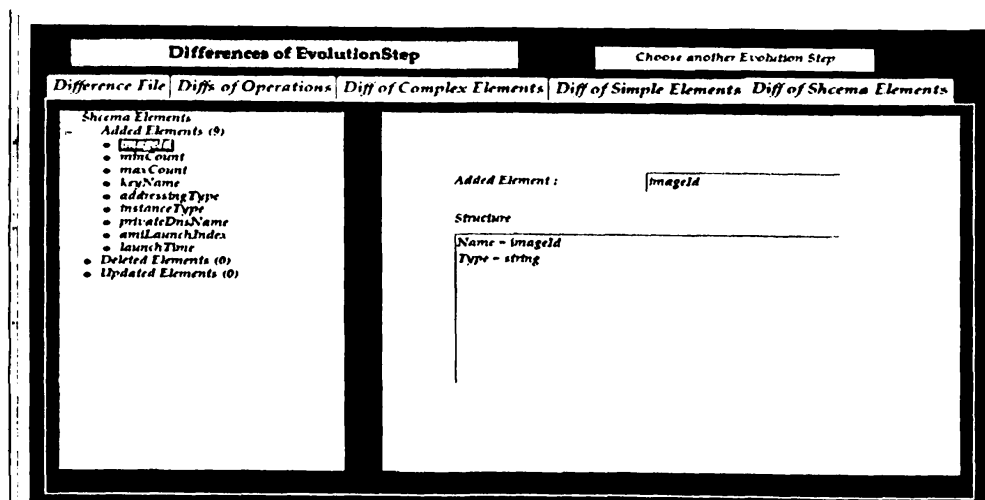
Σχήμα 3.10: Καρτέλα Διαφορές των Λειτουργιών (Diffs of Operations)



Σχήμα 3.11: Καρτέλα Διαφορές των σύνθετων τύπων (Diffs of Complex Types)



Σχήμα 3.12: Καρτέλα Διαφορές των απλών τύπων (Diffs of Simple Elements)



Σχήμα 3.13: Καρτέλα Διαφορές των στοιχείων (Diffs of Schema Elements)

Η επιλογή «Continue With Total Report Of all Evolution Steps» μας οδηγεί σε καινούργια διεπαφή που παρουσιάζονται συνολικά οι διαφορές για καθένα από τα εξελικτικά βήματα που δημιουργήσαμε (Σχήμα 3.14).

Total Differences Report

Differences per Evolution Step

Evolution Step	Operations	Complex Types	Simple Types	Elements
2009-05-15 ElasticLoadBalancing.wsdl to Versio	Added=4 Deleted=0 Updated=4			
2009-11-25 ElasticLoadBalancing.wsdl to Versio		Added=25 Deleted=0 Updated=9		
2010-07-01 ElasticLoadBalancing.wsdl to Versio			Added=1 Deleted=0 Updated=0	
2011-01-05 ElasticLoadBalancing.wsdl to Versio				Added=1 Deleted=0 Updated=10
2011-08-15 ElasticLoadBalancing.wsdl to Versio				
2011-11-15 ElasticLoadBalancing.wsdl to Versio				

Generate Report(excel form)

Σχήμα 3.14: Καρτέλα Αναφορά Διαφορών (Total Differences Report)

Η επιλογή «Generate Report Excel» δημιουργεί αναφορά για όλες τις διαφορές από κάθε ένα εξελικτικό βήμα που έχει δημιουργηθεί (Σχήμα 3.15).

	A	B	C	D	E	F	G	H	I	J
1	Versions	Operations Added	Operations Deleted	Operations Updated	Complex Types Added	Complex Types Deleted	Complex Types Updated	Simple Elements Added	Simple Elements Deleted	Simple Elements Updated
2	2006-06-26(mibal)	0	0	0	0	0	0	0	0	0
3	2007-08-29	6	0	14	27	2	4	0	0	0
4	2008-02-01	6	0	4	26	0	10	0	0	0
5	2008-05-05	8	0	0	26	0	0	0	0	0
6	2008-12-01	4	0	34	20	0	37	0	0	0
7	2009-07-15	27	0	3	102	0	10	0	0	0
8	2009-10-31	9	0	10	41	4	29	0	0	0
9	2009-11-30	7	0	2	30	0	6	0	0	0
10	2010-06-15	6	0	26	24	0	14	0	0	0
11	2010-08-31	4	0	30	16	0	67	0	0	0
12	2010-11-15	4	0	3	24	0	9	0	0	0
13	2011-01-01	23	0	21	76	2	55	0	0	0
14	2011-02-28	0	0	8	1	0	28	0	0	0
15	2011-11-01	1	0	7	10	0	20	0	0	0
16	2011-12-01	8	0	14	29	0	34	0	0	0
17	2011-12-15	1	0	1	7	0	3	0	0	0
18	2012-04-01	4	0	10	17	0	20	0	0	0
19	2012-05-01	3	0	2	15	0	9	0	0	0

Σχήμα 3.15: Παράδειγμα Excel αρχείου

ΚΕΦΑΛΑΙΟ 4. ΠΕΡΙΓΡΑΦΗ ΥΠΗΡΕΣΙΩΝ AMAZON ΚΑΙ ΔΕΔΟΜΕΝΩΝ

4.1 Ορολογία, ορισμοί και συμβολισμοί

4.2 Περιγραφή υπηρεσιών και δεδομένων

Σε αυτό το κεφάλαιο θα παρουσιάσουμε έξι σύνολα δεδομένων WSDL αρχείων, από τις προσφερόμενες Web Services της Αμερικανικής εταιρείας ηλεκτρονικού εμπορίου Amazon που συλλέξαμε, αναλύσαμε και μελετήσαμε με την χρήση του εργαλείου WS-Versioning tool που δημιουργήσαμε ειδικά για τον σκοπό αυτό.

Πριν προχωρήσουμε στην περιγραφή των υπηρεσιών και των χαρακτηριστικών τους πρέπει να καθορίσουμε μια σαφή ορολογία, μαζί με ένα σύνολο ορισμών για τις μετρούμενες ιδιότητες των συνόλων δεδομένων που έχουμε αναλύσει.

4.1. Ορολογία, ορισμοί και συμβολισμοί

- *Έκδοση ή Version*: Ένα στιγμιότυπο της προσφερόμενης υπηρεσίας που δημοσιοποιήθηκε από την εταιρεία με σκοπό την διόρθωση, την επέκταση ή την βελτίωση αυτής. Κάθε καινούργια έκδοση την καταχωρούμε με βάση το όνομα της χρονικής εμφάνισης της. Έτσι, το σύνολο είναι ταξινομημένο με τις συνεχείς εκδόσεις που παρουσιάζονται από τις παλαιότερες προς τις νεότερες.
- *Βήμα ή Step*: Καθορίζουμε σαν βήμα μεταξύ ζεύγους αρχείων WSDL την ενέργεια μετάβασης από την παλαιότερη έκδοση V_i στην νεότερη V_j ($V_i \rightarrow V_j$). Συνώνυμα: Μετάβαση ή Transition.



Ιδιότητες βήματος(step)

- Num of Operations: Ο αριθμός των λειτουργιών που εμφανίζονται σε ένα αρχείο WSDL.
- Num New Operations: Ο αριθμός των λειτουργιών του νεότερου αρχείου WSDL.
- Total Num of Changes Operations: Ο αθροιστικός αριθμός των αλλαγών μεταξύ 2 αρχείων WSDL.
- Num of Changes at Operations: Ο αριθμός αλλαγών των λειτουργιών του μεταξύ 2 αρχείων WSDL.
- Num of Changes at Types: Ο αριθμός αλλαγών των τύπων μεταξύ 2 αρχείων WSDL.
- Added Operations: Ο αριθμός των λειτουργιών που προστεθήκαν στην νεότερη έκδοση.
- Deleted Operations: Ο αριθμός των λειτουργιών που αφαιρέθηκαν από την νεότερη έκδοση.
- Update Operations: Ο αριθμός των λειτουργιών που τροποποιήθηκαν στην νεότερη έκδοση.
- Num of Types: Ο αριθμός των σύνθετων τύπων και των απλών τύπων του σχήματος WSDL.
- Added ComplexTypes: Ο αριθμός των σύνθετων τύπων του σχήματος που προστεθήκαν στην νεότερη έκδοση.
- Deleted ComplexTypes: Ο αριθμός των σύνθετων τύπων του σχήματος που αφαιρέθηκαν από την νεότερη έκδοση.
- Updated ComplexTypes: Ο αριθμός των σύνθετων τύπων του σχήματος που τροποποιήθηκαν στην νεότερη έκδοση.
- Added Simple Types: Ο αριθμός των απλών τύπων που προστέθηκαν στην νεότερη έκδοση.
- Deleted Simple Types: Ο αριθμός των απλών τύπων που αφαιρέθηκαν από την νεότερη έκδοση.



- **Updated Simple Types:** Ο αριθμός των απλών τύπων που τροποποιήθηκαν στην νεότερη έκδοση.
- **Added Elements:** Ο αριθμός στοιχείων (elements) που προστέθηκαν στην νεότερη έκδοση.
- **Deleted Elements:** Ο αριθμός στοιχείων (elements) που αφαιρέθηκαν από την νεότερη έκδοση.
- **Updated Elements:** Ο αριθμός στοιχείων (elements) που τροποποιήθηκαν στην νεότερη έκδοση.
- **Μέγεθος σχήματος ή schema size:** Ο συνολικός αριθμός των στοιχείων που δηλώνονται στην περιγραφή WSDL της υπηρεσίας και αποτελείται από portTypes, ports, services, operations, Complex Types, Simple Types, elements.
- **Ανάπτυξη σχήματος ή schema growth:** Η διαφορά που προκύπτει μεταξύ δυο διαδοχικών εκδόσεων και των υπό εξέταση στοιχείων μεταξύ της νεότερης έκδοσης και της παλαιότερης έκδοσης χρονικά π.χ., ο αριθμός των operation της νεότερης έκδοσης μείον του αριθμού αυτών της παλαιότερης έκδοσης. (Χρησιμοποιείται η πραγματική αξία της αφαίρεσης και όχι η απόλυτη).
- **Αλλαγές μεταξύ διαδοχικών εκδόσεων ή Version changes:** ο αριθμός των αλλαγών αθροιστικά στον οποίο περιέχονται οι επιμέρους αλλαγές (προσθήσεις, αφαιρέσεις, τροποποιήσεις) από κάθε συστατικό στοιχείο που εξετάζουμε για το συγκεκριμένο βήμα.

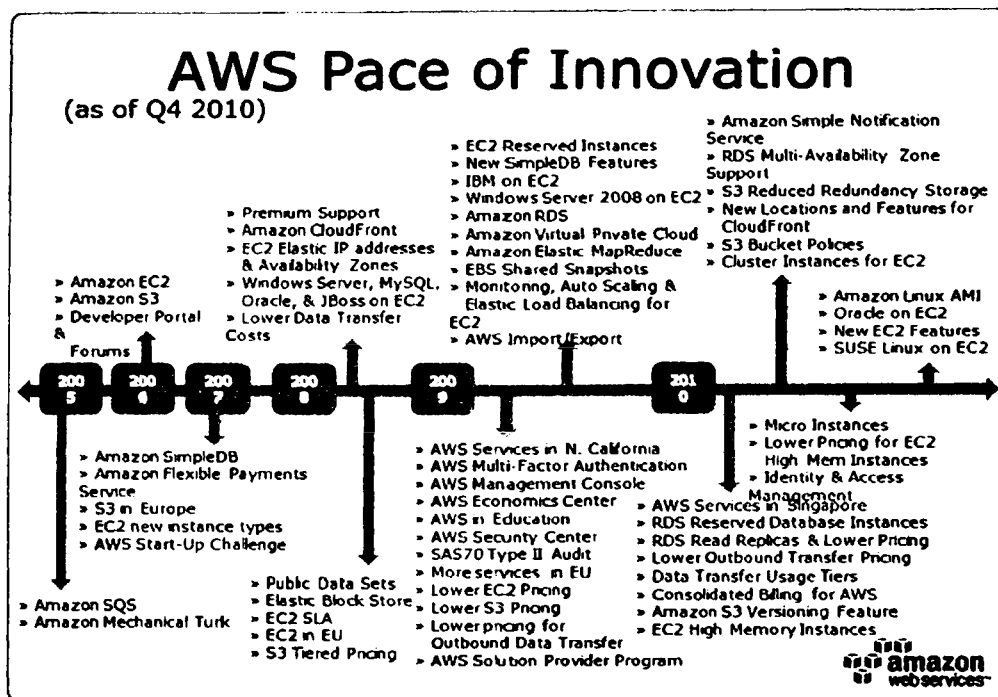
Ορισμοί που αφορούν το σύνολο των Δεδομένων.

- **Συνολικός αριθμός Αλλαγών ή Total Num Changes:** Ο συνολικός αριθμός των αλλαγών από όλα τα βήματα που έχουν καταγράψει για την εξεταζόμενη υπηρεσία.
- **Πολυπλοκότητα σχήματος ή complexity of schema:** Η μέτρηση της πολυπλοκότητας του λογισμικού κατά την διάρκεια εξέλιξης του. Είναι η άθροιση όλων των επιμέρους υπολογισμών πολυπλοκότητας ανά βήμα και δίνεται από τον τύπο, τροποποιημένα συστατικά/ νέα συστατικά (updated modules/new modules).



- Αλλαγές ανά ημέρα (Changes per Day): Ο αριθμός των αλλαγών ανά ημέρα προκύπτει από τον αριθμό των αλλαγών που έγιναν σε μια συγκεκριμένη έκδοση, και περιλαμβάνει τις προσθήσεις και τις τροποποιήσεις αυτής, προς των αριθμό των ημερών που μεσολάβησαν μεταξύ αυτής της έκδοσης και της προηγούμενης χρονικά έκδοσης.

4.2. Περιγραφή υπηρεσιών και δεδομένων



Σχήμα 4.1: Η υπολογιστική πλατφόρμα υπηρεσιών της Amazon. Πηγή: Jurg van Vliet and Flavia Paganelli. Programming Amazon EC2, page xi, 2010.

Τα Amazon Web Services (AWS συντομογραφία) είναι μια συλλογή απομακρυσμένων υπηρεσιών υπολογιστών που συνθέτουν μια υπολογιστική πλατφόρμα(cloud), και προσφέρονται μέσω του Διαδικτύου από την Amazon.com.

Η Amazon επίσημα ξεκίνησε το 2006, και τα Amazon Web Services παρέχουν online υπηρεσίες άλλων web sites ή client-side εφαρμογές. Οι περισσότερες από αυτές τις υπηρεσίες δεν εκτίθενται απευθείας στους τελικούς χρήστες, αλλά, αντίθετα, προσφέρεται η λειτουργικότητα που άλλοι προγραμματιστές μπορούν να



χρησιμοποιήσουν στις εφαρμογές τους. Οι Amazon Web Services είναι προσβάσιμες μέσω HTTP, χρησιμοποιώντας REST και SOAP πρωτόκολλα. Όλες οι υπηρεσίες χρεώνονται με βάση τη χρήση, η οποία διαφέρει από υπηρεσία σε υπηρεσία.

Η Amazon στην σελίδα που αφορούν τις υπηρεσίες (<http://aws.amazon.com/releasenotes/>), δημοσιοποιεί όλα τα αρχεία της εξέλιξης των υπηρεσιών της και έχει ανοιχτή πρόσβαση από όλους τους ενδιαφερόμενους χρήστες. Η συλλογή των δεδομένων μας ξεκίνησε τον Ιούνιο του 2012 και συνεχίστηκε μέχρι και τον Οκτώβριο του 2013. Συγκεντρώσαμε τα επίσημα αρχεία από την συγκεκριμένη σελίδα που αφορούν τις υπηρεσίες Rest που χρησιμοποιούν WSDL αρχεία, καθώς οι περισσότερες υπηρεσίες δεν χρησιμοποιούν WSDL περιγραφές αλλά επικοινωνούν απευθείας με το πρωτόκολλο HTTP. Αυτά τα αρχεία στη συνέχεια μετονομάστηκαν βάση ονομάτων που συσχετίζονται με την ημερομηνία που δημοσιοποιήθηκαν. Τα αρχεία υποβλήθηκαν σε επεξεργασία ανά ζεύγη από το εργαλείο μας, και συλλέξαμε (α) τις διαφορές μεταξύ δύο διαδοχικών εκδόσεων και (β) τις μετρικές που απαιτούνται για την μελέτη των συστημάτων.

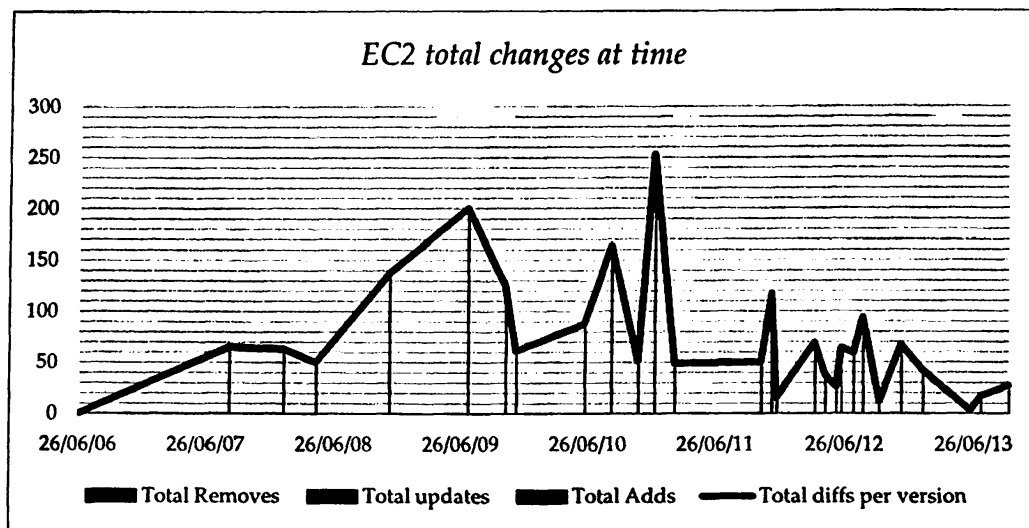
4.2.1. Amazon Elastic Compute Cloud (Amazon EC2)

Η υπηρεσία Elastic Compute Cloud (EC2) αποτελεί ένα κεντρικό κόμβο της cloud computing πλατφόρμας της Amazon, και παρέχει την δυνατότητα χρησιμοποίησης υπολογιστικής δύναμης χρησιμοποιώντας την τεχνική των εικονικών εξυπηρετητών (virtual server). Χρησιμοποιώντας την Amazon EC2 εξαλείφεται η ανάγκη για επένδυση σε τεχνολογικό εξοπλισμό κατά την δημιουργία ή επέκταση μια εταιρίας πληροφορικής, έτσι ώστε να μπορούν να αναπτύξουν εφαρμογές και ηλεκτρονικά προϊόντα γρηγορότερα. Το Amazon EC2 προσφέρει μεταβλητότητα όσο αφορά την υπολογιστική ικανότητα προσφέροντας από ένα ή μερικά virtual servers ανάλογα τις ανάγκες μιας επιχείρησης, να ρυθμίσετε την ασφάλεια και τη δικτύωση, αποθήκευση και διαχείριση.

Η επίσημη έναρξη της υπηρεσίας αυτής έγινε της 25 Αύγουστου 2006 και η τελευταία έκδοση ήταν στις 2013-10-09. Η συλλογή των αρχείων WSDL έγινε από την σελίδα <http://aws.amazon.com/releasenotes/Amazon-EC2> και συγκεντρώσαμε συνολικά 28 αρχεία με τελευταίο τον Οκτώβριο του 2013.



Στο παράρτημα παρουσιάζουμε πίνακα με όλες τις δημοσιευμένες εκδόσεις καθώς και πληροφορίες για αυτές. Πρέπει να επισημάνουμε ότι δεν έχουν όλες οι εκδόσεις νέες WSDL περιγραφές καθώς πολλές από αυτές υλοποιούνται από την πλευρά της Amazon και δεν προκαλούν επιπτώσεις στην διαπροσωπεία της εφαρμογής καθώς επίσης και κάποιες δεν παρέχονται.

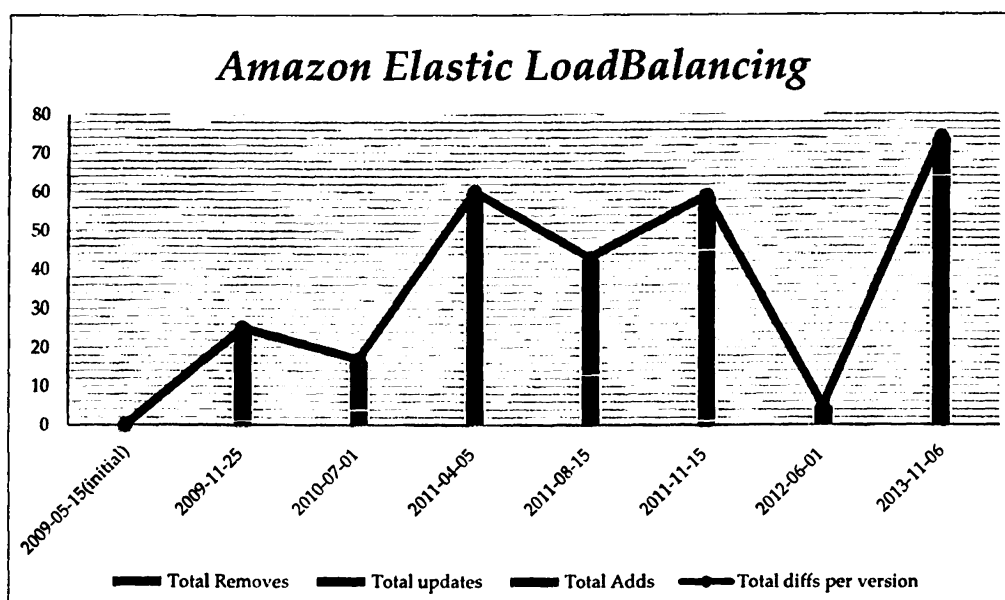


Σχήμα 4.2 Total num changes for EC2 related on time

Από το γράφημα 4.2 παρατηρούμε δυο μεγάλες αλλαγές. Η πρώτη έγινε στην έκδοση 2009-07-05 η δεύτερη στην έκδοση 2011-01-01. Και στις δυο αυτές αλλαγές είχαμε προσθήκες στο σύστημα καινούργιων σύνθετων λειτουργιών που επιτρέπουν να συνεργάζονται με καινούργιες διαθέσιμες υπηρεσίες όπως η Amazon Virtual Private Cloud (Amazon VPC) καθώς και θέματα αναβάθμισης και καινούργιων στάνταρ σε θέματα ασφάλειας και ιδιωτικότητας.

4.2.2. Amazon Elastic Load Balancing

Η υπηρεσία Elastic Load Balancing αυτόματα διανέμει την εισερχόμενη κίνηση μιας εφαρμογής σε πολλαπλούς εικονικούς εξυπηρετητές (virtual servers) της υπηρεσίας Amazon EC2 που ονομάζονται στιγμιότυπα (instances). Με την υπηρεσία αυτή επιτυγχάνονται υψηλότερα επίπεδα ανοχής σε σφάλματα στις εφαρμογές, αδιάλειπτη παροχή της απαιτούμενης υπολογιστικής ικανότητας για την εξισορρόπηση του φορτίου που απαιτείται για την ισορροπημένη κατανομή της κυκλοφορίας της εφαρμογής. Η επίσημη έναρξη της υπηρεσίας αυτής έγινε της 18 Μαΐου 2009 και η τελευταία έκδοση δημοσιεύτηκε στις 30 Ιουλίου 2013. Η συλλογή των αρχείων WSDL έγινε από την σελίδα <http://aws.amazon.com/releasenotes/> και περιλαμβάνει οκτώ WSDL αρχεία. Στο παράρτημα παρατίθεται ο πίνακας με όλο το ιστορικό των εκδόσεων



Σχήμα 4.3: Total num changes for elastic load balancing related on time

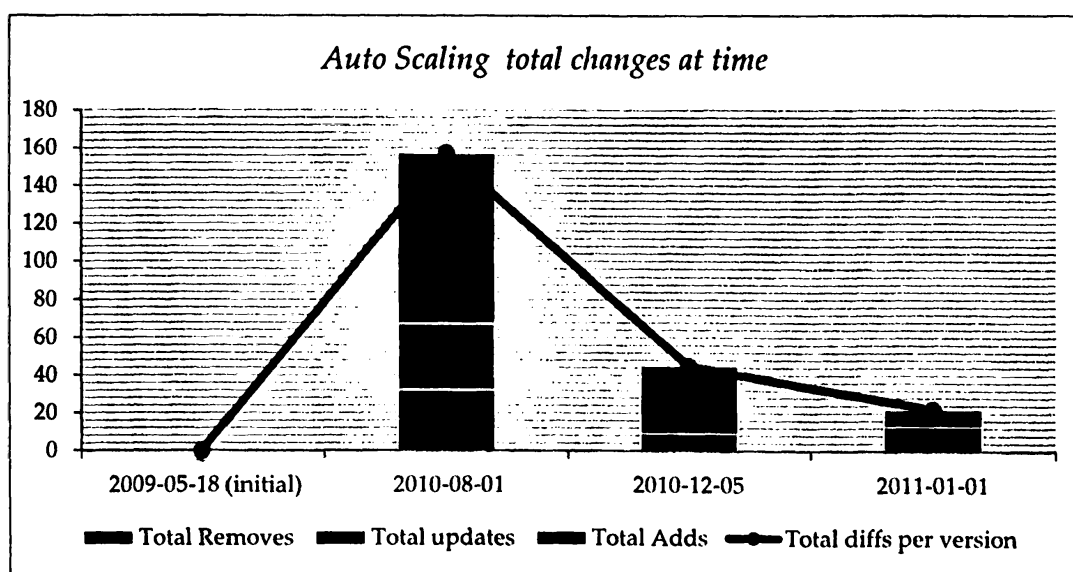
Από το γράφημα 4.3 παρατηρούμε τρεις μεγάλες αλλαγές. Η πρώτη έγινε στην έκδοση 2013-11-065 η δεύτερη στην έκδοση 2011-04-05 και η τρίτη στην έκδοση 2011-11-15. Και στις τρεις αυτές αλλαγές είχαμε προσθήκες στο σύστημα καινούργιων σύνθετων λειτουργιών όπως η υποστήριξη του IPV6, προσθήκες νέων τύπων δεδομένων, προσθήκες καινούργιων στάνταρ σε θέματα ασφάλειας και ιδιωτικότητας. Ειδικότερα στην τελευταία έκδοση γίνεται προσθήκη λειτουργίας προαιρετικής επαναδρομολόγησης της υπηρεσίας (λειτουργία Cross zone) στο



σύνολο των υπό χρήση υποδομών που μπορεί να βρίσκονται σε διαφορετικές γεωγραφικές περιοχές.

4.2.3. Amazon AutoScaling

Η υπηρεσία AutoScaling αποτελεί μια επέκταση λειτουργίας της υπηρεσίας Amazon Elastic Compute Cloud (EC2). Η υπηρεσία αυτή επιτρέπει την αναπροσαρμογή της υπολογιστικής ικανότητας που παρέχεται από την υπηρεσία Amazon EC2 προς τα πάνω ή προς τα κάτω αυτόματα ανάλογα με τις επιθυμητές συνθήκες που ορίζονται από τον χρήστη. Με την υπηρεσία αυτή μπορεί να εξασφαλιστεί ότι ο αριθμός των Virtual Servers που χρησιμοποιούνται αυξάνονται απρόσκοπτα κατά τη διάρκεια αιχμής της ζήτησης ώστε να διατηρούνται βέλτιστες επιδόσεις και να μειώνονται αυτόματα κατά τη διάρκεια περιόδων ύφεσης της ζήτησης για την ελαχιστοποίηση του κόστους και είναι ιδιαίτερα κατάλληλη για εφαρμογές που βιώνουν ωριαία, ημερήσια ή εβδομαδιαία μεταβλητότητα στη χρήση. Η επίσημη έναρξη της υπηρεσίας αυτής έγινε της 18 Μαΐου 2009. Η συλλογή των αρχείων WSDL έγινε από την σελίδα <http://aws.amazon.com/releasenotes/> και περιλαμβάνει τέσσερα WSDL αρχεία



Σχήμα 4.4: Total num changes for AutoScaling related on time

Η έκδοση που παρουσιάζει ενδιαφέρον είναι η 2010-08-01 κατά τη οποία γίνεται η σύνδεση με άλλες υπηρεσίες της Amazon όπως η Virtual Private Cloud, CloudWatch, καθώς και προσθήκες καινούργιων τύπων δεδομένων και διόρθωση θεμάτων ασφάλειας.

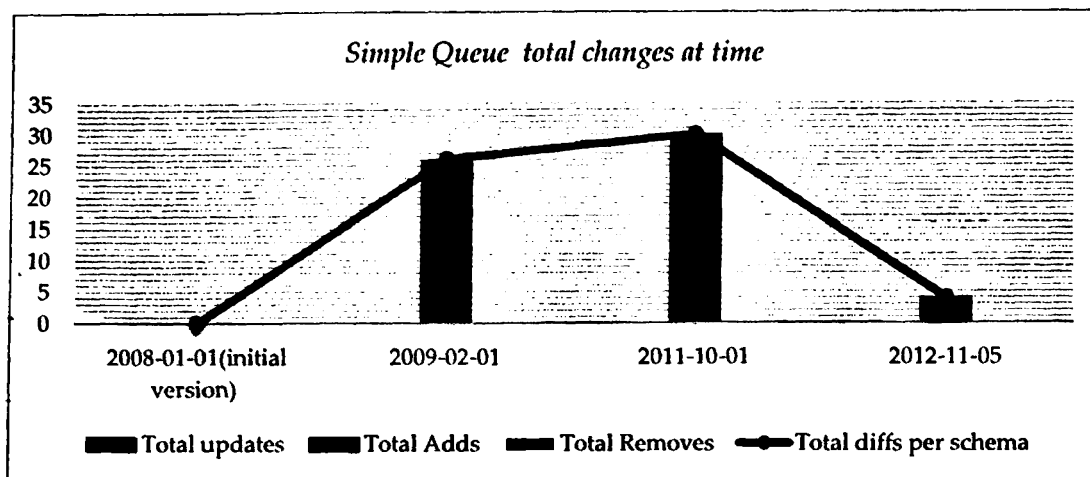
4.2.4. Amazon Simple Queue Service

Η υπηρεσία Amazon Simple Service Queue (Amazon SQS) είναι μια κατανεμημένη υπηρεσία μηνυμάτων που παρουσίασε η Amazon.com τον Απρίλιο του 2006. Υποστηρίζει την αποστολή μηνυμάτων μεταξύ Web Services εφαρμογών ως ένας τρόπος για να επικοινωνούν μέσω του Internet. Η υπηρεσία SQS παρέχει μια γρήγορη, εξαιρετικά επεκτάσιμη, πλήρως διαχειρίσιμη υπηρεσία μηνυμάτων αναμονής που επιλύει ζητήματα που προκύπτουν μεταξύ παραγωγού και καταναλωτή όπως η συνδεσιμότητα.

Με τη χρήση της υπηρεσίας αυτής, οι προγραμματιστές μπορούν απλά να μετακινήσουν δεδομένα μεταξύ διαφορετικών κατανεμημένων εφαρμογών που εκτελούν διαφορετικές εργασίες, χωρίς να χάνουν μηνύματα ή να απαιτείται οι εφαρμογές να είναι πάντα διαθέσιμες (online). Κάθε υπολογιστής στο διαδίκτυο μπορεί να προσθέσει ή να διαβάσει τα μηνύματα χωρίς να χρειάζεται η εγκατάσταση συγκεκριμένου λογισμικού ή κάποια ειδική διαμόρφωση τείχους προστασίας.

Συγκεντρώσαμε τέσσερα αρχεία καθώς τα δυο πρώτα αρχικά αρχεία (2006-04-01, 2007-05-01) περιγράφουν δεν είναι διαθέσιμα γιατί το 2008 έγινε μια μεγάλης κλίμακας ανακατασκευή στην υπηρεσία και η Amazon με ανακοίνωση της ζήτησε από τους χρήστες να μετακινηθούν στην έκδοση 2008-01-01 και δεν τα διατηρεί πλέον στο ιστορικό της. Τελευταία δημοσιοποιημένη έκδοση έγινε στις 2014-01-29





Σχήμα 4.5: Total num changes for simple queue related at time

Για την υπηρεσία SQS παρατηρούμε αλλαγές στις εκδόσεις 2009-02-01 και 2011-10-01 και στις 2 υπηρεσίες γίνονται προσθήκες νέων λειτουργιών και προστίθενται καινούργιοι κανόνες πολιτικής της εταιρείας.

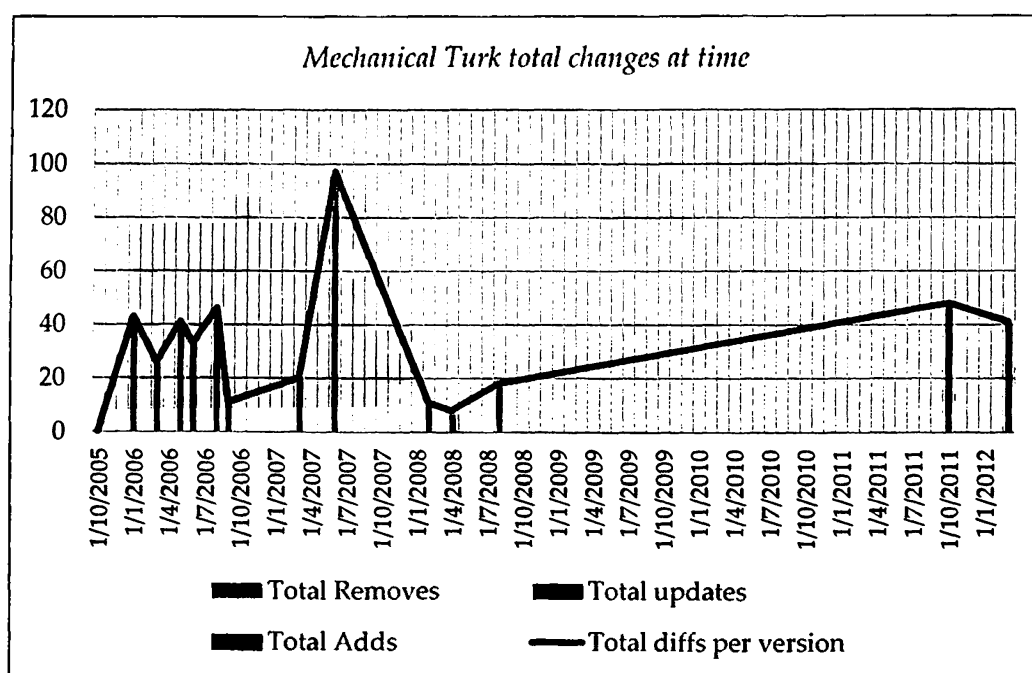
4.2.5. Amazon Mechanical Turk Service

Η υπηρεσία Amazon Mechanical Turk επιτρέπει σε άτομα ή επιχειρήσεις (γνωστοί και ως Requesters) να συντονίσουν τη χρήση της ανθρώπινης νοημοσύνης για την εκτέλεση των καθηκόντων που οι υπολογιστές δεν είναι σήμερα σε θέση να κάνουν.

Ενώ η τεχνολογία των υπολογιστών συνεχίζει να βελτιώνεται, υπάρχουν ακόμα πολλά πράγματα που τα ανθρώπινα όντα μπορούν να κάνουν πολύ πιο αποτελεσματικά από τους υπολογιστές, όπως ο προσδιορισμός των αντικειμένων σε μια φωτογραφία ή ένα βίντεο, ή την λεπτομερή έρευνα των στοιχείων ενός συνόλου δεδομένων. Παραδοσιακά, οι εργασίες αυτές διεκπεραιώνονται με την πρόσληψη ενός μεγάλου εργατικού δυναμικού πρόσκαιρης απασχόλησης η οποία είναι χρονοβόρα και δαπανηρή. Οι Αιτούντες (Requesters) δημοσιεύουν τις εργασίες προς διεκπεραίωση σε ένα κόμβο ο οποίος ονομάζεται HIT (Human Intelligence Tasks). Οι εργαζόμενοι (που ονομάζονται Providers στην ορολογία της υπηρεσίας Mechanical



Turk ή κοινώς, Turkers) μπορούν να περιηγηθούν μεταξύ των υφιστάμενων εργασιών και να την διεκπεραιώσουν βάση συγκεκριμένης οικονομικής συμφωνίας που καθορίζεται από τον αιτούντα. Για να δημιουργηθεί ένας κόμβος με μια εργασία, οι Requesters χρησιμοποιούν μια API διαπροσωπεία. Πρέπει να επισημάνουμε ότι η πολιτική της Amazon επιτρέπει οι αιτούντες να έχουν έδρα μονό στις ΗΠΑ. Η υπηρεσία ξεκίνησε δημόσια στις 2 Νοεμβρίου 2005 και, είναι ακόμα σε δοκιμαστική μορφή (beta) με τελευταία έκδοση στις 2012-03-29. Το σύνολο των δεδομενων αποτελείται από 14 αρχεία WSDL.

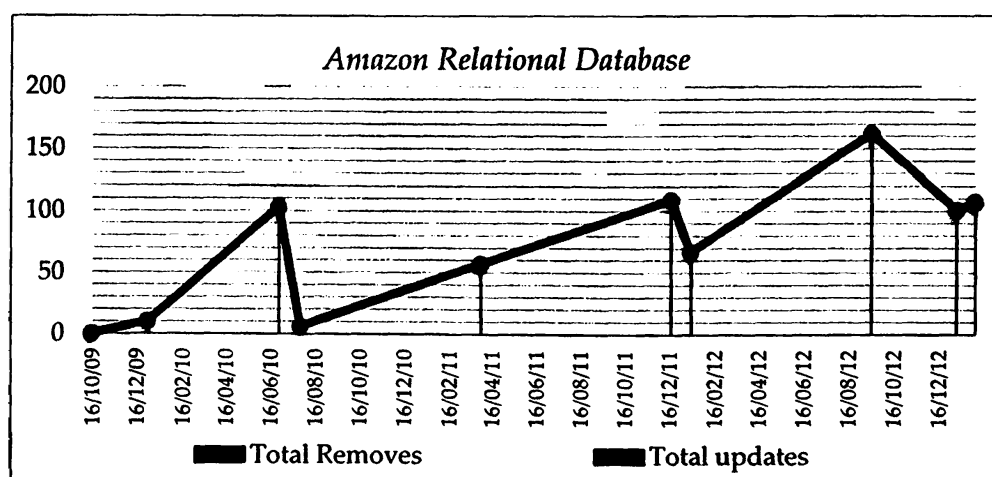


Σχήμα 4.6: Total num changes for Mechanical Turk

Από το γράφημα των συνολικών αλλαγών παρατηρούμε όχι μεγάλες διακυμάνσεις μεταξύ των εκδόσεων εκτός από την έκδοση 2007-06-21 στην οποία έχουμε τον μεγαλύτερο αριθμό αλλαγών και μετά βλέπουμε μια πτώση των αλλαγών και ειδικότερα στην έκδοση 2008-08-02 βλέπουμε να γίνονται σχεδόν μόνο διαγραφές.

4.2.6. Amazon Relational Database Service

Η Amazon Relational Database Υπηρεσία (Amazon RDS) είναι μια κατανεμημένη διαδικτυακή υπηρεσία που παρέχει ένα εύκολο τρόπο για να δημιουργήσουμε και να λειτουργήσουμε, μια ευέλικτη σχεσιακή βάση δεδομένων. Περίπλοκες και χρονοβόρες διαδικασίες, όπως η συντήρηση, η επιδιόρθωση του λογισμικού βάσεων δεδομένων, και η δημιουργία αντιγράφων ασφαλείας των βάσεων δεδομένων διαχειρίζονται αυτόματα κερδίζοντας χρόνο και προσφέροντας σημαντικά οικονομικά οφέλη για τους χρήστες τις υπηρεσίας. Επίσης παρέχεται η δυνατότητα αλλαγής μεγέθους χωρητικότητας. Η υπηρεσία Amazon RDS κυκλοφόρησε για πρώτη φορά στις 22 Οκτωβρίου 2009 υποστηρίζοντας την βάση δεδομένων MySQL και ακολούθησε η υποστήριξη της Oracle Database τον Ιούνιο του 2011, της Microsoft SQL Server τον Μάιο του 2012 της και PostgreSQL, το Νοέμβριο του 2013.



Σχήμα 4.7: Total num changes for Relational Database over time

Εδώ παρατηρούμε τις περισσότερες αλλαγές στην έκδοση 2012-09-17 στην οποία ενσωματώνονται αλλαγές που υποστηρίζουν τροποποιήσεις τον SQL Server 2012 DB, RDS Oracle micro DB Instances.



ΚΕΦΑΛΑΙΟ 5. ΑΞΙΟΛΟΓΗΣΗ ΤΩΝ ΝΟΜΩΝ ΤΟΥ LEHMAN

5.1 Νόμοι του Lehman και Ταξινόμηση των συστημάτων λογισμικού

5.2 Ορισμοί των Νόμων του Lehman

5.3 Αποτελέσματα μετρήσεων

5.4 Εκτίμηση της ισχύος των νόμων του Lehman βάση των μετρήσεων για τα Amazon Web Services

Οι νόμοι του Lehman που αφορούν την εξέλιξη λογισμικού διατυπώθηκαν για πρώτη φορά στις αρχές της δεκαετίας του 1970, στην μελέτη των inBelady, Lehman για την εξέλιξη του συστήματος OS/360 της IBM. [15]

Ο Lehman ισχυρίστηκε ότι το λογισμικό είναι ένα εργαλείο που προσομοιώνει τον πραγματικό κόσμο. Δεδομένου ότι ο πραγματικός κόσμος υπόκειται στην δυναμική της εξέλιξης, έτσι και ένα λογισμικό θα πρέπει να αλλάζει και να ενημερώνεται ώστε να είναι σε θέση να αντικατοπτρίζει τις αλλαγές στον πραγματικό κόσμο [17]. Ως εκ τούτου, διορθώσεις, ενημερώσεις και μικρές ή μεγάλες αλλαγές είναι αναπόσπαστα μέρη του κύκλου ζωής του λογισμικού.

Οι νόμοι της εξέλιξης του λογισμικού αναλύθηκαν, τροποποιήθηκαν, και επαναδιατυπώθηκαν από τον Lehman και τους συνεργάτες στα επόμενα τριάντα χρόνια για να αποκτήσουν την τελική τους μορφή σαν ένα σύνολο από οκτώ κανόνες. Οι τρεις πρώτοι νόμοι δημοσιεύθηκαν το 1974, οι επόμενα τρεις έξι χρόνια αργότερα, και οι τελευταίοι δύο το 1996.



5.1. Ταξινόμηση των συστημάτων λογισμικού

Ο Lehman και η ομάδα του μέχρι τις αρχές τις δεκαετίας του '80 κατηγοριοποιούσαν τα συστήματα λογισμικού με βάση το μέγεθος τους, σε μεγάλα συστήματα (Large programmes), και σε όχι μεγάλα συστήματα (non Large programmes) [16].

Προσπαθώντας να εκλεπτύνουν τις κατηγορίες αυτές, οι οποίες τους δημιουργούσαν προβλήματα και μετά από μελέτη βασισμένοι στην παρατήρηση ότι «τουλάχιστον κάθε πρόγραμμα αποτελεί ένα μοντέλο ενός μεγαλύτερου θεωρητικού μοντέλου που με την σειρά του αποτελεί κομμάτι ενός μεγαλύτερου αφαιρετικού μοντέλου κάποιου τμήματος του κόσμου, ή κάποιου σύμπαντος του λόγου» [16] κατέληξαν να ταξινομήσουν τα συστήματα λογισμικού - προγράμματα σε τρεις διαφορετικές κατηγορίες:

- Τα S-type προγράμματα που έχουν συνταχθεί σύμφωνα με μια ακριβώς προσδιορισμένη περιγραφή του τι μπορεί να κάνει το πρόγραμμα. Ουσιαστικά, αυτά τα συστήματα είναι εκείνα των οποίων οι ενδιαφερόμενοι έχουν κατανοήσει το πρόβλημα και ξέρουν ακριβώς τι χρειάζεται να κάνουν και ποιες προδιαγραφές να υλοποιήσουν για να το επιλύσουν. Ένα παράδειγμα ενός τύπου αυτού του συστήματος είναι ένα που εκτελεί μαθηματικούς υπολογισμούς όπου τόσο η επιθυμητή λειτουργικότητα και πώς να το πετύχουμε είναι καλά κατανοητή εκ των προτέρων και τα οποία δεν είναι πιθανό να αλλάξουν. Συστήματα των τύπων αυτών είναι τα απλούστερα μεταξύ των τριών τύπων και υπόκεινται λιγότερο στους εξελικτικούς νομούς του Lehman.
- Τα P-types προγράμματα είναι γραμμένα για να υλοποιούν συγκεκριμένες διαδικασίες που καθορίζουν ολοκληρωτικά τη λειτουργία επιτελεί το πρόγραμμα (π.χ. σκάκι). Αυτά είναι τα συστήματα που οι ενδιαφερόμενοι γνωρίζουν το τελικό αποτέλεσμα που είναι απαραίτητο να παραχθεί, αλλά δεν μπορούν να περιγράψουν με βέλτιστο τρόπο πώς να φτάσουν στην επιθυμητή λύση. Στα προγράμματα αυτά το ενδιαφέρον δεν επικεντρώνεται στην περιγραφή του προβλήματος, αλλά στην αξία και την εγκυρότητα των λύσεων που λαμβάνονται από αυτό. Για αυτό τα προβλήματα αυτά είναι πολύ πιθανό να περιέλθουν σε



διαδικασία ατερμόνων αλλαγών η να καθίστανται ολοένα λιγότερο λειτουργικά και κοστοβόρα. [16].

- Ο τελευταίος τύπος λογισμικών συστημάτων που προτείνεται από τον Lehman είναι τα E-types (embedded types). Τα E-types προγράμματα είναι γραμμένα για να υλοποιούν δραστηριότητες του πραγματικού κόσμου. Το πώς πρέπει να συμπεριφέρονται συνδέεται στενά με το περιβάλλον στο οποίο λειτουργούν. Αυτά τα προγράμματα θα πρέπει να προσαρμόζονται στις διαφορετικές απαιτήσεις και τις συνθήκες με το περιβάλλον με το οποίο αλληλεπιδρούν και να βελτιώνονται συνεχώς αν πρόκειται να παραμείνουν αποτελεσματικά σε ένα μεταβαλλόμενο κόσμο.

Η μελέτη των E-type συστημάτων είναι σημαντική καθώς η πλειοψηφία των επιχειρήσεων και των οργανισμών βασίζονται πάνω σε αυτά τα συστήματα. Τα συστήματα αυτά υπόκεινται σε συχνές αλλαγές για να εξελιχθούν αναλογικά με το περιβάλλον στο οποίο λειτουργούν, και συνεπώς είναι αυτά τα συστήματα όπου υπόκεινται στους νόμους του Lehman [16], [17].

Προφανώς οι Web Services που μελετάμε στην παρούσα εργασία ανήκουν ως επί το πλείστον στην κατηγορία E-type, και αυτή αποτελεί βασική υπόθεση της μελέτης μας.



5.2. Ορισμοί των Νόμων του Lehman

Πίνακας 5.1: Οι νόμοι της εξέλιξης λογισμικού όπως δημοσιεύθηκαν το 2006
[17]

I) LAW OF CONTINUING CHANGE An E-type system must be continually adapted, else it becomes progressively less satisfactory in use
II) LAW OF INCREASING COMPLEXITY As an E-type is changed its complexity increases and becomes more difficult to evolve unless work is done to maintain or reduce the complexity.
III) LAW OF SELF REGULATION Global E-type system evolution is feedback regulated.
IV) LAW OF CONSERVATION OF ORGANIZATIONAL STABILITY The work rate of an organization evolving an E-type software system tends to be constant over the operational lifetime of that system or phases of that lifetime.
V) LAW OF CONSERVATION OF FAMILIARITY In general, the incremental growth (growth rate trend) of E-type systems is constrained by the need to maintain familiarity.
VI) LAW OF CONTINUING GROWTH The functional capability of E-type systems must be continually enhanced to maintain user satisfaction over system lifetime.
VII) LAW OF DECLINING QUALITY Unless rigorously adapted and evolved to take into account changes in the operational environment, the quality of an E-type system will appear to be declining.
VIII) LAW OF FEEDBACK SYSTEM E-type evolution processes are multi-level, multi-loop, multi-agent feedback systems

1ος Νόμος του Lehman: Συνεχής αλλαγή (Continuing Change)

«An E-type system must be continually adapted, else it becomes progressively less satisfactory in use»

Ένα E-type σύστημα πρέπει συνεχώς να προσαρμόζεται, διαφορετικά η χρησιμότητά του θα φθίνει μέχρι το σημείο που να είναι συμφερότερο το σύστημα να κατασκευαστεί εκ νέου.

Δεδομένου ότι το περιβάλλον του πραγματικού κόσμου είναι συνεχώς μεταβαλλόμενο (π.χ. νέοι νόμοι και κανονισμοί προσθέτονται, όσο αυξάνεται η εμπειρία των χρηστών νέες προσδοκίες και νέες λειτουργίες αναγκαίες απαιτούνται από τους χρήστες), προκειμένου το σύστημα να συνεχίσει να χρησιμοποιείται, θα πρέπει να προσαρμοστεί (δηλαδή να εξελιχθεί). Εάν αυτό δεν συμβεί, το σύστημα γίνεται λιγότερο ικανοποιητικό [17].

Τρόποι μέτρησης για την εκτίμηση της ισχύος του νόμου. Για να διαπιστώσουμε την ισχύ του νόμου στα web services, πρέπει να αποδείξουμε ότι το λογισμικό δείχνει σημάδια εξέλιξης όσο περνάει ο καιρός. Πιθανές μετρήσεις που εξετάζουμε περιλαμβάνουν: (α) το συνολικό αριθμό των αλλαγών μελετώντας το γράφημα των συνολικών αλλαγών και (β) την ανάλυση των αλλαγών σε επιμέρους κατηγορίες, όπως τις γραφικές παραστάσεις των αλλαγών των λειτουργιών από την οποία συμπεραίνουμε την εξέλιξη των λειτουργιών που προσφέρονται από τα services που εξετάζουμε (γ) και τις αλλαγές στους τύπους δεδομένων.

2ος Νόμος του Lehmann: Αυξανόμενη πολυπλοκότητα (Increasing complexity)

«As an E-type is changed its complexity increases and becomes more difficult to evolve unless work is done to maintain or reduce the complexity»

Καθώς ένα πρόγραμμα εξελίσσεται η πολυπλοκότητά του αυξάνεται και γίνεται πιο δύσκολη η διαδικασία εξέλιξης, του εκτός και αν γίνουν εργασίες συντήρησης η μείωσης της πολυπλοκότητας του.



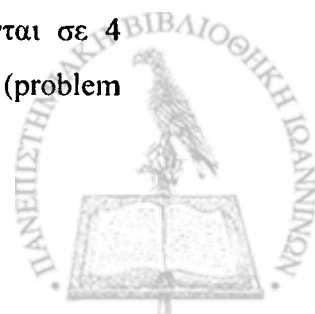
Η βασική ιδέα στην οποία βασίζεται αυτός ο νόμος είναι ότι η πολυπλοκότητα αυξάνεται λόγω της προσθήκης των αλλαγών που χρειάζονται, ή για να επιτευχθεί αύξηση της λειτουργικότητας του συστήματος, ή να επιτευχθεί ικανοποίηση των αναγκών που προκύπτουν από τις μεταβολές στους επιχειρησιακούς τομείς που αυτό αλληλεπιδρά. Αυτό οδηγεί σε αύξηση της εσωτερικής διασυνδεσιμότητας και, ως εκ τούτου, στην επιδείνωση της δομής του συστήματος, αυξάνοντας έτσι την εσωτερική ανισορροπία του συστήματος. Ακολουθώντας, αυτό οδηγεί σε αύξηση της πολυπλοκότητας των εσωτερικών και εξωτερικών διεπαφών σε όλα τα επίπεδα [17].

Η βασική έννοια του νόμου συμπληρώνεται με την προσπάθεια ελέγχου της πολυπλοκότητας που είναι σε μεγάλο βαθμό οπισθοδρομική (anti-regressive) [17]. Η δραστηριότητα ελέγχου της πολυπλοκότητας περιλαμβάνει την κατάργηση επαναλαμβανόμενου κώδικα (repetitive code), αναδιάρθρωση, τεκμηρίωση, ενημέρωση του κώδικα και πρέπει να γίνεται ανά τακτά χρονικά διαστήματα.

Σε γενικές γραμμές, οι δραστηριότητες αυτές έχουν μικρή ή καμία επίδραση στην λειτουργικότητα, την απόδοση ή άλλες ιδιότητες του λογισμικού και γενικά τα άμεσα οφέλη είναι σχετικά μικρά. Ωστόσο μακροπρόθεσμες επιπτώσεις μπορεί να είναι σημαντικές, τόσο ώστε σε κάποιο στάδιο, να κάνουν τη διαφορά μεταξύ της επιβίωσης του συστήματος ή της αντικατάστασής του. Το ισοζύγιο αυτών των δυο δραστηριοτήτων που πρέπει να συνυπάρχουν και να συνδιαμορφώνουν ταυτόχρονα ένα σύστημα είναι τέτοιο ώστε αυτό να αποκτά την δυναμική για σωστά εξελικτικά βήματα, και αποτελεί το κρίσιμο σημείο το νόμου.

Σημαντικό βήμα στην αξιολόγηση μας των Web Services είναι το πώς γίνεται η μέτρηση της πολυπλοκότητας.

Πρέπει να επισημάνουμε ότι η διατύπωση του 2^{ου} νόμου δεν καθορίζει συγκεκριμένα τον τρόπο με τον οποίο υπολογίζεται η πολυπλοκότητα. Το θέμα έχει τεθεί και ερευνάτε από την επιστημονική κοινότητα. Στην προσπάθεια αποσαφήνισης του προβλήματος έχουν δοθεί λύσεις οι οποίες λειτουργούν βέλτιστα κατά περίπτωση μελέτης. Δίνοντας ένα παράδειγμα οι Fenton – Pfleeger [18] αναφέρονται σε 4 κατηγορίες πολυπλοκότητας) την πολυπλοκότητα προβλημάτων (problem



complexity) που χαρακτηρίζονται από την υπολογιστική πολυπλοκότητα της λύσης του προβλήματος 2) αλγοριθμική πολυπλοκότητα (algorithmic complexity) που μελετά την πολυπλοκότητα του αλγορίθμου που επιλύει το πρόβλημα 3) την πολυπλοκότητα της διάρθρωσης – δομής (structural complexity) του προγράμματος που προκύπτει από την μέτρηση των διαγραμμάτων ροής αυτού ή από την ιεραρχία των κλάσεων του 4) τη γνωστική πολυπλοκότητα που αναφέρεται στην υπολογισμό της προσπάθειας να κατανοηθεί η λειτουργία του λογισμικού.

Για τους Lehman-Ramil [17] η πολυπλοκότητα ενός συστήματος αποτελείται από ένα σύνολο παραμέτρων που πρέπει να υπολογιστούν για ποιο ακριβή μέτρηση της:

- Πολυπλοκότητα της εφαρμογής και της λειτουργικότητας.
- Πολυπλοκότητα των προδιαγραφών και των απαιτήσεων.
- Πολυπλοκότητα αρχιτεκτονικής.
- Πολυπλοκότητα σχεδιασμού.
- Πολυπλοκότητα δομής.

Η μέτρηση της πολυπλοκότητας λαμβάνοντας υπόψη το σύνολο των παραπάνω αξιολογήσεων είναι δύσκολο να οριστεί και να μετρηθεί, ειδικά όταν αυτού του είδους μετρήσεις πρέπει να γίνουν με βάση δεδομένα που προκύπτουν από ηλεκτρονικά αρχεία περιγραφής WSDL. Επομένως, η προσέγγισή μας στο να υπολογίσουμε την πολυπλοκότητα δίνει έμφαση στην πληροφορία που παίρνουμε από την διάρθρωση-δομή του WSDL αρχείου. Έχουν προταθεί αρκετοί τρόποι μέτρησης της δομικής πολυπλοκότητας όπως ο αριθμός των γραμμών του κώδικα (LOC), η κυκλική πολυπλοκότητα (cyclomatic complexity) που βασίζεται στην θεωρία γραφημάτων, πολυπλοκότητα συζεύξεων (coupling), η οποία βασίζεται στον βαθμό εξαρτήσεων μεταξύ των δομικών στοιχείων του προγράμματος κ.α..

Ποιο συγκεκριμένα ο Lehman [16] διατυπώνει σαν συμπέρασμα εμπειρικής μελέτης τον τύπο: $IR = OMC/NM$ με OMC να είναι ο αριθμός των τροποποιημένων στοιχείων της παλαιάς έκδοσης (updated operations) και NM ο αριθμός των νέων στοιχείων που προστεθήκαν (operations added), ως δείκτη της πολυπλοκότητας του συστήματος. Απλοποιώντας την ερμηνεία του παραπάνω τύπου η πολυπλοκότητα είναι η αναλογία



της προσπάθειας (effort) για να τροποποιηθούν στοιχεία της παλαιότερης έκδοσης, προς την προσθήκη νέων στοιχείων του συστήματος στην νεότερη έκδοση.

Το μέγεθος των πόρων που χρειάζονται να διατεθούν στην εξέλιξη του λογισμικού σε ορισμένες οργανώσεις είναι πολύ σημαντική με συμμετοχή δεκάδων ή ακόμη και εκατοντάδων προγραμματιστών. Είναι σημαντικό για τους διαχειριστές να είναι σε θέση να εκτιμήσουν το μέγεθος της προσπάθειας και των συναφών χρονοδιαγραμμάτων που απαιτείται την εξέλιξη του λογισμικού. Ένα μοντέλο υπολογισμού της προσπάθειας, σχετίζεται επίσης με την αξιολόγηση της παραγωγικότητας προγραμματισμού, τις τάσεις της, ακόμη και να διαπιστωθεί το κατά πόσον θα μειωθεί όσο ένα σύστημα μεγαλώνει ηλικιακά και γίνεται περιπλοκότερο. Μοντέλα εκτίμησης του ρυθμού εργασίας (work rate) μπορούν επίσης να διαδραματίσουν σημαντικό ρόλο στην απόφαση πότε είναι σκόπιμο να αντικατασταθεί ένα σύστημα λόγω παλαιότητας. Η προσπάθεια που απαιτείται για την αλλαγή, την προσθήκη, την διαγραφή και την αντικατάσταση πηγαίου κώδικα μπορεί να είναι ένα σημαντικό κλάσμα του συνολικής προσπάθειας εξέλιξης ενός συστήματος [20].

Στην περίπτωση των Amazon Web Services για να επαληθεύσουμε τον 2^ο νόμο επικεντρωνόμαστε στις εξής παρατηρήσεις: Πρώτα προσπαθούμε να επιβεβαιώσουμε την ουσιαστική έννοια του νόμου, προσπαθώντας να βρούμε εκδόσεις (versions) κατά τις οποίες έχουμε πτώση στο μέγεθος του συστήματος ή πτώση στο ρυθμό ανάπτυξης του συστήματος. Θεωρούμε ότι σε συνδυασμό με την γενικότερη τάση των συστημάτων να αυξάνεται το μέγεθος τους και ο ρυθμός ανάπτυξης τους η παρατήρηση αυτή αποτελεί καλή ένδειξη ότι έχουμε προσπάθεια συγκράτησης της πολυπλοκότητας του συστήματος.

Σαν δεύτερο αποδεικτικό στοιχείο ισχύος του νόμου θα χρησιμοποιήσουμε τις μετρήσεις της πολυπλοκότητας IR ανά βήμα διαδοχικών εκδόσεων που έχουμε αναλογία των τροποποιήσεων σε λειτουργίες του Web Service μεταξύ δυο διαδοχικών εκδόσεων προς την διαφορά των προστιθέμενων λειτουργιών στο Web Service.



3ος Νόμος του Lehmann: Αυτορρύθμιση συστήματος (Self Regulation)

«Global E-type system evolution is feedback regulated»

Η εξέλιξη μεγάλων συστημάτων είναι μια αυτορυθμιζόμενη διεργασία. Η βασική ιδέα είναι ότι στην εξέλιξη ενός προγράμματος λαμβάνουν χώρα διεργασίες θετικής αλλά και αρνητικής τροφοδότησης. Όσο οι ανάγκες που πρέπει να εξυπηρετήσει το σύστημα αυξάνονται λόγω των καινούργιων απαιτήσεων για περισσότερη λειτουργικότητα από τους χρηστές τους, τόσο το μέγεθος του αυξάνει, και ταυτόχρονα λαμβάνουν χώρα εργασίες συντήρησης του συστήματος, διόρθωσης από λάθη και αστοχίες λογισμικού, καθώς και βελτιώσεις της εσωτερικής ποιότητας του λογισμικού [24].

Συνυπολογίζοντας τις δυο αυτές παραμέτρους της θετικής και αρνητικής τροφοδότησης του καταλαβαίνουμε ότι η ανάπτυξη του συστήματος δεν μπορεί να συνεχίζει να αυξάνεται διαρκώς με τον ίδιο ρυθμό και αυτό που περιμένουμε να δούμε είναι ανά διαστήματα να διακόπτεται από εκδόσεις συντήρησης του συστήματος. Αυτή η τάση είναι παρά πολύ συχνά εμφανίσιμη, τόσο που ο Lehman την χαρακτήρισε “*patterns of incremental growth*”, επαναλαμβανόμενα πρότυπα ανάπτυξης [17].

Μετρικές για την επαλήθευση του νόμου. Τυπικά η επαλήθευση του νόμου προκύπτει από τις μετρήσεις της ανάπτυξης του συστήματος (growth) και των παρατηρήσεων στο διάγραμμα που προκύπτει για διακυμάνσεις (ripples). Για κάθε εξελικτικό βήμα η ανάπτυξη του συστήματος ορίζεται σαν η διάφορα μεγέθους μεταξύ 2 διαδοχικών εκδόσεων. Η παρουσία διακυμάνσεων υποδεικνύει ύπαρξη θετικών και αρνητικών μηχανισμών τροφοδότησης του συστήματος με τον θετικό μηχανισμό να επεκτείνει το σύστημα και τον αρνητικό να εμπλέκεται στην μείωση του ρυθμού ανάπτυξης αυτού.

Για να επαληθεύουμε τον νόμο εξετάζουμε τις μετρήσεις που προκύπτουν από την ανάπτυξη του συστήματος (growth) που σχετίζεται με τις λειτουργίες (operations) και



τους τύπους δεδομένων (complex types και elements) που περιγράφονται στα WSDL αρχεία.

Κριτήρια που επαληθεύουν το νόμο είναι:

- Επαναλαμβανόμενα μοτίβα: στην ανάπτυξη των operation (growth at operations) και των types (growth at types) του συστήματος, αποτελούν ένδειξη για την ύπαρξη μηχανισμού, που προκαλεί την εμφάνιση των επαναλαμβανόμενων μοτίβων κατά την διάρκεια ανάπτυξης ενός συστήματος Web Service.
- Ενδείξεις που αποκαλύπτουν την ύπαρξη ενός μηχανισμού αρνητικής τροφοδότησης όπως εκδόσεις ή χρονικοί περίοδοι κατά τις οποίες έχουμε εργασίες συντήρησης και βελτίωσης των συστημάτων και εμφανίζονται στα γραφήματα των μετρήσεων της ανάπτυξης ή σαν εμφάνιση αρνητικής, μηδενικής, ή γενικότερα μικρότερης ανάπτυξης σε σχέση με την μέση τιμή της ανάπτυξης.

Επίσης από την στιγμή που τα δεδομένα μας δείχνουν μείωση του ρυθμού ανάπτυξης του συστήματος περιμένουμε να δούμε και τα επαναλαμβανόμενα μοτίβα σε περιοχές με μειωμένες τιμές.

4^{ος} Νόμος του Lehman: Διατήρηση της Οργανωτικής Σταθερότητας (Conservation of Organizational Stability)

«The work rate of an organization evolving an E-type software system tends to be constant over the operational lifetime of that system or phases of that lifetime»

Ο ρυθμός εργασιών ενός οργανισμού που εξελίσσει ένα E-type σύστημα λογισμικού τείνει να είναι σταθερός κατά τη διάρκεια του χρόνου λειτουργίας του ή των φάσεων που περνά κατά την διάρκεια της ζωής του. Ο νόμος αυτός είναι γνωστός και ως «νόμος του αμετάβλητου ρυθμού εργασίας» («invariant work rate»).

Κατανοώντας την ουσία του νόμου ο ρυθμός βελτίωσης της παραγωγικότητας από αποφάσεις ή ενέργειες που παίρνονται από τον οργανισμό, ισοσταθμίζεται από την αυξανόμενη πολυπλοκότητα του λογισμικού όσο περνά ο καιρός και από εξωγενείς



δυνάμεις που μπορεί να επιδράσουν στην εξέλιξη του λογισμικού (π.χ. διαθεσιμότητα των πόρων, διαθεσιμότητα προσωπικού, κλπ).

Πρέπει να επισημάνουμε ότι νόμος αυτός έχει επαναδιατυπωθεί περιλαμβάνοντας στην μετέπειτα διατύπωση του την πρόβλεψη ότι ο ρυθμός της παραγωγικότητας μπορεί να διαταραχτεί από απότομες αλλαγές που μπορεί να προκληθούν από επείγουσες υποχρεωτικές προς το σύστημα επεμβάσεις εξωγενών παραγόντων (π.χ. Υ2Κ πρόβλημα, συγχωνεύσεις οργανισμών κ.α.).

Μετρικές για την επαλήθευση του νόμου. Οι παρατηρήσεις επί των οποίων ο νόμος αυτός βασίζεται χρονολογούνται από τα τέλη της δεκαετίας του εβδομήντα. Βασισμένοι στα στοιχεία που συγκεντρώθηκαν μέσω του FEAST PROJECT παρατήρησαν ότι ο ρυθμός έργου μπορεί να μετρηθεί είτε με την προσφορά εργασίας είτε με το παραγόμενο έργο. Το πιο προφανές μέτρο είναι ο χρόνος που ένας εργαζόμενος ή εργαζόμενοι έχουν δαπανήσει στην ενασχόληση τους με την εξέλιξη του συστήματος. Στο πλαίσιο της μελέτης όμως τα συγκεκριμένα δεδομένα είναι πολύ δύσκολο να βρεθούν και αυτά που μπορεί να βρεθούν απαιτούν εξονυχιστική ανάλυση με αμφιβόλου ακριβείας αποτελέσματα. [21].

Στην περίπτωση της μελέτης των Web Services θα χρησιμοποιήσουμε την προσέγγιση που χρησιμοποιεί ο Lehman σε μια από τις εργασίες του [16] χρησιμοποιώντας την αναλογία των τροποποιημένων στοιχείων (operation και types) της εξεταζόμενης έκδοσης προς την χρονική περίοδο σε ημέρες που μεσολάβησαν μεταξύ αυτής και της προηγούμενης έκδοσης. Με την μετρική αυτή ορίζει τον αριθμό των στοιχείων που άλλαξαν ανά ημέρα:

$$\text{Modules per day} = \frac{\text{number of modules changed in a released}}{\text{release interval in days}} [5.3].$$



5^{ος} Νόμος του Lehman: Διατήρηση της οικειότητας (Conservation of familiarity)

«In general, the incremental growth (growth rate trend) of E-type systems is constrained by the need to maintain familiarity»

Σε γενικές γραμμές, η σταδιακή αύξηση (η τάση του ρυθμού αύξησης) των E-type συστημάτων περιορίζεται από την ανάγκη να διατηρηθεί η οικειότητα με το σύστημα. Καθώς ένα σύστημα εξελίσσεται, όλοι όσοι εμπλέκονται σε αυτό, χρήστες, προγραμματιστές, πωλητές, κ.ά., πρέπει να παραμένουν οικείοι με το περιεχόμενό του και της λειτουργίες του.

Η βασική ιδέα του νόμου είναι ότι όταν συμβεί μια μεγάλη εξελικτική αλλαγή στο σύστημα τότε ο μηχανισμός τροφοδότησης του συστήματος τείνει να μειώνει τις αλλαγές στις επόμενες εκδόσεις ώστε να δοθεί ο κατάλληλος χρόνος και προσπάθεια από τους εμπλεκόμενους στο σύστημα να απορροφήσουν και να κατανοήσουν το περιεχόμενο των αλλαγών.

Με δεδομένη την αυξανόμενη πολυπλοκότητα του συστήματος, τη λειτουργία του και τη λειτουργικότητά του, το να επιτύχεις νέα εξοικείωση με το σύστημα, μετά από πολλές αλλαγές, προσθήκες και αφαιρέσεις και την αποκατάσταση της εξοικείωσης με την προ-αλλαγής κατάσταση του συστήματος, γίνεται όλο και πιο δύσκολη με το πέρασμα του χρόνου [17]. Η προηγούμενη παρατήρηση και μετά από εμπειρική μελέτη πολλών συστημάτων οδήγησε Lehman-Ramil να συσχετίσουν το 5ο νόμο με το συμπέρασμα ότι σε βάθος χρόνου υπάρχει μείωση του ρυθμού αύξησης της ανάπτυξης του συστήματος [17].

Επίσης μια σημαντική παρατήρηση που αναφέρεται στο [22] είναι ότι μετά από μια νέα έκδοση με μεγάλο αριθμό σημαντικών αλλαγών ακολουθούν εκδόσεις με διορθώσεις και εργασίες συντήρησης του συστήματος.

Μετρικές για την επαλήθευση του νόμου. Μια λίστα μετρικών που μπορούν να χρησιμοποιηθούν προτείνονται από τους Lehman-Ramil [17] όπως και η μεθοδολογία της δημιουργίας των μετρικών αυτών: συλλογή δεδομένων για τις αλλαγές στο σύστημα και σχεδίαση ενός μοντέλου για την ανάπτυξη του συστήματος ώστε να σκιαγραφηθεί η τάση της ανάπτυξης του συστήματος. Στοιχεία που μπορεί να μετρηθούν και να εκτιμηθούν για τις αλλαγές ενός συστήματος είναι αντικείμενα,



γραμμές κώδικα, υποσυστήματα, είσοδοι – έξοδοι, διασυνδέσεις, κ.α.[17]. Μετρικές που προτείνονται και από τους Guowu Xie, et.al [22] και υιοθετούμε στην εργασία μας είναι:

- Η μέτρηση της ανάπτυξης του συστήματος
- Μέτρηση του συνολικού αριθμού των αλλαγών για κάθε καινούργια έκδοση
- Επίσης θα εξετάσουμε και θα λάβουμε υπόψη μας τις επιμέρους μετρήσεις των αλλαγών που αφορούν τα operations και τα types των Web Services

Για να επαληθεύουμε το νομό περιμένουμε να δούμε:

- ότι η ανάπτυξη του σχήματος κατά την διάρκεια του χρόνου δεν αυξάνεται και πιο συγκεκριμένα να δούμε τάσεις σταθερότητας η μείωσης.
- Θα προσπαθήσουμε να εντοπίσουμε μοτίβα στις εκδόσεις που ακολουθούν, εκδόσεις μεγάλων αλλαγών με μικρότερο ρυθμό ανάπτυξης και εκδόσεις διορθωτικές και πιο συγκεκριμένα ύπαρξη απότομων μεταβολών μετά από υψηλές τιμές (peaks) στο γράφημα.

6^{ος} Νόμος του Lehmann: Συνεχής ανάπτυξη (Continuing Growth)

«The functional capability of E-type systems must be continually enhanced to maintain user satisfaction over system lifetime»

Οι λειτουργίες ενός συστήματος πρέπει διαρκώς να αυξάνονται προκειμένου να μείνει ο χρήστης ικανοποιημένος καθ' όλη τη ζωή του συστήματος.

Με την πρώτη ανάγνωση του νόμου φαίνεται ότι δεν διαφέρει και πολύ από τον πρώτο νόμο, αλλά στην ουσία ο Lehman [23] εξηγεί ότι στην πραγματικότητα, οι δύο νόμοι εκφράζουν διαφορετικά φαινόμενα. Ο πρώτος νόμος αναγνωρίζει, μεταξύ άλλων, ότι, για παράδειγμα, οι αλλαγές στον επιχειρησιακό τομέα με το πέρασμα του χρόνου είναι πιθανό να ακυρώσουν λειτουργίες και παραδοχές που είναι ενσωματωμένες στο σύστημα λογισμικού και έτσι να προκληθεί απροσδόκητη συμπεριφορά κατά την εκτέλεση. Καθώς οι χρήστες θα αντιληφθούν την επακόλουθη μη ικανοποιητική ή μη ορθή λειτουργία του συστήματος θα απαιτήσουν μια διόρθωση. Δηλαδή ο νόμος αντανακλά την ανατροφοδότηση του συστήματος από τις



επιπτώσεις της χρήσης του συστήματος, και συνεπώς η αλλαγή είναι αναπόφευκτη στο λογισμικό όπως είναι σε κάθε ενεργό σύστημα [23].

Από την άλλη ο 6^{ος} νόμος αντλεί την διαφορετικότητα του από άλλη πηγή. Το γεγονός ότι κατά την αρχική ανάπτυξη ενός συστήματος (είτε σχεδιασμός εξ'αρχής, είτε επαναχρησιμοποίηση κάποιου παλαιού) ή αναβάθμιση ενός συστήματος, το πρώτο και βασικότερο βήμα είναι η λεπτομερής περιγραφή του μοντέλου αυτού και του επιχειρησιακού πεδίου δράσης του, και συνυπολογίζοντας του περιορισμούς που επιδρούν την δημιουργία του συστήματος όπως, ο προϋπολογισμός του συστήματος, ημερομηνίες παραδόσεις συστήματος κ.α., οδηγούν τις απαιτήσεις και τις προδιαγραφές σε ότι αφορά το εύρος των λειτουργιών του συστήματος σε οριοθέτηση και αποκλεισμό από την αρχική έκδοση. Όσο περνά ο καιρός αυτές οι λειτουργίες προστίθενται σε επόμενες εκδόσεις με αποτέλεσμα της αναπόφευκτης ανάπτυξης του συστήματος[23]. Επίσης σημαντική παρατήρηση ότι σε γενικές γραμμές η ισχύς του νόμου οδηγεί αποκλειστικά σε προσθήκες στο υπάρχον σύστημα και συνεπώς σε ανάπτυξη αυτού [17]. Πιθανές μετρικές προτείνονται από τους Guowu Xie, et.al. [22] είναι (α) Μέτρηση γραμμών κώδικα (LOC) (β) Μέτρηση των ορισμών του συστήματος (τύποι, λειτουργίες, μεταβλητές συστήματος) (γ) Μέτρηση των υπομονάδων (modules) ενός συστήματος.

Μετρικές για την επαλήθευση του νόμου

Πρέπει να παρατηρήσουμε ότι είναι αδύνατο να διακρίνει κανείς, από τις παραπάνω μετρικές, τις λειτουργικές προσθήκες στο σύστημα “που θα έπρεπε να υπήρχαν” και το ποσοστό της μεταβολής που σχετίζεται με το φαινόμενο της συνεχόμενης ανάπτυξης του συστήματος. Στην ιδανική περίπτωση, θα πρέπει να υπάρχουν λεπτομερή δεδομένα για τον αριθμό των υποψήφιων «λειτουργικών εκκρεμοτήτων” που αναμειγνύονται σε κάθε έκδοση κάτι που στην παρούσα μελέτη δεν έχουμε.

Στην μελέτη μας αυτή θα εξετάσουμε τις μετρήσεις που προκύπτουν από (i) τον αριθμό των λειτουργιών (operations) (ii) και τον αριθμό των τύπων του συστήματος (types) ανά έκδοση (version). Αυτό που θέλουμε να δούμε είναι η ανάπτυξη μεταξύ διαδοχικών εκδόσεων να δείχνουν μια γενική τάση επέκτασης την πάροδο του χρόνου



τότε και οι προσφερόμενες λειτουργίες των υπηρεσιών θα επεκτείνονται και αυτό θα αποτελεί ισχυρή ένδειξη ότι ο νόμος ισχύει.

7^{ος} Νόμος του Lehmann: Φθίνουσα ποιότητα (Declining quality)

Αν ένα E-type σύστημα δεν προσαρμόζεται και αναπτύσσεται στις αλλαγές του περιβάλλοντος, η ποιότητα του εν λόγω συστήματος θα εμφανίζεται να μειώνεται.

Ο νόμος αυτός προκύπτει άμεσα από τον πρώτο και τον έκτο νόμο, καθώς ένα E-τύπου σύστημα πρέπει να υποστεί αλλαγές και προσθήκες για να προσαρμοστεί και να αναπτυχθεί, εάν είναι να παραμένει ικανοποιητική η χρήση του σε ένα συνεχώς μεταβαλλόμενο περιβάλλον. Επομένως η λειτουργικότητα του πρέπει να επεκταθεί και να αλλάξει όπου και όταν χρειαστεί. Για να επιτευχθεί αυτό, νέα τμήματα κώδικα, υπομονάδες, και υποσυστήματα προστίθενται και νέες αλληλεπιδράσεις και διασυνδέσεις δημιουργούνται. Αν δεν γίνουν οι αλλαγές αυτές, οι αναντιστοιχίες με το επιχειρησιακό πεδίο του συστήματος αυξάνεται. Όπως αναφέρθηκε ήδη, η πολυπλοκότητα του συστήματος αυξάνεται όσον αφορά τις αλληλεπιδράσεις μεταξύ των μερών του [17]

Η αυξανόμενη πολυπλοκότητα και οι αναντιστοιχίες με τις επιχειρησιακές περιοχές που αλληλεπιδρά το σύστημα, η αύξηση του αριθμού των ελαττωμάτων, η αυξανόμενη δυσκολία στην προσαρμογή κατά την ανάπτυξη του συστήματος, θα συνδράμουν στο να μειωθεί η ικανοποίηση των ενδιαφερόμενων μερών, εκτός και αν αναληφθούν δράσεις, για την διατήρηση της ποιότητας του. Η ουσία του νόμου καθορίζεται από την έννοια της ποιότητας λογισμικού, και η διατήρηση της κατά την διάρκεια που αυτό αναπτύσσεται [17].

Υπάρχουν πολλές προσεγγίσεις για τον καθορισμό της ποιότητας του λογισμικού. Συμπερασματικά οι Lehman-Ramil [17] αναφέρουν ότι ουσιαστικά η ποιότητα λογισμικού αποτελεί μια συνάρτηση πολλών παραγόντων των οποίων η σχετική σημασία θα ποικίλλει ανάλογα με τις περιστάσεις. Οι χρήστες του συστήματος αντιμετωπίζουν τη έννοια της ποιότητας με ορούς όπως η επίδοση, η αξιοπιστία, η λειτουργικότητα και η προσαρμοστικότητα σε αυτό. Ο διευθύνων σύμβουλος της εταιρείας που διαχειρίζεται το σύστημα, αντιμετωπίζει την ποιότητα του, με την



συνεισφορά του συστήματος στην εταιρική κερδοφορία, το μερίδιο αγοράς που καταλαμβάνει, οι πόροι που απαιτούνται για την υποστήριξή του, και ούτω καθεξής. Έτσι καταλαβαίνουμε ότι η ποιότητα του συστήματος ορίζεται διαφορετικά με βάση την οπτική που έχουν όλοι οι εμπλεκόμενοι στο σύστημα και αφήνουμε ανοιχτό το θέμα μέτρησης της καθώς αυτή μπορεί να οριστεί διαφορετικά από αυτούς. Σημαντική και η παρατήρηση ότι η μείωση στην ποιότητα του συστήματος με το πέρασμα του χρόνου φαίνεται ότι συνδέεται με την αύξηση της πολυπλοκότητας λόγω γήρανσης του [17]

Οι Guowu Xie, et. al. [22] υιοθετούν την πρώτη οπτική του Lehman [23] και αναφέρονται σε εξωτερική ποιότητα του συστήματος που καθορίζεται ανά περίπτωση όπως αιτιολογήσαμε στην προηγούμενη παράγραφο και στην εσωτερική που αναφέρεται στις μετρικές που χαρακτηρίζουν την πολυπλοκότητα του συστήματος. Πιθανές μετρικές προτείνονται για την εσωτερική ποιότητα ενός συστήματος από τους Guowu Xie, et. al. [22] είναι (α) Ο αριθμός των γνωστών ελαττωμάτων του συστήματος για κάθε έκδοση (β) πυκνότητα των ελαττωμάτων για κάθε έκδοση (γ) ποσοστό των υπομονάδων (modules) που έχουν αλλάξει

Μετρικές για την επαλήθευση του νόμου. Όπως προαναφερθήκαμε είναι δύσκολο να μετρήσουμε την εξωτερική ποιότητα του λογισμικού. Αυτό θα απαιτούσε λεπτομερή στοιχεία από διορθώσεις, προσθήκες λειτουργιών, καθώς και σχέδια για επεκτάσεις του συστήματος τα οποία είναι σχεδόν αδύνατα να βρεθούν αφού κατά την γνώμη μας αποτελούν μη δημοσιεύσιμα έγγραφα (και ειδικότερα όσο αφορά τις διαδικτυακές υπηρεσίες της Amazon).

Όσο αφορά την εσωτερική ποιότητα του συστήματος την προσεγγίζουμε με μετρικές που συνδέονται με την πολυπλοκότητα του συστήματος και αναφερθήκαμε σε αυτές στην ενότητα που καλύπτεται ο 2ος νόμος. Στην μελέτη αυτή υιοθετούμε την άποψη των Lehman - Ramil [17] ότι η ισχύς του νόμου επαληθεύεται έμμεσα αν και μονό αν υπάρχουν ισχυρές ενδείξεις ότι υπάρχει μηχανισμός τροφοδότησης του συστήματος (νόμοι 3,8) και ισχύει και ο 2ος νόμος.

Όσο αφορά την εξωτερική ποιότητα θα την προσεγγίσουμε μελετώντας τις περιγραφές που συνοδεύουν τις καινούργιες εκδόσεις που έχουν δημοσιοποιηθεί από την εταιρεία Amazon, σεβόμενοι πάντα την παρατήρηση ότι η ουσία των αλλαγών



όσο αφορά την ποιότητα του λογισμικού μονό από στοιχειοθετημένα έγγραφα μπορεί να επαληθευτεί.

8ος Νόμος του Lehman: Ανατροφοδοτούμενο σύστημα (Law of Feedback System)

«E-type evolution processes are multi-level, multi-loop, multi-agent feedback systems»

Η διαδικασία ανάπτυξης E-type συστημάτων είναι ένα ανατροφοδοτούμενο σύστημα σε πολλά επίπεδα, με πολλές αναδράσεις και πολλούς εμπλεκόμενους.

Η συμπεριφορά των πολύπλοκων συστημάτων ανατροφοδότησης δεν είναι και δεν μπορεί, σε γενικές γραμμές, να περιγραφεί άμεσα με ορούς που περιγράφουν μόνο την προς τα εμπρός εξελικτική διαδρομή και μηχανισμούς του. Οι θετικοί και αρνητικοί μηχανισμοί ανατροφοδότησης και οι μηχανισμοί ελέγχου ενός E-type συστήματος περιλαμβάνουν δραστηριότητες από πολλά διαφορετικά πεδία, όπως το μάρκετινγκ, των επιχειρήσεων, η χρήση και η οργάνωση του συστήματος, κ.α. Η διαδικασία αυτή αναπτύσσει μια δυναμική που οδηγεί σε βελτίωση άλλα και ταυτόχρονα το περιορίζει από ανεξέλεγκτη ανάπτυξη [17].

Μετρικές για την επαλήθευση του νόμου. Υιοθετούμε την προσέγγιση και τους τύπους που δίνονται στην εργασία του Lehman [11] στην οποία αναπτύχτηκε το μοντέλο του αντίστροφου τετράγωνου («inverse square»), η οποία αποτελεί μια μη γραμμική-διακριτού χρόνου- δυναμική αναδρομή. Ο τύπος είναι: $\hat{S}_i = \hat{S}_{i-1} + \frac{\bar{E}}{\hat{S}_{i-1}^2}$

Όπου το $\hat{S}$ είναι η εκτίμηση για το μέγεθος του συστήματος στην έκδοση i και το $\bar{E}$ είναι μια παράμετρος που σχετίζεται με την μέτρηση της προσπάθειας (effort) που έχει καταβληθεί για την μετάβαση από την μια έκδοση στην άλλη και αποτελεί την μέση τιμή ενός συνόλου από προηγούμενες εκτιμήσεις τις ίδιες παραμέτρους.

Στην ενότητα με την αξιολόγηση του νόμου θα αναλύσουμε τις διαφορετικές προσεγγίσεις που θα ακολουθήσουμε για την μεταβλητή E .



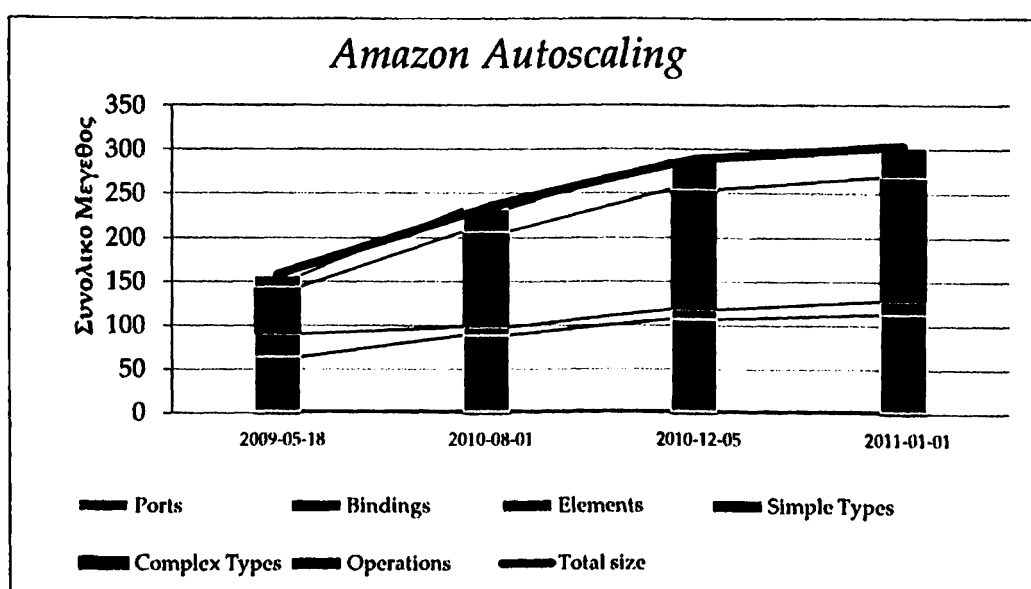
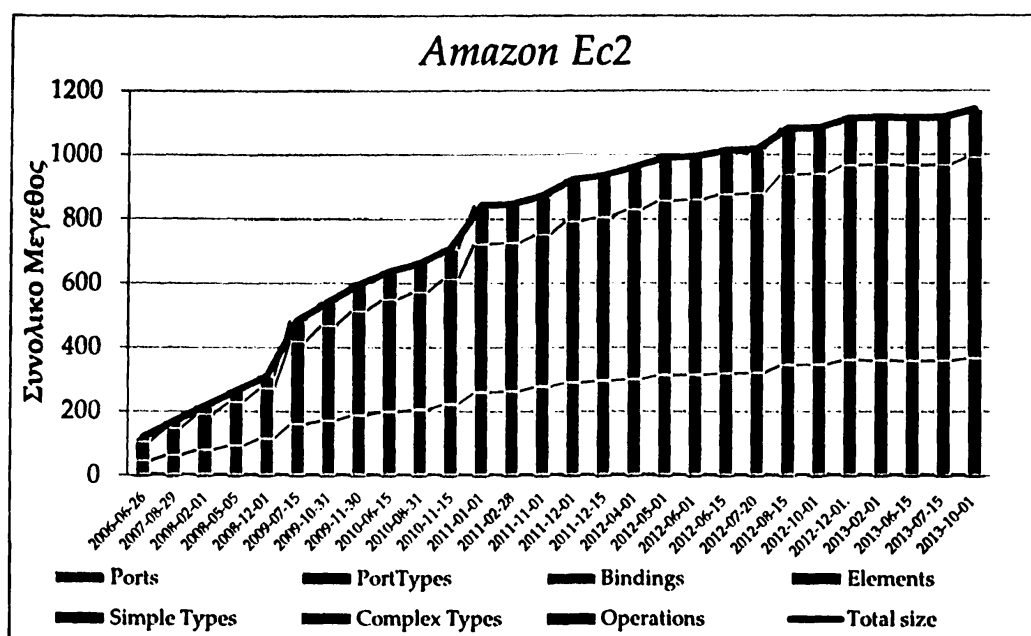
Για να εκτιμήσουμε τον νόμο πρέπει να επιβεβαιώσουμε ότι είναι δυνατό να προσομοιώσουμε την εξέλιξη των εκδόσεων ενός Web Service μέσω του τύπου με την πραγματική εξέλιξη του Web Services.

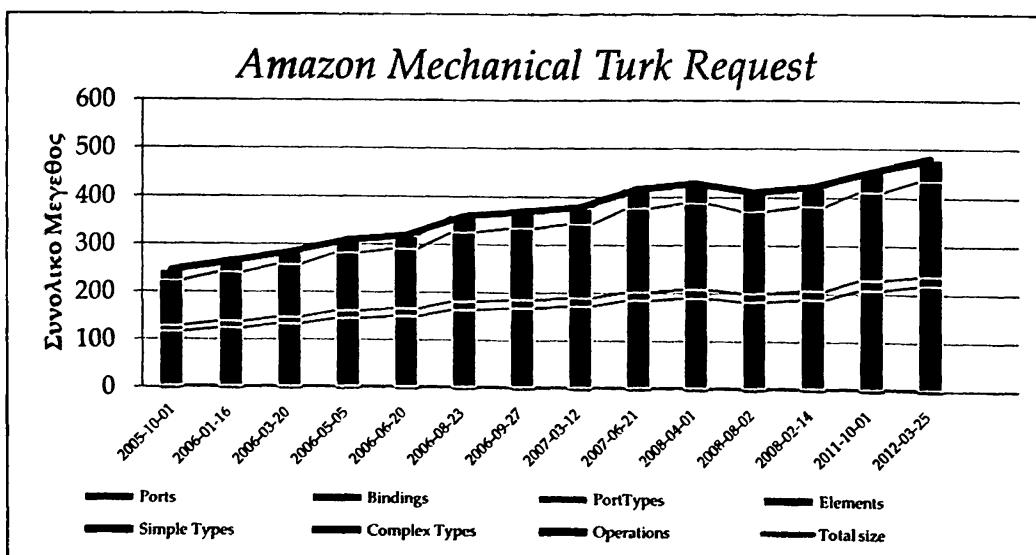
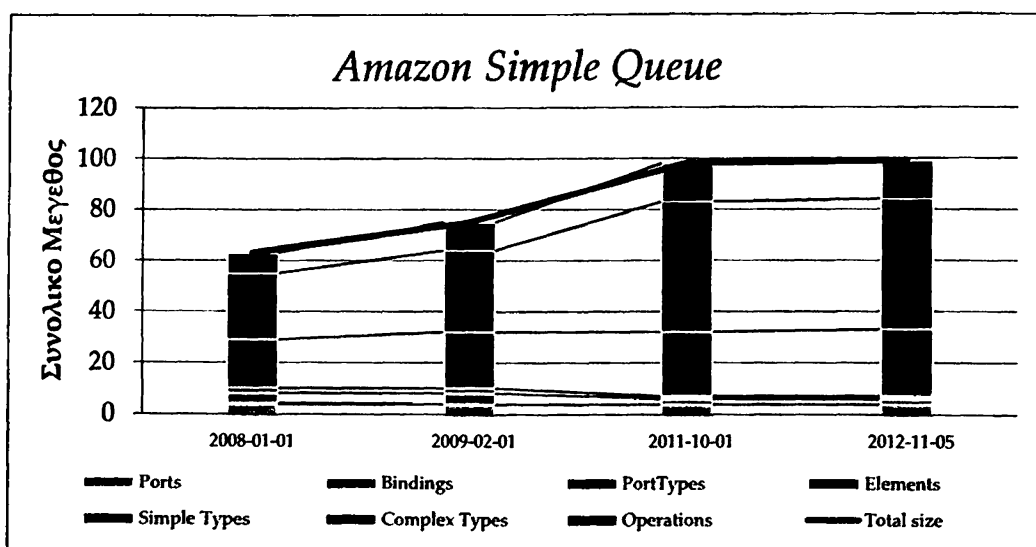
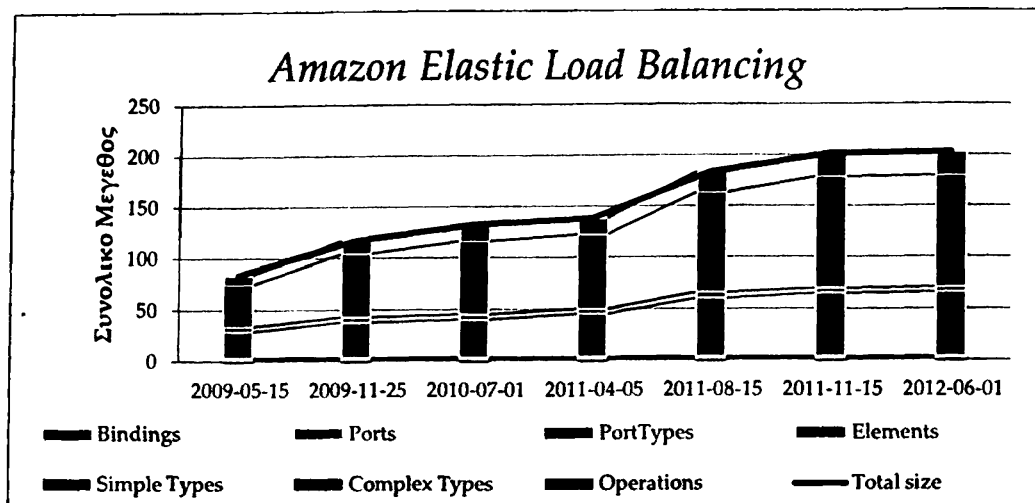


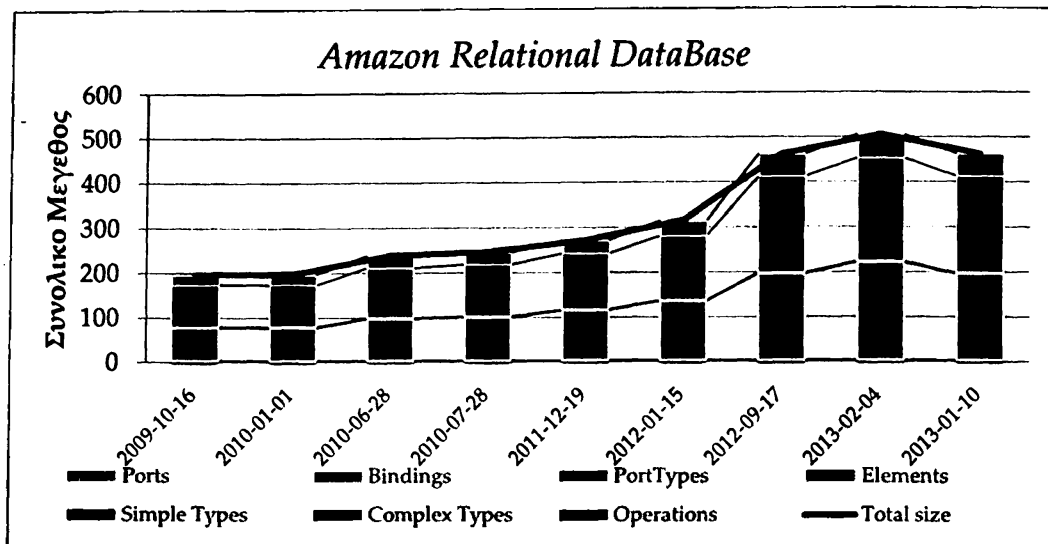
5.3. Αποτελέσματα μετρήσεων

Σε αυτή την ενότητα, παρουσιάζουμε τα αποτελέσματα για διάφορες μετρήσεις που πραγματοποιήσαμε, για τις έξι διαδικτυακές υπηρεσίες της Amazon.com.

5.3.1. Γραφήματα αλλαγής του συνολικού μεγέθους των σχημάτων ανά εξελικτικό βήμα.

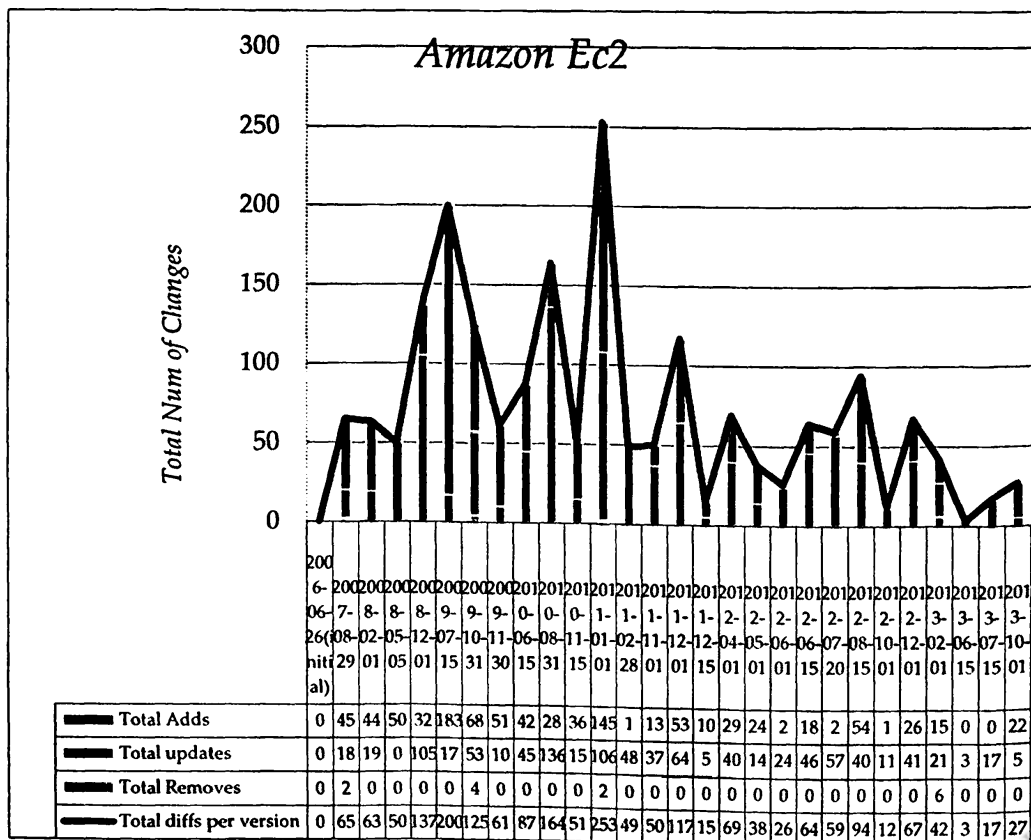


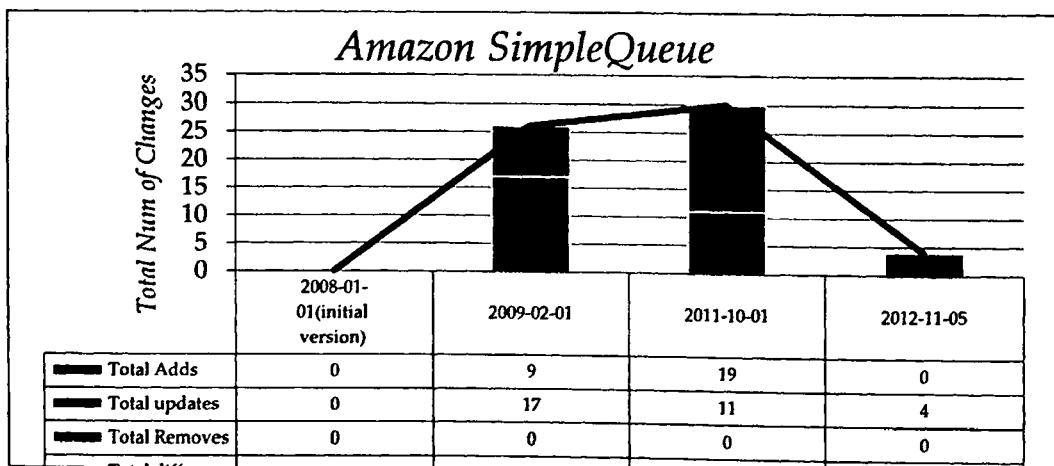
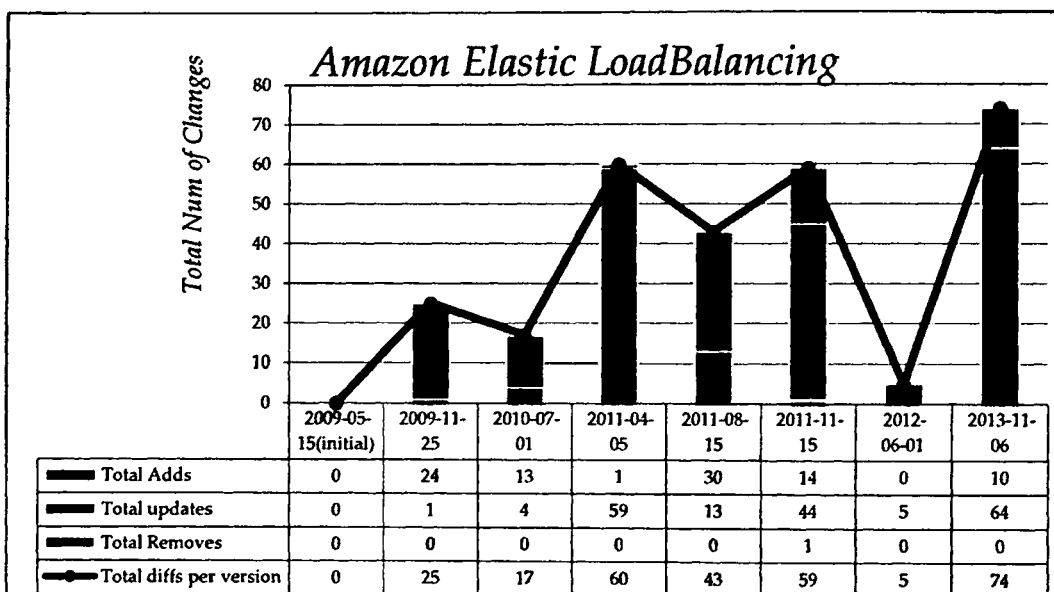
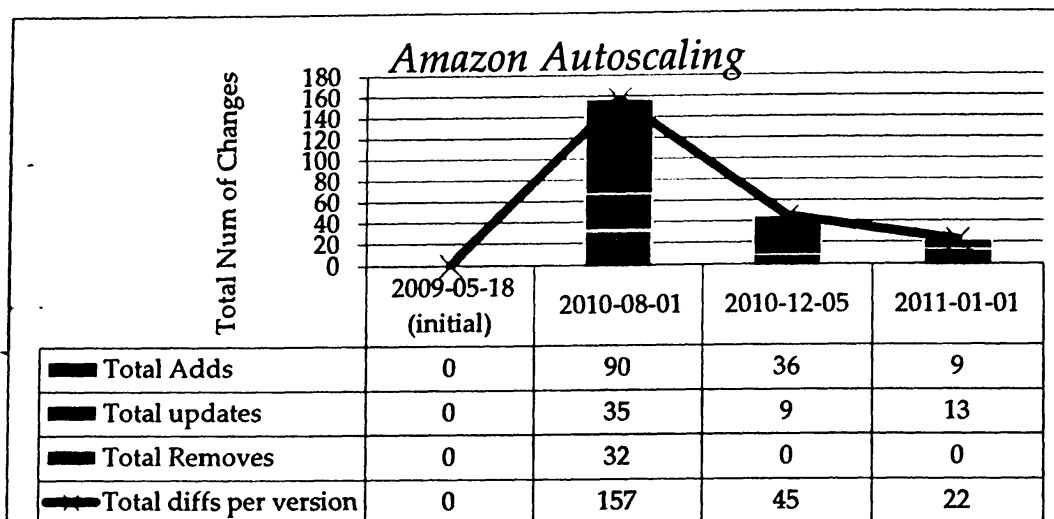


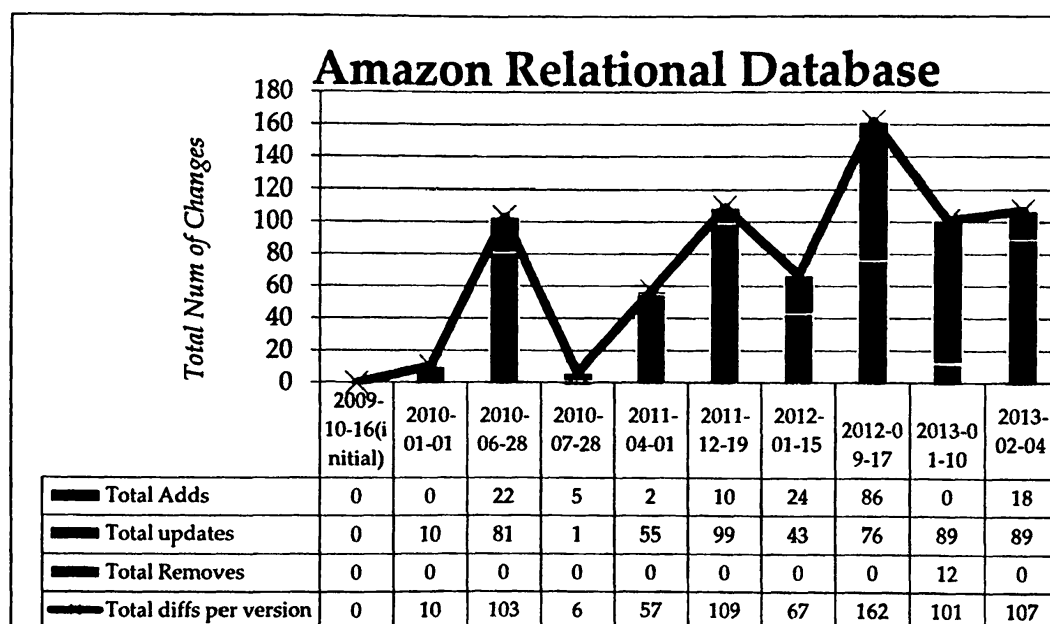
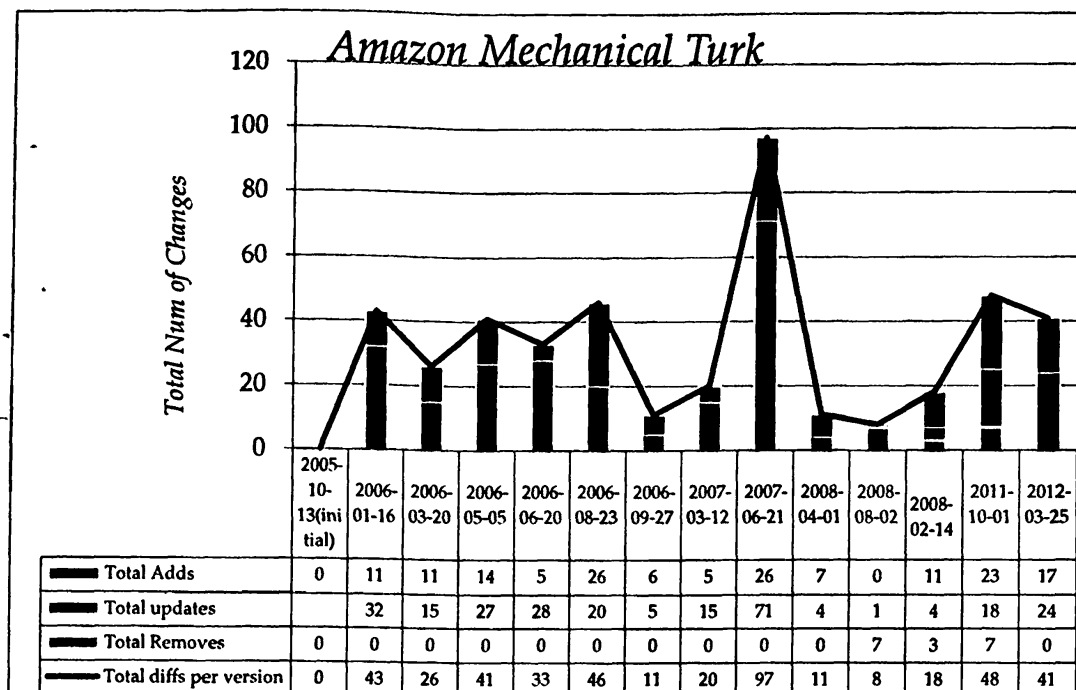


Σχήμα 5.1: Συγκριτική παρουσίαση της εξέλιξης του συνολικού μεγέθους των υπηρεσιών ανά έκδοση

5.3.2. Μετρήσεις των αλλαγών μεταξύ διαδοχικών εκδόσεων

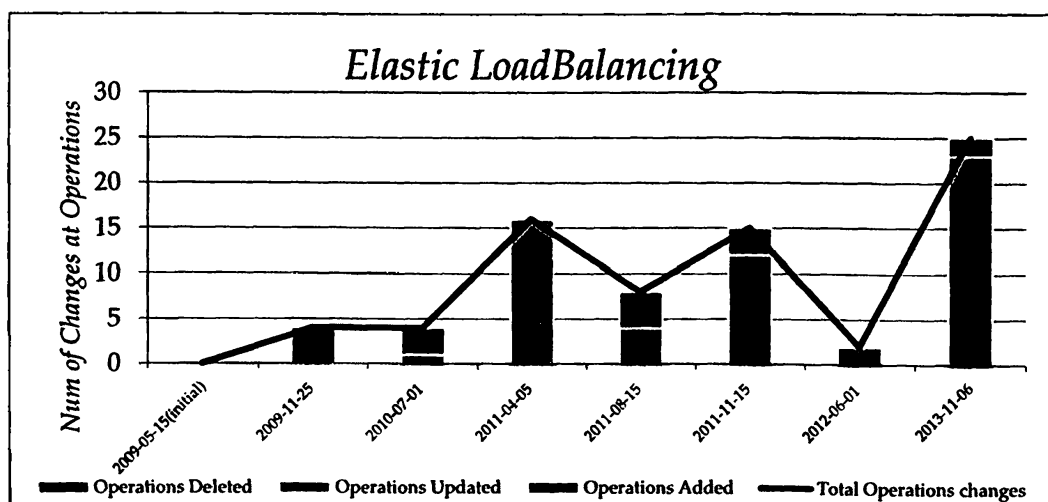
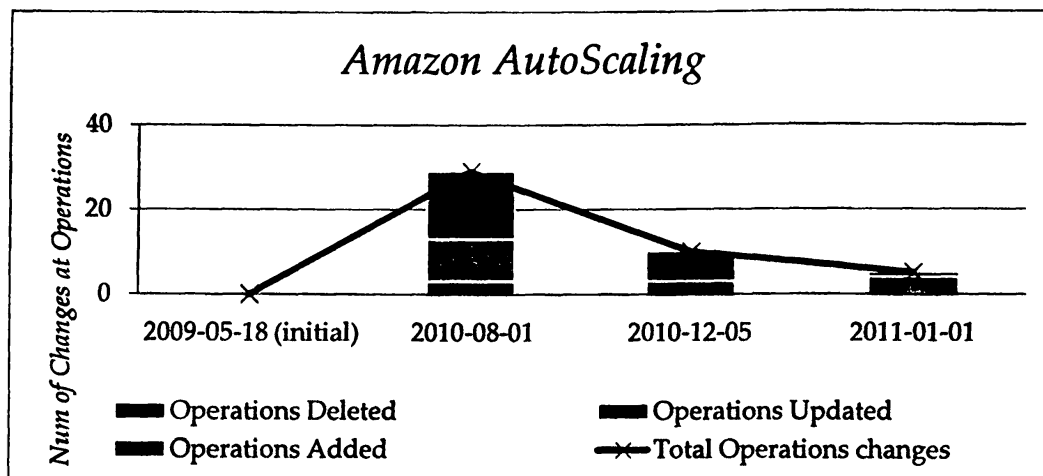
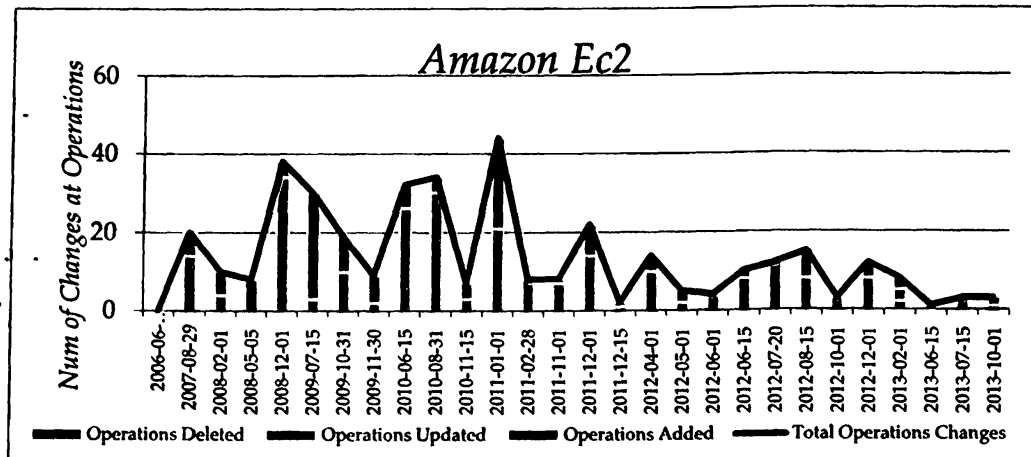


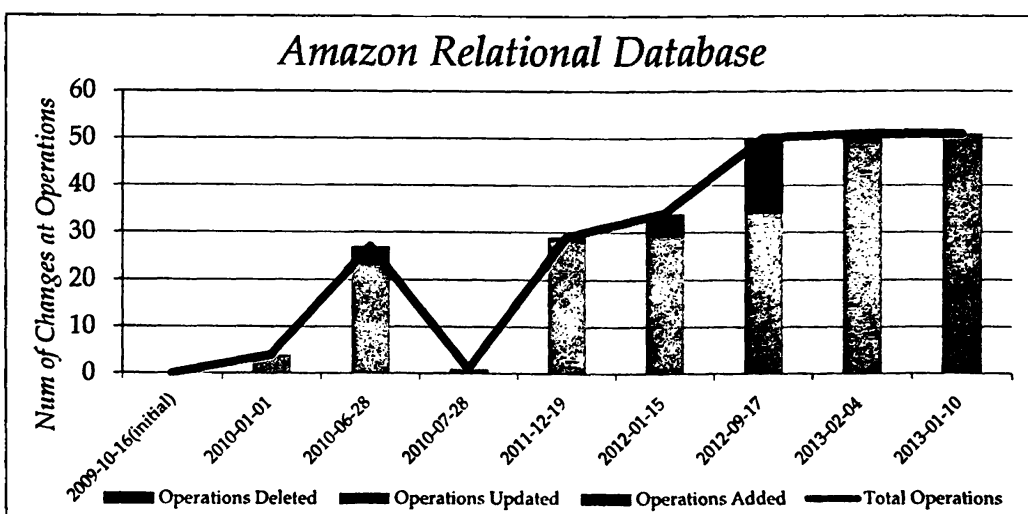
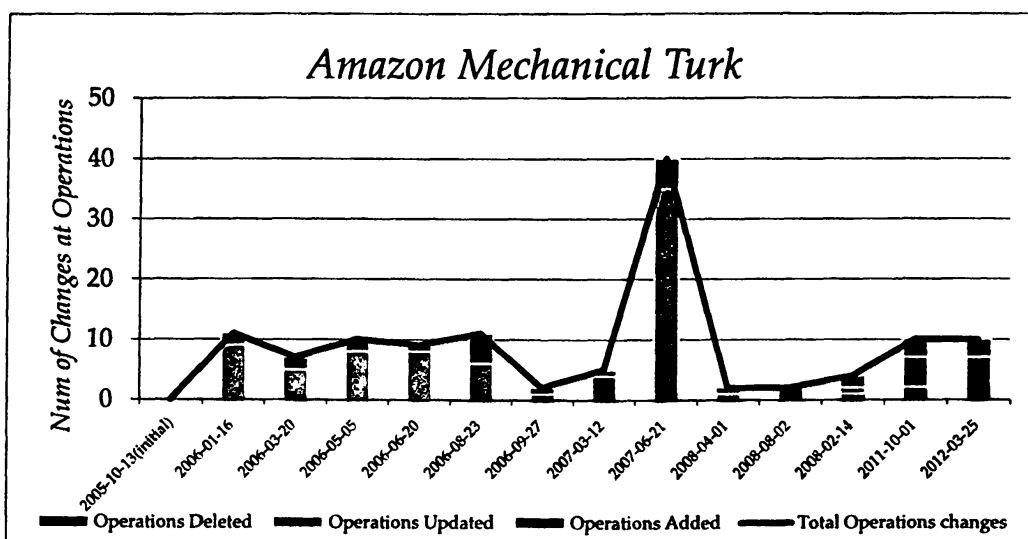
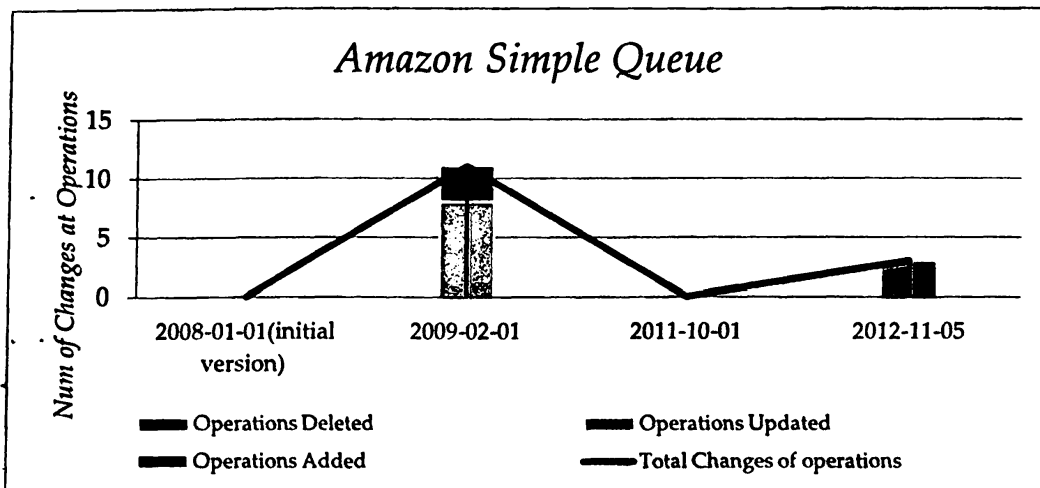




Σχήμα 5.2: Συγκριτική παρουσίαση της εξέλιξης των αλλαγών των υπηρεσιών ανά έκδοση

5.3.3. Γραφήματα αριθμού αλλαγών ανά έκδοση σε σχέση με τα operation

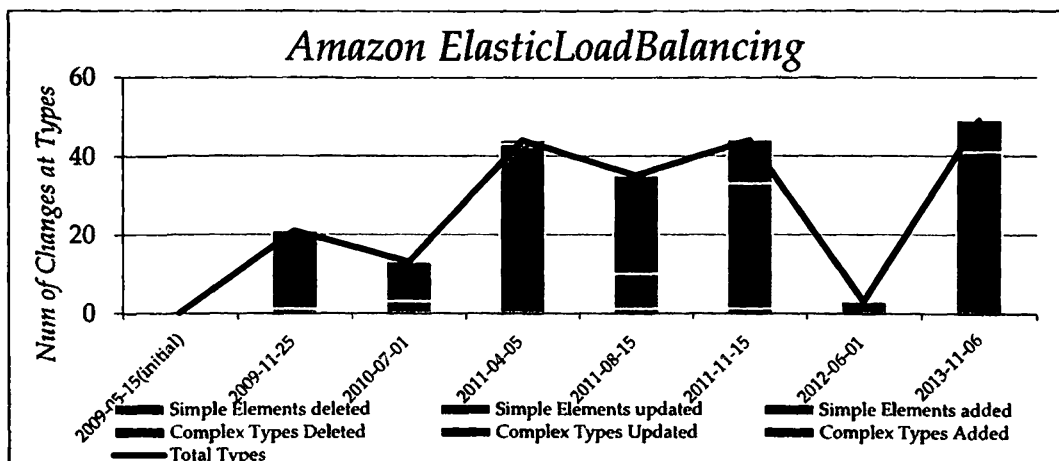
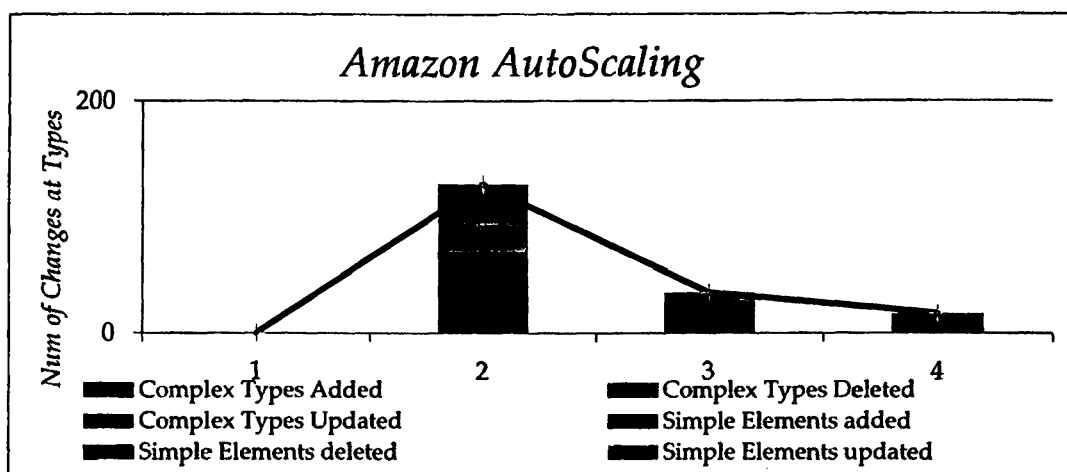
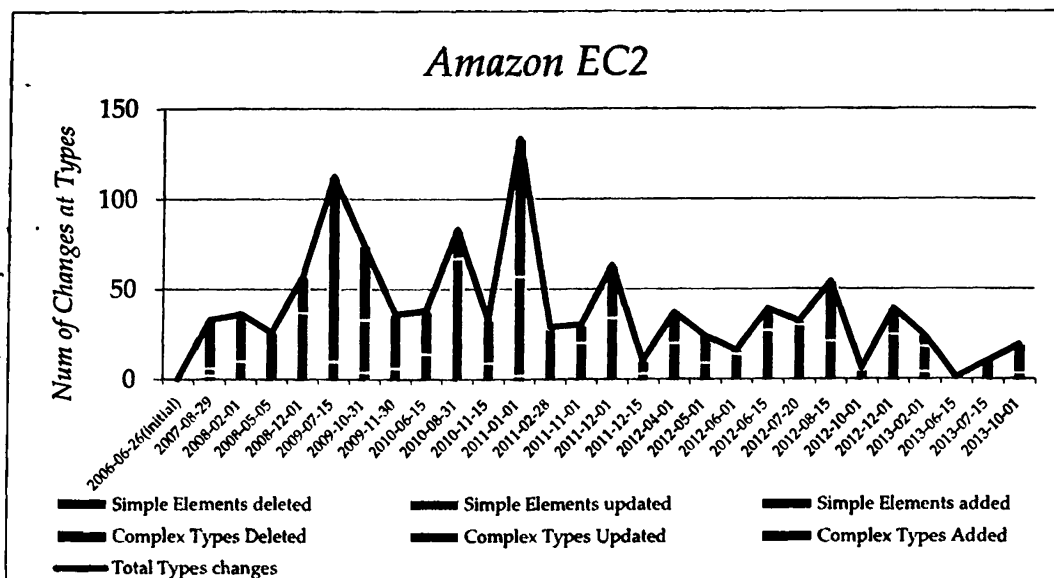


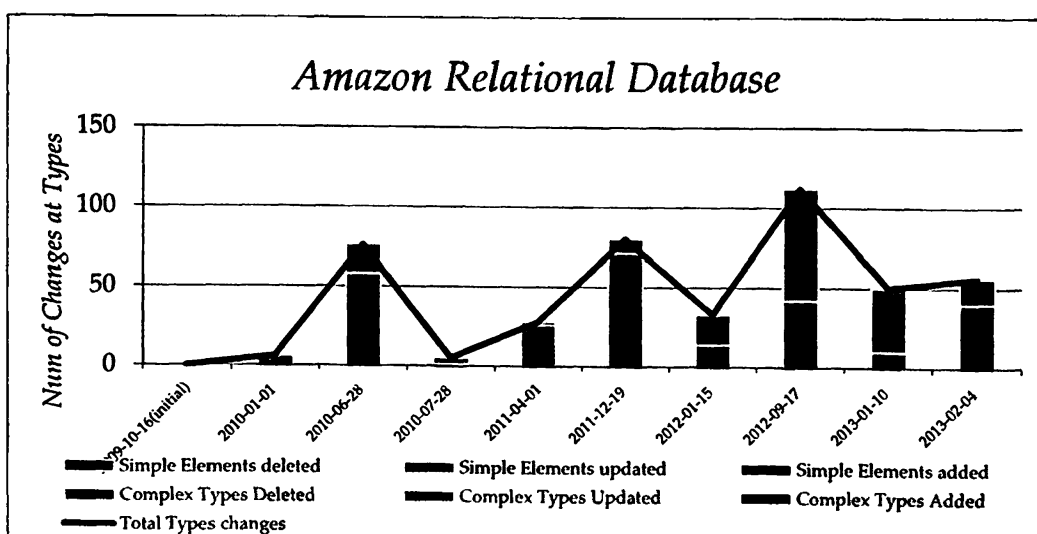
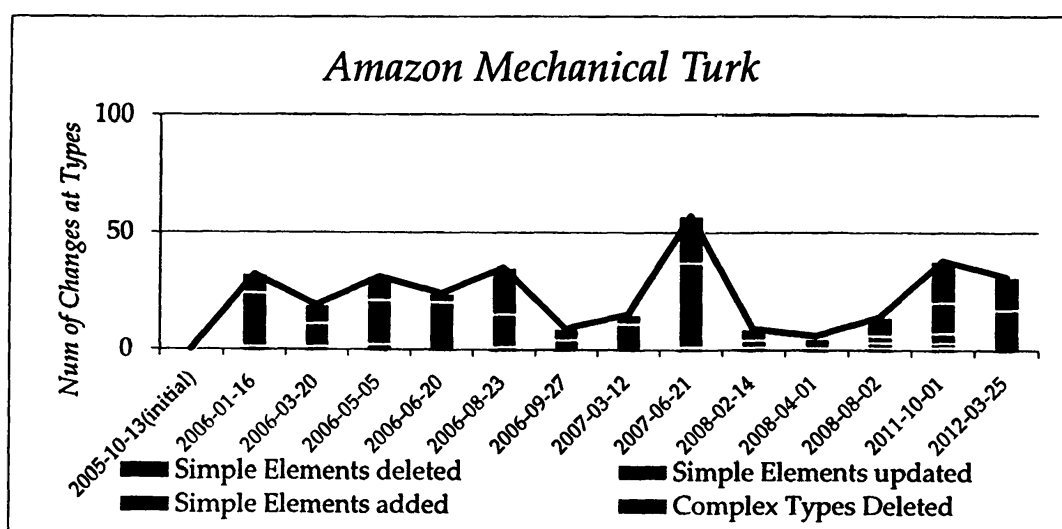
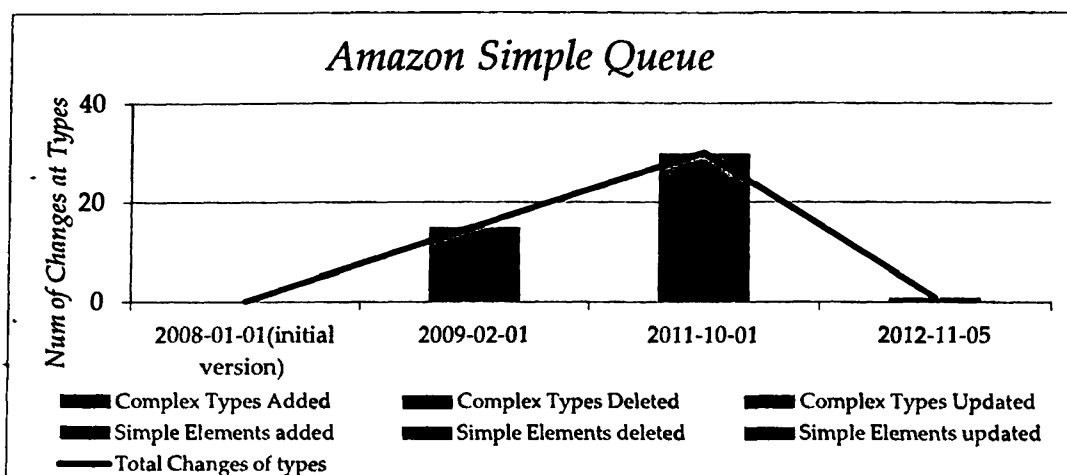


Σχήμα 5.3: Συγκριτική παρουσίαση της εξέλιξης των αλλαγών των υπηρεσιών σε σχέση με τα operations



5.3.4. Γραφήματα αριθμού αλλαγών ανά έκδοση σε σχέση με τα types (complex and simple types)

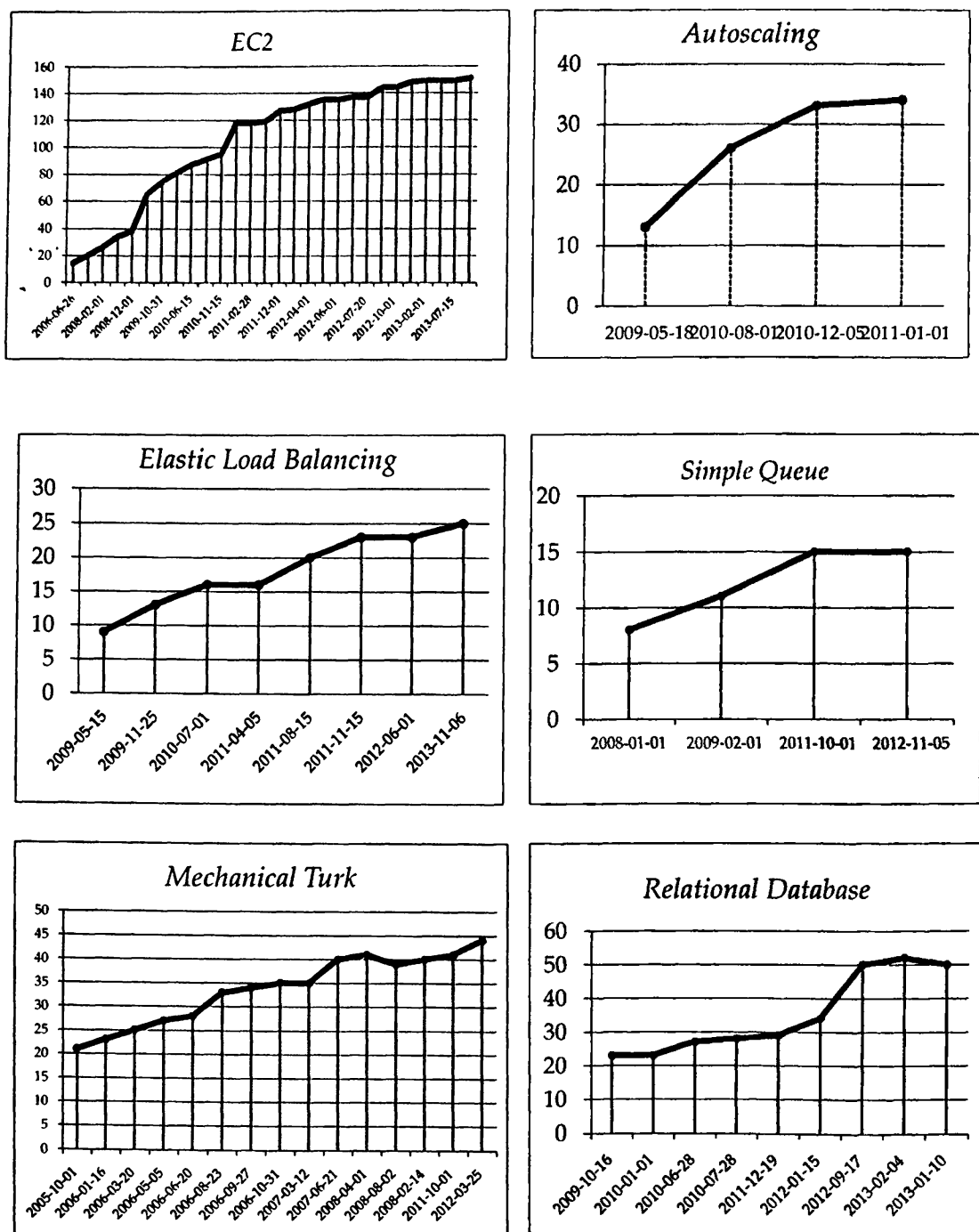




Σχήμα 5.4: Συγκριτική παρουσίαση της εξέλιξης των αλλαγών των υπηρεσιών σε σχέση με τα types

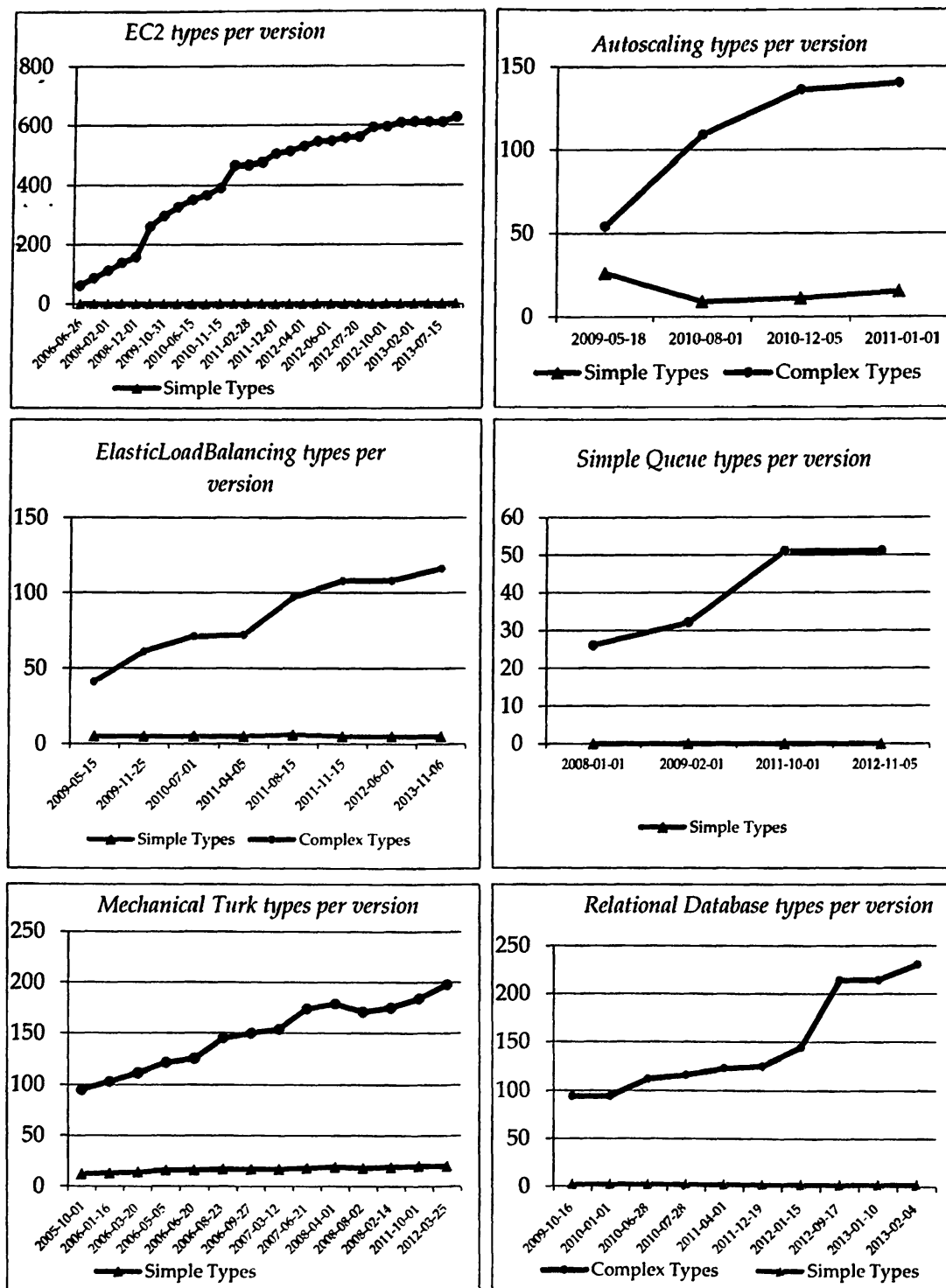


5.3.5. Μετρήσεις των αλλαγών του μεγέθους των εκδόσεων βάση των operations



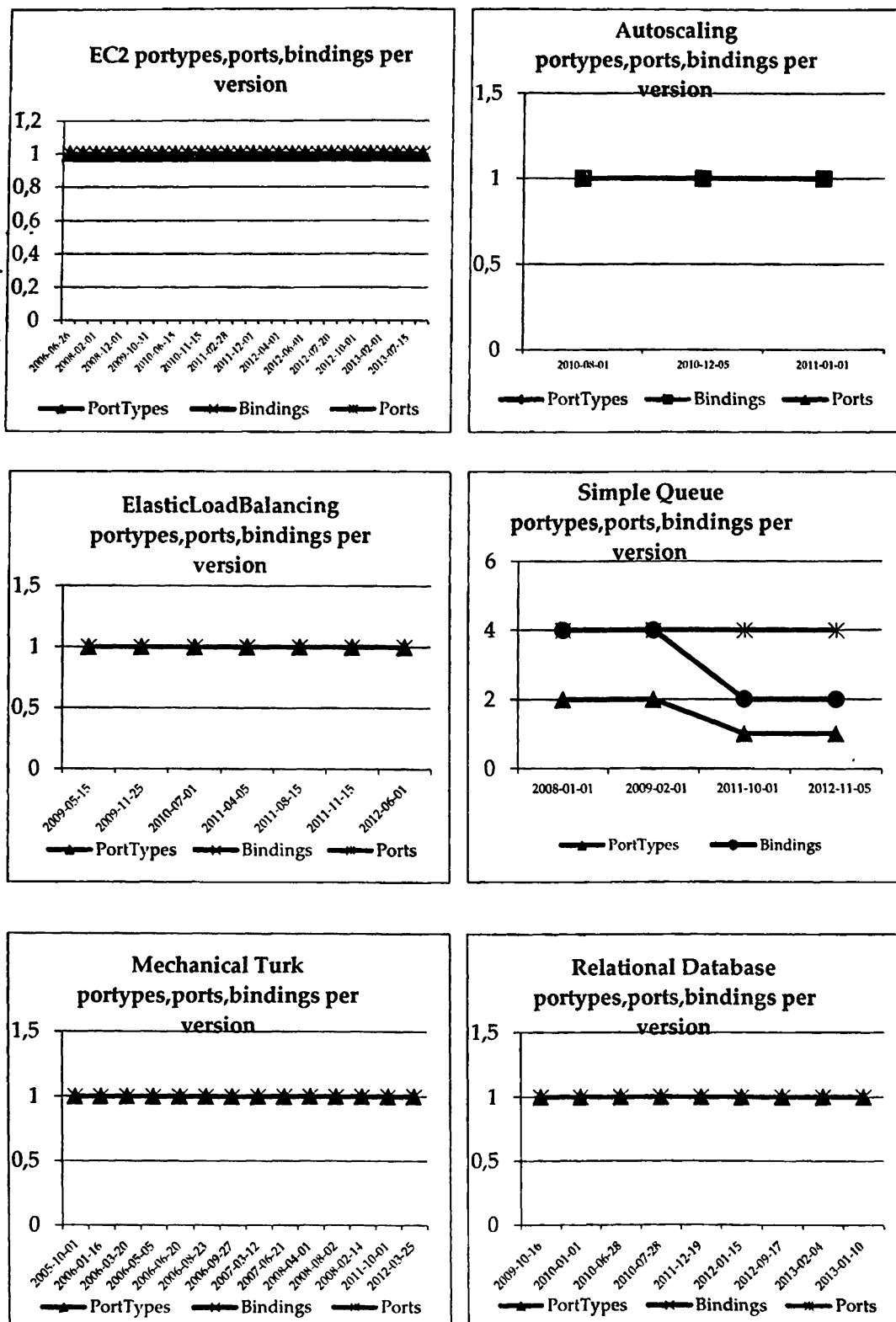
Σχήμα 5.5: Συγκριτική παρουσίαση των αλλαγών του μεγέθους των υπηρεσιών σε σχέση με τα operations

5.3.6. Μετρήσεις των αλλαγών του μεγέθους των εκδόσεων βάσει των types (Complex, Simple)



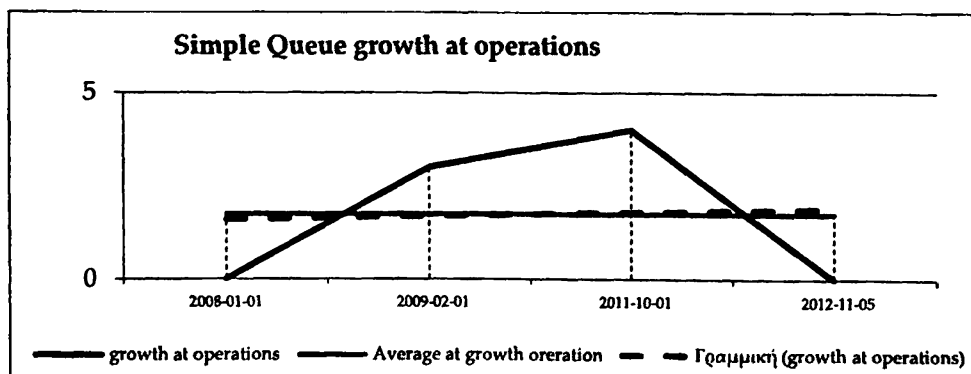
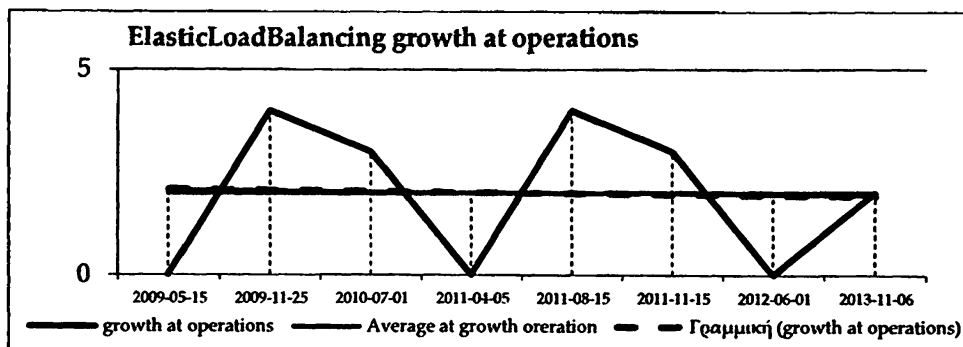
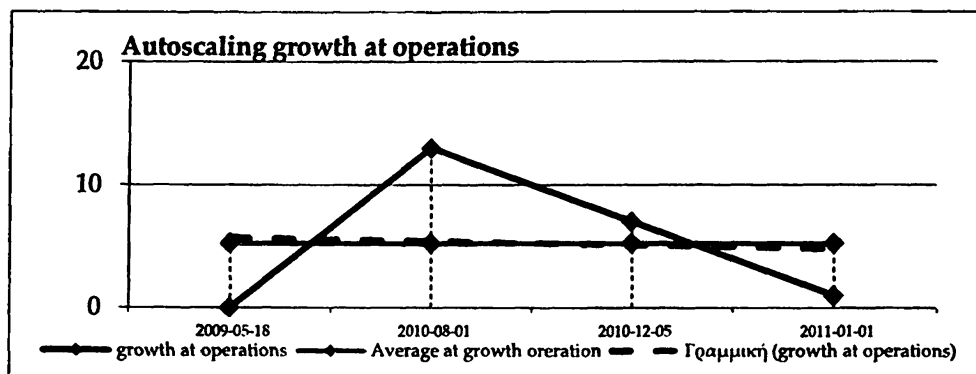
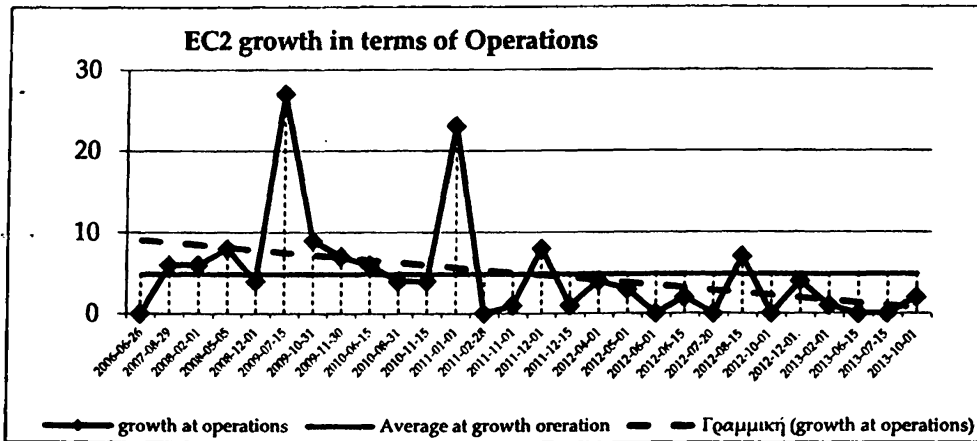
Σχήμα 5.6: Συγκριτική παρουσίαση των αλλαγών του μεγέθους των υπηρεσιών σε σχέση με τα types

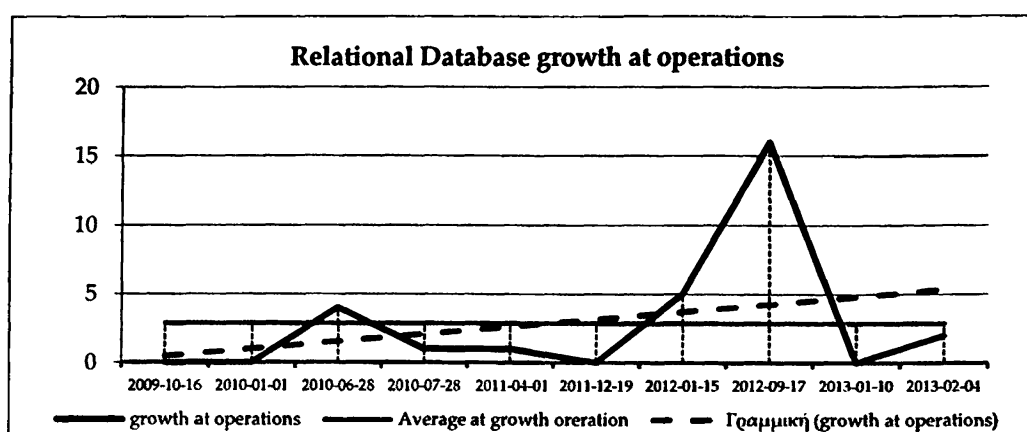
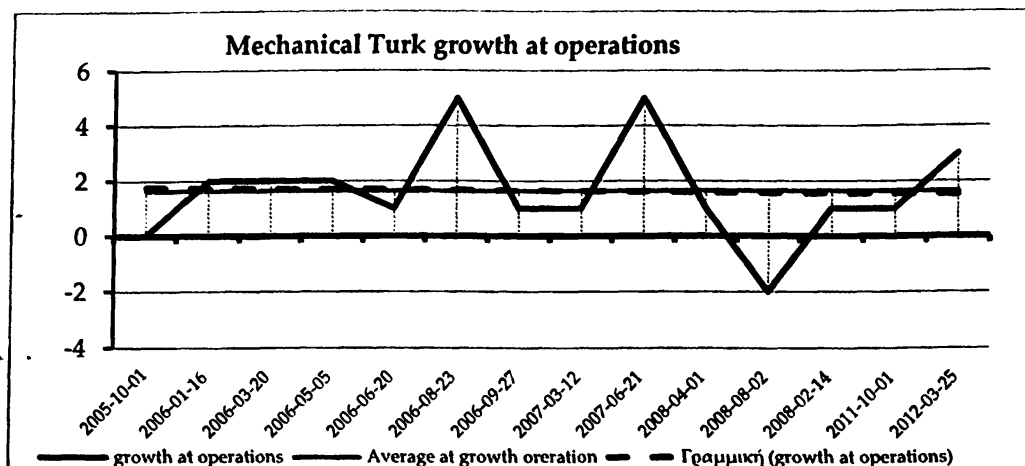
5.3.7. Μετρήσεις, portTypes, bindings, ports



Σχήμα 5.7: Συγκριτική παρουσίαση των αλλαγών του μεγέθους των υπηρεσιών σε σχέση με τα portTypes,ports,services

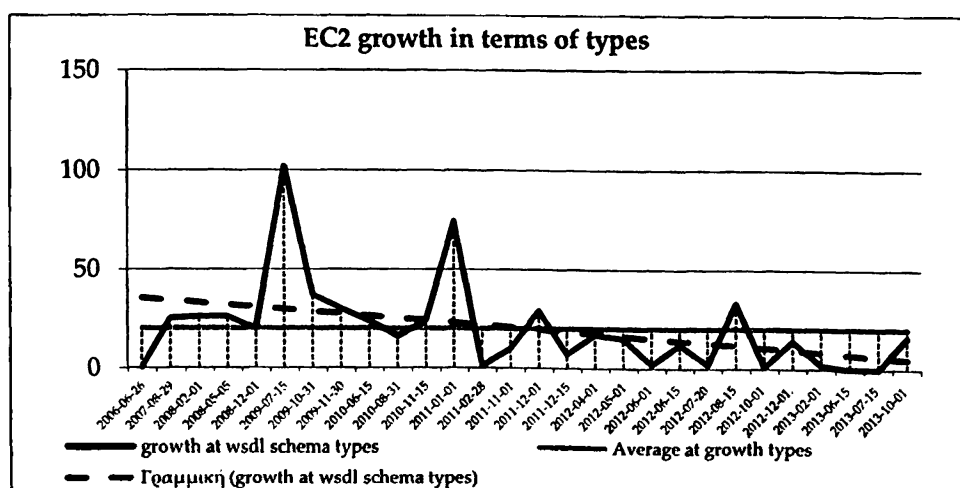
5.3.8. Μετρήσεις ανάπτυξης (growth) των διαδοχικών εκδόσεων σε σχέση με τα operations

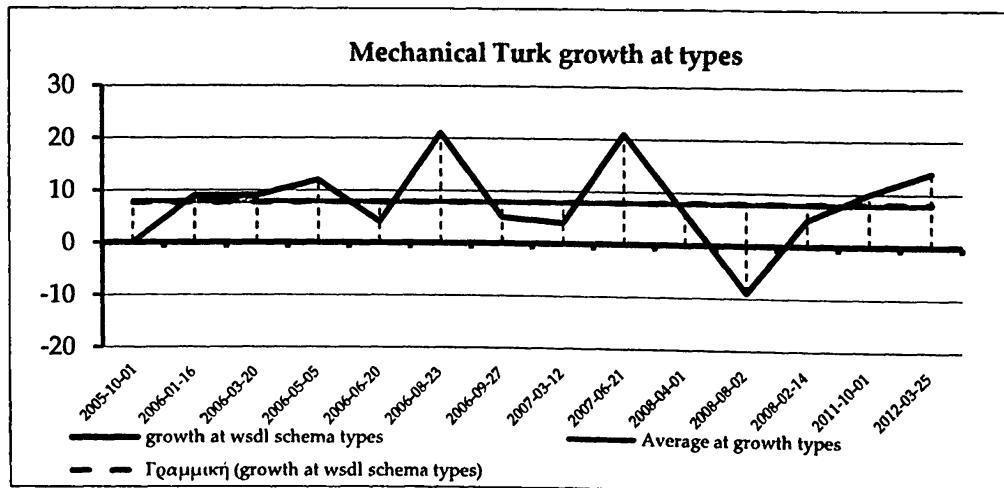
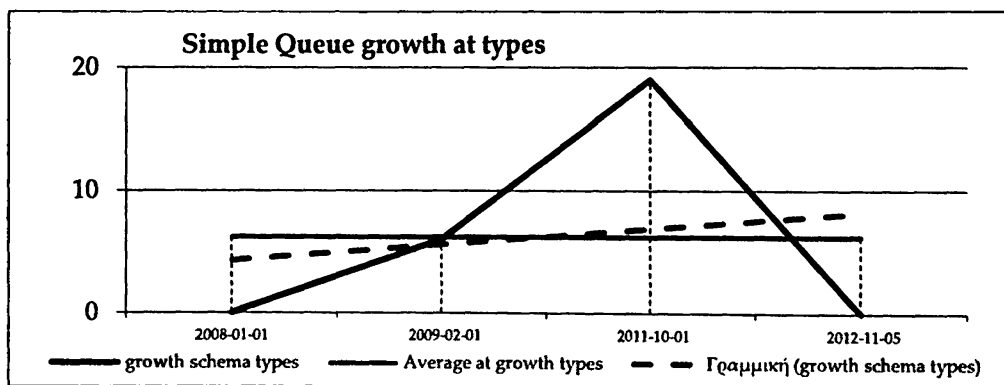
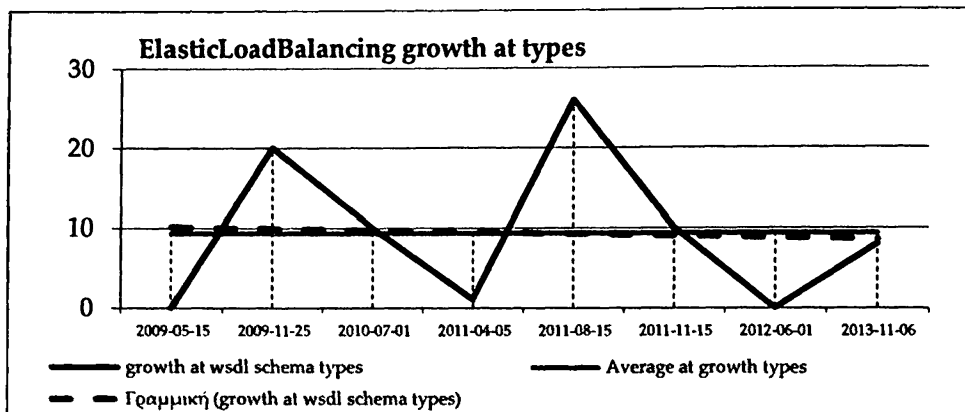
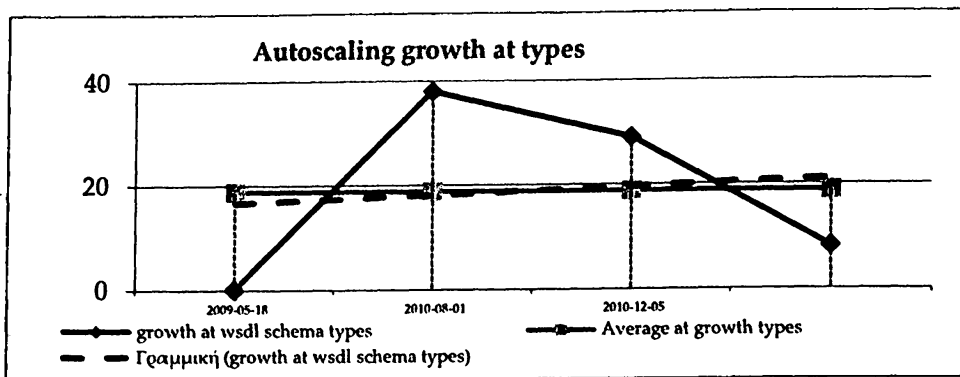


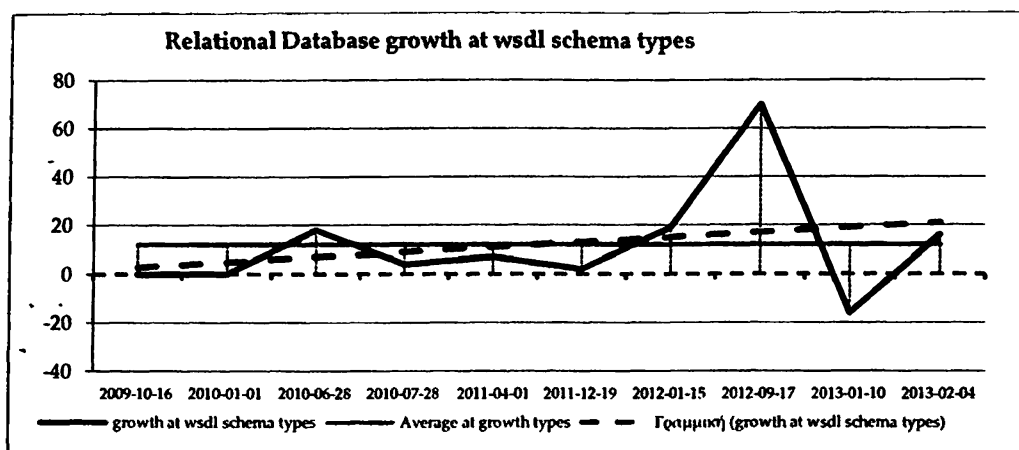


Σχήμα 5.8: Μετρήσεις ανάπτυξης (growth) των διαδοχικών εκδόσεων σε σχέση με τα operation

5.3.9. Μετρήσεις ανάπτυξης (growth) των διαδοχικών εκδόσεων σε σχέση με types

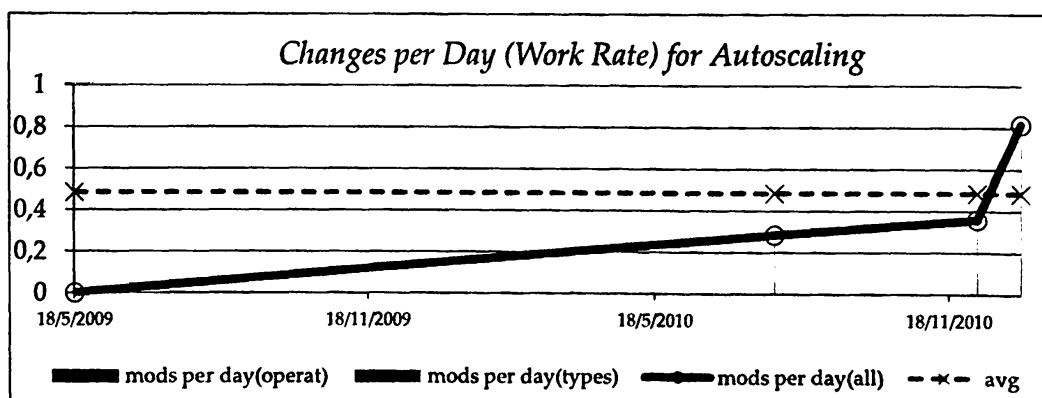
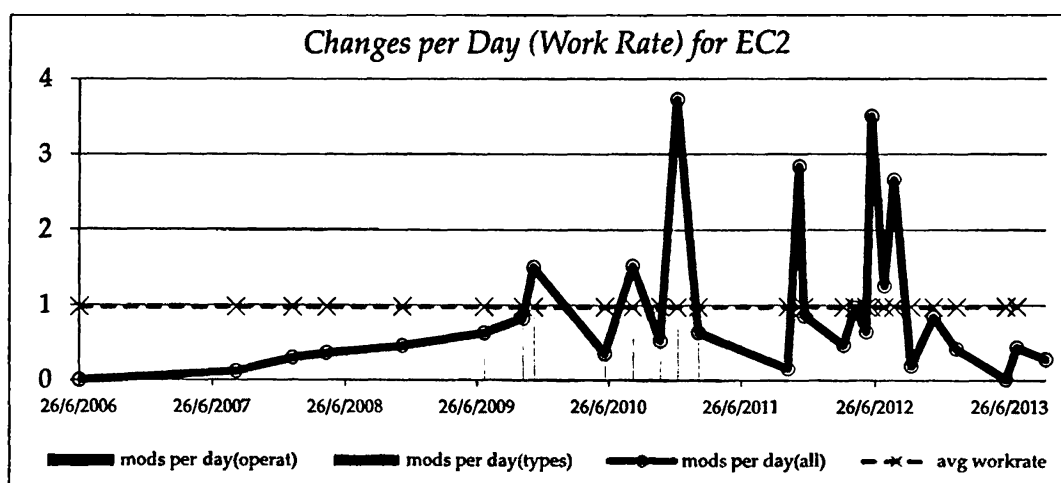


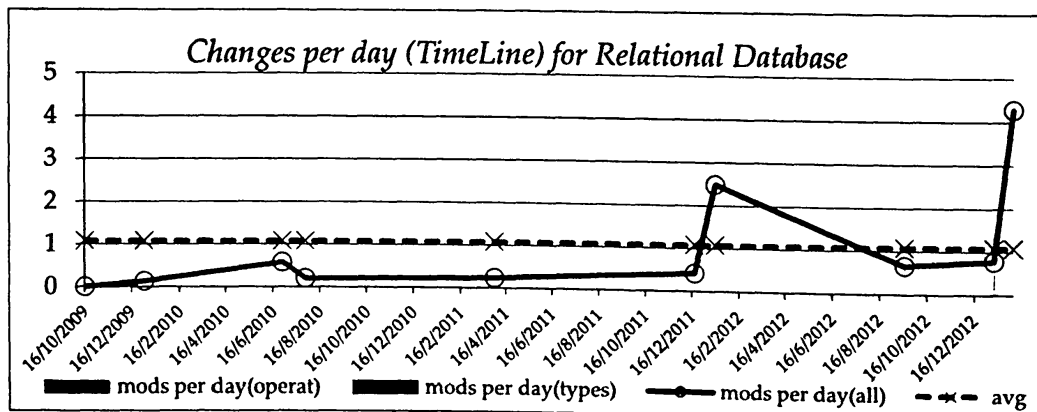
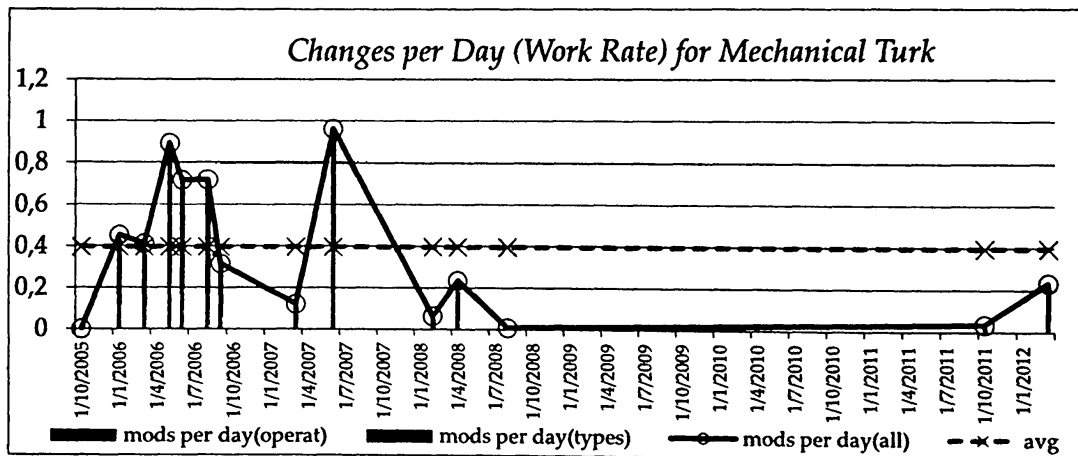
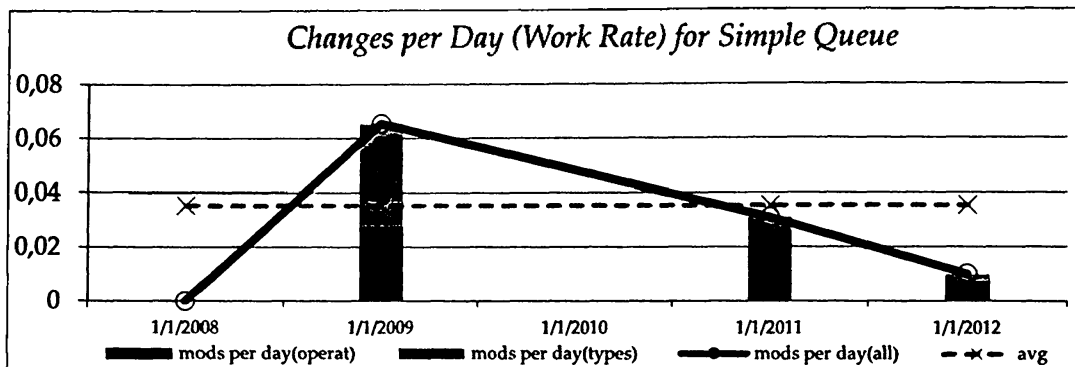
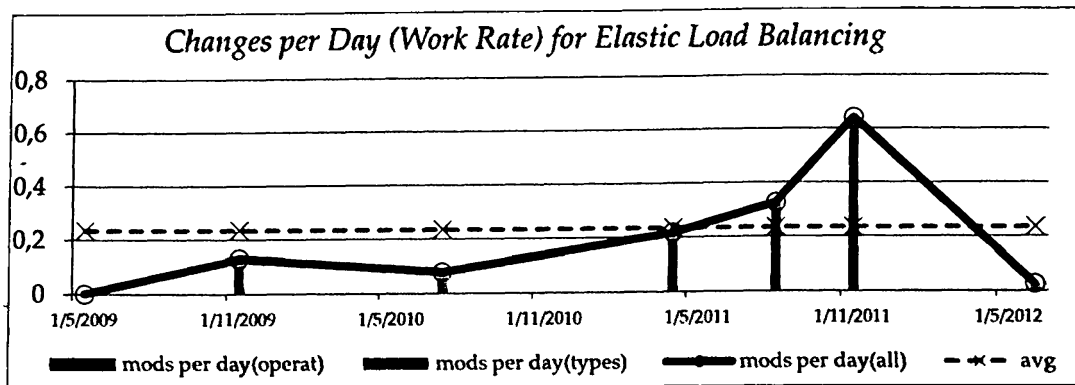




Σχήμα 5.9: Μετρήσεις ανάπτυξης (growth) των διαδοχικών εκδόσεων σε σχέση με τα types

5.3.10. Μετρήσεις ρυθμού εργασίας (work rate)

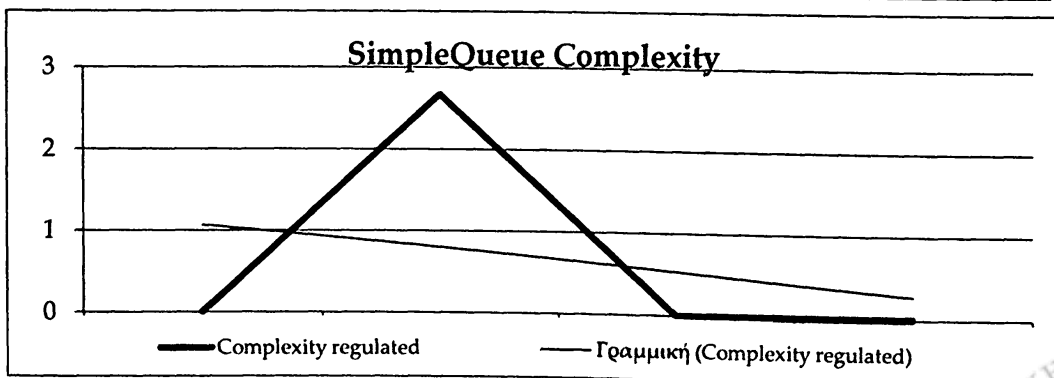
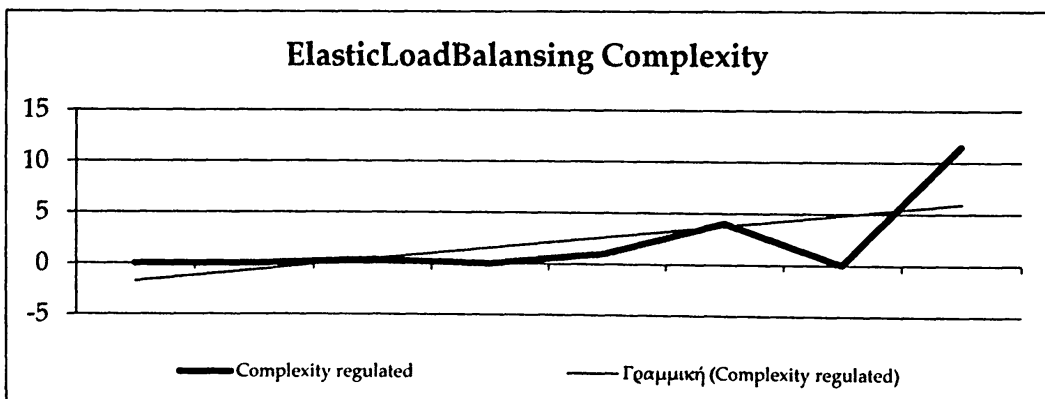
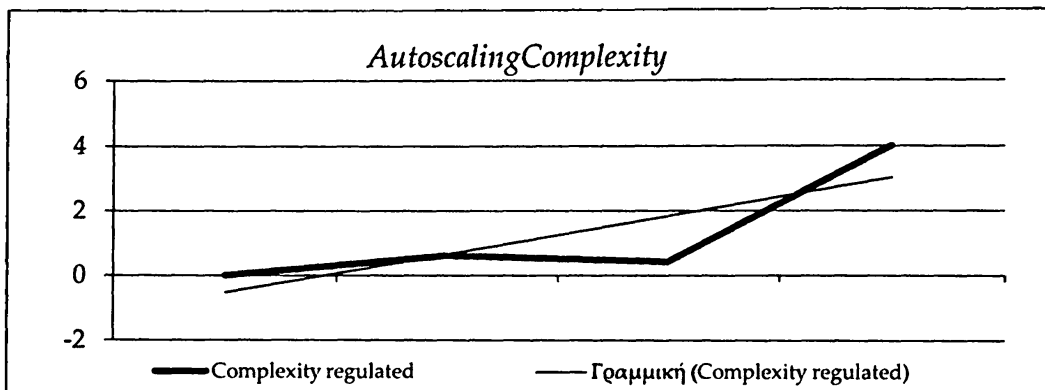
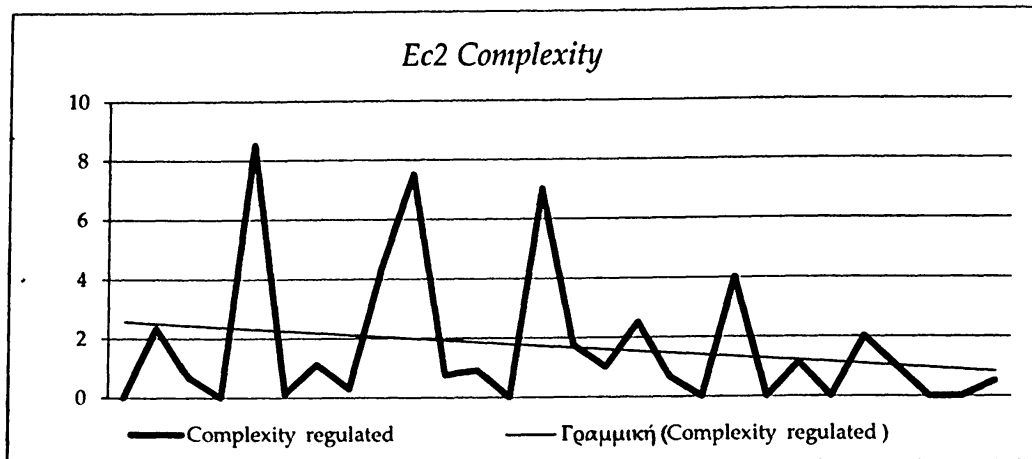


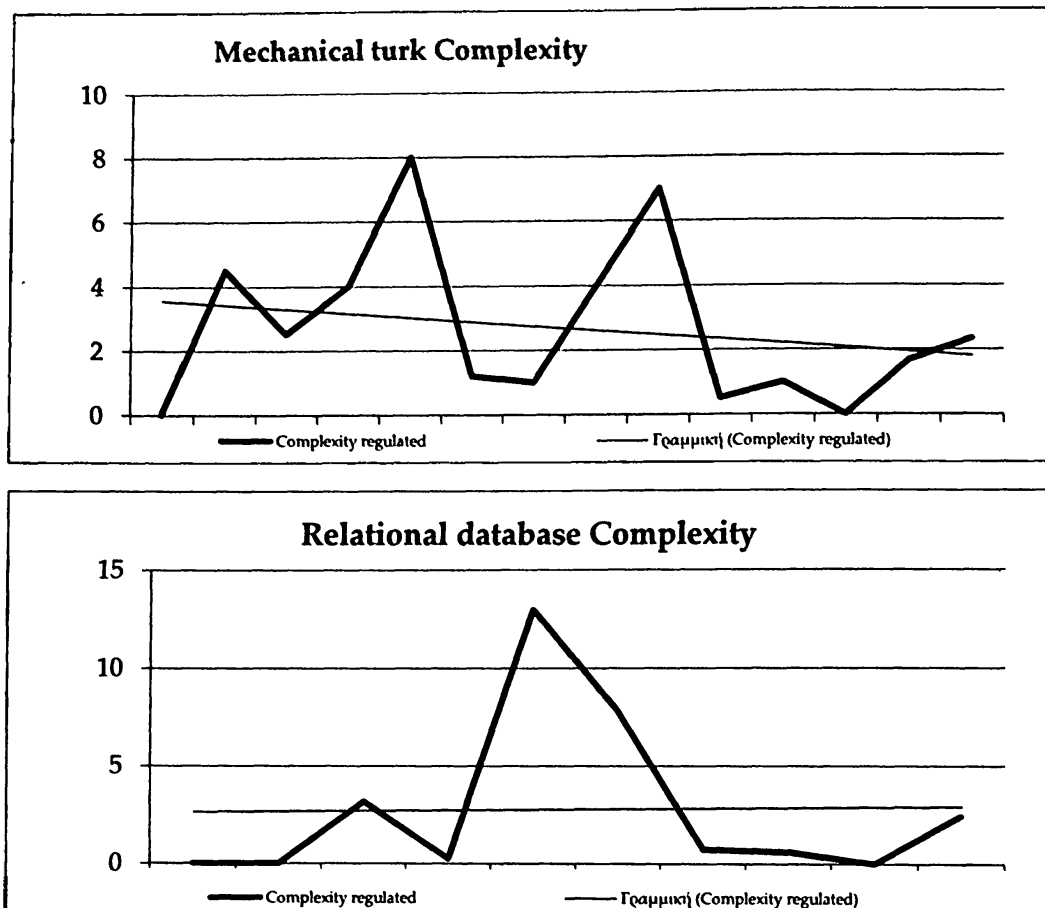


Σχήμα 5.10: Αλλαγές ανά ημέρα (work rate)



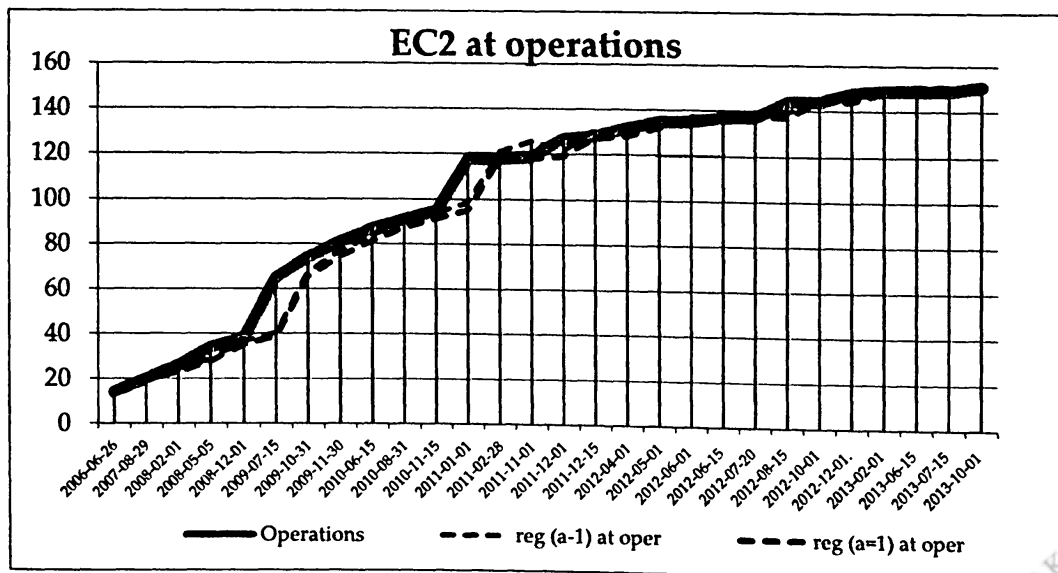
5.3.11. Μετρήσεις Πολυπλοκότητας με βάση τις operations

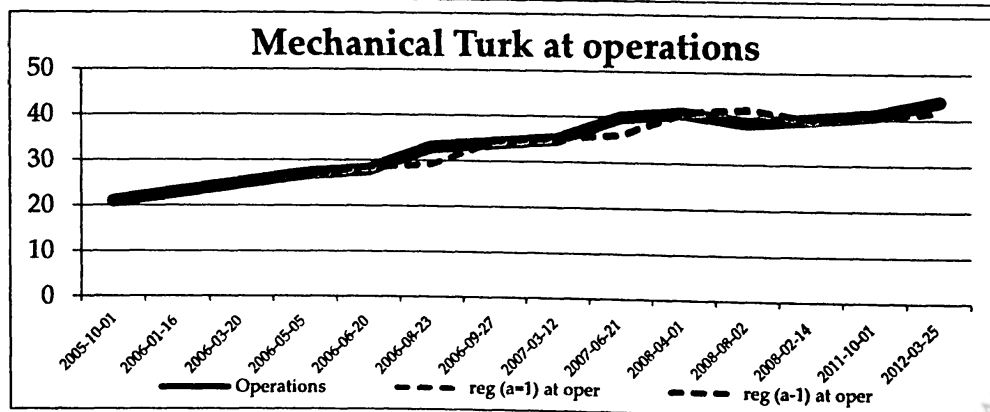
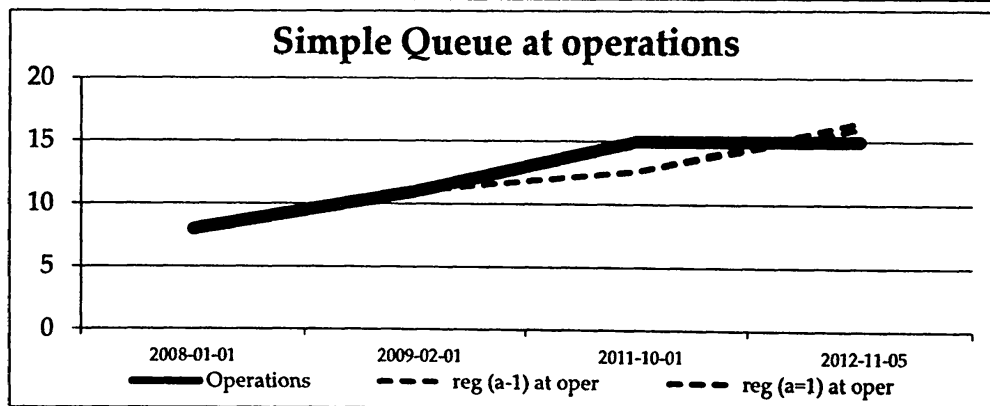
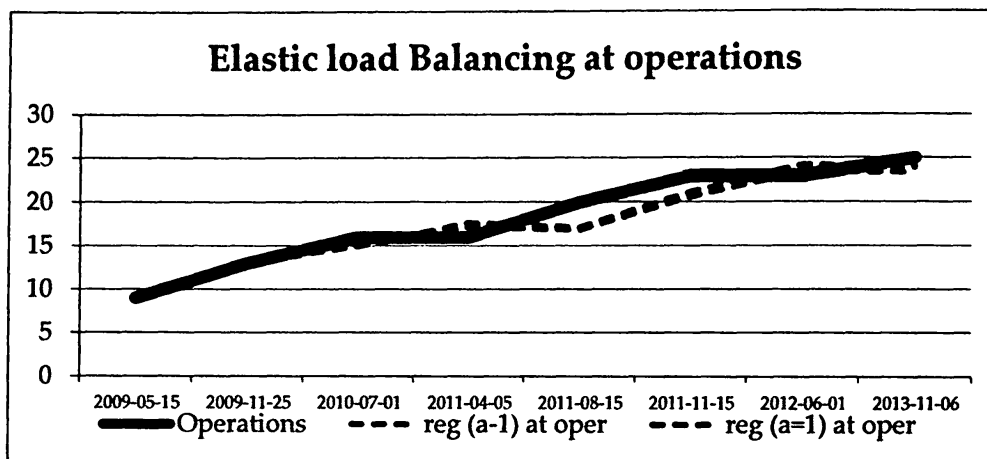
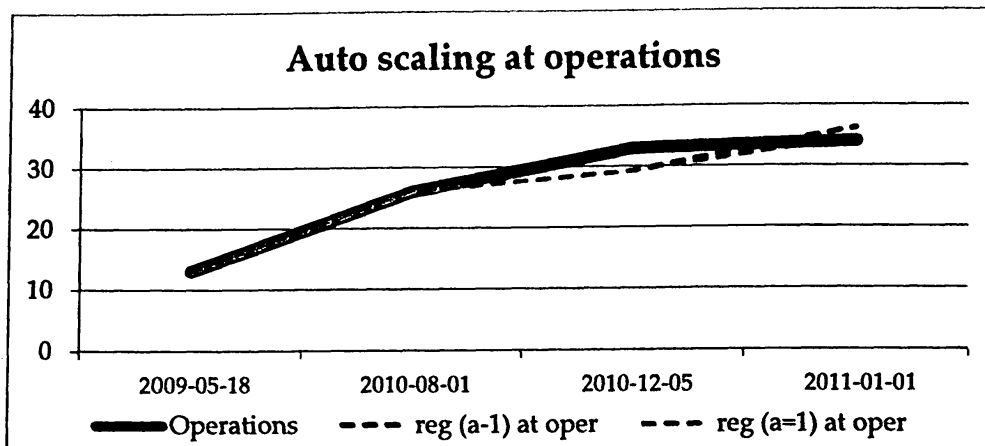


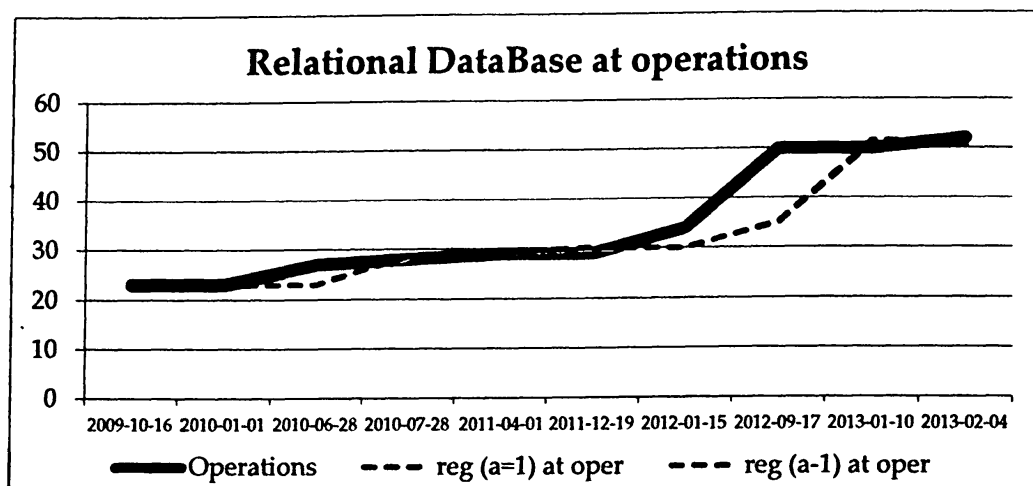


Σχήμα 5.11: Μετρήσεις Πολυπλοκότητας σε σχέση με τα operation

5.3.12. Regression analysis at operations

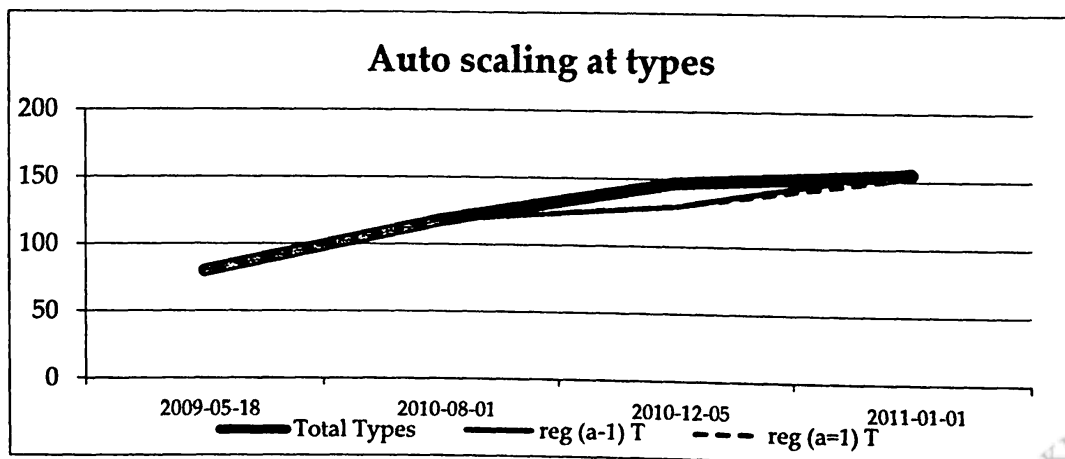
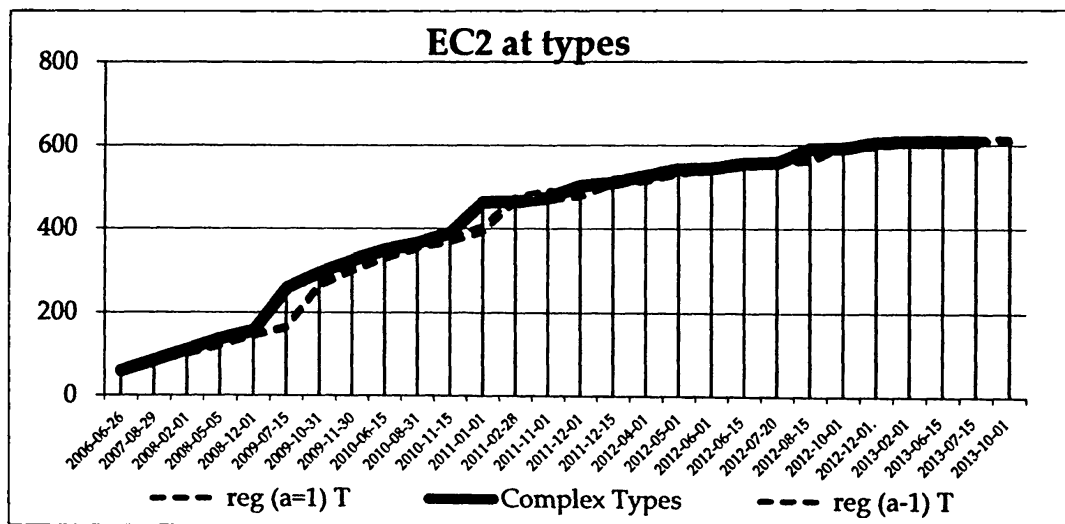


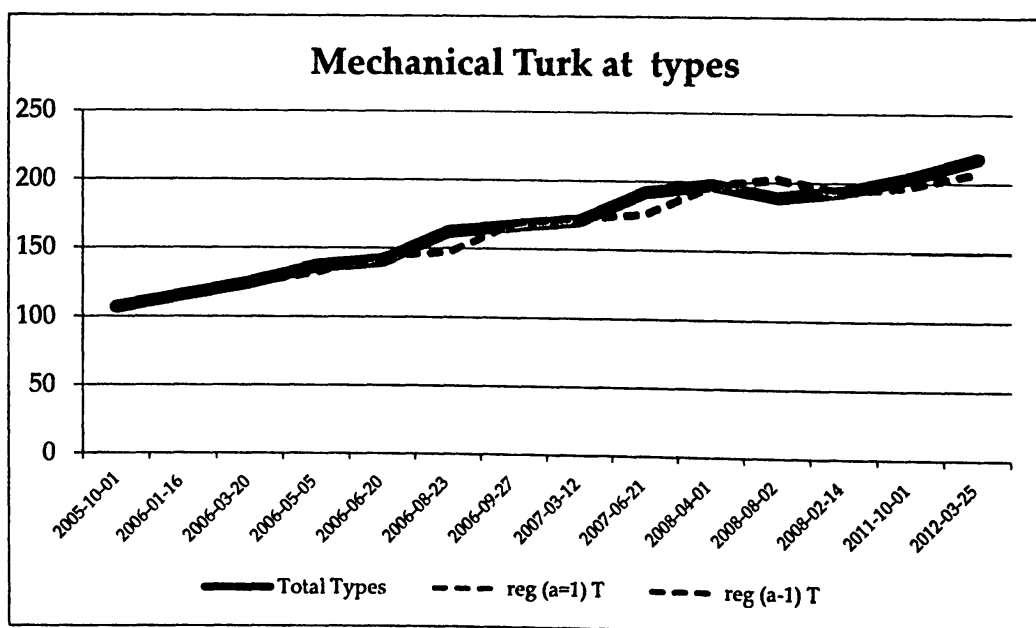
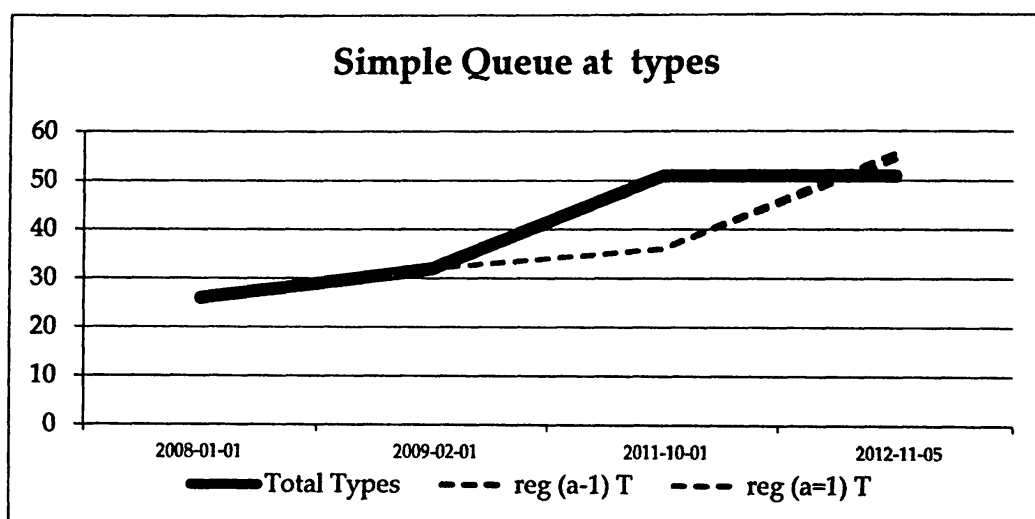
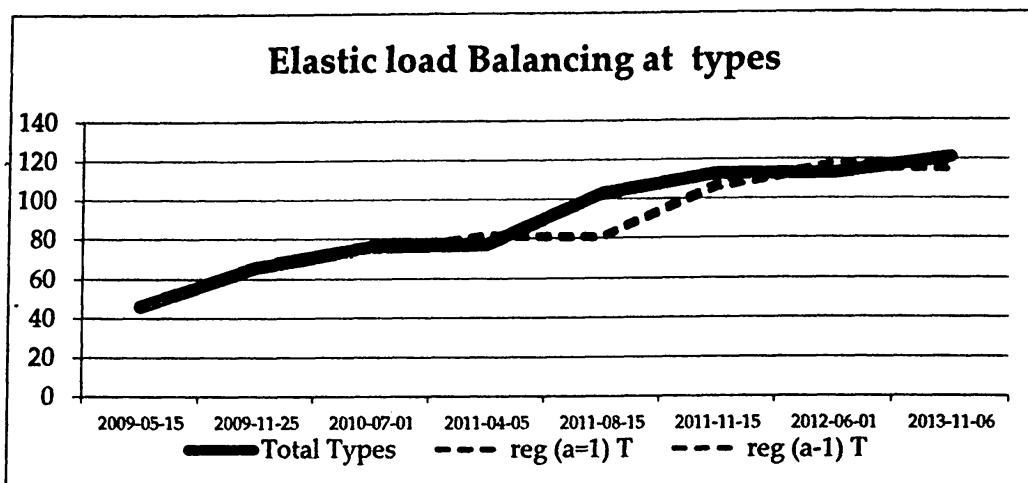


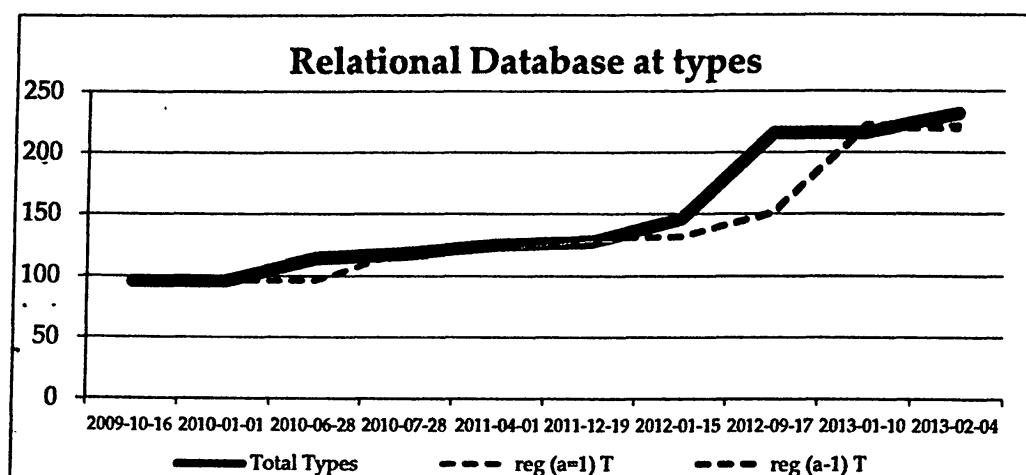


Σχήμα 5.12: Μετρήσεις Regression analysis σε σχέση με τα operation

5.3.13. Regression analysis at types







Σχήμα 5.13: Μετρήσεις Regression analysis σε σχέση με τα types

5.4. Εκτίμηση της ισχύος των νόμων του Lehman βάση των μετρήσεων για τα Amazon Web Services.

Αξιολόγηση 1ου νόμου Lehman

Το βασικό επιχείρημα του πρώτου νόμου είναι ότι το σχήμα πρέπει να αλλάζει συνεχώς με την πάροδο του χρόνου. Για να επικυρώσουμε την υπόθεση ότι ο νόμος της συνεχούς αλλαγής ισχύει, πρέπει να μελετήσουμε τις αλλαγές του σχήματος συνολικά άλλα και σε σχέση με τα operations, και τα types να εντοπίσουμε αν αυτό ισχύει.

Πιστεύουμε ότι ο πρώτος νόμος ισχύει για την περίπτωση των Amazon Web Services. Σε όλες τις περιπτώσεις των υπό μελέτη υπηρεσιών (Σχήματα 5.2, 5.3, 5.4) παρατηρούμε την ύπαρξη αλλαγών σε όλη τη διάρκεια της εξέλιξης τους.

Είναι ενδιαφέρον να επισημάνουμε ότι μπορούμε να βρούμε ακόμη εκδόσεις με ελάχιστες ή μηδενικές αλλαγές. Όπως έχει ήδη αναφερθεί, οι εκδόσεις με μηδενικές αλλαγές είναι εκδόσεις όπου οι αλλαγές μπορεί να αφορούν τον εξωραϊσμό, την τεκμηρίωση των υπηρεσιών, ή να αφορούν τεχνικού επιπέδου βελτιώσεις.

Πρέπει να επισημάνουμε ότι στα γραφήματα των αλλαγών φαίνεται να υπάρχει ένα χρονικό κενό μεταξύ των τελευταίων εκδόσεων από τις υπηρεσίες που εξετάζουμε μέχρι σήμερα. Συγκεκριμένα για τις Auto Scaling, Simple Queue Service και Mechanical Turk οι τελευταίες εκδόσεις τους με βελτιώσεις είναι όλες μέσα στο έτος 2013 ανεξάρτητα από το ότι δεν παρουσιάζουν καινούργια περιγραφή WSDL και από τις οποίες συμπεραίνουμε ότι η εξέλιξη των αλλαγών για αυτές τις υπηρεσίες δεν έχει σταματήσει απότομα.



Για να επιβεβαιώσουμε την υπόθεση ότι ο νόμος της αυξανόμενης πολυπλοκότητας ισχύει, θα πρέπει να επιβεβαιωθούν οι ακόλουθες παρατηρήσεις: (α) υπάρχει δραστηριότητα συντήρησης των υπηρεσιών που προσπαθεί να μειώσει την πολυπλοκότητα. (β) Τα γραφήματα μέτρησης της πολυπλοκότητας να παρουσιάζουν τάση αύξησης (με την πολυπλοκότητα να προσεγγίζεται από τον αριθμό των τροποποιημένων λειτουργιών προς τις προσθήκες νέων λειτουργιών ανά βήμα εξέλιξης).

Από τα γραφήματα μέτρησης πολυπλοκότητας (Σχήμα 5.10) παρατηρούμε ότι τρεις υπηρεσίες (Auto Scaling, Elastic Load Balancing, Relational Database) δείχνουν μικρή αύξηση τάσης της πολυπλοκότητας ενώ οι υπόλοιπες τρεις μικρή τάση μείωσης.

Διαισθητικά η έννοια της μέτρησης της πολυπλοκότητας έχει να κάνει με την δυσκολία υλοποίησης των αλλαγών. Η παρατήρηση αυτή δείχνει κατ'έπекταση το ποσό πολύπλοκο είναι αυτό που θέλουμε να τροποποιήσουμε στην υπηρεσία. Σύμφωνα με τον ισχυρισμό του Lehman και με απλή κατανόηση του τύπου περισσότερες τροποποιήσεις σημαίνουν μεγαλύτερη προσπάθεια στο να γίνουν αλλαγές ενώ οι προσθήκες καινούργιων λειτουργιών μικρότερη προσπάθεια.

Πρέπει να παρατηρήσουμε ότι όταν δεν έχουμε προσθήκες νέων λειτουργιών τότε ο παρονομαστής είναι μηδέν, και δεν έχουμε άλλη επιλογή από το να ορίσουμε την πολυπλοκότητα ως μηδέν. Η προσέγγιση αυτή δημιουργεί ακόμη μια διάκριση: τα σημεία στα γραφήματα που είναι μηδέν μπορεί να είναι είτε από τις τροποποιήσεις σε operations και types (στον αριθμητή του κλάσματος) είτε από τις προσθήκες σε operations και types (στον παρονομαστή του κλάσματος). Όταν τα σημεία αυτά είναι από τις τροποποιήσεις που ουσιαστικά δεν είχαμε στην νέα έκδοση, τότε το υπόλοιπο service δεν έχει επηρεαστεί με κάποια μορφή αρνητικής τροφοδότησης, ενώ όταν τα σημεία αυτά είναι από μη προσθήκες νέων operations και types μπορούμε να θεωρήσουμε ότι στο σύστημα έχουμε μόνο λειτουργίες συντήρησης. Το σύνολο των



σημείων που είναι μηδέν λόγω απουσίας τροποποιήσεων παρουσιάζονται σχηματικά στο γράφημα σαν τετράγωνα σημεία.

Στα γραφήματα της ανάπτυξης των υπηρεσιών (growth) (Σχήμα 5.8) παρατηρούμε απότομες πτώσεις σε σημεία με τιμές μηδενικές ή αρνητικές οι οποίες είναι σχεδόν στο σύνολο τους από μη προσθήκες νέων λειτουργιών το οποίο αποτελεί ισχυρή ένδειξη ότι έχουμε αρνητική τροφοδότηση.

Η δυσκολία υλοποίησης των αλλαγών δεν έχει την τάση να αυξάνεται δραστικά αν και παρατηρούμε σε όλες τις μετρήσεις, εκδόσεις με υψηλές τιμές άλλα γενικά μικρή τάση ανάπτυξης, ένδειξη ότι υπάρχει ένας μηχανισμός που βοηθάει στην συγκράτηση της. Για τους παραπάνω λόγους συμπεραίνουμε ότι ο νόμος επαληθεύεται για τις εξεταζόμενες Web Services.

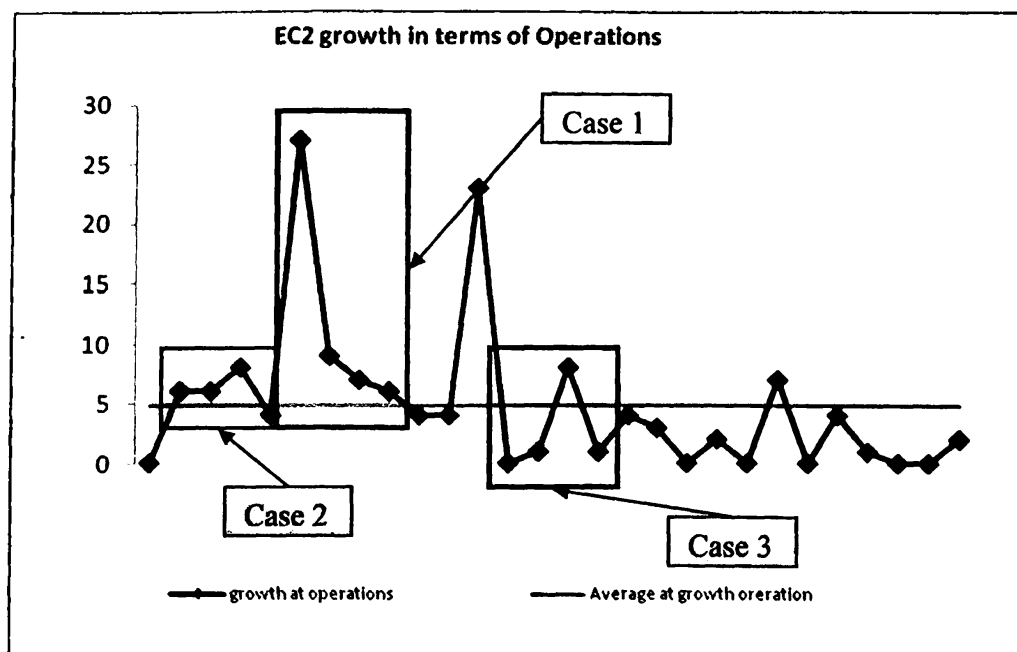
Αξιολόγηση 3^{ου} νόμου του Lehman

Για να επαληθεύσουμε την υπόθεση ότι ο νόμος της αυτορρύθμισης ισχύει, πρέπει να παρατηρήσουμε (α) επαναλαμβανόμενα μοτίβα στα γραφήματα της ανάπτυξης των υπηρεσιών, (β) μηδενικές τιμές ή αρνητικές στα γραφήματα της ανάπτυξης, (γ) με διακυμάνσεις στα γραφήματα της ανάπτυξης στα operations και τα types γύρω από τη μέση τιμή.

Μετά την μελέτη των σχημάτων της ανάπτυξης υπηρεσιών, τόσο για τα operations, όσο και για τα types (Σχήμα 5.8, Σχήμα 5.9) μπορούμε να κάνουμε κάποιες ενδιαφέρουσες παρατηρήσεις για κάθε μία από τις τρεις αναγκαίες συνθήκες.

Για το φαινόμενο το επαναλαμβανόμενων μοτίβων διακρίναμε τρεις δομές:





Σχήμα 5.14: Διαφορετικά εμφανιζόμενα μοτίβα κατά την ανάπτυξη των υπηρεσιών

- Η πρώτη περίπτωση (case 1) είναι όταν έχουμε ένα σημείο (κορυφή) και μετά ένα ή πολλά σημεία με χαμηλότερες τιμές. Αυτό το μοτίβο εμφανίζεται πιο συχνά σε όλα τα γραφήματα (operations, types) των υπηρεσιών. Συγκεκριμένα στις υπηρεσίες EC2, Elastic Load Balancing, Mechanical Turk, και Relational Database το μοτίβο επαναλαμβάνεται πάνω από δυο φορές και στα δυο γραφήματα που αφορούν τα operation και τα types, ενώ στις Auto Scaling, και Simple Queue που εξετάζουμε μικρό αριθμό νέων εκδόσεων εμφανίζεται μια φορά.
- Η δεύτερη περίπτωση παρατηρούμε μετρήσεις εκδόσεων σταθερές ή σχεδόν σταθερές (με μικρές αποκλίσεις μεταξύ τους) που ακολουθούνται από πτωτικές τιμές (μια ή περισσότερες). Το μοτίβο αυτό εμφανίζεται στις EC2, Mechanical Turk, και Relational Database.
- Στην τρίτη περίπτωση είναι όταν έχουμε σημεία ανόδου (τουλάχιστον ένα ή περισσότερα) που ακολουθούνται από εκδόσεις με μικρότερες τιμές ανάπτυξης. Το φαινόμενο αυτό είναι σπάνιο και εμφανίζεται μόνο στο EC2.

Πιστεύουμε ότι το φαινόμενο των επαναλαμβανόμενων μοτίβων ισχύει με βάση την πρώτη περίπτωση, με μια μικρή επιφύλαξη για τις δυο υπηρεσίες που ναι μεν εμφανίζεται το μοτίβο άλλα μόνο μια φορά.

Η δεύτερη παρατήρηση που πρέπει να επιβεβαιώσουμε είναι αυτής της εμφάνισης μηδενικών τιμών ή αρνητικές στα γραφήματα της ανάπτυξης. Η παρουσία τέτοιων σημείων καταδεικνύει εργασίες συντήρησης των υπηρεσιών αφού σε αυτές δεν έχουμε αύξηση του μεγέθους, και παρατηρούμε ότι οι εκδόσεις αυτές αφορούν τροποποιήσεις των λειτουργιών ή των τύπων.

Παρατηρήσαμε σε όλα τα γραφήματα ανάπτυξης των operations και των types, των εξεταζόμενων υπηρεσιών, εκδόσεις με μηδενικές τιμές το οποίο αποτελεί ισχυρή ένδειξη της ύπαρξης μηχανισμού αρνητικής τροφοδότησης για κάθε μια από αυτές.

- Επίσης παρατηρήσαμε με βάση την τρίτη περίπτωση του επαναλαμβανόμενου μοτίβου ότι σε τρεις υπηρεσίες (EC2, Mechanical Turk, και Relational Database), υπάρχουν περίοδοι σταθερότητας μεταξύ διαδοχικών εκδόσεων (δηλαδή, το μέγεθος παραμένει σταθερό, ή με μια μικρή απόκλιση). Η συμπεριφορά αυτή ενισχύει την παρατήρηση της ύπαρξης αρνητικής τροφοδότησης των υπηρεσιών αυτών καθώς λειτουργίες συντήρησης μπορεί να υπάρχουν και κρυμμένες μέσα στις εκδόσεις.
- Επίσης παρατηρούμε διακυμάνσεις πάνω και κάτω από την μέση τιμή της ανάπτυξης για όλες τις υπηρεσίες.

Βασιζόμενοι στις παρατηρήσεις μας συνολικά, μπορούμε να πούμε ο νόμος ισχύει, παρά το γεγονός ότι δεν μπορούμε να επιβεβαιώσουμε την επανάληψη του μοτίβου για δυο υπηρεσίες.



Αξιολόγηση 4^{ου} νόμου του Lehman

Για να επαληθεύσουμε την υπόθεση ότι ο νόμος του αμετάβλητου ρυθμού εργασίας ισχύει, πρέπει να παρατηρήσουμε στα γραφήματα του ρυθμού εργασίας, σε σχέση με τον χρόνο (Σχήμα 5.10), μια σχετικά ομοιόμορφη διακύμανση των τιμών πάνω και κάτω από τον μέσο όρο των τιμών. Αυτό αποτελεί ένδειξη ότι η εργασίες αλλαγών γίνονται στο χρόνο ομαλά και διαμοιρασμένες με ένα σχετικά σταθερό ρυθμό.

Για την υπηρεσία EC2 ο μέσος όρος είναι 0,9 αλλαγές ανά ημέρα και παρατηρούμε ότι οι προσθήκες και οι τροποποιήσεις που διαμοιράζονται μέσα στο χρονικό διάστημα της λειτουργίας της υπηρεσίας εντείνονται με την πάροδο του χρόνου και ειδικότερα από το 2010 και μετά, και δεν παρατηρούμε ομαλό διαμοιρασμό των αλλαγών στο χρόνο ζωής της υπηρεσίας σε σχέση με τον μέσο όρο. Πιθανολογούμε ότι η υπηρεσία γίνεται πιο αποδέκτη από το επιχειρηματικό περιβάλλον με το πέρασμα του χρόνου με αποτέλεσμα οι αλλαγές να αυξάνονται ώστε να ικανοποιηθούν οι απαιτήσεις των εταιρειών ή των χρηστών. Για τις υπηρεσίες Auto Scaling, με μέσο όρο 0,5 αλλαγές ανά ημέρα και Elastic Load Balancing με μέσο όρο 0,2 αλλαγές ανά ημέρα ισχύει η παρατήρηση που εντοπίσαμε στην υπηρεσία EC2.

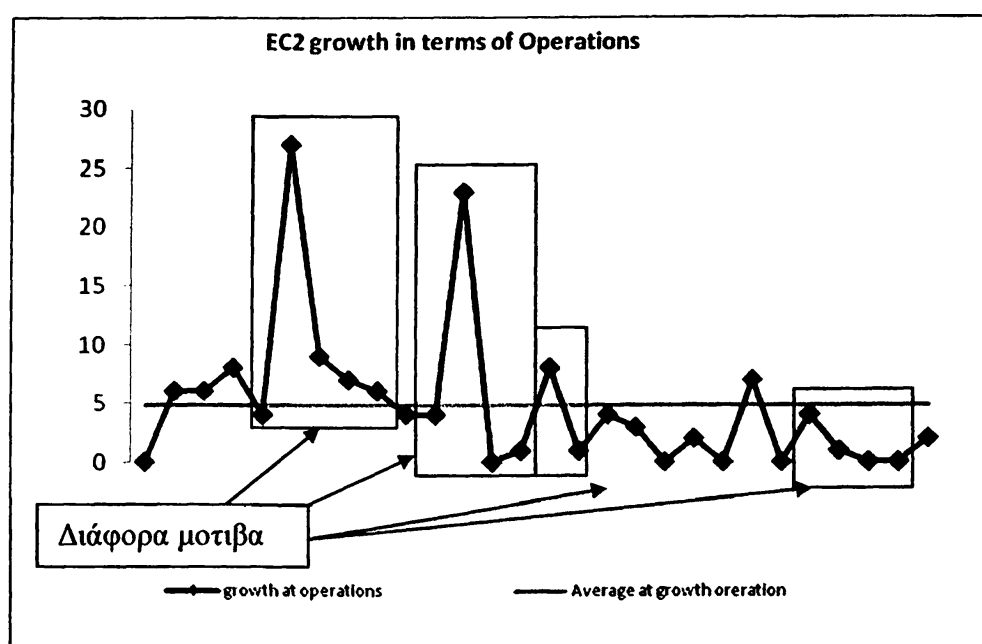
Στις υπηρεσίες Simple Queue, Mechanical Turk, οι τιμές των αντίστοιχων μέσων όρων είναι 0.035, 0.4, και παρατηρούμε ότι οι προσθήκες και οι τροποποιήσεις που διαμοιράζονται μέσα στο χρονικό διάστημα της λειτουργίας της υπηρεσίας δεν παρουσιάζουν μια σταθερότητα σε σχέση με τον μέσο όρο τους.

Η υπηρεσία Relational Database παρουσιάζει μέσο όρο αλλαγών σχεδόν μια ανά ημέρα. Σε αυτή την υπηρεσία φαίνεται μια πιο σταθερή ανάπτυξη κατά το αρχικό στάδιο της ζωής της μέχρι το 2011 όπου έχουμε μια απότομη άνοδο των αλλαγών. Ούτε για την συγκεκριμένη υπηρεσία μπορούμε να διαπιστώσουμε σταθερότητα στη διανομή των αλλαγών κατά την διάρκεια της ζωής της.

Συνολικά από τις παρατηρήσεις μας είναι προφανές ότι ο νόμος δεν έχει επαληθευτεί για τα Amazon Web Services.



Για να επαληθεύσουμε την υπόθεση ότι ο νόμος της διατήρησης της οικειότητας ισχύει, (α) πρέπει να εντοπίσουμε μοτίβα τα οποία χαρακτηρίζονται από εκδόσεις μεγάλης ανάπτυξης που ακολουθούνται από εκδόσεις με μικρότερο ρυθμό ανάπτυξης, και πιο συγκεκριμένα ύπαρξη απότομων μεταβολών μετά από υψηλές τιμές (peaks) σε χαμηλότερες τιμές (valleys) και (β) ότι η ανάπτυξη του σχήματος κατά την διάρκεια του χρόνου έχει τάσεις σταθεροποίησης.



Σχήμα 5.15: Μοτίβα με υψηλές τιμές που ακολουθούν χαμηλές τιμές.

Χρησιμοποιώντας το παράδειγμα της ανάπτυξης των operation του EC2 (Σχήμα 5.14) παρατηρούμε μοτίβα με εκδόσεις που έχουμε υψηλές τιμές και ακολουθούνται από εκδόσεις με μικρότερες τιμές.

Τα μοτίβα αυτά εμφανίζονται σε όλα τα γραφήματα ανάπτυξης που αφορούν τα operation και τα types με συχνότερη την εμφάνιση τους στις EC2, Mechanical Turk, και Elastic Load Balancing

Επίσης η παρατήρηση που κάναμε στον τρίτο νόμο ότι η ανάπτυξη δείχνει μικρή συγκρατημένη αύξηση στο EC2, σταθερή για τέσσερις υπηρεσίες τις Auto Scaling,

Simple Queue, Mechanical Turk, και Elastic Load Balancing και ελαφρά πτωτική τάση για την Relational Database αποτελεί ακόμη μια ένδειξη ότι μηχανισμός τροφοδότησης των υπηρεσιών τείνει να μειώνει τις αλλαγές στις επόμενες εκδόσεις ώστε να δοθεί ο κατάλληλος χρόνος και προσπάθεια από τους εμπλεκόμενους στο σύστημα να απορροφήσουν και να κατανοήσουν το περιεχόμενο των αλλαγών.

Οι παραπάνω παρατηρήσεις μας οδηγούν στο συμπέρασμα ότι ο 5^{ος} νόμος επαληθεύεται για την μελέτη των Web Services.

Αξιολόγηση 6^{ου} νόμου του Lehman

Ο έκτος νόμος της συνεχιζόμενης αύξησης απαιτεί από εμάς να εξακριβώσουμε κατά πόσον η παρεχόμενη λειτουργικότητα των υπηρεσιών αυξάνεται συνεχώς. Για το σκοπό αυτό, μελετάμε τα μεγέθη των υπηρεσιών, για τις λειτουργίες (operations) και για τους τύπους δεδομένων (types) (Σχήμα 5.5, Σχήμα 5.6).

Σε όλες τις περιπτώσεις, το μέγεθος του υπηρεσιών αυξάνει μακροπρόθεσμα, και όλα τα σύνολα δεδομένων που αφορούν τα operations και τα types καταδεικνύουν την τάση να αυξάνονται με την πάροδο του χρόνου.

Επίσης από τα δεδομένα που συλλέξαμε προκύπτει ότι:

- Για την υπηρεσία EC2 από τις 29 εκδόσεις μόνο στις 5 δεν έχουμε προσθήκες νέων λειτουργιών, ενώ έχουμε προσθήκες τύπων δεδομένων στις 27 από τις 29.
- Για τη υπηρεσία Auto Scaling, είχαμε σε όλες προσθήκες νέων λειτουργιών.
- Για τη υπηρεσία Mechanical Turk, είχαμε σε όλες προσθήκες νέων λειτουργιών εκτός μιας.
- Για τη υπηρεσία Elastic Load Balancing, δεν είχαμε προσθήκες νέων λειτουργιών σε 2 εκδόσεις από το σύνολο των 9.
- Για την υπηρεσία Simple Queue, παρατηρούμε ότι μόνο σε μια έκδοση είχαμε προσθήκες νέων λειτουργιών αλλά μπορούμε να δικαιολογήσουμε την εξαίρεση αυτή γνωρίζοντας ότι η υπηρεσία αυτή, η βασική της λειτουργία



είναι να μεταφέρει μηνύματα, και η εξέλιξη σε επίπεδο λειτουργιών είναι πολύ περιορισμένη.

Από τις παραπάνω παρατηρήσεις, μπορούμε να συμπεράνουμε ότι ο νόμος ισχύει.

Αξιολόγηση 7^{ου} νόμου του Lehman

Ο έβδομος νόμος μεταφράζεται για την περίπτωση των Web Services, ως πτώση της ποιότητας των υπηρεσιών σε σχέση με την ηλικιακή γήρανση τους, εκτός και αν πειθαρχημένα προσαρμόζονται στο εξωτερικό περιβάλλον τους.

Όπως έχουμε ήδη αναφερθεί, η ποιότητα εξετάζεται υπό το πρίσμα της εσωτερικής ποιότητας άλλα και της εξωτερικής ποιότητας.

Η εσωτερική ποιότητα σύμφωνα με την θεώρηση του Lehman συνδέεται με την μέτρηση της πολυπλοκότητας των υπηρεσιών. Το συμπέρασμα για την ισχύ του 2^{ου} νόμου αποτελεί και ισχυρή ένδειξη ότι η εσωτερική ποιότητα των υπηρεσιών που εξετάσαμε δεν φθίνει καθώς οι μετρήσεις μας έδειξαν ότι η πολυπλοκότητα στο σύνολο των υπηρεσιών δεν αυξάνεται σημαντικά.

Όσο αφορά στην εξωτερική ποιότητα μελετήσαμε τις περιγραφές των εργασιών που συνοδεύουν την δημοσιοποίηση νέων εκδόσεων μελετήσαμε τις περιγραφές των εκδόσεων και εντοπίσαμε εκδόσεις με βελτιώσεις, στην ασφάλεια, στις επιδόσεις. Επίσης εντοπίσαμε εκδόσεις με εργασίες αποσφαλμάτωσης του λογισμικού των υπηρεσιών, εκδόσεις με βελτιώσεις στην διαθεσιμότητα των υπηρεσιών, και εκδόσεις που εμπλουτίζουν την λειτουργικότητα αυτών.

Οι παρατηρήσεις μας όσο αφορά στην ποιότητα (εσωτερική, εξωτερική) των υπηρεσιών μας οδηγούν στο συμπέρασμα ότι το γενικότερο νόημα του νόμου φαίνεται να ισχύει σεβόμενοι πάντα την παρατήρηση ότι η ποιότητα του συστήματος ορίζεται διαφορετικά με βάση την οπτική που έχουν όλοι οι εμπλεκόμενοι στο σύστημα και αυτή μπορεί να οριστεί διαφορετικά από αυτούς.



Αξιολόγηση 8^{ου} νόμου του Lehman

Για να επικυρώσουμε τον όγδοο νόμο, θα πρέπει να αποδείξουμε ότι ο παρακάτω τύπος μπορεί να εκτιμήσει την εξέλιξη του μεγέθους των υπηρεσιών, με ακρίβεια ή, με μικρό σφάλμα σε σύγκριση με τις πραγματικές τιμές. Στην μελέτη μας αυτή εξετάζουμε τα αποτελέσματα που παράγει ο τύπος σε σχέση με τα operations και τα types.

$$\text{Ο τύπος είναι της μορφής: } \hat{S}_i = \hat{S}_{i-1} + \frac{\bar{E}}{\hat{S}_{i-1}^2}$$

όπου S είναι το εκτιμώμενο μέγεθος του συστήματος και E είναι η παράμετρος του μοντέλου που προσεγγίζει την προσπάθεια (Effort).

Στην εργασία μας ακολουθούμε την προσέγγιση του Lehman στην εργασία του [5.11] στην οποία προτείνει τον υπολογισμό της παραμέτρου E ως την μέση τιμή των επιμέρους E , που προκύπτουν ένα ανά βήμα εξέλιξης. Στη συνέχεια, υπολογίζουμε τις επιμέρους παραμέτρους για κάθε βήμα εξέλιξης από τον τύπο

$$E_i = \frac{s_i - s_a}{\sum_{j=a}^{i-1} \frac{1}{s_j^2}}$$

Όπου το s_i αντιπροσωπεύει το πραγματικό μέγεθος της υπηρεσίας στην έκδοση i και με a αναφέρεται στην έκδοση από την οποία ξεκινά την καταμέτρηση. Συγκεκριμένα, [5.11] προτείνονται δύο τιμές για το a , και συγκεκριμένα (i) $a = 1$ (η πρώτη έκδοση) και (ii) $a = s_{i-1}$ (η προηγούμενη έκδοση)

Από τα αποτελέσματα που παρουσιάζονται στα γραφήματα (Σχήμα 5.12, Σχήμα 5.13) συμπεραίνουμε ότι

- Η εκτίμηση που παράγει ο τύπος με υπολογισμό της παραμέτρου E για $a = 1$, στο σύνολο των εξεταζόμενων υπηρεσιών προκύπτει μια πολύ μικρή απόκλιση στα σημεία των απότομων αυξητικών αλλαγών.



- Η εκτίμηση που παράγει ο τύπος με υπολογισμό της παραμέτρου E για, $\alpha = s_{t-1}$ στο σύνολο των εξεταζομένων υπηρεσιών είναι πιο ακριβής.

Οι παραπάνω παρατηρήσεις μας οδηγούν στο συμπέρασμα ότι ο νόμος ισχύει για την περίπτωση των εξεταζόμενων Amazon Web Services.



ΚΕΦΑΛΑΙΟ 6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην εργασία αυτή αξιολογήσαμε τη δυνατότητα εφαρμογής των νόμων του Lehman στις υπηρεσίες ιστού της Amazon. Διεξάγαμε την πρώτη ουσιαστική μελέτη για τις υπηρεσίες αυτές με σκοπό εξαγωγή χρήσιμων συμπερασμάτων για τους εξελικτικούς νόμους που διέπουν ένα σύνολο των υπηρεσιών αυτής.

Συλλέξαμε σύνολα δεδομένων για έξι διαφορετικές υπηρεσίες τα οποία αναλύσαμε με την βοήθεια εργαλείου λογισμικού που δημιουργήσαμε.

Η επεξεργασία των δεδομένων μας βοήθησε να εντοπίσουμε πλήθος αλλαγών κατά την διάρκεια των εξελικτικών βημάτων. Επικεντρωθήκαμε στα κυρίαρχα χαρακτηριστικά που παρέχουν οι υπηρεσίες αυτές, τις λειτουργίες (operations), και στις αλλαγές αυτών (προσθήκες, αφαιρέσεις, τροποποιήσεις), καθώς και στους τύπους δεδομένων που σχετίζονται με τις λειτουργίες. Παρουσιάσαμε μετρήσεις, για τα μεγέθη, το ύψος των αλλαγών ανά έκδοση που εντοπίσαμε, την πολυπλοκότητα, την ανάπτυξη των υπηρεσιών, σε σχέση με τις εκδόσεις και το χρόνο ζωής των υπηρεσιών.

Αναλύσαμε τη συμπεριφορά των υπηρεσιών βασισμένοι σε διαφορετικές μετρήσεις και εξάγαμε συμπεράσματα για την ισχύ των νόμων του Lehman σε σχέση με την ύπαρξη τους στην εξελικτική διαδικασία των υπηρεσιών. Σύμφωνα με τις παρατηρήσεις μας, η συμπεριφορά των υπηρεσιών καταδεικνύει την ύπαρξη μηχανισμού ανατροφοδότησης, και επαληθεύει το σύνολο των νόμων πλην του τετάρτου νόμου.



ΑΝΑΦΟΡΕΣ

- [1] John Wilkins. "Defining Evolution". ACM - Reports of the National Center for Science Education, Volume: 21, Issue: 1-2, pg. 29-37, Year: 2001
- [2] Software Engineering Standards Committee. "IEEE Standard for Software Maintenance". IEEE Std 1219-1998 (Revision of IEEE Std 1219-1992), pg 3, 1998.
- [3] Israel Herraiz, Daniel Rodriguez, Gregorio Robles and Jesus M. Gonzalez-Barahona. "The evolution of the laws of software evolution. A discussion based on a systematic literature review". ACM - Computing Surveys, Vol.46, Issue 2, Article No. 28, November 2013.
- [4] M. M. Lehman. "Programs, Cities, Students: Limits to Growth?". In: PROGRAMMING METHODOLOGY: A Collection of Articles by Members of IFIP WG2.3, Edit by David Gries, Part I, pg 42-69, 1974.
- [5][2.1] IBM Developer Works. "New to Soa and web services". Available at <http://www.ibm.com/developerworks/webservices/newto/service.html>
- [6] Vasilios Andrikopoulos, Salima Benbernou, and Michael P. Papazoglou. "On the Evolution of Services". IEEE Transactions on Software Engineering, Vol. 38, Issue 3, pg 609-628, May 2012,.
- [7] Jinesh Varia - Sajee Mathew, "Overview of Amazon Web Services". Available at: http://media.amazonwebservices.com/AWS_Overview.pdf, March 2013,



[8] Microsoft Developer Network. "XML Web Services Basics". Available at: <http://msdn2.microsoft.com/en-us/library/ms996507.aspx>

[9] The Developer.com NetWork. "Understanding XML and XML Schema-Part 1" <http://www.developer.com/xml/article.php/2195981/web-services-tutorial-understanding-xml-and-xml-schemamdashpart-1.htm>

[10] W3C. "SOAP Version 1.2 Part 0: Primer (Second Edition)". W3C Recommendation (Second Edition), 27 April 2007. Available at: <http://www.w3.org/tr/2007/rec-SOAP12-part0-20070427/>

[11], Oracle and/or its affiliates. "Oracle® Fusion Middleware. Publishing and Finding Web Services Using UDDI". Chapter 22, November 2011. Available at: http://docs.oracle.com/cd/E23943_01/web.1111/e13734.pdf

[12] W3C. "Web Services description language (WSDL) 1.1", w3c note.15, March 2001. Available at: <http://www.w3.org/tr/WSDL>

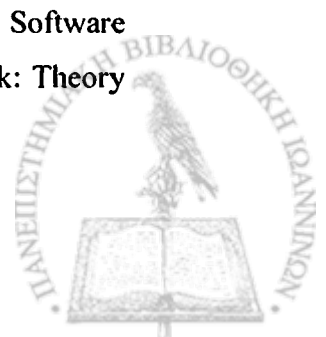
[13] Marios Fokaefs, Rimon Mikhael, Nikolaos Tsantalis, Eleni Stroulia. "An Empirical Study on Web Service Evolution", Proc. International Conference on Web Services, pg.49-56, 2011.

[14] M. M. Lehman. "Laws of Software Evolution Revisited". 5th European Workshop on Software Process Technology (EWSPT '96), pg: 108-124, Nancy, France, October 9-11, 1996.

[15] Laszlo A. Belady, M. M. Lehman. "A Model of Large Program Development". IBM Systems Journal 15(3), pg: 225-252, 1976.

[16] Lehman. M.M., "Programs, life cycles, and laws of software evolution," *Proceedings of the IEEE*, vol.68, no.9, pg.1060-1076, Sept. 1980.

[17] Meir M. Lehman and Juan C. Fernandez-Ramil. Rules and Tools for Software Evolution Planning and Management. In: Software Evolution and Feedback: Theory



and Practice. Edited by Nazim H. Madhavji, Juan C. Fernandez-Ramil, and Dewayne E. Perry, John Wiley & Sons Ltd, 2006.

[18] Norman E. Fenton, Shari Lawrence Pfleeger. Software metrics - a practical and rigorous approach. PWS Publishing Co., pg 14-20, Boston, MA, 1998.

[19] M M Lehman, J F Ramil. "Software Evolution - Part II Modelling Long-Term Growth Trends in Software as an Example of FEAST Output". Annual Distinguished Lecture, De Montfort Univ., Leicester, 20 Dec. 2001. Available at:
<http://www.eis.mdx.ac.uk/staffpages/mml/feast2/papers/pdf/jfr103c.pdf>

[20] Juan F. Ramil, M. M. Lehman. Effort estimation from change records of evolving software (poster). Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000), pg: 777, Limerick Ireland, June 4-11, 2000.

[21] M. M. Lehman, D. E. Perry and J. F. Ramil "On Evidence Supporting the FEAST Hypothesis and the Laws of Software Evolution", Proc. 5th IEEE International Software Metrics Symposium (METRICS 1998), pg: 84-88. March 1998, USA.

[22] Guowu Xie, Jianbo Chen, Neamtui, I. "Towards a better understanding of software evolution: An empirical study on open source software.". In *Proc. 25th IEEE Int'l Conf. on Soft. Maint. (ICSM 2009)*, pg.51-60, 20-26 Sept. Edmonton, Canada 2009.

[23] M. M. Lehman. Laws of Software Evolution Revisited. 5th European Workshop on Software Process Technology (EWSPT '96), pg: 108-124. Nancy, France, October 9-11, 1996,

[24] Lehman, M.M.; Ramil, J.F. Wernick, P. D. Perry, D.E. Turski, W.M., "Metrics and Laws of Software Evolution - The Nineties View". Proc. Fourth International Software Metrics Symposium, pg.20-32, 5-7 Nov 1997.



ΠΑΡΑΡΤΗΜΑ

Πίνακας Π.6.1: Το σύνολο των εκδόσεων για την υπηρεσία EC2.

Version Release	Latest WSDL	Detail Description
Release: Amazon EC2 on 2009 03 11	2009--03--01	This release introduces the Amazon EC2 Reserved Instances.
Release: Amazon EC2 on 2009 04 14	2009--03--01*	This release introduces Amazon EC2 Reserved Instances in Europe.
Release: Amazon EC2 on 2009 04 22	2009--03--02	This release introduces Amazon EC2 running IBM software.
Release: Amazon EC2 on 2009 05 17	2009--04--04	This release introduces Amazon CloudWatch, Auto Scaling, and Elastic Load Balancing.
Release: Amazon EC2 on 2009 08 26	2009--07--15	This release introduces changes to Amazon EC2 to support Amazon Virtual Private Cloud.
Release: Amazon EC2 on 2009 09 23	2009--08--15	This release introduces shared snapshots.
Release: Amazon EC2 on 2009 10 01	2009--08--15	This release introduces the consolidation of Windows AMIs.
Release: Amazon EC2 on 2009 10 26	2009--08--17	This release introduces High-Memory instance types.
Release: Amazon EC2 on 2009 11 11	2009--08--18	This release introduces .NET language-specific APIs.
Release: Amazon EC2 on 2009 12 02	2009--10--31	This release introduces AMIs backed by Amazon EBS.
Release: Amazon EC2 on 2009 12 09	2009--10--32	This release introduces support for Windows 2008.
Release: Amazon EC2 on 2009 12 14	2009--11--30	This release introduces Spot Instances.
Release: Amazon EC2 on 2010 02 22	2009--11--30	This release introduces the new High-Memory Extra Large instance type, and support for Reserved Instances with Windows.
Release: Amazon EC2 on 2010 03 22	2009--11--30	New AWS SDK for Java.
Release: Amazon EC2 on 2010 04 28	2009--11--30	Support for Asia Pacific (Singapore) Region.
Release: Amazon EC2 on 2010 06 14	2009--11--30	This release introduces Amazon CloudWatch monitoring for Amazon EBS volumes
Release: Amazon EC2 on 2010 07 12	2010--06--15	Support for the new cluster compute instance type (cc1.4xlarge), and the ability to specify the IP address for an instance launched in an Amazon VPC subnet.
Release: Amazon EC2 on 2010 09 02	2010--06--15	Support for the new cluster compute instance type (cc1.4xlarge), and the ability to specify the IP address for an instance launched in an Amazon VPC subnet.
Release: Amazon EC2 on 2010 09 08	2010--06--15	This release introduces the new micro instance type
Release: Amazon EC2 on 2010 09 19	2010--08--31	Support for tags, filters, idempotent RunInstance calls, and ImportKeyPair.
Release: Amazon EC2 on 2010 09 28	2010--08--31	New AWS SDK for PHP.
Release: Amazon EC2 on 2010 11 14	2010--08--31	Support for the new cluster GPU instance type (cg1.4xlarge).
Release: Amazon EC2 on 2010 12 02	2010--08--31	Support for Basic Monitoring for EC2 instances.
Release: Amazon EC2 on 2010 12 15	2010--11--15	Support for VM Import, and a new API version: 2010-11-15

Version Release	Latest WSDL	Detail Description
Release: Amazon EC2 on 2010 12 16	2010-11-15	Support for Oracle enterprise applications on Amazon EC2.
Release: Amazon EC2 on 2011 03 14	2011-01-01	Support for Amazon VPC Internet gateway, updated metadata, and a new API version: 2011-01-01
Release: Amazon EC2 on 2011 03 15	2011-01-01	Support for Windows Server 2008 R2
Release: Amazon EC2 on 2011 03 27	2011-02-28	New Dedicated Instances option for instances launched into a VPC, and a new API version: 2011-02-28
Release: Amazon EC2 on 2011 05 24	2011-04-05	Support for IPv6, zone apex domain names, application instance lockdown, and a new API version: 2011-04-05.
Release: Amazon EC2 on 2011 05 26	2011-05-15	Support for Amazon Spot Instances Availability Zone Local Pricing Changes and a new API version: 2011-05-15.
Release: Amazon EC2 on 2011 08 24	2011-07-15	Amazon EC2 VM Import Support for Windows 2003 and VHD file format.
Release: Amazon EC2 on 2011 08 30	2011-08-15	This release provide health checks for back-end server using HTTPS, back-end server authentication and a new API version: 2011-08-15.
Release: Amazon EC2 on 2011 09 15	2011-07-15	Amazon EC2 VM Import CLI Updates.
Release: Amazon EC2 on 2011 10 11	2011-07-15	This release of Amazon EC2 introduces support for EC2 Spot Instances in the Amazon Virtual Private Cloud (Amazon VPC).
Release: Amazon EC2 on 2011 11 17	2011-11-01	This release of Amazon EC2 introduces the DescribeInstanceStatus API.
Release: Amazon EC2 on 2011 12 01	2011-11-01	This release of Amazon EC2 introduced new choices of Reserved Instance offerings that address different projected uses of instances.
Release: Amazon EC2 on 2011 12 21	2011-12-01	Support for New Elastic Network Interfaces (ENIs) for EC2 Instances in a VPC.
Release: Amazon EC2 on 2011 12 30	2011-12-15	Support for Amazon instance status checks and a new API version: 2011-12-15
Release: Amazon EC2 on 2012 03 05	2011-12-15	This release of Amazon EC2 introduces Reserved Instance Pricing Tiers and On-Demand price updates
Release: Amazon EC2 on 2012 03 07	2011-12-15	This release of Amazon EC2 introduces 64-bit option for all Amazon Machine Images, a new medium instance type, m1.medium and support for the MindTerm SSH client to allow customers to connect to their Linux instances through a web browser.
Release: Amazon EC2 2012 03 12	2012-03-01	This release of Amazon EC2 introduces volume status checks and a new API version: 2012-03-01.
Release: Amazon EC2 on 2012 04 19	2012-04-01	This release of Amazon EC2 introduces support for AWS Marketplace and a new API version: 2012-04-01.
Release: Amazon EC2 on 2012 04 26	2012-04-01	This release of Amazon EC2 introduces support for cc2.8xlarge instances in Amazon Virtual Private Cloud (Amazon VPC).
Release: Amazon EC2 on 2012 05 18	2012-04-02	This release of Elastic Load Balancing adds console support for managing listeners, certificates, and cipher settings for existing load balancers. This release also includes changes to allow Marketplace AMIs to be registered with a load balancer.

Version Release	Latest WSDL	Detail Description
Release: Amazon EC2 on 2012 05 25	2012--05--01	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces support for exporting Windows Server instances.
Release: Amazon EC2 on 2012 06 07	2012--05--01	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces several new Spot Instance, new features include integration with Auto Scaling and AWS CloudFormation.
Release: Amazon EC2 on 2012 06 11	2012--06--01	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces support for AWS Identity and Access Management (IAM) roles on EC2 instances.
Release: Amazon EC2 on 2012 07 11	2012--06--15	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces support multiple IP addresses for Amazon EC2 instances in Amazon.
Release: Amazon EC2 on 2012 07 18	2012--06--15	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces support for high I/O instances (hi1.xlarge)
Release: Amazon EC2 on 2012 07 31	2012--07--20	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces a new Amazon Elastic Block Store (EBS) . This release also introduces the ability to launch selected Amazon EC2 instance types as EBS-Optimized instances.
Release: Amazon EC2 on 2012 09 11	2012--08--15	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces a new Reserved Instance Marketplace. This release also includes a new WSDL 2012-08-15.
Release: Amazon EC2 on 2012 10 31	2012--10--01	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces the new M3 extra-large and M3 double-extra-large instance types. These instance types are only available in the US East (N. Virginia) Region.
Release: Amazon EC2 on 2012 10 14	2012--10--01	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces the new Spot Instance Request Status. This release also includes a new WSDL 2012-10-01.
Release: Amazon EC2 on 2012 11 01	2012--10--01	This release of Amazon Elastic Compute Cloud (Amazon EC2) increases the maximum number of IOPS (input/output operations per second) that you can provision for an Amazon EBS volume.
Release: Amazon EC2 on 2012 12 17	2012--12--01	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces the ability to copy an Amazon Elastic Block Store (Amazon EBS) snapshot.
Release: Amazon EC2 on 2013 03 11	2013--02--01	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces support for copying an AMI from one AWS region to another and support for default virtual private clouds (VPC).
Release: Amazon EC2 on 2013 07 08	2013--06--15	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces support for creating IAM policy statements that apply to particular Amazon EC2 resources.
Release: Amazon EC2 on 2013 08 20	2013--07--15	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces support for controlling the assignment of public IP addresses to instances launched in a VPC.
Release: Amazon EC2 on 2013 09 11	2013--08--15	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces support for modifying Reserved Instances in a region.
Release: Amazon EC2 on 2013 10 09	2013--10--01	This release of Amazon Elastic Compute Cloud (Amazon EC2) introduces support for Reserved Instance instance type modifications.

Πίνακας Π.6.2: Το σύνολο των εκδόσεων για την υπηρεσία Elastic Load Balancing

Release Date	WSDL	Description
Release: Elastic Load Balancing on 2009-05-18	15/5/2009	This is an initial public beta release of Elastic Load Balancing.
Release: Elastic Load Balancing on 2009-11-11	15/5/2009	New AWS SDK for .NET.
Release: Elastic Load Balancing on 2009-12-02	15/5/2009	Updates for the new US-West (Northern California) Region.
Release: Elastic Load Balancing on 2010-03-22	15/5/2009	New AWS SDK for Java.
Release: Elastic Load Balancing on 2010-04-07	25/11/2009	This release adds support for sticky sessions.
Release: Elastic Load Balancing on 2010-04-28	25/11/2009	Adds support for the Asia Pacific (Singapore) region.
Release: Elastic Load Balancing on 2010-09-02	25/11/2009	Adds support for AWS Identity and Access Management (IAM).
Release: Elastic Load Balancing on 2010-10-14	1/7/2010	Adds HTTPS Support.
Release: Elastic Load Balancing on 2010-10-27	1/7/2010	This update adds X-Forwarded-Proto and X-Forwarded-Port headers.
Release: Elastic Load Balancing on 2011-05-24	5/4/2011	Support for IPv6, zone apex domain names, application instance lockdown, and a new API version: 2011-04-05.
Release: Elastic Load Balancing on 2011-08-30	15/8/2011	This release provides support for secure communication from the corresponding Elastic Load Balancer to application servers, health checks for back-end server using HTTPS, back-end server authentication, and a new API version: 2011-08-15.
Release: Elastic Load Balancing on 2011-11-21	15/11/2011	This release adds support for using Elastic Load Balancing in Amazon VPC applications, and a new API version: 2011-11-15.
Release: Elastic Load Balancing on 2012-06-10	1/6/2012	This release of Elastic Load Balancing introduces Internal Load Balancing in Amazon Virtual Private Cloud (Amazon VPC).
Release: Elastic Load Balancing on 2013-05-20	1/6/2012	This release of Elastic Load Balancing adds support for all HTTP methods.
Release: Elastic Load Balancing on 2013-07-30	2/6/2012	This release of Elastic Load Balancing adds support for Proxy Protocol.
Release: Elastic Load Balancing on 2013-11-06	1/6/2012	This release of Elastic Load Balancing provides an option to enable Cross-Zone Load Balancing for your load balancer.

Πίνακας Π.6.3: Το συνολο των εκδόσεων για την υπηρεσία Auto Scaling.

Release Date	WSDL Version	Description
Release: Auto Scaling on 2009-05-18	15/5/2009	Introducing Auto Scaling. This is an initial public beta release of Auto Scaling.
Release: Auto Scaling on 2009-11-11	15/5/2009	New AWS SDK for .NET.
Release: Auto Scaling on 2009-12-02	15/5/2009	Updates for the new US-West (Northern California) Region
Release: Auto Scaling on 2010-03-22	15/5/2009	New AWS SDK for Java.
Release: Auto Scaling on 2010-04-28	15/5/2009	Added support for new Region.
Release: Auto Scaling on 2010-09-02	15/5/2009	Auto Scaling now supports the AWS Identity and Access Management (IAM) web service
Release: Auto Scaling on 2011-07-25	1/1/2011	This release of Auto Scaling includes support for Amazon SNS notifications. You can now run Spot Instances in Auto Scaling groups by specifying a Spot Instance bid price in your launch configuration.
Release: Auto Scaling on 2012-06-07	1/1/2011	This release of Auto Scaling introduces support for launching EC2 instances with an AWS Identity and Access Management (IAM) role.
Release: Auto Scaling on 2012-06-11	1/1/2011	This release of Auto Scaling supports configuring an instance termination policy for your Auto Scaling Group.
Release: Auto Scaling on 2012-09-17	1/1/2011	Auto Scaling Group.
Release: Auto Scaling on 2012-11-16	1/1/2011	This release of Auto Scaling supports launching Amazon Elastic Compute Cloud (Amazon EC2) instance types as Amazon Elastic Block Store (EBS)-optimized instances.
Release: Auto Scaling on 2012-12-13	1/1/2011	This release of Auto Scaling supports Amazon Elastic Compute Cloud (Amazon EC2) instance status checks.
Release: Auto Scaling on 2013-12-10	1/1/2011	This release of Auto Scaling supports accessing Auto Scaling using the AWS Management Console.
Release: Auto Scaling on 2014-01-02	1/1/2011	This release of Auto Scaling adds support for creating launch configuration or Auto Scaling group using an Amazon EC2 instance

Πίνακας Π.6.4: Το σύνολο των εκδόσεων για την υπηρεσία Simple Queue Service

Version Release	WSDL Version Detail Description
Release: Amazon SQS on 2014-01-29	5/11/2012 Amazon SQS (Amazon SQS) adds support for dead letter queues.
Release: Amazon SQS on 2013-10-30	Amazon SQS (Amazon SQS) adds support for cross-origin resource sharing (CORS) with the AWS SDK for JavaScript.
Release: Amazon SQS on 2013-06-18	Amazon SQS adds support for 256 KB payloads in SQS. Previously, SQS messages were capped at 64 KB.
Release: Amazon SQS on 2012-11-21	You can now subscribe an Amazon SQS queue to an Amazon SNS topic using the AWS Management Console for Amazon SQS.
Release: Amazon SQS on 2012-11-05	Amazon SQS adds long polling, client-side buffering, and request batching, and also adds support for signature version 4.
Release: Amazon SQS on 2011-10-20	1/10/2011 Amazon SQS adds Delay Queues, Message Timers, GetQueueUrl, and batch API actions.
Release: Amazon SQS on 2011-10-06	1/2/2009 Amazon SQS is now available in the AWS Management Console
Release: Amazon SQS on 2010-09-28	1/2/2009 New AWS SDK for PHP.
Release: Amazon SQS on 2010-09-02	1/2/2009 Added support for AWS Identity and Access Management.
Release: Amazon SQS on 2010-06-30	1/2/2009 Added two new queue attributes.
Release: Amazon SQS on 2010-04-28	1/2/2009 Added support for new Region.
Release: Amazon SQS on 2010-03-22	1/2/2009 New AWS SDK for Java.
Release: Amazon SQS on 2009-12-02	1/2/2009 Added new endpoint.
Release: Amazon SQS on 2009-11-11	1/2/2009 Added new AWS SDK for .NET library.
Release: Amazon SQS on 2009-09-18	Removed the API reference from the Developer Guide and created a separate API Reference document.
Release: Amazon SQS on 2009-09-10	Added shared queues and anonymous access on 2009-04-07, and added new valid values for parameters in two operations on 2009-09-10.
Release: Amazon SQS on 2008-12-17	1/1/2008 New signature version and recommendations for secure requests.
Release: Amazon SQS on 2008-02-06	New price for Amazon SQS and changes to the service functionality.
Release: Amazon SQS on 2007-06-11	1/5/2007 Bug fixes for GetVisibilityTimeout, SetVisibilityTimeout, AddGrant, and ListGrants.
Release: Amazon SQS on 2007-05-18	Queue message counts, changeable message visibility timeouts, delimiters in queue names, and support for large messages through the Query API.
Release: Amazon SQS on 2006-07-11	1/4/2006 Amazon SQS (Amazon SQS™) is now generally available.

Πίνακας Π.6.5: Το σύνολο των εκδόσεων για την υπηρεσία Amazon Mechanical Turk

Release: AMT on 2011-12-01	1/10/2011	This release adds new operations that support Mechanical Turk Review Policies including ScoreMyKnownAnswers and SimplePlurality.
Release: AMT on 2008-11-04	2/8/2008	This release removes two operations: GetWorkerAcceptLimit and SetWorkerAcceptLimit, and adds a new qualification, Worker_NumberHITSApproved, which specifies the number of approved HITs a Worker submitted.
Release: AMT on 2008-05-09	1/4/2008	With this release, we have added a new API to our Web Service that enables Requesters to remove their Qualification Types.
Release: AMT on 2008-03-17	14/2/2008	To better serve US based workers, payment for completed HITs will now be deposited into an Amazon Payments Personal account.
Release: AMT on 2008-03-03	14/2/2008	This release of Amazon Mechanical Turk includes improvements to the ExternalQuestion functionality and the NotifyWorkers API.
Release: AMT on 2007-12-18	21/6/2007	This release of Amazon Mechanical Turk includes two major new pieces of Requester web site functionality that enhances your payment options and transaction history.
Release: AMT on 2007-09-04	21/6/2007	This release of Amazon Mechanical Turk includes several enhancements to the QuestionForm schema to allow Requesters more flexibility in designing their HITs.
Release: AMT on 2007-07-20	21/6/2007	This includes a new version of the API, known as "2007-06-21" that allows Requesters to block/unblock workers from completing the Requester's HITs .
Release: AMT on 2007-04-30	12/3/2007	This release of the Amazon Mechanical Turk includes a new developer sandbox that mirrors the production system. This new feature allows you to test your Mechanical Turk applications in a controlled and free environment.
Release: AMT on 2007-03-20	12/3/2007	This release of the Amazon Mechanical Turk includes a new version of the API, known as "2007-03-12," with a new feature that lets you configure the number of assignments a Worker can accept for your HITs.
Release: AMT on 2006-12-01	31/10/2006	This release of the Amazon Mechanical Turk includes a new version of the API, known as "2006-10-31," with new features such as the ability to assign a Qualification to a Worker without a Qualification request.
Release: AMT on 2006-10-20	27/9/2006	This release of the Amazon Mechanical Turk includes a new operation for querying your bonus payments, and a minor change to the Assignment data structure.
Release: AMT on 2006-09-20	23/8/2006	This release of the Amazon Mechanical Turk includes a new version of the API, known as "2006-08-23" with several new features.
Release: AMT on 2006-08-07	20/6/2006	This release of Amazon Mechanical Turk introduces two new features for creating HITs: XHTML formatted content, and questions hosted on external web sites.

Release: AMT on 2006-06-30	20/6/2006	This release of Amazon Mechanical Turk includes a new API, version 2006-06-20, with several new features, including bonus payments and private HITs.
Release: AMT on 2006-06-14	5/5/2006	This release of the Amazon Mechanical Turk includes a new version of the API, 2006-05-05, with several new features, including HIT event notifications, multimedia content in HITs, and locale-based Qualification requirements.
Release: AMT on 2006-04-27	20/3/2006	A new API version introduces a number of new features.
Release: AMT on 2006-03-03	16/1/2006	A new API version that includes HIT types, more flexibility in reviewing HITs, and updates to HIT Qualifications.
Release: AMT on 2005-10-13	1/10/2005	Introducing Amazon Mechanical Turk.

Πίνακας Π.6.6: Το σύνολο των εκδόσεων για την υπηρεσία Amazon Relational Database

<u>Release: Amazon RDS</u>	16/10/2009	This is the initial release of the Amazon RDS.
<u>Release: Amazon RDS on 2010-03-22</u>	16/10/2009	New AWS SDK for Java.
<u>Release: Amazon RDS on 2010-04-28</u>	16/10/2009	Adds support for the Asia Pacific (Singapore) region.
<u>Release: Amazon RDS on 2010-05-06</u>	16/10/2009	Added support for the US-West (Northern California) Region.
<u>Release: Amazon RDS on 2010-05-17</u>	1/1/2010	Adds support for the Multi-AZ Deployments; API updated to version 2010-01-01.
<u>Release: Amazon RDS on 2010-05-24</u>	1/1/2010	Adds AWS Console support.
<u>Release: Amazon RDS on 2010-08-16</u>	28/6/2010	Adds support for Reserved DB Instances and DB Engine Version Management.
<u>Release: Amazon RDS on 2010-09-02</u>	28/6/2010	Adds support for AWS Identity and Access Management (IAM).
<u>Release: Amazon RDS on 2010-09-02</u>	28/6/2010	Adds support for AWS Identity and Access Management (IAM).
<u>Release: Amazon RDS on 2010-10-04</u>	28/7/2010	Adds support for Read Replicas.
<u>Release: Amazon RDS on 2010-12-22</u>	28/7/2010	This release introduces AWS Management Console support for Reserved DB Instances for Amazon RDS (RDS).
<u>Release: Amazon RDS on 2011-05-23</u>	1/4/2011	Adds support for the Oracle database engine and updates the API version to 2011-04-01.
<u>Release: Amazon RDS on 2011-01-09</u>	19/12/2011	This release of Amazon RDS introduces two new types of Reserved DB Instance offerings.
<u>Release: Amazon RDS on 2012-01-24</u>	15/1/2012	This release of Amazon RDS adds support for running Amazon RDS DB Instances in an Amazon Virtual Private Cloud (VPC).
<u>Release: Amazon RDS on 2012-05-02</u>	23/4/2012	This release adds forced failover features for Oracle and MySQL as well as additional features for Oracle.
<u>Release: Amazon RDS on 2012-05-08</u>	23/4/2012	This release adds support for Microsoft SQL Server.
<u>Release: Amazon RDS on 2012-05-17</u>	23/4/2012	This release adds support for read replicas for MySQL DB instances in Amazon Virtual Private Cloud.
<u>Release: Amazon RDS on 2012-05-29</u>	23/4/2012	This release of the Amazon RDS adds support for option groups. The option to be supported with this release is Oracle Enterprise Management Database Control.
<u>Release: Amazon RDS on 2012-06-11</u>	23/4/2012	This release of Amazon RDS (Amazon RDS) introduces MySQL support for db.t1.micro DB instances.
<u>Release: Amazon RDS on 2012-07-11</u>	23/4/2012	This release of Amazon RDS (Amazon RDS) introduces Microsoft SQL Server support for the Database Engine Tuning Advisor.

Release: Amazon RDS on 2012-08-21	23/4/2012	This release of Amazon RDS (Amazon RDS) introduces support for DB Instance tagging.
Release: Amazon RDS on 2012-08-22	23/4/2012	This release of Amazon RDS (Amazon RDS) introduces support for SQL Server Agent. This release of Amazon RDS (Amazon RDS) introduces a new API version (2012-07-31) to support DB Instance tagging.
Release: Amazon RDS on 2012-08-23	31/7/2012	
Release: Amazon RDS on 2012-09-25	17/9/2012	This release of Amazon RDS (Amazon RDS) introduces support for provisioned IOPS.
Release: Amazon RDS on 2012-09-26	17/9/2012	This release of Amazon RDS (Amazon RDS) introduces support for RDS SQL Server 2012 DB Instances.
Release: Amazon RDS on 2012-09-27	17/9/2012	This release of Amazon RDS (Amazon RDS) introduces support for RDS Oracle micro DB Instances.
Release: Amazon RDS on 2012-10-10	17/9/2012	This release of Amazon RDS (Amazon RDS) introduces support for SSL in a SQL Server DB Instance.
Release: Amazon RDS on 2012-10-11	17/9/2012	This release of Amazon RDS (Amazon RDS) introduces support for promoting a Read Replica to a standard DB Instance.
Release: Amazon RDS on 2012-11-06	17/9/2012	This release of Amazon RDS (Amazon RDS) introduces support for m1.medium and m1.xlarge instance classes for all DB engines.
Release: Amazon RDS on 2013-01-14	10/1/2013	This release of Amazon RDS (Amazon RDS) introduces support for DB instance renaming and for DB security group migration.
Release: Amazon RDS on 2013-02-04	10/1/2013	This release of Amazon RDS (Amazon RDS) introduces support for DB event notification subscriptions.
Release: Amazon RDS on 2013-03-04	12/2/2013	This release of Amazon RDS (Amazon RDS) introduces support for log file access.
Release: Amazon RDS on 2013-03-13	15/5/2013	This release of Amazon RDS (Amazon RDS) introduces new functionality for Provisioned IOPS and adds support for upgrading a SQL Server 2008 R2 DB instance to a SQL Server 2012 DB instance.
Release: Amazon RDS on 2013-04-18	15/5/2013	This release of Amazon RDS (Amazon RDS) introduces support for Oracle Advanced Security features for native network encryption and transparent data encryption.
Release: Amazon RDS on 2013-05-23	15/5/2013	This release of Amazon RDS (Amazon RDS) introduces a new API version, 2013-05-15, and adds support for read replica status.

Release: Amazon RDS on 2013-06-20	This release of Amazon RDS (Amazon RDS) introduces support for upgrading your MySQL DB instances from version 5.1 to 5.5.
Release: Amazon RDS on 2013-07-01	This release of Amazon RDS (Amazon RDS) introduces support for using MySQL 5.6 for new DB instances, and for the MySQL 5.6 memcached interface and binary log access.
Release: Amazon RDS on 2013-08-28	This release of the Amazon RDS introduces support for using read replicas as a replication source.
Release: Amazon RDS on 2013-09-04	This release of Amazon RDS (Amazon RDS) introduces support for the db.cr1.8xlarge DB instance class for MySQL 5.6.
Release: Amazon RDS on 2013-09-05	This release of the Amazon RDS introduces support for using replication to import and export data between Amazon RDS and instances of MySQL running on-premises or in Amazon EC2.
Release: Amazon RDS on 2013-09-26	This release of Amazon RDS (Amazon RDS) introduces support for Oracle Statspack
Release: Amazon RDS on 2013-10-31	This release of the Amazon RDS (RDS) introduces support for copying DB snapshots from one AWS region to another. The release also includes a new Amazon RDS API version.
Release: Amazon RDS on 2013-11-07	This release of Amazon RDS (Amazon RDS) adds support for Microsoft SQL Server TDE.
Release: Amazon RDS on 2013-11-14	This release of Amazon RDS (Amazon RDS) introduces support for the PostgreSQL DB engine.
Release: Amazon RDS on 2013-11-26	This release of the Amazon RDS (RDS) introduces support for replicating between Amazon RDS MySQL DB instances in different AWS regions.
Release: Amazon RDS on 2014-01-13	This release of Amazon RDS (Amazon RDS) introduces support for Oracle Timezone option

ΣΥΝΤΟΜΟ ΒΙΟΓΡΑΦΙΚΟ

Ο Ιωάννης Ντίνος γεννήθηκε στα Ιωάννινα το 1972. Αποφοίτησε το 1989 από το Γενικό Λύκειο Ζωσιμαίας Σχολής. Οι βασικές σπουδές πραγματοποιήθηκαν στο Τμήμα πληροφορικής του Ελληνικού Ανοικτού Πανεπιστημίου, όπου εισήχθη το 2001 και αποφοίτησε το 2006. Συνέχισε για Μεταπτυχιακές Σπουδές στο Τμήμα πληροφορικής του Πανεπιστημίου Ιωαννίνων και εξειδικεύτηκε στο Λογισμικό. Τα ενδιαφέροντά του περιλαμβάνουν την τεχνολογία λογισμικού και τις τεχνολογίες διαδικτύου.



