

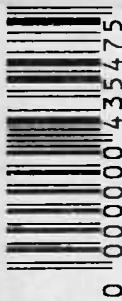


ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΒΙΒΛΙΟΘΗΚΗ
68661
005.711
10Y

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΔΙΠΛΩΜΑ ΕΙΔΙΚΕΥΣΗΣ (MSc)
στα ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΚΑΤΑΛΟΓΟΣ

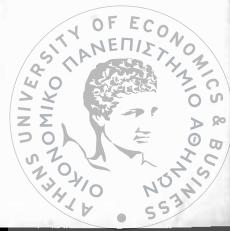


ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

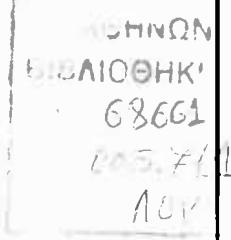
«Μοντελοποίηση Ανάκτησης Δέλτα»

Λουκής Λευτέρης
M3990017

ΑΘΗΝΑ, ΔΕΚΕΜΒΡΙΟΣ 2000



**ΜΕΤΑΠΤΥΧΙΑΚΟ ΔΙΠΛΩΜΑ ΕΙΔΙΚΕΥΣΗΣ (MSc)
στα ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ**



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Μοντελοποίηση Ανάκτησης Δέλτα»

**Λουκής Λευτέρης
M3990017**

Επιβλέπων Καθηγητής: Ε. Γιακουμάκης
Εξωτερικός Κριτής: Ν. Μαλεύρης

**ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

ΑΘΗΝΑ, ΔΕΚΕΜΒΡΙΟΣ 2000



Περιεχομένα

*Μοντελοποίηση
Ανάκτησης
Δέλτα*



Μοντελοποίηση Ανάκτησης Δέλτα

Πίνακας Περιεχομένων

1. ΠΡΟΛΟΓΟΣ	6
2. ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΚΑΙ ΟΡΙΣΜΟΙ	8
2.1. Πρόλογος	8
2.2. Γενική αναφορά στη θεωρία συστημάτων	8
2.3. Κυβερνητική – Αυτοματισμός και Ρομποτική - Έξυπνοι πράκτορες	10
2.4. Χρόνος – Χρονικές Βάσεις δεδομένων – Η έννοια του ΔΤ	11
2.5. Ορολογία	13
3. ΤΟΠΟΘΕΤΗΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ	16
3.1. Πρόλογος	16
3.2. Η ανάγκη ύπαρξης μοντέλου	18
3.3. Σενάρια χρήσης του μοντέλου	19
3.4. Απαιτήσεις μοντέλου – Σκοπός και στόχοι	21
4. ΑΝΑΛΥΣΗ ΜΟΝΤΕΛΟΥ	25
4.1. Πρόλογος	25
4.2. Η έννοια του συστήματος	25
4.3. Η έννοια της αλλαγής	29
4.4. Δύο βασικές Παρατηρήσεις	34
4.5. Το μοντέλο της κινηματογραφικής ταινίας	36
5. ΤΟ ΜΟΝΤΕΛΟ ΑΝΑΚΤΗΣΗΣ ΔΕΛΤΑ	39
5.1. Πρόλογος	39
5.2. Η παρατήρηση του συστήματος	40
5.3. Η καταγραφή των αλλαγών	44
5.4. Η διαχείριση των καταγραφών	53
5.5. Η ανάκτηση των καταγραφών	54
5.6. Η επαναφορά του συστήματος	56
6. ΕΦΑΡΜΟΓΗ ΜΟΝΤΕΛΟΥ ΑΝΑΚΤΗΣΗΣ ΔΕΛΤΑ	58
6.1. Πρόλογος	58
6.2. Εφαρμογή σε XML – Document Type Definition	58
7. ΑΝΑΦΟΡΕΣ	63
8. ΠΑΡΑΡΤΗΜΑΤΑ.....	66
8.1. Παράρτημα Α	66
8.2. Παράρτημα Β.....	69
9. ΠΕΡΙΛΗΨΗ	71
10. EXECUTIVE SUMMARY	73

1. Πρόλογος

Το θέμα της παρούσας εργασίας ασχολείται με τις αλλαγές που συμβαίνουν σε ένα σύστημα. Η ιδέα ξεκίνησε από την ανάγκη της αυτοματοποιημένης και τυποποιημένης ενημέρωσης ενός Χρήστη του Διαδικτύου, για τις αλλαγές που συμβαίνουν στους δικτυακούς Χώρους, με σκοπό την απλοποίηση της περιήγησής του στους δαιδαλώδεις ηλεκτρονικούς διαδρόμους του *Παγκόσμιου Ιστού Υπολογιστών*.

Η ανάγκη αυτή οδήγησε στην επέκταση του ζητούμενου θέματος και έτσι στη γενίκευση του προβλήματος. Δηλαδή οδήγησε στην αναζήτηση της προτυποποίησης ενός «μηχανισμού-μοντέλου» που θα μπορεί να παρατηρεί, να αντιλαμβάνεται, να καταγράφει και να αναπαριστά τις διαφοροποιήσεις στην κατάσταση ενός συστήματος κατά τη δράση στο χρόνο.

Στην παρούσα εργασία, επιχειρούμε την παρουσίαση ενός τέτοιου μοντέλου που να είναι ικανό να περιγράφει και να καταγράφει με τυπικό τρόπο τις αλλαγές που συμβαίνουν σε ένα σύστημα σε διάφορα επίπεδα λεπτομέρειας, έτσι ώστε να είναι δυνατή η ανάκτηση αυτών των αλλαγών είτε σε μορφή στιγμιότυπου (δηλαδή την κατάσταση του συστήματος σε μία συγκεκριμένη χρονική στιγμή), ή σε μορφή διαφορών ανάμεσα σε δύο στιγμιότυπα. Παράλληλα καταγράφονται και οι ενέργειες που «τυχόν» ευθύνονται για τις καταγραμμένες αλλαγές, για να μπορούμε να καλύψουμε έτσι όσο το δυνατόν περισσότερες περιπτώσεις μελέτης συστημάτων.

Παρουσιάζεται επομένως, μία σειρά από ορισμούς, προτάσεις και ιδέες για τον τρόπο σκέψης που θα πρέπει να χρησιμοποιήσουμε με σκοπό την ανάλυση, σχεδίαση και εφαρμογή ενός τέτοιου μηχανισμού καταγραφής και διαχείρισης των διαφορών που συμβαίνουν στην κατάσταση ενός συστήματος, στο πέρασμα του χρόνου, ή όπως αναφέρεται και στον τίτλο αυτής της εργασίας, η *Μοντελοποίηση της Ανάκτησης Δέλτα*.

Επιπλέον, παρουσιάζεται μία ψευδογλώσσα - σημειογραφία η οποία μπορεί να χρησιμοποιηθεί ως πρότυπο για τις διάφορες υλοποιήσεις που μπορούν να γίνουν με βάση αυτό το μοντέλο.

Το περιεχόμενο της εργασίας κατανέμεται ως εξής: Στην αρχική ενότητα παρουσιάζονται κάποιες βασικές έννοιες και ορισμοί που θα είναι χρήσιμοι για την κατανόηση του υπόλοιπου κειμένου και για την απόκτηση μιας σφαιρικής άποψης για την τεχνολογία που θα αναλυθεί στις επόμενες ενότητες. Αμέσως μετά γίνεται η τοποθέτηση του προβλήματος, η εξήγηση για την ανάγκη και τα σενάρια χρήσης για το μοντέλο ανάκτησης δέλτα, καθώς και οι απαιτήσεις που θα πρέπει να έχουμε από ένα τέτοιο μοντέλο. Στην επόμενη ενότητα αναλύουμε και ορίζουμε τις βασικότερες έννοιες που θα είναι χρήσιμες για την παρουσίαση του μοντέλου με σκοπό τη λεπτομερέστερη προσέγγιση στο πρόβλημά μας. Ακολουθεί η περιγραφή του ίδιου του μοντέλου και της ψευδογλώσσας – σημειογραφίας που χρησιμοποιεί ο μηχανισμός αυτός και, τέλος, στην επόμενη ενότητα, δίνεται μια εφαρμογή της σημειογραφίας σε γλώσσα XML (DTD), η οποία αποτελεί και έναν από τους στόχους αυτής της εργασίας. Τέλος, στα παραπόμπατα υπάρχει ολοκληρωμένη η σημειογραφία και το σύνολο εντολών διαχείρισης αυτής.

Κεφαλαιο 2

*Βασικές έννοιες και
ορισμοί*



2. Βασικές έννοιες και ορισμοί

2.1. Πρόλογος

Σ' αυτό το κεφάλαιο γίνεται μια εισαγωγή σε ορισμένες βασικές έννοιες, οι οποίες αναλύονται στις επόμενες ενότητες. Χωρίζεται στα επιμέρους θέματα της θεωρίας συστημάτων, σε θέματα αυτοματοποίησης, κυβερνητικής και έξυπνων πρακτόρων και τέλος σε θέματα χρόνου και χρονικών βάσεων δεδομένων. Επιπλέον, υπάρχουν οι ορισμοί για κάποιες έννοιες και τεχνολογίες που αναφέρονται στα επόμενα κεφαλαία και είναι χρήσιμες για την κατανόησή τους.

2.2. Γενική αναφορά στη θεωρία συστημάτων

Για την έννοια σύστημα υπάρχουν μία σειρά από ορισμούς και θεωρήσεις καθώς η καθημερινή χρήση αυτής της λέξης για την περιγραφή αντικειμένων, λειτουργιών, αφηρημένων έννοιών και καταστάσεων έχει διαστρεβλώσει την αντίληψή μας για τον αυστηρό προσδιορισμό του. Ίσως, τελικά, ένα από τα γενικά χαρακτηριστικά του συστήματος να είναι αυτή η απροσδιοριστία, που αφήνει περιθώρια για ευρεία εφαρμογή του μέσα στο Επιστητό.

Στην προσπάθειά του να μελετήσει αυτά που συμβαίνουν γύρω του, ο άνθρωπος χρησιμοποίησε διάφορους τρόπους σκέψης. Έτσι, μέχρι τον 17 μΧ αιώνα, κυριαρχούν οι ιδέες της μηχανοκρατίας, σύμφωνα με την οποία, ο κόσμος είναι συγκροτημένος με τις αρχές της μαθηματικής σκέψης. Η μηχανιστική σκέψη βασίζεται στην ανάλυση σε επιμέρους τμήματα μιας ύστατης μηχανής: της ίδιας της φύσης. Μέσα από αυτήν την ανάλυση και την επαγωγή, όλα τα επιμέρους στοιχεία είναι αυστηρά καθορισμένα και συμπεριφέρονται ως ένας αξιόπιστος μηχανισμός, δηλαδή ως ολότητα. Αυτή είναι και η έννοια του αναγωνισμού. Έτσι, για κάθε φαινόμενο υπάρχει μία αντιστοιχία «αιτία – αιτιατό» σύμφωνα με την οποία μπορούν να ερμηνευτούν όλα τα φαινόμενα. Η μηχανιστική σκέψη έχει τις ρίζες της στον Πλάτωνα και στον Πυθαγόρα. Ωστόσο, παρόλο που η μηχανιστική σκέψη μπορεί να δίνει ικανοποιητικά αποτελέσματα στην έρευνα πολλών φυσικών φαινομένων, παρουσιάζει δυσκολίες σε περιπτώσεις έμβιων οργανισμών ή σε πολιτικοκοινωνικές επιστήμες και επιστήμες συμπεριφοράς. Έτσι, μέσα στη δεκαετία του 40 προτάθηκε η συστημική προσέγγιση που έχει τις ρίζες της στις θεωρίες του Αριστοτέλη. Σύμφωνα με αυτή, το Όλον είναι κάτιο περισσότερο από το άθροισμα των επιμέρους στοιχείων του. Σε αυτήν τη θεώρηση λαμβάνονται υπόψη και οι αλληλεπιδράσεις των επιμέρους στοιχείων, ωστόσο τα ίδια τα επιμέρους στοιχεία δεν μπορούν να θεωρηθούν ανεξάρτητα. Έτσι, για να αναλυθεί το Όλον σε επιμέρους τμήματα θα πρέπει να ληφθούν υπόψη οι δεσμοί συναλλαγής ή σχέσης είτε αυτοί είναι στατικοί είτε είναι δυναμικοί. [1]

Ο όρος σύστημα αναπτύχθηκε κατά τη μελέτη βιολογικών οργανισμών από τον Ludwig Von Bertalanffy. Ένας γενικός ορισμός που δόθηκε από τους Schoderbek et al. είναι ο εξής [7] :

Ως σύστημα ορίζεται ένα σύνολο από αντικείμενα μαζί με τις σχέσεις μεταξύ των αντικειμένων και των χαρακτηριστικών γνωρισμάτων τους, τα οποία είναι σε αλληλοσυσχέτιση μεταξύ τους και με το περιβάλλον έτσι ώστε να αποτελούν μία ενιαία ολότητα.



Σε κάθε επιστήμη, η έννοια σύστημα λαμβάνει και διαφορετική όψη στον τρόπο αντίληψής της.

Για τις φυσικές επιστήμες, σύστημα είναι δύο ή περισσότερα σώματα που εξετάζονται ως ένα ενιαίο οργανικό σύνολο. Υπάρχουν εκτός από τα ίδια τα σώματα και οι αλληλεπιδράσεις μεταξύ τους. Για παράδειγμα αν θεωρήσουμε ως σύστημα δύο σφαίρες πάνω σε ένα επίπεδο που συγκρούονται τότε μπορούμε να μιλάμε για κινητική και δυναμική ενέργεια. Υπάρχουν εξωτερικές και εσωτερικές δυνάμεις, ανάλογα με το αν αυτές προέρχονται από το περιβάλλον ή από το ίδιο το σύστημα. Μπορούμε να μελετάμε τα αποτελέσματα αυτών των δυνάμεων και των ενεργειών, να περιγράφουμε με μαθηματικές εξισώσεις τις αλληλεπιδράσεις των σωμάτων που ανήκουν σε ένα σύστημα και να χρησιμοποιούμε μετρικές για τον υπολογισμό των τιμών των ιδιοτήτων κάθε σώματος (π.χ. ταχύτητα, απόσταση, ενέργεια κ.α.).

Στην επιστήμη των Υπολογιστών, μπορούμε να μιλάμε για λειτουργικά συστήματα και συστήματα αρχείων. Εδώ, τα επιμέρους στοιχεία του συστήματος, οι λειτουργίες και οι σχέσεις τους, καθώς και τα όρια του περιβάλλοντος είναι καλά καθορισμένα.

Για την επιστήμη της Πληροφορικής, ο όρος Πληροφοριακό σύστημα, με τη σημερινή του σημασία, αποτελείται από 5 επιμέρους τμήματα τα οποία αλληλεπιδρούν μεταξύ τους: **Τους Ανθρώπους, τις Μηχανές, τα Δεδομένα, το Λογισμικό, και τις Διαδικασίες**. Το Π. Σ. έχει 3 διαστάσεις – όψεις, ανάλογα με το εκάστοτε πρόβλημα ανάλυσης που καλούμαστε να επιλύσουμε: μία *Τεχνική*, μία *Διοικητική* και μία *Κοινωνική* [1]. Ένα Πληροφοριακό σύστημα είναι πολύ σημαντικό πλέον στη σύγχρονη εποχή και αποτελεί αναπόσπαστο μέρος ενός Οργανισμού. Η σημαντικότητα αυτή ενός Π. Σ. καθιστά απαραίτητα, εκτός από την προσεκτική ανάπτυξη του, τη σαφώς ορισμένη λειτουργία του, τις καθορισμένες απαιτήσεις του, και το δραστικό έλεγχό του.

Η έννοια σύστημα έχει επίσης ταυτιστεί και με τη μεθοδολογία, (π.χ. σύστημα επίλυσης εξισώσεων, σύστημα μελέτης κλπ). Υπάρχουν πολλοί άλλοι ορισμοί συστημάτων σε κάθε ένα κλάδο των επιστήμων, θετικών ή μη, ωστόσο εμάς μας ενδιαφέρουν οι ορισμοί των συστημάτων που εμπειρικείουν πληροφορία προς αποθήκευση.

Τα συστήματα μπορούμε να τα ταξινομήσουμε σε κατηγορίες, χρησιμοποιώντας διάφορα κριτήρια. [1]

1. Έτσι, σε σχέση με τη φύση των επιμέρους στοιχείων του συστήματος, μπορούμε να τα χωρίσουμε σε **φυσικά**, όταν αυτά υπάρχουν αυτούσια στη φύση, και **σχεδιασμένα** όταν είναι τεχνητά. Παραδείγματα φυσικών συστημάτων είναι ο άνθρωπος ή ένα πλανητικό σύστημα, ενώ παράδειγμα τεχνητών συστημάτων είναι ένα πληροφοριακό σύστημα, ένα κοινωνικό σύστημα κλπ.
2. Σε σχέση με το περιβάλλον μπορούμε να τα χωρίσουμε σε **ανοικτά**, όταν υπάρχουν εισερχόμενα και εξερχόμενα από το περιβάλλον και **κλειστά** όταν το σύστημα θεωρείται απομονωμένο. Παράδειγμα ανοικτού συστήματος είναι επίσης ο άνθρωπος, ενώ παράδειγμα κλειστού συστήματος είναι η χημική αντίδραση σε ένα κλειστό δοχείο.
3. Σε σχέση με την συμπεριφορά των συστημάτων, μπορούμε να τα χωρίσουμε σε **σταθερής κατάστασης, συγκεκριμένου στόχου, πολλών στόχων και τέλος συστήματα προθέσεως**. Έτσι, αντίστοιχα, ένας θερμοστάτης έχει πάντα ίδια συμπεριφορά, ένα σύστημα αυτόματης πλοϊγησης έχει πάντα ίδιο στόχο, ένα

υπολογιστικό σύστημα έχει περισσότερους από έναν στόχους, ενώ ένας άνθρωπος μπορεί να έχει πολλές προθέσεις.

4. Σε σχέση με τα γεγονότα που συμβαίνουν, ένα σύστημα μπορεί να χαρακτηριστεί ως **στατικό**, **δυναμικό** και **ομοιοστατικό**. Στην πρώτη περίπτωση δεν συμβαίνουν γεγονότα, άρα η κατάσταση του είναι σταθερή και στη δεύτερη περίπτωση η κατάστασή του συστήματος αλλάζει με το χρόνο. Στην τρίτη περίπτωση τα συστήματα θεωρούνται στατικά ως ολότητες, ωστόσο τα επιμέρους στοιχεία του αλλάζουν δυναμικά. Δηλαδή είναι αυτά τα συστήματα που έχουν σταθερή κατάσταση σε σχέση με το περιβάλλον.
5. Τέλος, σε σχέση με το βαθμό πρόβλεψης της συμπεριφοράς, μπορούμε να χωρίσουμε τα συστήματα σε **αιτιοκρατικά** και σε **πιθανολογικά**.

Σε επόμενες ενότητες θα αναλύσουμε τον όρο σύστημα με αναλυτικές περιγραφές των επιμέρους όρων που αποτελούν το ορισμό του και θα προσπαθήσουμε να γενικεύσουμε αυτό τον ορισμό έτσι ώστε να βρίσκει εφαρμογή σε όσο το δυνατό περισσότερες θεωρήσεις συστημάτων που υπάρχουν μέχρι σήμερα.

2.3. Κυβερνητική – Αυτοματισμός και Ρομποτική - Εξυπνοι πράκτορες

Με την έννοια *Κυβερνητική* εννοούμε την επιστήμη της μελέτης ενός αυτορυθμιζόμενου συστήματος. Έχει τις ρίζες της στην εποχή του Πλάτωνα (400 π.Χ.), όπου ορίζεται η τέχνη της διακυβέρνησης πλοίων, αρμάτων, ανθρώπων και γενικότερα μηχανών. Ένα απλό παράδειγμα ενός αυτορυθμιζόμενου συστήματος είναι ο αυτόματος πιλότος του αεροσκάφους που συνεχώς ρυθμίζει και διορθώνει την πορεία του σύμφωνα πάντοτε με τον αρχικό προορισμό και σκοπό.[5]

Το πρώτο βιβλίο με τη σημερινή έννοια της λέξεως «κυβερνητική», το δημοσίευσε το 1948 ο N. Wiener με τίτλο «*Έλεγχος και επικοινωνία στο ζωντανό Οργανισμό και στη μηχανή*». Ο Wiener μελέτησε τα συστήματα αυτόματου έλεγχου με σκοπό την αυτόματη σκόπευση των αντιαεροπορικών πυροβόλων εξαιτίας της ολοένα αυξανόμενης ταχύτητας των πολεμικών αεροσκαφών. Έτσι, γεννήθηκε η ανάγκη για σχεδιασμό αυτόματων μηχανισμών που θα μπορούσαν να αντικαταστήσουν αποτελεσματικά τον άνθρωπο.[4]

Η έννοια που σχετίζεται άμεσα με την επιστήμη της κυβερνητικής είναι η **ανάδραση**. Με τον όρο αυτό εννοούμε ότι οι εξερχόμενες πληροφορίες ή ένα μέρος από αυτές, κατά τη λειτουργία του συστήματος, χρησιμοποιούνται ως επιστροφή στο σημείο εισαγωγής ενός μηχανισμού ή μιας λειτουργίας. Η επαναπληροφόρηση αυτή οδηγεί στην ανάλογη συμπεριφορά του συστήματος με αποτέλεσμα να έχουμε τα επιθυμητά αποτελέσματα, δηλαδή την αυτορύθμιση. Το σύστημα επομένως με δάφορα αισθητήρια όργανα (*sensors*) επικοινωνεί και ανταλλάσσει πληροφορίες με το περιβάλλον του, έτσι ώστε να δοθούν κατάλληλες εντολές ή να πραγματοποιηθούν οι κατάλληλες ενέργειες για την επίτευξη του στόχου της λειτουργίας του συστήματος.

Η κυβερνητική δεν περιορίζεται μόνο στη θεώρηση της ανάδρασης, ως στιγμαία πληροφορία που επανατροφοδοτεί το σύστημα, αλλά και στην αποθήκευση πληροφοριών ώστε να μπορούν να γίνουν συνδυασμοί ενεργειών ή να ληφθούν ορισμένες αποφάσεις από το σύστημα, έτσι ώστε να επιλεχθεί η αποτελεσματικότερη,

κάθε φορά, δράση του μηχανισμού. Πάνω σε αυτό έχουν στηριχτεί τα συστήματα αυτόματου έλεγχου (αυτοματισμοί, ρομποτική), που αποτελούν κόμβους επεξεργασίας πληροφοριών και λήψης αποφάσεων σχετικά με τη ρύθμιση της παραγωγικής διαδικασίας [4]. Έτσι, ένα ρομπότ βασίζεται στην αποθήκευση γνώσης και στην τεχνητή μάθηση. Αυτά τα δύο στηρίζονται στην καταγραφή πληροφορίας που, στην περίπτωσή μας, είναι οι αλλαγές. Έτσι, για παράδειγμα, μπορούμε να καταγράψουμε μία σειρά από κινήσεις ενός τεχνητού βραχίονα και μετά να τις επαναλάβουμε όσες φορές θέλουμε πετυχαίνοντας πάντα το ίδιο αποτέλεσμα. Αυτό, σε συνδυασμό με το σύστημα απόφασης ενός αυτοματοποιημένου μηχανισμού αποτελεί και την έννοια των αυτόματων παραγωγής (ρομπότ).

Οι έξυπνοι πράκτορες αποτελούν ένα είδος αυτόματου συστήματος σύμφωνα με την παραπάνω ανάπτυξη. Ένας έξυπνος πράκτορας είναι ένα αυτοδύναμο τμήμα λογισμικού που είναι υπεύθυνο για την εκτέλεση μιας ορισμένης εργασίας σε κάποιο περιβάλλον. Λέγονται αυτόνομοι διότι είναι προσαρμοσμένοι σε ένα περιβάλλον και μπορούν να το «αισθανθούν», να το παρατηρήσουν και να ενεργήσουν πάνω σε αυτό. Στην πραγματικότητα, ένας πράκτορας μπορεί να είναι ένα πρόγραμμα που λειτουργεί μέσα σε ένα κατανεμημένο σύστημα και είναι προγραμματισμένο για μία ή περισσότερες συγκεκριμένες εργασίες. Για παράδειγμα μπορεί να έχει αναλάβει την εύρεση και συλλογή πληροφοριών για κάποιο θέμα μέσα από τις σελίδες του Διαδικτύου. Η έννοια του πράκτορα είναι πολύ σημαντική για την κατανόηση και για τη μελέτη αυτής της εργασίας.

2.4. Χρόνος – Χρονικές Βάσεις δεδομένων – Η έννοια του ΔΤ

Ο χρόνος είναι αφηρημένη έννοια. Ένα γεγονός είναι, ότι ο άνθρωπος δεν διαθέτει αισθητήρια όργανα για να αντιληφθεί το χρόνο. Στην πραγματικότητα, δεν μπορούμε να τον καταγράψουμε όσο και αν αυτό ακούγεται παράδοξο. Για να τον καταγράψουμε χρησιμοποιούμε φυσικά μέσα όπως την περιοδικότητα ενός σώματος (περιστροφή Γης), τις ιδιότητες που έχει ένα φυσικό φαινόμενο ή σώμα (ταχύτητα του Φωτός, παλμός Quartz) ή βασισμένοι σε μηχανισμούς και σε μαθηματικά (μηχανικό ρόλοι). Αυτό δηλαδή που καταγράφουμε είναι το αποτέλεσμα του χρόνου που, στην πραγματικότητα, είναι η αιτία, ή και το ανάποδο: καταγράφουμε δηλαδή την αιτία κάποιου φυσικού φαινομένου που, στην πραγματικότητα, είναι το αποτέλεσμα αυτού! Αυτά και αλλά παράδοξα όπως η χρονική ασυνέχεια δημιουργούνται σχετικά με τη μελέτη του χρόνου και την προσπάθεια ανάλυσής του. Σ' αυτό προστίθεται και το γεγονός ότι ο άνθρωπος δεν μπορεί να αντιληφθεί το άπειρο και το κενό. Χρησιμοποιεί έννοιες και σύμβολα γι' αυτό, και το λαμβάνει υπόψη του κατά μοντελοποίηση και τη μέτρηση του φυσικού κόσμου. Για παράδειγμα στα μαθηματικά χρησιμοποιείται κατά κόρον η έννοια του άπειρου, του κενού κλπ. Πραγματικά, αυτό βοηθάει να κατανοηθούν και να ορισθούν νέες έννοιες και σχέσεις οι οποίες είναι έτοιμες να καταρρεύσουν όταν κάποια στιγμή αλλάξουμε την άποψή μας για το χρόνο.

Η Γενική Θεωρία της Σχετικότητας, στις αρχές του προηγούμενου αιώνα, έφερε επανάσταση στον τρόπο αντιμετώπισης της αντίληψης του Χώρου και του Χρόνου. Ο ίδιος ο Αϊνστάιν είπε ότι όταν μελετάμε το χρόνο θα πρέπει να τον θεωρούμε τοπικά [4]. Αυτό πλέον, με διάφορα πειράματα, έχει αποδειχτεί θριαμβευτικά. Έτσι, και εμείς, θα θεωρούμε το χρόνο τοπικά. Δηλαδή, για ένα κομμάτι του σύμπαντος που αποτελεί το περιβάλλον του υπό εξέταση συστήματος, θα χρησιμοποιούμε την

παραδοσιακή θεώρηση της συνέχειας του χρόνου για την περαιτέρω ανάπτυξη του θέματός μας.

Η μοντελοποίηση του χρόνου σε ένα σύστημα αποθήκευσης δεδομένων στα οποία εμπλέκεται ο χρόνος, πχ μία χρονική βάση δεδομένων, απαιτεί να έχουμε ορίσει επακριβώς το ΤΙ θεωρούμε ως χρόνο, χωρίς αυτό να σημαίνει ότι πρέπει να καλύψουμε όλες τις περιπτώσεις του πραγματικού κόσμου [3]. Η κοινότητα που ασχολείται με τις χρονικές εφαρμογές έχει χρησιμοποιήσει τρία μοντέλα για το χρόνο. Συνήθως ο χρόνος «χαρτογραφείται» με ένα σύνολο από αριθμούς που είναι πλήρως διατεταγμένοι και υπακούονται στο σύμβολο σύγκρισης (<). Έτσι, ο χρόνος μπορεί να αναπαρασταθεί με έναν άξονα, σε κάποιο σύστημα συντεταγμένων.

Τα μοντέλα που χρησιμοποιούμε για το χρόνο είναι τα εξής:[8]

1. **Συνεχές** : που χρησιμοποιεί το σύνολο των πραγματικών αριθμών για την αναπαράσταση του χρόνου. Με αυτό τον τρόπο, ανάμεσα σε δύο σημεία στο χρονικό άξονα, πάντα θα υπάρχει ένα άλλο σημείο που θα αντιστοιχεί σε έναν άλλο πραγματικό αριθμό.
2. **Πυκνό** : που είναι η αντιστοιχία του χρόνου στους ρητούς αριθμούς.
3. **Διακριτό**: που είναι η αντιστοιχία του χρόνου σε ακέραιους αριθμούς.

Από τα τρία αυτά μοντέλα, το *Συνεχές* είναι και το πιο ακριβές στην αναπαράσταση του χρόνου. Ωστόσο, η χρήση του χρόνου σε μία ψηφιακή (υπολογιστική) συσκευή καθιστά αδύνατη τη συνεχή αναπαράσταση του χρόνου, χωρίς απώλεια πληροφορίας. Επομένως από τη στιγμή που θα πρέπει να καταγράψουμε το χρόνο σε μία υπολογιστική συσκευή, θα θεωρούμε το χρόνο, απαραιτήτως, *Διακριτό*.[9]

Με βάση την τελευταία παρατήρηση, ορίζεται η έννοια του **Χρόνον**, ως το μικρότερο κομμάτι που μπορεί να αναπαρασταθεί πάνω στο χρονικό άξονα [10]. Έτσι, όταν θα μιλάμε για υπολογιστικές συσκευές οι οποίες μπορούν να παραστήσουν πραγματικούς αριθμούς μέχρι ένα ορισμένο δεκαδικό ψηφίο, θα εννοούμε αυτή τη διακριτή τιμή ανάμεσα σε δύο τέτοιους αριθμούς. Η έννοια που περιγράφεται σε επόμενες ενότητες ως dt θα είναι αυτή ακριβώς η διαφορά ανάμεσα σε δύο «διπλανούς» αριθμούς που παριστάνουν το χρόνο στην υπολογιστική μας συσκευή.

Επιπλέον, ένα **χρονικό διάστημα** θα ορίζεται ως ένα διάστημα πάνω στο χρονικό άξονα ανάμεσα σε δύο συγκεκριμένους αριθμούς [10] και θα μπορεί να χαρακτηρίζεται από ανοικτότητα και κλειστότητα με την αντιστοιχία των εννοιών, δανεισμένων από τα μαθηματικά.

Οι χρονικές βάσεις δεδομένων είναι, στην ουσία, βάσεις δεδομένων που σχετίζονται με την παράσταση χρονικών στοιχείων. Οι χρόνοι μιας χρονικής βάσης δεδομένων μπορούν να χωριστούν στις εξής κατηγορίες:[3]

1. **Χρόνοι Εγκυρότητας (Valid time)**: Είναι οι χρόνοι που αντιστοιχίζονται σε δεδομένα που έχουν σημασία στον πραγματικό χρόνο. Για παράδειγμα, μπορούμε να πούμε ότι είναι ο χρόνος που είχε μία ορισμένη τιμή, κάποια μετοχή, κατά τη διάρκεια μιας συνεδρίασης του χρηματιστηρίου.
2. **Χρόνοι Δοσοληψίας (Transaction time)**: Είναι οι χρόνοι που αντιστοιχίζονται στη χρονική στιγμή που έγινε μία καταγραφή ενός γεγονότος.

3. **Χρόνοι ορισμού Χρήστη (User Defined time):** Είναι οι χρόνοι που έχουν σημασία μόνο για το Χρήστη που καταγράφει ένα γεγονός. Για το σύστημα είναι ένα απλό δεδομένο.

Οι χρονικές βάσεις δεδομένων χωρίζονται σε τέσσερις κατηγορίες. [11]

1. **Βάσεις στιγμιότυπου (Snapshot).** Σε αυτές τις βάσεις αποθηκεύονται δεδομένα που σχετίζονται με την τρέχουσα κατάσταση ενός συστήματος. Παλαιότερες καταστάσεις δεν είναι δυνατό να αποθηκευτούν και επομένως όταν αλλάζει η τιμή σε ένα πεδίο τότε αυτό ανανεώνεται χωρίς να κρατείται ιστορικό για τις αλλαγές του.
2. **Ιστορικές βάσεις (Historical).** Αυτές οι βάσεις αποθηκεύουν το χρόνο με τη σημασία που αυτός έχει στον πραγματικό κόσμο. Δηλαδή κρατείται μόνο ο Χρόνος Εγκυρότητας χωρίς να είναι δυνατή η επαναφορά της βάσης σε παλαιότερες χρονικές στιγμές.
3. **Βάσεις επαναφοράς (Rollback).** Σε αυτές τις βάσεις αποθηκεύεται ο χρόνος δοσοληψίας, δηλαδή ο χρόνος των αλλαγών που συμβαίνουν στην ίδια τη βάση. Έτσι, είναι δυνατή η επαναφορά του συστήματος σε παλαιότερες χρονικές στιγμές.
4. **Διχρονικές βάσεις (Bitemporal).** Πρόκειται για το συνδυασμό μιας Ιστορικής και μιας Βάσης Επαναφοράς. Μία διχρονική βάση έχει τα πλεονεκτήματα όλων των άλλων βάσεων καθώς μπορεί να αποθηκεύσει και πληροφορίες ιστορικής φύσεως αλλά και πληροφορίες επαναφοράς του συστήματος.

Κατά μία έννοια, όλα τα συστήματα, με όλους τους ορισμούς και τις κατηγοριοποιήσεις που τους έχουν δοθεί, έχουν άμεση σχέση με το χρόνο. Αυτή η σχέση δεν είναι άλλη από την έννοια της αλλαγής που συμβαίνει σε ένα σύστημα. Εκεί θα εστιάσουμε και την προσοχή μας για την περαιτέρω τοποθέτηση και ανάλυση του προβλήματος. Μας ενδιαφέρουν, δηλαδή, τα συστήματα που επηρεάζονται από τον παράγοντα χρόνο. Όλη, λοιπόν, η μελέτη θεωρεί το χρόνο ως μετρήσιμο, πάντα με κάποιο σημείο αναφοράς όπως για παράδειγμα μία αρχική κατάσταση. Άλλα για τις καταστάσεις του συστήματος και τη σχέση τους με το χρόνο και τις αλλαγές, θα μιλήσουμε στις επόμενες παραγράφους αφού πρώτα δώσουμε τους ορισμούς για κάποιες έννοιες οι οποίες θα αναφέρονται ευρέως στη συνέχεια, και κάποιες λέξεις οι οποίες επίσης θα χρησιμοποιούνται για την περιγραφή του μοντέλου.

2.5. Ορολογία

Δεδομένα : (Data) Ο όρος δεδομένα, σύμφωνα με τα διεθνή πρότυπα ISO είναι μια παράσταση γεγονότων εννοιών ή εντολών σε τυποποιημένη μορφή που είναι κατάλληλη για επικοινωνία, ερμηνεία ή επεξεργασία από τον άνθρωπο ή από αυτόματα μέσα. [1]

Πληροφορία : (Information) Είναι η τρέχουσα σημασία που αποδίδεται στα δεδομένα, χρησιμοποιώντας τις συμβατικές παραδοχές που εφαρμόζονται σ' αυτά. Έτσι, θα έχουμε την εξίσωση : δεδομένα + ερμηνεία = πληροφορία.[1]

Γλώσσα markup : Ιστορικά η έννοια mark up χρησιμοποιήθηκε σε αρχεία κειμένου για να μπορούμε να δηλώνουμε μεταδεδομένα και άλλα χαρακτηριστικά, εντός του κειμένου. Για παράδειγμα, θα μπορούσαμε να αποθηκεύσουμε τη γραμματοσειρά κάποιας παραγράφου ή ένα σύνδεσμο αναφοράς σε άλλο έγγραφο κλπ. Έτσι, μια γλώσσα markup, είναι ένα σύνολο από τέτοιες συμβάσεις για την κωδικοποίηση κειμένων. Σε μια τέτοια γλώσσα πρέπει να καθορίζεται, ποια θα είναι τα σύμβολα και ποια θα είναι η σημασία αυτών, για ένα κείμενο. Η πρότυπη μεταγλώσσα SGML (Standard Generalized Markup Language), η οποία παρουσιάστηκε το 1986, είναι ακριβώς, ένας τρόπος για τη δημιουργία τέτοιων γλωσσών. (ISO 8879 [13]) Η XML αποτελεί ένα υποσύνολο της γλώσσας SGML, χρησιμοποιείται ως επί το πλείστον για εφαρμογές διαδικτύου και επιτρέπει τη δημιουργία markup γλωσσών (CDF, SMIL, CML κλπ) που χρησιμοποιούνται ευρέως από την κοινότητα της Πληροφορικής.[14] Η γλώσσα HTML που χρησιμοποιείται για την αναπαράσταση Χώρων στο Διαδίκτυο, είναι μια υλοποίηση της SGML.[15]

URN : (Uniform Resource Name). Είναι η ονομασία κάποιου πόρου (αρχείου, σελίδας, εικόνας κλπ) που υπάρχει στο Διαδίκτυο, η οποία έχει μόνιμη και μοναδική σημασία. Λέγοντας μόνιμη, εννοούμε ότι οποιοσδήποτε χρήστης του Διαδικτύου μπορεί, όποτε αυτός το θελήσει, να βρει τον πόρο στον οποίο αναφέρεται ένα URN. Το URN έχει παρόμοια συμβολογραφία με αυτή του γνωστού, στο Διαδίκτυο, URL (Uniform Resource Locator). Τα URN και URL είναι τύποι μιας γενικότερης έννοιας που αναπτύσσεται για τις ονομασίες σε κατανεμημένα περιβάλλοντα, του URI (Uniform Resource Identifier) από τον οργανισμό I.E.T.F. [12]

Δέλτα : Εκφράζει τη Διαφορά και συμβολίζεται με το ελληνικό γράμμα «Δ». Στην περίπτωσή μας θα αναφέρεται στις διαφοροποιήσεις της κατάστασης του συστήματος που μελετάμε κατά την διάρκεια των αλλαγών που συμβαίνουν σε αυτό. Η ανάκτηση δέλτα, αναλύεται και ορίζεται στις επόμενες ενότητες.

Χρονική βάση δεδομένων : (Temporal database) είναι μια βάση δεδομένων στην οποία εμπλέκεται, αποθηκευτικά αλλά και σημασιολογικά, ο παράγοντας χρόνος. Περισσότερες λεπτομέρειες υπάρχουν στο αντίστοιχο κεφάλαιο.

Έξυπνος πράκτορας : (Intelligent agent) είναι ένα αυτοδύναμο τμήμα λογισμικού που είναι υπεύθυνο για την εκτέλεση μιας ορισμένης εργασίας σε κάποιο περιβάλλον, συνήθως κατανεμημένο. Περισσότερες λεπτομέρειες υπάρχουν στο αντίστοιχο κεφάλαιο.

Version control : Είναι η τεχνολογία που επιτρέπει τον έλεγχο του Ιστορικού, των πόρων ενός έργου Λογισμικού. Λέγοντας πόρους, εννοούμε αρχεία τα οποία αλλάζουν με την εξέλιξη του έργου. Οι παλαιότερες εκδόσεις (version) των αρχείων αυτών αποθηκεύονται κατάλληλα σε φακέλους (archives) έτσι ώστε να είναι δυνατή η ανάκτησή τους.[6]

Κεφαλαιο 3

*Τοποθέτηση του
προβλήματος*

3. Τοποθέτηση του προβλήματος

3.1. Πρόλογος

Ας θεωρήσουμε μία συνηθισμένη προβληματική κατάσταση. Κάποιος Χρήστης του διαδικτύου, ανάλογα με τα ενδιαφέροντα που έχει, επισκέπτεται τακτικά αντίστοιχους Χώρους (Sites) στο διαδίκτυο, όπου κάθε φορά βρίσκει τις πληροφορίες που θέλει. Πολλοί από αυτούς τους Χώρους, συχνά αλλάζουν το περιεχόμενο των ιστοσελίδων τους. Άλλοι ανανεώνουν τις πληροφορίες τους σε τακτά χρονικά διαστήματα, ενώ άλλοι, κάθε φορά που έχουν να παρουσιάσουν κάτι καινούριο, απλώς το δηλώνουν σε κάποιο σημείο των σελίδων τους.

Για παράδειγμα ας θεωρήσουμε ένα ηλεκτρονικό κατάστημα που επισκέπτεται τακτικά ο εν λόγω Χρήστης. Εξ ορισμού, το ηλεκτρονικό κατάστημα θα διαθέτει κάθε φορά, ένα σύνολο από νέα προϊόντα που ενδεχομένως θα ενδιαφέρεται να προωθήσει. Το ίδιο το ηλεκτρονικό κατάστημα θα έχει νέες τιμές για τα προϊόντα, θα έχει προσφορές, δώρα, κοκ. Από τη μεριά του ο Χρήστης θέλει να ενημερωθεί για όλα αυτά, και ο μόνος τρόπος για να το καταφέρει αυτό είναι από τις αντίστοιχες «στατικές» σελίδες νέων (what's new), που πιθανόν να διαθέτει το ηλεκτρονικό κατάστημα. Φυσικά δεν θα του δίνεται η δυνατότητα να μάθει παλαιότερες αλλαγές που συνέβησαν στις σελίδες ή στα προϊόντα του ηλεκτρονικού καταστήματος όπως επίσης δεν θα είναι σε θέση να κάνει επιλογή από τα νέα, που του προσφέρονται μέσω αυτών των σελίδων. Τελικά δεν θα είναι σε θέση να παρακολουθήσει την εξέλιξη της αγοράς, με τον τρόπο που αυτός επιθυμεί.

Συνεχώς λοιπόν, και για διαφορετικούς λόγους κάθε φορά, υπάρχουν νέες πληροφορίες οι οποίες αλλάζουν με άγνωστους ρυθμούς, τις οποίες θα ενδιαφερόταν να μάθει ο εν λόγω Χρήστης του διαδικτύου. Το πρόβλημα είναι ότι αν οι Χώροι αυτοί είναι αρκετοί σε πλήθος, τότε η εργασία της «συγκομιδής» των νέων πληροφοριών από το Χρήστη θα είναι αρκετά επίπονη και χρονοβόρα...

Μέχρι τώρα, η εργασία της καταγραφής του ιστορικού των αλλαγών που συμβαίνουν σε έναν Χώρο Διαδικτύου, τις περισσότερες φορές, παραλείπεται. Σε άλλες περιπτώσεις γίνεται χειρωνακτικά και με διαφορετικό τρόπο για κάθε Χώρο. Ο κάθε διαχειριστής του Χώρου δημιουργεί μία ή περισσότερες σελίδες με τις αλλαγές που έχουν συμβεί σε χρονολογική σειρά αλλά τις καταγράφει με αυθαίρετο τρόπο, σε αυθαίρετη μορφή και με αμφισβητούμενη, πολλές φορές, εγκυρότητα. Το αποτέλεσμα είναι, ο Χρήστης που επισκέπτεται το χώρο αυτό, να βρίσκεται μπροστά σε ένα μεγάλη σελίδα με ακατέργαστο κείμενο που περιέχει καταγραφές αλλαγών, το οποίο πρέπει να διαβάσει και να επιλέξει αυτά που τον ενδιαφέρουν για να μπορέσει να έχει μία ιδέα του τι άλλαξε από την τελευταία φορά που επισκέφτηκε το Χώρο.

Έχουν γίνει κάποιες προσπάθειες για να λυθεί, επιμέρους, αυτό το πρόβλημα, όπως για παράδειγμα τα **Ενεργά Κανάλια** από τη Microsoft (Active Channels)[17]. Σύμφωνα με αυτήν την τεχνολογία, ο Χρήστης εγγράφεται στα κανάλια που ενδεχομένως να υποστηρίζει ο εξυπηρετητής του Χώρου. Έτσι, κάθε φορά που ο Χρήστης είναι συνδεμένος στο διαδίκτυο, γίνεται ένας προκαθορισμένος τακτικός έλεγχος στον εξυπηρετητή σχετικά με τη διαφοροποίηση των σελίδων από την τελευταία φορά που επισκέφτηκε το χώρο ο Χρήστης. Αν έχει αλλάξει μία σελίδα ή άλλο αρχείο από τον εξυπηρετητή τότε ο Χρήστης ειδοποιείται κατάλληλα. Ωστόσο αυτή την τεχνολογία, παρόλο που έχει ως γλώσσα υλοποίησης μία μορφή της

γλώσσας *XML* (για την οποία θα μιλήσουμε παρακάτω), δεν προσφέρει μεγάλη ευελιξία σε θέματα γενίκευσης της πληροφορίας. Είναι δηλαδή μόνο ένας μηχανισμός που προσφέρει ανανέωση στις ιστοσελίδες που έχει ήδη επισκεφτεί ο Χρήστης. Με δύο λόγια δεν απαντάει σε ερωτήματα τύπου «τι έχει αλλάξει από τη χρονική στιγμή t και μετά» ή «ποια ήταν η μορφή του Χώρου τη χρονική στιγμή t».

Μία άλλη προβληματική περίπτωση που μπορούμε να συναντήσουμε είναι αυτή των νέων εκδόσεων σε ένα έγγραφο ή σε μία επιστημονική εργασία. Σε περίπτωση που κάποιος ενδιαφέρεται να μάθει τις αλλαγές που έχουν γίνει μέσα σε ένα κείμενο, θα πρέπει, πολλές φορές, να το εξετάσει γραμμή προς γραμμή και ίσως να χρειαστεί να το ξαναδιαβάσει. Επιπλέον, θα ήθελε κάποιος να μάθει την εξέλιξη της επιστημονικής έρευνας σε σχέση με το χρόνο και να μελετήσει τα στάδια της προόδου αυτής.

Παρόμοιο θέμα με το παραπάνω είναι και ο **έλεγχος έκδοσης** (*version control*) [6]. Ας θεωρήσουμε μία εταιρία παραγωγής λογισμικού (αλλά και γενικότερα σε μία εταιρία υλοποίησης έργων). Ένα έργο, ως γνωστόν, δουλεύεται σε ομάδες, η καθεμία από τις οποίες περιλαμβάνει άτομα με διάφορες ειδικότητες για διαφορετική εργασία κάθε φορά. Έτσι, οι πόροι και τα αλλά επιμέρους στοιχεία του έργου, συνεχώς αλλάζουν. Υπάρχει επιτακτική ανάγκη, ο *Διαχειριστής* του έργου να γνωρίζει κάθε αλλαγή που συνέβη και να μπορεί να διαχειριστεί αυτές τις αλλαγές. Δηλαδή θα χρειάζεται να ξέρει Πότε, από Ποιον, και με Ποιο Τρόπο συνέβησαν τροποποιήσεις σε έναν ορισμένο πόρο του έργου, καθώς επίσης θα χρειάζεται να μπορεί να επιλέγει και το επίπεδο λεπτομέρειας για τον κάθε πόρο. Έτσι, για παράδειγμα, ίσως ο *Διαχειριστής* του έργου να ενδιαφέρεται για το γεγονός ότι άλλαξε ένας αλγόριθμος ενός τμήματος του κώδικα που περιέχεται σε μία συγκεκριμένη συνάρτηση, ή, ανεβάζοντας το επίπεδο λεπτομέρειας, να ενδιαφέρεται μόνο για το γεγονός ότι άλλαξε η ίδια η συνάρτηση χωρίς να τον απασχολούν οι λεπτομέρειες του περιεχομένου της αλλαγής.

Για το θέμα του *version control* έχουν γίνει αρκετές προσπάθειες από μερικές εταιρίες παραγωγής λογισμικού ή σχεδίασης μεταγλωττιστών ανά τον κόσμο. Έτσι, έχουμε ονόματα εταιριών όπως LBMS, MICROSOFT, INTEROLV, MKS, POWERSOFT οι οποίες έχουν παρουσιάσει στην αγορά αξιόλογα εργαλεία για τον χειρισμό αυτού του θέματος. Ωστόσο, αυτά τα εργαλεία είναι εξειδικευμένα για τη χρήση αντικειμένων σε ένα έργο λογισμικού. Δεν προσφέρουν ευελιξία και επεκτασιμότητα καθώς επίσης και γενίκευση σε αφηρημένα αντικείμενα ή έννοιες, προς μελέτη. Επιπλέον, δεν υπάρχει ενιαίο **πρότυπο μορφής** (*format*) αυτών των αρχείων καταγραφής των αλλαγών με αποτέλεσμα να μην υπάρχει η παραμικρή δυνατότητα επικοινωνίας μεταξύ τους και διασυνδεσιμότητας με άλλες, ανομοιογενείς, εφαρμογές και συσκευές.

Ένα άλλο παράδειγμα μέσα από το χώρο της πληροφορικής είναι η **καταγραφή των ενεργειών** (*logging*) που συμβαίνουν μέσα σε ένα υπολογιστικό σύστημα με σκοπό τον έλεγχο του ιστορικού από το διαχειριστή του συστήματος για λόγους ασφάλειας, στατιστικής μελέτης, διόρθωσης σφάλματος, επαναφοράς συστήματος, κ.ο.κ. Έως τώρα, υπάρχουν τέτοιοι αυτόματοι μηχανισμοί από μία ποικιλία εταιριών παραγωγής λογισμικού, λειτουργικών συστημάτων και τεχνολογίας Υλικού. Ωστόσο δεν υπάρχει ένα πρότυπο που θα μπορεί να περιγράψει σε τυποποιημένη μορφή αυτές τις αλλαγές, έτσι ώστε να υπάρχει **διαεπικοινωνία** μεταξύ αυτών των μηχανισμών για τη διαχείριση της πληροφορίας σε κοινή βάση.

Πέρα από το Διαδίκτυο και την Πληροφορική, στην καθημερινή ζωή, τα πράγματα αλλάζουν συνεχώς. Πολλές φορές οι αλλαγές που συμβαίνουν, δεν μας ενδιαφέρουν, άλλες πάλι φορές ενδιαφερόμαστε ΜΟΝΟ γι' αυτές. Για παράδειγμα ένας ιστορικός ή ένας μελετητής ενός οργανισμού, ενδιαφέρεται για την καταγραφή των αλλαγών που συμβαίνουν στο σύστημα που είναι υπεύθυνος να παρακολουθεί, είτε αυτό είναι πολιτικό, κοινωνικό, είτε είναι διοικητικό και πληροφοριακό σύστημα κ.ο.κ. Το αποτέλεσμα είναι να προκύπτουν δεδομένα τα οποία ΔΕΝ είναι σε μορφή αυστηρά τυποποιημένη, έτσι ώστε να είναι δυνατή η κατανόηση και η αξιοποίηση αυτών των δεδομένων σε μορφή πληροφορίας από τον ίδιο τον ιστορικό ή μελετητή, αλλά και από την ευρύτερη επιστημονική κοινότητα.

Έτσι λοιπόν, όπως είδαμε από τα προηγούμενα παραδείγματα, τίθενται θέματα ανάκτησης της κατάστασης ενός συστήματος σε μία παλαιότερη χρονική στιγμή, ανάκτησης των αλλαγών-διαφοροποιήσεων που συνέβησαν σε ένα σύστημα από μία παλαιότερη χρονική στιγμή μέχρι την τωρινή του κατάσταση και τέλος τίθενται θέματα επαναφοράς ενός συστήματος σε μία παλαιότερη κατάστασή του.

3.2. Η ανάγκη ύπαρξης μοντέλου

Με τις αμφίδρομες επικοινωνίες (Internet, Κινητή Τηλεφωνία, κ.α.) να καταλαμβάνουν πλέον ένα σημαντικό μερίδιο του επαγγελματικού και μη, χώρου, υπάρχει ανάγκη για φθηνή επικοινωνία και μετάδοση πληροφοριών, με ελάχιστο πλεόνασμα, σε τυποποιημένη μορφή και με δυναμικότητα στη χρήση και επεξεργασία τους. Όπως καταλαβαίνουμε, ο σύγχρονος άνθρωπος – επαγγελματίας - χρήστης ενδιαφέρεται για την περιεκτική, ταξινομημένη και ουσιώδη πληροφορία, η οποία θα του παρέχεται με αυτοματοποιημένο τρόπο, έτσι ώστε να ελαχιστοποιήσει τα έξοδά του αλλά και τη δικτυακή συμφόρηση.

Το πρόβλημα της ανομοιογένειας της πληροφορίας και της αταξινόμητης ροής αυτής, στη γενική της μορφή, υπάρχει από την αρχή της Ιστορίας. Έτσι, η αποθήκευση της γνώσης πέρασε από αρκετά στάδια μέχρι να καταλήξει στη σημερινή της ηλεκτρονική μορφή. Αυτό το πρόβλημα έχει αρχίσει να γίνεται κίνδυνος, όσο οι εξελικτικές καταστάσεις της σημερινής εποχής θα οδηγούν στην παγκοσμιοποίηση της επικοινωνίας και της ανταλλαγής πληροφοριών.

Καθώς η Ερμηνεία δίνει νόημα στα Δεδομένα και τα μετατρέπει σε Πληροφορία, η ταξινόμηση και η τυπική σύνταξη της πληροφορίας προσθέτει άξια στην ίδια την πληροφορία δίνοντας της ευελιξία στη διαχείρισή της.

Πιο συγκεκριμένα, η ίδια η ιδέα του υπερκειμένου έφερε επανάσταση με την έναρξη της προηγούμενης δεκαετίας στο χώρο του Διαδικτύου, καθώς πλέον έδωσε ένα μεγάλο πλεονέκτημα στην προσβάσιμη πληροφορία: την διασυνδετικότητα. Αυτό σε συνδυασμό με το πλεονέκτημα που μας δίνει ο τομέας των πολυμεσικών τρόπων παρουσίαστης της πληροφορίας, προκάλεσε μία έκρηξη στον τρόπο σκέψης που είχαν μέχρι τότε οι άνθρωποι για την αποθήκευση και την ανάκτηση της Γνώσης.

Η «διαδικτυοποίηση» όμως δεν είχε μόνο σκοπό τη μονόδρομη πληροφόρηση, όπως έκαναν μέχρι τότε τα μέσα μαζικής ενημέρωσης. Έτσι, το Δίκτυο Υπολογιστών προώθησε την αμφίδρομη πληροφόρηση και επομένως την ανταλλαγή των πληροφοριών. Το υπερκείμενο (HTML) όμως συναντά δυσκολίες στο να πετύχει αυτό το σκοπό. Το πρόβλημα είναι ότι δεν μπορεί να διαχωρίσει την πληροφορία παρουσίασης από την ίδια την πληροφορία [2]. Έτσι, στην πραγματικότητα, με τις σελίδες υπερκείμενου HTML είχαμε απλά «ηλεκτρονικοποίηση» μιας λειτουργίας

που γινόταν εδώ και χρόνια: την τυπογραφία, μαζί με κάποια «ηλεκτρονικά» πλεονεκτήματα. Ο Χρήστης είναι υποχρεωμένος να πάρει όλη την πληροφορία και μετά να την επεξεργαστεί και να αποφασίσει και να επιλέξει τα κομμάτια που τον ενδιαφέρουν. Με βάση το πρόβλημα αυτό, ο σχεδιασμός της γλώσσας XML σε συνδυασμό με όλες τις συναφείς τεχνολογίες (XSL, XLINK, XPATH κ.ο.κ) [15] που τη συνοδεύουν, καλείται να δώσει λύση, φροντίζοντας να διαχωρίσει τις ίδιες τις πληροφορίες από τις πληροφορίες παρουσίασης. Έτσι, δίνει μεγαλύτερη ευελιξία στον χειρισμό των θεμάτων τα οποία αναφέραμε πιο πάνω. Με δύο λόγια, έγινε μία μελέτη από πολλές και έμπειρες τεχνικές ομάδες από όλο τον κόσμο με σκοπό τη μοντελοποίηση αυτών των μηχανισμών διαχωρισμού και επανένωσης της πληροφορίας παρουσίασης και της μετά-πληροφορίας. Αυτομάτως αποκτήθηκαν πλεονεκτήματα ευελιξίας, αυτοματισμού, επεκτασιμότητας και αυστηρού έλεγχου της πληροφορίας.

Έτσι, μπορεί πλέον ένας Χρήστης να διαχειριστεί τα δεδομένα του και μετά να τα παρουσιάσει όπως αινός θέλει, με συνάφεια, σε μία πλατφόρμα απαλλαγμένη από τεχνικά προβλήματα, ασυμβατότητες μέσων και μηχανισμών κ.ο.κ., με μερικούς εξαιρετικά ευέλικτους μηχανισμούς (XSL, CSS κλπ) [15]. Ταυτόχρονα έχει τη δυνατότητα να επαναχρησιμοποιήσει τις πληροφορίες δημιουργώντας έτσι μία ισχυρότατη, δυναμική βάση γνώσης, μέσα από μία απλή σελίδα υπερκείμενου.

Με το χαρακτηρισμό όμως “δυναμική” βάση γνώσης εννοούμε ότι θα έχουμε αλλαγές στο περιεχόμενο και τη δομή αυτής της βάσης σε πραγματικό ή μη χρόνο, πράγμα που σημαίνει ότι αυτομάτως τίθεται το θέμα της διαχείρισης αυτών των αλλαγών. Τίθεται λοιπόν η ανάγκη για μία ενιαία, πρότυπη, και συναφή διαχείριση της καταγραφής των αιτιών και των αποτελεσμάτων μιας αλλαγής, αλλά και της ανάκτησης αυτών των αλλαγών σε επόμενες χρονικές στιγμές.

Επόμενο βήμα λοιπόν, ακολουθώντας το δρόμο που χάραξε το υπερκείμενο και αμέσως μετά η XML, και με βάση τα προβλήματα που αναπτύξαμε στις προηγούμενες παραγράφους, είναι η μοντελοποίηση στον τρόπο που θα γίνεται η διαχείριση των αλλαγών σε ένα σύστημα που αφορά το Διαδίκτυο.

Κατ’ επέκταση, μπορούμε να γενικεύσουμε το ζητούμενο θέμα, έτσι ώστε το μοντέλο μας να μην είναι κατάλληλο μόνο για συστήματα αποκλειστικά για το Διαδίκτυο, αλλά για ένα οποιοδήποτε σύστημα προς μελέτη, στο γενικό ορισμό του.

Πριν προχωρήσουμε στην ανάπτυξη των απαιτήσεων που θα πρέπει να έχουμε από ένα τέτοιο μοντέλο, ας δούμε μερικές περιπτώσεις κατά τις οποίες μπορεί να φανεί η αξία που έχει η προτυποποίηση των αλλαγών, σε διαφόρων ειδών συστήματα.

3.3. Σενάρια χρήσης του μοντέλου

Η βάση στην οποία στηρίζεται η έρευνα και η μελέτη σχεδίασης αυτού του μοντέλου, είναι ο χώρος των υπολογιστών διότι εκεί εστιάζεται ένα μεγάλο μέρος των προβλημάτων που συζητήσαμε στις προηγούμενες παραγράφους. Όμως το μοντέλο μπορεί να βρει εφαρμογές και έξω από το χώρο των δικτύων, των υπολογιστών και γενικά της ηλεκτρονικής τεχνολογίας.

Ξεκινώντας όμως πρώτα από το Διαδίκτυο, ως ένα κύριο σενάριο χρήσης είδαμε ότι είναι η αυτοματοποιημένη ενημέρωση του Χρήστη του Διαδικτύου, σχετικά με τις αλλαγές που συμβαίνουν σε Χώρους που επισκέπτεται τακτικά. Έτσι, για παράδειγμα, υλοποιώντας έναν έξυπνο πράκτορα είτε στη μεριά του υπολογιστή του

Χρήστη, ή στη μεριά του εξυπηρετητή του Διαδικτύου (web server), θα μπορούμε να καταγράφουμε τις αλλαγές που συμβαίνουν, κάθε φορά που ο Διαχειριστής του Χώρου αλλάζει τα περιεχόμενα του δικτυακού τόπου, με αποτέλεσμα ο Χρήστης να ενημερώνεται είτε άμεσα, ή όποτε αυτός το θελήσει για τις συγκεκριμένες αλλαγές που τυχόν να τον ενδιαφέρουν. Παράλληλα, ο Χρήστης θα μπορεί να διαλέγει και το επίπεδο λεπτομέρειας που θα γίνεται η ενημέρωση. Έτσι, θα μπορεί να διαλέξει μεγάλο βαθμό λεπτομέρειας και επομένως να ενημερώνεται ακόμα και όταν αλλάζει έστω και μία λέξη ή ένα σύμβολο από κάποια σελίδα. Από την άλλη μεριά θα μπορεί να διαλέξει μικρό βαθμό λεπτομέρειας όποτε να ενημερώνεται κάθε φορά που αλλάζει εξ' ολόκληρου μία σελίδα ή ένα αρχείο στο δικτυακό Χώρο.

Ένα άλλο παράδειγμα για τη χρήση του μοντέλου στο Διαδίκτυο αλλά και, γενικά, στη δημοσίευση και ανταλλαγή εγγράφων είναι το **ιστορικό των αλλαγών** για νεότερες εκδόσεις. Δηλαδή, θα μπορούν να κρατούνται σε αυστηρά δομημένο αρχείο οι αλλαγές που έκανε ο συγγραφέας (οι συγγραφείς) του έγγραφου, έτσι ώστε ένας αναγνώστης που έχει μία παλαιότερη έκδοση να μπορεί να μελετήσει τις νέες τροποποιήσεις. Επιπλέον ο αναγνώστης, έχοντας την παλαιότερη έκδοση και το αρχείο καταγραφής των αλλαγών, θα μπορεί, μέσω κατάλληλης διαδικασίας, να μετατρέψει την παλαιά έκδοση στην καινούρια, χωρίς να χρειάζεται να «κατεβάσει» ξανά ολόκληρο το νέο έγγραφο από το Διαδίκτυο (**document patching**).

Το ίδιο ακριβώς συμβαίνει και στα αρχεία εγκατάστασης μιας εφαρμογής, όπου εκεί το πρόβλημα της ανανέωσης έκδοσης γίνεται ακόμη εντονότερο καθώς αυξάνονται τα μεγέθη των αρχείων που θα πρέπει να «κατεβάσει» ο Χρήστης από το Διαδίκτυο (**application patching**). Πρέπει βέβαια να τονίσουμε ότι η ανανέωση εφαρμογών γίνεται ήδη από τις μεγάλες εταιρείες λογισμικού, ωστόσο αναζητάμε, όπως είπαμε, την ενοποίηση και την προτυποποίηση αυτών των μηχανισμών, έτσι ώστε να μας δίνεται η δυνατότητα να κάνουμε και επιλεκτική ανανέωση, ακόμα και από τον πλοηγητή (browser) του Διαδικτύου που διαθέτουμε.

Επιπροσθέτως, ένας έξυπνος πράκτορας που θα βρίσκεται μόνιμα φορτωμένος και θα ελέγχεται από το **λειτουργικό σύστημα** του υπολογιστή μας θα μπορεί να καταγράφει κάθε ενέργεια που συμβαίνει στους πόρους του συστήματος, έτσι ώστε να είναι δυνατόν, κατά πρώτον, να ελέγχουμε για θέματα ασφάλειας (ιοί, εισβολείς) τον υπολογιστή μας και, κατά δεύτερον, να μπορούμε να επαναφέρουμε το σύστημά μας σε παλαιότερες καταστάσεις σε περίπτωση δυσλειτουργίας ή κατάρρευσης.

Παρόμοια εφαρμογή για θέματα ασφάλειας, εκτός από τα λειτουργικά συστήματα, είναι και ένα πιο γενικευμένο **σύστημα συναγερμού** που θα καταγράφει τις ενέργειες και τα αποτελέσματα που συμβαίνουν κατά τη διάρκεια παρακολούθησης, έτσι ώστε, εκ των υστερών, να μπορούμε να ανακτήσουμε, επιλεκτικά, αυτά τα γεγονότα και να τα μελετήσουμε.

Όπως είπαμε, το μοντέλο που αναζητάμε θα πρέπει ακολουθεί τα πλεονεκτήματα της γλώσσας XML και αυτό σημαίνει ότι θα πρέπει υπάρχει δυνατότητα ανταλλαγής πληροφορίας ανάμεσα σε ανομοιογενή συστήματα. Έτσι, για παράδειγμα, ένα δορυφορικό **σύστημα** θα μπορεί να στέλνει πληροφορίες για τον καιρό και για το χρηματιστήριο κατευθείαν στη σελίδα μας, σε μορφή XML, η οποία θα μετατρέπεται και θα παρουσιάζεται όπως εμείς θέλουμε και, επιπλέον, θα μπορούν να στέλνονται και οι αλλαγές που συμβαίνουν στους δύο αυτούς τομείς και να απεικονίζονται δυναμικά στη σελίδα μας. Παράλληλα θα υπάρχει δυνατότητα **ανταλλαγής δοσοληψιών (transactions)** ανάμεσα σε ετερογενείς βάσεις δεδομένων που χρησιμοποιούνται κάτω από τον ελέγχο του ίδιου πληροφοριακού συστήματος

κάποιου οργανισμού. Για παράδειγμα, θα μπορούμε να έχουμε επικοινωνία ενός προγράμματος CAD με μία βάση δεδομένων, ή θα μπορούμε να ανταλλάξουμε εντολές αλλαγής σχήματος (**database schema**) και περιεχομένου σε βάσεις δεδομένων που ανήκουν σε διαφορετικούς κατασκευαστές.

Μία άλλη εφαρμογή μπορούμε να συναντήσουμε στον **τομέα των πολυμέσων**. Τα πολυμέσα χαρακτηρίζονται από την αλληλεπιδραστική επικοινωνία με το Χρήστη και τη δυναμική αλλαγή στην κατάσταση των στοιχείων του (κινούμενοι χαρακτήρες, ήχοι κ.λ.π.). Έτσι, σε μία εφαρμογή πολυμέσων από το Διαδίκτυο θα μπορούν να στέλνονται οι ενέργειες και τα αποτελέσματα των φορέων των αλλαγών και των αντικειμένων που αλληλεπιδρούν κατά τη λειτουργία αυτής της εφαρμογής. Για παράδειγμα, αν ένας Χρήστης που έχει εγκαταστήσει ένα εικονικό πολυμεσικό θέατρο στον υπολογιστή του, έχει «εγγράψει» στο σύστημα του, κάποιους ηθοποιούς - αντικείμενα, τότε θα χρειάζεται να «κατεβάζει» από το Διαδίκτυο MONO τις ενέργειες και τα αποτελέσματα κάθε επεισοδίου που ενδιαφέρεται να δει, και μετά θα θέτει σε λειτουργία αυτό το εικονικό σύστημα, το οποίο θα αναλαμβάνει το ρόλο του σκηνοθέτη...

Στον τομέα της **Αυτοματοποίησης** και της **Κυβερνητικής** η καταγραφή του δέλτα, δηλαδή των αλλαγών που συμβαίνουν σε ένα σύστημα, είναι πολύ σημαντικό ζήτημα. Έτσι, το πλεονέκτημα που θα μπορεί να δώσει το μοντέλο που αναζητάμε είναι η επαναχρησιμοποίηση των καταγεγραμμένων πληροφοριών. Παραδείγματα εφαρμογής είναι αυτά της εξομοίωσης συστήματος, της αυτοματοποίησης μηχανισμού παραγωγής προϊόντος, της τηλεδιαχείρισης συσκευής κ. λ. π.

Ολοκληρώνοντας την περιγραφή των σεναρίων στα οποία μπορεί να βρει εφαρμογή το μοντέλο αυτό, θα αναφερθούμε και στη γενικότερη θεώρηση ενός συστήματος το οποίο μπορεί να είναι ένα **κοινωνικό**, ένα **πληροφοριακό**, ένα **φυσικό**, ένα **βιολογικό** σύστημα και οποιαδήποτε άλλη μορφή μπορούμε να συναντήσουμε στη θεωρία. Σε όλα αυτά τα συστήματα συμβαίνουν αλλαγές, η μελέτη των οποίων πολλές φορές είναι σημαντική για τον τομέα της εκάστοτε επιστήμης. Έτσι, σε πολλά παραδείγματα-σενάρια τίθεται η ανάγκη για την καταγραφή των αλλαγών, η ανάγκη για την ανάκτηση των παλαιοτέρων καταστάσεων ενός συστήματος και η ανάγκη για την πιθανή επαναφορά του συστήματος σε παλαιότερες καταστάσεις. Ταυτόχρονα, ενδιαφερόμαστε για την ομοιόμορφη, αυστηρά δομημένη καταγραφή αυτών των πληροφοριών, έτσι ώστε να είναι δυνατή η (αυτόματη) ανταλλαγή και η κατανόησή τους, ανάμεσα στις επιστημονικές κοινότητες ανά τον κόσμο. Συνοπτικά, θα αναφέρουμε μερικούς τομείς όπου βρίσκει εφαρμογή το παραπάνω : την **Αστρονομία** όπου μας ενδιαφέρει η κατάσταση ενός πλανητικού συστήματος, την **Ιατρική** όπου μας ενδιαφέρει η κατάσταση της υγείας ενός ασθενή σε συνδυασμό με μία θεραπεία, την **Οργάνωση** και **Διοίκηση** στην οποία μας ενδιαφέρουν οι «δράσεις» και οι «αντιδράσεις» στα επιμέρους τμήματα ενός οργανισμού κ.ο.κ.

Βλέπουμε, δηλαδή, ότι οι εφαρμογές για ένα τέτοιο μοντέλο είναι θεωρητικά απεριόριστες, καθόσον θα συμβαίνουν γύρω μας συνεχώς αλλαγές και όσο θα μπορούμε να καταγράφουμε και να μελετάμε αυτές τις αλλαγές.

3.4. Απαιτήσεις μοντέλου – Σκοπός και στόχοι

Με βάση την παραπάνω ανάπτυξη του προβλήματος, είμαστε σε θέση να δούμε αναλυτικότερα τις **απαιτήσεις** που θα πρέπει να έχουμε από ένα τέτοιο μοντέλο, δηλαδή τους στόχους που θα πρέπει να θέσουμε, κατά τη μελέτη και τη σχεδίασή του.

Επίσης, θα μπορέσουμε να καθορίσουμε και τον κύριο σκοπό αυτής της μελέτης, δηλαδή τις συγκεκριμένες λειτουργίες που θα πρέπει να έχει ένα τέτοιο μοντέλο.

Έτσι, αναλυτικά, ζητείται ένα μοντέλο για τις εξής λειτουργίες:

- Παρατήρηση ενός συστήματος
- Καταγραφή αλλαγών
- Διαχείριση αλλαγών
- Ανάκτηση αλλαγών
- Επαναφορά του συστήματος

Χωρίζουμε τη λειτουργικότητα του μοντέλου με αυτόν τον τρόπο, διότι καθεμία από τις λειτουργίες αυτές αποτελεί μια ξεχωριστή λογική ενότητα. Έτσι, σε καθεμία από αυτές, τίθενται ορισμένα θέματα και προβλήματα για συζήτηση και επιπλέον προτείνονται κάποιες λύσεις ή ιδέες με σκοπό την, όσο το δυνατόν, καλύτερη εφαρμογή τους, κατά την υλοποίηση του μοντέλου.

Για τη λειτουργία αυτού του μοντέλου, χρειάζεται ένα είδος *σημειογραφίας* - ψευδογλώσσας, την οποία θα χρησιμοποιήσουμε για την αναπαράσταση των αλλαγών με δομημένο και τυπικό τρόπο, καθώς επίσης και ένα σύνολο εντολών για τη διαχείριση αυτών των καταγραφών.

Όπως είπαμε και στις προηγούμενες ενότητες, τέτοιες σημειογραφίες υπάρχουν ήδη στο χώρο της πληροφορικής. Ωστόσο, αυτές αφορούν ένα συγκεκριμένο πρόβλημα κάθε φορά και δεν μπορούν να καλύψουν όλες τις περιπτώσεις που αναφέραμε στα σενάρια χρήσης. Τίθεται λοιπόν η απαίτηση για **Γενίκευση**, δηλαδή την ύπαρξη ενός μηχανισμού, ο οποίος θα μπορεί να εφαρμόζεται σε όσο το δυνατό περισσότερες περιπτώσεις συστημάτων, ξεκινώντας από το Διαδίκτυο και καταλήγοντας σε ένα κοινωνικό ή σε ένα μαθηματικό σύστημα.

Με την απαίτηση της γενίκευσης, αυτομάτως, απαιτούμε και την **Επεκτασιμότητα**, δηλαδή τη δυνατότητα δημιουργίας νέων «δομικών στοιχείων» για την ικανοποίηση του προβλήματος που καλούμαστε να λύσουμε κάθε φορά. Έτσι, χωρίς να υπάρχει ασυμβατότητα προς τις παλαιότερες εκδόσεις, θα μπορούμε να προσθέτουμε κανόνες, τύπους, συμβολισμούς κλπ, έτσι ώστε να μπορεί το μοντέλο να προσαρμοστεί στις εξελίξεις που συμβαίνουν στις διάφορες τεχνολογίες. Ακόμα και στην περίπτωση που το Υλικό θα θεωρείται απαρχαιωμένο, η πληροφορία που, αυτό, θα επεξεργάζεται, θα μπορεί να χρησιμοποιείται σε νέα συστήματα, χωρίς όμως να χάνεται η άξια και η σημασία της.

Επιπλέον, ζητείται να υπάρχει **Επαναχρησιμοποιησιμότητα** και **Μεταφερσιμότητα** της πληροφορίας που θα αποθηκεύουμε κατά τη λειτουργία του μοντέλου μας. Αυτό σημαίνει ότι η πληροφορία θα μπορεί να (ξανά)χρησιμοποιείται από πολλές συσκευές και μηχανισμούς, οι οποίοι δεν θα χρειάζεται να είναι ομογενείς ή να έχουν λογική και φυσική σχέση μεταξύ τους. Με αυτό τον τρόπο θα μπορούμε να διευρύνουμε το φάσμα εφαρμογών του μοντέλου και έξω από το χώρο του διαδικτύου. Επίσης η πληροφορία θα μπορεί να ανταλλάζεται μεταξύ συστημάτων, οργανισμών, πρακτόρων, βάσεων δεδομένων κλπ, χωρίς να χρειάζεται κάποια μετατροπή ή αλλαγή.

Μια άλλη απαίτηση είναι η **Ευελιξία** και ο **Έλεγχος** της πληροφορίας που θα αποθηκεύουμε και θα ανακτούμε. Ζητάμε, δηλαδή, έναν τρόπο αναπαράστασης ο οποίος θα μπορεί να περιγράψει όλες τις περιπτώσεις συστημάτων, με οποιοδήποτε

επίπεδο λεπτομέρειας εμείς θελήσουμε να έχουμε. Ταυτόχρονα, και καθώς η πληροφορία θα είναι αυστηρά δομημένη και θα έχει καθορισμένη σημασία, θα διευκολύνεται το έργο της αυτοματοποίησης της επεξεργασίας της.

Τέλος, η γλώσσα καταγραφής θα πρέπει να είναι απλή, έτσι ώστε να είναι εύκολα **εφαρμόσιμη, αναγνώσιμη και υλοποιήσιμη**, από όλους τους τεχνολογικούς ή μη, τομείς οι οποίοι ενδιαφέρονται να υιοθετήσουν ένα τέτοιο μοντέλο για το σύστημά τους.

Από τις λειτουργικές απαιτήσεις του μοντέλου, θα έχουμε κατά κύριο λόγο, την **ευθυγράμμιση** του μοντέλου με τους ήδη υπάρχοντες μηχανισμούς της τεχνολογίας SGML και, ειδικότερα, της XML. Οι τεχνολογίες των γλωσσών *markup* όπως η SGML και η XML, προσφέρουν ένα σύνολο από πλεονεκτήματα και νέες δυνατότητες σε προβλήματα που ήδη έχουν κάνει την εμφάνισή τους κατά τη διάρκεια της εξέλιξης του διαδικτύου και των κατανεμημένων συστημάτων, γενικότερα.

Επιπλέον, ζητείται **σφαιρικότητα** στη χρήση του μοντέλου, δηλαδή μια δομή της γλώσσας καταγραφής των αλλαγών η οποία θα μπορεί να αναφέρεται σε αντικείμενα, συστήματα ή άλλους πόρους, τα οποία θα βρίσκονται έξω από το χώρο δράσης του συστήματος που μελετάμε. Δηλαδή, με τη χρήση μοναδικών, παγκοσμίως, διευθύνσεων, θα μπορούμε να χρησιμοποιούμε στοιχεία από αλλά συστήματα, οργανισμούς ή οντότητες επεξεργασίας δεδομένων (πράκτορες, βάσεις δεδομένων κλπ).

Κεφαλαιο 4

Ανάλυση μοντέλου



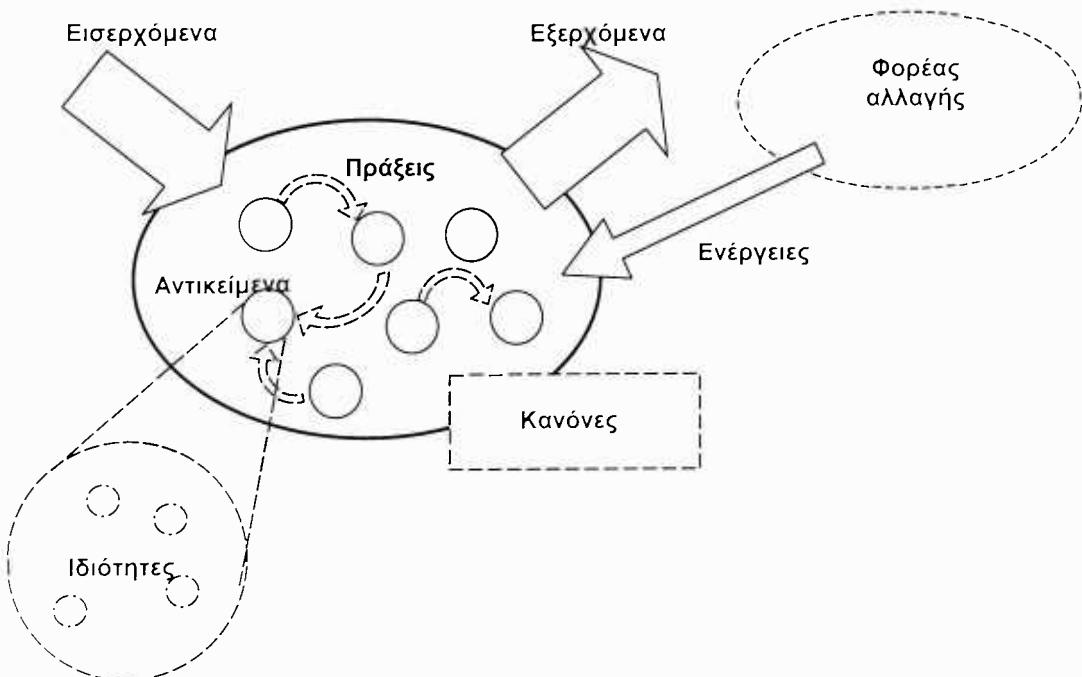
4. Ανάλυση μοντέλου

4.1. Πρόλογος

Πριν ξεκινήσουμε την ανάλυση του μοντέλου που προτείνουμε για την ανάκτηση δέλτα ενός συστήματος, υπάρχει ανάγκη να αποσαφηνίσουμε τα πράγματα γύρω από το σύστημα, το οποίο θα αναλύσουμε πριν εφαρμόσουμε το ίδιο το μοντέλο. Αυτό θα μας βοηθήσει στη μεθοδικότερη, άρα και ευκολότερη αντίληψη του τι ακριβώς θέλουμε να πετύχουμε.

4.2. Η έννοια του συστήματος

Αφού είδαμε στις προηγούμενες παραγράφους τους ορισμούς και τις κατηγορίες των συστημάτων από διάφορες οπτικές γωνίες, ας δώσουμε έναν, όσο γίνεται περισσότερο γενικευμένο ορισμό περί του συστήματος που θα μας βοηθήσει στην περαιτέρω ανάλυση του μοντέλου για το οποίο ενδιαφερόμαστε.



Σχήμα 1. Το σύστημα

Μπορούμε να ορίσουμε ένα σύστημα ως ένα σύνολο από καλά καθορισμένα αντικείμενα και ένα σύνολο από αλληλεπιδράσεις μεταξύ τους οι οποίες ακολουθούν κάποιους κανόνες που εφαρμόζονται πάνω στα αντικείμενα του συστήματος.

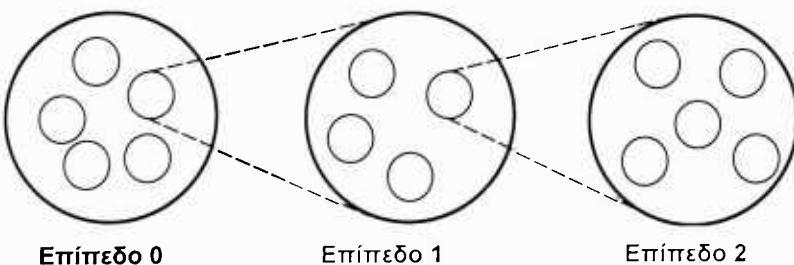
Λέγοντας αντικείμενα, εννοούμε τα επιμέρους τμήματα ενός συστήματος, τα οποία «υπάρχουν» και αλληλεπιδρούν μέσα στο χώρο δράσης του συστήματος. Το αντικείμενο μπορεί να αποτελεί μία αφηρημένη έννοια, δηλαδή να μην είναι απαραίτητα, κατά έναν τρόπο, αντίληπτό με τη σημασία της λέξεως «αντικείμενο». Αυτό συμβαίνει συχνά σε θεωρητικής υφής συστήματα όπου δεν γνωρίζουμε ακριβώς τη φύση ενός ή περισσότερων αντικειμένων του συστήματος που αναλύουμε. Επίσης συμβαίνει σε αναλύσεις συστημάτων που χρησιμοποιούμε τους όρους «έννοια» ή/και «οντότητα». Για παράδειγμα, η ανάλυση των κοινωνικών ή πληροφοριακών συστημάτων έχει να επιδείξει πολλά τέτοια παραδείγματα αφηρημένων εννοιών των αντικειμένων ενός συστήματος. Αυτό όμως δεν σημαίνει ότι τα αντικείμενα του συστήματός μας δεν πρέπει να είναι καλά καθορισμένα.

Λέγοντας «καλά καθορισμένα» αντικείμενα, αυτομάτως εννοούμε ότι έχουμε ξεχωρίσει ποια αντικείμενα δεν ανήκουν στο σύστημα και άρα αποτελούν το περιβάλλον του συστήματος. Το περιβάλλον του συστήματος κατά ένα τρόπο «περικλείει» το χώρο δράσης του συστήματος και ανάλογα με το είδος του, αλληλεπιδρά ή όχι με αυτόν. Έχουμε δηλαδή τα ανοιχτά και τα κλειστά συστήματα, αντίστοιχα.

Αλληλεπίδραση με το περιβάλλον σημαίνει ότι υπάρχουν εισερχόμενα και εξερχόμενα από αυτό. Για τα εισερχόμενα και εξερχόμενα θα μιλήσουμε στις επόμενες παραγράφους όταν αναφερθούμε στις αλλαγές του συστήματος.

Χρησιμοποιώντας την επαγωγή, ένα αντικείμενο μπορεί να αποτελεί από μόνο του ένα σύστημα του αρχικού συστήματος. Αυτό το σύστημα, το οποίο θα το ονομάζουμε υποσύστημα, πρέπει, ανάλογα με τους αυστηρά καθορισμένους κανόνες που έχουμε ορίσει για την ανάλυση του συστήματος, να είναι σαφώς ορισμένο αλλά και περιορισμένο. Δηλαδή η επιλογή των αντικειμένων που αποτελούν υποσυστήματα και η επιλογή μας για μεγαλύτερη (σε βάθος) ανάλυση αυτών, εξαρτάται, κατά πρώτον, από την όψη που βλέπουμε το σύστημα και, κατά δεύτερον, από τις απαιτήσεις μας κατά την ανάλυση του συστήματος. Πριν δώσουμε ένα παράδειγμα, ας εισάγουμε την έννοια του επιπέδου λεπτομέρειας σε ένα σύστημα.

Όταν αναλύουμε ένα σύστημα, όλα τα υποσυστήματα (και μόνο αυτά) που ανήκουν άμεσα σε ένα σύστημα, θα λέμε ότι ανήκουν στο ίδιο επίπεδο λεπτομέρειας.



Σχήμα 2 Επίπεδα Λεπτομέρειας

Χρησιμοποιώντας το μοντέλο της γενετικής, θα λέμε ότι στο πρώτο επίπεδο λεπτομέρειας ενός συστήματος με ρίζα τον πατέρα-σύστημα, θα ανήκουν όλοι οι άμεσοι απόγονοι – συστήματα αυτού του πατέρα. Με τον ίδιο τρόπο, στο δεύτερο επίπεδο λεπτομέρειας θα ανήκουν οι άμεσοι απόγονοι των υποσυστημάτων του πατέρα, δηλαδή τα «εγγόνια» κ.ο.κ (βλ Σχήμα.2)

Όπως αναφέραμε και προηγουμένως, πρέπει να διευκρινίζονται σαφώς τα επιμέρους στοιχεία του κάθε συστήματος και υποσυστήματος και να καθορίζονται ακριβώς τα σύνορα του κάθε ενός, από τα επίπεδα λεπτομέρειας. Σε κάθε επίπεδο λεπτομέρειας, συνήθως, υπάρχουν ομογενή αντικείμενα δηλαδή αντικείμενα τα οποία, με την έννοια που τα αναλύουμε, έχουν κάτι κοινό. Αυτό δεν είναι απαραίτητο, αλλά βοηθάει στην καλύτερη τεκμηρίωση της ανάλυσής μας.

Ας εξετάζουμε, ως σύστημα, ένα εμπορικό κέντρο με τρεις ορόφους.

Ας υποθέσουμε ότι θέλουμε να μελετήσουμε τον κάθε όροφο ξεχωριστά. Τότε μπορούμε να θεωρήσουμε ότι κάθε ένας όροφος αποτελεί ένα υποσύστημα και ανήκει στο πρώτο επίπεδο λεπτομέρειας. Ένα δεύτερο επίπεδο λεπτομέρειας θα μπορούσε να είναι τα καταστήματα που έχει ο κάθε όροφος, και ένα τρίτο θα μπορούσε να είναι οι χώροι ή αλλιώς τα δωμάτια του κάθε καταστήματος. Στην περίπτωση που, για παράδειγμα, ο τρίτος όροφος αποτελούνταν από ένα μόνο κατάστημα τότε θα μπορούσαμε να είχαμε θεωρήσει αυτόν τον όροφο ως κατάστημα και, έτσι, να είχαμε βάλει το κατάστημα στο πρώτο επίπεδο λεπτομέρειας. Αυτό είναι επιτρεπτό. Ωστόσο αν χρειαστούμε, αργότερα, να κάνουμε μία μελέτη για τα επίπεδα λεπτομέρειας του συστήματος του εμπορικού κέντρου τότε θα υπήρχε, ίσως, πρόβλημα στην τεκμηρίωση του κάθε επιπέδου.

Τα αντικείμενα έχουν **ιδιότητες** οι οποίες θεωρούνται γνωστές, μετρήσιμες και καταγραφείσες, με τα μέσα που διαθέτουμε μέχρι σήμερα. Φυσικά, μπορούν να υπάρχουν και ιδιότητες που δεν ανήκουν στην παραπάνω κατηγορία, που σημαίνει ότι μπορεί να μην είναι γνωστές ή καταγραφείσες, είτε μετρήσιμες. Μας ενδιαφέρουν μόνο όσες από τις ιδιότητες μπορούν να «καταγραφούν», με την κυριολεκτική σημασία της λέξεως. Ωστόσο, πρέπει να τονίσουμε ότι, παρόλο που μπορεί μία ιδιότητα να είναι αφηρημένη, μπορεί να καταγραφεί, αρκεί να μπορούμε να την αποθηκεύσουμε ή να την ανακτήσουμε αργότερα. Για παράδειγμα, μία ιδιότητα ενός ανθρώπου – αντικειμένου θα μπορούσε να είναι η ψυχολογική κατάστασή του, της οποίας, έστω και ασαφώς, μπορούμε να της αναθέσουμε διακριτές τιμές...

Στη συνέχεια θα μιλήσουμε για τους φορείς, τις ενέργειες, τις πράξεις, τα γεγονότα και τους κανόνες μιας αλλαγής η οποία συμβαίνει σε ένα σύστημα, χωρίς να έχουμε μιλήσει για την ίδια την αλλαγή, που είναι το κυρίως θέμα της επόμενης ενότητας.

Προς το παρόν θα αρκεστούμε στην ιδέα ότι, «αλλαγή» σε ένα σύστημα συμβαίνει, όταν κάτι αλλάζει...

Για να συμβεί μία αλλαγή στο σύστημα πρέπει να υπάρχει και ένας **φορέας αλλαγής**. Λέγοντας φορέα της αλλαγής, εννοούμε, με αρκετό επίπεδο αφαίρεσης, την αιτία για την οποία συνέβη ένα «αιτιατό». Φυσικά κάθε αιτιατό έχει και μία αιτία, εννοώντας όμως ότι μία αιτία μπορεί να είναι ...οτιδήποτε! Ο φορέας της αλλαγής είναι αυτός που είναι υπεύθυνος για μία αλλαγή που θα συμβεί σε ένα σύστημα. Ο φορέας είναι ικανός να επιφέρει αλλαγές σε ένα ή περισσότερα αντικείμενα του συστήματος. Ο φορέας της αλλαγής μπορεί να είναι εσωτερικός ή εξωτερικός ανάλογα με το αν ανήκει στο σύστημα που εξετάζουμε ή όχι. Αυτό, βέβαια, ισχύει, πάντα για τις περιπτώσεις στις οποίες γνωρίζουμε το φορέα της αλλαγής.

Πρέπει να τονίσουμε ότι ο φορέας της αλλαγής είναι έννοια αφηρημένη. Δηλαδή, σε πολλές περιπτώσεις συστημάτων ο φορέας δεν ανήκει στο, υπό μελέτη, σύστημα, ενώ άλλες φορές, μας είναι εντελώς άγνωστος. Στην πραγματικότητα εισάγουμε την έννοια του φορέα αλλαγής, για να απαντήσουμε στο ερώτημα «Ποιος έκανε την αλλαγή». Αυτό σημαίνει ότι ο φορέας μπορεί να μην έχει σχέση με την ίδια την αλλαγή ή να υπάρχουν και άλλοι συνυπεύθυνοι φορείς τους οποίους δεν έχουμε καταγράψει και δε γνωρίζουμε. Ακόμη, υπάρχει περίπτωση, ο φορέας να είναι υπεύθυνος με έμμεσο τρόπο για την αλλαγή που εξετάζουμε. Για παράδειγμα, σε ένα πολύπλοκο κοινωνικό σύστημα, ο φορέας της αλλαγής μπορεί να είναι περισσότερα τους ενός διοικητικά στελέχη. Επομένως, ο φορέας είναι αυτός που πιστεύουμε ΕΜΕΙΣ ότι είναι υπεύθυνος για μία αλλαγή και όχι αυτός που μπορεί είναι πραγματικά υπεύθυνος. Η επιλογή γίνεται από τη φύση του προβλήματος που καλούμαστε να λύσουμε. Εδώ αναδύεται η απαίτηση μας για Αντικειμενικότητα του μοντέλου. Ο καταγράφεας της αλλαγής θα πρέπει να είναι παρά πολύ προσεκτικός για να αποφασίσει Ποιος θα καταγραφεί ως φορέας της αλλαγής. Ένα παράδειγμα είναι αυτό με τους ιστορικούς οι οποίοι πρέπει να καταγράφουν τα γεγονότα αντικειμενικά. Παρόλα αυτά, σε πολλά συστήματα, ο φορέας της αλλαγής είναι σαφώς καθορισμένος και μπορούμε να γνωρίζουμε ακριβώς τις πράξεις του και να τις καταγράφουμε, όπως θα δούμε στη συνέχεια. Αυτό είναι εντονότερο ιδίως στα συστήματα που αφορούν πληροφορική και υπολογιστές, όπως το λειτουργικό σύστημα, όπου όχι μόνο μπορούμε να γνωρίζουμε τη ροή των μηνυμάτων που ανταλλάσσονται μεταξύ των δομικών στοιχείων του συστήματος, αλλά μπορούμε να πετύχουμε και αυτοματοποίηση της καταγραφής των ενεργειών του φορέα. Ένα παράδειγμα είναι το αρχείο καταγραφής (logfile) του διαχειριστή (administrator) του συστήματος, στο οποίο φαίνεται καθαρά ποιος έχει μπει στο σύστημα και τι ενέργειες έχει κάνει, όπως επίσης και ποιες ήταν οι επιπτώσεις στην κατάσταση του συστήματος.

Ο φορέας ή οι φορείς της αλλαγής, με την έννοια που συζητήσαμε παραπάνω, εκτελούν κάποιες ενέργειες πάνω στα αντικείμενα. Έτσι, ως αποτέλεσμα αυτών των ενέργειών, θα έχουμε κάποιες πράξεις πάνω στα στοιχεία του συστήματος.

Ο συνδυασμός των συνόλου των ενέργειών του φορέα της αλλαγής, με το σύνολο των πράξεων των στοιχείων του συστήματος αποτελούν τον ορισμό του γεγονότος.

Επομένως, γεγονός είναι μία σχέση «αιτίας-αιτιατού», αλλά όπως είπαμε και στις προηγούμενες παραγράφους αυτή η σχέση μπορεί να είναι εικονική.



Σχήμα 3 Το γεγονός

Ένας φορέας αλλαγής μπορεί να ανήκει στο σύστημα και, επομένως, να είναι ένα αντικείμενο. Τότε θα μιλάμε για την **αλληλεπίδραση** των αντικειμένων του συστήματος. Όταν ο φορέας δεν ανήκει στο σύστημα τότε η ενέργεια του φορέα έρχεται από το εξωτερικό περιβάλλον του συστήματος, επομένως θα έχουμε **εισερχόμενα**. Αντιστοίχως, όταν μία ενέργεια δεν απευθύνεται σε κάποιο αντικείμενο του συστήματος, θα έχουμε **εξερχόμενα**.

Οι **κανόνες** που διέπουν ένα σύστημα, είναι αυτοί που ακολουθούν οι φορείς της αλλαγής για να προκαλέσουν την αλληλεπίδραση των αντικειμένων.

Οι κανόνες που διέπουν ένα σύστημα είναι αυστηρά καθορισμένοι και, υποτίθεται ότι, μπορούν να καλύψουν όλους τους συνδυασμούς καταστάσεων στις οποίες μπορεί να βρεθεί ένα σύστημα.

Οι κανόνες αυτοί μπορεί να μας είναι άγνωστοι, δηλαδή να μην γνωρίζουμε πώς ακριβώς λειτουργεί ένα σύστημα. Αυτό άλλωστε, πολλές φορές, είναι και το ζητούμενο της μελέτης, δηλαδή μπορούμε να χρησιμοποιήσουμε το μοντέλο ανάκτησης δέλτα για να μελετήσουμε τη συμπεριφορά ενός συστήματος και να καταλάβουμε πώς λειτουργεί. Μπορούμε να μελετήσουμε ακριβώς τα αντικείμενα που μας ενδιαφέρουν, αποκλείοντας κάποια άλλα που για την εκάστοτε έρευνά μας δεν μας απασχολούν. Για παράδειγμα, στη μελέτη ενός διοικητικού συστήματος, μπορεί να μας ενδιαφέρει άμεσα μία αλλαγή στη διοικητική πυραμίδα του συστήματος, ωστόσο για κάποιο λόγο, να μην μας απασχολεί μία αλλαγή που συνέβη στο τομέα του Λογισμικού ή του Υλικού, παρόλο που και η πρώτη αλλαγή και η δεύτερη μπορεί να συμβαίνουν ταυτόχρονα.

Από την άλλη μεριά, οι κανόνες που διέπουν ένα σύστημα, μπορεί να μας είναι γνωστοί. Έτσι, πολλές φορές, είναι δυνατό να γνωρίζουμε όχι μόνο τα αντικείμενα και τις τιμές των ιδιοτήτων αυτών, αλλά και τις συναρτήσεις με τις οποίες μεταβάλλεται μία τιμή κάποιας ιδιότητας. Αυτό φαίνεται περισσότερο στα φυσικά συστήματα, όπου μπορούμε να γνωρίζουμε, για παράδειγμα, τις εξισώσεις της κίνησης ενός σώματος ή της επιτάχυνσής του κλπ. Αυτό μας διευκολύνει ιδιαίτερα στον ακριβή υπολογισμό μιας τιμής κάποιας ιδιότητας συναρτήσει του χρόνου και, επομένως, μας βοηθά στην καλύτερη μελέτη του συστήματος μας. Έτσι, μπορεί μία ιδιότητα ενός αντικειμένου να ακολουθεί τις τιμές από μία συνάρτηση. Για παράδειγμα, αν εξετάζουμε το Ηλιακό σύστημα τότε είμαστε σε θέση να γνωρίζουμε την τροχιά του κάθε πλανήτη και επομένως η θέση του κάθε πλανήτη ορίζεται σε σχέση με την τροχιά του, αλλά και σε σχέση με το χρόνο. Περισσότερα όμως γι' αυτό θα συζητήσουμε στη συνέχεια αφού πρώτα εισάγουμε τον ορισμό της αλλαγής και άλλες έννοιες που σχετίζονται με την αλλαγή σε ένα σύστημα.

4.3. Η έννοια της αλλαγής

Ένα σύστημα «δρα», όπως έχουμε αναφέρει, μέσα στο χρόνο. Επομένως συμβαίνουν κάποιες αλλαγές στο σύστημα μας και, μερικές από αυτές, είναι εκείνες που θα πρέπει να καταγράψουμε. Στη συνέχεια θα δώσουμε το σαφή ορισμό μερικών εννοιών που είναι απαραίτητοι για τον ορισμό της αλλαγής οι οποίοι θα μας βοηθήσουν στην περαιτέρω μελέτη του μοντέλου ανάκτησης.

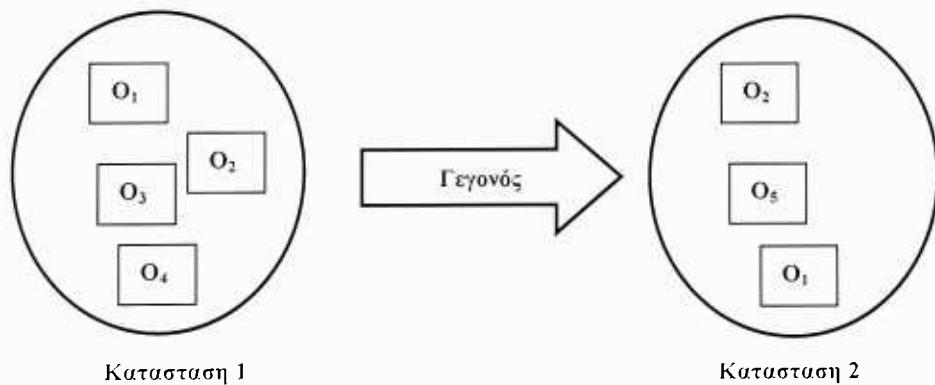
Κατάσταση ενός συστήματος είναι το σύνολο όλων των αντικειμένων και των ιδιοτήτων αυτών, που ανήκουν στο σύστημα.

Φυσικά, όπως είπαμε και παραπάνω, αυτό το σύνολο περιέχει μόνο τα απαραίτητα στοιχεία του συστήματος, εκείνα δηλαδή που αφορούν τη μελέτη μας ή το πρόβλημά μας. Η κατάσταση ενός συστήματος δεν έχει σχέση με το χρόνο. Δηλαδή αυτό το σύνολο των αντικειμένων και ιδιοτήτων είναι ανεξάρτητο από το χρόνο και αυτή είναι η διαφορά της κατάστασης από το στιγμιότυπο του συστήματος, για το στιγμιότυπο θα μιλήσουμε αργότερα.

Η κατάσταση σε ένα σύστημα είναι το κριτήριο για να αποφασίσουμε αν έχει γίνει μία αλλαγή στο σύστημα, όπως το έχουμε ορίσει μέχρι εδώ.

Mία αλλαγή συμβαίνει όταν αλλάζει η κατάσταση ενός συστήματος.

Όταν λέμε, επομένως, αλλαγή στο σύστημα, θα εννοούμε ότι το σύνολο των αντικειμένων και των ιδιοτήτων του συστήματος άλλαξε για κάποιο «λόγο» ή «αιτία». Δηλαδή, όπως αναφέραμε και στην προηγούμενη ενότητα, συνέβησαν κάποια γεγονότα στο σύστημα και, πιο συγκεκριμένα, έγιναν κάποιες ενέργειες από τους φορείς μιας αλλαγής, με αποτέλεσμα να γίνουν κάποιες πράξεις στα επιμέρους στοιχεία του συστήματος. Φυσικά, όπως μπορούμε να αντιληφθούμε, μία αλλαγή έχει άμεση σχέση με το χρόνο. Δηλαδή μία αλλαγή συμβαίνει όσο το σύστημα «δρα» μέσα στο χρόνο.



Σχήμα 4 Η Αλλαγή στο σύστημα

Στην πραγματικότητα, κατά τη διάρκεια μιας αλλαγής μπορεί να συνέβησαν πολλά γεγονότα ταυτόχρονα. Δηλαδή, πολλοί φορείς μπορεί να επηρέασαν πολλά αντικείμενα του συστήματος ταυτόχρονα, και μάλιστα χωρίς να έχει σχέση ο ένας φορέας με τον άλλο και τα αντικείμενα που επηρέασε ο ένας, με αυτά του άλλου. Τότε όμως δεν θα μιλάμε για πολλές αλλαγές. Ήα εξακολουθούμε να μιλάμε για ΜΙΑ αλλαγή, εφόσον μελετάμε το σύστημα ως ολότητα. Έτσι, όταν μιλάμε για «αλλαγή» θα εννοούμε την αλλαγή του συνόλου των αντικειμένων και ιδιοτήτων του συστήματος.

Θα θεωρήσουμε ότι μία αλλαγή συμβαίνει **στιγμιαία**, σε μία χρονική στιγμή, ή αλλιώς σε ένα απειροελάχιστο χρονικό διάστημα dt .

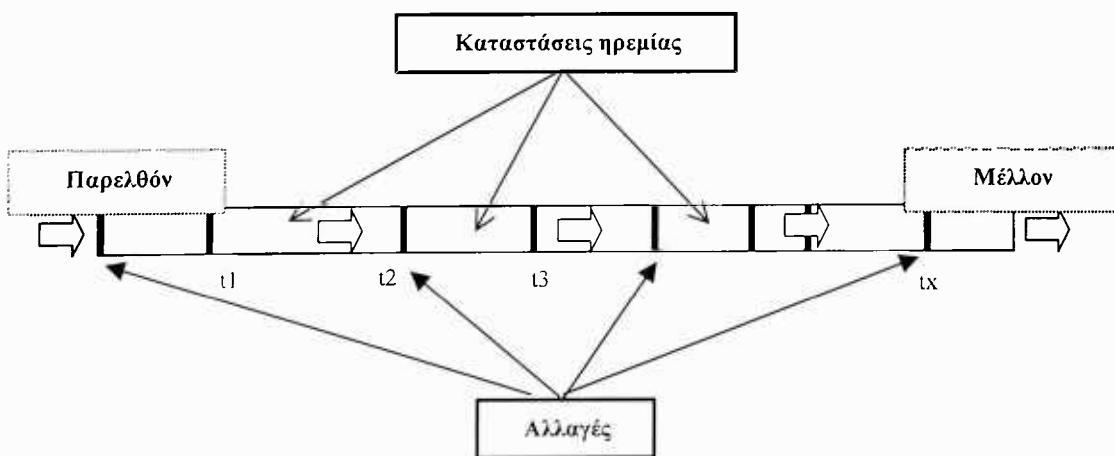
Μας ενδιαφέρει πολύ αυτή η στιγμή... δηλαδή μας ενδιαφέρει η καταγραφή της τιμής, αυτής της χρονικής στιγμής καθώς επίσης και η καταγραφή της κατάστασης του συστήματος. Αυτή τη στιγμή, επειδή ακριβώς θεωρείται αμελητέα, θα την εννοήσουμε ως **διακριτή στιγμή** και όχι έως χρονικό διάστημα.

Η έννοια απειροελάχιστο χρονικό διάστημα dt , φυσικά, είναι σχετική και εξαρτάται από το πόσο «ευαίσθητο» στις αλλαγές είναι το σύστημά μας. Λέγοντας ευαίσθητο, εννοούμε την περίπτωση όπου κάποια ή κάποιες ιδιότητες που ανήκουν στο σύστημά μας, παίρνει τιμές από μία συνεχή συνάρτηση, σε σχέση με το χρόνο. Τέτοιουν είδους συστήματα συναντάμε πολύ συχνά, για παράδειγμα στη φυσική. Ένα τέτοιο παράδειγμα είναι η ελεύθερη πτώση μιας σφαίρας. Σ' αυτή την περίπτωση, ιδιότητες της σφαίρας όπως η κινητική ή η δυναμική ενέργεια παίρνουν τιμές σύμφωνα με συναρτήσεις που εξαρτώνται από το χρόνο. Έτσι, ανάλογα με την κλίμακα όπου θέλουμε να παρατηρήσουμε αυτές τις ιδιότητες της σφαίρας, μπορούμε να ορίσουμε το dt όσο μικρό ή όσο μεγάλο θέλουμε. Επιπλέον, κατά τη χρονική στιγμή dt θα θεωρούμε ότι η κινητική ενέργεια της σφαίρας ή η ταχύτητά της είναι σταθερά.

Από την άλλη μεριά, και εφόσον ισχυριζόμαστε ότι μία αλλαγή συμβαίνει σε μία μετρήσιμη, διακριτή στιγμή, τότε, μέχρι να συμβεί η επόμενη αλλαγή, δηλαδή μέχρι να έρθει η επόμενη διακριτή χρονική στιγμή όπου θα συμβεί επίσης μία διαφοροποίηση στην κατάσταση του συστήματος που μελετάμε, θα έχει μεσολαβήσει ένα χρονικό διάστημα.

Σε αυτό το χρονικό διάστημα, όσο μεγάλο και να είναι αυτό, θα αναγκαστούμε να παραδεχτούμε, εξ ορισμού, ότι δε συμβαίνει αλλαγή στην κατάσταση του συστήματος. Έτσι, ανάμεσα σε δύο στιγμές όπου το σύστημα αλλάζει δύο φορές την κατάστασή του, το σύστημα μας θα θεωρείται «ακινητοποιημένο».

Επομένως, κατάσταση ηρεμίας συστήματος είναι το ανοικτό χρονικό διάστημα ανάμεσα σε δύο αλλαγές του συστήματος ή διαφορετικά, όταν ένα σύστημα δεν αλλάζει θα λέμε ότι βρίσκεται σε κατάσταση ηρεμίας.



Σχήμα 5 Το σύστημα στο χρόνο

Το πιο αντιπροσωπευτικό παράδειγμα συστήματος όπου φαίνεται η διαφορά της κατάστασης ηρεμίας και της χρονικής στιγμής της αλλαγής είναι το ίδιο το ρολόι. Αν θεωρήσουμε ένα ηλεκτρονικό ρολόι ως σύστημα, και εξετάζουμε την τιμή που έχει κάθε φορά ο δείκτης των δευτερολέπτων, τότε, σε κάθε δευτερόλεπτο αυτή η τιμή αλλάζει. Η αλλαγή αυτή γίνεται στιγμιαία. Κατά τη διάρκεια όμως του δευτερόλεπτου, το ρολόι βρίσκεται σε κατάσταση ηρεμίας.

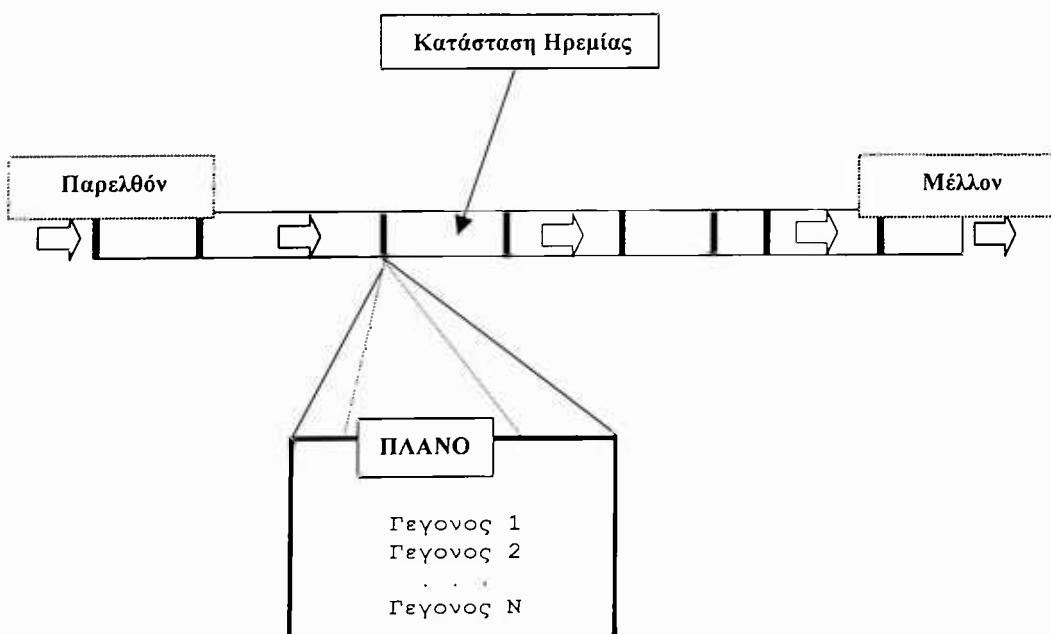
Ένα άλλο θέμα που τίθεται, σύμφωνα με τους ορισμούς που έχουμε δώσει και με όσα έχουμε αναφέρει μέχρι τώρα, είναι ότι ορισμένα γεγονότα μπορεί να συμβαίνουν, κατά μία έννοια, «ταυτόχρονα». Ωστόσο, ακόμα και ταυτόχρονα να συμβαίνουν, πολλές φορές θα μπορούμε να μιλάμε για ακολουθία γεγονότων ιδίως όταν αυτή η ακολουθία αφορά το ίδιο αντικείμενο.

Πιο συγκεκριμένα, για ένα αντικείμενο, κατά τη διάρκεια μιας αλλαγής, μπορούμε να έχουμε πάνω από μία πράξεις που συμβαίνουν στις ιδιότητες του. Υπάρχει περίπτωση, αυτή η σειρά να μην μας ενδιαφέρει ή να είναι ανεξάρτητη από το αποτέλεσμα που θα έχει η εκτέλεση αυτών των πράξεων. Υπάρχει, όμως, και η περίπτωση, η διατεταγμένη αυτή σειρά των πράξεων που συμβαίνουν σε ένα αντικείμενο, να μας ενδιαφέρει.

Για παράδειγμα, στην περίπτωση που το σύστημά μας είναι ένα έγγραφο, όταν γίνει μία αλλαγή στα περιεχόμενα μιας παραγράφου και μετά αυτή η παράγραφος σβηστεί, τότε η σειρά που θα γίνουν αυτές οι δύο πράξεις μας ενδιαφέρει, διότι διαφορετικά θα είχαμε πρόβλημα αν, κατά τη διαδικασία της ανάκτησης, αλλάζαμε το περιεχόμενο σε μία παράγραφο που δεν υπάρχει...

Γι' αυτό το λόγο ορίζουμε την έννοια του πλάνου όπου:

Πλάνο συστήματος είναι η ακολουθία των γεγονότων που συνέβησαν κατά τη διάρκεια μιας αλλαγής στο σύστημα.



Σχήμα 6 Η έννοια του πλάνου

Η έκφραση «κατά τη διάρκεια μιας αλλαγής», εξηγήσαμε πως είναι αρκετά μεταφορική, από την άποψη ότι η αλλαγή στο σύστημα γίνεται στιγμιαία.

Επομένως, τα γεγονότα σε ένα πλάνο του συστήματος θα θεωρούνται *tautóchrona* από άποψη χρόνου, αλλά από άποψη στιγμής, θα θεωρούνται με τη σειρά που αυτά περιγράφονται.

Μέχρι τώρα, είμαστε σε θέση να καταλάβουμε πώς θα διακρίνουμε τις αλλαγές που συμβαίνουν σε ένα σύστημα και, επομένως, να τις καταγράψουμε. Επόμενη έννοια είναι αυτή του στιγμιότυπου η οποία μας βοηθάει να ανακτήσουμε αυτές τις καταγραφές των αλλαγών

Με το όρο στιγμιότυπο συστήματος, θα εννοούμε την κατάσταση στην οποία βρίσκεται ένα σύστημα, τη χρονική στιγμή ΔΤ.

Όπως αναφέραμε και προηγουμένως, ένα στιγμιότυπο είναι στη ουσία η κατάσταση ενός συστήματος, συνδεμένη με το χρόνο, και, για την ακρίβεια, συνδεμένη με ένα απειροελάχιστο διάστημα χρόνου, όπως ορίσαμε το ΔΤ, όπου το σύστημα μπορεί να θεωρηθεί σε κατάσταση ηρεμίας.

Επομένως, λέγοντας στιγμιότυπο, θα αναφερόμαστε και πάλι σε διακριτή, μετρήσιμη χρονική στιγμή.

Ένα στιγμιότυπο μπορεί να τύχει σε χρονική στιγμή όπου το σύστημα βρίσκεται, θεωρητικά πάντα, σε κατάσταση ηρεμίας. Από την άλλη μεριά, ένα στιγμιότυπο μπορεί να συμπέσει με τη χρονική στιγμή της αλλαγής. Τότε, θα κάνουμε την παραδοχή ότι η αλλαγή, έχει ήδη συμβεί.

Επομένως, αν έχουμε καταγράψει ένα πλάνο, δηλαδή μία αλλαγή, για τη χρονική στιγμή t , τότε αν ζητάμε να μάθουμε το στιγμιότυπο αυτή τη χρονική στιγμή t , θα θεωρούμε ότι η αλλαγή στο σύστημα έχει ήδη συμβεί.

Αυτό σημαίνει ότι ανάμεσα σε δύο αλλαγές του συστήματος που συμβαίνουν τις χρονικές στιγμές t_1 και t_2 , το πλάνο της πρώτης αλλαγής εκτελέστηκε τη χρονική στιγμή t_1 , και αυτή η κατάσταση του συστήματος ισχύει κατά τη διάρκεια του διαστήματος $[t_1, t_2]$.

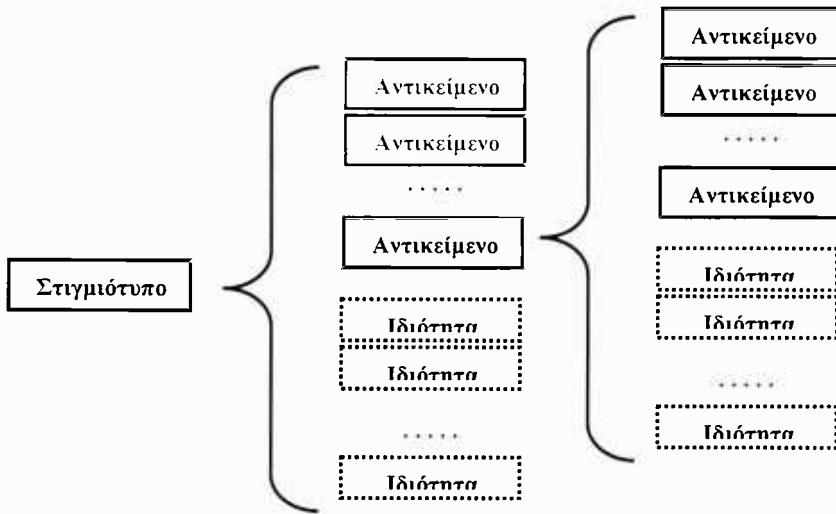
Το στιγμιότυπο, όπως το ορίσαμε παραπάνω, αποτελεί μία από τις πληροφορίες που καλείται να μας δώσει το μοντέλο ανάκτησης. Δηλαδή, σε πολλές περιπτώσεις εφαρμογής του μοντέλου θα χρειαζόμαστε να μάθουμε την κατάσταση που είχε το σύστημα μία δεδομένη χρονική στιγμή t . Το αποτέλεσμα θα είναι να μας δοθεί ένα στιγμιότυπο, όπου και θα μπορούμε να το «διαβάσουμε». Έτσι, επόμενο θέμα για συζήτηση είναι ο τρόπος αναπαράστασης του στιγμιότυπου.

Οι δομές που μπορούμε να χρησιμοποιήσουμε για να αναπαραστήσουμε ένα στιγμιότυπο είναι τρεις:

1. Η σχεσιακή
2. Η δενδρική
3. Η αναπαράσταση ενός γραφήματος.

Θα προτιμήσουμε τη δενδρική αναπαράσταση για τους εξής λόγους: κάθε άλλη δομή μπορεί να μετατραπεί στη δενδρική. Έτσι, σε ορισμένες περιπτώσεις, μία άλλη δομή εκτός της δενδρικής, θα μας βόλευε περισσότερο από άποψη τεκμηρίωσης και ευκολίας διαχείρισης της πληροφορίας. Για παράδειγμα, ένα δίκτυο υπολογιστών

έχει, αποδεδειγμένα, ως καλύτερο τρόπο αναπαράστασης, τη δομή του γραφήματος. Παρόλα αυτά, ένα γράφημα μπορεί να θεωρηθεί ότι είναι ένα δένδρο του οποίου τα φύλλα έχουν κάποιες επιπλέον ιδιότητες (αυτές των δεσμών μεταξύ τους).



Σχήμα 7 Η αναπαράσταση του στιγμιότυπου

Η δενδρική δομή είναι ο πιο διαδεδομένος τρόπος αναπαράστασης διότι βρίσκει εφαρμογή σε ήδη υπάρχουσες τεχνολογίες. (Sgml, Xml). Επιπλέον, η αντικειμενοστραφής σκέψη έχει τη βάση της στη δενδρική δομή και, έτσι, υποστηρίζεται από τις περισσότερες γλώσσες προγραμματισμού.

Το τελευταίο αποτελεί και το κίνητρο για τη χρησιμοποίηση της δενδρικής δομής, εφόσον ένας από τους στόχους που έχουμε θέσει για το μοντέλο ανάκτησης, είναι η ευθυγράμμισή του με τη γλώσσα XML.

Πριν προχωρήσουμε στην αναλυτική περιγραφή του μοντέλου, υπάρχουν ακόμη δύο βασικά θέματα για ανάπτυξη.

4.4. Δύο βασικές Παρατηρήσεις

A) Η ανάπτυξη αυτού του μοντέλου ανάκτησης δέλτα γίνεται διότι μας ενδιαφέρει η μελέτη των στιγμιότυπων και η καταγραφή της διαφοράς μεταξύ τους, και όχι το ίδιο το σύστημα και η λειτουργία του.

Αυτό είναι πολύ σημαντικό για την κατανόηση της λειτουργίας του μοντέλου. Σε αυτό το μοντέλο δεν προτείνεται κάποιος τυπικός τρόπος για την αναπαράσταση και τη λειτουργία ενός συστήματος. Δεν πρόκειται δηλαδή για γλώσσα μοντελοποίησης ενός συστήματος, όπως η UML. Άλλωστε αυτό είναι σε παρά πολλές περιπτώσεις αδύνατο με τα μέσα που διαθέτουμε μέχρι σήμερα. Για παράδειγμα, είναι σχεδόν αδύνατο να μπορέσουμε να μοντελοποιήσουμε ένα πολύπλοκο σύστημα ζωντανού οργανισμού. Είναι αδύνατο να καταγράψουμε κάθε αλληλεπίδραση και κατάσταση κάθε ενός από τα κύτταρα και, γενικά, θα ήταν παρά πολύ επίπονο αυτό το έργο της αναπαράστασης ενός τέτοιου συστήματος που, πολλές φορές, δεν έχουμε ούτε καν την απαραίτητη γνώση ή τα απαραίτητα όργανα μέτρησης για να το πετύχουμε αυτό.

Άλλες φορές είναι αδύνατη η αναπαράσταση εννοιών και διεργασιών που έχουν κάποιο υψηλό επίπεδο αφαίρεσης, όπως ακριβώς συμβαίνει στη μελέτη των κοινωνικών συστημάτων Επιπλέον, μία αλλαγή συνοδεύεται από αντιδράσεις. Δηλαδή, μία αλλαγή είναι δυνατό να προκαλέσει, πάντα σύμφωνα με την εφαρμογή των κανόνων του συστήματος, μία αλυσίδα πράξεων και γεγονότων τα οποία μερικές φορές είναι τόσο πολύπλοκα ώστε είναι αδύνατο, με τα σημερινά μέσα, να μπορέσουμε να τα μετρήσουμε, να τα αναλύσουμε ή να βγάλουμε συμπεράσματα από αυτά.

Έτσι, η αναπαράσταση της λειτουργίας ενός συστήματος δεν είναι θέμα αυτής της εργασίας.

Αυτό που μας ενδιαφέρει είναι η καταγραφή των αλλαγών, δηλαδή των πλάνων ενός συστήματος, το οποίο εξετάζουμε σε συνάρτηση με το χρόνο και η ανάκτηση ή η επαναφορά του, από τις πληροφορίες που έχουμε για τα στιγμιότυπα. Αυτό σημαίνει ότι θα έχουμε μία δομή με αντικείμενα και ιδιότητες αυτών, για τις οποίες θα είμαστε υπεύθυνοι να αποθηκεύσουμε και, αργότερα, να ανακτήσουμε κατά βούληση και επιλεκτικά.

Έτσι, στο μοντέλο περιγράφεται αυτός, ακριβώς, ο τρόπος αναπαράστασης της κατάστασης και της καταγραφής των γεγονότων που διαφοροποιούν την κατάστασή του συστήματος από πλάνο σε πλάνο. Πρόκειται δηλαδή για ένα μηχανισμό πλάνων και στιγμιότυπων και όχι για ένα τρόπο αναπαράστασης του συστήματος και της λειτουργίας του...

B) Στην Ανάλυση του συστήματος παίζει μεγάλο ρόλο η όψη με την οποία βλέπουμε το σύστημά μας.

Ο καθένας, ανάλογα με τις πληροφορίες που έχει, αλλά και με τις απαιτήσεις που έχει συγκεντρώσει από την τοποθέτηση του προβλήματός του, βλέπει το ίδιο σύστημα με διαφορετικό τρόπο. Κατά την ανάλυση του συστήματος χρειάζεται ιδιαίτερη προσοχή, έτσι ώστε να αποσαφηνίσουμε ορισμένα πράγματα σχετικά με το ΤΙ ακριβώς ζητάμε από τη χρήση αυτού του μοντέλου.

Επίσης θα πρέπει, όπως είπαμε και σε προηγούμενες παραγράφους, να οριοθετήσουμε ακριβώς το, υπό μελέτη, σύστημα μας, διότι, ανάλογα με την περίπτωση, υπάρχει διαφορά του ΤΙ θεωρούμε εμείς ως σύστημα από το ΠΟΙΟ ακριβώς είναι το σύστημα.

Ας χρησιμοποιήσουμε το γνωστό παράδειγμα του ρολογιού ως αντικείμενο ανάλυσης. Τότε κάποιος που ενδιαφέρεται για την κίνηση των δεικτών του ρολογιού σε σχέση με το εκκρεμές του ρολογιού, μπορεί να θεωρήσει ότι το σύστημά του αποτελείται από τα αντικείμενα, των δεικτών και του εκκρεμούς. Κάποιος άλλος όμως ενδιαφέρεται για το μηχανισμό του ρολογιού, όποτε θεωρεί ως αντικείμενο μελέτης του, το σύστημα των γραναζιών. Έτσι, ο καθένας έχει ορίσει το δικό του σύστημα προς μελέτη και το έχει ξεχωρίσει από το εξωτερικό περιβάλλον του και τις αλληλεπιδράσεις του με αυτό.

4.5. Το μοντέλο της κινηματογραφικής ταινίας

Ας ξεκινήσουμε από την αρχή ή καλύτερα από την κορυφή επιστρέφοντας στις έννοιες της συστηματικής θεώρησης.

Χωρίς βλάβη της γενικότητας και για τα επόμενα ... χρόνια μπορούμε να θεωρήσουμε το Σύμπαν ως το ανώτερο σύστημα, τη μητρική οντότητα όλων των υπόλοιπων συστημάτων που μελετήσαμε, μελετάμε και θα μελετάμε.

Σε όλες τις περιπτώσεις μπορούμε να θεωρήσουμε το Σύμπαν, ως ένα σύστημα που αποτελεί τη ρίζα όλων των υπόλοιπων γνωστών μέχρι τώρα συστημάτων. Όλα αυτά τα συστήματα «περιέχονται» μέσα στο Σύμπαν και αποτελούν τα επιμέρους τμήματα και τις ιδιότητές του. Όλα αυτά τα αντικείμενα του Σύμπαντος έχουν αλληλεπιδράσεις μεταξύ τους και το καθένα από αυτά έχει υποσυστήματα-αντικείμενα και ιδιότητες. Αυτά τα υποσυστήματα αποτελούνται, με τη σειρά τους, από άλλα υποσυστήματα και αυτό επαναλαμβάνεται πολλές φορές, μέχρι να φτάσουμε σε ένα επιθυμητό επίπεδο λεπτομέρειας το οποίο θα αποτελεί και τη ρίζα του συστήματος για το οποίο ενδιαφερόμαστε.

Πρέπει βέβαια να τονίσουμε ότι υπάρχει δυνατότητα, αυτή η επαγγελματική διεύρυνση των επιμέρους τμημάτων του συστήματος, ξεκινώντας από μία ρίζα, να μην τελειώσει ποτέ. Αυτό όμως δεν αποτελεί πρόβλημα διότι, δεν υπάρχει φυσικός περιορισμός στην ανάλυση του επιπέδου λεπτομέρειας ενός συστήματος, εκτός και αν το επιτρέψουμε εμείς.

Το Σύμπαν «δρα» με παράγοντα το χρόνο. Δηλαδή, με το πέρασμα του χρόνου συμβαίνουν κάποιες αλλαγές και σε κάθε χρονική στιγμή η κατάστασή του αλλάζει. Επομένως, μπορούμε να θεωρήσουμε ότι με κάποια «κοσμική» συσκευή, για παράδειγμα, μία ιδεατή κινηματογραφική μηχανή, μπορούμε να καταγράψουμε τις αλλαγές που συμβαίνουν σε κάθε χρονική στιγμή. Έτσι, θα είμαστε σε θέση να μπορούμε να προβάλουμε αυτές τις καταγραφές της ταινίας σε ένα επίσης ιδεατό τοίχο και να «δούμε» πώς ήταν το σύστημα σε προηγούμενες στιγμές του χρόνου.

Επίσης, αν αυτό «επιτρέπεται» θα μπορούμε επαναφέρουμε ένα σύστημα σε μία κατάσταση που βρισκόταν σε παλαιότερες χρονικές στιγμές.

Όπως ακριβώς συμβαίνει και σε μία κινηματογραφική ταινία, θα μπορούμε να πλοηγηθούμε μέσα σε αυτή την ιδεατή ταινία και να προβάλουμε το σημείο που μας ενδιαφέρει. Έτσι, θα είμαστε σε θέση να απαντήσουμε σε διάφορες ερωτήσεις σχετικά με το «ποιος» έκανε «τι», «που» και «πότε», και έτσι θα μπορούμε να βγάλουμε συμπεράσματα και για το «πώς» και το «γιατί».

Φυσικά, αυτή η ιδεατή κατάσταση με την ιδεατή, κοσμική, μηχανή λήψης δεν μπορεί να υπάρξει, οπότε θα αναγκαστούμε να κάνουμε κάποιους συμβιβασμούς...

Έτσι, για παράδειγμα, όπως είδαμε και στις προηγούμενες παραγράφους, θα μελετάμε ένα σύστημα περιορίζοντας την προσοχή μας στα στοιχεία του συστήματος που μας ενδιαφέρουν την κάθε φορά, και θα δεχτούμε τον τεμαχισμό της συνέχειας του χρόνου σε διακριτά χρονικά διαστήματα, επίσης, ανάλογα με τις ανάγκες μας.

Πριν προχωρήσουμε στην περιγραφή του μοντέλου ας δώσουμε επίσης δύο βασικούς χαρακτηρισμούς για ένα σύστημα.

Ένα σύστημα θα είναι ανακτήσιμο όταν μπορούμε να ανακτήσουμε μία παρελθοντική κατάστασή του, δηλαδή ένα παλαιότερο στιγμιότυπό του, από την καταγραφή των γεγονότων που έχουν συμβεί, μέχρι τη χρονική στιγμή μιας νεότερης κατάστασής του, δηλαδή ενός νεώτερου στιγμιότυπου του.

Ένα σύστημα θα είναι αντιστρέψιμο, όταν είναι ανακτήσιμο και υπάρχει η δυνατότητα να το επαναφέρουμε σε μία παρελθοντική κατάστασή του.

Από τους δύο αυτούς ορισμούς προκύπτουν και οι ορισμοί των αντίστοιχων λειτουργιών «**ανάκτηση κατάστασης**» και «**επαναφορά κατάστασης**». Ήα αναφερόμαστε σ' αυτές απλά ως ανάκτηση και επαναφορά συστήματος, για λόγους απλότητας.

Μία υποπερίπτωση για τη λειτουργία της ανάκτησης συστήματος είναι η «**ανάκτηση δέλτα**» κατά την οποία δεν ανακτούμε τα ίδια τα στιγμιότυπα, αλλά τα γεγονότα που συνέβησαν κατά τη διάρκεια ενός χρονικού διαστήματος.

Οι λειτουργίες αυτές θα είναι θέματα των επόμενων ενοτήτων, κατά την περιγραφή του μοντέλου.

Κεφαλαιο 5

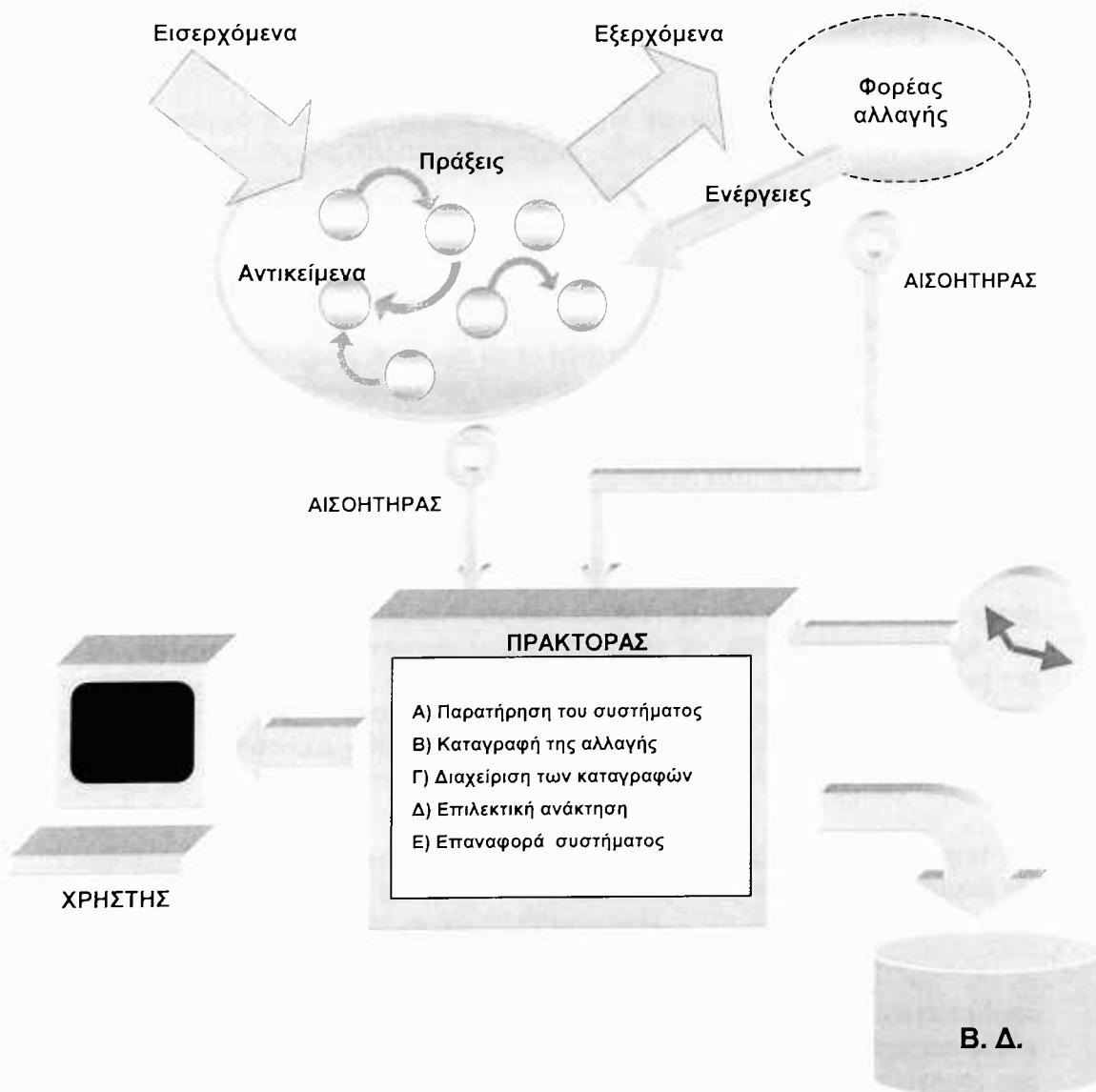
*To μοντέλο
ανάκτησης δέλτα*

5. Το μοντέλο ανάκτησης δέλτα

5.1. Πρόλογος

Το μοντέλο ανάκτησης δέλτα είναι μία μεθοδολογία που θα μας βοηθήσει να πετύχουμε όλα αυτά που έχουμε αναπτύξει μέχρι τώρα. Σύμφωνα με το μοντέλο της κινηματογραφικής ταινίας, ζητάμε έναν μηχανισμό παρατήρησης, καταγραφής, διαχείρισης, ανάκτησης και επαναφοράς των καταστάσεων ενός συστήματος που δρα μέσα στο χρόνο.

Παρακάτω βλέπουμε αυτό το μοντέλο σχηματικά, και αμέσως μετά θα περιγράψουμε τη λειτουργία του



Το κυρίως μέρος του μοντέλου αυτού είναι η έννοια του *Πράκτορα*. Λέγοντας Πράκτορα, θα εννοούμε εκείνο το μηχανισμό που, με μία σειρά από λειτουργίες-διεργασίες, επιτελεί τον προαναφερόμενο σκοπό μας.

Οι λειτουργίες αυτές είναι πέντε:

1. **Η παρατήρηση του συστήματος**
2. **Η καταγραφή της αλλαγής (σε αποθηκευτικό χώρο)**
3. **Η διαχείριση των καταγραφών**
4. **Η (επιλεκτική) ανάκτηση των καταγραφών**
5. **Η πιθανή επαναφορά του συστήματος σε μία παλαιότερη κατάσταση**

Επίσης, υπάρχει ένας αποθηκευτικός χώρος για τις καταγραφές, ένα σύνολο από αισθητήρες εναίσθητους στις αλλαγές και ένα ρολόι για ...να μας λέει την ώρα!

Τέλος υπάρχει και ο Χρήστης δηλαδή ο χειριστής του μηχανισμού αυτού.

Δεν είναι απαραίτητο όλα τα τμήματα του πράκτορα να βρίσκονται μέσα στο ίδιο φυσικό και λογικό χώρο και δεν χρειάζεται να αποτελούν συγκεκριμένη υλοποίηση, από πλευράς εξοπλισμού. Ανάλογα με το σύστημα που εξετάζουμε, ένας πράκτορας μπορεί να είναι, από Λογισμικό και Υλικό, μέχρι Άνθρωπος

Ο τρόπος σκέψης για τη λειτουργία αυτού του μοντέλου είναι ο εξής.

5.2. Η παρατήρηση του συστήματος

Αρχικά χρειαζόμαστε κάποιον να παρατηρεί το σύστημα για το οποίο ενδιαφερόμαστε. Ο παρατηρητής αυτός θα είναι σε «θέση παρατήρησης» στα κατάλληλα σημεία που θα έχουμε ορίσει εμείς εξ' αρχής, και θα μας ειδοποιεί για οτιδήποτε συμβαίνει στο «πεδίο ορατότητάς» του. Έτσι, εισάγουμε την έννοια του *Αισθητήρα* και φυσικά χρησιμοποιούμε εισαγωγικά για τις προηγούμενες εκφράσεις σχετικά με την παρατήρηση διότι, όπως είναι φανερό, ο Αισθητήρας είναι έννοια αφηρημένη.

Όταν λέμε *Αισθητήρας*, επομένως, θα εννοούμε την οντότητα που παρατηρεί και «αισθάνεται» μία αλλαγή, και μας στέλνει κατάλληλο μήνυμα με πληροφορίες για αυτήν την αλλαγή, έτσι ώστε να την αποθηκεύσουμε.

Χωρίζουμε τους αισθητήρες σε δύο κατηγορίες.

Αυτούς που αντιλαμβάνονται τις αλλαγές στην κατάσταση των αντικειμένων (αιτιατό) και αυτούς που αντιλαμβάνονται τα σήματα – ενέργειες του ίδιου του φορέα (αιτία). Έτσι, θα έχουμε τους αισθητήρες *της πράξης* και τους αισθητήρες *της ενέργειας*, αντίστοιχα. Ο τελευταίος παρακολουθεί το φορέα της αλλαγής, ενώ ο πρώτος παρακολουθεί τις πράξεις που συμβαίνουν στα αντικείμενα του συστήματος.

Και στις δύο περιπτώσεις υπάρχει δυνατότητα να μπορούμε να καταγράψουμε το συνολικό γεγονός που συμβαίνει κάθε φορά.

Ο Αισθητήρας της πράξης διαφέρει από τον Αισθητήρα της ενέργειας, στο ότι ο πρώτος, επιβάλλεται να υπάρχει (υποχρεωτικά) ενώ ο δεύτερος όχι. Δηλαδή μπορούμε και να μην καταγράψουμε την ενέργεια όταν αναφερόμαστε σε ένα γεγονός.

Πολλές, ή μάλλον τις περισσότερες, φορές, δεν είναι δυνατό να αναλύσουμε την αιτία ενός γεγονότος. Άλλες πάλι φορές, μπορεί να γνωρίζουμε ακριβώς και με σαφήνεια την αιτία ή τις αιτίες μιας πράξης, όμως μπορεί να μην είναι δυνατό να «τοποθετήσουμε» αισθητήρα πάνω στους φορείς της αλλαγής και να καταγράψουμε τις ενέργειές τους.

Σε άλλες περιπτώσεις μπορεί να μην είμαστε σε θέση να βγάλουμε τα συμπεράσματα που εμείς θέλουμε (π.χ. Ποιος είναι ο πραγματικός υπεύθυνος για την αλλαγή που συνέβη). Αυτό συμβαίνει, για παράδειγμα, στην περίπτωση καταγραφής ενός κοινωνικού συστήματος.

Κάτι τέτοιο θα μπορούσε να επηρεάσει την απαίτηση για αντικειμενικότητα του παρατηρητή και του πράκτορα που θα καταγράφει τις αλλαγές και θα χρειάζεται να βγάλει κάποια συμπεράσματα.

Βέβαια υπάρχουν και περιπτώσεις όπου δεν μας ενδιαφέρει η καταγραφή του φορέα της αλλαγής, όπως, για παράδειγμα, στην καταγραφή των αλλαγών, σε μία κονσόλα ήχου. Εκεί, απλά μας ενδιαφέρουν οι αλλαγές στα ρυθμιστικά και στους διακόπτες της κονσόλας και όχι το ποιος έκανε την αλλαγή.

Ένα άλλο πρόβλημα είναι ότι με τον Αισθητήρα της πράξης μπορεί να έχουμε πλεονασμό στη αποθήκευση της πληροφορίας σε σύγκριση με την αποθήκευση των πληροφοριών που μας δίνει ο φορέας. Για παράδειγμα, μία ενέργεια του φορέα μπορεί να αντιστοιχεί σε πλήθος πράξεων, για τα αντικείμενα. Η δημιουργία μακρό-πράξεων, όπως θα δούμε και παρακάτω, μας βοηθάει σε αυτό το πρόβλημα.

Ανάλογα με το σύστημα που εξετάζουμε, ο Αισθητήρας διαφέρει κάθε φορά. Ακόμα και μέσα στο ίδιο το σύστημα μπορεί να υπάρχουν πολλά είδη αισθητήρων. Τον ρόλο του Αισθητήρα μπορεί να τον παίξει ένα είδος Λογισμικού, ένα ρομπότ, μία ειδική συσκευή, αλλά τον ίδιο το ρόλο μπορεί να τον παίξει και ένας Άνθρωπος ή ένα υβριδικό σύστημα ανθρώπου μηχανής.

Σημαντικό επίσης είναι να ξέρουμε το πού θα τοποθετήσουμε τους αισθητήρες. Αυτό εξαρτάται από την περίπτωση που έχουμε, κάθε φορά, να αντιμετωπίσουμε. Είναι σαν να βάζουμε φίλτρα μπροστά από μία ταινία που προβάλλεται, το καθένα από τα οποία είναι προγραμματισμένο να αφαιρεί πληροφορία από την πρωτότυπη ταινία, και έτσι ο τελικός θεατής να βλέπει μόνο αυτά που τον ενδιαφέρουν. Για παράδειγμα, σε ένα πολύπλοκο Πληροφοριακό σύστημα, όπως αυτό το ορίσαμε στις προηγούμενες παραγράφους, συμβαίνουν συνεχώς αλλαγές στα επιμέρους τμήματά του. Συμβαίνουν αλλαγές στο προσωπικό, στις θέσεις εργασίας, στο Λογισμικό κλπ. Αν όμως δεν μας απασχολεί η αλλαγή στη θέση και στη διοικητική οργάνωση των εργαζόμενων, μπορούμε να μην τοποθετήσουμε αισθητήρες σε αυτήν την περιοχή.

Ένα άλλο θέμα είναι το πόσο ευαίσθητος θα είναι ο Αισθητήρας στις πράξεις που αντιλαμβάνεται. Λέγοντας ευαίσθησία, εννοούμε το επίπεδο λεπτομέρειας, το οποίο θα μπορεί να αναλύσει ο Αισθητήρας που είναι υπεύθυνος για την παρακολούθηση κάποιου αντικειμένου. Ας πάρουμε για παράδειγμα ένα σύστημα επεξεργασίας

κειμένου. Αν υπάρχει ένας Αισθητήρας που παρακολουθεί τις αλλαγές που συμβαίνουν στο κείμενο με ευαισθησία παραγράφου, τότε όταν πληκτρολογούμε μία νέα λέξη σε μία παράγραφο, ο Αισθητήρας δεν θα μας ειδοποιήσει για τυχόν αλλαγή. Ωστόσο αν σβηστεί, ή αν αλλάξει θέση μία παράγραφος τότε θα μας ειδοποιήσει κατάλληλα.

Αυτή η ευαισθησία, επομένως, του Αισθητήρα εξαρτάται από το TI λέμε στον Αισθητήρα να παρακολουθεί και σε ποιο βαθμό λεπτομέρειας.

Μέχρι τώρα αναφέραμε σχετικά με τις αρμοδιότητες του Αισθητήρα, ωστόσο δε μιλήσαμε για το τρόπο και το χρόνο που θα εφαρμόζονται αυτές. Η ερώτηση λοιπόν είναι, κάθε πότε θα μας ειδοποιεί ένας Αισθητήρας για την αλλαγή που συνέβη, δεδομένου ότι σε πολλές περιπτώσεις συστημάτων δεν υπάρχει η δυνατότητα σύλληψης των ενεργειών ενός φορέα.

Με βάση το συλλογισμό αυτό, θα χωρίσουμε τα συστήματα σε δύο κατηγορίες.

Θα τα χωρίσουμε σε αυτά, όπου μπορούμε να καταλάβουμε τα γεγονότα όταν αυτά συμβαίνουν (**eventdriven**), και σ' αυτά που ορίζουμε εμείς κάθε πότε θα ελέγχουμε το σύστημα μας για τυχόν αλλαγές που συνέβησαν (**timedriven**)

Για την πρώτη περίπτωση (**eventdriven**), μας δίνεται η δυνατότητα να ξέρουμε πότε ο φορέας της αλλαγής έκανε μία ενέργεια η οποία είναι υπεύθυνη για κάποιες πράξεις, για τις οποίες ενδιαφερόμαστε να καταγράψουμε. Στην περίπτωση που δεν έχουμε Αισθητήρα ενέργειας, μας δίνεται η δυνατότητα να αντιλαμβανόμαστε κάθε πότε άλλαξε η κατάσταση σε ένα σύστημα. Έτσι, ο πράκτορας καταγράφει την ώρα και το γεγονός, κάθε φορά που ειδοποιείται για την ύπαρξη μιας αλλαγής.

Τα πιο συχνά παραδείγματα για την έννοια **eventdriven**, είναι αυτά που αφορούν τους υπολογιστές, όπως για παράδειγμα τα λειτουργικά συστήματα ή τα συστήματα αρχείων και τα προγράμματα εφαρμογής (**applications**).

Ένα άλλο θέμα για αυτή την περίπτωση είναι η ομαδοποίηση των πράξεων σε μικρότερο επίπεδο λεπτομέρειας από αυτό που ανήκουν. Πάλι εδώ παίζει ρόλο η ευαισθησία του Αισθητήρα. Έτσι, στο γνωστό παράδειγμα της επεξεργασίας κειμένου, μπορεί ένας αισθητήρας που είναι ευαίσθητος σε επίπεδο χαρακτήρων να ομαδοποιεί τις πράξεις σε επίπεδο λέξης. Με αυτό τον τρόπο, ο Αισθητήρας θα ειδοποιεί τον πράκτορα κάθε φορά που αλλάζει μία λέξη (πρόσθεση, αφαίρεση, αλλαγή) και όχι κάθε φορά που γίνεται μία πράξη πάνω στους χαρακτήρες του κειμένου.

Στη δεύτερη περίπτωση (**timedriven**), δεν μας δίνεται η δυνατότητα να ξέρουμε πότε έγινε μία αλλαγή στο σύστημα. Σ' αυτές τις περιπτώσεις, συνήθως, δεν μας ενδιαφέρει, από άποψη μελέτης, ο φορέας μιας αλλαγής. Τότε είμαστε αναγκασμένοι να ελέγχουμε το σύστημα κατά τακτά χρονικά διαστήματα, να συγκρίνουμε την τρέχουσα κατάσταση με αυτή που έχουμε κρατήσει από την τελευταία καταγραφή, και αν έχει υπάρξει κάποια διαφοροποίηση, τότε μόνο να ειδοποιούμε τον πράκτορα για την αλλαγή που συνέβη.

Εδώ, τίθεται το πρόβλημα του καθορισμού του χρονικού διαστήματος που θα μεσολαβεί ανάμεσα σε δύο συγκρίσεις. Το διάστημα αυτό θα το ονομάζουμε **interval**. Έτσι, είναι φανερό ότι πριν το καθορισμό αυτής της παραμέτρου, θα πρέπει να έχει προηγηθεί από τον αναλυτή συστήματος, μία προσεκτική εξέταση της λειτουργίας του συστήματος και μία μελέτη του πεδίου τιμών αυτής της παραμέτρου,

σχετικά με τις απαιτήσεις μας για λεπτομέρεια, ως αναφορά στη διάσταση του χρόνου. Για παράδειγμα, αν εξετάζουμε την τροχιά ενός κομήτη θα μιλάμε για τάξεις μεγέθους αιώνων, ενώ αν εξετάζουμε τη δοκιμαστική κρούση ενός αυτοκίνητου, θα μιλάμε για τάξεις μεγέθους νανοδευτερολέπτων.

Πολλά συστήματα μπορεί να έχουν έναν *υβριδικό* χαρακτήρα σε σχέση με το κριτήριο διαχωρισμού που κάναμε παραπάνω (*timedriven*, *eventdriven*). Δηλαδή μπορεί ένας πράκτορας (ένα σύστημα ασφάλειας, για παράδειγμα), να ελέγχει για τυχόν αλλαγές στην κατάσταση του συστήματος, ανά τακτά χρονικά διαστήματα, αλλά, και κάθε φορά που συμβαίνει μία ενέργεια, σε κάποιον από τους αισθητήρες του μοντέλου (στο παράδειγμα μας, σε μια παραβίαση ιδιοκτησίας).

Ένα άλλο θέμα που τίθεται, είναι αυτό της αποθήκευσης, στη βάση μας, πράξεων που έχουν γίνει την ίδια χρονική στιγμή. Σ' αυτή την περίπτωση, μπορούμε να διαχωρίσουμε τα συστήματα με βάση τη θεώρηση της ταυτόχρονης πράξης, ως εξής.

- Επιτρέπονται ταυτόχρονες πράξεις για το ίδιο αντικείμενο
- Επιτρέπονται ταυτόχρονες πράξεις για διαφορετικά αντικείμενα

Οι συνδυασμοί αυτών των δύο περιπτώσεων (συνολικά τέσσερις) μας δίνουν ένα σύνολο από επιμέρους τύπους συστημάτων τα οποία θα πρέπει να αντιμετωπίσουμε κατάλληλα, ανάλογα με τις απαιτήσεις μας.

Φυσικά, δεν πρέπει να αγνοούμε το γεγονός ότι αν αναφερόμαστε σε σύστημα πληροφορικής τότε ο πράκτοράς μας θα είναι ένας υπολογιστής, ο οποίος θα καταγράφει τα γεγονότα. Οι υπολογιστές, όμως, με τη σημερινή τεχνολογία που ακολουθούν, είναι κατά κανόνα σειριακοί, πράγμα που δεν επιτρέπει την ταυτόχρονη αντίληψη των γεγονότων από έναν αισθητήρα. Έτσι, αν καταγράφουμε, τυπικά, τον πραγματικό χρόνο που καταφθάνουν τα γεγονότα από τον Αισθητήρα στον πράκτορα, κάθε χρονική στιγμή θα διαφέρει από την προηγούμενη της κατά ένα ελάχιστο, αλλά όχι πάντα αμελητέο, χρονικό διάστημα. Παρόλα αυτά, ο χρόνος καταγραφής των γεγονότων μπορεί να είναι ακριβής, εφόσον έχουμε ορίσει ένα **κατώφλι διαστήματος χρόνου (threshold)**. Αν η διαφορά μεταξύ δύο στιγμών που μελετάμε είναι μικρότερη από αυτή του κατώφλιού, τότε μπορούν αυτές οι στιγμές να θεωρηθούν ίσες και να στρογγυλοποιηθούν με βάση κάποια συνάρτηση. Για παράδειγμα και οι δύο στιγμές θα αποκτήσουν την τιμή της πρώτης, ή θα αποκτήσουν την τιμή της δεύτερης ή θα αποκτήσουν την τιμή του μέσου όρου των δύο στιγμών. Θα θεωρήσουμε όμως, ότι αυτές οι δύο στιγμές θα αποκτήσουν την τιμή της πρώτης, και αυτό ακριβώς πετυχαίνει και η έννοια του πλάνου για την οποία μιλήσαμε σε προηγούμενο κεφάλαιο. Μία άλλη λύση θα ήταν να στρογγυλοποιηθούν και οι δύο ως προς κάποια, καλά καθορισμένα ή διακριτά, σημεία του χρόνου (**quantize**). Για παράδειγμα μπορούμε να ορίσουμε διακριτά σημεία στο χρόνο, κάθε 1/16 του δευτερόλεπτου και έτσι όλες οι στρογγυλοποιήσεις να γίνονται με βάση αυτή την κλίμακα, δηλαδή κάθε καταγεγραμμένη χρονική στιγμή να στρογγυλοποιείται στο πλησιέστερο διακριτό σημείο αυτής της κλίμακας.

Με όλα αυτά, θα πρέπει να λάβουμε υπόψη ότι ο χρόνος αντίληψης των γεγονότων είναι σχετικός και δεν ανταποκρίνεται στην πραγματική του τιμή, παρόλο που ο χρόνος καταγραφής του γεγονότος στο αποθηκευτικό μέσο μπορεί να θεωρηθεί αμελητέος. Έτσι, σε συστήματα κρίσιμων ενεργειών, θα πρέπει να λάβουμε υπόψη αυτό το φαινόμενο και θα πρέπει να έχουμε κάνει από πριν μία ανάλυση σχετικά με

τις έννοιες του κατωφλίου και της στρογγυλοποίησης, καθώς και του μέγιστου χρόνου ανταπόκρισης του υπολογιστικού μας συστήματος.

Υπάρχουν περιπτώσεις όπου οι ενέργειες δεν μπορούν να γίνουν ταυτόχρονα ή που δεν θέλουμε να γίνονται ταυτόχρονα. Στην αντίθετη περίπτωση, μπορούμε να επιτρέψουμε την καταγραφή και τη διαχείριση ενεργειών που έχουν γίνει την ίδια χρονική στιγμή.

Με βάση τα παραπάνω, ορίζουμε ως **chronon** το μικρότερο χρονικό διάστημα που μπορεί να διαχειριστεί ο πράκτορας μας. Ο όρος αυτός είναι δανεισμένος από τη θεωρία χρονικών βάσεων δεδομένων και είναι χρήσιμος στην περίπτωση που η βάση υλοποίησης του μοντέλου μας είναι μία τέτοια βάση.

5.3. Η καταγραφή των αλλαγών

Το επόμενο βήμα μετά από τη σύλληψη των αλλαγών, είναι η καταγραφή τους σε έναν αποθηκευτικό χώρο. Αυτός ο χώρος μπορεί να είναι οτιδήποτε μπορεί να κρατήσει τις πληροφορίες μας, ακολουθώντας το μοντέλο της κινηματογραφικής ταινίας για το οποίο μιλήσαμε σε προηγούμενες παραγράφους. Από εδώ και στο εξής θα αναφερόμαστε σε αυτόν το χώρο ως **αρχείο καταγραφής**.

Το μοντέλο της κινηματογραφικής ταινίας, υπονοεί ότι ο τρόπος προσπέλασης του μέσου θα πρέπει να είναι σειριακός χωρίς όμως αυτό να σημαίνει ότι τρόπος αποθήκευσης θα πρέπει να είναι, επίσης, σειριακός.

Πιο συγκεκριμένα, ο στόχος μας είναι να αποθηκεύουμε τα πλάνα ενός συστήματος κατά τη λειτουργία του. Από τον ορισμό που έχουμε δώσει σε προηγούμενες ενότητες, ένα πλάνο θα έχει μία χρονική στιγμή και μία ακολουθία γεγονότων τα οποία συνέβησαν εκείνη τη χρονική στιγμή. Τα πλάνα αυτά μπορούν να βρίσκονται με τυχαία σειρά στο μέσο αποθήκευσης που διαθέτουμε. Ωστόσο, για να πετύχουμε ανάκτηση ή επαναφορά του συστήματος θα πρέπει να ακολουθήσουμε τη χρονικά διατεταγμένη σειρά των πλάνων. Γι' αυτό το θέμα θα μιλήσουμε αναλυτικά στη λειτουργία της ανάκτησης και επαναφοράς.

Είπαμε ότι σε ένα πλάνο υπάρχουν ακολουθίες από γεγονότα. Επιπλέον, σε ένα πλάνο, θα χρειάζεται να δηλώνουμε και την κατάσταση των ίδιων των αντικειμένων, δηλαδή, κατά κάποιο τρόπο, θα αποθηκεύουμε και τα στιγμιότυπα των αντικειμένων. Ο λόγος που γίνεται αυτό είναι διότι η ίδια η κατάσταση ενός αντικειμένου χρειάζεται για την αντιστροφή των πράξεων, κατά την ανάκτηση ή την αντιστροφή του συστήματος. Τέλος, σε ένα Πλάνο θα υπάρχουν δηλώσεις σχετικά με το ίδιο το σύστημα, με τους φορείς των αλλαγών, με τις ενέργειες καθώς και άλλες δηλώσεις λειτουργικού σκοπού όπως μακροεντολές και συναρτήσεις. Ας δούμε όμως πρώτα πώς θα αναπαριστάνουμε την κατάσταση ενός αντικειμένου.

Όπως αναλύσαμε σε προηγούμενες ενότητες, ένα αντικείμενο αποτελείται από ιδιότητες και πιθανόν από άλλα αντικείμενα με αποτέλεσμα να δημιουργείται ένα δένδρο το οποίο αναπαριστά την κατάσταση του αντικειμένου. Αν θεωρήσουμε και το ριζικό σύστημα ως αντικείμενο, τότε η αναπαράσταση αυτού θα αποτελεί ένα στιγμιότυπο.

Έτσι, ένα στιγμιότυπο ενός συστήματος θα δηλωθεί με τον εξής τρόπο :

```
Snapshot { Property+ Object* }
Object { Property+ Object* }
```

Ας εξηγήσουμε τους παραπάνω συμβολισμούς.

Κατ' αρχήν για να ονομάζουμε τις οντότητες, θα χρησιμοποιούμε λεκτικά με λατινικούς χαρακτήρες. Αυτό το κάνουμε διότι τα λεκτικά – έναντι των συμβόλων, είναι πιο εύκολα αναγνώσιμα, και έτσι πιο εύκολα κατανοητά.

Έτσι, στις παραπάνω δηλώσεις θα έχουμε τις δύο οντότητες, του στιγμιότυπου και του αντικειμένου.

Δεύτερη παρατήρηση είναι τα άγκιστρα «{ » , « } » που υπάρχουν μετά από τις δύο οντότητες που δηλώθηκαν. Αυτά χρησιμοποιούνται για να δείξουν τη σχέση κατοχής (has). Έτσι, θα υπάρχουν σχέσεις Προγόνου – Απόγονου χωρίς όμως να συγχέεται αυτή η ορολογία, με αυτή της αντικειμενοστραφούς προσέγγισης.

Τρίτη παρατήρηση είναι τα σύμβολα + και * που υπάρχουν δίπλα από κάθε οντότητα. - απόγονο. Αυτά δηλώνουν τη συμμετοχή της κάθε οντότητας στην προγονική - οντότητα και τα συγκεκριμένα σύμβολα είναι τα ίδια με αυτά που χρησιμοποιούνται και στις γλώσσες SGML και XML. Ο λόγος που γίνεται αυτό είναι, για να υπάρχει εύκολη ευθυγράμμιση της σημειογραφίας με αυτή της XML όπως θα δούμε σε επόμενες ενότητες. Το σύμβολο * σημαίνει «μηδέν ή περισσότερα» και το σύμβολο + σημαίνει «ένα ή περισσότερα». Το πλήρες σύνολο των συμβόλων για αυτήν τη σημειογραφία είναι το παρακάτω

,	: Διατεταγμένη σειρά
?	: Προαιρετικό
#	: Απαριτούμενο
+	: ένα ή περισσότερα
*	: μηδέν ή περισσότερα
	: επιλογή (διάζευξη)
()	: Ομαδοποίηση
< >	: αναφορά σε οντότητα
&	: Λίστα ιδιοτήτων

Από εδώ και στο εξής θα χρησιμοποιούμε τον πίνακα αυτόν, ως αναφορά.

Βλέπουμε λοιπόν, στις παραπάνω δηλώσεις, ότι ένα στιγμιότυπο έχει τουλάχιστο μία ιδιότητα και, προαιρετικά, ένα ή περισσότερα αντικείμενα. Αυτό θα πει ότι μπορεί να υπάρχει και στιγμιότυπο με μία μόνο ιδιότητα και τίποτα περισσότερο. Δηλαδή, κατά τη μελέτη του συστήματος, μπορεί να ενδιαφερόμαστε για την τιμή μίας μόνο

ιδιότητας και έτσι να καταγράφουμε μόνο τις αλλαγές αυτής, παρόλο που το σύστημά μας μπορεί να αποτελείται και από άλλες ιδιότητες και αντικείμενα.

Το ίδιο ακριβώς συμβαίνει και με το αντικείμενο που, στην πραγματικότητα, αποτελεί ένα υποσύνολο του στιγμιότυπου, και έτσι, είναι ακριβώς του ίδιου τύπου με αυτό. Με βάση αυτήν τη σκέψη, ένα στιγμιότυπο θα μπορούσε να δηλωθεί ως αντικείμενο. Ο λόγος που γίνεται διαχωρισμός μεταξύ των δύο οντοτήτων είναι για τεκμηρίωση και μόνο.

Η οντότητα της *Ιδιότητας* (property), αποτελείται από ένα όνομα και μία τιμή αντιστοιχισμένη σε αυτό το όνομα. Θα δηλώνεται ως εξής:

```
Property : (PropertyName = value)
value : ( id | number | text | bin | nil )
```

Το όνομα της ιδιότητας αποτελεί και το μοναδικό αναγνωριστικό για την αναφορά μας σε αυτήν, σε συνδυασμό με τη θέση της στο δένδρο αναπαράστασης, όπως θα δούμε παρακάτω.

Η τιμή μιας ιδιότητας, από ότι βλέπουμε, μπορεί να πάρει έναν από τους βασικούς τύπους δεδομένων που φαίνονται στην παραπάνω δήλωση. Οι τύποι αυτοί των δεδομένων περιέχουν ένα υψηλό επίπεδο αφαιρέσης και εξαρτώνται από τη γλώσσα υλοποίησης του μοντέλου. Έτσι, όταν λέμε ότι ο τύπος δεδομένων είναι *number* θα εννοούμε όλους τους τύπους δεδομένων που χρησιμοποιεί η συγκεκριμένη υλοποίηση του μοντέλου για να αναπαριστά τους αριθμούς. Δηλαδή θα μπορεί να είναι από ακέραιος μέχρι πραγματικός κινητής υποδιαστολής κ.ο.κ.

Ας δούμε αναλυτικά αυτούς τους τύπους

- **Number** : περιλαμβάνει όλους τους αριθμητικούς τύπους
- **Text** : περιλαμβάνει όλους τους τύπους αλφαριθμητικών συμπεριλαμβανόμενου και του τύπου **χαρακτήρα**.
- **Bin** : περιλαμβάνει όλους τους τύπους δυαδικών δεδομένων. Χρησιμοποιείται για την αναπαράσταση πολυμεσικών δεδομένων και άλλων δεδομένων ψηφιακής πληροφορίας.
- **ID** : αυτός είναι ένας πολύ σημαντικός τύπος δεδομένων και περιλαμβάνει όλους τους τύπους αναφορών, μιας γλώσσας υλοποίησης. Λέγοντας «αναφορά» θα εννοούμε, κατά μία έννοια, μία «διεύθυνση» από όπου θα μπορούμε να βρίσκουμε την αντίστοιχη τιμή μιας μεταβλητής, η οποία βρίσκεται εκτός της περιοχής της δήλωσης. Αυτή, θα μπορούσε να είναι μία αναφορά σε ένα αντικείμενο που βρίσκεται κάπου μέσα στο αρχείο καταγραφής που έχουμε κρατήσει. Θα μπορούσε να είναι η αναφορά σε μία ιστοσελίδα ή σε ένα αρχείο ενός απομακρυσμένου δίσκου, ή θα μπορούσε να είναι ένα αντικείμενο CORBA/DCOM ή, τελικά, οτιδήποτε μπορούμε να φανταστούμε και μπορούμε να αναφερθούμε μοναδικά σε αυτό. Η μοναδικότητα του ID παίζει μεγάλο ρόλο στο «πεδίο ορατότητας» της εφαρμογής μας. Για παράδειγμα αν η εφαρμογή μας απευθύνεται στο Διαδίκτυο, ένας διαισθητικός τρόπος για την αναπαράσταση του ID θα ήταν ένα URL ή ένα URI. Ωστόσο, επειδή ο τύπος του ID χρησιμοποιείται

για την αναφορά σε αντικείμενα και δηλώσεις εντός του αρχείου καταγραφής, θα πρέπει να είμαστε πολύ προσεκτικοί ως προς τις συμβάσεις που θα κάνουμε στο θέμα αυτό.

- *Nil* : αυτός ο τύπος αναφέρεται σε μη αντιστοιχισμένες οντότητες ή σε απροσδιόριστου τύπου μεταβλητές και ακολουθεί τη σημασιολογία μιας αντίστοιχης δήλωσης στις γλώσσες υψηλού επιπέδου που υπάρχουν σήμερα.

Τα στιγμιότυπα και τα αντικείμενα, όπως ήδη αναφέραμε, δηλώνονται μέσα στο αρχείο καταγραφής και συγκεκριμένα μέσα σε κάθε πλάνο. Εκτός όμως από αυτά, έχουμε και άλλες δηλώσεις που αφορούν ένα πλάνο, καθώς επίσης και πληροφορίες συστήματος και, επιπλέον, έχουμε και απλά σχόλια. Ας δούμε μερικά από αυτά. :

```
Actor { Property+ }
Action { Property+ }
Declaration { Property+ }
Comment { (id | text)+ }
```

Έτσι, βλέπουμε ότι θα έχουμε τη δήλωση των φορέων της αλλαγής καθώς και τις ενέργειές τους. Εδώ, καθεμία από τις δύο αυτές οντότητες θα μπορεί να έχει ένα σύνολο από ιδιότητες οι οποίες ορίζονται κατά βούληση για τις ανάγκες τις εκάστοτε εφαρμογής. Τέτοιες ιδιότητες θα μπορεί να είναι, για παράδειγμα, το όνομά τους, κάποιες μετρικές, κάποιες μεταβλητές, ή οτιδήποτε άλλο μας χρειάζεται. Βλέπουμε επίσης ότι έχουν υποχρεωτικά, τουλάχιστο, μία ιδιότητα, και αυτή είναι η ιδιότητα ID η οποία είναι απαραίτητη για την αναφορά σε αυτούς τους φορείς ή τις ενέργειες από αλλά σημεία του αρχείου καταγραφής ή ακόμα και από εξωτερικά αρχεία καταγραφής, ανάλογα με την περίπτωσή μας. Για τις υποχρεωτικές και προκαθορισμένες ιδιότητες των οντοτήτων θα μιλήσουμε παρακάτω. Όπως έχουμε πει και σε προηγούμενες παραγράφους, οι έννοιες των φορέων και των ενεργειών είναι αφηρημένες, ορίζονται υποκειμενικά από το χρήστη και μπορούν να είναι προαιρετικές για το σύστημα μας.

Η οντότητα *Declaration* χρησιμοποιείται για να δηλώσουμε κάτι που θέλουμε. Αυτή η δήλωση θα μπορεί να απευθύνεται στο parser του συστήματος μας για να τον ειδοποιήσει για μία παράμετρο που είναι αναγκαία για την εγκυρότητα του συστήματος. Αποτελείται από κάποιες προκαθορισμένες ιδιότητες οι οποίες είναι, ο τύπος του συστήματος, ο τύπος του χρόνου που καταγράφουμε και άλλες χρήσιμες πληροφορίες οι οποίες θα αναφερθούν αργότερα. Επίσης, οι δηλώσεις μπορούν να απευθύνονται και στο χρήστη, δίνοντάς του τη δυνατότητα να ορίσει δικές του ιδιότητες - δηλώσεις ανάλογα με τις ανάγκες και τις απαιτήσεις του.

Η οντότητα *Comment* χρησιμοποιείται για να γράφουμε σχόλια μέσα στο αρχείο καταγραφής και παίζουν το ρόλο της τεκμηρίωσης των δηλώσεων ή άλλων πληροφοριών. Δεν υπάρχει κάτι το ιδιαίτερο σε αυτή την οντότητα εκτός από το γεγονός ότι τα σχόλια, εκτός από κείμενο, θα μπορούν να περιέχουν και ID αναφορές σε άλλους πόρους του συστήματος μας.

Τέλος, ένα πλάνο πρέπει να περιέχει μία οντότητα χρόνου η οποία το ορίζει μοναδικά μέσα στο αρχείο καταγραφής και αποτελεί τον «αύξοντα αριθμό ταυτότητας» του κάθε πλάνου. Έτσι, θα έχουμε τη δήλωση.

Time { Property+ }

Όπως βλέπουμε, η οντότητα του χρόνου έχει μία προκαθορισμένη ιδιότητα η οποία είναι η διακριτή τιμή του χρόνου που αναφερόμαστε. Ωστόσο, η οντότητα του χρόνου μπορεί να πάρει και άλλες ιδιότητες καθορισμένες από το Χρήστη.

Πριν προχωρήσουμε στη δήλωση της οντότητας του πλάνου, ας δούμε πρώτα αυτή του γεγονότος.

Fact { <Actor*> <Action*> Operation* <Macro*> }

Οι χαρακτήρες <> που περικλείουν μία οντότητα σημαίνουν ότι η συγκεκριμένη δήλωση δεν είναι δήλωση οντότητας, αλλά μία αναφορά σε αυτή. Έτσι, όταν γράφουμε <actor> θα εννοούμε μία καταγραφή ενός ID τύπου, η οποία αναφέρεται σε μια οντότητα Actor. Έτσι, θα αναφερόμαστε σε κάποια μοναδική οντότητα actor που θα είναι δηλωμένη κάπου μέσα στο αρχείο καταγραφής ή σε ένα οποιοδήποτε άλλο εξωτερικό αρχείο.

Με βάση αυτό, βλέπουμε ότι μία οντότητα γεγονότος, θα έχει αναφορές σε έναν οι περισσότερους φορείς, όπως επίσης και σε μία ή περισσότερες ενέργειες. Αυτό μας δίνει την ελευθερία να δηλώσουμε περισσότερες από μία «αιτίες» για ένα γεγονός, με οποιεσδήποτε συνέπειες μπορεί να έχει αυτό για τη διεξαγωγή συμπερασμάτων, αργότερα, κατά τη διαδικασία της ανάκτησης.

Επόμενες δηλώσεις, για ένα γεγονός, είναι αυτή της πράξης και αυτή της Μακροπράξης για τις οποίες θα μιλήσουμε αμέσως μετά.

Όπως είχαμε αναφέρει κατά την Ανάλυση του Μοντέλου, οι πράξεις συμβαίνουν στα αντικείμενα και στις ιδιότητες μιας αλλαγής κατά τη στιγμή της αλλαγής. Επίσης μιλήσαμε για ακολουθία πράξεων που συμβαίνουν κατά τη στιγμή της αλλαγής και έτσι ορίσαμε την έννοια του πλάνου.

Μπορούμε να πούμε ότι υπάρχουν πράξεις πολλών ειδών, αν αυτές τις δούμε από τη λογική όψη. Απ' τη φυσική όψη όμως, στην πραγματικότητα όλες οι πράξεις αναλύονται σε μία σειρά από στοιχειώδεις πράξεις που είναι και αυτές τις οποίες «καταλαβαίνει» το σύστημα. Έτσι, εισάγουμε την έννοια της μακροπράξης η οποία στην ουσία είναι μία σειρά από στοιχειώδεις πράξεις. Έτσι, ένα πλάνο θα μπορεί να περιέχει και μακροπράξεις αλλά και στοιχειώδεις πράξεις

Οι βασικές πράξεις που μπορούν να γίνουν στα αντικείμενα είναι οι εξής

Για τα ίδια τα αντικείμενα είναι:

```
Addobject (ParentPosition, <Object>)
Deleteobject (Position, <Object>)
```

Και για τις ιδιότητες των αντικειμένων, θα είναι:

```
Addproperty (Position, Propertynname, value)
DeleteProperty (Position, Propertynname, value)
ChangeProperty (Position, Propertynname, value,
oldvalue)
```

Μία αρχική παρατήρηση εδώ, είναι ότι δεν υπάρχει η πράξη *updateobject* διότι αυτή μεταφράζεται ως σύνολο των εντολών των ιδιοτήτων του αντικειμένου. Δηλαδή, οι βασικές αλλαγές που μπορούν να υπάρξουν για το αντικείμενο, είναι αυτές της εισαγωγής και της διαγραφής κάποιων αντικειμένων απόγονων καθώς και η αλλαγή (εισαγωγή διαγραφή και ανανέωση) στο σύνολο των ιδιοτήτων του.

Επειδή ο ορισμός του αντικειμένου έχει επαγωγικό χαρακτήρα, με αυτές τις πράξεις μπορούμε να αλλάξουμε οτιδήποτε θέλουμε από το αντικείμενο και σε οποιοδήποτε επίπεδο λεπτομέρειας θέλουμε.

Οι πράξεις της εισαγωγής και της διαγραφής ιδιοτήτων κατανοούνται καλύτερα αν θεωρήσουμε το παράδειγμα της επεξεργασίας κειμένου. Εκεί, ένα αντικείμενο παραγράφου, για παράδειγμα, θα μπορεί να αποκτήσει μία ιδιότητα **BOLD** για το κείμενο που περιέχει, ή θα μπορεί να χάσει την ιδιότητα της αρίθμησης κ.ο.κ

Επίσης, οι πράξεις της εισαγωγής και της διαγραφής ιδιοτήτων μας δίνουν την ελευθερία να διαχειριστούμε ιδιότητες των οποίων μπορεί να μην είχαμε προβλέψει την ύπαρξη, από την αρχική ανάλυση του συστήματος.

Ο τύπος *position*, καθώς και όλοι οι υπόλοιποι, που βρίσκονται εντός των παραμέτρων αυτών των στοιχειωδών πράξεων φαίνονται παρακάτω

```
Position : ObjectPath
oldvalue : value
ParentPosition : Position
```

Βλέπουμε δηλαδή ότι ο τύπος *position* είναι ένα *μονοπάτι αντικειμένου*. Λέγοντας *μονοπάτι αντικειμένου* θα εννοούμε τον τρόπο με τον οποίο αναφερόμαστε στα αντικείμενα – απόγονους, ενός αντικειμένου - πατέρα. Χρησιμοποιείται όπως ακριβώς και το *μονοπάτι* σε ένα *σύστημα αρχείων*, ωστόσο δεν υπάρχει τυπική *σύνταξη* στον ορισμό του διότι ο Χρήστης θα πρέπει να είναι ελεύθερος να υλοποιήσει αυτή τη σύνταξη, ανάλογα με τις προδιαγραφές του συστήματός του. Πιο

συγκεκριμένα, ας θεωρήσουμε ότι θα χρησιμοποιήσουμε το διαχωριστή «/» για τον ορισμό του μονοπατιού μας. Τότε, αν υποθέσουμε ότι το αντικείμενο-πρόγονος είναι το ίδιο το στιγμιότυπο, για να αναφερθούμε σε ένα αντικείμενο ή μία θέση κάπου μέσα σε αυτό το στιγμιότυπο, θα χρησιμοποιούμε ένα μονοπάτι της μορφής

Snapshot/object1/object11/object113

Με αυτόν τον τρόπο θα μπορούμε να αναφερόμαστε σε μία θέση ή ένα αντικείμενο του συστήματός μας. Επίσης θα μπορούμε να χρησιμοποιούμε και σύμβολα σχετικότητας μονοπατιού, δανεισμένα από τα αντίστοιχα σύμβολα των συστημάτων αρχείων, όπως για παράδειγμα την τελεία (.), και τις δύο τελείες (..) που δηλώνουν τον τρέχοντα κατάλογο και τον κατάλογο – πρόγονο, αντίστοιχα.

Ας δούμε τις παραμέτρους των πράξεων αναλυτικά.

Η addobject σημαίνει ότι έγινε μία εισαγωγή αντικειμένου μέσα στο στιγμιότυπο, στη θέση που ορίζει το parentposition.

Το αντικείμενο το οποίο εισήχθη βρίσκεται μέσα σε άγκιστρα <, > οπότε αυτό, όπως είπαμε, σημαίνει ότι πρόκειται για μία αναφορά σε κάποια δήλωση αντικειμένου που βρίσκεται μέσα στο αρχείο καταγραφής μας.

Τις ίδιες ακριβώς παραμέτρους χρησιμοποιεί και η διαγραφή, με τη διαφορά ότι η αναφορά δηλώνει το αντικείμενο το οποίο διαγράφηκε. Η παρατήρηση εδώ είναι ότι κάθε φορά που διαγράφεται ένα αντικείμενο πρέπει να δηλώνεται, καθώς και να καταγράφεται το περιεχόμενό του. Ο λόγος για αυτή τη φαινομενικά περιττή πράξη είναι ότι έτσι η πράξη της διαγραφής θα μπορεί να αντιστρέφεται και αυτό χρησιμεύει κατά τη διαδικασία της ανάκτησης ή της επαναφοράς. Πιο συγκεκριμένα, η πράξη της διαγραφής κατά την αντιστροφή μετατρέπεται σε πράξη εισαγωγής. Επομένως στη περίπτωση που δεν κρατάμε πληροφορίες για το ΤΙ διαγράφηκε τότε δεν θα μπορούμε να ανακτήσουμε το στιγμιότυπο που μας ενδιαφέρει.

Έτσι, για την περίπτωση που δεν μας ενδιαφέρει η ανάκτηση συστήματος αλλά μόνο η ανάκτηση δέλτα, δηλαδή η ανάκτηση των γεγονότων θα μπορούμε να αφήνουμε αυτή την παράμετρο σε μορφή NIL, για την οποία μιλήσαμε παραπάνω.

Αντίστοιχα σχόλια με αυτά που κάναμε για τις πράξεις των αντικειμένων έχουμε να κάνουμε και για τις πράξεις των ιδιοτήτων των αντικειμένων.

Επιπλέον, εδώ υπάρχει και η παράμετρος της ονομασίας της ιδιότητας, η οποία, όπως αναφέραμε και παραπάνω, αποτελεί το αναγνωριστικό της, το οποίο σε συνδυασμό με την παράμετρο position αποτελούν τον τρόπο με τον οποίο θα αναφερόμαστε μοναδικά σε μία ιδιότητα. Αυτό σημαίνει ότι, για παράδειγμα, η ονομασία «βάρος» μπορεί να βρίσκεται σε πολλά αντικείμενα στα δάφφορα επίπεδα λεπτομέρειας του συστήματος, όμως για κάθε ένα αντικείμενο αποτελεί μοναδική έκφραση στην ιδιότητα αυτού.

Επίσης στην πράξη updateproperty, για λόγους αντιστρεψιμότητας, όπως αναφέραμε και στις προηγούμενες παραγράφους, χρειάζεται να κρατάμε και την παλαιά τιμή, δηλαδή αυτήν που αντικαταστάθηκε από τη νέα. Φυσικά, μπορούμε να αφήσουμε αυτό το πεδίο σε μορφή NIL, στην περίπτωση που δεν μας ενδιαφέρει η αντιστρεψιμότητα του συστήματος.

Αυτές ήταν οι στοιχειώδεις πράξεις ανάμεσα στα αντικείμενα και τις ιδιότητές τους. Όμως μπορούμε να ορίσουμε και σύνθετες πράξεις και να τους δώσουμε μάλιστα και όνομα, έτσι ώστε να μπορούμε να αναφερόμαστε σε αυτές κατά την καταγραφή των

γεγονότων μας. Έτσι, ανακαλώντας την απαίτηση για επεκτασιμότητα, θα μπορούμε να δηλώσουμε νέες πράξεις ανάμεσα στα αντικείμενα, δηλαδή στα αποτελέσματα, όπως, για παράδειγμα, την πράξη «μεταμόρφωση» ή «περιστροφή».

Για να το πετύχουμε αυτό χρησιμοποιούμε τη δήλωση της μακροπράξης.

Macro { Operation+ }

Για να ορίσουμε ένα macro χρειαζόμαστε τουλάχιστο μία βασική πράξη, όπως υποδηλώνει το σύμβολο +.

Η μακρό-πράξη έχει σαν προκαθορισμένη ιδιότητα ένα ID, έτσι ώστε να μπορούμε να αναφερθούμε σε αυτήν, μέσα από τα γεγονότα.

Αφού είδαμε όλες τις δηλώσεις που μπορούμε να χρησιμοποιήσουμε για την καταγραφή ενός πλάνου, το μόνο που μένει είναι να δώσουμε και τον τυπικό ορισμό του πλάνου.

```
Plan {Time+
      Declaration*
      Comment*
      Actor*
      Action*
      Snapshot*
      Object*
      Macro*
      Fact* }
```

Παρατηρούμε ότι το μόνο υποχρεωτικό στοιχείο της δήλωσης του πλάνου, είναι αυτό του χρόνου. Ο χρόνος, (εκτός και αν έχουμε προσθέσει την ιδιότητα ID) αποτελεί και το προσδιοριστή της μοναδικότητας ενός πλάνου και είναι αυτός που θα μας επιτρέψει να ανακτήσουμε τα γεγονότα που συνέβησαν σε κάποια χρονική στιγμή.

Όλες οι άλλες οντότητες μπορούν να παραλειφθούν, υποδηλώνοντας έτσι την ύπαρξη του *Kενού Πλάνου* (dummy). Αυτό σε ορισμένες περιπτώσεις μπορεί να φανεί χρήσιμο, ενώ σε άλλες μπορεί να οδηγήσει σε παρερμηνείες. Γι' αυτό, θα πρέπει να δίνεται ιδιαίτερη προσοχή στο στάδιο της ανάλυσης για την εγκατάσταση των αισθητήρων, που μιλήσαμε στην προηγούμενη ενότητα.

Επιπλέον, όλες οι παραπάνω οντότητες μπορούν να περιέχονται ελεύθερα μέσα στο πλάνο και αυτό μας δίνει το πλεονέκτημα της ευελιξίας κατά τη διαδικασία της υλοποίησης του μοντέλου.

Ένα τελευταίο θέμα σε αυτή την ενότητα είναι οι προκαθορισμένες ιδιότητες που έχουμε, ήδη, αναφέρει. Ωα χρησιμοποιούμε το σύμβολο & μπροστά από μία οντότητα, για να δηλώσουμε ότι ακολουθεί η λίστα των προκαθορισμένων ιδιοτήτων αυτής της οντότητας. Έτσι, θα έχουμε:

```

&Snapshot : (ID#)
&Object : (ID#)
&Time : (Value# TimeFormat?)
&Actor : (ID#)
&Action : (ID#)
&Macro : (ID#)
&Fact : (ID?)
&Declaration : ( SystemType#
                  Interval?
                  Threshold?
                  Quantize?
                  TimeFormat?
                  Chronon?
                  Version?
                )

```

```

SystemType : ( Eventdriven | Timedriven | Hybrid )
Interval : (number)
Threshold : (number)
Quantize : (number)
Timeformat : (datetime | frames | plans)
Chronon : (number)

```

Παρατηρούμε ότι οι περισσότερες από τις οντότητες έχουν ένα υποχρεωτικό ID, δηλαδή μία προκαθορισμένη ιδιότητα, που δεν μπορεί να διαγραφεί ή να εισαχθεί ξανά, ωστόσο μπορεί να αλλάξει τιμή. Αυτή η ιδιότητα είναι ο προσδιοριστής της οντότητας και έχουμε ήδη σχολιάσει τη χρησιμότητά του.

Επίσης η οντότητα TIME έχει την προκαθορισμένη ιδιότητα *value* και τη μη υποχρεωτική ιδιότητα *timeformat*, η οποία μας πληροφορεί για το είδος του χρόνου που μετράει στο σύστημα μας. Τα είδη του χρόνου μπορεί να είναι τρία:

- **Datetime**, που είναι το γνωστό σύστημα ημερομηνίας και ώρας.
- **Frames**, που είναι ένα σύστημα καταμέτρησης το οποίο χρησιμοποιείται κυρίως σε πολυμεσικές εφαρμογές.
- **Plans**, που είναι ένας αύξοντας αριθμός για κάθε νέο πλάνο που προστίθεται στο αρχείο μας και χρησιμεύει για τα συστήματα στα οποία δεν μας ενδιαφέρει η καταγραφή του πραγματικού χρόνου αλλά τα «βήματα» που έχουν γίνει κατά τη λειτουργία του συστήματος.

Τέλος, η λίστα των προκαθορισμένων ιδιοτήτων της οντότητας declaration περιλαμβάνει μία σειρά από έννοιες οι οποίες έχουν αναλυθεί στις προηγούμενες παραγράφους (threshold, quantize, κλπ).

5.4. Η διαχείριση των καταγραφών

Αφού εξηγήσαμε τον τρόπο με τον οποίο θα αποθηκεύονται τα πλάνα μέσα στο αρχείο καταγραφών, θα χρειαστούμε και ένα σύνολο από εντολές για τη διαχείριση αυτού του αρχείου. Σε αυτήν, καθώς και στις επόμενες δύο, ενότητες θα μιλήσουμε για αυτά τα σύνολα εντολών τα οποία χρησιμοποιούμε στην κάθε περίπτωση.

Πρέπει να πούμε ότι αυτά τα σύνολα των εντολών που θα περιγράψουμε είναι απλώς τα συστήματα διεπαφής (Interface), πράγμα που σημαίνει ότι ο κάθε ένας θα μπορεί να ακολουθήσει το δικό του τρόπο υλοποίησης, σύμφωνα με τη γλώσσα προγραμματισμού ή με οτιδήποτε άλλο υπόβαθρο υλοποίησης, χρησιμοποιεί.

Πριν όμως αναπτύξουμε αυτά τα σύνολα εντολών, θα χρειαστεί να εισάγουμε κάποιες συναρτήσεις οι οποίες θα είναι χρήσιμες για τη λειτουργικότητα αυτών των εντολών.

```
Now()  
Beginning()  
Prevtime(time)  
Nexttime(time)
```

Αυτές οι συναρτήσεις επιστρέφουν μονάδα χρόνου (time).

Με τη συνάρτηση now, θα μπορούμε να διαβάζουμε την τρέχουσα ώρα του συστήματος.

Με τη συνάρτηση beginning θα αναφερόμαστε στην παλαιότερη καταγραφή του πλάνου που υπάρχει στο αρχείο μας.

Οι δύο επόμενες συναρτήσεις μπορούν να πάρουν μία παράμετρο χρόνου. Η πρώτη μας δίνει το αμέσως προηγούμενο χρονικά πλάνο από την τιμή της παραμέτρου time. Αντιστοίχως, η δεύτερη μας δίνει το αμέσως επόμενο χρονικά πλάνο. Αυτές οι δύο συναρτήσεις χρησιμεύουν για την πλοήγησή μας στο αρχείο καταγραφής, όπως θα δούμε παρακάτω.

Ας δούμε τώρα τα σύνολα εντολών για τα οποία μιλήσαμε στην αρχή αυτής της ενότητας.

Το πρώτο σύνολο εντολών χρησιμεύει για την περίπτωση που θέλουμε να εισάγουμε ή να διαγράψουμε ένα πλάνο από το αρχείο καταγραφής μας.

```
Insert(time, plan)  
Delete(time)  
PlanExists(time)
```

Σε αυτό το σύνολο υπάρχουν μόνο δύο απλές εντολές και μία συνάρτηση. Εφόσον, όπως προαναφέραμε, το μέσο αποθήκευσης των πλάνων μας είναι τυχαίας προσπέλασης σε φυσικό επίπεδο, δεν θα χρειαστεί να έχουμε εντολές σειριακής διαχείρισης όπως για παράδειγμα εντολές αναδιοργάνωσης ή εντολές επικόλλησης πλάνου.

Έτσι, θα έχουμε την εισαγωγή (*insert*) νέου πλάνου τη χρονική στιγμή *time*. Στην περίπτωση που σ' αυτή τη χρονική στιγμή υπάρχει ήδη ένα πλάνο καταχωρημένο, τότε αυτό θα διαγράφεται για να εισαχθεί το νέο. Δεν πρέπει να ξεχνάμε ότι δεν επιτρέπεται, εξ ορισμού, να υπάρχουν δύο πλάνα με την ίδια καταγραφή χρόνου.

Σε περίπτωση που θέλουμε να βεβαιωθούμε για την, τυχόν, ύπαρξη πλάνου στο σύστημα πριν από την εισαγωγή νέου, τότε θα χρησιμοποιούμε τη συνάρτηση *planexists* που μας επιτρέπει να ελέγξουμε αν υπάρχει αποθηκευμένο πλάνο για μία συγκεκριμένη στιγμή του χρόνου

Τέλος, υπάρχει και η εντολή της διαγραφής ενός πλάνου τη συγκεκριμένη χρονική στιγμή.

Ένα θέμα που τίθεται με αυτές τις εντολές εισαγωγής και διαγραφής πλάνων, είναι αυτό της ασφάλειας και της ακεραιότητας. Υπάρχει πιθανότητα, σε περίπτωση που διαγραφεί ή εισαχθεί εκ των υστερών ένα πλάνο, να δημιουργηθεί παρερμηνεία κατά την ανάκτηση ή επαναφορά του συστήματος ή να δημιουργηθεί θέμα ασφάλειας, σε περίπτωση που καταγράφονται κρίσιμες πληροφορίες. Αυτό είναι ένα ζήτημα που θα πρέπει να δώσει αρκετή προσοχή ο αναλυτής του συστήματος κατά τη διαδικασία υλοποίησης του μοντέλου.

5.5. Η ανάκτηση των καταγραφών

Το επόμενο σύνολο εντολών αφορά τέσσερις επιμέρους λειτουργίες. Αυτές είναι της πλοήγησης, της ανάκτησης πλάνων, της ανάκτησης αντικειμένων και της ανάκτησης γεγονότων.

Σε όλες τις περιπτώσεις, πάντως, πρέπει να έχουμε υπόψη αυτό που είχαμε αναφέρει στις προηγούμενες ενότητες για το σειριακό τρόπο προσπέλασης των πλάνων του αρχείου καταγραφής μας. Έτσι, θα μπορούμε να φανταστούμε έναν δείκτη ο οποίος θα μετακινείται μέσα στο αρχείο μας και κάθε φορά θα δείχνει ένα πλάνο, το οποίο θα είναι και το τρέχων πλάνο.

Έτσι, για να μπορέσουμε να χειριστούμε αυτό το λογικό αρχείο καταγραφών, θα χρειαστούμε αρχικά εντολές πλοήγησης:

```
GotoPlan(time)  
GotoNextPlan()  
GotoPrevPlan()
```

Η πρώτη εντολή έχει παράμετρο μία χρονική στιγμή και μας βοηθάει να μετακινηθούμε γρήγορα σε μία δεδομένη χρονική στιγμή. Για την περίπτωση που δεν υπάρχει πλάνο με αυτή τη χρονική στιγμή τότε ο δείκτης μεταφέρεται στο πλησιέστερο πλάνο.

Φυσικά, μπορούμε να χρησιμοποιήσουμε κάποια από τις συναρτήσεις που αναφέραμε στην αρχή αυτής της ενότητας. Για παράδειγμα, αν θέλουμε να μετακινήσουμε το δείκτη στη θέση του πιο πρόσφατου πλάνου, θα δώσουμε την εντολή:

```
GotoPlan(now())
```

Με την ίδια λογική, οι άλλες δύο εντολές μετακινούν το δείκτη στο αμέσως επόμενο και στο αμέσως προηγούμενο, χρονικά, πλάνο, αντίστοιχα.

Οι επόμενες δύο εντολές, είναι εντολές συστήματος.

```
SetDetailLevel (number)  
GetTimeIndex ()
```

Η εντολή *Setdetaillevel* προστάζει το σύστημα να χειρίζεται τα αντικείμενα με συγκεκριμένο επίπεδο λεπτομέρειας. Αυτό είναι χρήσιμο σε περίπτωση που ενδιαφερόμαστε για πληροφορίες ενός επιπέδου, χωρίς να μας απασχολούν οι λεπτομέρειες ενός κατώτερου επιπέδου. Έτσι, θέτοντας ως επίπεδο λεπτομέρειας το μηδέν, θα εννοούμε ότι μας ενδιαφέρει μόνο το αντικείμενο στο οποίο αναφερόμαστε. Κάθε απόγονος αυτού του αντικειμένου θα παραλείπεται. Αντίστοιχα, όταν έχουμε επίπεδο λεπτομέρειας 1, τότε θα λαμβάνουμε υπόψη και τους άμεσους απόγονους του αντικειμένου στο οποίο αναφερόμαστε κ.ο.κ.

Η εντολή *Gettimeindex* μας επιστρέφει τη θέση του δείκτη στο αρχείο καταγραφής, έτσι ώστε να έχουμε μία ιδέα σχετικά με την πλοήγησή μας μέσα στο αρχείο καταγραφής.

Τέλος, έχουμε να παρουσιάσουμε τα σύνολα εντολών με τα οποία μπορούμε να ανακτήσουμε ένα πλάνο, να ανακτήσουμε μία σειρά γεγονότων για ένα συγκεκριμένο αντικείμενο και, τέλος, να ανακτήσουμε ένα αντικείμενο, αντίστοιχα:

```
GetPlan()  
GetPlanByTime (time)  
GetNextPlan ()  
GetPrevPlan ()  
  
GetFacts (position)  
GetFactsByTime (position, time)  
GetNextFacts (position)  
GetPrevFacts (position)  
  
GetObject (position)  
GetObjectByTime (position, time)  
GetNextObject (position)  
GetPrevObject (position)
```

Όλα τα παραπάνω έχουν τη λειτουργία που φανερώνει η ονομασία τους, με την παρατήρηση ότι σε περίπτωση που δεν έχουμε παράμετρο χρόνου, θα εννοείται ότι αναφερόμαστε στο χρόνο του δείκτη του αρχείου καταγραφής μας. Επίσης, με τις εντολές *getnext...* και *getprev...* θα αναφερόμαστε στο επόμενο, χρονικά,

αντικείμενο, δηλαδή αυτό που ανήκει στο αμέσως επόμενο πλαίσιο ή στο αμέσως προηγούμενο πλαίσιο, αντίστοιχα.

5.6. Η επαναφορά του συστήματος

Τελευταίο σύνολο εντολών για τη διαχείρισή μας, είναι αυτές που μας βοηθούν στην επαναφορά του συστήματος σε μία παλιότερη κατάσταση. Πρέπει να τονίσουμε ότι δεν είναι όλα τα συστήματα αντιστρέψιμα, και εξαρτάται από τη φύση του ιδίου του συστήματος αν επιτρέπεται αυτό κάτω από ορισμένες συνθήκες ή όχι. Για παράδειγμα, ένα σύστημα επεξεργασίας κειμένου μπορεί να επαναφέρει ένα κείμενο σε παλαιότερες καταστάσεις του, ενώ σε ένα πληροφοριακό σύστημα μπορεί να μην είναι δυνατό να το επαναφέρουμε σε μία παλαιότερη κατάσταση.

Οι λειτουργίες είναι δύο:

Undo (*time*)

Redo (*time*)

είναι οι γνωστές λειτουργίες με τις επίσης γνωστές συνέπειες. Η μόνη παρατήρηση που πρέπει να γίνει εδώ, είναι σχετικά με την παράμετρο *time*. Εδώ, αναφερόμαστε στο χρόνο στον οποίο θέλουμε να επιστρέψει ένα σύστημα, σχετικά με τον τρέχοντα χρόνο. Έτσι, θα ζητάμε να γίνει επαναφορά του συστήματος, όπως αυτό ήταν πριν τ χρόνο και, σε περίπτωση που ο τύπος του χρόνου είναι *plan*, τότε θα μιλάμε για βήματα.

Υπάρχουν αρκετά είδη μοντέλων για τις διαδικασίες UNDO και REDO[19] [20] [18]. Ωστόσο, η φύση του μοντέλου ανάκτησης δέλτα είναι γραμμική, όπως έχουμε αναφέρει και στις προηγούμενες ενότητες. Παρόλα αυτά, ο υλοποιητής του μοντέλου, έχει την ελευθερία να διαλέξει το μηχανισμό UNDO που ταιριάζει περισσότερο στο πρόβλημα που καλείται κάθε φορά να λύσει.

Κεφαλαιο 6

*Εφαρμογή του
μοντέλου*

6. Εφαρμογή μοντέλου ανάκτησης δέλτα

6.1. Πρόλογος

Σε αυτό το κεφάλαιο θα παρουσιάσουμε μια εφαρμογή του μοντέλου που αναλύσαμε σε προηγούμενες παραγράφους. Η εφαρμογή αυτή θα είναι στη γλώσσα XML. Όπως γνωρίζουμε, το σημαντικότερο κομμάτι σε ένα έγγραφο γλώσσας XML είναι ο προσδιορισμός του περιεχομένου του, δηλαδή των *elements* και των *attributes* του ριζικού στοιχείου που, στην περίπτωση μας, θα ονομάζεται **drml** (**D**elta **R**etrievi**ng** **M**odeling **L**anguage). Υπάρχουν πολλοί τρόποι για να ορίσουμε ένα *Σχήμα* σε XML [16]. Ως προτιμήσουμε τον τρόπο του DTD (Document Type Definition) ο οποίος είναι και ο πιο κοινός τρόπος και, επιπλέον, αποτελεί ένα πρότυπο το οποίο είναι ευρέως διαδεδομένο.

6.2. Εφαρμογή σε XML – Document Type Definition

```
<!--  
This is the XML document type definition (DTD) for  
Delta Retrieving Modelling Language (DRML)  
  
Date: December 2000  
Authors: Lefteris Loukis  
Version: 1.00  
  
-->  
  
<!-- Generally useful entities -->  
<!ENTITY % id-attr "id ID #REQUIRED">  
  
<===== DRML Document =====>  
<!--  
The root element DRML contains all other elements.  
-->  
<!ELEMENT drml plan?>  
<!ATTLIST drml  
      %id-attr;  
>  
  
<===== Plan Element =====>  
<!--  
This is the basic element of DRML  
-->  
<!ELEMENT plan (Time
```

```

        Declaration*
        Comment*
        Actor*
        Action*
        Snapshot*
        Object*
        Macro*
        Fact* }

>

<!ATTLIST plan
          %id-attr;
>

<!--===== Snapshot Element =====-->
<!ELEMENT snapshot object*>

<!ATTLIST snapshot
          %id-attr;
>

<!--===== Object Element =====-->
<!ELEMENT object ANY>

<!ATTLIST object
          %id-attr;
>

<!--===== Time Element =====-->
<!ELEMENT time ANY>

<!ATTLIST time
          %id-attr;
          value CDATA
          timeformat (datetime | frames | plans)
>

<!--===== Actor Element =====-->
<!ELEMENT actor ANY>

<!ATTLIST actor
          %id-attr;
>

<!--===== Action Element =====-->
<!ELEMENT action ANY>

<!ATTLIST action
          %id-attr;
>

<!--===== Declaration Element =====-->
<!ELEMENT declaration ANY>

<!ATTLIST declaration
          %id-attr;

```

```

        systemtype ( Eventdriven | Timedriven | Hybrid )
        interval CDATA
        threshold CDATA
        quantize CDATA
        timeformat (datetime | frames | plans)
        chronon CDATA
        version CDATA
    >

<!--===== Macro Element =====>
<!ELEMENT macro operation+>

<!ATTLIST declaration
      %id-attr;
>

<!--===== Comment Element =====>
<!ELEMENT comment EMPTY>

<!ATTLIST declaration
      %id-attr;
      text CDATA
>

<!--===== Fact Element =====>
<!ELEMENT fact  ActorID*
              ActionID*
              operation*
              MacroID*
>

<!ATTLIST fact
      %id-attr;
>

<!--===== Operation Element =====>
<!ELEMENT operation addobject*
              deleteobject*
              addproperty*
              deleteProperty*
              changeProperty*
>

<!ATTLIST operation
      %id-attr;
>

<!--===== Addobject Element =====>
<!ELEMENT addobject EMPTY>

<!ATTLIST addobject
      parentposition CDATA
      object ObjectID
>

```

```

<!----- Deleteobject Element ----->
<!ELEMENT deleteobject EMPTY>

<!ATTLIST deleteobject
      position CDATA
      object ObjectID
>

<!----- Addproperty Element ----->
<!ELEMENT addproperty EMPTY>

<!ATTLIST addproperty
      position CDATA
      propertynname CDATA
      value CDATA
>

<!----- DeleteProperty Element ----->
<!ELEMENT deleteproperty EMPTY>

<!ATTLIST deleteproperty
      position CDATA
      propertynname CDATA
      value CDATA
>

<!----- ChangeProperty Element ----->
<!ELEMENT changeproperty EMPTY>

<!ATTLIST changeproperty
      position CDATA
      propertynname CDATA
      oldvalue CDATA
      value CDATA
>

```

>

Κεφαλαιο 7

Αναφορές

7. Αναφορές

- [1] Ε.Κιουντούζης, *Μεθοδολογίες ανάλυσης και σχεδιασμού Πληροφοριακών Συστημάτων*, σελ. 17-79, 1997
- [2] P. Spencer, *Professional XML Design and Implementation*, Wrox Press, 1999
- [3] A. Steiner, *A generalisation Approach to Temporal Data Models and their Implementations*, Swiss Federal Institute of Technology, Zurich, 1998
- [4] Α.Σταυρόπουλος, *Φυσικές Επιστήμες*, Πειραιάς, Σταμούλης 1988
- [5] I.X. Παναγιωτόπουλος, *Δομημένος Προγραμματισμός Pascal*, σελ.17-20, Πειραιάς, Σταμούλης 1990
- [6] Sybase Inc, *Version Control Interfaces*, part no.AA0309, 1996
- [7] Schoderberk,P. , Schoderberk, C, and Kefalas, A. , *Management Systems: Conceptual Considerations*, 4th edition, Richard Irwin Inc. Burr Ridge, Illinois, USA, 1990
- [8] J. Clifford and A.U. Tansel, *An Algebra for Historical Relational Databases: Two Views*, S.Navathe editor, Sigmod Record, 1985
- [9] R.Snodgrass, editor, *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers, 101 Philip Drive, Assinippi Park, Norwell, 1995
- [10] A.Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass. *Temporal Databases: Theory, Design, and Implementation*, Benjamin Publishing Company, 1993
- [11] R. Snodgrass and I. Ahn. *A Taxonomy of Time in Databases*. In S. Navathe, editor, Proceedings of the ACM SIGMOD Internationnal Conference on Management og Data, pp236-246, 1985
- [12] Internet Engineering Task Force, IETF, (2000), διαθέσιμο στην <http://www.ietf.org/home.html>
- [13] International Standard ISO 8879 *Information Processing - Text and Office Systems -Standardized Generalized Markup Language (SGML)*, First Edition - 1986-10-15 UDC 681.3.06 Ref. No. ISO 8879-1986(E)
- [14] The World Wide Web Consortium (W3C), and *XML specification*, διαθέσιμο στο: <http://www.w3.org/TR/REC-xml>
- [15] HyperText Markup Language (RFC 1866) <http://ds.internic.net/rfc/rfc1866.txt>
- [16] Oasis, Robin Cover, *The XML Cover Pages* (2000), διαθέσιμο στην: <http://www.oasis-open.org/cover/sgml-xml.html>,

- [17] Microsoft Corporation, 2000, *Active Channels, CDF Reference* at <http://msdn.microsoft.com/workshop/delivery/cdf/reference/channels.asp>
- [18] T. Berlage, *A Selective Undo Mechanism for Graphical User Interfaces Based On Command Objects*, GMD (German National Research Center for Computer Science) 53754 Sankt Augustin, Germany
- [19] G. B. LEEMAN, JR, *A Formal Approach to Undo Operations in Programming Languages*, IBM Thomas J. Watson Research Center
- [20] W. Keith Edwards, T. Igarashi, A. LaMarca, E. D. Mynatt, *A Temporal Model for Multi-Level Undo and Redo.*



Κεφαλαιο 8

Παρατήματα



8. Παραρτήματα

8.1. Παράρτημα A

Σημειογραφία

Συμβόλισμοί

- , : διατεταγμένη σειρά
? : Προαιρετικό
: Απαιτούμενο
+ : ένα ή περισσότερα
* : μηδέν ή περισσότερα
| : επιλογή (διάζευξη)
() : Ομαδοποίηση
< > : αναφορά σε οντότητα
& : Λίστα ιδιοτήτων

Τύποι δεδομένων

```
Property : (PropertyName = value)
value :( id | number | text | bin | nil )

oldvalue : value
Position : ObjectPath
ParentPosition : Position
```

Ορισμοί

```
Snapshot { Property+ Object* }
Object { Property+ Object* }

Time { Property+ }
Actor { Property+ }
Action { Property+ }
Declaration { Property+ }

Macro { Operation+ }
Comment { (id | text)+ }

Fact { <Actor*> <Action*> Operation* <Macro*> }

Plan { Time+
       Declaration*
       Comment*
       Actor*
       Action*
       Snapshot*
       Object*
       Macro*
       Fact* }

Operation : ( . . . Ένα από τα Παρακάτω . . . )

Addobject(ParentPosition, <Object>)
Deleteobject(Position, <Object>)

Addproperty(Position, Propertynname, value)
DeleteProperty(Position, Propertynname, value)

ChangeProperty(Position, Propertynname, value, oldvalue)
```

Λίστες ιδιοτήτων

```
&Snapshot : (ID#)
&Object : (ID#)
&Time : (Value# TimeFormat?)
&Actor : (ID#)
&Action : (ID#)
&Macro : (ID#)
&Fact : (ID?)
&Declaration : ( SystemType#
                  Interval?
                  Threshold?
                  Quantize?
                  TimeFormat?
                  Chronon?
                  Version?
                )
```

```
SystemType : ( Eventdriven | Timedriven | Hybrid )
Interval : (number)
Threshold : (number)
Quantize : (number)
Timeformat (datetime | frames | plans)
Chronon : (number)
```

8.2. Παράρτημα B

Σύστημα διεπαφής

Συναρτήσεις

```
Now()  
Beginning()  
Prevtime(time)  
Nexttime(time)  
  
Snapshot = the root position
```

Εντολές διαχείρισης

```
Insert(time, plan)  
Delete(time)  
PlanExists(time)
```

Εντολές ανάκτησης

```
GotoPlan(time)  
GotoNextPlan()  
GotoPrevPlan()
```

```
SetDetailLevel(number)
GetTimeIndex()

GetPlan()
GetPlanByTime(time)
GetNextPlan()
GetPrevPlan()
TracePlans(time, time)

GetFacts(position)
GetFactsByTime(position, time)
GetNextFacts(position)
GetPrevFacts(position)
TraceFacts(position, time, time)

GetObject(position)
GetObjectByTime(position, time)
GetNextObject(position)
GetPrevObject(position)
TraceObject(position, time, time)
```

Εντολές επαναφοράς

```
Undo(time)
Redo(time)
```

9. Περίληψη

Το θέμα της εργασίας ασχολείται με τις αλλαγές που συμβαίνουν σε ένα σύστημα. Η ιδέα ξεκίνησε από την ανάγκη της αυτοματοποιημένης και τυποποιημένης ενημέρωσης ενός Χρήστη του Διαδικτύου, για τις αλλαγές που συμβαίνουν στους δικτυακούς Χώρους, με σκοπό την απλοποίηση της περιήγησής του στους δαιδαλώδεις ηλεκτρονικούς διαδρόμους του *Παγκόσμιου Ιστού Υπολογιστών*.

Με τις αμφιδρομες επικοινωνίες (Internet, Κινητή Τηλεφωνία, κ.α.) να καταλαμβάνουν πλέον ένα σημαντικό μερίδιο του επαγγελματικού και μη, χώρου, υπάρχει ανάγκη για φθηνή επικοινωνία και μετάδοση πληροφοριών, με ελάχιστο πλεόνασμα, σε τυποποιημένη μορφή και με δυναμικότητα στη χρήση και επεξεργασία τους. Όπως είναι φανερό, ο σύγχρονος άνθρωπος – επαγγελματίας - χρήστης ενδιαφέρεται για την περιεκτική, ταξινομημένη και ουσιώδη πληροφορία, η οποία θα του παρέχεται με αυτοματοποιημένο τρόπο, έτσι ώστε να ελαχιστοποιήσει τα έξοδά του αλλά και τη δικτυακή συμφόρηση.

Τίθεται λοιπόν η ανάγκη για μία ενιαία, πρότυπη, και συναφή διαχείριση της καταγραφής των αιτιών και των αποτελεσμάτων μιας αλλαγής, αλλά και της ανάκτησης αυτών των αλλαγών σε επόμενες χρονικές στιγμές.

Η ανάγκη αυτή οδήγησε στην επέκταση του ζητούμενου θέματος και έτσι στη γενίκευση του προβλήματος. Δηλαδή οδήγησε στην αναζήτηση της προτυποποίησης ενός «μηχανισμού-μοντέλου» που θα μπορεί να παρατηρεί, να αντιλαμβάνεται, να καταγράφει και να αναπαριστάνει τις διαφορές στην κατάσταση ενός συστήματος κατά τη δράση του στο χρόνο.

Στην εργασία, επιχειρούμε την παρουσίαση ενός τέτοιου μοντέλου που να είναι ικανό να περιγράφει και να καταγράφει με τυπικό τρόπο τις αλλαγές που συμβαίνουν σε ένα σύστημα σε διάφορα επίπεδα λεπτομέρειας, έτσι ώστε να είναι δυνατή η ανάκτηση αυτών των αλλαγών είτε σε μορφή στιγμιότυπου (δηλαδή την κατάσταση του συστήματος σε μία συγκεκριμένη χρονική στιγμή), ή σε μορφή διαφορών ανάμεσα σε δύο στιγμιότυπα. Παράλληλα καταγράφονται και οι ενέργειες που «τυχόν» ευθύνονται για τις καταγραμμένες αλλαγές, για να μπορούμε να καλύψουμε έτσι όσο το δυνατόν περισσότερες περιπτώσεις μελέτης συστημάτων.

Παρουσιάζεται επομένως, μία σειρά από ορισμούς, προτάσεις και ιδέες για τον τρόπο σκέψης που θα πρέπει να χρησιμοποιήσουμε με σκοπό την ανάλυση, σχεδίαση και εφαρμογή ενός τέτοιου μηχανισμού καταγραφής και διαχείρισης των διαφορών που συμβαίνουν στην κατάσταση ενός συστήματος, στο πέρασμα του χρόνου, ή όπως αναφέρεται και στον τίτλο αυτής της εργασίας, η *Μοντελοποίηση της Ανάκτησης Δελτα*.

Επιπλέον, παρουσιάζεται μία ψευδογλώσσα - σημειογραφία η οποία μπορεί να χρησιμοποιηθεί ως πρότυπο για τις διάφορες υλοποιήσεις που μπορούν να γίνουν με βάση αυτό το μοντέλο.

Το περιεχόμενο της εργασίας κατανέμεται ως εξής:

Στο πρώτο μέρος, γίνεται μια εισαγωγή σε ορισμένες βασικές έννοιες, οι οποίες αναλύονται στις επόμενες ενότητες. Χωρίζεται στα επιμέρους θέματα της θεωρίας συστημάτων, σε θέματα αυτοματοποίησης, κυβερνητικής και έξυπνων πρακτόρων και τέλος σε θέματα χρόνου και χρονικών βάσεων δεδομένων. Επιπλέον, υπάρχουν οι

ορισμοί για κάποιες έννοιες και τεχνολογίες που αναφέρονται στα επόμενα κεφαλαία και είναι χρήσιμες για την κατανόησή τους.

Αμέσως μετά γίνεται η τοποθέτηση του προβλήματος, η εξήγηση για την ανάγκη και τα σενάρια χρήσης για το μοντέλο ανάκτησης δέλτα, καθώς και οι απαιτήσεις που θα πρέπει να έχουμε από ένα τέτοιο μοντέλο.

Στην επόμενη ενότητα παρουσιάζονται κάποιες βασικές έννοιες και ορισμοί που θα είναι χρήσιμοι για την κατανόησή του υπόλοιπου κειμένου και για την απόκτηση μιας σφαιρικής άποψης για την τεχνολογία που θα περιγραφεί στο επόμενο μέρος. Έτσι αναλύουμε και ορίζουμε τις έννοιες των επιμέρους στοιχείων γύρω από το σύστημα, το οποίο θα αναλύσουμε πριν εφαρμόσουμε το ίδιο το μοντέλο. Τέτοιες έννοιες είναι αυτές του στιγμιότυπου, του πλάνου, της κατάστασης, του γεγονότος κλπ. Αυτό θα μας βοηθήσει στη μεθοδικότερη, άρα και ευκολότερη αντίληψη του τι ακριβώς θέλουμε να πετύχουμε.

Το κυρίως μέρος της εργασίας ασχολείται με την περιγραφή του μοντέλου. Το βασικότερο κομμάτι αυτού, είναι η έννοια του *Πράκτορα*. Λέγοντας Πράκτορα, εννοούμε εκείνο το μηχανισμό που, με μία σειρά από λειτουργίες-διεργασίες, επιτελεί τον προαναφερόμενο σκοπό μας.

Οι λειτουργίες αυτές είναι οι εξής:

6. **Η παρατήρηση του συστήματος**
7. **Η καταγραφή της αλλαγής (σε αποθηκευτικό χώρο)**
8. **Η διαχείριση των καταγραφών**
9. **Η (επιλεκτική) ανάκτηση των καταγραφών**
10. **Η πιθανή επαναφορά του συστήματος σε μία παλαιότερη κατάσταση**

Ακολουθεί η λεπτομερής περιγραφή των παραπάνω λειτουργιών και της ψευδογλώσσας – σημειογραφίας που χρησιμοποιεί ο μηχανισμός αυτός

Στο τελευταίο μέρος της εργασίας, παρουσιάζουμε μια εφαρμογή του μοντέλου που αναλύσαμε. Η εφαρμογή αυτή είναι στη γλώσσα XML. Όπως γνωρίζουμε, το σημαντικότερο κομμάτι σε ένα έγγραφο γλώσσας XML είναι ο προσδιορισμός του περιεχομένου του, δηλαδή των *elements* και των *attributes* του ριζικού στοιχείου του εγγράφου που, στην περίπτωση μας, θα ονομάζεται **drml** (**D**elta **R**etrievi**ng** **M**odeling **L**anguage). Για να ορίσουμε το Σχήμα της drml, χρησιμοποιούμε ένα DTD (Document Type Definition).

10. Executive Summary

The project deals with the *changes* that occur in a system. The idea was created by the need to give the internet user, automated and standardized information about the changes that take place in the Internet Web Sites, with the aim of making more simple his exploration in the complicated electronic paths of the World Wide Web.

Because of the significant share of the professional space the two-way communications (Internet, mobile phones, etc.) occupy, there is a need of cheap communications and cheap information's transmission.

It is evident that the modern man (professional, user) is interested in the comprehensive, classified information, which is provided in an automated way, in order to minimize the costs and the network traffic.

However, there is a need not only for a unified and exemplary handling of the recording of the causes and the effects of a change, but also for the ability of retrieving that change at subsequent moments.

That need led to the generalization of the problem. It led to the search of a standardization of a “model-mechanism” which could be able to observe, understand, record and depict the differences in the state of a system during its action from time to time.

In that project we try to present a model with the ability to describe and record, in a typical way, the changes that take place in a complicated system, so that the retrieval of those changes in the form of a *snapshot* (namely the state of the system in a given moment) or in the form of the *differences* between two snapshots, could be possible.

At the same time we record the acts which could be responsible for the recorded changes, in order to cover in that way as more cases of systems' study as possible.

Therefore, a range of definitions, proposals and ideas comes forward, about the way of thinking we must follow in order to analyze, plan and put in practice such a mechanism. A mechanism for recording and handling the differences that take place in a system during the pass of time; in other words the *Delta Retrieval Modeling*.

In addition, a pseudolanguage appears, a notation, which can be used as a pattern for the projects that could be created with such a model.

The content of the project is divided as follows:

In the first part, there is an introduction to some basic concepts, which are analyzed in the subsequent parts. That part is subdivided in topics about the system theory, automation topics, cybernetics and smart agents topics and in topics about time and temporal databases. Furthermore there are definitions of concepts and technologies that are mentioned in the following chapters and are useful for the understanding of the above.

Subsequently, we have the address of the problem, the explanation for the need, the usage scenarios and also the requirements we must have from such a model.

In the next part are introduced some basic concepts and definitions which could be useful for the understanding of the rest of the text as well as for the acquisition of a spherical view for the technology that is to be introduced in the next part. So, we

analyze the terms of the *system* as well as the terms of the *snapshot*, *the plan*, *the state*, *the fact*, e.t.c

The basic part of the project deals with the description of the model. The most important part of it, is the concept of the *agent*. With the word agent we refer to the mechanism which through a series of functions, fulfils the above-mentioned goal.

These functions are:

1. The observation of the system
2. The recording of the change (in a storing section)
3. The handling / management of the recordings
4. The recovery of the recordings
5. The possible restoration of the system in an old state

A detail description of the above-mentioned functions an

In the last part of the project there is a presentation of an application of the model analyzed above. That application is in the XML language. As we know, the most important part of a document in XML language is the specification of its content, namely of the elements and the attributes of the root element of the document, which in our case is named drml (Delta Retrieving Modeling Language). In order to define



80025 75540

