

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΤΕΧΝΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΔΙΚΤΥΑΚΩΝ ΤΟΠΩΝ

Όνοματεπώνυμα Σπουδαστών

ΖΑΡΚΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΚΟΥΤΣΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ

ΣΤΑΜΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΠΑΤΡΑ – 2013

ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία πραγματοποιήθηκε στο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Πατρών, στο τμήμα Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων. Στόχος αυτής της πτυχιακής είναι η εξέλιξη της τεχνολογίας και η σύνδεση της με το διαδίκτυο.

Θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα καθηγητή μας Δρ. Στάμο Κωνσταντίνο, ο οποίος μας βοήθησε σημαντικά, προκειμένου να ολοκληρωθεί αυτή η εργασία. Τέλος, ευχαριστούμε τις οικογένειες μας και το φιλικό περιβάλλον μας που στάθηκε αρωγός στην περάτωση αυτού του πονήματος.

ΠΕΡΙΛΗΨΗ

Στην παρούσα εργασία θα γίνει μια σύντομη ιστορική αναδρομή του Διαδικτύου, στη συνέχεια θα αναφερθούμε στην ψηφιακή επικοινωνία και τους λόγους για τους οποίους χαρακτηρίζεται αδήριτη η Πληροφορική σαν επιστήμη και η εξέλιξή της, η οποία συνδέεται άρρηκτα με την εν γένει εξέλιξη της κοινωνίας.

Επίσης θα παρουσιαστούν γλώσσες προγραμματισμού, όπως η html, η php, αλλά και η νέα γλώσσα, η οποία είναι ιδιαίτερα δημοφιλής, η XHTML5. Ακόμα, δεν θα παραλειφθούν οι αρχιτεκτονικές, τα πρωτόκολλα και οι βασικές υπηρεσίες που προσφέρει το Διαδίκτυο στους χρήστες του, ενώ θα σημειώνονται και οι βάσεις δεδομένων και τα συστήματα διαχείρισης περιεχομένου, που είναι ιδιαίτερα χρήσιμα.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ.....	1
1.1 Τι είναι πληροφορία	1
1.2 Οριοθέτηση της έρευνας	3
1.3 Μεθοδολογία έρευνας	3
1.4 Δομή της εργασίας	4
ΚΕΦΑΛΑΙΟ 2: ΔΙΑΔΙΚΤΥΟ	6
2.1 Ιστορική Αναδρομή.....	6
2.2 Ψηφιακή επικοινωνία.....	11
2.3 Λόγοι ανάπτυξης.....	13
ΚΕΦΑΛΑΙΟ 3: ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΔΙΑΔΙΚΤΥΟΥ.....	16
3.1 Αρχιτεκτονικές Διαδικτύου-Δίκτυα.....	16
3.2 Πρωτόκολλα Διαδικτύου	19
3.3 Βασικές Υπηρεσίες Διαδικτύου.....	23
ΚΕΦΑΛΑΙΟ 4: ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ.....	27
4.1 Τεχνολογίες Διαδικτύου και Γλώσσες Προγραμματισμού.....	27
<i>Μετάφραση.....</i>	28
<i>Η πληθώρα αλγοριθμικών γλωσσών</i>	30
<i>Ορισμοί Γλωσσών</i>	33
4.1.1 HTML	34
4.1.2 PHP	39
4.1.3 XML.....	42
4.1.4 XHTML	44
4.1.5 JAVASCRIPT.....	47
4.2 Βάσεις Δεδομένων και Διαδίκτυο.....	50
4.2.1 Πλεονεκτήματα Βάσεων Δεδομένων.....	51
4.2.2 Ανεξαρτησία Δεδομένων.....	54
4.2.3 Μια αρχιτεκτονική συστημάτων Βάσεων Δεδομένων.....	56
4.3 Συστήματα Διαχείρισης Περιεχομένου.....	60

ΚΕΦΑΛΑΙΟ 5: ΕΦΑΡΜΟΓΕΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ WEB 2.0.....	64
5.1 Παγκόσμιος Πληροφοριακός Ιστός (World Wide Web-WWW).....	64
5.2 Η μετάβαση από το web 1.0 στο web 2.0.....	65
5.3 Χαρακτηριστικά web 2.0.....	66
5.4 Τεχνολογία web 2.0.....	68
5.5 Εφαρμογές web 2.0	69
5.6 Ηλεκτρονικό Εμπόριο (E-Commerce).....	71
5.6.1 Οφέλη Ανάπτυξης Διαδικτυακών Τόπων.....	72
Σύνοψη - Συμπεράσματα	74
Βιβλιογραφία	76

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

Για πολλούς ανθρώπους η Πληροφορική είναι συνώνυμη με τη νέα τεχνολογία: τη χρήση μηχανών που στηρίζονται στη λειτουργία μικροεπεξεργαστών. Ωστόσο η χρήση εργαλείων που έχουν κατασκευαστεί από τον άνθρωπο για τη συλλογή, αναπαραγωγή, μετάδοση, εγγραφή, ταξινόμηση και αξιοποίηση των πληροφοριών, ανάγεται σε χρόνο πολύ παλιότερο από την εποχή μας - την εποχή της επανάστασης των μικροϋπολογιστών.

Για άλλους, η ιδιαιτερότητα για την εισαγωγή ενός νέου όρου στηρίζεται στην πίστη ότι η επεξεργασία των πληροφοριών μπορεί πια να γίνει σε μια βάση συστηματική και ενοποιημένη.

Οπωσδήποτε, δεν είμαστε ακόμα σε θέση να προτείνουμε έναν ακριβή ορισμό της Πληροφορικής. Για το λόγο αυτό, κρίνεται σκόπιμο να προσπαθήσουμε να παρουσιάσουμε με παραδείγματα όλα όσα σήμερα μοιάζει να μορφοποιούνται κάτω από τον όρο-ομπρέλα Πληροφορική.

1.1 Τι είναι πληροφορία

Οι άνθρωποι έχουν πολύ διαφορετικές ιδέες για το τι είναι πληροφορία. Ακόμη και τα λεξικά δεν συμφωνούν. Άλλα λεξικά την εξισώνουν με τη γνώση. Άλλα τονίζουν τη μετάδοση της γνώσης. Και άλλα δίνουν έμφαση στον ενεργητικό τρόπο άντλησης της γνώσης.

Η αιτία αυτής της διαφοροποίησης στην κοινή χρήση του όρου βρίσκεται στην αφηρημένη έννοια της ίδιας της πληροφορίας, καθώς την αντιλαμβανόμαστε μέσω των αποτελεσμάτων της. Αντλούμε πληροφορίες από στοιχεία (data), από παρατηρήσεις του περιβάλλοντος. Μεταβιβάζουμε πληροφορίες μέσα από την επικοινωνία.

«Πληροφορία είναι η σημασία την οποία ο άνθρωπος αποδίδει σε αναπαραστάσεις γεγονότων και ιδεών, χρησιμοποιώντας τους συμβατικούς όρους αναπαράστασης» (Καρολίδης, 2005). Πρόκειται για έναν αξιόλογο ορισμό, με το μειονέκτημα ότι περιλαμβάνει τη λέξη "σημασία", που είναι κι αυτή τόσο αφηρημένη όσο και η λέξη "πληροφορία". Ένα ενδιαφέρον σημείο, στο οποίο ο προηγούμενος ορισμός προκαλεί την προσοχή μας είναι οι "συμβατικοί όροι αναπαράστασης". Όταν η αναπαράσταση γίνεται με τη βοήθεια της γλώσσας (όπως συμβαίνει συνήθως), η σύνταξη και η σημαντική της γλώσσας σχηματίζουν ένα δεδομένο πλαίσιο για οποιαδήποτε πληροφορία εκφράζεται μέσα από αυτό. Για

παράδειγμα, αν ακούσουμε κάποιον να λέει "είκοσι οχτώ υπό σκιά", μπορούμε να καταλάβουμε ότι μιλά για τη θερμοκρασία, ότι ο αριθμός αναφέρεται σε βαθμούς Κελσίου, ότι η σκιά δεν είναι κάποιου συγκεκριμένου αντικειμένου κλπ. Επομένως, υποθέτοντας πως χρησιμοποιούνται οι συμβατικοί όροι της ελληνικής γλώσσας και κουλτούρας, μπορούμε να αντλήσουμε περισσότερες πληροφορίες από αυτές που μεταφέρονται άμεσα από την αρχική πρόταση.

Οι σημαντικές και συντακτικές όψεις της πληροφορίας έχουν απασχολήσει πολλούς γλωσσολόγους, αλλά δεν έχουν ακόμη οδηγήσει σε έναν γενικά παραδεκτό ορισμό της πληροφορίας. Η πληροφορία φαίνεται να στηρίζεται, πάνω απ' όλα, στην έννοια της "δύναμης επιλογής". Για παράδειγμα, μπορούμε να σκεφτούμε ότι ένας τηλεφωνικός κατάλογος περιέχει μεγάλη ποσότητα πληροφοριών γιατί κάθε εγγραφή επιλέγει ένα πρόσωπο, μια οικογένεια ή έναν οργανισμό μέσα από έναν πολύ μεγάλο αριθμό ατόμων, οικογενειών και οργανισμών που βρίσκονται στη γεωγραφική περιοχή που καλύπτει ο κατάλογος. Επίσης, συνδέει αυτό το πρόσωπο με μια μοναδική επιλογή ψηφίων (τον αριθμό τηλεφώνου) μέσα από εκατομμύρια δυνατών συνδυασμών αυτών των ψηφίων. Ακόμη περισσότερο, ο κατάλογος επαναλαμβάνει αυτή τη διαδικασία επιλογής για όλους τους συνδρομητές. Η ταχυδρομική διεύθυνση κάθε συνδρομητή αποτελεί επίσης παράδειγμα της δύναμης επιλογής της πληροφορίας. Η διεύθυνση προσδιορίζεται από μια διαδικασία αύξουσας εξειδίκευσης: περιοχή, πόλη, οδός, αριθμός κλπ.

Ας σημειωθεί ότι πίσω από την επιλεκτική όψη της πληροφορίας βρίσκεται η υπόθεση ότι υπάρχει ένας πεπερασμένος, αν και μεγάλος, αριθμός εναλλακτικών δυνατοτήτων επιλογής, ο οποίος είναι γνωστός και σ' αυτόν που παρέχει και σ' αυτόν που χρησιμοποιεί τις πληροφορίες. Έτσι, στην περίπτωση των ονομάτων ενός τηλεφωνικού καταλόγου, οι εναλλακτικές δυνατότητες βασίζονται στους διάφορους τρόπους με τους οποίους μπορούν να συνδυαστούν τα γράμματα του αλφαβήτου. Αλλά το σύνολο των δυνατών γνώσεων - έννοιες, ιδέες, γεγονότα, ονόματα κλπ. – είναι πρακτικά απεριόριστο.

Αυτή η προφανής διαφορά ανάμεσα σε εκείνα που μπορεί να θέλουμε να μεταδώσουμε και στον τρόπο μετάδοσης τους, διευθετείται με τον χωρισμό του περιεχομένου της πληροφορίας από την αναπαράσταση της.

Η αναπαράσταση (σύμβολα, σημεία, σινιάλα κλπ.) μπορεί, λοιπόν, να ανήκει σε ένα εντελώς καθορισμένο, πεπερασμένο σύνολο (το αλφάβητο, τελείες και παύλες κλπ). Σε αυτή την περίπτωση, η διαδικασία μετάδοσης των πληροφοριών αποτελεί διαδικασία μετάδοσης

της αναπαράστασης τους. Νέες ιδέες θα μεταδίδονται με νέους συνδυασμούς παλιών σημάτων. Αυτός ο αντικειμενικός τρόπος αντιμετώπισης της μετάδοσης των πληροφοριών είναι που ενδιαφέρει τους μηχανικούς (και τους θεωρητικούς) των τηλεπικοινωνιών και αποτελεί τη βάση της θεωρίας των πληροφοριών (Καρολίδης, 2005).

1.2 Οριοθέτηση της έρευνας

Από τα παραπάνω, κατέστη σαφές ότι είναι δύσκολο να οριοθετήσει κανείς επακριβώς την επιστήμη της Πληροφορικής. Το γεγονός αυτό οφείλεται στην ίδια τη φύση της εν λόγω επιστήμης, καθώς βασίζεται στην έννοια της «πληροφορίας». Για το λόγο αυτό, συχνά επιλέγει κανείς να εξετάσει διάφορες επιμέρους πτυχές ή θεματικά αντικείμενα που άπτονται της Πληροφορικής.

Η παρούσα λοιπόν εργασία καταπιάνεται με θέματα που άπτονται του Διαδικτύου. Πιο συγκεκριμένα, επιχειρείται να πραγματοποιηθεί μία εκτενής ιστορική αναδρομή στο θεμελιώδες αυτό για τη σημερινή πραγματικότητα επίτευγμα, καθώς επίσης να παρουσιαστεί η πληθώρα των εφαρμογών που απορρέουν του Διαδικτύου.

Στα πλαίσια αυτά, θα αναφερθούμε στις αρχιτεκτονικές του Διαδικτύου και τα πρωτόκολλά του. Επιπλέον, θα αναφερθούμε σε γλώσσες προγραμματισμού, όπως html, php, xml, xhtml, javascript, css και dhtml, ενώ δε θα παραλειφθούν οι Βάσεις Δεδομένων (ΒΔ) και τα συστήματα διαχείρισης περιεχομένου.

Τέλος, δεδομένου ότι ένα από τα πλέον δημοφιλή προς μελέτη θέματα τα τελευταία χρόνια στο χώρο του Διαδικτύου και της Πληροφορικής αποτελεί η μετάβαση από το παραδοσιακό web 1.0 στο web 2.0, θα παρατεθούν ορισμένες βασικές πληροφορικές για το θέμα αυτό. Πρόκειται για την παροχή δεύτερης γενιάς υπηρεσιών βασισμένων στο Διαδίκτυο, με έμφαση στην ηλεκτρονική συνεργασία και την επικοινωνία των χρηστών.

1.3 Μεθοδολογία έρευνας

Η έννοια της μεθοδολογίας, αναφέρεται στον τρόπο με τον οποίο συλλέγονται τα αναγκαία στοιχεία και οι απαιτούμενες πληροφορίες. Για την εκπόνηση λοιπόν της πτυχιακής αυτής εργασίας, επιλέχθηκε να λάβει χώρα μία πρωτογενής ποιοτική έρευνα.

Ως βάση άντλησης των πληροφοριών χρησιμοποιήθηκε αφενός το Διαδίκτυο, αλλά και έντυπο υλικό. Σε κάθε περίπτωση η αναφορά σε πρωτογενές υλικό ολοκληρώνεται με την παραπομπή στην αντίστοιχη αναφορά στο τελευταίο κεφάλαιο («Βιβλιογραφία»).

Ως βασική μεθοδολογία στα πλαίσια της ποιοτικής αυτής έρευνας είναι η παρατήρηση και η κριτική σκέψη σε ένα σύνολο μελετών, απόψεων και άρθρων που είναι διαθέσιμα στο ευρύ κοινό. Με αυτή την προσέγγιση, ο χαρακτηρισμός «πρωτογενής, τυπική μελέτη» δεν είναι απόλυτος, διότι αρκετές φορές, παρουσιάζονται στην πορεία της εργασίας θέματα και συμπεράσματα που βασίζονται σε υπάρχουσες δημοσιευμένες θέσεις.

Το τελικό παραγόμενο αποτέλεσμα είναι η εργασία αυτή. Όχι μόνο μία «Θεωρητική» εργασία, αλλά μία εργασία «Εκτενούς Ανασκόπησης» αναλύοντας σε βάθος και προβάλλοντας τις προεκτάσεις της τεχνολογίας του Διαδικτύου, καθώς επίσης και των εφαρμογών του.

1.4 Δομή της εργασίας

Η εργασία αποτελείται από πέντε (5) αυτοτελή κεφάλαια. Το πρώτο εξ αυτών είναι εισαγωγικό και περιλαμβάνει την οριοθέτηση της εργασίας, τον προσδιορισμό του αντικειμένου μελέτης, όπως επίσης και τη μεθοδολογία ανάπτυξης. Πιο συγκεκριμένα, η εργασία αυτή εξετάζει τα θέματα που άπτονται της τεχνολογίας Διαδικτύου.

Αναφορικά με τη μεθοδολογία, πρόκειται κατά βάση για μία ποιοτική, πρωτογενής έρευνα, με τελικό αποτέλεσμα την ανάπτυξη μίας βιβλιογραφικής εργασίας εκτενούς ανασκόπησης.

Το δεύτερο κεφάλαιο φέρει τον τίτλο «Διαδίκτυο» και περιλαμβάνει αναλυτικά όλες τις βασικές πληροφορίες που συνθέτουν την ανάπτυξη και τη δημιουργία του γνωστού σε όλους Διαδικτύου.

Το επόμενο κεφάλαιο ονομάζεται «Βασικά Στοιχεία Διαδικτύου» και σε αυτό επιχειρείται η παρουσίαση του θέματος από τεχνολογική άποψη, ως προς τον τρόπο δόμησης και λειτουργίας. Στα πλαίσια αυτά παρατίθεται η χρησιμοποιούμενη αρχιτεκτονική, τα πρωτόκολλα επικοινωνίας, αλλά και οι βασικές παρεχόμενες υπηρεσίες.

Το τέταρτο κατά σειρά κεφάλαιο ονομάζεται «Τεχνολογίες Διαδικτύου» και αναφέρεται μεταξύ άλλων σε ζητήματα προγραμματισμού στο Διαδίκτυο και Βάσεις Δεδομένων.

Το επόμενο κεφάλαιο φέρει τον τίτλο «Εφαρμογές και Τεχνολογίες Web 2.0» και σε αυτό επιχειρείται να καταγραφεί επιγραμματικά η πληθώρα των εφαρμογών που απορρέουν του Διαδικτύου, όπως επίσης και η μετάβαση από το web 1.0 στο περιβάλλον web 2.0 με έμφαση στις συνέπειες και τα οφέλη για τον τελικό χρήστη. Με αυτή την προσέγγιση εκτιμάται ότι προσεγγίζεται το θέμα που επιλέχθηκε πιο σφαιρικά με την έννοια ότι εξετάζεται το ζήτημα τεχνολογικά, αλλά και πρακτικά. Άλλωστε, αρκεί κανείς να αντιληφθεί ότι ο μη Τεχνικός αναγνώστης, αλλά και ο απλός χρήστης του Διαδικτύου ενδιαφέρεται πρωτίστως για τις εφαρμογές ενός μέσου, παρά για τον τρόπο λειτουργίας αυτού.

Το τελευταίο κεφάλαιο χρησιμοποιείται για τη σύνοψη των κυριότερων σημείων της εργασίας και την παράθεση των συμπερασμάτων. Τέλος, στην ενότητα «Βιβλιογραφία» καταγράφονται οι πρωτογενείς πηγές αναφοράς.

ΚΕΦΑΛΑΙΟ 2: ΔΙΑΔΙΚΤΥΟ

2.1 Ιστορική Αναδρομή

Οι μαθηματικοί υπολογισμοί συνάρπαζαν τον άνθρωπο από τα πιο παλιά χρόνια, και, μέχρι πριν από μερικούς ακόμη αιώνες, η υπολογιστική ικανότητα θεωρούνταν -περισσότερο και από τη γραφή- ένα είδος τέχνης, που μόνο λίγοι, εκλεκτοί, ήταν ικανοί να ασκούν. Σήμερα θεωρείται αυτονόητο το να έχει κάποιος απόλυτη και αυτόματη ευχέρεια στις τέσσερις βασικές πράξεις της αριθμητικής, τουλάχιστον όσο πρόκειται για ακέραιους αριθμούς, και μας είναι δύσκολο να κατανοήσουμε ότι τα πράγματα δεν ήταν πάντοτε έτσι και ότι αυτή η εξέλιξη είναι δημιούργημα των δύο τελευταίων αιώνων. Ούτε και έχουμε επίγνωση πόσο πολύπλοκο εγχείρημα είναι ακόμη και η απλούστερη πράξη, η πρόσθεση, όταν εμπλέκονται πολυψήφιοι αριθμοί. Τα πράγματα ήταν πολύ χειρότερα μέχρι και το τέλος του Μεσαίωνα, επειδή οι υπολογισμοί δε γίνονταν με το δικό μας δεκαδικό σύστημα, αλλά με λατινικούς αριθμούς, κάτι που περιέπλεκε πολύ περισσότερο την κατάσταση. Μόλις το 1524 ο Adam Riese (1492-1559) εξέδωσε μια "Προπαίδεια" όπου περιέγραφε τους κανόνες των πράξεων σύμφωνα με το ινδικής προέλευσης δεκαδικό σύστημα. Από την εποχή εκείνη το δεκαδικό σύστημα επικράτησε σε ολόκληρη την Ευρώπη! (Πολλάλης, 1999)

Σήμερα, ο καθένας μας εκτελεί τις τέσσερις πράξεις της αριθμητικής, χωρίς να ξέρει για ποιο λόγο η μέθοδος που ακολουθεί οδηγεί πάντοτε σε ορθό αποτέλεσμα. Πώς εξηγείται αυτό; Δε θα έπρεπε άραγε να "κατανοήσουμε" πρώτα από μαθηματική άποψη τη μέθοδο της διαίρεσης, για παράδειγμα, για να μπορούμε να την εφαρμόσουμε; Και όμως, κάτι τέτοιο δεν είναι απαραίτητο, επειδή στην ουσία πρόκειται για μια διαδικασία που εκτελείται μηχανικά, και θα μπορούσε επομένως να διεκπεραιωθεί καταρχήν και από μια μηχανή. Δεν είναι ανάγκη να γνωρίζουμε τις μαθηματικές αρχές που "περιέχονται" στη μέθοδο, ούτε και να νοιαζόμαστε για ειδικές περιπτώσεις στις οποίες η μέθοδος δε δίνει αποτελέσματα, επειδή ξέρουμε από το σχολείο και από την πείρα μας ότι λειτουργεί πάντοτε.

Αυτή η έννοια της μηχανικά εκτελέσιμης υποκοριστικής διαδικασίας, αποτελεί τον πυρήνα της πληροφορικής. Η διαδικασία αυτή ονομάζεται «αλγόριθμος» λέξη που προέρχεται από τον Πέρση μαθηματικό Abu Ja'far Mohammed ibn Musa Al-Khowarismi, που έγραψε ένα ονομαστό βιβλίο περί άλγεβρας.

Στη συνέχεια όμως η θεωρία των αλγορίθμων πήρε μια περιθωριακή θέση στα μαθηματικά. Πολύ λίγοι μαθηματικοί ενδιαφέρθηκαν για μηχανικές υπολογιστικές διαδικασίες και, ως τον αιώνα μας, με τον όρο "αλγόριθμος" εννοούνταν κυρίως οι αλγόριθμοι των τεσσάρων στοιχειωδών πράξεων. Ένας λόγος για τούτο είναι ότι οι μαθηματικοί ενδιαφέρονταν κυρίως για νόμους παρά για μεθόδους ή διαδικασίες, δηλαδή περισσότερο για την εξεύρεση σχέσεων μεταξύ μαθηματικών μεγεθών, παρά για τον υπολογισμό αριθμητικών τιμών (Καρολίδης, 2005). Ένας άλλος λόγος είναι ότι προτού εμφανιστεί ο ηλεκτρονικός υπολογιστής δεν ήταν δυνατή η εκτέλεση περίπλοκων αλγορίθμων με χιλιάδες ή και εκατομμύρια βήματα. Έτσι, ο ηλεκτρονικός υπολογιστής άνοιξε νέους κόσμους στην "αλγοριθμική", όπως ακριβώς το τηλεσκόπιο στην αστρονομία και το μικροσκόπιο στη βιολογία και στην ιατρική. Ωστόσο, στην αναλογία μας υπάρχει μια σημαντική διαφορά: το τηλεσκόπιο και το μικροσκόπιο, όσο καθοριστική κι αν ήταν η σημασία τους ως οργάνων, δε δημιούργησαν τα ίδια μια νέα επιστήμη. Προφανώς ο ηλεκτρονικός υπολογιστής είναι κάτι περισσότερο από μια απλή υπολογιστική μηχανή. (Πολλάλης, 1999)

Παρά το ότι οι αλγόριθμοι δεν υπήρξαν ουσιώδες αντικείμενο της μαθηματικής έρευνας, ορισμένοι μαθηματικοί του παρελθόντος προσπάθησαν να κατασκευάσουν μηχανικές υπολογιστικές μηχανές για τις τέσσερις βασικές πράξεις. Ανάμεσα στις γνωστότερες συγκαταλέγονται οι μηχανές του von Schickard (1592-1635), του Pascal (1623-1662) και του Leibniz (1646-1716). Όλες αυτές οι προσπάθειες όμως δε στέφθηκαν από επιτυχία, γιατί οι λεπτουργικές δυνατότητες της εποχής δεν ήταν τόσο αναπτυγμένες που να επιτρέπουν για μεγάλο χρονικό διάστημα την αλάνθαστη επανάληψη υπολογιστικών βημάτων με γρανάζια ακριβείας. Μόλις στον αιώνα μας εμφανίστηκαν οι μηχανικές αριθμομηχανές γραφείου που έκαναν πιο εύκολη την υπολογιστική εργασία σε τράπεζες, τεχνικά γραφεία, και άλλες επιχειρήσεις (Ζαφείρη, 2005).

Όμως, καμιά από τις μηχανές αυτές δε διαθέτει ένα βασικό στοιχείο του ηλεκτρονικού υπολογιστή: τη δυνατότητα προγραμματισμού. Οι υπολογιστικές μηχανές μπορούσαν βεβαίως να εκτελούν τους αλγορίθμους των τεσσάρων βασικών πράξεων της αριθμητικής, αλλά τίποτε περισσότερο. Για το σκοπό αυτόν είχαν τέσσερα προγράμματα, τα οποία εκτελούσαν με κίνηση των τροχών τους και τα οποία ήταν «αναπόσπαστο στοιχείο» της κατασκευής τους.

Τόσο η δυνατότητα προγραμματισμού όσο και η αποθήκευση προγραμμάτων σε υπολογιστή έχουν και αυτές τους προδρόμους τους. Το 1805, στη Γαλλία, ο Jacquard εφεύρε έναν αργαλειό που ελεγχόταν από ένα σύστημα διάτρητων ξύλινων πλακιδίων. Το 1886, ο Hermann Hollerith εφεύρε τις διάτρητες κάρτες, ενώ γνωστή από παλιά είναι και η αποθήκευση προγραμμάτων σε μουσικά «αυτόματα μηχανήματα» με κυλίνδρους που φέρουν καρφιά. Οι τρύπες στο χαρτί ή τα καρφιά στους κυλίνδρους κινούνται πλάι σε μηχανικούς αισθητήρες, οι οποίοι μπορούν να τα ψηλαφίσουν. Πρόκειται για τις πρώτες συσκευές αποθήκευσης δεδομένων: τα δεδομένα "καταχωρούνται" μόνο μια φορά, κατά την κατασκευή, με τη μορφή της διάτρησης πάνω στο χαρτί, και στη συνέχεια μπορούν να «διαβάζονται» οποτεδήποτε και οσοδήποτε συχνά. Για το λόγο αυτό, μια τέτοια μνήμη ονομάζεται και μνήμη ανάγνωσης.

Ο σπουδαιότερος πρόδρομος της πληροφορικής είναι ο Άγγλος καθηγητής των Μαθηματικών Charles Babbage (1792-1871). Ο Babbage όχι μόνο εφεύρε μια μηχανική υπολογιστική μηχανή, όπως και άλλοι προηγούμενοι του, αλλά είχε και την ιδέα να καθοδηγήσει τη λειτουργία της με διάφορα αποθηκευμένα προγράμματα, κατά την αρχή του Jacquard και οι δικές του προσπάθειες όμως προσέκρουσαν στις αδυναμίες της τεχνικής υλοποίησης και λησμονήθηκαν. Ωστόσο, η χρήση των διάτρητων καρτών επεκτάθηκε και οδήγησε σε νέες εφαρμογές, οι οποίες ξεπέρασαν την απλή αποθήκευση δεδομένων (Πολλλάλης, 1999). Σύντομα, μηχανές με διάτρητες κάρτες χρησιμοποιούνταν σε γραφεία και εργοστάσια και για την ταξινόμηση, πινακοποίηση, και πρόσθεση δεδομένων.

Ένα νέο βήμα προόδου σημειώθηκε το 1934, όταν ο Γερμανός μηχανικός Konrad Zuse (γενν. 1910) είχε την ιδέα να εφαρμόσει το δυαδικό αριθμητικό σύστημα στις μηχανικές υπολογιστικές μηχανές. Μέχρι τότε χρησιμοποιούσαν αποκλειστικά το, αυτονόητο για μας, δεκαδικό σύστημα και τούτο απαιτούσε, για κάθε ψηφίο, δέκα διαφορετικές μηχανικές θέσεις υπό τη μορφή γранаζιών ή ράβδων. Και ο Leibniz είχε, στην εποχή του, ασχοληθεί με το δυαδικό σύστημα, αλλά ο Zuse δεν το ήξερε αυτό. Το δυαδικό σύστημα έχει μόνο δύο αριθμούς, τους 0 και 1, πράγμα που σημαίνει μεγάλη απλοποίηση - σε αντίβαρο, όμως, οι δυαδικοί αριθμοί έχουν πολύ περισσότερα ψηφία από τους δεκαδικούς αντίστοιχους τους, και ένας άνθρωπος θα δυσκολευόταν πολύ να τους γράφει ή να τους διαβάζει σε μόνιμη βάση. Την εποχή εκείνη, ιδίως εξαιτίας της τηλεφωνίας, η τεχνική των ηλεκτρονόμων είχε Φτάσει σε υψηλό επίπεδο. Ηλεκτρονόμος (ρελέ) είναι, με απλά λόγια, ένας τηλεχειριζόμενος ηλεκτρομηχανικός διακόπτης, που μπορεί να ανοίξει ή να κλείσει μία ή περισσότερες επαφές

ταυτόχρονα μέσα σε κλάσματα του δευτερολέπτου. Και εδώ ήρθε η μεγάλη έμπνευση: οι δύο θέσεις της επαφής ενός ηλεκτρονόμου (ανοιχτή, κλειστή) μπορεί να θεωρηθούν ως αντίστοιχες με τους δυαδικούς αριθμούς 0 και 1. Κατά το δεύτερο παγκόσμιο πόλεμο, στις αρχές της δεκαετίας του '40, ο Zuse κατασκεύασε έναν υπολογιστή με ηλεκτρονόμους που στηριζόταν σε αυτήν την αρχή και ελεγχόταν από πρόγραμμα μέσω ατέρμωνων διάτρητων ταινιών που περιπλέκονταν σε βρόχους. Έτσι αρχίζει η εποχή των ηλεκτρονικών υπολογιστών.

Επίσης κατά τη διάρκεια του Δεύτερου Παγκόσμιου Πόλεμου, το 1944, κατασκευάστηκε στις ΗΠΑ από τον Αϊκεν μια μηχανή που ονομάστηκε Mark I και λειτουργούσε με παρόμοιο τρόπο, αν και χρησιμοποιούσε ακόμη το δεκαδικό σύστημα. Στο μεταξύ είχε διαπιστωθεί ότι κατάλληλες να χρησιμοποιηθούν ως διακόπτες είναι και οι ηλεκτρονικές λυχνίες, οι οποίες τότε είχαν εφαρμογή στη ραδιοφωνία. Οι λυχνίες δεν περιλαμβάνουν μηχανικά κινητά μέρη, και κατά συνέπεια ο χρόνος για την εκτέλεση μιας διέλευσης από το κύκλωμα έγινε δυνατό να μειωθεί από την τάξη μεγέθους των 100 mS (mS = χιλιοστό του δευτερολέπτου), που ήταν στους ηλεκτρονόμους, στα 0.1 mS, δηλαδή κατά 1000 περίπου φορές.¹ Σύμφωνα με την αρχή αυτή, οι Eckert και Mauchly κατασκεύασαν το 1946, στο Πανεπιστήμιο της Πενσυλβανίας, τον πρώτο ηλεκτρονικό υπολογιστή, τον ENIAC. Το όνομα του προέρχεται από τα αρχικά των λέξεων electronic numerical integrator and computer (ηλεκτρονικός αριθμητικός ολοκληρωτής και υπολογιστής), πρόκειται δηλαδή για αρκτικόλεξο. Από τότε τα αρκτικόλεξα έγιναν μόδα στην πληροφορική.

Όπως έχει συμβεί και άλλοτε στην ιστορία της τεχνικής, στην περίπτωση αυτή είχαμε σε δύο διαφορετικά μέρη του κόσμου και την ίδια σχεδόν στιγμή δύο εντελώς όμοιες εφευρέσεις. Τίθεται κατά συνέπεια το ερώτημα της προτεραιότητας. Ποιος είναι ο εφευρέτης του ηλεκτρονικού υπολογιστή; Τα ευρήματα του Zuse προηγήθηκαν των Αμερικανών, εκείνοι όμως χρησιμοποίησαν πρώτοι την ηλεκτρονική. Στο ευρύτερο κοινό ο υπολογιστής έγινε γνωστός σαν καθαρά αμερικανική εφεύρεση, επειδή οι Γερμανοί εργάζονταν υπό καθεστώς μυστικότητας. Κατά συνέπεια, οι Αμερικανοί αγνοούσαν για πολύ καιρό τον Zuse και ο δρόμος προς τα μεγάλα υπολογιστικά συστήματα των επόμενων δεκαετιών καθορίστηκε από τις αμερικανικές προόδους.

Υπάρχει όμως και κάτι άλλο που ξεχωρίζει τις εφευρέσεις του Zuse και των Αμερικανών από τους σημερινούς υπολογιστές: ο τρόπος της αποθήκευσης προγραμμάτων. Θυμίζουμε ότι το πρόγραμμα πρέπει να αποθηκεύεται και ότι ο Zuse πέτυχε αυτό με

διάτρητες ταινίες. Στον ENIAC χρησιμοποιήθηκαν πίνακες ανάλογοι με εκείνους των τηλεφωνικών κέντρων, που ο χειρισμός τους γινόταν με το χέρι. Και οι δύο λύσεις κάθε άλλο παρά ικανοποιητικές ήταν διότι δεν προσέφεραν καμιά ευελιξία, και, αν είχαμε αρκεστεί σε αυτές, η παραπέρα εξέλιξη της τεχνολογίας των υπολογιστών θα έμενε ασφυκτικά περιορισμένη. Οι πρώτοι υπολογιστές όμως χρειάζονταν μνήμη και για τα ενδιάμεσα αποτελέσματα, στον υπολογισμό των οποίων ακολουθούσαν επίσης την τεχνική των ηλεκτρονόμων ή των λυχνιών" εδώ ήταν που ο ουγγρικής καταγωγής Αμερικανός μαθηματικός John von Neumann (1902-1957) είχε την ιδέα να αποθηκεύσει το πρόγραμμα στη μνήμη του υπολογιστή. Έτσι, η πρόσβαση στα επιμέρους βήματα του προγράμματος έγινε όχι μόνο πολύ ταχύτερη απ' ό,τι με τις διάτρητες ταινίες, αλλά και πολύ πιο ευέλικτη. Όμως, το σημαντικότερο ήταν ότι, με τον τρόπο αυτόν το πρόγραμμα καθώς και τα ενδιάμεσα αποτελέσματα έμεναν στη μνήμη, εκτελούνταν απευθείας από τον υπολογιστή, και μάλιστα μπορούσαν να μεταβληθούν από ένα άλλο πρόγραμμα. Ο von Neumann είχε εκθέσει γραπτώς την ιδέα του αυτή ήδη από το 1946, αλλά μόλις το 1949 μπόρεσε να την υλοποιήσει, στην υπολογιστική συσκευή EDSAC του Πανεπιστημίου του Καίμπριτζ στην Αγγλία (Πολλάλης, 1999). Με την αποθήκευση του προγράμματος στη μνήμη ολοκληρώνεται αυτό που ονομάζουμε σήμερα ηλεκτρονικό υπολογιστή, ή ακριβέστερα υπολογιστή τύπου von Neumann, για να τον διακρίνουμε από τις μεταγενέστερες εξελίξεις.

Αυτοί που πρώτοι κυρίως συνέβαλαν στην θεωρία πληροφοριών – κωδίκων, δηλ. που σε αυτούς οφείλεται η γένεση της, ήταν οι μαθηματικοί C. E. Shannon και N. Wiener αμέσως μετά τον Β' Παγκόσμιο πόλεμο. Μετά την συμβολή αυτή των Shannon και Wiener, ένας μεγάλος αριθμός από επιστημονικά άρθρα ήλθε στο φως στη σχετική με το αντικείμενο βιβλιογραφία σ' ένα μικρό χρονικό διάστημα. Το γεγονός αυτό δείχνει την μεγάλη σπουδαιότητα και τον ενθουσιασμό των επιστημόνων στο νέο για την εποχή, και ίσως και για σήμερα, πεδίο. Επίσης αυτό δείχνει την μεγάλη σημασία που αποδίδουν οι επιστήμονες στο πεδίο αυτό. Είναι αδύνατο να αναφερθούμε στα άρθρα αυτά. Για ιστορικούς όμως λόγους, μπορούμε ν' αναφέρουμε εδώ μερικά επιστημονικά ονόματα που συνέβαλαν πάρα πολύ στην θεωρία κωδίκων, όπως οι: R. W. Hamming, P. Elias, R. M. Fano, D. Huffman, N. Abramson P. Fire, I. M. Jacobs, S. Golomb, A. Viterbi, E. Berlekamp, F. Reza, D. M. Hageibarger, D.C.Voukalis κ.α.

Τα ονόματα που αναφέραμε είναι μόνο ένας μικρός αριθμός από τους μαθηματικούς και τους επικοινωνιακούς επιστήμονες που συνέβαλαν στην θεωρία κωδίκων. Ένας μεγάλος

αριθμός από επιστήμονες ασχολείται ακόμα και σήμερα στο σπουδαίο αυτό πεδίο, που δεν σταμάτησε ν' αναπτύσσεται, και η συμβολή τους σ' αυτό φαίνεται αρκετά μεγάλη. Δεν κρίνουμε εδώ αναγκαίο ν' αναφέρουμε ονόματα, αλλά ο αναγνώστης μπορεί να βρει πολλά περισσότερα γύρω από την ιστορική εξέλιξη της θεωρίας κωδίκων και τους επιστήμονες που συμβάλλουν καθημερινά στην θεωρία κωδίκων και την σημερινή της ανάπτυξη στα επιστημονικά περιοδικά: IRE Transactions of Information Theory, IEEE Transactions on Information Theory, International Journal of Electronic. Beel System Technical Journal, κ.α.

Από τότε που ήρθε στο φώς η εργασία του Shannon (1948), η θεωρία κωδίκων, έφθασε σε περιόδους υψηλής ευφορίας με ανακαλύψεις διαφόρων κατηγοριών κωδίκων, αλγορίθμων αποκωδικοποίησης και με υποσχέσεις επαναστατικοποίησης των ψηφιακών επικοινωνιών.

Βλέπουμε ν' αυξάνουν καθημερινά οι εφαρμογές των κωδίκων στους επικοινωνιακούς δίαυλους με διαφορετικές τεχνικές μεθόδους με σκοπό να ελαττωθούν τελικά το κόστος και η πολυπλοκότητα τους και να βελτιωθεί η απόδοση τους. Στον τομέα αυτό δεν έχει ακόμη επέλθει καμία επαναστατική πρόοδος.

Ο βασικός σκοπός ήταν να μεταφραστούν οι ιδέες του Shannon σε μηχανικά συστήματα και ν' αποδειχθεί ότι η θεωρία κωδίκων είναι απλή και αποδοτική σε ανάλογες εφαρμογές της ψηφιακής επικοινωνίας. Έχουν βέβαια βρεθεί πολλά τέτοια συστήματα. Άλλα είναι εντελώς θεωρητικά και μερικά πρακτικά, χωρίς φυσικά ν' αποκλείεται και σήμερα η ανακάλυψη ενός επαναστατικού συστήματος που θα επιλύει πρακτικά το πρόβλημα της θεωρίας κωδίκων, οπωσδήποτε όμως παραμένουν ακόμη σ' αυτήν πολλά άλυτα προβλήματα. Έτσι φαίνεται ότι θα υπάρξει μια μελλοντική έρευνα στην κατασκευή κωδίκων, την ανάλυση τους και τέλος την οικονομική πρακτική εφαρμογή τους. (Norton, 1995).

2.2 Ψηφιακή επικοινωνία

Η πληροφορική συνδέεται στενά με τους ηλεκτρονικούς υπολογιστές. Όσο δεν υπήρχαν ηλεκτρονικοί υπολογιστές, δεν υπήρχε ούτε και πληροφορική -ορισμένες φορές μάλιστα, η πληροφορική ορίζεται ως η επιστήμη των υπολογιστών. Η λέξη "πληροφορική" είναι σύνθετη τεχνητή λέξη, προερχόμενη από τη λέξη "πληροφορία" και την κατάληξη "-ική" (όπως στην "ιατρική"). Εμφανίστηκε για πρώτη φορά κατά τη δεκαετία του '60 στη Γερμανία ("Informatik") και στη Γαλλία ("Informatique") και έγινε πολύ γρήγορα αποδεκτή

για τη νέα επιστήμη που βρισκόταν ακόμη στα σπάργανα. Στον αγγλόφωνο χώρο, ο αντίστοιχος όρος "informatics" δεν έχει γίνει μέχρι σήμερα ιδιαίτερα αποδεκτός" αντίθετα, χρησιμοποιείται ο όρος "computer science", κατά λέξη δηλαδή "επιστήμη των υπολογιστών". Ωστόσο, πολλοί ειδικοί της πληροφορικής θεωρούν άστοχη την υπερβολικά στενή σύνδεση με τον υπολογιστή ως συσκευή, και γι' αυτό προτιμούν να κάνουν λόγο για "computing science", που θα μπορούσε να μεταφραστεί ως η επιστήμη των υπολογισμών. Για την ώρα ας μην ασχοληθούμε με το αν η λύση αυτή είναι καλύτερη, ούτε και με το τι σημαίνει, τελικά, ο όρος "computing" σε σχέση με την πληροφορική. Την απάντηση θα τη δώσει το βιβλίο αυτό, στο σύνολο του.

Ο ηλεκτρονικός υπολογιστής γεννήθηκε στον αιώνα μας, στη δεκαετία του 1940. Πρόδρομοι του είναι οι μηχανικές υπολογιστικές μηχανές και η τεχνική υπολογισμού με διάτρητες κάρτες. Αλλά και στον κόσμο των ιδεών, η πληροφορική έχει τους προγόνους της, που μπορούμε να τους ανιχνεύσουμε ως τα βάθη του χρόνου. Μπορούμε μάλιστα να κάνουμε διάκριση ανάμεσα στην προϊστορία και την ιστορία της πληροφορικής. (Norton, 1995).

Όταν λέμε ψηφιακή επικοινωνία θεωρούμε τις εξής περιπτώσεις (Καρολίδης, 2005):

1. Την επιλογή σ' αυτόματα τηλεφωνικά ή τηλεγραφικά δίκτυα. Πολλοί συνδρομητές επιλέγουν ταυτόχρονα και χρησιμοποιούν παράλληλες δρομολογήσεις σ' όλο το σύστημα του δικτύου. Μια εσφαλμένη δρομολόγηση θα επιδράσει μόνο σ' ένα συνδρομητή κάθε φορά και δεν θα επιδράσει σ' όλο ανεξαρτήτως το σύστημα. Τα επικοινωνιακά κανάλια, υποτίθεται ότι δεν έχουν ραδιοκυκλώματα, και δεν είναι τόσο επιρρεπή στις εξωτερικές επιδράσεις.
2. Η διαβίβαση της ψηφιακής πληροφορίας σε τηλεγραφικά κανάλια μεγάλων αποστάσεων αποτελείται από πολλούς φορείς. Ασυρματικοί φορείς είναι επιρρεπείς στις εξωτερικές επιδράσεις λ.χ. ραδιοεπιδράσεις.
3. Η διαβίβαση των πληροφοριών στους υπολογιστές. Οι πληροφορίες περνούν δια μέσου των ιδίων φορέων πολλές φορές πριν να ληφθεί το πραγματικό αποτέλεσμα. Οι φορείς αυτοί δεν είναι σε μεγάλο αριθμό επιρρεπείς στις εξωτερικές επιδράσεις, αλλά ο αριθμός των πληροφοριών που περνούν από τα ίδια στοιχεία μπορεί να είναι τόσο μεγάλος που και είναι πολύ μικρό φαινόμενο σε κάποιο στοιχείο μπορεί να επιδράσει στον ορθό χειρισμό της αυτόματης συσκευής.

Η εκπομπή της ψηφιακής πληροφορίας μπορεί να προστατευθεί με την λήψη των σφαλμάτων ή και την διόρθωση τούτων με ειδικούς κώδικες (Καλαφατούδης, 2004).

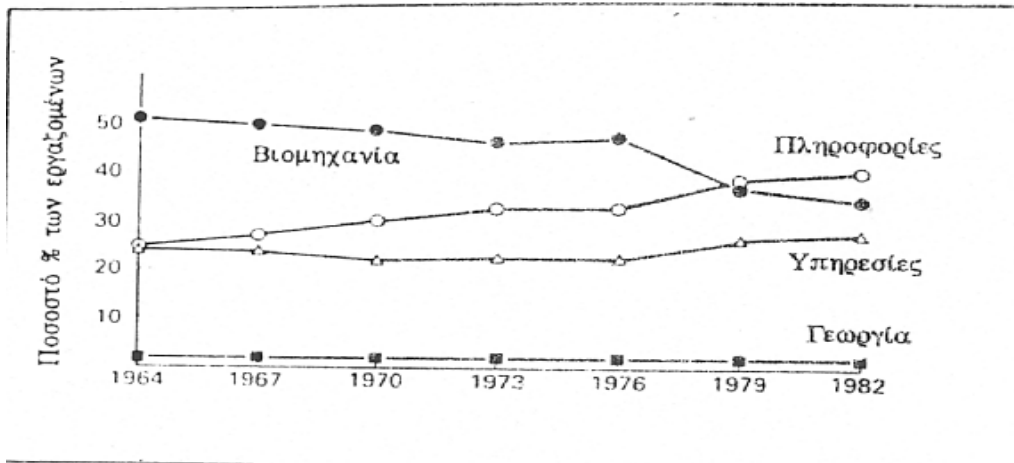
2.3 Λόγοι ανάπτυξης

Υπάρχουν πολλοί λόγοι για τους οποίους η Πληροφορική έχει γίνει αντικείμενο πλατιάς συζήτησης και μελέτης. Καθένας από τους λόγους αυτούς είναι σημαντικός από μόνος του, αλλά όλοι μαζί δημιουργούν μια πειστική. ανάγκη να κατανοήσουμε τα τεχνικά και κοινωνικά θέματα που ανακύπτουν.

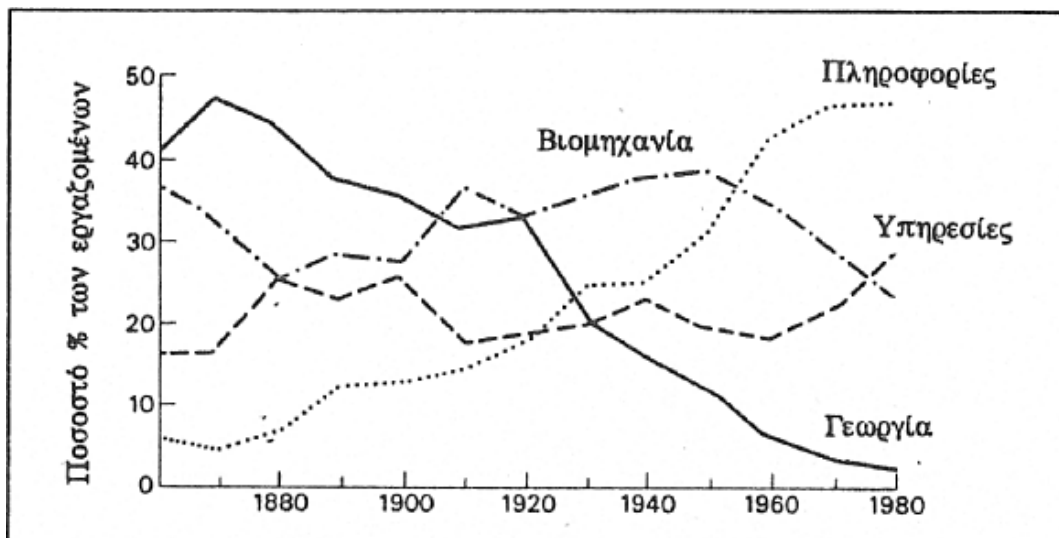
Από κοινωνική σκοπιά, η Πληροφορική υπόσχεται αλλαγές στον τρόπο με τον οποίο επικοινωνούμε και παίρνουμε αποφάσεις. Ακόμα και πριν από την εποχή των υπολογιστών, η ανάπτυξη των τηλεπικοινωνιών - για παράδειγμα, τηλέφωνο, ραδιόφωνο και τηλεόραση - άνοιξε τους ορίζοντες των ατόμων και των κοινωνιών και έθεσε στη διάθεση τους πληροφορίες για απομακρυσμένα γεγονότα και νέες ιδέες. Το γεγονός αυτό μας έχει βοηθήσει να κατανοήσουμε ένα μέρος της πολυπλοκότητας του σύγχρονου κόσμου, αλλά ταυτόχρονα έχει αυξήσει αυτή την πολυπλοκότητα, επιτρέποντας μια μεγαλύτερη αλληλεπίδραση μεταξύ των ανθρώπων.

Η χρήση του υπολογιστή στην επεξεργασία των πληροφοριών έχει οδηγήσει σε μια νέα διάσταση πολυπλοκότητας, μέσα από την ικανότητα για αποθήκευση τεράστιων ποσοτήτων από στοιχεία και την επεξεργασία τους με μεγάλη ταχύτητα. (Πολλάλης, 1999)

Παράλληλα, με την αύξηση των δαπανών στον τομέα των πληροφοριών, έχει υπάρξει μια μετατόπιση του εργατικού πληθυσμού προς επαγγέλματα που έχουν σχέση με τη διαχείριση πληροφοριών. Στη Μεγάλη Βρετανία, για παράδειγμα, ένας στους τρεις εργαζόμενους στη βιομηχανία χαρακτηρίζεται ως διοικητικός, ενώ το 1965 η αναλογία ήταν ένας στους τέσσερις. Στις Ηνωμένες Πολιτείες οι τάσεις είναι παρόμοιες.



Σχ. 1.1. Αλλαγές στα ποσοστά των εργαζομένων στους διάφορους τομείς της οικονομίας στη Μεγάλη Βρετανία. Ο τομέας των Πληροφοριών περιλαμβάνει τη δημόσια διοίκηση, τους διοικητικούς και τεχνικούς υπαλλήλους της βιομηχανίας, τις επιστημονικές υπηρεσίες, τις ασφαλίσεις, τις τραπεζικές και τις οικονομικές υπηρεσίες. Ο τομέας των Υπηρεσιών περιλαμβάνει τις εμπορικές αντιπροσωπείες, τις μεταφορές, τις τηλεπικοινωνίες και διάφορες άλλες υπηρεσίες. (Πηγή: Capron H. L., Εφαρμογές πληροφορικής, computers: εργαλεία για μια εποχή πληροφοριών, Αθήνα, 1999)



Σχ. 1.2. Αλλαγές στα ποσοστά των εργαζομένων στους διάφορους τομείς της οικονομίας στις Ηνωμένες Πολιτείες. (Πηγή: Capron H. L., Εφαρμογές πληροφορικής, computers: εργαλεία για μια εποχή πληροφοριών, Αθήνα, 1999)

Τέτοιες κοινωνικοοικονομικές τάσεις δεν είναι εύκολα αντιστρέψιμες και δεν υπάρχει κανένας λόγος να πιστέψουμε πως ο τομέας της πληροφορίας θα χάσει τη δεσπόζουσα θέση

του. Επομένως, η Πληροφορική θα έχει μια άμεση επίδραση στην πλειοψηφία των εργαζομένων στις αναπτυγμένες χώρες. (Norton, 1995)

Από οικονομική σκοπιά, τα στοιχεία των παραπάνω πινάκων δείχνουν μια αυξανόμενη ανισορροπία ανάμεσα στις παραδοσιακά παραγωγικές και διοικητικές απασχολήσεις. Για την ώρα, η επεξεργασία των πληροφοριών προσελκύει μεγάλους αριθμούς ατόμων και οι εργαζόμενοι στον τομέα αυτό απαιτούν υψηλές αμοιβές. Ακόμα, η απόδοση των εργαζομένων είναι δύσκολο να μετρηθεί. Πώς μπορεί κανείς να συγκρίνει την απόδοση ενός δασκάλου ή ενός ερευνητή με αυτήν ενός εργαζόμενου σε μια αλυσίδα παραγωγής; (Ζαφείρη, 2005)

Ποιό είναι το πραγματικό κόστος της πληροφορίας; Ποιά είναι η αξία της στην αγορά; Πώς συνδέεται με τον σημερινό σκελετό της οικονομίας;

Ένα άλλο ερέθισμα έρχεται από τις βιομηχανίες που έχουν πρόσφατα αναπτυχθεί γύρω από προϊόντα που σχετίζονται με την πληροφορία. Βιομηχανίες ειδικευμένες στους τομείς των υπολογιστών και των τηλεπικοινωνιών - ακρογωνιαίων λίθων της Πληροφορικής - ενώνονται τώρα με τη βιομηχανία των τσιπς από σιλικόνη, σε μια προσπάθεια να ανοίξουν νέες αγορές για τα προϊόντα τους: στο γραφείο, το σχολείο και την εκδοτική βιομηχανία - τις κατεξοχήν θέσεις όπου η πληροφορία βρίσκεται στο επίκεντρο του ενδιαφέροντος. Αυτές οι βιομηχανίες θα προσπαθήσουν να παράγουν μια μεγάλη ποικιλία προϊόντων για όλο τον πληθυσμό. Αλλά η ταχύτατη εξέλιξη της μικροηλεκτρονικής έχει βρει ανέτοιμο ακόμα και τον ίδιο τον κόσμο των υπολογιστών, που δεν μπορεί να φανταστεί το πλήθος των εφαρμογών οι οποίες θα ανταποκρίνονταν στη μεγιστοποίηση της απόδοσης των χρηστών.

Τέλος, από τη σκοπιά του ατόμου, η κατοχή της πληροφορίας ή η πρόσβαση σ' αυτήν έχουν γίνει θέματα προσωπικού ενδιαφέροντος. Η κατοχή της πληροφορίας φαίνεται, όλο και περισσότερο, σαν ένα κλειδί για την επαγγελματική εξέλιξη. Έτσι, ο έλεγχος της πρόσβασης στην πληροφορία θα μπορούσε (ακόμα πιο πολύ απ' ότι παλιότερα) να γίνει ένα όπλο υπέρ ή εναντίον του ατόμου. Η ανάπτυξη της Πληροφορικής έχει κιόλας οδηγήσει σε σκέψεις γύρω από τον κίνδυνο περιορισμού της ατομικής ελευθερίας. Ταυτόχρονα, όμως, η Πληροφορική μπορεί να δώσει τη δυνατότητα για τον σχηματισμό μιας προσωπικής γνώμης που θα στηρίζεται σε περισσότερα δεδομένα.

ΚΕΦΑΛΑΙΟ 3: ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΔΙΑΔΙΚΤΥΟΥ

3.1. Αρχιτεκτονικές Διαδικτύου-Δίκτυα

Τα μοντέρνα δίκτυα υπολογιστών έχουν σχεδιαστεί μ' έναν υψηλό βαθμό δόμησης. Για να ελαττώσουν την πολυπλοκότητα της σχεδίασης, τα περισσότερα δίκτυα έχουν οργανωθεί σε σειρές από στρώματα ή επίπεδα (layers ή levels), που το καθένα χτίζεται πάνω στο προηγούμενό του. Ο αριθμός των επιπέδων, τα ονόματά τους, τα περιεχόμενά τους και η λειτουργία του καθενός διαφέρουν από δίκτυο σε δίκτυο. Σε όλα όμως τα δίκτυα ο σκοπός κάθε επιπέδου είναι να προσφέρει συγκεκριμένες υπηρεσίες στα υψηλότερα επίπεδα, απομονώνοντας αυτά τα επίπεδα από τις λεπτομέρειες σχετικά με το πώς πραγματικά υλοποιούνται οι παρεχόμενες υπηρεσίες. Το επίπεδο n μιας μηχανής επικοινωνεί με το επίπεδο n μιας άλλης μηχανής. Οι κανόνες και οι συνθήκες που χρησιμοποιούνται σ' αυτή την επικοινωνία είναι γνωστές ως το πρωτόκολλο του επιπέδου n (layer n protocol). Οι οντότητες που περιλαμβάνονται στα αντίστοιχα επίπεδα σε διαφορετικά μηχανήματα ονομάζονται ομότιμες διεργασίες (peer processes).

Ανάμεσα σε κάθε ζεύγος γειτονικών επιπέδων υπάρχει μια διασύνδεση (interface). Η διασύνδεση αυτή καθορίζει ποιες πρωτογενείς λειτουργίες και υπηρεσίες προσφέρει ένα επίπεδο στο επίπεδο πάνω από αυτό. Το σύνολο των επιπέδων και πρωτοκόλλων ονομάζεται αρχιτεκτονική δικτύου (network architecture). (Καρολίδης, 2005)

Το μοντέλο αναφοράς OSI (Open System Interconnection – Διασύνδεση Ανοιχτών Συστημάτων) αναπτύχθηκε από τον Διεθνή Οργανισμό Τυποποίησης (ISO – International Standards Organization) και ασχολείται με συνδέσεις ανοιχτών συστημάτων (αυτά τα οποία είναι ανοιχτά για επικοινωνία με άλλα συστήματα). Υποδιαιρεί τις λειτουργίες ενός τηλεπικοινωνιακού δικτύου σε μια «κατακόρυφη» στοίβα από επίπεδα, για το καθένα από τα οποία μπορεί να οριστεί κάποιο πρωτόκολλο σε μία συγκεκριμένη υλοποίηση. Κάθε επίπεδο αξιοποιεί τις λειτουργίες του κατώτερου του στη στοίβα επιπέδου, ενώ στόχος του είναι να παρέχει λειτουργικότητα στο αμέσως ανώτερο επίπεδό του. Μία συγκεκριμένη υλοποίηση του μοντέλου, με καθορισμένα πρωτόκολλα για κάθε επίπεδο, ονομάζεται στοίβα πρωτοκόλλων ή απλά στοίβα (Allen, 1998).

Το μοντέλο OSI είναι μια ιεραρχική δομή επτά επιπέδων που καθορίζει τις προδιαγραφές επικοινωνίας μεταξύ δύο υπολογιστών, ορίζοντας επακριβώς τον σκοπό κάθε επιπέδου αλλά και τα χρησιμοποιούμενα πρωτόκολλα (Ζαφείρης, 2005):

7 Επίπεδο Εφαρμογής (Application Layer)
6 Επίπεδο Παρουσίασης (Presentation Layer)
5 Επίπεδο Συνόδου (Session Layer)
4 Επίπεδο Μεταφοράς (Transport Layer)
3 Επίπεδο Δικτύου (Network Layer)
2 Επίπεδο Σύνδεσης Δεδομένων (Data Link Layer)
1 Φυσικό Επίπεδο (Physical Layer)

Το μοντέλο αναφοράς OSI

Το φυσικό επίπεδο (physical layer) ασχολείται με τη μετάδοση ακατέργαστων bits σε ένα κανάλι επικοινωνίας. Επίσης ορίζει όλες τις ηλεκτρικές και φυσικές προδιαγραφές της επικοινωνίας. Η κύρια αποστολή του επιπέδου σύνδεσης δεδομένων (data link layer) είναι να μετασχηματίσει το ακατέργαστο μέσο μετάδοσης σε μια γραμμή που εμφανίζεται ελεύθερη από σφάλματα μετάδοσης στο επίπεδο δικτύου. Μερικές από τις βασικές λειτουργίες αυτού του επιπέδου είναι η επιβεβαίωση μετάδοσης και λήψης καθώς και η ανίχνευση λαθών. Το επίπεδο δικτύου (network layer) ασχολείται με τον έλεγχο της λειτουργίας του υποδικτύου. Παρέχει σύνδεση και δρομολόγηση (routing) ανάμεσα σε δύο κόμβους ενός δικτύου. Η βασική λειτουργία του επιπέδου μεταφοράς (transport layer) είναι η αποδοχή δεδομένων από το επίπεδο συνόδου, η διάσπαση αυτών σε μικρότερες μονάδες αν χρειαστεί, η μεταφορά τους στο επίπεδο δικτύου και η διασφάλιση ότι όλα τα τμήματα φτάνουν σωστά στην άλλη πλευρά. Ουσιαστικά διεκπεραιώνει τη μεταφορά των δεδομένων από χρήστη σε χρήστη, απαλλάσσοντας έτσι τα ανώτερα επίπεδα από κάθε φροντίδα να προσφέρουν αξιόπιστη μεταφορά δεδομένων από το ένα άκρο της επικοινωνίας στο άλλο. Το επίπεδο συνόδου (session layer) επιτρέπει στους χρήστες διαφορετικών μηχανημάτων να εγκαθιστούν συνόδους (δηλαδή ανταλλαγές δεδομένων) μεταξύ τους. Μία σύνοδος επιτρέπει μια συνήθη μεταφορά δεδομένων, όπως και το επίπεδο μεταφοράς, αλλά παρέχει και μερικές πρόσθετες υπηρεσίες που είναι χρήσιμες σε πολλές εφαρμογές. Μια σύνοδος, μπορεί να χρησιμοποιηθεί για να επιτρέψει τη σύνδεση ενός χρήστη σ' ένα απομακρυσμένο σύστημα καταμερισμού

χρόνου (time-sharing) ή να μεταφέρει ένα αρχείο μεταξύ δύο μηχανών. Το επίπεδο παρουσίασης (presentation layer) εκτελεί συγκεκριμένες λειτουργίες οι οποίες ζητούνται αρκετά συχνά από τους χρήστες, για να εξασφαλίζουν την εύρεση μιας γενικής λύσης γι' αυτούς, ώστε να μην αφήνεται κάθε χρήστης να λύνει τα προβλήματα μόνος του (Ζαφείρης, 2005). Συγκεκριμένα, ενώ όλα τα κατώτερα επίπεδα ενδιαφέρονται μόνο για την αξιόπιστη μεταφορά bits από το ένα μέρος στο άλλο, το επίπεδο παρουσίασης ενδιαφέρεται για το συντακτικό και τη σημασιολογία των πληροφοριών που μεταδίδονται. Το επίπεδο εφαρμογής (application layer) χρησιμοποιεί τις υπηρεσίες του επιπέδου παρουσίασης για την εκτέλεση εφαρμογών των χρηστών. Μερικές χαρακτηριστικές λειτουργίες αυτού του επιπέδου είναι η μεταφορά αρχείων, η εισαγωγή εργασιών από απόσταση, η εμφάνιση καταλόγων (directory) αρχείων, το ηλεκτρονικό ταχυδρομείο κλπ. Αυτό το επίπεδο είναι η κύρια διασύνδεση του χρήστη με το δίκτυο.

Οι τεχνολογίες IP VPN, χρησιμοποιούν ένα υπαρκτό TCP/IP δίκτυο για να περάσουν IP πακέτα ενός άλλου δικτύου, συνήθως κρυπτογραφημένα. Ουσιαστικά το VPN προσφέρει έτοιμες ευκολίες, όπως διάφανη κρυπτογράφηση δεδομένων και μηχανισμούς αυθεντικοποίησης. Το IP-VPN (Virtual Private Network) επιτρέπει τη διασύνδεση όλων των σημείων παρουσίας μιας επιχείρησης (κεντρικά γραφεία, εργοστάσια, υποκαταστήματα, αποθήκες κλπ.) μέσω του δικτύου ενός Τηλεπικοινωνιακού φορέα (Παπαδάκης, 2004). Η υπηρεσία IP - VPN είναι υπηρεσία υψηλής ποιότητας, πλήρως διαχειριζόμενη από άκρη σε άκρη, βασισμένη στο πρωτόκολλο Internet (IP). Αυτή η λύση συνδυάζει τις πλέον ελκυστικές δυνατότητες των τεχνολογιών που βασίζονται στο Internet με αυτές των ιδιωτικών δικτύων - και υποστηρίζεται από Συμφωνίες Επιπέδου Υπηρεσιών (Service Level Agreement "SLA"). Ο Τηλεπικοινωνιακός φορέας αναλαμβάνει την ευθύνη για τη σχεδίαση, υλοποίηση, παρακολούθηση και διαχείριση του δικτύου ευρείας περιοχής. Η υπηρεσία VPN είναι η δημιουργία λογικών καναλιών (point – to – point link) τα οποία χρησιμοποιούνται για την διασύνδεση δύο σημείων πάνω από το Internet (Allen, D.W. και Johmson, S. 1998).

Λειτουργικά υπάρχουν δύο είδη VPN :

- **Site to Site VPN:** Σε αυτήν την περίπτωση και τα δύο σημεία έχουν σταθερές IP διευθύνσεις και μόνιμη σύνδεση με το Internet. Χρησιμοποιούνται για τη διασύνδεση ιδιωτικών δικτύων και έρχεται να υποκαταστήσει τα μισθωμένα κυκλώματα. Σύμφωνα με τη χρήση του χωρίζονται σε *Internal*, τα οποία συνδέουν σημεία που

ανήκουν στον ίδιο οργανισμό ή εταιρία (central site & remote site), και *External* που συνδέουν σημεία που δεν ανήκουν στο ίδιο οργανισμό ή εταιρία (central site & supplier site). (Παπαδάκης, 2004)

- **Access VPN:** Σε αυτήν την περίπτωση μόνο το ένα σημείο έχει σταθερή IP διεύθυνση και μόνιμη σύνδεση με το Internet ενώ συνήθως το απομακρυσμένο είναι υπολογιστής με ειδικό software. (Πολλάλης, 2005).

3.2. Πρωτόκολλα Διαδικτύου

Αρχίζουμε με μια περίληψη των χαρακτηριστικών των πρωτοκόλλων.

Χαρακτηριστικά

Μερικά σημαντικά χαρακτηριστικά ενός πρωτοκόλλου είναι τα παρακάτω:

- Άμεσο/έμμεσο
- Μονολιθικό/δομημένο
- Συμμετρικό/ασύμμετρο
- Πρότυπο/μη πρότυπο

Η επικοινωνία ανάμεσα σε δύο οντότητες μπορεί να είναι **άμεση** ή **έμμεση**. Αν δύο συστήματα μοιράζονται μία σημείου προς σημείο ζεύξη, οι οντότητες σε αυτά τα συστήματα μπορούν να επικοινωνούν άμεσα. Αυτό σημαίνει ότι τα δεδομένα και οι πληροφορίες ελέγχου περνούν άμεσα μεταξύ των οντοτήτων χωρίς κανέναν παρεμβαλλόμενο ενεργό δρώντα. Το ίδιο μπορούμε να πούμε και για έναν πολυσημειακό (multipoint) σχηματισμό, αν και εδώ οι οντότητες πρέπει να εμπλέκονται με τον έλεγχο πρόσβασης κάνοντας έτσι το πρωτόκολλο πιο πολύπλοκο (Παπαδάκης, 2004). Αν τα συστήματα συνδέονται μέσω ενός επικοινωνιακού δικτύου μεταγωγής δεν είναι δυνατόν να έχουμε ένα άμεσο πρωτόκολλο. Οι δύο οντότητες θα πρέπει να εξαρτώνται από τη λειτουργία άλλων οντοτήτων για να ανταλλάξουν δεδομένα. Μία πιο ακραία περίπτωση είναι η κατάσταση στην οποία δύο οντότητες δεν μοιράζονται καν το ίδιο δίκτυο μεταγωγής, αλλά είναι έμμεσα συνδεδεμένες μέσω δύο ή περισσότερων δικτύων. Ένα σύνολο από τέτοια διασυνδεδεμένα δίκτυα ονομάζεται διαδίκτυο. (Παπαδάκης, 2004)

Ένα άλλο χαρακτηριστικό ενός πρωτοκόλλου σχετίζεται με το εάν αυτό είναι **μονολιθικό** ή **δομημένο**. Θα πρέπει να ξεκαθαριστεί, ότι η διεργασία της επικοινωνίας ανάμεσα σε οντότητες διαφορετικών συστημάτων είναι αρκετά πολύπλοκη για να την χειριστούμε ως μια ενότητα. Για παράδειγμα, ας θεωρήσουμε ένα πρόγραμμα ηλεκτρονικού

ταχυδρομείου το οποίο τρέχει σε δύο υπολογιστές συνδεδεμένους με μια σύγχρονη HDLC ζεύξη. Για να είναι στην πραγματικότητα μονολιθικό το πρόγραμμα, θα πρέπει να περιλαμβάνει όλη τη λογική του HDLC (Πολλάλης, 2005). Αν η σύνδεση γινόταν μέσω ενός δικτύου μεταγωγής πακέτου, το πρόγραμμα θα εξακολουθούσε να χρειάζεται τη λογική του HDLC (ή κάποια ανάλογη) για να συνδεθεί στο δίκτυο. Θα χρειαζόταν επίσης λογική για να διασπάσει iron mail σε κομμάτια μεγέθους όσο ένα πακέτο, λογική για την αίτηση ενός νοητού κυκλώματος και ούτω καθεξής. Το mail θα πρέπει να σταλεί μόνο όταν το σύστημα και η οντότητα προορισμού είναι ενεργά και έτοιμα να λάβουν δεδομένα. Για αυτό το είδος συνεργασίας απαιτείται λογική. Επιπλέον, όπως θα δούμε, η λίστα συνεχίζεται. Μία αλλαγή σε οποιαδήποτε φάση σημαίνει ότι αυτό το τεράστιο πρόγραμμα θα πρέπει να τροποποιηθεί με τον κίνδυνο να παρουσιαστούν βλάβες που δύσκολα ανιχνεύονται. (Ζαφείρης, 2005)



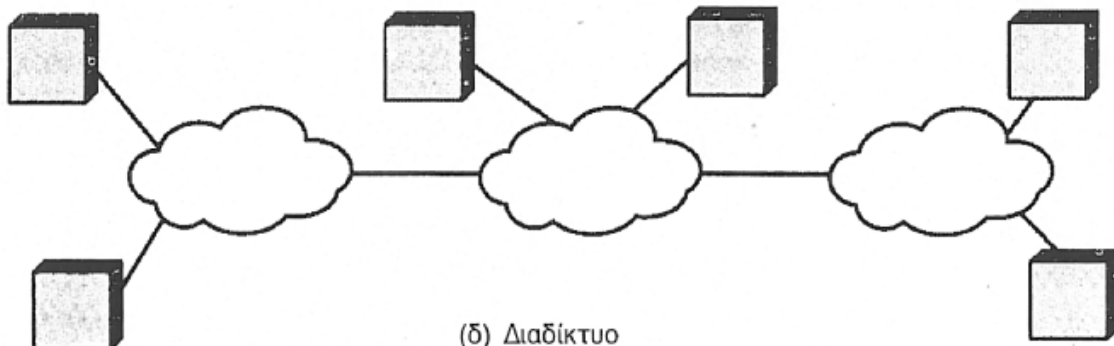
(α) Σημείου προς σημείο



(β) Πολυσημειακό Δίκτυο Ευρείας Εκπομπής



(γ) Δίκτυο Μεταγωγής



(δ) Διαδίκτυο

Πηγή: Capron H. L., Εφαρμογές πληροφορικής, computers: εργαλεία για μια εποχή πληροφοριών, Αθήνα, 1999

Μία εναλλακτική λύση είναι να χρησιμοποιηθούν δομημένες τεχνικές σχεδιασμού και υλοποίησης. Αντί για ένα μόνο πρωτόκολλο, υπάρχει ένα σύνολο πρωτοκόλλων που παρουσιάζουν μία ιεραρχική δομή ή δομή επιπέδων. Οι φυσικές λειτουργίες υλοποιούνται στις οντότητες των χαμηλότερων επιπέδων οι οποίες παρέχουν υπηρεσίες στις οντότητες των υψηλότερων επιπέδων. Για παράδειγμα, θα μπορούσε να υπάρχει μία μονάδα HDLC

(οντότητα) που όταν χρειαστεί να καλείται από το πρόγραμμα ηλεκτρονικού ταχυδρομείου. Σημειώνεται ότι αυτή είναι απλώς μια άλλη έμμεση μορφή. Οι οντότητες των υψηλότερων επιπέδων βασίζονται στις οντότητες των χαμηλότερων επιπέδων για να ανταλλάξουν δεδομένα. (Ζαφείρης, 2005)

Όταν χρησιμοποιείται σχεδιασμός δομημένου πρωτοκόλλου αναφερόμαστε στο υλικό και στο λογισμικό που χρησιμοποιούνται για την υλοποίηση της επικοινωνιακής λειτουργίας ως αρχιτεκτονική επικοινωνιών. Το υπόλοιπο αυτού του κεφαλαίου, μετά από αυτή την ενότητα, είναι αφιερωμένο σε αυτό το θέμα.

Ένα πρωτόκολλο μπορεί να είναι είτε συμμετρικό είτε ασύμμετρο. Τα περισσότερα από τα πρωτόκολλα που θα μελετήσουμε είναι συμμετρικά. Αυτό σημαίνει ότι περιλαμβάνουν επικοινωνία ανάμεσα σε ομότιμες οντότητες. Η ασυμμετρία μπορεί να υπαγορευθεί από τη λογική μιας ανταλλαγής (π.χ. μία διαδικασία client και server) ή από την επιθυμία να κρατηθεί μία από τις οντότητες ή τα συστήματα όσο πιο απλά γίνεται. Ένα παράδειγμα για το τελευταίο είναι η κατάσταση της κανονικής απόκρισης του HDLC (Πολλλάλης, 2005). Τυπικά, αυτή περιλαμβάνει έναν υπολογιστή ο οποίος καταμετρά και επιλέγει έναν αριθμό από τερματικά. Η λογική στο τερματικό άκρο είναι αρκετά απλή.

Λειτουργίες

Δεν έχουν όλα τα πρωτόκολλα όλες τις λειτουργίες. Αν συνέβαινε αυτό, τότε θα κάναμε διπλό κόπο. Ωστόσο, υπάρχουν πολλά παραδείγματα της ίδιας λειτουργίας σε πρωτόκολλα διαφορετικών επιπέδων.

Αυτή η συζήτηση θα είναι σκόπιμα περιληπτική. Παρέχει μία ολοκληρωμένη άποψη των χαρακτηριστικών και των λειτουργιών των επικοινωνιακών πρωτοκόλλων. Η αρχή του πρωτοκόλλου είναι θεμελιώδης σε πολλά από τα θέματα αυτού του βιβλίου και όπως προχωρούμε θα δούμε συγκεκριμένα παραδείγματα όλων αυτών των λειτουργιών (Ζαφείρης, 2005).

Μπορούμε να ομαδοποιήσουμε τις λειτουργίες ενός πρωτοκόλλου στις ακόλουθες κατηγορίες:

- Ενθυλάκωση
- Τεμαχισμός και επανασυναρμολόγηση
- Έλεγχος σύνδεσης
- Διατεταγμένη παράδοση

- Έλεγχος ροής
- Έλεγχος σφαλμάτων
- Διευθυνσιοδότηση
- Πολυπλεξία
- Υπηρεσίες μετάδοσης

3.3. Βασικές Υπηρεσίες Διαδικτύου

Οι βασικές υπηρεσίες του Διαδικτύου διακρίνονται σε (Παπαδάκης, 2004):

1. Διαχείριση
2. Αναζήτηση
3. Μεταφορά Αρχείων
4. Ηλεκτρονικό Ταχυδρομείο

Οι υπηρεσίες αυτές βασίζονται σε διάφορα πρότυπα και πρωτόκολλα. Πιο συγκεκριμένα (Allen, D.W. και Johnson, S., 1998):

1. Διαχείριση

- **ECHO:** Το πρωτόκολλο echo δημιουργήθηκε με απώτερο σκοπό τον έλεγχο για την απώλεια πακέτων στο υλικό του δικτύου και των κόμβων που μπορεί να προκύψει κατά τη μετάδοση τους. Πόροι του δικτύου όπως ξενιστές, δρομολογητές, εξυπηρετητές, υποστηρίζουν το πρωτόκολλο για τεχνικούς λόγους (Ζαφείρης, 2004).
- **PING:** Το ping είναι μια μέθοδος για τον εντοπισμό της διαθεσιμότητας και της απόδοσης ενός απομακρυσμένου πόρου του δικτύου. Θεωρείται ότι αποτελεί το ακρωνύμιο των λέξεων "Packet INternet Groper". Η διαδικασία με την οποία επιβεβαιώνεται η σύνδεση με έναν απομακρυσμένο υπολογιστή π.χ. μέσω Internet ή τοπικού δικτύου. Με το ping αποστέλλεται στον απομακρυσμένο υπολογιστή ένα πακέτο δεδομένων και στη συνέχεια ο υπολογιστής που έστειλε το πακέτο, περιμένει για μία echo απάντηση, δηλαδή την απάντηση στο πακέτο δεδομένων του ping.
- **TRACEROUTE:** Είναι μια μέθοδος παρόμοια με τη μέθοδο Ping. Με τη διαφορά ότι η μέθοδος traceroute παρουσιάζει λεπτομερείς πληροφορίες για την διαδρομή που

διασχίζουν τα πακέτα. Τα πακέτα πριν φτάσουν στον τελικό τους προορισμό διασχίζουν μια διαδρομή που είναι γεμάτη από μεγάλο αριθμό δρομολογητών και δικτύων κορμού (Πολλάλης, 2005). Ένα πακέτο απάντησης συνήθως ακολουθεί διαφορετική διαδρομή. Η μέθοδος traceroute παρουσιάζει λεπτομερώς τη διαδρομή που ακολουθεί το πακέτο από την στιγμή που θα φύγει από τον αποστολέα μέχρι την στιγμή που θα φτάσει στον παραλήπτη.

- **WHOIS-WHOIS++:** Είναι ένα πρωτόκολλο που χρησιμοποιείται για την εύρεση πληροφοριών σχετικά με δίκτυα, domains και οργανισμούς. Η υπηρεσία αυτή εφαρμόζεται σε μια σειρά εξυπηρετητών του διαδικτύου. Το πρωτόκολλο whois έχει το μειονέκτημα ότι δεν μπορεί να τρέξει σε όλους τους εξυπηρετητές. Μπορεί να υποστηρίξει απλές ερωτήσεις. Το 1992 το πρωτόκολλο whois εξελίχτηκε σε whois++ . Η αλλαγή στο πρωτόκολλο whois έγινε με την επέκταση κάποιων επιλογών, όπως η χρήση πολλαπλών γλωσσών που καθορίζουν το χαρακτήρα των πληροφοριών, δηλαδή δημιουργούν μια δομή στην πληροφορία η οποία επιστρέφεται από τον εξυπηρετητή (Ζαφείρης, 2004).

2. Αναζήτηση

Πριν εξαπλωθεί το πρωτόκολλο του παγκόσμιου ιστού η αναζήτηση εγγράφων στο διαδίκτυο γινόταν με διάφορες μεθόδους. Μερικές από αυτές είναι (Καλαφατούδης, 2004):

- ✓ **Archie:** είναι ένα σύστημα που έχει κατασκευαστεί για να αναζητεί δεδομένα που είναι αποθηκευμένα σε ftp εξυπηρετητές.
- ✓ **Gopher:** βάσει αυτής της μεθόδου η πληροφορία που αναζητείται παρουσιάζεται σε μια ιεραρχία (εξυπηρετητές, καταλόγους, υποκαταλόγους και αρχεία).
- ✓ **Veronica:** είναι ένα σύστημα όπου η αναζήτηση πληροφοριών γίνεται σε gopher εξυπηρετητές.
- ✓ **Wide Area Information Server (WAIS):** είναι ένα σύστημα για αναζήτηση πληροφοριών με βάση κάποιο απλό κείμενο.

3. Μεταφορά Αρχείων

- **File Transfer Protocol (FTP):** Το ftp πρωτόκολλο είναι ο απλούστερος και πιο διαδεδομένος τρόπος για την μεταφορά αρχείων μεταξύ υπολογιστών οι οποίοι είναι συνδεδεμένοι σε δίκτυα που υποστηρίζουν το πρωτόκολλο TCP/IP (όπως το internet). Με το TCP/IP πρωτόκολλο θα γίνει η μεταφορά των αρχείων για να είναι πιο αξιόπιστη και ασφαλής. Οι υπολογιστές που τρέχουν την εφαρμογή FTP client έχουν τη δυνατότητα μόλις συνδεθούν με το server να εκτελέσουν ένα πλήθος από διάφορες διεργασίες. Μερικές από τις οποίες είναι:

- ✓ **ανέβασμα αρχείων στο server (upload)**
- ✓ **κατέβασμα αρχείων από τον server (download)**
- ✓ **μετονομασία ή διαγραφή αρχείων από τον server**

Το πρωτόκολλο FTP είναι ένα ανοιχτό πρότυπο. Έτσι ένας υπολογιστής που είναι συνδεδεμένος σε ένα δίκτυο, έχει την δυνατότητα να διαχειρίζεται αρχεία σε έναν άλλο υπολογιστή του δικτύου ακόμη και αν ο υπολογιστής αυτός διαθέτει διαφορετικό λειτουργικό σύστημα.

- **Trivial file transfer protocol :**Είναι ένας απλός μηχανισμός μεταφοράς αρχείων μεταξύ δύο υπολογιστών. Αυτός ο μηχανισμός χρησιμοποιεί το UDP (User Datagram Protocol) για την μεταφορά των δεδομένων. Με αυτόν τον τρόπο υπάρχει η δυνατότητα μόνο της μεταφοράς αρχείων και όχι της περιήγησης μέσα σε καταλόγους.

4. Ηλεκτρονικό Ταχυδρομείο (e-mail)

Το ηλεκτρονικό ταχυδρομείο είναι ένα μέσο για την ανταλλαγή μηνυμάτων μεταξύ υπολογιστών. Για την αποστολή αλλά και την παράδοση των ηλεκτρονικών μηνυμάτων χρειάζεται η συνεργασία από ένα πλήθος διαφόρων πρωτοκόλλων (Πολλάλης, 2005). Ο πελάτης του ηλεκτρονικού ταχυδρομείου (user agent) στέλνει ένα ηλεκτρονικό έγγραφο στον εξυπηρετητή. Ο εξυπηρετητής ηλεκτρονικού ταχυδρομείου στον οποίο συνδέεται ο χρήστης αποτελεί το home ή outgoing server και αναλαμβάνει να διανείμει τα e-mails του χρήστη. Στον ίδιο εξυπηρετητή συνήθως απευθύνονται και τα e-mails που απευθύνονται στον χρήστη.

Ο εξυπηρετητής με την σειρά του προωθεί το μήνυμα στους άλλους εξυπηρετητές μέσω του Mail Transfer Agents (MTA). Ένα μήνυμα μπορεί να περάσει από πολλούς MTAs μέχρι να φτάσει στον εξυπηρετητή στον οποίον ο χρήστης που απευθύνεται έχει λογαριασμό (Ζαφείρη, 2005).

Ανάλογα με την κατάσταση στην οποία βρίσκεται το πρόγραμμα του χρήστη υπάρχουν τριών ειδών λειτουργίες (Carron, 1996):

- ✓ **Online:** ο πελάτης βρίσκεται σε σύνδεση με τον εξυπηρετητή. Η επεξεργασία των μηνυμάτων γίνεται κατά τη διάρκεια της σύνδεσης. Η επεξεργασία γίνεται στον εξυπηρετητή ενώ ο πελάτης δίνει τις εντολές. Ένα πρωτόκολλο που λειτουργεί με αυτόν τον τρόπο είναι το IMAP.
- ✓ **Offline:** ο πελάτης συνδέεται με τον εξυπηρετητή ανά διαστήματα και ελέγχει αν υπάρχουν νέα μηνύματα. Αν υπάρχουν τότε μεταφέρονται στον υπολογιστή του χρήστη και συνήθως διαγράφονται από τον εξυπηρετητή. Οποιαδήποτε άλλη λειτουργία διενεργείται από τον υπολογιστή και μόνο του χρήστη. Αυτό το είδος λειτουργίας χρησιμοποιείται από το Post Office Protocol.
- ✓ **Disconnected:** είναι ένα υβριδικό μοντέλο των καταστάσεων online και offline. Ο πελάτης συνδέεται με τον εξυπηρετητή και μεταφέρει ορισμένα μηνύματα που έχει επιλέξει, και τα επεξεργάζεται σε offline κατάσταση. Σε κάποια άλλη στιγμή ο πελάτης συνδέεται ξανά με τον εξυπηρετητή και μεταφέρει τις αλλαγές. Το IMAP πρωτόκολλο μπορεί να λειτουργήσει και με αυτήν την κατάσταση.

ΚΕΦΑΛΑΙΟ 4: ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ

4.1 Τεχνολογίες Διαδικτύου και Γλώσσες Προγραμματισμού

Μια πρώτη διάκριση στις γλώσσες προγραμματισμού σε δύο είδη είναι οι γλώσσες μηχανής, τα προγράμματα των οποίων αποτελούνται από διαταγές που είναι «κατανοητές» μόνο από μια συγκεκριμένη οικογένεια μηχανημάτων και εκτελούνται απευθείας από αυτά, και τις αλγοριθμικές γλώσσες, τα προγράμματα των οποίων αποτελούνται από εντολές που είναι ανεξάρτητες από οποιαδήποτε οικογένεια μηχανημάτων και για να εκτελεστούν πρέπει προηγουμένως να μεταφραστούν σε γλώσσα μηχανής. Οι εξωτερικές διαφορές ενός αλγορίθμου σε γλώσσα περιγραφής αλγορίθμων και του ίδιου αλγορίθμου σε αλγοριθμική γλώσσα προγραμματισμού είναι εμφανείς, όμως οι διαφορές μεταξύ ενός αλγοριθμικού προγράμματος και του αντίστοιχου προγράμματος σε γλώσσα μηχανής είναι πολύ μεγαλύτερες. Η παρακάτω εικόνα δείχνει σαφέστερα τη διαφορά, καθώς αντιπαραβάλλει τα προγράμματα του Ευκλείδειου αλγορίθμου, στη γλώσσα Modula-2 και στη γλώσσα μηχανής του μικροεπεξεργαστή MC 68000 της εταιρίας Motorola (Ζαφείρη, 2005).

Τα προγράμματα παρουσιάζουν τις εξής διαφορές:

Το πρόγραμμα μηχανής αποτελείται από διαταγές ειδικές για το συγκεκριμένο τύπο μηχανήματος και, αν χρειαστεί να μεταφερθεί από το μηχάνημα για το οποίο γράφτηκε σε άλλο, θα πρέπει να ξαναγραφεί εντελώς από την αρχή. Το αλγοριθμικό πρόγραμμα αποτελείται από εντολές ανεξάρτητες από το μηχάνημα επομένως, είναι αδύνατο να εκτελεστεί κατευθείαν από οποιοδήποτε μηχάνημα.

Το πρόγραμμα μηχανής περιγράφει συγκεκριμένες ενέργειες συσχετισμένες με το υλικό μέρος του υπολογιστή, ενέργειες όπως το φόρτωμα των τελεστών στον καταχωρητή, και η αποθήκευση των αποτελεσμάτων. Το αλγοριθμικό πρόγραμμα περιγράφει "αφηρημένες" ενέργειες, που δε συνδέονται άμεσα με το υλικό μέρος, ενέργειες όπου δεν εμφανίζονται η φόρτωση τελεστών και οι καταχωρητές (Blissmer, 1996).

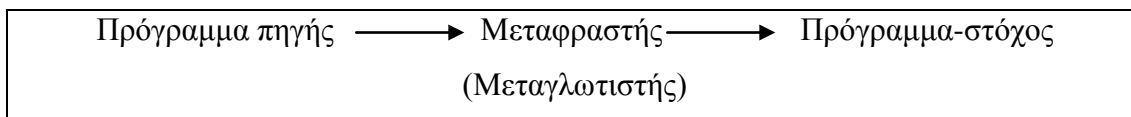
Σε ένα πρόγραμμα μηχανής είναι δύσκολο να ξεχωρίσουμε ποιος αλγόριθμος εκτελείται. Σε ένα αλγοριθμικό πρόγραμμα εκφράζεται πολύ σαφέστερα το τι κάνει ο αλγόριθμος" ένα τέτοιο πρόγραμμα είναι δηλαδή πολύ πιο ευανάγνωστο.

Στις αρχές της ανάπτυξης των υπολογιστών, στη δεκαετία του 1950, όλα τα προγράμματα έπρεπε να γράφονται σε γλώσσα μηχανής, κι αυτό είχε ως συνέπεια να είναι

δυσανάγνωστα, επιρρεπή σε λάθη, και εξαρτημένα από το συγκεκριμένο τύπο μηχανήματος. Στο τέλος όμως της δεκαετίας, οι προγραμματιστές είχαν μια εξαιρετική ιδέα: να αναπτύξουν γλώσσες προγραμματισμού ανεξάρτητες από συγκεκριμένα μηχανήματα και να μεταφράζουν αυτόματα με τον υπολογιστή, και με τη βοήθεια ενός ειδικού προγράμματος, τα προγράμματα που γράφονται στις γλώσσες αυτές σε γλώσσα μηχανής (Allen, 1998).

Μετάφραση

Πώς είναι όμως δυνατόν να μεταφράζεται ένα πρόγραμμα της Modula-2 σε γλώσσα μηχανής; Και τι σημαίνει «μετάφραση»; Πώς πρέπει να την φανταστεί κανείς; Ας αρχίσουμε από το «τι σημαίνει»! Για το πρόγραμμα μετάφρασης, το πρόγραμμα της Modula-2 που πρέπει να μεταφραστεί δεν είναι τίποτε περισσότερο από μια ακολουθία χαρακτήρων, ένα κείμενο. Το κείμενο εκτείνεται σε πολλές, ίσως και εκατοντάδες, γραμμές, και κάθε γραμμή αποτελείται από διατεταγμένα γράμματα, ψηφία, ειδικούς χαρακτήρες, και τον χαρακτήρα του κενού. Το κείμενο αυτό ονομάζεται **πρόγραμμα πηγής**. Ως μετάφραση ορίζεται τώρα η διαδικασία, με την οποία το πρόγραμμα διαβάζεται, αποθηκεύεται στη μνήμη, αναγνωρίζονται οι επιμέρους εντολές του και η σημασία τους, και από αυτές παράγεται ένα νέο πρόγραμμα, το **πρόγραμμα-στόχος**. Το πρόγραμμα-στόχος πρέπει να αναπαριστάνει τον ίδιο αλγόριθμο με το πρόγραμμα πηγής, αλλά στη γλώσσα μηχανής του μηχανήματος στο οποίο πρόκειται να εκτελεστεί. Στην κάτωθι Εικόνα βλέπουμε σχηματικά τη διαδικασία της μετάφρασης (Blissmer, 1996)



Τώρα πια θα πρέπει να σας έχει γίνει αρκετά σαφής η σημασία της έννοιας "μετάφραση" στην πληροφορική. Σημαίνει τον μετασχηματισμό ενός προγράμματος, που περιγράφει έναν αλγόριθμο, από μια γλώσσα πηγής, κατάλληλη για χρήση από ανθρώπους, σε μια γλώσσα-στόχο, κατάλληλη για το συγκεκριμένο μηχανήμα. Το πρόγραμμα που πραγματοποιεί τη μετάφραση ονομάζεται και **μεταγλωτιστής** (αγγλικά **compiler**). Επομένως, η μετάφραση στην πληροφορική διαφέρει αισθητά ως έννοια από τη μετάφραση με την κοινή σημασία του όρου και κάποιος που ασχολείται με τον τομέα αυτό στην πληροφορική δεν είναι φιλόλογος που επιχειρεί με τη βοήθεια υπολογιστή να μεταφράσει π.χ. αγγλικά σε ελληνικά.

Ο τρόπος λειτουργίας της μετάφρασης (μεταγλώττισης) είναι θέμα περίπλοκο και δε θα μας απασχολήσει περισσότερο. Η κατασκευή μεταγλωττιστών είναι ένα εκτεταμένο και πολυσχιδές πεδίο της πληροφορικής, με πολύ πλούσια βιβλιογραφία. Αξίζει να αναφέρουμε ότι ο μεταγλωττιστής πρέπει πρώτα να διερευνήσει τη δομή του προγράμματος πηγής, δηλαδή τη σύνθεση του σε μεταβλητές, σταθερές, εντολές, παραστάσεις κ.λπ., για να μπορέσει να προχωρήσει στην καθαυτή μετάφραση. Γι' αυτό είναι απαραίτητο το πρόγραμμα πηγής να έχει γραφεί με **ακρίβεια** από τον προγραμματιστή. Και η τελευταία τελεία, και το τελευταίο κόμμα, έχουν σημασία. Για παράδειγμα, δεν επιτρέπεται το ίδιο όνομα να γράφεται τη μια φορά Schmidt και την άλλη schmi **11**. Το f (**10**) είναι κάτι εντελώς διαφορετικό από το f [**10**] , και το **χ0** δεν έχει καμιά σχέση με το χO (το πρώτο έχει το ψηφίο 0, το δεύτερο το κεφαλαίο γράμμα O). Το φαινόμενο αυτό είναι τελείως αντίθετο με πολλές συνήθειες της καθημερινής ζωής, και υπερφαλαγγίζει κατά πολύ τη συνηθισμένη ακρίβεια έκφρασης στα μαθηματικά (Dologite, 1990). Κατά τη γραφή των προγραμμάτων, οι προγραμματιστές πρέπει να δείχνουν υπέρμετρη σχολαστικότητα, και να ελέγχουν βασανιστικά τα πάντα. Θα μπορούσε ίσως κανείς να συμβιβαστεί με το φαινόμενο αυτό, αν η ακρίβεια περιοριζόταν σε συγκεκριμένους, κρίσιμους τομείς, στους οποίους "απαιτείται απόλυτη ακρίβεια". Δυστυχώς, τα πράγματα δεν είναι έτσι. Ακόμη και τα απλούστερα, τα στοιχειωδέστερα τμήματα ενός προγράμματος, απαιτούν από τον προγραμματιστή την ίδια επιμέλεια όπως και τα πιο πολύπλοκα. Ο ανθρώπινος χαρακτήρας δεν μπορεί να συμφιλιωθεί με την ιδέα αυτή, γι' αυτό και (σχεδόν) κάθε πρόγραμμα περιέχει ορισμένα στοιχειώδη συντακτικά λάθη, που πρέπει να αφαιρεθούν για να μεταφραστεί σωστά από τον μεταγλωττιστή. Ευτυχώς σε κάθε προσπάθεια μετάφρασης, ο μεταγλωττιστής επισημαίνει τη θέση των λαθών και προβαίνει σε υποδείξεις για τη φύση τους, έτσι που να είναι εύκολο να εντοπιστούν, να διορθωθούν, και το πρόγραμμα να δοθεί ξανά προς μετάφραση, κάτι που οπωσδήποτε διευκολύνει αρκετά αυτήν την επίπονη διαδικασία.

Τον πρώτο καιρό μετά την εμφάνιση των πρώτων αλγοριθμικών γλωσσών, είχε δημιουργηθεί η εντύπωση ότι οι γλώσσες αυτές δεν επρόκειτο ποτέ να υποκαταστήσουν τις μηχανοστρεφείς γλώσσες, όπως ακριβώς και η τεχνητή γλώσσα Εσπεράντο δεν μπόρεσε να υποκαταστήσει τις φυσικές ανθρώπινες γλώσσες (Dologite, 1990). Εθεωρείτο ότι το αντίτιμο για την ανεξαρτησία από το συγκεκριμένο μηχάνημα ήταν η μεγάλη απώλεια της δυνατότητας έκφρασης, επειδή όπως ακριβώς οι φυσικές γλώσσες προσφέρουν πολύ μεγαλύτερη εκφραστικότητα από την Εσπεράντο, έτσι και η γλώσσα μηχανής καθιστά

δυνατή την εκμετάλλευση όλων των ειδικών δυνατοτήτων ενός μηχανήματος. Η συνέπεια αυτού είναι ότι τα μηχανοστρεφή προγράμματα είναι συντομότερα, και εκτελούνται ταχύτερα από τα αντίστοιχα τους που είναι μεταφρασμένα από μια αλγοριθμική γλώσσα. Με την πάροδο του χρόνου, η διαφορά αυτή, που υπήρχε στην αρχή, άρχισε να χάνει όλο και περισσότερο τη σημασία της, δεδομένου ότι οι σύγχρονες γλώσσες προγραμματισμού έγιναν όλο και περισσότερο εκφραστικές, και οι τεχνικές μετάφρασης τόσο προηγμένες που (με λιγιστές εξαιρέσεις) τα προγράμματα μηχανής που έχουν παραχθεί με μετάφραση εργάζονται σήμερα εξίσου αποτελεσματικά με τα προγράμματα μηχανής που είναι γραμμένα στο χέρι άλλωστε, σήμερα δεν είμαστε πλέον τόσο φειδωλοί όσο στο παρελθόν σε ό, τι αφορά τη χωρητικότητα της μνήμης (Carron, 1999).

Όσοι όμως ήταν διορατικοί διέκριναν αμέσως τα πλεονεκτήματα των γλωσσών προγραμματισμού υψηλού επιπέδου, δηλαδή την αποστασιοποίηση τους από τους επιμέρους τύπους μηχανημάτων και την επικέντρωσή τους στο αλγοριθμικό περιεχόμενο. Τελική συνέπεια ήταν σήμερα να εργαζόμαστε σχεδόν αποκλειστικά με γλώσσες υψηλού επιπέδου. Η εξέλιξη αυτή δεν ήταν το αποτέλεσμα μιας επανάστασης, μια και δεν συνέβη απότομα, αλλά επιβλήθηκε βαθμιαία, με την πάροδο του χρόνου. Επαναστατικός ήταν όμως ο αντίκτυπος της: άλλαξε την πρακτική πληροφορική, την ανύψωσε σε ανώτερο επίπεδο, συνέβαλε στο να αναδειχθεί σε επιστήμη. Η μέγιστη αξία της έγκειται ίσως στο ότι προώθησε εντονότατα την αλγοριθμική σκέψη. Ο σημερινός προγραμματιστής δε σκέπτεται πλέον με τις έννοιες του μηχανοστρεφούς προγραμματισμού, αλλά με τις δομές δεδομένων και τις δυνατότητες που θέτουν στη διάθεση του οι σύγχρονες γλώσσες προγραμματισμού. Ακόμη καλύτερα κάνει μάλιστα, όταν προχωρεί ένα ακόμα βήμα και μεταβαίνει από τα στοιχεία μιας εκείνα μιας γλώσσας περιγραφής συγκεκριμένης γλώσσας προγραμματισμού σε αλγορίθμων, όπως η Adele (McBride, 1999).

Η πληθώρα αλγοριθμικών γλωσσών

Από τις αρχές της δεκαετίας του '60 έχουν κάνει την εμφάνισή τους πολλές εκατοντάδες, ίσως μάλιστα και χιλιάδες γλώσσες προγραμματισμού. Λίγες όμως απ' αυτές κατόρθωσαν να γνωρίσουν ευρύτερη διάδοση. Και από αυτές τις λίγες, αν εξαιρέσουμε τις γλώσσες που εξυπηρετούν ειδικούς σκοπούς, σήμερα έχουν σημασία μόνο δέκα ως είκοσι.

Γιατί τόσο πολλές γλώσσες; Η αρχική ιδέα ήταν, θα πίστευε κανείς, να δημιουργηθεί μία οικουμενική γλώσσα. Πραγματικά" όλες αυτές οι γλώσσες είναι οικουμενικές, με την

έννοια ότι με καθεμία απ' αυτές μπορεί να επιλυθεί οποιοδήποτε προγραμματιστικό πρόβλημα σε μορφή ανεξάρτητη από το συγκεκριμένο τύπο μηχανήματος. Η πληθώρα έχει πολλές αιτίες. Ο στόχος της εισαγωγής μιας νέας γλώσσας προγραμματισμού ήταν συνήθως η βελτίωση της εκφραστικής ικανότητας. Στις πρώτες γλώσσες στο προσκήνιο βρισκόταν η ικανότητα αριθμητικών υπολογισμών, ενώ αργότερα η έμφαση δόθηκε στην επεξεργασία εγγραφών και άλλων δομών δεδομένων. Στις πρώτες γλώσσες, η ροή ρυθμιζόταν με εντολές άλματος υπό όρους ή χωρίς όρους, ενώ αργότερα έκαναν την εμφάνιση τους οι βρόχοι και άλλοι, όλο και περισσότεροι γενικευμένοι μηχανισμοί. Οι προσπάθειες που γίνονται αποσκοπούν στη δυνατότητα έκφρασης κατά τον απλούστερο και φυσικότερο δυνατό τρόπο όσων θέλει κανείς να διατυπώσει, με παράλληλη αποστασιοποίηση από τον τρόπο υλοποίησης από τον υπολογιστή. Ο δρόμος αυτός είναι μακρύς και πολλά από τα παρακλάδια του αδιέξοδα.

Επιπλέον διαπιστώθηκε ότι υπάρχουν τόσα διαφορετικά πεδία εφαρμογών, που αξίζει τον κόπο να σχεδιαστεί μια γλώσσα προγραμματισμού με στοιχεία που προσφέρονται άριστα για μια ειδική κατηγορία εφαρμογών, ενώ σε άλλους τομείς χρειάζονται ελάχιστα ή και καθόλου. Έτσι, στις τεχνικές-επιστημονικές εφαρμογές χρειάζονται τρισδιάστατοι πίνακες, σύνθετοι αριθμοί, και τριγωνομετρικές συναρτήσεις αντίθετα, στις εμπορικές εφαρμογές απαιτούνται ιδιαίτερες δυνατότητες για το χειρισμό εκτεταμένου και διατεταγμένου κειμένου και πινάκων (Glossbrenner, 1999).

Αυτά για την πληθώρα των γλωσσών προγραμματισμού. Όσο για τις επιμέρους γλώσσες, όσο κι αν δεν μπορούμε να εξετάσουμε αναλυτικά καμιά από αυτές, μπορούμε ωστόσο να αναφέρουμε τις πιο σημαντικές. Η Fortran είναι η αρχαιότερη γλώσσα προγραμματισμού για τεχνικές-επιστημονικές εφαρμογές, ενώ η Cobol είναι η αρχαιότερη γλώσσα για εμπορικές εφαρμογές. Και οι δύο τους, παρά τους επανειλημμένους εκσυγχρονισμούς που τους έχουν γίνει, έχουν πια γεράσει αθεράπευτα, κι όμως εξακολουθούν να χρησιμοποιούνται πολύ, απλώς και μόνο επειδή υπάρχουν πάρα πολλά προγράμματα γραμμένα σ' αυτές και δεν είναι δυνατόν, για οικονομικούς λόγους, να ξαναγραφούν όλα από την αρχή. Για μια περίοδο, η **PL/I** είχε συγκεντρώσει την προσοχή και εθεωρείτο ως γλώσσα που καλύπτει το όλο φάσμα των εφαρμογών. Ήταν όμως γλώσσα υπερβολικά εκτενής και από μερικές άλλες απόψεις η δομή της ήταν ατυχής. Ενώ οι Fortran, Cobol, και PL/I αναπτύχθηκαν από τη βιομηχανία, και κυρίως από την εταιρία IBM, η γλώσσα Algol-60 σχεδιάστηκε, λίγο καιρό μετά τη Fortran, από επιτροπή επιστημόνων. Από

επιστημονική άποψη, αποτελούσε σε σύγκριση με τη Fortran ένα βήμα προς τα εμπρός, ως προς τον πρακτικό χειρισμό της όμως ήταν ένα βήμα προς τα πίσω (Ζαφείρη, 2005).

Αντιδρώντας προς τις "γλώσσες-ελέφαντες" PL/I και Algol-68 (επίγονο της Algol-60), ο N. Wirth, καθηγητής πληροφορικής στη Ζυρίχη, δημιούργησε γύρω στο 1970 τη γλώσσα Pascal. Στην αρχή, ο Wirth είχε συλλάβει τη γλώσσα απλώς και μόνο ως βοήθημα για τη διδασκαλία του προγραμματισμού, με λίγες, αλλά σαφείς και νέες δομές. Προβληματίστηκε για την απόρριψη των περιττών στοιχείων όσο και για την προσθήκη νέων. Η Pascal επιβλήθηκε διεθνώς και χρησιμοποιείται μέχρι σήμερα ευρύτατα στην πράξη και στη συγγραφή προγραμμάτων. Περίπου 10 χρόνια αργότερα, δηλαδή γύρω στο 1980, ο Wirth ανέπτυξε μια διάδοχο της Pascal, τη Modula-2, η οποία σήμερα έχει εκτοπίσει την Pascal σε μεγάλο βαθμό, τουλάχιστον μεταξύ των επιστημόνων της πληροφορικής. Παράλληλα, κατά τη δεκαετία του '70 και ύστερα από πρωτοβουλία του Υπουργείου Αμύνης των ΗΠΑ αναπτύχθηκε η γλώσσα Ada (προς τιμήν της Ada Augusta Lovelace, 1816-1851, ετεροθαλούς αδελφής του Λόρδου Βύρωνος και συνεργάτιδος του Charles Babbage). Η Ada σχεδιάστηκε να γίνει, όπως και η PL/I, μια οικουμενική γλώσσα για κάθε πεδίο εφαρμογών, παραμένοντας ταυτόχρονα απλή και σαφής όσο και η Pascal. Έχοντας δύο τόσο αντιφατικούς στόχους, η Ada εξελίχθηκε σε μια ακόμη γλώσσα που πάσχει από γιγαντισμό και το μέλλον της είναι ακόμη αβέβαιο (Glossbrenner, 1999).

Περίεργο φαινόμενο αποτελεί η γλώσσα προγραμματισμού C. Αναπτύχθηκε στις αρχές της δεκαετίας του '70 και σήμερα έχει γεράσει από την άποψη της δομής της τόσο πολύ, που να είναι ανεπίδεκτη επιδιορθώσεων όσο και η Fortran. Όμως, η C συνδέεται στενά με το λειτουργικό σύστημα Unix, το οποίο έχει επιβληθεί παγκοσμίως, με αποτέλεσμα η C να εξαπλώνεται όλο και περισσότερο και να αποτελεί σήμερα μια πολύ ζωντανή γλώσσα, καθιερωμένη σε ευρύτατο φάσμα εφαρμογών.

Υπάρχει και μια άλλη εξέλιξη, την οποία εδώ απλώς θα αναφέρουμε: είναι η δημιουργία γλωσσών ανώτερου επιπέδου αφαίρεσης απ' ό,τι οι αλγοριθμικές γλώσσες.

Τα πλεονεκτήματα και τα μειονεκτήματα της καθεμίας γλώσσας προγραμματισμού είναι ένα ανεξάντλητο θέμα συζήτησης μεταξύ ειδικών της πληροφορικής. Μερικοί θεωρούν την εκλογή μιας συγκεκριμένης γλώσσας προγραμματισμού από τον αρχάριο ως γεγονός τόσο καθοριστικής σημασίας, που διατυπώνουν απόλυτες απόψεις του τύπου: «Όποιος ξεκινάει με Fortran, δε θα μάθει ποτέ να προγραμματίζει σωστά» (Glossbrenner, 1999). Άλλοι έχουν τη γνώμη ότι με κάθε γλώσσα προγραμματισμού μπορούν να γραφτούν και

καλά και κακά προγράμματα, και ότι για το λόγο αυτόν πολύ μεγαλύτερη σημασία έχει η ορθή κατανόηση των αλγοριθμικών εννοιών απ' ό,τι η χρήση μιας συγκεκριμένης γλώσσας. Από τις δύο αυτές ακραίες απόψεις, η πρώτη είναι ανόητη, ενώ η δεύτερη, αντίθετα, πολύ λογική. Μία μόνο βελτίωση της πρέπει να «απαιτήσουμε»: η γλώσσα προγραμματισμού που χρησιμοποιείται, θα πρέπει να προάγει την αλγοριθμική σκέψη' τούτο σημαίνει ότι δε θα πρέπει στη διατύπωση αλγορίθμων να χρησιμοποιούνται σχηματισμοί εννοιών που δεν υφίστανται στην επιλεγμένη για χρήση γλώσσα προγραμματισμού (Norton, 1995).

Ορισμοί Γλωσσών

Η ανάπτυξη των γλωσσών προγραμματισμού είχε και μια άλλη ενδιαφέρουσα, και τυπική για την πληροφορική, συνέπεια. Συγκεκριμένα, όποιος θέλει να χρησιμοποιήσει μια γλώσσα προγραμματισμού, πρέπει να ξέρει ακριβώς πώς εκφράζονται σε αυτήν οι διάφοροι αλγόριθμοι. Πρέπει να την γνωρίζει μέχρι την τελευταία λεπτομέρεια, τόσο από συντακτική όσο και από σημειολογική άποψη. Συντακτικό είναι η διδασκαλία της κατασκευής προτάσεων η γνώση μιας γλώσσας προγραμματισμού από συντακτική άποψη σημαίνει τη δυνατότητα συγγραφής προγραμμάτων, που αποτελούνται από δηλώσεις, εντολές, παραστάσεις, και την απαιτούμενη στίξη με σωστή γραμματική. Σημειολογία είναι η διδασκαλία της σημασίας, και η γνώση μιας γλώσσας προγραμματισμού από σημειολογική άποψη σημαίνει τη δυνατότητα κατανόησης όλων των δηλώσεων, παραστάσεων, και εντολών. Και τα δύο είναι περίπλοκα, πολύ πιο περίπλοκα απ' ό,τι η σύνταξη και η σημειολογία των μαθηματικών τύπων (Καλαφατούδης, 2004).

Κατά συνέπεια, είναι απαραίτητο κατά την εκμάθηση μιας γλώσσας προγραμματισμού να υπάρχει υποστήριξη από επίσημη τεκμηρίωση, όπου να περιγράφονται, διεξοδικά και με ακρίβεια, το συντακτικό και η σημειολογία της γλώσσας. Έτσι, ο προγραμματιστής θα μπορεί να συμπεραίνει χωρίς αμφιβολία αν το πρόγραμμα που έχει γράψει είναι συντακτικά ορθό και αν πραγματικά εκτελεί αυτά που ο ίδιος είχε κατά νου, αν δηλαδή ο υπολογιστής θα πραγματοποιήσει αυτά που ήθελε να πετύχει ο προγραμματιστής όταν έγραφε το πρόγραμμα. Η επίσημη αυτή τεκμηρίωση λέγεται ορισμός της γλώσσας.

Οι πλήρεις και ακριβείς ορισμοί γλωσσών είναι δύσκολο να γραφτούν και δύσκολο να διαβαστούν. Ο πρώτος ορισμός υπήρξε η διάσημη έκθεση Algol-60. Εκεί, το συντακτικό περιγραφόταν σε μεγάλο βαθμό με τύπους- αντίθετα, η σημειολογία περιγραφόταν με λέξεις. Και αυτή η μέθοδος έχει επικρατήσει ως σήμερα.

Επιπλέον, μέχρι τώρα δεν έχει υπάρξει απόλυτα ικανοποιητικός ορισμός γλώσσας. Ένας καλός ορισμός γλώσσας πρέπει να είναι σύντομος ώστε ο αναγνώστης να μην "πνίγεται" σε λεπτομέρειες- από την άλλη όμως πλευρά πρέπει να είναι πλήρης, δηλαδή να περιλαμβάνει όλες τις λεπτομέρειες. Δυστυχώς, αυτές οι δύο απαιτήσεις δεν είναι δυνατόν να συμφιλιωθούν. Επιπλέον, ο ορισμός πρέπει να είναι όσο το δυνατόν περισσότερο τυπικός, ώστε να μην αφήνει περιθώρια παρεξηγήσεων, από την άλλη όμως πλευρά πρέπει να είναι και ευανάγνωστος -άλλες δύο απαιτήσεις που πολύ δύσκολα συμβιβάζονται. Για το λόγο αυτό βρίσκει κανείς και πολύ σύντομους ορισμούς γλωσσών (Modula-2: 27 σελίδες) αλλά και πολύ εκτεταμένους (Ada: 340 σελίδες).¹ Άλλωστε, για τις περισσότερες γλώσσες προγραμματισμού δεν υπάρχει κανένας ορισμός, όσο παράδοξο κι αν ακούγεται αυτό. Περιγράφονται απλώς σε εγχειρίδια ή άλλες εκδόσεις με κανόνες και παραδείγματα, χωρίς φιλοδοξίες πληρότητας ή μονοσήμαντου.

Παρακάτω θα παρουσιαστούν ορισμένες από τις πλέον γνωστές γλώσσες προγραμματισμού:

4.1.1 HTML

Τα αρχικά HTML προέρχονται από τις λέξεις HyperText Markup Language. Η HTML δεν είναι μια γλώσσα προγραμματισμού. Είναι μια γλώσσα σήμανσης (markup language), δηλαδή ένας ειδικός τρόπος γραφής κειμένου. Ο καθένας μπορεί να δημιουργήσει ένα αρχείο HTML χρησιμοποιώντας απλώς έναν επεξεργαστή κειμένου. Ο browser αναγνωρίζει αυτόν τον τρόπο γραφής και εκτελεί τις εντολές που περιέχονται σε αυτόν. Αξίζει να σημειωθεί ότι η HTML είναι η πρώτη και πιο διαδεδομένη γλώσσα περιγραφής της δομής μιας ιστοσελίδας. Ο σκοπός ενός web browser είναι να διαβάζει τα έγγραφα HTML και τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Η HTML χρησιμοποιεί τις ειδικές ετικέτες (tags) να δώσει τις απαραίτητες οδηγίες στον browser. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.² Τα tags είναι εντολές που συνήθως ορίζουν την αρχή ή το τέλος μιας λειτουργίας. Τα tags βρίσκονται πάντα μεταξύ των συμβόλων < και >. Π.χ. <BODY>. Ένα αρχείο HTML πρέπει να έχει κατάληξη htm ή html. Τα έγγραφα HTML είναι απλά αρχεία κειμένου (γνωστά

¹ jcp.h.wikispaces.com/aboutlanguages, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

² www.webvistas.org/topic/43-html5/, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

σαν ASCII) που μπορούν να δημιουργηθούν με οποιονδήποτε κειμενογράφο (SimpleText σε Macintosh, Notepad σε PC).³ Ένα οποιοδήποτε κείμενο αποτελείται από επιμέρους στοιχεία όπως πίνακες παραγράφων, λίστες κ.α.

Τα πρώτα χρόνια, η HTML χρησιμοποιούταν μόνο για τη δημιουργία στατικών ιστοσελίδων όπως επίσης και για την διάταξη των εγγράφων. Η HTML συνεχίζει να παραμένει, ιεραρχικά δομημένη και αποτελούμενη από ετικέτες. Οι ετικέτες (tags) αυτές περικλείουν και διαφοροποιούν τα bit κειμένου, υποδεικνύοντας την λειτουργία και το σκοπό του κειμένου που βρίσκεται ανάμεσα στις ετικέτες. Οι ετικέτες είναι γραμμένες απευθείας σε μορφή απλού κειμένου στο html έγγραφο όπου μπορούν να διερμηνευτούν από το λογισμικό του υπολογιστή. Οι ετικέτες αυτές καθαυτές δεν παρουσιάζονται στον browser και είναι ξεχωριστές από το περιεχόμενο που περικλείουν. Σημειώνεται ότι ανοίγουν με τη μορφή <tag> και κλείνουν με τη μορφή </tag>. Για παράδειγμα το <p> </p> ανοίγει και κλείνει μία παράγραφο.⁴

Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από *ετικέτες*, οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα <html>), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <h1> και </h1>), με την πρώτη να ονομάζεται *ετικέτα έναρξης* και τη δεύτερη *ετικέτα λήξης* (ή σε άλλες περιπτώσεις *ετικέτα ανοίγματος* και *ετικέτα κλεισίματος* αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός web browser είναι να διαβάσει τα έγγραφα HTML και τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.⁵

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά

³ jcpw.wikispaces.com/aboutlanguages, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

⁴ www.tplanet.eu/showtutorial.php?id, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

⁵ www.tplanet.eu/showtutorial.php?id, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα.⁶ Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η Javascript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.⁷

Τον Ιανουάριο του 2008 δόθηκε στη δημοσιότητα ένα προσχέδιο της HTML 5, μια σημαντική αναθεώρηση της γλώσσας σήμανσης για τον Παγκόσμιο Ιστό. Η προδιαγραφή HTML 5 βοηθά στη βελτίωση της διαλειτουργικότητας και στη μείωση του κόστους λογισμικού δίνοντας ακριβείς κανόνες όχι μόνο για τη διαχείριση όλων των σωστών εγγράφων HTML αλλά και για τον τρόπο επιδιόρθωσης των λαθών. Οι συντάκτες της HTML5 είναι ο Ian Hickson της εταιρίας Google και ο Dave Hyatt της εταιρίας Apple.

Η ομάδα Web Hypertext Application Technology Working Group (WHATWG) άρχισε δουλειά σε αυτή την έκδοση τον Ιούνιο του 2004 με το όνομα Web Applications 1.0. Το Φεβρουάριο του 2010 το πρότυπο ήταν ακόμη σε κατάσταση "Last Call" στο WHATWG.⁸

Η HTML5 προορίζεται για αντικατάσταση της HTML 4.01, της XHTML 1.0, και της DOM Level 2 HTML. Ο σκοπός είναι η μείωση της ανάγκης για ιδιόκτητα plug-in και πολλές άλλες διαδικτυακές εφαρμογές (RIA) όπως το Adobe Flash, το Microsoft Silverlight, το Apache Pivot, και η Sun JavaFX.

Οι ιδέες πίσω από την HTML5 εμφανίστηκαν αρχικά το 2004 από την ομάδα WHATWG. Η HTML5 εμπεριέχει το πρότυπο *Web Forms 2.0* που είναι επίσης της WHATWG.⁹

⁶ jcpw.wikispaces.com/aboutlanguages, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

⁷ www.webvistas.org/topic/43-html5/, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

⁸ jcpw.wikispaces.com/aboutlanguages, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

⁹ www.webvistas.org/topic/43-html5/, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

Το πρότυπο HTML5 υιοθετήθηκε ως αρχικό βήμα για τις εργασίες της νέας ομάδας εργασίας HTML του W3C το 2007. Αυτή η ομάδα εργασίας δημοσίευσε το Πρώτο Δημόσιο Working Draft του προτύπου στις 22 Ιανουαρίου 2008. Το πρότυπο είναι ακόμη υπό ανάπτυξη, και αναμένεται να παραμείνει έτσι για πολλά χρόνια, παρόλο που μέρη της HTML5 θα τελειώσουν και θα υποστηριχτούν από περιηγητές πριν το όλο πρότυπο φτάσει στη τελική κατάσταση Recommendation.¹⁰

Οι HTML σελίδες αρχίζουν με τη δήλωση `<!DOCTYPE>` και ακολουθούνται από τις HTML ετικέτες HEAD και BODY ως εξής¹¹:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD>
<TITLE>Ο τίτλος της παρούσας σελίδας</TITLE>
... άλλα στοιχεία που τοποθετούνται στο στοιχείο HEAD ...
</HEAD>
<BODY>
... κύριος κορμός της σελίδας ...
</BODY>
</HTML>
```

Παρακάτω παρατίθενται οι πιο σημαντικές εντολές της HTML:

<head> : Η εντολή αυτή περιέχει τον τίτλο και άλλες σημαντικές πληροφορίες για το έγγραφο.

<title> : Μέσα σε αυτήν την εντολή περιλαμβάνεται ο τίτλος του εγγράφου. Ο τίτλος εμφανίζεται στην μπάρα του browser.

<p> : Η εντολή αυτή ορίζει μία νέα παράγραφο.

<h1> : Η εντολή αυτή καθορίζει το μέγεθος των γραμμών. Το `<h1>` είναι το μεγαλύτερο μέγεθος και το `<h6>` είναι το μικρότερο. Η πρώτη κεφαλίδα σε κάθε κείμενο θα πρέπει να έχει το σημάδι `<h1>`.

¹⁰ www.tplanet.eu/showtutorial.php?id, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

¹¹ jcp.h.wikispaces.com/aboutlanguages, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

**
** : Η εντολή **
** αναγκάζει το κείμενο (και όλο το υπόλοιπο υλικό που περιέχει η σελίδα) να συνεχίσει στη επόμενη σειρά.

**** : Η εντολή αυτή από την εμφάνισή της και μέχρι την λήξη της (δηλ. ****) αναγκάζει το κείμενο να φαίνεται με έντονη γραφή (bold).

<i> : Η εντολή αυτή από την εμφάνισή της και μέχρι την λήξη της (δηλ. **</i>**) αναγκάζει το κείμενο να φαίνεται με πλάγια γραφή (italic).

<u> : Η εντολή αυτή από την εμφάνισή της και μέχρι την λήξη της (δηλ. **</u>**) αναγκάζει το κείμενο να φαίνεται υπογραμμισμένο (underline).

Στην HTML υπάρχουν λίστες αριθμημένες ή μη. Η εντολή για αριθμημένη λίστα είναι η **** ενώ για μη αριθμημένη είναι ****. Για κάθε στοιχείο της λίστας υπάρχει η εντολή ****. Η εντολή για την εισαγωγή εικόνας σε HTML έγγραφο είναι ****. Όμως δεν γράφεται έτσι απλά, αλλά συνοδεύεται και από μερικές παραμέτρους. Η πιο σημαντική παράμετρος που δεν μπορεί να παραλειφθεί είναι η **src="ονομαεικονας"**. Η παράμετρος **src** καθορίζει το όνομα του αρχείου εικόνας που θα χρησιμοποιηθεί και σε ποιο φάκελο (directory). Όλες οι εικόνες για να εμφανιστούν στο Internet πρέπει να είναι GIF ή JPG. Για να ορίσουμε ένα πίνακα αρκεί η εντολή **<table>**. Έπειτα ορίζουμε τις σειρές και τις στήλες, προσθέτοντας ένα- ένα κελί. Για να ορίσουμε μία σειρά (row) γράφουμε την **<tr>**, ενώ για στήλη το **<td>**. Ως σύνδεσμο (link) στην HTML ορίζουμε ότι μπορεί να επιλεγεί και σαν αποτέλεσμα να εμφανιστεί μια άλλη σελίδα. Ένας σύνδεσμος δηλαδή είναι ένας δείκτης προς ένα άλλο έγγραφο HTML. Για να ορίσουμε κάτι ως σύνδεσμο (κείμενο, εικόνα) πρέπει να εισάγουμε την εντολή **<a>**. Από την εμφάνιση της **<a>** και ως τη λήξη ισχύος της (με το ****), ό,τι περιέχεται γίνεται αυτόματα σύνδεσμος και εμφανίζεται διαφορετικά από τον browser (Παπαδάκης, 2004).

Οι φόρμες HTML χρησιμοποιούνται για να περάσουμε δεδομένα σε ένα server. Μια φόρμα μπορεί να περιέχει στοιχεία εισόδου, όπως text fields, checkboxes, radio-buttons, submit buttons και πολλά άλλα. Μια φόρμα μπορεί να περιέχει επίσης λίστες για να επιλέγουμε ανάμεσα σε καταλόγους. Το πιο σημαντικό στοιχείο της φόρμας είναι το στοιχείο εισόδου. Το στοιχείο εισόδου χρησιμοποιείται για να επιλέξουμε πληροφορίες του χρήστη.

Ως σύνδεσμο (link) στην HTML ορίζουμε ότι μπορεί να επιλεγεί (κάνοντας click) και σαν αποτέλεσμα να εμφανιστεί μια άλλη σελίδα. Για να ορίσουμε κάτι ως σύνδεσμο (κείμενο, εικόνα) πρέπει να εισάγουμε την εντολή **<a>**. Από την εμφάνιση της **<a>** και ως τη

λήξη ισχύος της (με το ``) ότι περιέχεται γίνεται αυτόματα σύνδεσμος και εμφανίζεται διαφορετικά από τον browser (συνήθως υπογραμμισμένο και με μπλε γράμματα). Σημαντική παράμετρος είναι η `href="NameofData"` όπου καθορίζει σε ποιο στοιχείο είναι δείκτης ο σύνδεσμος.¹²

Η HTML έχει αλλάξει ελάχιστα από τότε που δημιουργήθηκε. Αντίθετα, αυτό που έχει αλλάξει είναι η μορφοποίηση των στοιχείων της HTML κάτι που σήμερα γίνεται με την χρήση του CSS.

4.1.2 PHP

Η PHP (Hypertext Preprocessor) είναι μία ευρέως χρησιμοποιούμενη, ανοιχτού κώδικα και γενικού σκοπού γλώσσα σεναρίου που είναι ειδικά σχεδιασμένη για την ανάπτυξη εφαρμογών διαδικτύου και μπορεί να ενσωματωθεί μέσα σε κώδικα HTML και να εκτελείται κάθε φορά που ο χρήστης επισκέπτεται τη σελίδα. Μια σελίδα PHP περνά από επεξεργασία από συμβατό web server και δημιουργεί κώδικα HTML ή άλλη έξοδο που θα δει ο επισκέπτης. Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δε θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται κώδικα PHP. (Ζαφείρη, 2005)

Ο κώδικας PHP μιας ιστοσελίδας περικλείεται από τα tags `<?php` και `?>` Την απεικόνιση κειμένου αναλαμβάνουν οι εντολές `print` ή `echo`. Για παράδειγμα, ο ακόλουθος κώδικας

```
<?php
print "Hello world!";
?>
```

θα εμφανίσει στο πρόγραμμα περιήγησης των επισκεπτών τη φράση Hello world! Θα πρέπει όμως να προσέξουμε ώστε η προβαλλόμενη φράση να ξεκινά και να τελειώνει με τον ίδιο

PHP¹³:

¹² ifestos.teilar.gr/index.php?option=com, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

¹³ ifestos.teilar.gr/index.php?option=com, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

τρόπο. Δηλαδή αν επιλέξουμε να ξεκινά με διπλά εισαγωγικά, θα πρέπει επίσης να κλείνει με διπλά και όχι με μονά, αλλιώς θα εμφανιστεί σφάλμα επεξεργασίας.

Όταν ο προχωρημένος προγραμματιστής φτάσει στο σημείο να δημιουργεί ολοκληρωμένες εφαρμογές PHP, το πιθανότερο είναι ότι θα χρειαστεί να τοποθετήσει σχόλια σε διάφορα σημεία του περίπλοκου κώδικα, ώστε να θυμάται τη λειτουργία που εκτελούν. Σχόλια μπορούν να εισαχθούν είτε με δύο χαρακτήρες καθέτου // στην αρχή κάθε γραμμής, είτε με /* και */ στην αρχή και στο τέλος του σχολίου αντίστοιχα (ανεξάρτητα από τον αριθμό των γραμμών του). Το κείμενο που σημειώνεται με αυτό τον τρόπο, αγνοείται κατά την επεξεργασία του κώδικα PHP από τον web server.

Όπως οι περισσότερες γλώσσες προγραμματισμού, έτσι και η PHP υποστηρίζει μεταβλητές. Στην PHP οι μεταβλητές φέρουν ως πρόθεμα τον χαρακτήρα του δολλαρίου (\$). Επίσης υποστηρίζει operators, οι οποίοι θέτουν προϋποθέσεις για την εκτέλεση ενός τμήματος κώδικα. Οι operators διατυπώνονται με τις εντολές if, elseif, else ή switch/case σε συνδυασμό με τα σύμβολα !=, ==, >, <, <=, >=.

Αναλυτικά¹⁴:

Operators σύγκρισης

- \$a == \$b Το \$a είναι ίσο με το \$b
- \$a != \$b Το \$a δεν είναι ίσο με το \$b
- \$a > \$b Το \$a είναι μεγαλύτερο του \$b
- \$a >= Το \$a είναι μεγαλύτερο ή ίσο του \$b

Operators μαθηματικών πράξεων

- \$a + \$b Πρόσθεση \$a συν \$b
- \$a - \$b Αφαίρεση \$a - \$b
- \$a * \$b Πολλαπλασιασμός του \$a επί το \$b
- \$a / \$b Διαίρεση \$a προς \$b
- \$a++ Αύξηση του \$a κατά 1 μονάδα
- \$a-- Μείωση του \$a κατά 1 μονάδα

¹⁴ [nefeli.lib.teicrete.gr/.../Chatzipaulis2010.pdf](#), όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

Λογικοί operators

&& και

|| ή

! όχι

Κατά τον ορισμό προϋποθέσεων για τις τιμές μεταβλητών στην PHP με τους operators if, elseif, else πρέπει απαραίτητα να χρησιμοποιείται το διπλό "ίσον". Αν χρησιμοποιήσουμε μονό "ίσον" στον κώδικα (π.χ. if (\$mynumber = 8) αντί για if (\$mynumber == 8), η PHP θα απέδιδε στη μεταβλητή \$mynumber τον αριθμό 8, κάτι που θα προκαλούσε προβλήματα στην εκτέλεση του υπόλοιπου κώδικα.

Η PHP και η HTML είναι δύο γλώσσες προγραμματισμού που συνεργάζονται απόλυτα μεταξύ τους. Επίσης η PHP "συνεργάζεται" τέλεια με βάσεις δεδομένων που υποστηρίζουν επερωτήματα SQL, όπως είναι οι MySQL, Microsoft SQL Server, PostgreSQL, Oracle κ.ά. Αυτό επιτρέπει τη δημιουργία αξιόλογων εφαρμογών που επιτρέπουν την αποθήκευση δεδομένων και ανταλλαγή τους μεταξύ online χρηστών. Η σύνδεση σε έναν MySQL server στον ίδιο ηλεκτρονικό υπολογιστή όπου "τρέχει" και ο web server, γίνεται με την εντολή mysql_connect() ως εξής¹⁵:

```
<?php
mysql_connect("localhost", "username", "password");
?>
```

και στη συνέχεια, η κατάλληλη βάση δεδομένων επιλέγεται με την εντολή mysql_select_db():

```
<?php
mysql_select_db("database_name");
?>
```

Η PHP ενσωματώνει την ισχύ και τη δυναμικότητα σχετικά παλαιότερων γλωσσών όπως η Perl αλλά καταργώντας τις αδυναμίες τους. Μερικά από τα βασικά χαρακτηριστικά της είναι:

- Ο πηγαίος κώδικάς της διανέμεται ελεύθερα στο διαδίκτυο δίνοντας τη δυνατότητα σε όποιον θέλει να κατασκευάζει και να διανέμει εφαρμογές για εμπορική και μη χρήση.

¹⁵ nefeli.lib.teicrete.gr/.../Chatzipaulis2010.pdf, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

- Μπορεί να μεταφραστεί και να τρέξει στα περισσότερα λειτουργικά συστήματα που κυκλοφορούν στην αγορά
- Συνεργάζεται χωρίς προβλήματα με τους πιο δημοφιλείς Web Servers που κυκλοφορούν όπως τον Apache
- Διαθέτει ενσωματωμένες εντολές υποστήριξης για ένα μεγάλο αριθμό βάσεων δεδομένων
- Μπορεί να χρησιμοποιηθεί στη δημιουργία εικόνων, ανάγνωση / εγγραφή σε αρχεία και για αποστολή email.

4.1.3 XML

Η XML (EXtensible Markup Language), όπως και η HTML, αποτελείται από στοιχεία και ιδιότητες, αλλά οι προγραμματιστές μπορούν να δημιουργήσουν ή να προσθέσουν νέα στοιχεία και ιδιότητες, χωρίς να αλλάζουν την XML τυποποίηση. Αυτό κάνει την XML επεκτάσιμη - αλλά συνεχίζει να είναι μια τυποποίηση. Μία άλλη πλευρά της XML είναι ότι οι προγραμματιστές μπορούν να δημιουργήσουν XML έγγραφα οποιασδήποτε πολυπλοκότητας, ακόμη και με εκατομμύρια στοιχεία και ιδιότητες σε ένα έγγραφο (παρόλο που για λόγους επεξεργασίας, το μικρότερο μέγεθος πάντα θα λειτουργεί καλύτερα). Για παράδειγμα, αν μία σχεσιακή βάση δεδομένων μετατραπεί σε XML μορφή, θα υπήρχαν εκατομμύρια στοιχεία που θα αναπαριστούσαν τα εκατομμύρια των εγγραφών της βάσης δεδομένων (Παπαδάκης, 2004).

Η XML είναι ιδανική για χρήση στο Internet, επειδή τα δεδομένα ενός XML εγγράφου, όπως και στην HTML, είναι κωδικοποιημένα με ετικέτες που αποτελούνται από απλό κείμενο. Τα περισσότερα λειτουργικά συστήματα μπορούν να χειριστούν εύκολα το απλό κείμενο και άρα είναι ευκολότερο να δημιουργηθούν επεξεργαστές και προγράμματα περιήγησης που θα χειρίζονται την XML. Και επειδή τα XML στοιχεία και ιδιότητες μπορεί να ορίζονται σε DTD και σε διατάξεις που είναι διαθέσιμα σε οποιοδήποτε άτομο ή μηχάνημα, όποτε υπάρχουν επεκτάσεις, αναγνωρίζονται αυτόματα (εφόσον συμμορφώνονται με τις XML τυποποιήσεις).

Μάλιστα, η XML μέθοδος σήμανσης δεδομένων προχωράει πολύ προς τη λύση του προβλήματος του άμεσου ορισμού των πληροφοριών με ένα τυπικό τρόπο, που μπορούν να

καταλάβουν οι υπολογιστές. Για παράδειγμα, σε ένα XML έγγραφο, οι πληροφορίες για την ηλικία ενός πελάτη μπορούν να γραφούν με αυτόν τον τρόπο¹⁶:

```
<P>Customer = <B>John Smith</B></P> <P>Age =  
<B>35</B></P>
```

Αυτό λέει σε έναν χρήστη που διαβάζει τη σελίδα, ότι ο πελάτης είναι 35 ετών, αλλά δεν λέει τίποτα για τον υπολογιστή που σαρώνει το έγγραφο -τουλάχιστον όσον αφορά τα στοιχεία. Πάντως, σε ένα XML έγγραφο, θα ήταν κωδικοποιημένα ως εξής¹⁷:

```
<customer>  
<name>John Smith</name>  
<age>35</age>  
</customer>
```

Αφού υπάρχει ένα στοιχείο ηλικίας (ένα εξαρτώμενο στοιχείο του στοιχείου πελάτη), ένας υπολογιστής που επεξεργάζεται το XML έγγραφο θα μπορούσε να συμπεράνει ότι αυτό το στοιχείο αναφέρεται στην ηλικία του πελάτη, αντί να πρέπει να συμπεράνει αυτό το γεγονός από τα συμφραζόμενα στην HTML, στη γραμμή που δείχνει το όνομα του πελάτη.

Σε μια XML διάταξη, το στοιχείο ηλικίας θα συμπεριλάμβανε έναν περιορισμό, όπως ότι η ηλικία δεν θα μπορούσε να είναι μικρότερη από 0 ή μεγαλύτερη από 200 και θα έπρεπε πάντα να είναι ακέραιος, που υπολογίζεται από την ημερομηνία γέννησης του πελάτη, στρογγυλοποιημένος προς τα κάτω. Αυτές οι επιπλέον οδηγίες επιτρέπουν ένα αρκετά ακριβή ορισμό του στοιχείου της ηλικίας, σε σημείο που να μην υπάρχουν σοβαρές αμφιβολίες σε σχέση με την έννοια του στοιχείου της ηλικίας.

Είναι επιθυμητό να μειώνεται η αμφιβολία, γιατί η επικοινωνία εξαρτάται σε μεγάλο βαθμό από συμφωνίες ανάμεσα στα επικοινωνούντα μέλη, ειδικά ανάμεσα σε μηχανήματα, τα περισσότερα εκ των οποίων δεν μπορούν εύκολα να καταλάβουν τα λανθασμένα δεδομένα (όπως μία ηλικία 210, που οι άνθρωποι αυτόματα θα υποψιάζονταν ότι είναι σφάλμα, ενώ οι μηχανές θα το θεωρούσαν σωστό).

Συστατικά Ενός XML Εγγράφου

¹⁶ www.tplanet.eu/showtutorial.php?id, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

¹⁷ www.tplanet.eu/showtutorial.php?id, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

Ένας όρος που θα δείτε να χρησιμοποιείται συνέχεια στα XML έγγραφα είναι η *σωστή σύνταξη*. Τα XML έγγραφα πρέπει να ακολουθούν έναν αριθμό κανόνων για να θεωρούνται σωστά και αν ένα έγγραφο δεν είναι σωστό, πιθανόν να εμφανίσει σφάλματα κατά την επεξεργασία του. Ασφαλώς, η δουλειά σας είναι να βεβαιωθείτε ότι τα έγγραφα σας είναι σωστά από κάθε άποψη προτού δημοσιευθούν, οπότε θα πρέπει να βεβαιωθείτε ότι ακολουθήσατε τους κανόνες. Αυτοί οι κανόνες θα καλυφθούν στις ενότητες που περιγράφουν τα συστατικά ενός XML εγγράφου. Είναι επίσης σημαντικό να ξέρετε ότι τα XML έγγραφα μπορεί να είναι σωστά, αλλά να μην είναι έγκυρα. Ένα έγκυρο έγγραφο συμμορφώνεται με τους κανόνες και περιορισμούς που ορίζονται στα DTD ή στη διάταξη, ενώ ένα σωστό έγγραφο συμμορφώνεται γενικά μόνο με τους κανόνες της XML¹⁸.

XML Οντότητες και Δεδομένα

Τα XML έγγραφα αποτελούνται από μονάδες που καλούνται *οντότητες* (entities). Σκεφτείτε τις ως αρχεία ή άλλες πηγές των XML δεδομένων. Ένα XML έγγραφο μπορεί να αποτελείται από μία ή περισσότερες από αυτές τις μονάδες, με τον ίδιο τρόπο που ένα HTML έγγραφο μπορεί να προέρχεται από ένα HTML αρχείο, μαζί με ένα αρχείο συμπερίληψης από την πλευρά του διακομιστή και από μερικά επιπλέον δεδομένα που προέρχονται από μία βάση δεδομένων. Οι οντότητες προσδιορίζονται από τα ονόματα τους και μπορεί είτε να περιέχουν όλα τα δεδομένα τους στην δική τους δήλωση είτε να αναφέρουν τα δεδομένα τους μέσω ενός URL (Allen, 1998).

Τα δεδομένα που είναι τμήμα ενός XML εγγράφου αποτελούνται από οντότητες που έχουν αναλυθεί και από οντότητες που δεν έχουν αναλυθεί. Οι οντότητες που δεν *έχουν αναλυθεί*, περιέχουν δεδομένα που μπορεί να είναι ή να μην είναι κείμενο. Αν είναι κείμενο, μπορεί να είναι κείμενο που επιτρέπεται σε XML οντότητες που έχουν αναλυθεί ή που δεν επιτρέπεται.

4.1.4 XHTML

Η XHTML είναι συνδυασμός της ***HTML και της XML***. Πιο συγκεκριμένα η XHTML είναι μια παραλλαγή της ***HTML*** γραμμένη με κανόνες σύνταξης της ***XML***, δηλαδή ο κώδικας ενός XHTML αρχείου μπορεί να περιέχει σχεδόν όλες τις ετικέτες που περιέχει και η HTML,

¹⁸ www.fme.aegean.gr/sites/.../xsd_askhsh_level1.pdf, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

με διαφορά την αυστηρότερη σύνταξη του κώδικα. Τα αρχικά της *XHTML* είναι από το *EXtensible HyperText Markup Language* το οποίο σημαίνει εκτεταμένη γλώσσα σήμανσης υπερκειμένου. Είναι μια πιο αυστηρή και πιο καθαρή έκδοση της HTML και μοιάζει πολύ με την HTML 4.0.¹⁹

Η XHTML είναι ο διάδοχος της HTML, μια αυστηρότερη μορφή κώδικα, που δημιουργήθηκε με σκοπό την κατασκευή ιστοσελίδων που θα είναι προσβάσιμες και από άλλες συσκευές εκτός από τους browsers (π.χ. κινητά τηλέφωνα). Πάνω σ' αυτή τη τεχνολογία στηρίζονται οι browsers που έχουν σχεδιαστεί ειδικά για τα άτομα με ειδικές ανάγκες, με αποτέλεσμα να επιτρέπεται ένα μεγαλύτερο κοινό πρόσβασης στις ιστοσελίδες που σχεδιάστηκαν με XHTML.²⁰

Πολλές ιστοσελίδες είναι γραμμένες με κακό τρόπο στην σύνταξη του HTML κώδικα τους. Παρόλ' αυτά το τελικό αποτέλεσμα θα εμφανιστεί σωστά στον browser του επισκέπτη. Υπάρχουν όμως αρκετοί web browsers οι οποίοι διαχειρίζονται τις τεχνολογίες ανάπτυξης ιστοσελίδων και web εφαρμογών (όπως html, css, php κτλ), ο καθένας με διαφορετικό τρόπο. Όταν έχουν να διαχειριστούν κακογραμμένους κώδικες, όλοι οι browsers χρειάζεται να δεσμεύουν μεγαλύτερη επεξεργαστική ισχύ από τον υπολογιστή στον οποίον "τρέχουν". Μερικοί από αυτούς τους browsers τρέχουν, εκτός από τους υπολογιστές γραφείου, και σε κινητά και σε άλλες μικροσυσκευές, οι οποίες δεν έχουν την επεξεργαστική ισχύ που έχει ένας υπολογιστής ώστε να διαχειριστεί κακογραμμένους κώδικες. Η XHTML αναπτύχθηκε λοιπόν, συνδυάζοντας την αυστηρότητα της XML με τις δυνατότητες της HTML. Η XHTML λόγω της αυστηρότητας στη σύνταξή της, μπορεί να "τρέξει" χωρίς να χρειάζεται μεγάλη επεξεργαστική ισχύ από τα υπολογιστικά συστήματα.

Σήμερα σπάνια βρίσκουμε ιστοσελίδες γραμμένες με το πρότυπο της HTML. Πλέον είναι απαραίτητο να ακολουθούμε το πρότυπο XHTML. Όλα τα CMS, όπως το Joomla, δημιουργούν σελίδες σύμφωνα με το πρότυπο της XHTML. Να σημειώσουμε εδώ ότι η επέκταση των αρχείων που είναι γραμμένα σε XHTML παραμένει η ίδια με την HTML, δηλαδή .html ή .htm.

¹⁹ www.fme.aegean.gr/sites/.../xsd_askhsh_level1.pdf, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

²⁰ www.fme.aegean.gr/sites/.../xsd_askhsh_level1.pdf, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

Οι πιο σημαντικές διαφορές στον τρόπο σύνταξης μεταξύ της XHTML και της HTML είναι οι εξής²¹:

- Στην XHTML όλες οι ετικέτες, οι ιδιότητες και οι τιμές τους, γράφονται με μικρά (πεζά). Επίσης όλες οι ετικέτες έχουν ετικέτα τέλους.
- Στην HTML έχουμε την δυνατότητα να γράφουμε κώδικα είτε με πεζούς είτε με κεφαλαίους χαρακτήρες. Ιδιαίτερα στην αρχή, είχε καθιερωθεί οι web designers να χρησιμοποιούν κεφαλαίους χαρακτήρες για τις ετικέτες και πεζούς για τις ιδιότητες και τις τιμές, ώστε οι ετικέτες να ξεχωρίζουν από τον υπόλοιπο κώδικα για να είναι πιο ευανάγνωστος. Στην XHTML δεν ισχύει αυτό. Οι ετικέτες καθώς και οι ιδιότητες με τις τιμές τους πρέπει να γράφονται με πεζούς χαρακτήρες.
- Στην XHTML οι τιμές των ιδιοτήτων των ετικετών πρέπει απαραίτητα να βρίσκονται μέσα σε διπλά εισαγωγικά, ενώ η HTML δίνει τη δυνατότητα να χρησιμοποιήσουμε και μονά εισαγωγικά.
- Σε ορισμένες περιπτώσεις στην HTML μπορούμε να παραλείψουμε την τιμή μιας ιδιότητας όταν αυτή είναι ίδια με την ιδιότητα, πράγμα που δε συμβαίνει με την XHTML.²² Π.χ. στην HTML μπορούμε να γράψουμε

```
<input readonly />
```

Ενώ στην XHTML πρέπει να γράψουμε

```
<input readonly =" readonly ">
```

- Κάθε αρχείο γραμμένο σε XHTML, πρέπει να αρχίζει με την ετικέτα **<!DOCTYPE>**, με τις ετικέτες `<html>`, `<head>` και `<body>` να ακολουθούν.
- Τα κενά στοιχεία πρέπει είτε να έχουν ένα τελικό tag ή το αρχικό tag πρέπει να τελειώνει με τους χαρακτήρες `</>`.
- Το χαρακτηριστικό id αντικαθιστά το χαρακτηριστικό name.

Ένα XHTML DTD (**Document Type Definition**) ορίζει την επιτρεπτή σύνταξη των XHTML στοιχείων μέσα σε ένα έγγραφο. Υπάρχουν τρία XHTML DTDs²³ :

²¹ www.webstandards.org/learn/articles/askw3c/oct2003/, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

²² www.webstandards.org/learn/articles/askw3c/oct2003/, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

²³ www.webstandards.org/learn/articles/askw3c/oct2003/, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

- **XHTML 1.0 Transitional:** Προορίζεται για την εύκολη μετάβαση από την έκδοση HTML 3.2. Η έκδοση αυτή πρέπει να προτιμάται όταν θέλουμε η ιστοσελίδα να είναι προσπελάσιμη από όσο το δυνατόν περισσότερους χρήστες του Διαδικτύου, επειδή υποστηρίζει σχεδόν όλες τις ετικέτες της XHTML ακόμη και αυτές οι οποίες είναι προς διαγραφή.
- **XHTML 1.0 Strict:** Διαχωρίζει το περιεχόμενο από τον τρόπο εμφάνισης. Προορίζεται για την ανάπτυξη ιστοσελίδων οι οποίες απαιτούν την ικανοποίηση ειδικών προδιαγραφών π.χ. των προδιαγραφών οι οποίες καθορίζονται από δημόσιους φορείς και υπηρεσίες.
- **XHTML 1.0 Frameset:** Η έκδοση αυτή θα πρέπει να χρησιμοποιείται όταν γίνεται χρήση πλαισίων σε μια ιστοσελίδα.

4.1.5 JAVASCRIPT

Η JavaScript είναι το αποτέλεσμα μιας κοινής προσπάθειας των εταιρειών Netscape Communications Corporation και Sun Microsystems, Inc. Η αναγγελία της νέας γλώσσας έγινε στις 4 Δεκεμβρίου του 1995, τότε που το Netscape Navigator 2.0 ήταν ακόμη στην beta έκδοση. (Ζαφείρη, 2005)

Η JavaScript έκδοση 1.0 έγινε διαθέσιμη με την νέα έκδοση αυτής της εφαρμογής browser. (Πριν από την κυκλοφορία της με το όνομα JavaScript, η γλώσσα αποκαλούνταν LiveScript).

Είναι μια σχετικά απλή γλώσσα η οποία παρέχει βασικές προγραμματιστικές λειτουργίες. Οποιαδήποτε προγραμματιστική εμπειρία, από τη Visual Basic μέχρι την Pascal ή την C (η οποία είναι πολύ πιο εξελιγμένη) είναι αρκετή για την κατανόηση της JavaScript. Η JavaScript είναι μία διερμηνευμένη (interpreted) γλώσσα προγραμματισμού με ιδιότητες αντικειμενοστραφούς γλώσσας προγραμματισμού, χωρίς όμως να μπορεί να χαρακτηριστεί ως πλήρης αντικειμενοστραφής. Η γλώσσα αυτή, κτίστηκε ουσιαστικά πάνω στο πρότυπο των γλωσσών C, C++ και Java. Στην JavaScript οι μεταβλητές δεν είναι απαραίτητο να έχουν ένα συγκεκριμένο τύπο ή ακόμη είναι δυνατόν να αλλάζουν τύπο κατά τη διάρκεια της ζωής τους. Επίσης, δεν πρέπει να συγχέεται η JavaScript με την Java. Η μία δεν έχει καμία σχέση με την άλλη. Η χρήση του ονόματος JavaScript έγινε για λόγους προώθησης της γλώσσας σε μία εποχή που η εξάπλωση της Java ήταν πολύ μεγάλη (Καλαφατούδης, 2004).

Η γλώσσα JavaScript χρησιμοποιείται κυρίως για την εξυπηρέτηση των παρακάτω σκοπών:

- Λιγότερος φόρτος των server: Ο έλεγχος και η επικύρωση των δεδομένων που εισάγονται από τους χρήστες γίνεται από τη μεριά του browser κι έτσι δεδομένα τα οποία δεν είναι σε κατάλληλη μορφή δεν αποστέλλονται στον server.
- Άμεση αλληλεπίδραση με τους χρήστες: Με την χρήση της JavaScript για τον έλεγχο των δεδομένων μειώνονται οι χρόνοι αναμονής του χρηστών, αφού αυτοί δεν χρειάζεται να περιμένουν μεγάλα χρονικά διαστήματα επαναφόρτωσης της σελίδας σε περίπτωση που έχουν ξεχάσει να εισάγουν κάποιο δεδομένο ή έχουν εισάγει κάτι λάθος.
- Αυτόματη διόρθωση λαθών: Ένα παράδειγμα που μπορεί να κάνει περισσότερο κατανοητό το πώς μπορεί να χρησιμοποιηθεί η JavaScript με αυτόν τον τρόπο είναι αυτό της ημερομηνίας. Πολλά συστήματα βάσεων δεδομένων αποθηκεύουν δεδομένα ημερομηνιών σε μορφή dd-mm-yyyy. Αν κάποιος χρήστης εισάγει κάποια ημερομηνία σε μορφή dd/mm/yyyy τότε κάτι τέτοιο θα μπορούσε να ανιχνευτεί αυτόματα από τον browser και να μετατραπεί στην σωστή μορφή πριν τα δεδομένα αποσταλούν στον server.
- Αυξημένη χρηστικότητα: Αυτό επιτυγχάνεται επιτρέποντας στον χρήστη την αλλαγή και αλληλεπίδραση με το γραφικό περιβάλλον χωρίς την επαναφόρτωση της σελίδας. Ένα τέτοιο παράδειγμα είναι τα πτυσσόμενα μενού.
- Αυξημένη δυνατότητα αλληλεπίδρασης: Ένα τέτοιο παράδειγμα όπου κάτι τέτοιο επιτυγχάνεται είναι τα μενού τα οποία αλληλεπιδρούν όταν ο χρήστης περάσει το mouse πάνω από αυτά, κάτι το οποίο έχει ως αποτέλεσμα να δημιουργηθεί μία σειρά από γεγονότα τα οποία έχουν προγραμματιστεί να λειτουργούν με έναν συγκεκριμένο τρόπο.
- Ελαφρότερα περιβάλλοντα: Αντί της απαίτησης download ενός μεγάλου αρχείου Java applet ή ενός Flash movie, τα προγράμματα γραμμένα σε JavaScript είναι μικρά σε μέγεθος και αποθηκεύονται στη μνήμη του browser μόλις κατέβουν. Επίσης είναι συμβατά σε microbrowsers, σε αντίθεση με το Flash.

Ο κώδικας της JavaScript είναι ενσωματωμένος σε ένα HTML αρχείο χρησιμοποιώντας την ετικέτα SCRIPT. Μπορούν να χρησιμοποιηθούν πολλαπλές ετικέτες

μέσα σε ένα αρχείο HTML, χρησιμοποιώντας πολλαπλές ετικέτες SCRIPT. Ένα παράδειγμα κομματιού κώδικα, μέσα σε μια HTML σελίδα, είναι το ακόλουθο²⁴:

```
<script language="JavaScript">
<!--
document.write("This is our new site!");
//-->
</script>
```

Ο κώδικας της JavaScript, όπως οι περισσότερες γλώσσες προγραμματισμού, έχει στη δομή του εντολές που χρησιμοποιούνται για αναθέσεις τιμών, σύγκριση τιμών, εντολές ελέγχου ροής και εκτέλεση σημείων του κώδικα.

Η JavaScript είναι μία βασιζόμενη σε αντικείμενα γλώσσα script η οποία τρέχει στην πλευρά του πελάτη (client-side), και μπορείτε να την χρησιμοποιείτε για να κάνετε πιο "δυναμικές" τις ιστοσελίδες σας. Για να μπορέσετε να βγάλετε νόημα από έναν τέτοιο ορισμό, θα πρέπει να εξετάσουμε τα σημαντικά μέρη του ένα προς ένα.

Βασιζόμενη σε Αντικείμενα

Ο όρος *βασιζόμενη σε αντικείμενα* σημαίνει ότι η JavaScript μπορεί να χρησιμοποιεί στοιχεία τα οποία αποκαλούνται *αντικείμενα (objects)*. Ωστόσο, τα αντικείμενα δεν *βασίζονται σε κλάσεις*, πράγμα που σημαίνει ότι δεν γίνεται καμία διάκριση μεταξύ μιας "κλάσης και ενός υποδείγματος της (ενός αντικειμένου), σε αντίθεση με τις γλώσσες αντικειμενοστραφούς προγραμματισμού. Τα αντικείμενα της JavaScript είναι απλώς αντικείμενα γενικού σκοπού.

Στην Πλευρά του Πελάτη

Ο όρος *"στην πλευρά του πελάτη (client)"* σημαίνει ότι η JavaScript τρέχει στην *εφαρμογή-πελάτη (client εφαρμογή)* που χρησιμοποιεί ένας χρήστης για να εμφανίσει μία σελίδα, και όχι στον Web server στον οποίο στεγάζεται αυτή η σελίδα. Σ' αυτή την περίπτωση, η client εφαρμογή είναι μία εφαρμογή Web browser. Για να αποσαφηνίσουμε περισσότερο τα πράγματα, θα εξετάσουμε πώς λειτουργεί μία βασιζόμενη στον server γλώσσα σε αντιδιαστολή με μία γλώσσα βασιζόμενη στην πλευρά του πελάτη (client).

²⁴ www.webstandards.org/learn/articles/askw3c/oct2003/, όπως αναρτήθηκε την 13^η Οκτωβρίου 2012.

Βασιζόμενες σε Server Γλώσσες

Μία βασιζόμενη σε server γλώσσα πρέπει να λαμβάνει πληροφορίες από την ιστοσελίδα ή την εφαρμογή Web browser, να τις στέλνει σε ένα πρόγραμμα το οποίο τρέχει στον server και κατόπιν να στέλνει τις πληροφορίες που είναι, το προϊόν της επεξεργασίας πίσω, στην εφαρμογή browser. Συνεπώς, πρέπει να υπάρχει ένα ενδιάμεσο βήμα το οποίο εκτελεί την αποστολή και την ανάκτηση των πληροφοριών από τον server, πριν παρουσιαστούν τα αποτελέσματα του προγράμματος στην εφαρμογή browser.

Συχνά, μία βασιζόμενη σε server γλώσσα δίνει στους προγραμματιστές δυνατότητες τις οποίες δεν υποστηρίζουν οι βασιζόμενες σε client γλώσσες, όπως για παράδειγμα η αποθήκευση πληροφοριών στον Web server για μελλοντική χρήση, ή η χρήση των νέων πληροφοριών για την ενημέρωση μιας ιστοσελίδας και η αποθήκευση των ενημερώσεων.

4.2 Βάσεις Δεδομένων και Διαδίκτυο

Ένα Σύστημα Διοίκησης Βάσης Δεδομένων (ΣΔΒΔ) (Database Management System) είναι ένα πακέτο λογικού (software package), που έχει σαν κύριους στόχους τη συστηματική αποθήκευση (storage), αναζήτηση (retrieval) και συντήρηση (maintenance) δεδομένων. Υπάρχουν πάρα πολλά πακέτα στην αγορά, που ταιριάζουν στον πιο πάνω ορισμό. Αξίζει να σημειώσουμε ότι πολλά από τα πακέτα αυτά, χαρακτηρίζονται από τους κατασκευαστές τους σαν Γενικευμένα ΣΔΒΔ, Συστήματα Επεξεργασίας Πληροφοριών (Information Processing Systems), κλπ.

Οι διάφορες επιχειρήσεις και Οργανισμοί χρησιμοποιούν τα ΣΔΒΔ σαν “εργαλεία” για την αποδοτικότερη λειτουργία, διοίκηση και προγραμματισμό τους. Αυτό σημαίνει ότι οι χρήστες (users) περιμένουν πολλά οφέλη από τη χρήση των ΣΔΒΔ όπως: βελτίωση της παραγωγικής διαδικασίας, ταχύτερη εξυπηρέτηση του πελάτη, ορθολογικότερες διοικητικές αποφάσεις, κλπ. Αυτό εξηγεί γιατί σήμερα λειτουργούν σε παγκόσμια κλίμακα περισσότερα από 250.000 ΣΔΒΔ και ξοδεύονται τεράστια ποσά για την ανάπτυξη, λειτουργία και συντήρηση τους (Allen, D.W. και Johnson, S., 1998).

Αν θελήσουμε να περιγράψουμε ένα τυπικό περιβάλλον λειτουργίας ενός ΣΔΒΔ, θα διαπιστώσουμε ότι υπάρχουν 4 κύρια συστατικά: δεδομένα (data), υλικό (hardware), λογικό (software) και χρήστες (users).

Η Βάση Δεδομένων (ΒΔ) είναι το τμήμα του ΣΔΒΔ που αποθηκεύει τα δεδομένα του Οργανισμού. Πριν από την εμφάνιση της τεχνολογίας των ΒΔ, τα δεδομένα φυλάσσονταν σ' ένα σύνολο διακεκριμένων αρχείων. Σήμερα, σε μια ΒΔ τα δεδομένα είναι ολοκληρωμένα (integrated) και καταμερισμένα (shared). Με τον όρο ολοκληρωμένα, εννοούμε ότι η ΒΔ αποτελείται από την ένωση διακεκριμένων αρχείων αλλά τα πλεονάζοντα (redundant) δεδομένα προσπαθούμε να τα αποθηκεύουμε όσο το δυνατόν λιγότερες φορές. Με τον όρο καταμερισμένα, εννοούμε ότι τα ίδια δεδομένα μπορούν να τα χρησιμοποιούν διαφορετικοί χρήστες σε διαφορετικές ή στην ίδια χρονική στιγμή (Allen, D.W. και Johnson, S., 1998).

Οι ΒΔ αποθηκεύονται, συνήθως, σε δευτερεύουσες μνήμες με δυνατότητες τυχαίας προσπέλασης (random access). Στη συνέχεια, θα υποθέτουμε ότι οι ΒΔ φυλάγονται σε μαγνητικούς δίσκους, με κινητές κεφαλές (magnetic disks with movable heads).

Η κύρια, όμως, λειτουργία του ΣΔΒΔ είναι το ότι επιτρέπει στους χρήστες να εκφράζουν τις απαιτήσεις τους, για πληροφόρηση, χωρίς να απασχολούνται καθόλου με τον τρόπο φύλαξης της ΒΔ στο δίσκο. Με την έννοια αυτή το ΣΔΒΔ λειτουργεί σαν ένας μεταφραστής, που δέχεται για είσοδο αυτό που επιθυμεί ο χρήστης και παράγει σαν αποτέλεσμα το πώς θα υλοποιηθεί η απαίτηση του χρήστη.

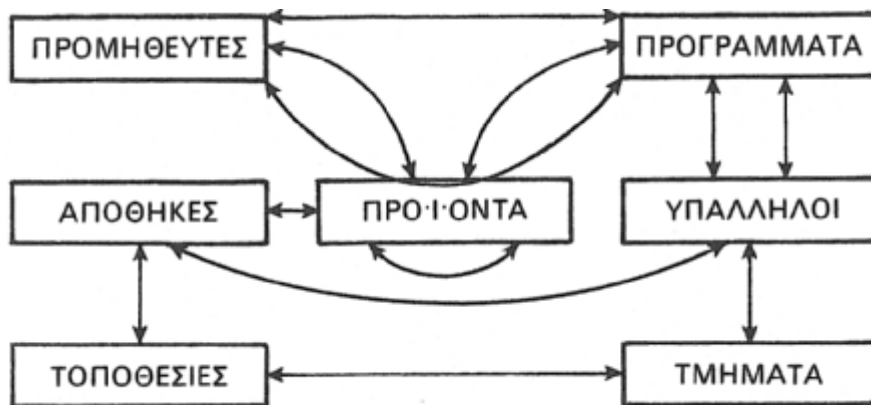
Σ' ένα ΣΔΒΔ, υπάρχει μεγάλη ποικιλία χρηστών οι κυριότεροι από τους οποίους είναι: ο *Προγραμματιστής Εφαρμογών* (Application Programmer), ο *τελικός χρήστης* (end user) και ο *Διαχειριστής της ΒΔ* (Database Administrator) (ΔΒΔ). Ο προγραμματιστής εφαρμογών είναι ο συμβατικός προγραμματιστής, που γνωρίζουμε, με τη διαφορά ότι τα προγράμματα του επεξεργάζονται τη ΒΔ αντί κάποια άλλη δομή δεδομένων (π.χ. αρχεία). Ο χρήστης είναι ένας χειριστής τερματικού σταθμού, που μπορεί να είναι από απλός υπάλληλος μέχρι ο Διοικητής του Οργανισμού. Η κατηγορία αυτή, των χρηστών, ζητά πληροφορίες από τη ΒΔ με τη χρήση κάποιας γλώσσας πολύ υψηλού επιπέδου, που αναφέρεται συνήθως σαν *γλώσσα ερωταπαντήσεων* ή *ερωτήσεων* (query language). Η χρήση μιας τέτοιας γλώσσας δεν απαιτεί συνήθως εξειδικευμένες γλώσσες προγραμματισμού. Το ρόλο του Διαχειριστή της ΒΔ θα τον εξετάσουμε διεξοδικά στη συνέχεια (Παπαδάκης, 2004).

4.2.1. Πλεονεκτήματα Βάσεων Δεδομένων

Στη συνέχεια, θα προσπαθήσουμε να παρουσιάσουμε τα πλεονεκτήματα της χρήσης των ΒΔ έναντι των συμβατικών μεθόδων φύλαξης και επεξεργασίας αρχείων. Πριν το κάνουμε, όμως, αυτό θα χρειαστεί να δώσουμε κάποιο ορισμό του τι είναι ακριβώς μια ΒΔ.

Ορισμός: Μια ΒΔ είναι μια συλλογή των λειτουργικών δεδομένων ενός οργανισμού. Τα δεδομένα αυτά φυλάγονται στις περιφερειακές μονάδες του Η/Υ και υφίστανται επεξεργασία από ένα αριθμό μηχανογραφικών εφαρμογών.

Στον ορισμό χρησιμοποιείται ο όρος *λειτουργικά* (operational) δεδομένα, για να διακρίνουμε τα δεδομένα της ΒΔ από άλλα, όπως είναι τα δεδομένα εισόδου και εξόδου. Πολλές φορές, στην πράξη, χρησιμοποιείται ο όρος *Τράπεζα Δεδομένων* (Data Bank) συνώνυμα με τον όρο ΒΔ. Στο βιβλίο αυτό έχουμε αποφύγει τη χρήση του όρου Τράπεζα Δεδομένων μια και ο όρος συνήθως αναφέρεται στο σύνολο των δεδομένων ενός Οργανισμού, τα οποία όμως δεν είναι απαραίτητο να τα επεξεργάζεται ένα ΣΔΒΔ. Τα λειτουργικά δεδομένα αφορούν κύρια (α) τις *οντότητες* (entities) που απασχολούν τον οργανισμό, και (β) τις *συσχετίσεις* (relationships), μεταξύ των οντοτήτων (Καλαφατούδης, 2004). Και τις δύο αυτές έννοιες πρέπει με κάποιο τρόπο να τις απεικονίσουμε σε μια ΒΔ. Στο παρακάτω σχήμα παρουσιάζουμε τα λειτουργικά δεδομένα μιας κατασκευαστικής εταιρίας.



Πηγή: Καλαφατούδης, Σ., Κοίλιας, Χ. και Μπανδήλα, Ε., *Εισαγωγή στην Πληροφορική και χρήση υπολογιστή*, Αθήνα, 2004

Με βάση το παραπάνω σχήμα θα κάνουμε τις ακόλουθες παρατηρήσεις:

Τα παραλληλόγραμμα αντιστοιχούν στις οντότητες Προμηθευτές, Αποθήκες, Τοποθεσίες, Προϊόντα, Προγράμματα (Projects), Υπάλληλοι και Τμήματα.

Τα βέλη αντιστοιχούν στις συσχετίσεις και είναι διπλής κατεύθυνσης για να δείξουν ότι η συσχέτιση μπορεί να λάβει χώρα κατά οποιαδήποτε κατεύθυνση. Για παράδειγμα, η σχέση Υπάλληλοι - Τμήματα μπορεί να χρησιμοποιηθεί για να απαντήσουμε τις ερωτήσεις (α) Βρες το Τμήμα όπου εργάζεται κάποιος Υπάλληλος, και (β) Βρες τους Υπάλληλους που εργάζονται σε κάποιο Τμήμα.

Τα βέλη, συνήθως, συσχετίζουν 2 οντότητες. Υπάρχει όμως στο σχήμα ένα βέλος που συσχετίζει τις 3 οντότητες Προμηθευτής, Προϊόν και Προγράμματα.

Υπάρχει ένα άλλο βέλος που συσχετίζει μόνο την οντότητα Προϊόν με τον εαυτό της. Αυτό ενδεχόμενα θέλει να δείξει ότι κάποια προϊόντα μπορεί να είναι στην πραγματικότητα συνάθροιση άλλων προϊόντων. Δηλαδή, σ' ένα ΣΔΒΔ μπορεί να έχουμε *αναδρομικούς* (recursive) τύπους δεδομένων.

Οι οντότητες Προγράμματα και Υπάλληλοι συσχετίζονται με 2 διαφορετικά βέλη. Αυτό συμβαίνει γιατί το πρώτο (δεύτερο) βέλος αφορά τη συσχέτιση όπου κάποιος υπάλληλος 'εργάζεται στο' ('διευθύνει το') χ πρόγραμμα.

Από το ανωτέρω σχήμα γίνεται φανερό ότι η ΒΔ δίνει τη δυνατότητα στον οργανισμό, να έχει κάποιο *συγκεντρωτικό έλεγχο* στα λειτουργικά του δεδομένα. Αυτό, βέβαια, βρίσκεται σε αντίθεση με τους συμβατικούς τρόπους μηχανογράφησης, όπου κάθε εφαρμογή χρησιμοποιεί τα δικά της αρχεία, τελείως ανεξάρτητα από άλλες εφαρμογές. Ένα άλλο αποτέλεσμα του συγκεντρωτισμού είναι ότι πρέπει να υπάρχει κάποιο άτομο που θα είναι υπεύθυνο, για τον έλεγχο των λειτουργικών δεδομένων. Το άτομο αυτό δεν είναι άλλο από το Διαχειριστή της ΒΔ (ΔΒΔ). Σημειώνουμε δε, ότι η θέση του ΔΒΔ είναι αρκετά υψηλή στην ιεραρχία του οργανισμού. Αυτό, δικαιολογείται και από το γεγονός ότι η 'φροντίδα' των λειτουργικών δεδομένων του οργανισμού έχει την ίδια (ή ίσως μεγαλύτερη) σημασία από τη διοίκηση των άλλων *ενεργητικών στοιχείων* (assets) του οργανισμού, π.χ. χρήματα, κτίρια, κλπ. Στη συνέχεια απαριθμούμε μερικά από τα πλεονεκτήματα της χρήσης ΒΔ, που σχετίζονται με την αποθήκευση των λειτουργικών δεδομένων (Καλαφατούδης, 2004).

Μείωση των πλεοναζόντων δεδομένων. Κάθε συμβατική μηχανογραφική εφαρμογή έχει τα δικά της αρχεία, με αποτέλεσμα, τη φύλαξη των *ίδιων* δεδομένων σε περισσότερες από μια εφαρμογές. Αυτό αυξάνει, κύρια, το κόστος φύλαξης και συντήρησης των επί πλέον δεδομένων. Με τη χρήση των ΒΔ ο ΔΒΔ μπορεί εύκολα να εντοπίσει τα πλεονάζοντα δεδομένα. Αξίζει να σημειώσουμε ότι, στην πράξη, υπάρχουν πολλοί λόγοι, όπου είναι επιθυμητό να έχουμε κάποιο πλεονασμό δεδομένων π.χ. ασφάλεια, επίδοση (performance), κλπ.

Αύξηση της συμβιβαστότητας των δεδομένων. Το πλεονέκτημα αυτό πηγάζει από το προηγούμενο. Ας υποθέσουμε ότι ο υπάλληλος Υ3 μετακινήθηκε στο τμήμα Τ8. Αν το τμήμα εργασίας των υπαλλήλων υπάρχει σε περισσότερα από ένα αρχεία τότε υπάρχει ο κίνδυνος να καταχωρηθεί η μετακίνηση του υπαλλήλου στο ένα αρχείο και να ξεχασθεί να καταχωρηθεί

σε κάποιο άλλο αρχείο. Στην περίπτωση αυτή, έχουμε *ασυμβίβαστα* (incosistent) δεδομένα. Η χρήση των ΒΔ αυξάνει την πιθανότητα να έχουμε συμβιβαστά δεδομένα.

Εύκολη ανάπτυξη νέων εφαρμογών. Αυτό συμβαίνει γιατί οι νέες εφαρμογές θα χρησιμοποιούν τη ΒΔ, οπότε δεν δαπανάται χρόνος για τη δημιουργία νέων αρχείων.

Διευκόλυνση εισαγωγής προτύπων. Ο συγκεντρωτικός έλεγχος των δεδομένων επιτρέπει στο ΔΒΔ να εισαγάγει *πρότυπα* (standards) για την υλοποίηση των εφαρμογών. Η τυποποίηση αυτή έχει πάρα πολλά πλεονεκτήματα, το κυριότερο από τα οποία είναι η εισαγωγή 'πειθαρχίας' στην ανάπτυξη των εφαρμογών. Ενδιαφέρον παρουσιάζει η περίπτωση της ανταλλαγής δεδομένων (data interchange), μεταξύ διαφόρων μηχανογραφικών κέντρων. (Σημ.: Είναι γνωστό ότι διάφοροι δημόσιοι οργανισμοί (π.χ. ΙΚΑ, ΜΗΚΥΟ, κλπ.) επιτρέπουν την ανταλλαγή στοιχείων με άλλους οργανισμούς κάνοντας χρήση μαγνητικών ταινιών).

Εύκολη ανάπτυξη μηχανισμών ασφάλειας. Σ' ένα ΣΔΒΔ ο Δ ΒΔ μπορεί να ανιχνεύσει τους διάφορους εναλλακτικούς τρόπους, με τους οποίους, οι χρήστες έχουν τη δυνατότητα να προσπελάσουν, σε κάποιο δεδομένο. Αυτό διευκολύνει την ανάπτυξη μηχανισμών *ασφαλείας* (security) για τη χρήση (π.χ. αναζήτηση, τροποποίηση, κλπ.) των δεδομένων.

Βελτίωση της ορθότητας των δεδομένων. Με τον όρο *ορθότητα* (integrity) των δεδομένων, εννοούμε ότι κάθε δεδομένο είναι σωστά καταχωρημένο στη ΒΔ. Ο συγκεντρωτισμός των ΒΔ επιτρέπει στο ΔΒΔ να εισάγει τους απαραίτητους αυτοματοποιημένους ελέγχους (validation procedures), που μειώνουν, κατά πολύ, το ενδεχόμενο λαθεμένων καταχωρήσεων. Για παράδειγμα, οι έλεγχοι αυτοί απαγορεύουν την καταγραφή περισσοτέρων από 25 ώρες υπερωρίες τη βδομάδα, κλπ.

Συμβιβασμός των απαιτήσεων χρήσης. Δεν υπάρχει αμφιβολία ότι κάθε είδος χρήσης μπορεί να υλοποιηθεί με διάφορους εναλλακτικούς τρόπους, που χρησιμοποιούν όμως διαφορετικές ποσότητες από τα διαθέσιμα μέσα (resources) του Η/Υ. Ο ΔΒΔ επιλέγει εκείνους τους τρόπους ικανοποίησης των χρήσεων, που φαίνεται ότι είναι ο 'καλύτερος' σε επίπεδο Οργανισμού και όχι σε επίπεδο μεμονωμένης χρήσης.

Το μεγαλύτερο, όμως, πλεονέκτημα των ΒΔ είναι η εξασφάλιση της *ανεξαρτησίας των δεδομένων* (data independence), μια έννοια που αναπτύσσουμε αμέσως.

4.2.2. Ανεξαρτησία Δεδομένων

Στην αρχή, θα εξηγήσουμε την πρόταση «Οι κλασσικές μηχανογραφικές εφαρμογές έχουν εξαρτημένα δεδομένα». Αυτό σημαίνει ότι τα προγράμματα εφαρμογών

(κωδικοποιημένα σε COBOL, PASCAL, κλπ.) γνωρίζουν τον τρόπο οργάνωσης και προσπέλασης των δεδομένων και ο τρόπος αυτός είναι ενσωματωμένος στη λογική των προγραμμάτων. Για παράδειγμα, το αρχείο X είναι οργανωμένο σαν σειριακό αρχείο με δείκτη (index sequential) και στο πρόγραμμα A το επεξεργαζόμαστε σειριακά. Ας υποθέσουμε τώρα ότι το αρχείο X αλλάζει δόμηση και οργάνωση σαν ένα τυχαίο (random file). Είναι φανερό ότι το πρόγραμμα A χρειάζεται να ξαναγραφεί (σχεδόν) από την αρχή για να επεξεργαστεί το αρχείο X.

Με βάση το πιο πάνω παράδειγμα, ορίζουμε σαν ανεξαρτησία των δεδομένων την *απαιτήση να μην τροποποιούνται τα προγράμματα εφαρμογών όταν αλλάζει ο τρόπος οργάνωσης και προσπέλασης των δεδομένων*. Στη συνέχεια, δίνουμε δύο αιτιολογίες γιατί αυτό είναι επιθυμητό σ' ένα ΣΔΒΔ.

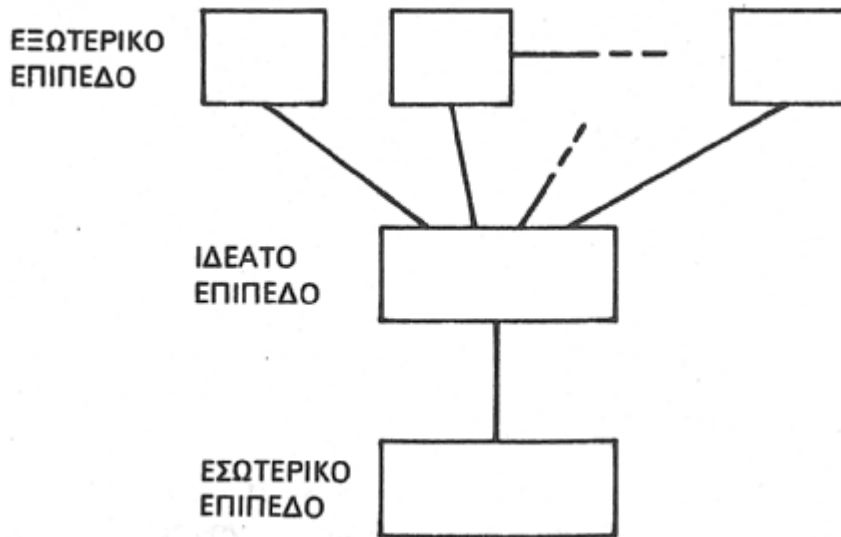
1. Διαφορετικές εφαρμογές μπορεί να απαιτούν άλλη απεικόνιση των αυτών δεδομένων. Για παράδειγμα η Εφαρμογή A θέλει το πεδίο μιας εγγραφής με το όνομα MISTHOS να είναι δυαδικό ενώ η Εφαρμογή B το θέλει δεκαδικό. Αυτό ίσως συμβαίνει γιατί η Εφαρμογή A(B) χρησιμοποιεί κύρια το πεδίο MISTHOS σε αριθμητικές πράξεις (σε εκτυπώσεις). Ο ΔΒΔ μπορεί να επιλέξει να φυλάζει στο δίσκο το πεδίο σαν δυαδικό αριθμό αλλά κάθε αναφορά στο πεδίο αυτό από την Εφαρμογή B θα έχει σαν αποτέλεσμα το ΣΔΒΔ να κάνει αυτόματα την απαιτούμενη μετατροπή. Το παράδειγμα μας δείχνει ότι μια εφαρμογή (δηλ. η B) μπορεί να 'βλέπει' ένα δεδομένο με τρόπο διαφορετικό από εκείνο, με τον οποίο έχει φυλαχτεί στην πραγματικότητα.

2. Ο ΔΒΔ επιθυμεί να τροποποιήσει τη φυσική δομή των δεδομένων ή/και τον τρόπο προσπέλασης των δεδομένων. Αυτό μπορεί να συμβεί γιατί άλλαξαν οι δίσκοι του υπολογιστή, τροποποιήθηκαν οι προτεραιότητες επεξεργασίας, κλπ. Είναι φανερό ότι τέτοιες αλλαγές μπορούν, πρακτικά, να γίνουν μόνον όταν οι εφαρμογές έχουν την ιδιότητα της ανεξαρτησίας των δεδομένων.

Για να εκτιμήσουμε καλύτερα τη σημασία της ανεξαρτησίας των δεδομένων θα αναφέρουμε ότι υπάρχουν περιπτώσεις όπου το 25% της προγραμματιστικής προσπάθειας, που καταβάλλεται σε μεγάλα μηχανογραφικά κέντρα, θα είχε αποφευχθεί αν οι εφαρμογές δεν είχαν εξαρτημένα δεδομένα. Στη συνέχεια, παρουσιάζουμε μια αρχιτεκτονική ενός ΣΔΒΔ που προσφέρεται για την επιτυχία της ανεξαρτησίας δεδομένων (Παπαδάκης, 2004).

4.2.3. Μια αρχιτεκτονική συστημάτων Βάσεων Δεδομένων

Στο κάτωθι σχήμα παρουσιάζουμε μια αρχιτεκτονική ΣΔΒΔ που φαίνεται ικανοποιητική για την περιγραφή των βασικών χαρακτηριστικών των σημερινών και των μελλοντικών ΣΔΒΔ. Η αρχιτεκτονική αυτή είναι παρόμοια μ' αυτή που προτάθηκε από την ANSI /SPARC (Study Group on Database Management Systems) και αποτελεί, ίσως, το 'πρότυπο' για τη μελέτη των ΣΔΒΔ.



Πηγή: Καλαφατούδης, Σ., Κοίλιας, Χ. και Μπανδήλα, Ε., Εισαγωγή στην Πληροφορική και χρήση υπολογιστή, Αθήνα, 2004

Η αρχιτεκτονική αποτελείται από 3 επίπεδα: το *εξωτερικό* (external), το *ιδεατό* (conceptual) και το *εσωτερικό* (internal). Το εσωτερικό επίπεδο αφορά στον τρόπο φύλαξης των δεδομένων, στις περιφερειακές μονάδες. Το εξωτερικό επίπεδο έχει να κάνει με τον τρόπο που οι διάφοροι χρήστες *βλέπουν* (φαντάζονται ή οραματίζονται) (view) τα δεδομένα της βάσης. Το ιδεατό επίπεδο βρίσκεται μεταξύ των 2 άλλων επιπέδων. Αν το εξωτερικό επίπεδο περιγράφει την όψη των δεδομένων από την σκοπιά των *μεμονωμένων* χρηστών (individual users) τότε το ιδεατό επίπεδο την περιγράφει από τη σκοπιά του *συνόλου* των χρηστών (user community). Με άλλα λόγια, υπάρχουν *πολλές* εξωτερικές όψεις της ΒΔ (όπου κάθε μία περιγράφει ένα τμήμα της ΒΔ) αλλά υπάρχει *μία* ιδεατή όψη (που περιγράφει όλη τη ΒΔ). Τέλος, παρατηρούμε ότι υπάρχει *μία* εσωτερική όψη (view).

Πριν προχωρήσουμε, σε λεπτομέρειες, θα προσπαθήσουμε να βρούμε κάποια αναλογία της έννοιας των 3 επιπέδων της αρχιτεκτονικής με το χειρισμό των πινάκων σε μια

γλώσσα υψηλού επιπέδου (π.χ. PASCAL). Στο ιδεατό επίπεδο μπορεί να έχουμε τον ορισμό του πίνακα:

A : array [1..m, 1..n] of integer;

Αν ο πίνακας έχει φυλαχτεί στη μνήμη του Η/Υ κατά στήλες σε συνεχόμενες θέσεις μνήμης τότε στο εσωτερικό επίπεδο μας ενδιαφέρει η θέση του στοιχείου A[i,j], Είναι γνωστό ότι η θέση αυτή ισούται με

$$\text{loc}(A[1,1]) + n(i-1) + j-1,$$

όπου $\text{loc}(A[1,1])$ είναι η θέση που έχει φυλαχτεί το στοιχείο A[1,1]. Στο εξωτερικό επίπεδο ένας χρήστης μπορεί να ορίσει μια συνάρτηση f(i) σαν το άθροισμα από j = 1 έως n των A[i,j]. Με τον τρόπο αυτό ο χρήστης αναφέρεται στο άθροισμα των στοιχείων μιας γραμμής και όχι στα μεμονωμένα στοιχεία της γραμμής, φυσικά ο χρήστης δεν χρειάζεται να γνωρίζει απολύτως τίποτε για τον τρόπο φύλαξης των δεδομένων.

Στο σχήμα που παρατίθεται κατωτέρω δίνουμε ένα παράδειγμα της περιγραφής των 3 επιπέδων της αρχιτεκτονικής. Η περιγραφή αφορά το μισθό των υπαλλήλων που εργάζονται σε διάφορα τμήματα ενός οργανισμού.

Με βάση το παρακάτω σχήμα, παρατηρούμε (Καλαφατούδης, 2004):

1. Στο εξωτερικό επίπεδο, υποθέτουμε την ύπαρξη 2 χρηστών που φαντάζονται αντίστοιχα τη ΒΔ σαν κάποια εγγραφή της PL/1 και COBOL.

2. Στο ιδεατό επίπεδο, η ΒΔ αποτελείται από μια περιγραφή της οντότητας Υπάλληλος, σε μορφή εγγραφής.

3. Στο εσωτερικό επίπεδο, δίνεται η περιγραφή της οντότητας όπως αυτή φυλάγεται στο δίσκο. Η λέξη BYTE, OFFSET, INDEX, κλπ. αφορούν τον τρόπο υλοποίησης της ΒΔ. Η εγγραφή στο επίπεδο αυτό συχνά αναφέρεται σαν *φυσική εγγραφή* (physical record).

4. Στο εξωτερικό επίπεδο οι χρήστες βλέπουν ένα μόνο μέρος της οντότητας, όπως αυτή απεικονίζεται στο ιδεατό (ή εσωτερικό) επίπεδο. Μια εγγραφή του ιδεατού ή εξωτερικού επιπέδου αναφέρεται και σαν *λογική εγγραφή* (logical record). Ένα αξιοσημείωτο γεγονός αυτού του επιπέδου είναι ότι ο πρώτος χρήστης θεωρεί ότι η οντότητα αποτελείται από τα πεδία Κωδικός Υπαλλήλου (YPAL#) και Μισθός (MIS) ενώ ο δεύτερος βλέπει την εγγραφή να περιέχει μόνο τα πεδία Κωδικός Υπαλλήλου (YPALAR) και Αρ. Τμήματος (TMHMAR). Αξίζει να σημειώσουμε, επίσης, ότι τα πεδία της εγγραφής έχουν, για τον κάθε χρήστη, διαφορετική ονομασία και προδιαγραφές αποθήκευσης. Κωδικός Υπαλλήλου (YPAL#) και Μισθός (MIS) ενώ ο δεύτερος βλέπει την εγγραφή να περιέχει μόνο τα πεδία Κωδικός

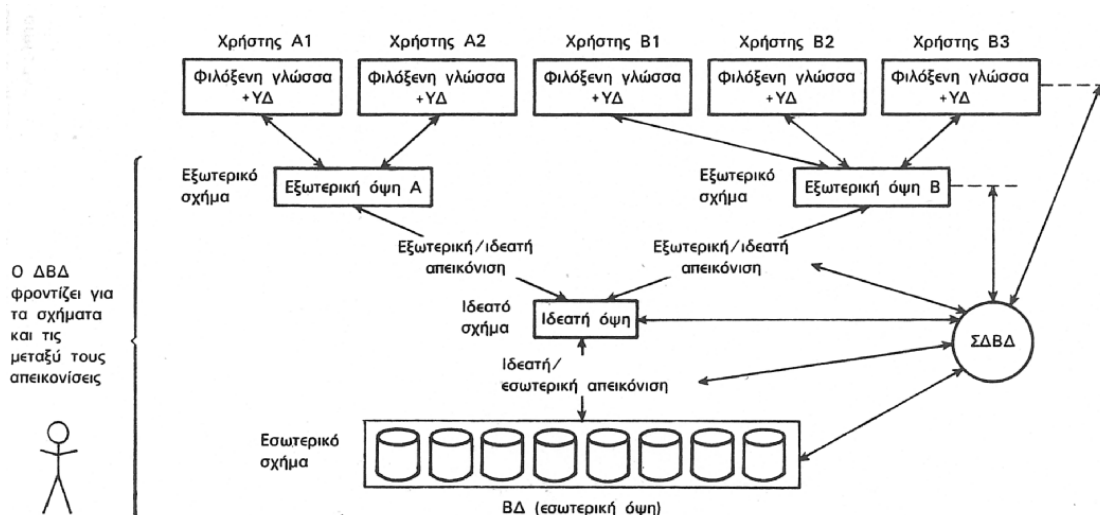
Υπαλλήλου (YPALAR) και Αρ. Τμήματος (TMHMAR). Αξίζει να σημειώσουμε, επίσης, ότι τα πεδία της εγγραφής έχουν, για τον κάθε χρήστη, διαφορετική ονομασία και προδιαγραφές αποθήκευσης.

<p><i>Εξωτερικό (PL / 1)</i></p> <p>DCL 1 YPALP, 2 YPAL# CHAR(6), 2 MIS FIXED BIN(31);</p>	<p><i>Εξωτερικό (COBOL)</i></p> <p>01 YPALC. 02 YPALAR PIC X(6). 02 TMHMAR PIC X(4).</p>
<p><i>Ιδεατό</i></p> <p>YPALLHLOS ARITH-YPALLHLOY CHARACTER (6) ARITH-TMHMATOS CHARACTER (4) MISTHOS NUMERIC (5)</p>	
<p><i>Εσωτερικό</i></p> <p>STORED-YPAL LENGTH = 18 PREFIX TYPE = BYTE (6), OFFSET = 0 YPAL # TYPE = BYTE (6), OFFSET = 6, INDEX = YPALX TMHM # TYPE = BYTE (4), OFFSET = 12 MISTH TYPE = FULLWORD, OFFSET = 16</p>	

Πηγή: Πολλάλης, Γ. Α., Γιαννακόπουλος, Δ. και Παπουτσής, Ι., 1999.

Στο σχήμα που ακολουθεί παρουσιάζεται η αρχιτεκτονική, όπως αυτής απεικονίζεται στην δεύτερη εικόνα του κεφαλαίου, με πολύ περισσότερες λεπτομέρειες. Πολλές φορές θα αναφερόμαστε στην αρχιτεκτονική σαν αρχιτεκτονική ANSI SPARC.

Με βάση την αρχιτεκτονική του παρακάτω σχήματος, θα παρατηρήσουμε: Υπάρχουν πολλοί χρήστες, που μπορούν να είναι από προγραμματιστές μέχρι απλοί χρήστες, που κάνουν προσπέλαση στη ΒΔ με *μαζικά* (batch) ή *άμεσα* (on line) προγράμματα. Κάθε χρήστης έχει στη διάθεση του μια γλώσσα. Η γλώσσα αυτή μπορεί να είναι είτε μια γλώσσα ερωταπαντήσεων (query language) είτε μια συμβατική γλώσσα (π.χ. COBOL) που έχει επεκταθεί κατά τρόπο ώστε να μπορεί να επεξεργάζεται τη ΒΔ. Η συμβατική γλώσσα ονομάζεται *φιλόξενη* (host) ενώ η επέκταση της που της επιτρέπει να χειρίζεται τη ΒΔ λέγεται Υπογλώσσα Δεδομένων (ΥΔ) ή DSL (Data Sublanguage).



Πηγή: Πολλάλης, Γ. Α., Γιαννακόπουλος, Δ. και Παπουτσής, Ι., 1999.

Η ΥΔ αποτελείται από 2 τμήματα: τη *Γλώσσα Ορισμού Δεδομένων* (ΓΟΔ) (Data Definition Language - DDL) και τη *Γλώσσα Χειρισμού Δεδομένων* (ΓΧΔ) (Data Manipulation Language - DML). Τις περισσότερες φορές η ΓΟΔ χρησιμοποιείται, εξωτερικά, από το πρόγραμμα ενώ η ΓΧΔ είναι στην πραγματικότητα μια σειρά από κλήσεις (calls) σε υπορουτίνες του ΣΔΒΔ. Τέλος, κάθε χρήστης έχει στη διάθεση του μια *περιοχή εργασίας* (work space), όπου τοποθετούνται όλα τα δεδομένα της επικοινωνίας με τη ΒΔ (Καλαφατούδης, 2004).

Από την αρχιτεκτονική του παραπάνω σχήματος, φαίνεται ακόμη, ότι κάθε χρήστης ή ομάδα από χρήστες έχουν μια *εξωτερική όψη* (external view) της ΒΔ, ο δε προγραμματισμός τους γίνεται σύμφωνα με αυτή την όψη. Κάθε διακεκριμένη εξωτερική όψη περιγράφεται (με τη χρήση της ΓΟΔ) και ονομάζεται *εξωτερικό σχήμα* (external schema). Η ΓΧΔ επεξεργάζεται τις *εξωτερικές εγγραφές* (external records). Στο *ιδεατό επίπεδο* έχουμε, αντίστοιχα, τις έννοιες του *ιδεατού σχήματος* (conceptual schema) και της *ιδεατής εγγραφής* (conceptual record). Αξίζει να σημειώσουμε ότι η περιγραφή της ιδεατής εγγραφής δεν αποτελείται μόνο από το όνομα και το μέγεθος των πεδίων της αλλά μπορεί να περιέχει επίσης και άλλες πληροφορίες που είναι χρήσιμες για την ορθότητα των δεδομένων, την προστασία τους, κλπ. Ενδεικτικά αναφέρουμε την ύπαρξη ψηφίου ελέγχου (check digit) και τον καθορισμό των οριακών τιμών που μπορεί να λάβει κάποιο πεδίο (Πολλάλης, 1999).

Το τρίτο επίπεδο της αρχιτεκτονικής αφορά στη φυσική υλοποίηση της ΒΔ, δηλαδή την *εσωτερική όψη* (internal view). Η όψη αυτή ορίζει το *εσωτερικό σχήμα* (internal

schema) και περιγράφεται με την *εσωτερική εγγραφή* (internal record) ή τη *φυσική εγγραφή*. Παρατηρούμε ότι για την περιγραφή της ιδεατής και της φυσικής εγγραφής υπάρχουν, αντίστοιχα, διαθέσιμες οι *ιδεατές ΓΟΔ* και οι *φυσικές ΓΟΔ*.

Παρατηρούμε, ότι σ' ένα ΣΔΒΔ έχουμε δύο είδη γλωσσών προγραμματισμού, από τα οποία το ένα αφορά τους ορισμούς των δεδομένων (Σχ. 1.3) και το άλλο είδος αναφέρεται στην επεξεργασία των δεδομένων. Οι ΓΟΔ χρησιμοποιούνται *μόνο* κατά το σχεδιασμό ή την τροποποίηση του σχεδιασμού μιας εφαρμογής. Ο λόγος, που οι συμβατικές γλώσσες προγραμματισμού εμφανίζουν ενοποιημένα τους ορισμούς και τις επεξεργασίες των δεδομένων είναι γιατί κάνουν την παραδοχή ότι τα δεδομένα παύουν να έχουν νόημα, μετά την εκτέλεση του προγράμματος. Σ' ένα ΣΔΒΔ, όμως, τα δεδομένα εξακολουθούν να υπάρχουν στη ΒΔ. (Σημ: Η γλώσσα προγραμματισμού COBOL έχει την εντολή COPY που επιτρέπει να γίνονται οι ορισμοί των αρχείων εξωτερικά από ένα πρόγραμμα).

Με αναφορά και πάλι στην τελευταία εικόνα παρατηρούμε ότι υπάρχουν 2 *απεικονίσεις ή μετασχηματισμοί* (mappings): (α) η απεικόνιση ιδεατού/ εσωτερικού, και (β) η εξωτερικού/ιδεατού σχήματος. Αυτοί οι μετασχηματισμοί στην πραγματικότητα, εξασφαλίζουν την ανεξαρτησία των δεδομένων. Για παράδειγμα, η απεικόνιση ιδεατού/εσωτερικού περιγράφει πώς μια ιδεατή όψη έχει υλοποιηθεί σε ένα ή περισσότερους δίσκους. Κάθε αλλαγή, στη φυσική υλοποίηση της ΒΔ, έχει σαν αποτέλεσμα να αλλάξει η απεικόνιση. Αυτό μπορεί να το κάνει ο ΔΒΔ, χωρίς να χρειαστεί καμιά τροποποίηση των προγραμμάτων εφαρμογής (Πολλάλης, 1999).

Ένα κύριο τμήμα της αρχιτεκτονικής είναι το ΣΔΒΔ, που είναι υπεύθυνο για την ορθή διεκπεραίωση των απαιτήσεων χρήσης. Συγκεκριμένα, το ΣΔΒΔ κάνει τις ακόλουθες διαδικασίες (α) δέχεται μια περιγραφή χρήσης σε κάποια (εξωτερική) ΓΧΔ, (β) ελέγχει την ορθότητα της χρήσης, (γ) συμβουλευεται το εξωτερικό σχήμα, τον εξωτερικό/ιδεατό μετασχηματισμό, το ιδεατό σχήμα, τον ιδεατό/εσωτερικό μετασχηματισμό και το εσωτερικό σχήμα, και (δ) εκτελεί τις απαραίτητες ενέργειες για την ικανοποίηση της χρήσης (Παπαδάκης, 2004).

4.3 Συστήματα Διαχείρισης Περιεχομένου

Τα Συστήματα Διαχείρισης Περιεχομένου (ΣΔΠ, Content Management Systems, CMS) είναι διαδικτυακές εφαρμογές που επιτρέπουν την online τροποποίηση του περιεχομένου ενός δικτυακού τόπου.

Οι διαχειριστές μέσω του διαδικτύου ενημερώνουν το περιεχόμενο στο ΣΔΠ, το οποίο είναι εγκατεστημένο σ' ένα διακομιστή. Οι αλλαγές αυτές γίνονται αυτόματα διαθέσιμες πάλι μέσω του διαδικτύου, σε όλους τους επισκέπτες και χρήστες του δικτυακού τόπου.

Η διαχείριση περιεχομένου ιστοσελίδων (WCM) είναι ένα σύστημα CMS σχεδιασμένο για να απλοποιήσει τη δημοσίευση του δικτυακού περιεχομένου των web sites και των κινητών συσκευών, μεταξύ άλλων, επιτρέπει στους δημιουργούς περιεχομένου να υποβάλουν περιεχόμενο χωρίς να απαιτεί τεχνική γνώση HTML ή μεταφόρτωση των αρχείων.

Πολλά web-based συστήματα διαχείρισης περιεχομένου υφίστανται τόσο Ανοιχτού Κώδικα όσο και με εμπορικά κατοχυρωμένα πνευματικά δικαιώματα. Ωστόσο, αυτός είναι ένας τομέας στον οποίο το OSS (Open Source Software) έχει αποκτήσει δεσπόζουσα θέση έναντι των ιδιοκτησιακής μορφής ομολόγων του (Παπαδάκης, 2004).

Το Σύστημα Διαχείρισης Περιεχομένου αυτοματοποιεί διάφορες διαδικασίες όπως η διατήρηση της εμφάνισης των σελίδων σε όλο το site, καθώς και η δημιουργία των σχετικών μενού, συνδέσμων και τα λοιπά. Προσφέρει έναν εύκολο και εύχρηστο τρόπο να ενημερώνει το περιεχόμενό τους. Το ίδιο απλό είναι και η προσθήκη νέων σελίδων, η διαγραφή παλαιότερων καθώς και η αναδιαμόρφωση του site ώστε να συμβαδίζει με τις αλλαγές στην επιχείρηση ή τις δραστηριότητες που εκφράζει.

Στόχος των Συστημάτων Διαχείρισης Περιεχομένου είναι να απαλλάξει τους χρήστες από τη διαδικασία της κατανόησης δημιουργίας ενός ιστοτόπου με τεχνολογίες όπως η HTML, η XML ή η DHTML. Το Σύστημα Διαχείρισης Περιεχομένου προσφέρει γραφικό περιβάλλον το οποίο δίνει άμεση πρόσβαση στο περιεχόμενο της ιστοσελίδας. Επιπρόσθετα, η τροποποίηση ή προσθήκη του περιεχομένου (κειμένου και φωτογραφιών) μπορεί να γίνει με έναν γραφικό editor όμοιο με αυτόν που χρησιμοποιείται στους κοινούς επεξεργαστές κειμένου. Η πληροφορία οργανώνεται αποδοτικά σε κατηγορίες και υποκατηγορίες και παρουσιάζεται με τρόπο φιλικό στο χρήστη αλλά και στο διαχειριστή (Καλαφατούδης, 2004).

Ο όρος Content Management Systems (CMS, Συστήματα Διαχείρισης Περιεχομένου) αναφέρεται στις εφαρμογές που επιτρέπουν στον πελάτη να διαχειρίζεται το δικτυακό του περιεχόμενο, όπως κείμενα, εικόνες, πίνακες κ.λπ., με εύκολο τρόπο, συνήθως παρόμοιο με αυτόν της χρήσης ενός κειμενογράφου. Οι εφαρμογές διαχείρισης περιεχομένου επιτρέπουν την αλλαγή του περιεχομένου χωρίς να είναι απαραίτητες ειδικές γνώσεις σχετικές με τη δημιουργία ιστοσελίδων ή γραφικών, καθώς συνήθως τα κείμενα γράφονται μέσω κάποιων online WYSIWYG ("What You See Is What You Get") html editors, ειδικών δηλαδή κειμενογράφων, παρόμοιων με το MS Word, που επιτρέπουν τη μορφοποίηση των κειμένων όποτε υπάρχει ανάγκη.

Οι αλλαγές του site μπορούν να γίνουν από οποιονδήποτε υπολογιστή που είναι συνδεδεμένος στο Διαδίκτυο, χωρίς να χρειάζεται να έχει εγκατεστημένα ειδικά προγράμματα επεξεργασίας ιστοσελίδων, γραφικών κ.λπ. Μέσω ενός απλού φυλλομετρητή ιστοσελίδων (browser), ο χρήστης μπορεί να συντάξει ένα κείμενο και να ενημερώσει άμεσα το δικτυακό του τόπο. Αυτό που αποκαλούμε πολλές φορές "δυναμικό περιεχόμενο" σε ένα website δεν είναι άλλο παρά οι πληροφορίες που παρουσιάζονται στο site και μπορούν να αλλάξουν από τους ίδιους τους διαχειριστές του μέσω κάποιας εφαρμογής, η οποία ουσιαστικά μπορεί να εισάγει (προσθέτει), διορθώνει και να διαγράφει εγγραφές σε πίνακες βάσεων δεδομένων, όπου τις περισσότερες φορές καταχωρούνται όλες αυτές οι πληροφορίες. Αυτό σημαίνει ότι δεν χρειάζεται να δημιουργηθούν πολλές ξεχωριστές ιστοσελίδες για την παρουσίαση των πληροφοριών στο site, αλλά αρκεί ένας ενιαίος σχεδιασμός στα σημεία όπου θέλουμε να εμφανίζεται το περιεχόμενό μας, καθώς και να υπάρχει ο ειδικός σε κάποια συγκεκριμένη γλώσσα προγραμματισμού (ASP, PHP, Coldfusion, Perl, CGI κ.λπ.), ο οποίος αναλαμβάνει να εμφανίσει τις σωστές πληροφορίες στις σωστές θέσεις (Ζαφείρη, 2005).

Έτσι, για το δικτυακό τόπο μιας εφημερίδας π.χ., που απαιτεί εύλογα καθημερινή ενημέρωση αλλά δεν χρησιμοποιεί κάποιο σύστημα Content Management, θα πρέπει ο υπεύθυνος για το σχεδιασμό του (designer) να δημιουργήσει μια σελίδα με τα γραφικά, την πλοήγηση και το περιβάλλον διεπαφής (interface) του website, ο υπεύθυνος ύλης να τοποθετήσει το περιεχόμενο στα σημεία της ιστοσελίδας που θέλει, και να ενημερωθούν οι σύνδεσμοι των υπόλοιπων σελίδων ώστε να συνδέονται με την καινούργια. Αφού την αποθηκεύσει, πρέπει να την ανεβάσει στο website μαζί με τις υπόλοιπες ιστοσελίδες που άλλαξαν.

Αντιθέτως, αν ο δικτυακός τόπος λειτουργεί με χρήση κάποιου συστήματος CM, το μόνο που έχει να κάνει ο διαχειριστής του είναι να ανοίξει τη σχετική φόρμα εισαγωγής νέου άρθρου στη διαχειριστική εφαρμογή του website και να γράψει ή να επικολλήσει (copy-paste) τα στοιχεία που επιθυμεί. Αυτόματα, μετά την καταχώριση γίνονται από το ίδιο το σύστημα διαχείρισης περιεχομένου όλες οι απαραίτητες ενέργειες, ώστε το άρθρο να είναι άμεσα διαθέσιμο στους επισκέπτες και όλοι οι σύνδεσμοι προς αυτό ενημερωμένοι.

Με την αυξητική τάση χρήσης των CMS στην Ελλάδα και το εξωτερικό, γίνεται εμφανές ότι το μέλλον του Διαδικτύου σε ό, τι αφορά περιεχόμενο και πληροφορίες που πρέπει να ανανεώνονται τακτικά, ανήκει στα προγράμματα διαχείρισης περιεχομένου, αφού προσφέρουν πολλά πλεονεκτήματα, ταχύτητα και ευκολίες στη χρήση τους (Ζαφείρη, 2005).

ΚΕΦΑΛΑΙΟ 5: ΕΦΑΡΜΟΓΕΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ WEB 2.0

5.1. Παγκόσμιος Πληροφοριακός Ιστός (World Wide Web-WWW)

Ο Παγκόσμιος Πληροφοριακός Ιστός (World Wide Web, WWW ή W3) θεωρείται η ταχύτερα αναπτυσσόμενη υπηρεσία του Διαδικτύου. Αποτελεί μία σχετικά πρόσφατη δικτυακή υπηρεσία που γνώρισε όμως καθολική αποδοχή. Για το λόγο αυτό μάλιστα μέσα σε μικρό χρονικό διάστημα, πραγματοποιήθηκε η μετάβαση στο web 2.0, το οποίο επίσης εξελίσσεται στο web 3.0. Η επιτυχία του web ήταν τόσο σημαντική που οδήγησε στη ραγδαία εξάπλωση του Διαδικτύου και μάλιστα ο περισσότερος κόσμος, θεωρεί λανθασμένα το web συνώνυμο του Διαδικτύου (Φούσκας, 2002).

Ο Παγκόσμιος Ιστός είναι ουσιαστικά μία τεράστια συλλογή πληροφοριών, αποθηκευμένων σε διάφορες μορφές (κείμενο, εικόνα, ήχος, γραφικά κτλ.). Αυτή η συλλογή είναι κατανομημένη και διατίθεται στο Διαδίκτυο μέσω ειδικών εφαρμογών, τους εξυπηρετητές Παγκοσμίου Ιστού (World Wide Web servers). Με άλλα λόγια, θα μπορούσε κανείς να ισχυριστεί ότι πρόκειται για ένα εργαλείο αναζήτησης και παρουσίασης των πληροφοριών που βρίσκονται στο Διαδίκτυο.

Αρχικά δημιουργήθηκε για τη διακίνηση επιστημονικών πληροφοριών μεταξύ πανεπιστημιακών ομάδων που βρίσκονταν σε γεωγραφικά διασκορπισμένες περιοχές. Βασίζεται στο αρχιτεκτονικό μοντέλο «πελάτη / εξυπηρετητή» (client/server) και χρησιμοποιεί την τεχνική του υπερκειμένου (hypertext) για τη δημιουργία πληροφοριακών αντικειμένων, των γνωστών ιστοσελίδων. Οι ιστοσελίδες διαμορφώνονται από κείμενο, εικόνα (στατική ή κινούμενη) και ήχο. Επομένως, το web υιοθετεί ένα εξαιρετικά αποτελεσματικό γραφικό περιβάλλον επικοινωνίας με το χρήστη και επιτρέπει την απεικόνιση και μετάδοση οποιουδήποτε τύπου πληροφορίας. Το περιβάλλον αυτό παρέχει τη δυνατότητα ενεργοποίησης διαφόρων συνδέσμων (hyperlinks) οι οποίοι οδηγούν σε πληροφοριακά αντικείμενα (ιστοσελίδες), οπουδήποτε και αν αυτές βρίσκονται στο Διαδίκτυο (Αδαμόπουλος, 2002).

Ειδικότερα, το υπερκείμενο είναι μια μορφή παρουσίασης ενός γραπτού κειμένου στην οποία η διαδοχή των τμημάτων του δεν ακολουθεί κατά ανάγκη τη φυσική σειρά παρουσίασης που επιβάλλεται από τη σελιδοποίηση του κειμένου. Οι αυτοτελείς ενότητες υπερκειμένου που εμφανίζονται στην οθόνη του υπολογιστή ονομάζονται ιστοσελίδες.

Τμήματα τα οποία παραπέμπουν σε άλλα τμήματα της ίδιας ή άλλων ιστοσελίδων ονομάζονται κόμβοι. Οι αναφορές ή παραπομπές ενός τμήματος σε ένα άλλο ονομάζονται σύνδεσμοι (links).

Με τη χρήση των συνδέσμων ο αναγνώστης ενός υπερκειμένου δε διαβάζει απλά ένα κείμενο, αλλά έχει τη δυνατότητα να «κινείται» μέσα σε αυτό. Ο αναγνώστης δηλαδή «περιηγείται» ανάμεσα στους κόμβους μιας ιστοσελίδας, αλλά και όλων των άλλων ιστοσελίδων στις οποίες αυτή οδηγεί. Το γραφικό περιβάλλον επικοινωνίας επιτρέπει στο χρήστη να επιλέγει κατάλληλα σε μια ιστοσελίδα μια έντονα φωτιζόμενη (highlighted) λέξη, η οποία μπορεί να τον οδηγήσει, είτε σε κάποιο άλλο τμήμα της ίδιας ιστοσελίδας, είτε σε κάποια άλλη ιστοσελίδα. Μέσω των έντονα φωτιζόμενων λέξεων πραγματοποιούνται οι μεταβάσεις (συνδέσεις) μεταξύ των τμημάτων των διάφορων ιστοσελίδων. Τα εργαλεία λογισμικού με τα οποία πραγματοποιείται η ανάγνωση των ιστοσελίδων αποτελούν το λογισμικό πλοήγησης – περιήγησης (browsers).

Ο όρος περιήγηση εκφράζει τη δυνατότητα του αναγνώστη να διαβάζει το κείμενο με τη σειρά που αυτός θεωρεί καλύτερη, κινούμενος μέσω των διαθέσιμων συνδέσμων μεταξύ των διαφόρων τμημάτων των ιστοσελίδων. Τα συστήματα υπερκειμένου που δεν περιλαμβάνουν μόνο κείμενο αλλά και άλλους τύπους πληροφορίας χαρακτηρίζονται ως υπερμέσα (hypermedia).

Κάθε ιστοσελίδα είναι προσπελάσιμη από οπουδήποτε στο Διαδίκτυο μέσω της διεύθυνσής της (Universal Resource Location, URL) που τη χαρακτηρίζει με τρόπο μοναδικό. Το web συνεπώς, αξιοποιώντας τις δυνατότητες των πολυμέσων, δημιουργεί έναν εύκολα προσβάσιμο παγκόσμιο πληροφοριακό χώρο. Τέλος, ένα ακόμη πλεονέκτημα του WWW είναι η ανεξάρτητη από οποιοδήποτε υπολογιστικό περιβάλλον γλώσσα ανάπτυξης των πληροφοριακών του αντικειμένων, η HTML (HyperText Markup Language) η οποία αναλύθηκε εκτενώς σε προηγούμενο κεφάλαιο της εργασίας.

5.2. Η μετάβαση από το web 1.0 στο web 2.0

Μέχρι στιγμής, έχουν περιγραφεί οι θεμελιώδεις έννοιες που είναι συνυφασμένες με τη δημιουργία, την ανάπτυξη και τη λειτουργία του Διαδικτύου. Επίσης, έχουν παρουσιαστεί τα βασικά θέματα που σχετίζονται με τον Παγκόσμιο Πληροφοριακό Ιστό. Στο σημείο αυτό λοιπόν, κρίνεται σκόπιμο να παρουσιαστούν τα χαρακτηριστικά του web 2.0, έτσι ώστε να

γίνει κατανοητό το πλαίσιο της τεχνολογίας αυτής, συγκριτικά με την προγενέστερη κατάσταση.

Το παραδοσιακό WEB, ή αλλιώς το web 1.0, αποτέλεσε το μέσο χάρη στο οποίο οι χρήστες απέκτησαν πρόσβαση σε πληθώρα δεδομένων. Σε αυτή τη βάση, θα μπορούσε κανείς να ισχυριστεί ότι το web 2.0 αποτελεί την εξέλιξη που έχει σαν κινητήριο δύναμη τους ίδιους τους χρήστες και διαμορφώνεται από τις ανάγκες τους (Καλτσογιάννης, 2007). Ανάγκες για αρτιότερη, ευκολότερη και πιο αποτελεσματική επικοινωνία, τροποποίηση των υπηρεσιών και λειτουργιών με βάση το πώς οι ίδιοι επιθυμούν να χρησιμοποιούν το Διαδίκτυο και να επηρεάζουν την ανθρώπινη δραστηριότητα. Ίσως είναι και η πρώτη φορά που οι χρήστες καθορίζουν τις εξελίξεις σε τόσο μεγάλο βαθμό.

Σε περιβάλλον web 1.0 ο χρήστης έχει στατικό ρόλο και χρησιμοποιεί τον Παγκόσμιο Πληροφοριακό Ιστό σαν μέσο πρόσβασης σε πληθώρα δεδομένων και σαν μέσο ανταλλαγής μηνυμάτων και αρχείων. Στο web 2.0 όμως η επικοινωνία λαμβάνει νέα διάσταση. Πλέον, ο χρήστης έχει ενεργό ρόλο στη χρήση και στη δημιουργία των πληροφοριών. Η έμφαση στο χρήστη και στις δυνατότητες που έχει πλέον επιβεβαιώνονται και από το γεγονός ότι η οντότητα αυτή επιλέχθηκε ως το «σημαντικότερο πρόσωπο της χρονιάς» για το έτος 2006 από το περιοδικό Time.

5.3. Χαρακτηριστικά web 2.0

Στον ακόλουθο πίνακα, παρουσιάζονται συνοπτικά οι κύριες διαφορές ως προς τρεις κεντρικές παραμέτρους (Μορφή και λειτουργία των ιστοσελίδων, ο ρόλος του χρήστη, αλλά και τεχνολογικά) που εντοπίζονται μεταξύ web 1.0 και web 2.0:

	Web 1.0	Web 2.0
Σελίδες :	Στατικές.	Δυναμικές.
Χρήστης :	Καμία παρέμβαση του χρήστη στο περιεχόμενο.	Ο χρήστης διαμορφώνει την πληροφορία.

Προγραμματιστικά :

- Framesets
- GIF κουμπιά
- HTML extensions
(πχ <blink>, <marquee>)
- Δυναμικές Γλώσσες
Προγραμματισμού.
- CSS styling
- AJAX
- Database Systems.

Αναλυτικότερα, τα βασικά χαρακτηριστικά που διέπουν το περιβάλλον web 2.0 σε σχέση με την υλοποίηση, αλλά και σχετικά με το χρήστη και τις παρεχόμενες λειτουργίες είναι τα εξής (Θεοδωράκης, «Επιχειρηματικότητα και Web 2.0»):

- Το Διαδίκτυο και όλες οι συσκευές που είναι συνδεδεμένες σε αυτό, αποτελούν μια παγκόσμια πλατφόρμα επαναχρησιμοποιούμενων υπηρεσιών και δεδομένων, τα οποία προέρχονται κυρίως από τους ίδιους τους χρήστες και στις περισσότερες περιπτώσεις διακινούνται ελεύθερα.
- Αρκεί ένας browser, ώστε να χρησιμοποιείται σαν διεπαφή με αυτή την πλατφόρμα, η οποία λειτουργεί ανεξαρτήτως συσκευής πρόσβασης (π.χ. Η/Υ, PDA, κινητό τηλέφωνο) και λειτουργικού συστήματος. Μοναδική προϋπόθεση, η ύπαρξη σύνδεσης στο Διαδίκτυο.
- Λογισμικό, περιεχόμενο και εφαρμογές ανοιχτού κώδικα (open source).
- Χρήση κυρίως ελαφριάς τεχνολογίας σε ό,τι αφορά τα πρωτόκολλα, τις γλώσσες προγραμματισμού, τις διεπαφές, ενώ διαπιστώνεται και μια τάση για απλότητα στον προγραμματιστικό σχεδιασμό τους.
- Πλούσια και διαδραστικά interfaces χρηστών (Rich Internet Applications-RIA), δυναμικό περιεχόμενο, ιστοσελίδες που ανανεώνουν μόνο όποιο περιεχόμενό τους αλλάζει (τεχνολογία Ajax).
- Συνεχής και άμεση ανανέωση των δεδομένων και του λογισμικού.

- Προώθηση του δημοκρατικού χαρακτήρα του Διαδικτύου, με τους χρήστες να έχουν τον πρωταγωνιστικό ρόλο.
- Υιοθέτηση της τάσης για αποκέντρωση των δεδομένων, υπηρεσιών και προτύπων.
- Δυνατότητα κατηγοριοποίησης του περιεχομένου από το χρήστη με σημασιολογικές έννοιες για ευκολότερη αναζήτηση της πληροφορίας.
- Δυνατότητα για ανοιχτή επικοινωνία, ανάδραση, διάχυση πληροφοριών, άμεση συγκέντρωση και εκμετάλλευση της γνώσης των χρηστών για διάφορα ζητήματα.
- Αμφίδρομη επικοινωνία του χρήστη με επιχειρήσεις ή οργανισμούς που μπορεί να έχει σαν αποτέλεσμα την επίδρασή του στη υιοθέτηση κατευθύνσεων και λήψη αποφάσεων.

5.4. Τεχνολογία web 2.0

Στη συνέχεια, παρουσιάζονται συνοπτικά οι σημαντικότερες τεχνολογίες που χρησιμοποιούνται από το web 2.0 και το διαφοροποιούν ως προς τον τρόπο λειτουργίας και παρουσίασης των ιστοσελίδων σε σχέση με το web 1.0 (Καλτσογιάννης, 2007):

- Πλούσιες και διαδραστικές διεπαφές χρήστη (Rich Internet Applications-RIA) που χρησιμοποιούν τεχνολογία Flash, Javascript και Ajax, χαρακτηριστικό που αντιπροσωπεύει την τάση του web 2.0 για όσο το δυνατόν καλύτερη εκμετάλλευση του δικτύου. Αντί να φορτώνεται ολόκληρη η σελίδα, ανανεώνονται μόνο τα δεδομένα που αλλάζουν όσο ο χρήστης βρίσκεται ή επανέρχεται σε αυτή (π.χ. Στο Gmail ο υπολογισμός του διαθέσιμου αποθηκευτικού χώρου ανανεώνεται σε πραγματικό χρόνο και από όλη τη σελίδα αλλάζει μόνο αυτός).
- Χρήση CSS (Cascading Style Sheets) για να διαχωρίζονται τα δεδομένα πληροφορίας από τα δεδομένα μορφοποίησης σε μια ιστοσελίδα. Αυτό, πέρα από την οικονομία στο εύρος ζώνης του δικτύου, προσφέρει και ευελιξία στον τρόπο παρουσίασης των δεδομένων, αφού ο χρήστης βλέπει τα δεδομένα ανάλογα με το CSS που ο ίδιος έχει (

π.χ. τα ίδια δεδομένα ανάλογα με το CSS μπορούν να παρουσιαστούν σε οθόνη υπολογιστή, κατευθείαν σε εκτυπωτή, σε μορφή ανάγνωσης για τυφλούς ή και να μετατραπούν σε φωνή και με χρήση κατάλληλου λογισμικού).

- Χρήση σημασιολογικών δεδομένων και microformats για να περιγράφεται η σημασία των δεδομένων που περιέχουν οι ιστοσελίδες (μεταδεδομένα). Έτσι, αυτά κατηγοριοποιούνται και η αναζήτησή τους γίνεται ευκολότερη και πιο αποδοτική.
- Χρήση RSS feeds ή και Atom.
- Χρήση ανοικτού λογισμικού (π.χ. Linux σαν λειτουργικό σύστημα, Apache σαν Web server, MySQL σαν βάση δεδομένων και PHP, Pearl, Python, σαν γλώσσες προγραμματισμού).
- Πρωτόκολλα δικτύου REST και SOAP που χρησιμοποιούν απλές HTTP εντολές (get, post, put, κλπ) για ανάκτηση δεδομένων από τους servers.
- Αρχιτεκτονικές SOA (Service Oriented Architecture) που επιτρέπουν το διαμοιρασμό και την επαναχρησιμοποίηση υπηρεσιών-εφαρμογών από διαφορετικά προγράμματα λογισμικού και SaaS (Software as a Service) όπου οι εφαρμογές είναι εγκατεστημένες σε κεντρικό server στο δίκτυο και οι χρήστες τις χρησιμοποιούν μέσω browser ανεξαρτήτως Η/Υ, τόπου, και χρονικής στιγμής.

5.5. Εφαρμογές web 2.0

Ο όρος «web 2.0» οφείλει την ύπαρξή του μεταξύ άλλων σε μία σειρά εφαρμογών, υπηρεσιών, εργαλείων και λειτουργιών που διέπονται από συγκεκριμένα χαρακτηριστικά. Πρόκειται για καινοτομίες που αναζητούσαν οι χρήστες σχετικά με τη δυνατότητα διαχείρισης και επεξεργασίας των πληροφοριών που διατίθενται μέσω του Παγκόσμιου Πληροφοριακού Ιστού. Έτσι, η αλληλουχία ή η ροή των γεγονότων ήταν η εξής: υπάρχει μία συγκεκριμένη ανάγκη, εντοπίζεται και περιγράφεται αυτή η ανάγκη, υλοποιούνται νέες εφαρμογές και παρέχονται νέες υπηρεσίες που καλύπτουν αυτή την ανάγκη.

Web 1.0

Web 2.0

Εφαρμογές /**Υπηρεσίες :**

Ηλεκτρονικό Ταχυδρομείο
(email)

Ηλεκτρονικό Ταχυδρομείο
(email)

Μεταφορά αρχείων (file
transfer, ftp)

Μεταφορά αρχείων (file
transfer, ftp)

Ομάδες ειδήσεων (newsgroup)

Ομάδες ειδήσεων (newsgroup)

Usenet

Ανοικτή συνομιλία (internet
relay chat, IRC)

Ανοικτή συνομιλία (internet
relay chat, IRC)

Τηλεδιάσκεψη

Τηλε-ιατρική

Τηλε-εκπαίδευση

Ηλεκτρονικό εμπόριο

Social Media

Wikis

Blogs

RSS

Tagging

Youtube

Twitter

...

Σήμερα λοιπόν για παράδειγμα οι περισσότεροι χρήστες του Διαδικτύου διατηρούν έναν λογαριασμό στο Facebook που επισκέπτονται και ενημερώνουν σχεδόν σε καθημερινή βάση. Οι «οπαδοί» του twitter είναι και αυτοί μία άλλη κατηγορία χρηστών με ιδιαίτερα χαρακτηριστικά. Πέραν αυτών των ενδεικτικών εφαρμογών, υλοποιήθηκαν διάφορες

εφαρμογές και υπηρεσίες που έτυχαν ευρείας αποδοχής και διάδοσης. Οι εφαρμογές λοιπόν είναι αυτές που συνθέτουν το πλαίσιο του web 2.0 και τη μετάβαση σε μία «νέα εποχή».

Μια τάση που παρατηρείται στις εφαρμογές αυτές είναι η προσαρμογή των προγραμμάτων και υπηρεσιών στις προτιμήσεις των χρηστών και η βελτίωσή τους ανάλογα με τον αριθμό των ατόμων και τη συχνότητα που τα χρησιμοποιούν. Τέτοια παραδείγματα είναι τα προγράμματα διαμοίρασης αρχείων μεταξύ των χρηστών (π.χ. BitTorrent), αρχιτεκτονικής P2P (Peer to Peer networks), τα οποία γίνονται αποτελεσματικότερα όσο τα χρησιμοποιούν περισσότεροι χρήστες.

Επιπλέον, ιστοσελίδες αξιολόγησης προϊόντων και υπηρεσιών οφείλουν τη δημοτικότητά τους στο πλήθος των δημοσιευμένων απόψεων των χρηστών, ενώ προγράμματα anti spam βασίζονται και στη γνώμη των χρηστών για το τι είναι spam και τι όχι.

Ιστοσελίδες οι οποίες μιμούνται τη λειτουργία ενός λειτουργικού συστήματος και παρέχουν τα βασικά χαρακτηριστικά και εφαρμογές του. Άλλες, προσφέρουν χωρίς χρέωση, εναλλακτικά προγράμματα παραδοσιακών εμπορικών εφαρμογών π.χ. της σουίτας προγραμμάτων Office (Google Docs).

Ιστοσελίδες που ο καθένας μπορεί να “ανεβάσει” ό,τι video, φωτογραφίες ή τραγούδια επιθυμεί (YouTube.com, myspace.com) και να κάνει γνωριμίες μέσα από αυτές. Παιχνίδια, όπως το Second Life, που φέρνουν τα όρια της πραγματικότητας και της εικονικής πραγματικότητας πολύ κοντά, αφού ο χρήστης μπορεί να πραγματοποιήσει online πολλές από τις δραστηριότητες της καθημερινής του ζωής.

5.6. Ηλεκτρονικό Εμπόριο (E-Commerce)

Η ελεύθερη, ανταγωνιστική αγορά θα μπορούσε να αποτελείται από έναν παγκόσμιο ηλεκτρονικό χώρο ανοικτό για κάθε επιχειρηματική δραστηριότητα. Το γεγονός αυτό συνεπάγεται πρόσβαση σε μεγαλύτερες αγορές, αύξηση της επιχειρηματικής κινητικότητας με σκοπό τον εντοπισμό των καταλληλότερων συνθηκών δραστηριοποίησης (ευνοϊκό φορολογικό σύστημα, χαμηλό κόστος εργατικού δυναμικού, κτλ), καθώς επίσης και ευελιξία για απασχόληση εργαζομένων, παραγωγή προϊόντων και παροχή υπηρεσιών οπουδήποτε, χρησιμοποιώντας μία παγκόσμια δύναμη εργασίας που εργάζεται από απόσταση.

Ενώ τα γεωγραφικά όρια της αγοράς καταρρίπτονται, οι παγκόσμιες κοινότητες με βάση συγκεκριμένα ενδιαφέροντα εξαπλώνονται. Η εμφάνιση του Διαδικτύου και των extranets είχε σαν αποτέλεσμα μία οικονομική και ευέλικτη υποδομή που αποτελεί κινητήριο δύναμη για το παγκόσμιο εμπόριο. Η κεντρική ιδέα και το βασικό πλεονέκτημα του Ηλεκτρονικού Εμπορίου είναι η δυνατότητα να κάνει κανείς δουλειές ανά πάσα στιγμή, από οπουδήποτε, οι απαιτούμενες εργασίες να γίνονται γρήγορα και με εύλογο κόστος.

Στο σημείο αυτό είναι σημαντικό να διευκρινιστεί ότι η έννοια του Ηλεκτρονικού Εμπορίου είναι ευρύτερη και κατά συνέπεια δεν περιλαμβάνει μόνο ένα δικτυακό τόπο μέσου του οποίου ένας επιχειρηματίας πουλάει τα προϊόντα του. Αυτό είναι απλά ένα υποσύνολο. Αντίθετα, περιλαμβάνει τραπεζικές συναλλαγές από το σπίτι, αγορές σε online καταστήματα και εμπορικά κέντρα, αγορά μετοχών, εύρεση εργασίας, διεξαγωγή μίας δημοπρασίας και ηλεκτρονική συνεργασία σε ερευνητικά και αναπτυξιακά έργα.

Αντιλαμβάνεται λοιπόν κανείς ότι η διαδικασία ανάπτυξης ενός διαδικτυακού τόπου ποικίλει με την έννοια ότι μπορεί να αναφέρεται στην ανάπτυξη μίας στατικής ιστοσελίδας, είτε μίας δυναμικής που θα μπορούσε να είναι ένα ηλεκτρονικό κατάστημα. Έπειτα και αυτό θα μπορούσε να εμπλουτιστεί με επιμέρους διαδραστικές υπηρεσίες (σχολιασμός, επικοινωνία, κτλ), είτε να παρέχει εξατομικευμένες υπηρεσίες όπως για παράδειγμα το amazon. Τέλος, είναι σημαντικό να γίνει μνεία στο γεγονός ότι ένας διαδικτυακός τόπος θα μπορούσε να εξυπηρετεί όχι μόνο άτομα μεμονωμένα, ούτε μόνο επιχειρήσεις, αλλά και φορείς (οργανισμούς με την ευρύτερη έννοια) και κράτη (κυβερνητικές ιστοσελίδες).

5.6.1 Οφέλη Ανάπτυξης Διαδικτυακών Τόπων

Τα οφέλη από την ανάπτυξη ενός διαδικτυακού τόπου ή απλούστερα μίας ιστοσελίδας, είτε αυτή είναι στατική, είτε δυναμική είναι πολλαπλά. Είναι ωστόσο σαφές ότι η περίπτωση μία δυναμικής ιστοσελίδας μπορεί να καλύψει την ανάγκη ανάπτυξης ενός ηλεκτρονικού καταστήματος, χαρακτηριστικό που θα κάλυπτε τις ουσιαστικές ανάγκες μίας πραγματικής επιχείρησης η οποία αντιλαμβάνεται το ανταγωνιστικό πλεονέκτημα που απορρέει από την αξιοποίηση της τεχνολογίας και ειδικότερα του Διαδικτύου.

Στα πλαίσια αυτά, παρακάτω παρατίθενται ορισμένα βασικά οφέλη που εξασφαλίζονται από την ανάπτυξη ενός διαδικτυακού τόπου μέσω του οποίου πραγματοποιούνται εμπορικές συναλλαγές (Turban et al.):

- Επέκταση της θέσης αγοράς σε εθνικές και διεθνείς αγορές.

- Μείωση του κόστους δημιουργίας, επεξεργασίας, διανομής, αποθήκευσης και επαναφοράς πληροφοριών που είναι καταγεγραμμένες σε χαρτί.
- Δυνατότητα δημιουργίας άκρως εξειδικευμένων επιχειρήσεων.
- Μείωση επιπέδων αποθεμάτων και εξόδων.
- Καλύτερη εξυπηρέτηση πελατών.
- Εντοπισμός νέων επιχειρηματικών εταίρων.
- Ανάπτυξη απλοποιημένων και δομημένων επιχειρηματικών διαδικασιών.
- Μείωση χρόνου παραγωγής και παράδοσης αναπτύσσοντας συνέργιες.
- Εξάλειψη χαρτιών (paperwork).
- Ταχύτερη πρόσβαση σε πληροφορίες.
- Αυξημένη ευελιξία.
- Μείωση τηλεπικοινωνιακού κόστους, διότι το Διαδίκτυο ως επικοινωνιακό μέσο είναι πιο οικονομικό συγκριτικά με τις υπόλοιπες εναλλακτικές (πχ, Τηλέφωνο, VPN, κτλ).

ΣΥΝΟΨΗ - ΣΥΜΠΕΡΑΣΜΑΤΑ

Οι επιχειρήσεις που βασίζονται στη χρήση της νέας τεχνολογίας διακρίνονται από μια τάση συνεχούς εξέλιξης σε επίπεδο τεχνολογικών εφαρμογών και είναι ανοιχτές στις αλλαγές. Οι περισσότερες επιχειρήσεις του τομέα επικοινωνίας και όχι μόνο, χρησιμοποιούν ηλεκτρονικούς υπολογιστές. Είναι γνωστό πλέον ότι όσο ευρύτερη είναι η χρήση της νέας τεχνολογίας σε έναν χώρο εργασίας τόσες περισσότερες πιθανότητες επιτυχίας και αναγνώρισης υπάρχουν για την εκάστοτε επιχείρηση.

Στην παρούσα εργασία στόχο είχαμε να αναδείξουμε τις τεχνολογίες ανάπτυξης διαδικτυακών τόπων. Στα πλαίσια αυτά πραγματοποιήθηκε μία θεωρητική επισκόπηση του ερευνητικού αντικειμένου και παρουσιάστηκε η αρχιτεκτονική, η δομή, τα πρωτόκολλα, οι υπηρεσίες αλλά και οι εφαρμογές του Διαδικτύου.

Πιο αναλυτικά, αρχικά έγινε μια σύντομη ιστορική αναδρομή του Διαδικτύου, στη συνέχεια έγινε λόγος για την ψηφιακή επικοινωνία και τους λόγους για τους οποίους χαρακτηρίζεται αδήριτη η Πληροφορική σαν επιστήμη και η εξέλιξή της, η οποία συνδέεται άρρηκτα με την εν γένει εξέλιξη της κοινωνίας.

Έπειτα, τέθηκαν οι βάσεις του υπολογιστικού περιβάλλοντος, δηλαδή σημειώνεται εκτενής αναφορά στα βασικά στοιχεία του Διαδικτύου, ήτοι τις αρχιτεκτονικές, τα πρωτόκολλα και τις βασικές υπηρεσίες που προσφέρει το Διαδίκτυο στους χρήστες του. Στη συνέχεια παρουσιάστηκαν επιμέρους συναφείς γλώσσες προγραμματισμού, όπως η html, η php, αλλά και η νέα γλώσσα, η οποία είναι ιδιαίτερα δημοφιλής, η XHTML5.

Στη συνέχεια, έγινε ιδιαίτερη αναφορά στις βάσεις δεδομένων, κάνοντας λόγο για τα πλεονεκτήματα και τις ανάγκες τους, ενώ δεν παραλείψαμε να αναφέρουμε συστήματα διαχείρισης περιεχομένου, που είναι ιδιαίτερα χρήσιμα, καθώς επιτρέπουν την online τροποποίηση του περιεχομένου μιας ιστοσελίδας.

Είναι κοινός τόπος, λοιπόν, ότι αργά ή γρήγορα, οι περισσότερες επιχειρήσεις επενδύουν στην Πληροφορική. Η ταχύτητα και η ακρίβεια των λογιστικών πακέτων, καθώς και οι διαχειριστικές πληροφορίες που παρέχουν, τα καθιστούν σημαντική καινοτομία. Οι επιχειρήσεις που έχουν ενσωματώσει στη στρατηγική τους διαδικασίες ορθολογικής διαχείρισης της τεχνολογίας, απολαμβάνουν σημαντικά οφέλη ως προς την καλύτερη οργάνωση των λειτουργιών τους, την μείωση του λειτουργικού τους κόστους καθώς και την συμβολή τους στην μείωση του περιβαλλοντικού τους αποτυπώματος. Αυτό που καλείται να

επιτύχει η Πληροφορική είναι να ελαχιστοποιήσει το κόστος και τον κόπο, ενώ συγχρόνως να επιφέρει σημαντική πρόοδο ως προς την αποτελεσματικότητα της επιχείρησης. Το κλειδί για την επιτυχία είναι η σωστή προσέγγιση, καθώς επίσης και η αξιοποίηση της υπάρχουσας τεχνολογίας.

Ολοκληρώνοντας την παρούσα εργασία εκτιμάται ότι ο στόχος που αρχικά τέθηκε έχει επιτευχθεί, αποκομίζοντας πολύτιμη γνώση καθόλη τη διάρκεια της εκπόνησης του έργου, αλλά και διαμορφώνοντας έναν συνοπτικό οδηγό για έναν αναγνώστη που ενδιαφέρεται να ενημερωθεί σχετικά με θεμελιώδη θέματα που άπτονται του Διαδικτύου, ως προς την τεχνολογική υποδομή, αλλά και τις εφαρμογές του.

Σχετικά με τα συμπεράσματα που απορρέουν, αυτά εστιάζουν στο γεγονός ότι στο περιβάλλον του Διαδικτύου συντελούνται σημαντικές αλλαγές. Ο αρχικός σκοπός που θα μπορούσε να περιγραφεί ως «στατική παρουσίαση πληροφοριών» εξελίσσεται σε κάτι πολύ πιο δυναμικό, πολύ πιο ενδιαφέρον και μάλιστα με κεντρική ιδέα τη συμμετοχή του ίδιου του χρήστη στη δημιουργία των πληροφοριών. Στα πλαίσια αυτά, οι έννοιες όπως «διαδραστικότητα», «δυναμικό περιεχόμενο», «συνεργασία», «συνεισφορά», «κοινωνική δικτύωση», έρχονται στο προσκήνιο. Ορισμένοι μάλιστα ισχυρίζονται ότι πρόκειται για μία τεχνολογική και συγχρόνως κοινωνική επανάσταση που είναι σε εξέλιξη (Καλτσογιάννης, 2007)!

Οι αλλαγές αυτές, θα μπορούσαν να αποδοθούν μέσω της τεχνολογίας και των εφαρμογών web 2.0, για τις οποίες έγινε αναφορά στο τελευταίο κεφάλαιο της εργασίας και κρίνεται σκόπιμο να αποτελέσει έναυσμα για περαιτέρω μελέτη, ώστε να εξεταστεί η μετάβαση σε περιβάλλον web 3.0.. Δεν πρόκειται για μία νέα έκδοση του Παγκόσμιου Πληροφοριακού Ιστού, ούτε για κάποιο νέο πρωτόκολλο επικοινωνίας διαφορετικό από το TCP/IP. Πρόκειται όμως για αλλαγές του τρόπου χρήσης των τεχνολογιών, αλλαγές στην αρχιτεκτονική των εφαρμογών και κυρίως αλλαγές στον «προσανατολισμό» των υπηρεσιών. Έτσι, η κεντρική ιδέα είναι η χρήση του Διαδικτύου ως μία ενιαία, κοινή πλατφόρμα, μέσω της οποίας διατίθενται πολλαπλές εφαρμογές και υπηρεσίες μέσω των οποίων καθίσταται δυνατή η επικοινωνία, η συνεργασία και η αλληλεπίδραση μεταξύ των χρηστών σε κατάσταση πραγματικού χρόνου (on-line).

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Allen, D.W. και Johnson, S. (1998) Μαθαίνοντας το Internet, Αθήνα: Γκιούρδας.
- Blissmer, R. H (1996) Introducing Computers, Wiley.
- Carron, H. L. (1999) Εφαρμογές πληροφορικής computers: εργαλεία για μια εποχή πληροφοριών, Αθήνα: Ίων.
- Dologite, D. G. (1990) Εισαγωγή στη χρήση των υπολογιστών, Αθήνα: Γκιούρδας.
- ECDL 4 (7 σε 1) MS Office 2003 γρήγορα και απλά (2006), Αθήνα: Κλειδάριθμος.
- Efraim Turban, Jae Lee, David King, H. Michael Chung, Ηλεκτρονικό Εμπόριο, Αρχές-Εξελίξεις-Στρατηγική από τη σκοπιά του manager, Εκδόσεις Μ. Γκιούρδας.
- Glossbrenner, A. & E. (1999) Search engines for the World Wide Web, Berkeley, Calif, Peachpit press.
- McBride, P. K. (1999) Internet, Αθήνα: Δίαυλος.
- Microsoft Office System έκδοση 2003 βήμα-βήμα (2004), Αθήνα: Κλειδάριθμος.
- Norton, P. (1995) Εισαγωγή στους υπολογιστές, Αθήνα: Τζιόλας.
- Αδαμόπουλος Δ. (2002). Τηλεματική, Διαδίκτυα και Κοινωνία, «Τηλεματική».
- Ελληνικά Microsoft Windows xp βήμα-βήμα, Online training solutions (2001), Αθήνα: Κλειδάριθμος.
- Ελληνικό Microsoft Office System (2004), Αθήνα: Κλειδάριθμος.
- Ελληνικό Microsoft Office Word 2003 βήμα βήμα (2004), Αθήνα: Κλειδάριθμος.
- Ελληνικός Internet Explorer 6, βήμα-βήμα (2002), Αθήνα: Κλειδάριθμος.
- Ζαφείρη, Κ. Και Θεοδώρου, Σ. (2005) Μηχανές Αναζήτησης και Directories, Αυτοέκδοση.
- Θεοδωράκης Δ., JADE Hellas, Καινοτομείν-Επιχειρείν, Επιχειρηματικότητα και Web 2.0.
- Καλαφατούδης, Σ., Κοίλιας, Χ. και Μπανδήλα, Ε (2004) Εισαγωγή στην Πληροφορική και χρήση υπολογιστή, Αθήνα: Εκδόσεις Νέων Τεχνολογιών.
- Καλτσογιάννης Α.(2007) Παρατηρητήριο για την Κοινωνία της Πληροφορίας, web 2.0: Χαρακτηριστικά και επίδρασή του σε επιχειρήσεις, κεντρική διοίκηση και χρήστες.
- Καρολίδης, Ξ. (2005) Microsoft Office Word 2003 Θεωρία και Πράξη, Αθήνα: Αβακας.
- Παπαδάκης, και Χατζηπέρης, Ν. (2004) Εισαγωγή στην Πληροφορική και τη χρήση υπολογιστή, Αθήνα: Μεταίχμιο.
- Πολλάλης, Γ. Α.. Γιαννακόπουλος, Δ. και Παπουτσή, Ι. (1999) Εισαγωγή στους ηλεκτρονικούς υπολογιστές. Αθήνα: Ίων.
- Φούσκας Γ. (2002). Βασικά Ζητήματα Δικτύων Η/Υ, «Δίκτυα Υπολογιστών Ι».

Ιστοσελίδες

www.webstandards.org/learn/articles/askw3c/oct2003/

www.fme.aegean.gr/sites/.../xsd_askhsh_level1.pdf

www.tplanet.eu/showtutorial.php?id

www.nefeli.lib.teicrete.gr/.../Chatzipaulis2010.pdf

www.jcph.wikispaces.com/aboutlanguages

www.ifestos.teilar.gr/index.php?option=com

www.webvistas.org/topic/43-html5/