

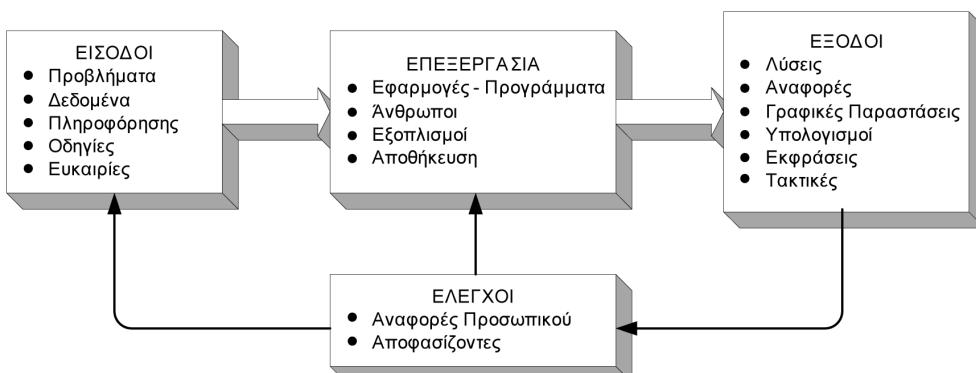
κεφάλαιο 2

Πληροφοριακά Συστήματα

2.1 Εισαγωγή

Ως σύστημα (information system – σχήμα 2.1) μπορούμε να θεωρήσουμε γενικά, ένα σύνολο από μέσα συλλογής, αποθήκευσης, επεξεργασίας, ανάλυσης και διάχυσης της πληροφόρησης με στόχο την επίτευξη ενός συγκεκριμένου σκοπού. Αν και με το ορισμό αυτό καλύπτουμε τα πάσης φύσεως συστήματα πληροφόρησης, εν τούτοις στην πράξη έχει επικρατήσει ο όρος αυτός να αναφέρεται στα πληροφοριακά συστήματα που βασίζονται στη χρήση ηλεκτρονικού υπολογιστή (computer-based information systems).

Έτσι, ως Πληροφοριακά Συστήματα που βασίζονται στη χρήση ηλεκτρονικού υπολογιστή ορίζουμε τα πληροφοριακά συστήματα που χρησιμοποιούν την τεχνολογία των ηλεκτρονικών υπολογιστών (hardware and software), για την υλοποίηση των εργασιών και των στόχων των χρηστών τους.



Σχήμα 2.1. Πληροφοριακό σύστημα (πηγή: Turban et al., 1996)

Ο όρος τεχνολογία της πληροφορίας (information technology), αναφέρεται συνήθως στην τεχνολογική πλευρά των πληροφοριακών συστημάτων (hardware, software, databases, cdrom, cdrec, printers, ...), ενώ αρκετές φορές χρησιμοποιείται και ως ισοδύναμος των πληροφοριακών συστημάτων ή της συνένωσης πληροφοριακών συστημάτων ενός οργανισμού ή επιχείρησης.

2.2 Ο στρατηγικός ρόλος των Πληροφοριακών Συστημάτων

Οι περισσότεροι επιχειρηματίες και διευθυντές θεωρούσαν ότι τα πληροφοριακά συστήματα είναι απαραίτητα για την υποβοήθηση των λειτουργιών της επιχείρησης αλλά αγνοούν τη χρήση τους σε κύριες περιοχές όπως: υπολογισμό και πρόβλεψη μεριδίων αγοράς, ανάπτυξη νέων προϊόντων, αξιολόγηση και επιλογή νέων επιχειρηματικών πρωτοβουλιών, κα. Η άποψη αυτή με την πάροδο του χρόνου μεταβάλλεται και διαμορφώνεται η άποψη ότι ο ρόλος των πληροφοριακών συστημάτων είναι κυρίως για να προσδίδει ανταγωνιστικά πλεονεκτήματα στις επιχειρήσεις. Αυτό δεν δημιουργεί και την βεβαιότητα ότι όποια επιχείρηση αναπτύσσει και χρησιμοποιεί πληροφοριακά συστήματα έχει ή priori την προσδοκώμενη επιτυχία υλοποίησης των στόχων της. Η κύρια αιτία αποτυχίας των, έχει να κάνει με το ότι τα πληροφοριακά συστήματα που αναπτύσσονται προσαρμόζονται στην υφιστάμενη απηρχαιωμένη οργανωτική δομή της επιχείρησης και ασχολούνται με την επίλυση τρεχόντων προβλημάτων. Η επίτευξη ανταγωνιστικών πλεονεκτημάτων με τη βοήθεια των ΠΣ έχει να κάνει με την αναδιοργάνωση των λειτουργιών και διαδικασιών μιας επιχείρησης καθώς και με την αύξηση της χρήσης εξειδικευμένων πληροφοριακών συστημάτων (ΣΥΑ-DSS, ΔΠΣ-EIS, ΕΣ-ES) από τα στελέχη – αποφασίζοντες υψηλού επιπέδου διοίκησης για την επίλυση εξειδικευμένων προβλημάτων.

Στο σημερινό έντονα ανταγωνιστικό περιβάλλον, όπου όλα τα δεδομένα αλλάζουν ταχύτατα και επικρατεί μεγάλη αβεβαιότητα, μία επιχείρηση προκειμένου να είναι σε θέση να ανταπεξέλθει στον ανταγωνισμό θα πρέπει να αποδίδει σημαντικό ρόλο στα πληροφοριακά συστήματα και να τα χρησιμοποιεί για να:

- αυξήσουν την παραγωγικότητα (μείωση του κόστους, αύξηση της αποτελεσματικότητας, ...),
- βελτιώσουν την ποιότητα των προσφερόμενων προϊόντων και υπηρεσιών,
- δημιουργήσουν ανταγωνιστικά πλεονεκτήματα,
- διαμορφώσουν τη στρατηγική της εταιρείας,

κεφ. 2 Πληροφοριακά Συστήματα

59

- αναδιοργανώσουν και επανασχεδιάσουν τις επιχειρησιακές τους διαδικασίες,
- υποστηρίζει την βελτίωση της λήψης αποφάσεων,
- ανταποκρίνεται ταχύτερα στις απαιτήσεις των πελατών και στις αλλαγές της επιχείρησης ή του περιβάλλοντός της,
- έχει πρόσβαση σε σημαντικές και χρήσιμες πληροφορίες (πχ. μέσω διαδικτύου, εξωτερικών βάσεων δεδομένων, ...),
- διαχειρίζεται καλύτερα τεράστιους όγκους δεδομένων-πληροφοριών,
- βελτίωσει τη δημιουργικότητα και να προωθεί τις καινοτομίες, κ.α.

Τα Πληροφοριακά Συστήματα (ΠΣ) μίας επιχείρησης-οργανισμού, τα οποία συνεισφέρουν στην επιχειρησιακή της στρατηγική, καλούνται Στρατηγικά Πληροφοριακά Συστήματα (ΣΠΣ), ορίζονται δε ως τα ΠΣ, τα οποία υποστηρίζουν ή/και καθορίζουν τη στρατηγική των επιχειρησιακών λειτουργιών (Wiseman, 1985; 1988). Το κύριο κριτήριο αναγνώρισης ενός ΠΣ ως στρατηγικού, είναι ο βαθμός στον οποίο μπορεί να συμβάλλει στην αλλαγή του τρόπου λειτουργίας της επιχείρησης. Η κύρια σκοπιμότητα ανάπτυξης τέτοιων ΠΣ, είναι η βελτίωση της παραγωγικότητας, της ποιότητας και της ανταγωνιστικής θέσης μιας επιχείρησης-οργανισμού.

Αρχικά η χρήση των ΣΠΣ ήταν μόνο εξωτερική και απευθυνόταν στην αγορά της εταιρείας στοχεύοντας στον άμεσο ανταγωνισμό με τις αντίστοιχες εταιρείες του κλάδου, παρέχοντας για παράδειγμα νέες υπηρεσίες στους πελάτες ή/και στους προμηθευτές. Στη συνέχεια όμως, τα ΣΠΣ άρχισαν να χρησιμοποιούνται και για εσωτερική χρήση, υποστηρίζοντας προσπάθειες αναδιοργάνωσης και διαχείρισης των λειτουργιών των επιχειρήσεων, βελτίωσης της παραγωγικότητας των εργαζομένων και των συνθηκών συνεργασίας και επικοινωνίας. Σύμφωνα με τους Sass and Keeffe (1988), τα ΣΠΣ μπορούν να συνεισφέρουν στην ανάπτυξη ανταγωνιστικού πλεονεκτήματος με τους ακόλουθους τρόπους:

- Ανάπτυξη εμποδίων για την εισαγωγή ανταγωνιστών στην αγορά.
- Ολοκληρωτική αλλαγή της βάσης του ανταγωνισμού, με την παροχή νέων υπηρεσιών, προϊόντων ή πληροφοριών που δεν παρέχονται από τους ανταγωνιστές.
- Αύξηση του κόστους αλλαγής πελατών (προμηθευτών) και/ή λειτουργική εξάρτηση, με σκοπό να αποθαρρυνθούν οι πελάτες από το να αλλάξουν προμηθευτές (πχ. με την παροχή ειδικών υπηρεσιών προς τον πελάτη, οι οποίες δύσκολα μπορούν να αντιγραφούν από τους ανταγωνιστές).

- Επανασχεδίαση των επιχειρησιακών λειτουργιών της εκμεταλλευόμενοι τις δυνατότητες που μας παρέχουν τα ΠΣ.
- Επίσπευση επιχειρησιακών λειτουργιών, όπως για παράδειγμα η τιμολόγηση, που επιτρέπει στην εταιρεία να επιλέξει μία βέλτιστη στρατηγική τιμολόγησης για τα προϊόντα και τις υπηρεσίες που διαθέτει σε διαφορετικούς πελάτες.
- Υποστήριξη σημαντικών εργασιών με την ανάπτυξη εξειδικευμένων ΠΣ. Αυτό έχει ως αποτέλεσμα τη μείωση του κόστους και του χρόνου διανομής του τελικού προϊόντος-υπηρεσιών στον πελάτη.

2.3 Ολοκλήρωση Πληροφοριακών Συστημάτων

Ο όρος τεχνολογία λογισμικού (software engineering) εμφανίζεται αρχικά σε δύο συνέδρια του NATO το 1968 και 1969 (Naur and Randell, 1969; Randell and Buxton, 1970), στην προσπάθεια να οργανωθεί με συλλογικό τρόπο η παραγωγή προϊόντων λογισμικού. Ο Constantine υπήρξε ένας από τους πρωτοπόρους της δομημένης σχεδίασης ο οποίος επινόησε μια ιεραρχική μέθοδο διάσπασης. Πάνω στις πρώτες ιδέες του Constantine βασίσθηκαν οι Yourdon (1972) και οι DeMarco (1978) καθώς και οι Gane and Sarson (1979) για να γράψουν τα πρώτα βιβλία τους για τη δομημένη ανάλυση. Μέχρι τότε οι υπάρχουσες μέθοδοι κάλυπταν μόνο μερικές από τις φάσεις του κύκλου ζωής του λογισμικού. Η μέθοδος της δομημένης ανάλυσης (DeMarco, 1978), αρχικά εφαρμόσθηκε στην ανάλυση και σχεδίαση συστημάτων αλλά γρήγορα επεκτάθηκε και κάλυψε όλες τις φάσεις του κύκλου ζωής. Από τότε έχουν δημιουργηθεί νέες τεχνικές οι οποίες όμως βασίζονται στα κύρια σημεία τους στις παραπάνω αρχικές ιδέες. Γενικά δύο είδη ολοκλήρωσης συστημάτων υπάρχουν, η τεχνική (technical integration) και η λειτουργική (functional integration) (Hale et al., 1989; Brooks, 1989).

Η τεχνική ολοκλήρωση υποδιαιρείται στα ακόλουθα τρία είδη:

1. Κάθετη ολοκλήρωση πληροφοριακών συστημάτων: Κατά την αρχιτεκτονική σχεδίαση των διαφόρων επιμέρους συστημάτων λαμβάνεται πρόνοια να υπάρχει δυνατότητα μεταφοράς δεδομένων μεταξύ των διαφόρων ανεξάρτητων εφαρμογών του συστήματος. Ως εκ τούτου σημαντικότατο ρόλο παίζει η καλοσχεδιασμένη δυνατότητα μεταφοράς δεδομένων μέσω σταθερών παραμέτρων ενδοεπικοινωνίας (Hale et al., 1989).
2. Οριζόντια ολοκλήρωση πληροφοριακών συστημάτων: Κατά την αρχιτεκτονική σχεδίαση των συστημάτων αυτών λαμβάνεται πρόνοια να υπάρχει δυνατότητα

κεφ. 2 Πληροφοριακά Συστήματα**61**

συνεργασίας των διάφορων λειτουργικών υποσυστημάτων του ΠΣ, ούτως ώστε να παρέχεται η δυνατότητα μεταφοράς δεδομένων από το ένα υποσύστημα σε ένα άλλο. Τα συστήματα αυτά μπορούν να διαχωριστούν στις ακόλουθες τέσσερις υποκατηγορίες, ανάλογα με το τρόπο της παρεχόμενης διασύνδεσης των υποσυστημάτων τους, όσον αφορά τη μεταφορά δεδομένων (Brooks, 1989):

- αυτόνομα (standalone),
- απλής διασύνδεσης (simple linked),
- περίπλοκης διασύνδεσης (sophisticated linked), και
- ολοκληρωμένα (integrated).

3. Ολοκλήρωση ανά έργο (project integration): Αναφέρεται στην ανάπτυξη μεγάλων και πολύπλοκων συστημάτων, που περιλαμβάνουν βάσεις δεδομένων και δυνατότητες τηλεπικοινωνίας και αφορούν την υλοποίηση ενός συγκεκριμένου έργου (project).

Με στόχο τη βελτίωση της αποτελεσματικότητας γίνεται ολική επανασχεδίαση αρκετών λειτουργικών διαδικασιών, γεγονός που επιφέρει πρόσθετα ανταγωνιστικά πλεονεκτήματα στην επιχείρηση (Benjamins and Scott-Morton, 1988; Clemons and Row, 1991). Δύο κύριοι τρόποι λειτουργικής ολοκλήρωσης περιγράφονται στη συνέχεια:

1. Κάθετη ολοκλήρωση των λειτουργιών: Γίνεται επανασχεδίαση των εργασιών και ριζική μεταβολή των παραδοσιακών τρόπων καταμερισμού της ροής των εργασιών στις διάφορες ομάδες χρηστών ενός συστήματος που υποστηρίζει συγκεκριμένες λειτουργίες μιας επιχείρησης (πχ. υποστηρίζει τις λειτουργίες μάρκετινγκ).
2. Οριζόντια ολοκλήρωση των λειτουργιών: Με τον όρο αυτό εννοείται η χρήση των συστημάτων αυτών για να μεταβάλλουν τις δομικές διαφορές (Clemons and Row, 1991) μεταξύ των στρατηγικών μέσων μιας εταιρείας ούτως, ώστε να επιτύχουν τη βελτίωση της χρησιμοποίησης των μέσων με αποτέλεσμα τη μείωση του κόστους. Αφορά την υποστήριξη οριζόντιων δράσεων μιας επιχείρησης που διατρέχουν οριζοντιώς τις διάφορες λειτουργίες της επιχείρησης.

2.4 Κύκλος ζωής συστημάτων

Τα προϊόντα λογισμικού έχουν και αυτά ένα κύκλο ζωής που ξεκινά από τη σύλληψη μιας νέας ιδέας για την ανάπτυξη ενός συγκεκριμένου προϊόντος και φθάνει μέχρι τη

χρονική στιγμή της απόσυρσής του από την αγορά ή από το σταμάτημα της χρήσης του, περιλαμβάνει δε όλες εκείνες τις δραστηρότητες που είναι απαραίτητες για την ανάπτυξη, τη λειτουργία και την συντήρησή του.

Ένα μοντέλο κύκλου ζωής περιγράφει τις φάσεις, τις εργασίες που γίνονται και τα προϊόντα που παράγονται σε κάθε φάση καθώς και τη σειρά διαδοχής των. Με στόχο την περιγραφή των διαφόρων σταδίων του κύκλου ζωής του λογισμικού, έχουν αναπτυχθεί διάφορα μοντέλα ανάπτυξής του (Yourdon, 1989; Norris and Rigby, 1992; Γιακουμάκης, 1994), όπως:

- Μοντέλο του καταρράκτη - waterfall model (Royce, 1970; Boehm, 1984; IEEE, 1983; Rook, 1986).
- Σπειροειδές μοντέλο - spiral model (Boehm, 1988).
- Μοντέλο σταδιακής εκλέπτυνσης και επαναληπτικής προσαύξησης - stepwise refinement and iterative enhancement model (Wirth, 1971).
- Μοντέλο προτυποποίησης - prototyping model (Agresti, 1986; Carey, 1990; McLeod, 1993).
- Λειτουργικό μοντέλο - operational model (Zave, 1984).
- Μοντέλο αυτόματου προγραμματισμού - automatic programming model (Agresti, 1986; Stahl, 1986; Parnas, 1985).
- Μοντέλο της επαναχρησιμοποίησης λογισμικού (Davis, 1988).

Τα μοντέλα αυτά περιγράφουν κατά κανόνα γενικές κατηγορίες και περιπτώσεις ανάπτυξης λογισμικού, παρακάμπτοντας λεπτομερείς αναφορές σε συγκεκριμένες περιπτώσεις. Στην πραγματικότητα τα μοντέλα αυτά καλύπτουν τις γενικές γραμμές ανάπτυξης λογισμικού, ενώ η ανάπτυξη και συντήρηση ενός προϊόντος λογισμικού είναι προσαρμοσμένα στις συνθήκες ανάπτυξής του (επιχείρηση, χρήστες, ομάδα ανάπτυξης, ύπαρξη στελεχών, εμπειρία αναλυτών, υφιστάμενος εξοπλισμός, ...).

Σε κάθε φάση ανάπτυξης εφαρμόζονται οι τεχνικές που καλύπτονται από την επιστήμη της τεχνολογίας λογισμικού. Στον επόμενο πίνακα (2.1), παρουσιάζονται οι κύριες δραστηριότητες του κύκλου ζωής πληροφοριακών συστημάτων.

2.5 Ανάλυση και Σχεδίαση Συστημάτων

Η ανάπτυξη πληροφοριακών συστημάτων δίνει έμφαση στα ακόλουθα χαρακτηριστικά:

- Επιτάχυνση της ανάπτυξης ΠΣ

κεφ. 2 Πληροφοριακά Συστήματα

63

Πίνακας 2.1. Φάσεις κύκλου ζωής Π.Σ.

Φάσεις	Δραστηριότητες	Τεκμηρίωση
Ανάλυση Απαιτήσεων	Εκτίμηση αναγκών χρήστη Μελέτη σκοπιμότητας Καθορισμός απαιτήσεων χρήστη Προετοιμασία προγραμματισμού εργασιών	Αιτήσεις χρηστών Προτάσεις και εκτίμηση Κόστους Μελέτη σκοπιμότητας Ανάλυση απαιτήσεων Προγραμματισμός έργου
Λογική Σχεδίαση	Προετοιμασία γενικών προδιαγραφών σχεδίασης Διευκρίνισε τις απαιτήσεις των χρηστών	Λειτουργική περιγραφή Τεκμηρίωση απαιτήσεων σε δεδομένα
Φυσική Σχεδίαση	Προετοιμασία λεπτομερείς προδιαγραφές σχεδίασης Καθόρισε υποσυστήματα Σχεδίαση δομής βάσεων δεδομένων	Προδιαγραφές συστήματος υποσυστήματος Προδιαγραφές βάσεων δεδομένων Προδιαγραφές προγράμματος
Σχεδίαση Προγράμματος	Κωδικοποίηση προγράμματος Έλεγχοι τμημάτων προγράμματος Τεκμηρίωση τμημάτων	Τεκμηρίωση προγράμματος
Ολοκλήρωση Συστήματος	Διενέργεια ελέγχων υποσυστημάτων Διενέργεια ελέγχων συστημάτων Εκπαίδευση χρηστών Διενέργεια μετατροπών δεδομένων	Προγραμματισμός ελέγχων Αναφορά αποτελεσμάτων ελέγχων Εγχειρίδιο χρηστών
Εγκατάσταση και Λειτουργία	Λειτουργία συστήματος Συντήρηση νέου συστήματος Εκτίμηση συστήματος	Εγχειρίδιο λειτουργιών Εγχειρίδιο συντήρησης Αναφορά αποτίμησης

- Στη χρήση γλωσσών 4^{ης} γενιάς, οι οποίες διαθέτουν χαρακτηριστικά: γραφικών, συστημάτων διαχείρισης βάσεων δεδομένων, ικανότητες αλληλεπιδραστικής επικοινωνίας, υποβολή ερωτήσεων – απάντηση, γεννήτριες αναφορών, λεξικά δεδομένων, επεξεργασία κειμένων, επαναχρησιμοποιούμενο κώδικα, βιβλιοθήκη ανάπτυξης λογισμικού, ασφάλεια πρόσβασης στα δεδομένα, κ.α. και γλωσσών δης γενιάς οι οποίες είναι συμβολικές γλώσσες που παρέχουν αποτελεσματικούς τρόπους αναπαράστασης αντικειμένων και μεθόδων που χρησιμοποιούνται στην τεχνητή νοημοσύνη.
- Στην ανάπτυξη πρωτοτύπων για την πειραματική δοκιμή των εφαρμογών και τη διενέργεια ελέγχων. Πρωτότυπο θεωρείται ένα προϊόν λογισμικού το οποίο ανα-

πιπύσσεται γρήγορα και περιλαμβάνει τις βασικές λειτουργίες του τελικού προϊόντος. Με βάση αυτό γίνονται οι έλεγχοι, εντοπίζονται σφάλματα, διερευνώνται νέες απαιτήσεις, νέες λειτουργίες. Στη συνέχεια το σύστημα υφίσταται τις βελτιώσεις και η διαδικασία επαναλαμβάνεται μέχρι να πάρουμε το τελικό προϊόν.

- Στη χρήση τεχνικών δομημένης ανάπτυξης εφαρμογών με χρήση εργαλείων CASE (Computer-Aided Software Engineering). Τα εργαλεία αυτά και σε συνεργασία με τις τεχνικές δομημένης ανάλυσης και σχεδίασης, στοχεύουν στην αυτοματοποίηση της ανάπτυξης μεγάλων και σύνθετων ΠΣ.
- Ανάπτυξη εφαρμογών με χρήση αντικειμενοστραφούς προγραμματισμού (Object-oriented programming).
- Ανάπτυξη συστημάτων ανοιχτής αρχιτεκτονικής. Στοχεύει στην ανεξαρτησία δεδομένων, λογικής και διαδικασιών. Οι εφαρμογές που αναπτύσσονται θα είναι ανεξάρτητες από την πλατφόρμα ανάπτυξης, εύκολες σε μεταβολές και ολοκληρώσιμες με άλλες εφαρμογές.

Δομημένη ανάλυση - Σχεδίαση Συστήματος

Η ανάπτυξη ενός συστήματος γίνεται για να καλύπτει τις ανάγκες μιας επιχείρησης, ενός οργανισμού κ.λπ. Έτσι, αρχικά τίθενται οι αντικειμενοί στόχοι που πρέπει να πληροί το υπό ανάπτυξη σύστημα. Η διαδικασία ανάπτυξης ενός συστήματος έχει ως πρώτο βήμα τον προσδιορισμό των απαιτήσεων του συστήματος. Κατά τη φάση αυτή γίνεται καταγραφή, ανάλυση και καθορισμός των απαιτήσεων του συστήματος. Με τον καθορισμό των απαιτήσεων του συστήματος γίνεται μια περιγραφή των δυνατότήτων του, ούτως ώστε να μπορεί να ελεγθεί αν αυτό ικανοποιεί το σκοπό ανάπτυξής του. Στη φάση αυτή προσδιορίζονται οι απαιτήσεις σε μορφές πληροφοριών και σε λειτουργίες - επεξεργασίες που τις δημιουργούν ή τις μετασχηματίζουν. Διευκρινίζονται επίσης τα μη λειτουργικά θέματα που άπονται θεμάτων όπως η αξιοπιστία και η ασφάλεια των παρεχομένων πληροφοριών, οι παράγοντες που επηρεάζουν τη λειτουργία του συστήματος κ.α.

Κατά τη φάση της σχεδίασης του συστήματος οι λειτουργίες-επεξεργασίες μετασχηματίζονται έτσι ώστε να είναι εύκολη η υλοποίηση του συστήματος με τα διαθέσιμα μέσα υλικού και λογισμικού. Έτσι, καταλήγουμε στο καθορισμό των τμημάτων (modules) που υλοποιούν συγκεκριμένες λειτουργίες-επεξεργασίες, όπως του τρόπου διαχείρισης των δεδομένων, του συστήματος επικοινωνίας χρήστη -συστήματος και άλλων εργασιών.

Για μια περισσότερη σε βάθος μελέτη των διαφόρων μεθόδων και των εργαλείων που χρησιμοποιούνται για το προσδιορισμό των απαιτήσεων και τη σχεδίαση ενός συστή-

ματος μπορεί κανείς να ανατρέξει στη βιβλιογραφία (IBM, 1974; DeMarco, 1978; Gane and Sarson, 1979; Yeh, 1982; 1983; Alford, 1985; Ross, 1985; Whitten et al., 1989; Yourdon, 1989; Sommerville, 1992).

Εργαλεία ανάλυσης

Για την ανάπτυξη ποιοτικού λογισμικού με βάση τις αρχές της δομημένης ανάλυσης είναι απαραίτητη η χρήση των κατάλληλων εργαλείων. Έτσι, για τη γραφική αναπαράσταση των λειτουργιών και της λογικής του συστήματος γίνεται χρήση των ακολούθων εργαλείων (Yourdon and Constantine, 1975; Myers, 1975; Gane and Sarson, 1979; DeMarco, 1978; Yourdon, 1989; Gibson, 1989; Norman and Nunamaker, 1989; McCusker, 1991; Lindholm, 1992): Διαγραμμάτων Ροής Δεδομένων - ΔΡΔ (Data Flow Diagrams - DFD), των Λεξικών Δεδομένων (Data Dictionary - DD), των Διαγραμμάτων Δομής ή Αρχιτεκτονικής (Structure Chart - SC) και των Διαγραμμάτων Δομής Δεδομένων (Data Structure Diagrams - DSD). Για μια γενικότερη επισκόπηση των παραπάνω εννοιών μπορεί κανείς να ανατρέξει στη σχετική βιβλιογραφία (Yourdon, 1989; Whitten et al., 1989; McLeod, 1993 κ.α.).

2.6 Αντικειμενοστραφής προγραμματισμός

Στον αντικειμενοστραφή προγραμματισμό (object-oriented programming), τα βασικά δομικά συστατικά είναι οι κλάσεις και τα αντικείμενα. Το αντικείμενο μοντελοποιεί ένα τμήμα της πραγματικότητας και για αυτό είναι κάτι το οποίο υπάρχει στο χώρο και το χρόνο. Σαν αντικείμενο μπορεί να θεωρηθεί κάτι το οποίο έχει σαφώς ορισμένα όρια. Σύμφωνα με τον Booch (1994), ένα αντικείμενο έχει: κατάσταση, συμπεριφορά και ταυτότητα. Η δομή και η συμπεριφορά όμοιων αντικειμένων καθορίζεται στην κοινή τους κλάση. Ο Joned (1979), ξεκαθαρίζει περισσότερο τον όρο επισημαίνοντας ότι "στο μοντέλο του αντικειμένου, η έμφαση δίνεται στον χαρακτηρισμό των συστατικών του φυσικού ή αφηρημένου συστήματος που μοντελοποιεί το λογισμικό σύστημα. Τα αντικείμενα έχουν μια συγκεκριμένη ακεραιότητα που δεν μπορεί να παραβιαστεί. Ένα αντικείμενο μπορεί μόνο να αλλάξει κατάσταση, να συμπεριφερθεί, να διαχειριστεί και να σταθεί σε σχέση με άλλα αντικείμενα με τρόπους που είναι συγκεκριμένοι για το αντικείμενο. Οι Stefik και Bobrow (1986), ορίζουν τα αντικείμενα ως οντότητες που συνδυάζουν τις ιδιότητες των διαδικασιών με δεδομένα, μια και εκτελούν υπολογισμούς και σώζουν την κατάστασή τους. Μπορούμε να πούμε ότι ένα αντικείμενο είναι μία σαφής οντότητα που παρουσιάζει μία καλά καθορισμένη συμπεριφορά.

Η αντικειμενοστραφής διαδικασία (Booch, 1994), είναι μία καλώς ορισμένη επαναλαμβανόμενη διαδικασία. Υπάρχουν δύο χαρακτηριστικά τα οποία είναι κοινά σε όλα

τα πετυχημένα έργα λογισμικού: η ύπαρξη μίας ισχυρά διορατικής αρχιτεκτονικής και η εφαρμογή ενός καλά διαχειριζόμενου επαναληπτικού και σταδιακού κύκλου ζωής ανάπτυξής των. Τα επιτυχημένα έργα αντικειμενοστραφούς προγραμματισμού δεν εμφανίζουν ούτε άναρχους αλλά και ούτε τρομερά αυστηρούς και γραφειοκρατικούς κύκλους ζωής της ανάπτυξης. Αντίθετα, η διαδικασία που οδηγεί στην επιτυχή κατασκευή αντικειμενοστραφών αρχιτεκτονικών είναι σταδιακή με την έννοια ότι κάθε πέρασμα από ένα κύκλο ανάλυσης-σχεδιασμού-αξιολόγησης, οδηγεί στη βαθμιαία εκλέπτυνση των στρατηγικών και τακτικών αποφάσεων, με αποτέλεσμα την σύγκλιση σε μία λύση που ικανοποιεί τις απαιτήσεις του τελικού χρήστη και που παρόλα αυτά είναι απλή, αξιόπιστη και προσαρμοζόμενη.

Ο αντικειμενοστραφής προγραμματισμός Booch (1994), είναι μία μέθοδος ανάπτυξης στην οποία τα προγράμματα είναι οργανωμένα ως συλλογές συνεργαζόμενων αντικειμένων, κάθε ένα από τα οποία αναπαριστά ένα στιγμιότυπο κάποιας κλάσης. Οι κλάσεις του προγράμματος είναι μέλη μίας ιεραρχίας κλάσεων που διέπεται από μία κληρονομικότητα σχέσεων. Υπάρχουν τρία σημαντικά στοιχεία σε αυτό τον ορισμό:

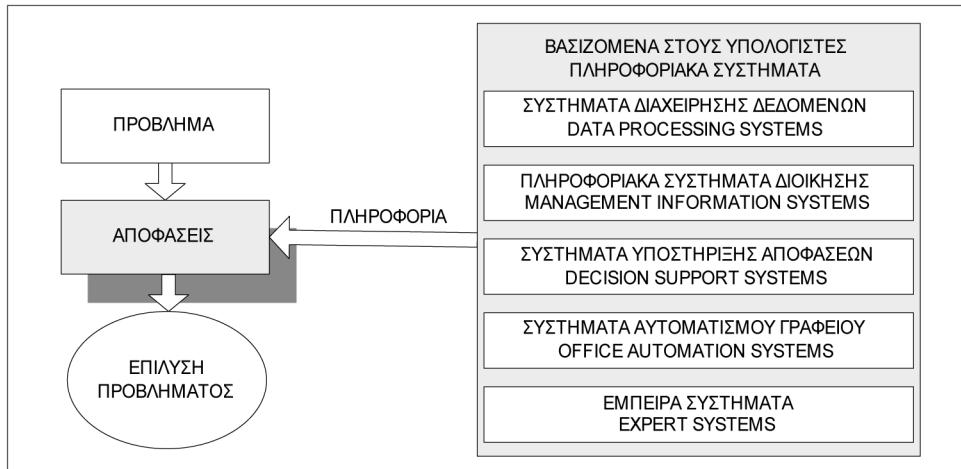
- ο αντικειμενοστραφής προγραμματισμός χρησιμοποιεί αντικείμενα και όχι αλγόριθμους ως βασικά λογικά δομικά στοιχεία,
- κάθε αντικείμενο είναι ένα στιγμιότυπο κάποιας κλάσης, και
- οι κλάσεις σχετίζονται μεταξύ τους μέσω σχέσεων κληρονομικότητας.

Ένα πρόγραμμα μπορεί να φαίνεται αντικειμενοστραφές, αν όμως κάποιο ή κάποια από τα στοιχεία αυτά απουσιάζει τότε δεν είναι.

Οι Cardelli και Wegner (1985), ορίζουν ότι μία γλώσσα προγραμματισμού είναι αντικειμενοστραφής εάν και μόνο εάν:

- Υποστηρίζει αφαιρέσεις αντικειμένων με ένα σύστημα επικοινωνίας αποτελούμενο από επώνυμες λειτουργίες και μία κρυφή τοπική κατάσταση.
- Τα αντικείμενα έχουν ένα αντίστοιχο τύπο (κλάση).
- Οι τύποι (κλάσεις) μπορεί να κληρονομούν χαρακτηριστικά από υπερτύπους (υπερκλάσεις).

Μία γλώσσα υποστηρίζει τη κληρονομικότητα εάν είναι δυνατό να εκφραστούν σχέσεις της μορφής 'is a' ανάμεσα στους τύπους. Έτσι, για παράδειγμα ένα κόκκινο τριαντάφυλλο είναι ένα λουλούδι και ένα λουλούδι είναι ένα είδος φυτού. Αν μία γλώσσα δεν παρέχει απευθείας υποστήριξη κληρονομικότητας τότε δεν είναι αντικειμενοστραφής αλλά βασισμένη σε αντικείμενα (Object Based).



Σχήμα 2.2. Μοντέλο πληροφοριακών συστημάτων βασιζόμενων στη χρήση υπολογιστών
(πηγή: McLeod, 1993)

2.7 Πληροφοριακά συστήματα βασιζόμενα στους πλεκτρονικούς υπολογιστές

Με την ανάπτυξη φθηνότερης, φιλικότερης και καλύτερης τεχνολογίας, ξεκίνησε η εξελικτική πορεία των Βασιζόμενων στους Υπολογιστές Πληροφοριακών Συστημάτων (Computer-Based Information Systems - CBIS). Η εξέλιξη αυτή ήταν το αποτέλεσμα μιας συλλογικής μάθησης των ανθρώπων, οι οποίοι έμαθαν τα θετικά της τεχνολογίας και της δύναμής της. Ο McLeod (1993), χρησιμοποιεί τον όρο CBIS για να περιγράψει τις πέντε κατηγορίες εφαρμογών που χρησιμοποιούν Η/Υ (σχήμα 2.2) και οι Bocic et al. (1999), για να τα διαχωρίσουν από τα βασιζόμενα στο χαρτί ΠΣ (paper-based IS). Σήμερα, με τον όρο Πληροφοριακά Συστήματα εννοούμε κατά κανόνα αυτά που βασίζονται στη χρήση υπολογιστών.

2.7.1 Ιστορική εξέλιξη Πληροφοριακών Συστημάτων

Μετά τα απλά Συστήματα Επεξεργασίας (Διεκπεραίωσης) Συναλλαγών (Δοσοληψιών) (ΣΕΣ, Transaction Processing Systems - TPS), στη δεκαετία του '60 έκαναν την εμφάνισή τους τα Πληροφοριακά Συστήματα Διοίκησης (ΠΣΔ, Management Information Systems - MIS), τα οποία χαρακτηρίζονται κυρίως από την ικανότητά τους να παράγουν περιοδικές αναφορές, όπως μια καθημερινή λίστα εργαζομένων