



## ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

### ΤΜΗΜΑ ΤΕΧΝΟΛΟΓΩΝ ΓΕΩΠΟΝΩΝ

<b>Τίτλος Έργου:</b>	Εφαρμογές σύγχρονων Τεχνολογιών Κινητών Συσκευών για την υποστήριξη του Συστήματος Ολοκληρωμένης Διαχείρισης στη γεωργική παραγωγή και την Ιχνηλασιμότητα αγροτικών προϊόντων  New services and mobile technologies for supporting an Integrated Management System in agricultural production and Traceability of agricultural products
<b>Ακρωνύμιο:</b>	AGROTRACE
<b>ΕΣΠΑ 2007-2013 Δράση:</b>	«ΣΥΝΕΡΓΑΣΙΑ 2011»
<b>Κωδικός Έργου :</b>	11ΣΥΝ-3-2032
<b>Φορέας στην Σύμπραξη:</b>	Συντονιστής
<b>Επωνυμία του Κύριου Φορέα:</b>	Τμήμα Τεχνολόγων Γεωπόνων ΤΕΙ Δυτικής Ελλάδας

<b>Παραδοτέο</b>	Π3.1
<b>Τίτλος Παραδοτέου</b>	Αναφορά Υλοποίησης Ολοκληρωμένου Κινητού Πληροφοριακού Συστήματος

<b>Συντελεστές-Συγγραφείς:</b>	Αγλαΐα Λιόπα Τσακαλίδου Παντελής Μπαρούχας Αθανάσιος Κουλόπουλος Νικόλαος Μαλάμος Ιωάννης Τζήμας Τσώλης Δημήτριος Καρατζάς Εμμανουήλ Καρατζάς Παναγιώτης Κανδύλη Αναστασία Ντάκουλα Λαμπρινή
--------------------------------	---

<http://www.agrotrace.gr>

## Περιεχόμενα

---

Αρχιτεκτονική συστήματος .....	5
Αρχιτεκτονική Συστήματος και Βασικά Στοιχεία.....	6
Οι τεχνολογίες που αξιοποιούνται.....	9
Οι τεχνολογίες αυτές περιγράφονται αναλυτικά στα παρακάτω κεφάλαια.	10
Καινοτόμα χαρακτηριστικά.....	10
Τεχνολογίες Υλοποίησης .....	13
Λογισμικό Ανοικτού Κώδικα .....	13
Εισαγωγή .....	13
Συμμετοχή στην ανάπτυξη λογισμικού Ανοικτού κώδικα .....	14
Πλεονεκτήματα χρήσης λογισμικού Ανοικτού Κώδικα.....	14
Μειονεκτήματα χρήσης λογισμικού Ανοικτού Κώδικα .....	15
Γλώσσες προγραμματισμού .....	16
PHP .....	16
Επεκτάσεις αρχείων και διακομιστές.....	16
Χρησιμοποίηση PHP .....	17
Εκτέλεση αρχείων php .....	18
Παράδειγμα κώδικα PHP .....	18
MySQL.....	18
Βάσεις Δεδομένων .....	19
Η Σχεσιακή Βάση Δεδομένων MySQL .....	20
SQL.....	21
Πρότυπα της SQL.....	22
SQL Server .....	22
Ενδιάμεση μνήμη .....	23
Ταυτοχρονισμός .....	23
Ανάκτηση δεδομένων .....	24
CLR .....	25
HTML .....	25
Η HTML σήμερα.....	26
Εντολές HTML – Παραδείγματα .....	26
Σύνδεσμοι (links) .....	27

CSS .....	28
Τι είναι η CSS .....	29
Σύνταξη Των Στοιχείων Id.....	29
Σύνταξη Των Στοιχείων Class.....	29
Σύνδεση CSS Με HTML.....	30
Εξωτερικό Αρχείο .....	30
Φύλλα Στυλ.....	30
Εισαγωγή ενός Φύλλου Στυλ (Style Sheet) .....	31
Εξωτερικά Φύλλα Στυλ (External Style Sheets).....	31
Εσωτερικά Φύλλα Στυλ (Internal Style Sheets).....	32
Javascript .....	32
Τρόπος σύνταξης.....	33
Δείγμα κώδικα javascript .....	33
Οφέλη της JavaScript.....	34
Τρέχοντας JavaScript .....	35
Events - Γεγονότα .....	35
XML.....	35
Εισαγωγικά.....	36
Προέλευση και στόχοι (Origin and goals) .....	37
Βασική Ορολογία.....	38
Χαρακτήρας Unicode.....	38
Επεξεργαστής και Εφαρμογή .....	38
Σήμανση και Περιεχόμενο.....	38
Ετικέτα .....	38
Στοιχεία.....	39
Χαρακτηριστικά.....	39
Παράδειγμα .....	39
Χαρακτήρες και διαφυγή. ....	39
Βήματα Υλοποίησης Συστήματος.....	40
Δημιουργία web server μέσω Xampp .....	40
Δημιουργία βάσης.....	40
Ρυθμίσεις.....	41
Εγκατάσταση διαδικτυακής εφαρμογής.....	41
Σύστημα λήψης απόφασης.....	43

Έμπειρα συστήματα .....	43
Ορισμός Έμπειρου Συστήματος .....	43
Λειτουργικά στοιχεία .....	44
Δομικά Στοιχεία .....	45
Αρχιτεκτονική .....	46
Ταξινόμηση.....	49
Εργαλεία ανάπτυξης έμπειρων συστημάτων .....	50
Γλώσσες 5ης γενιάς .....	50
Πλεονεκτήματα .....	50

## Αρχιτεκτονική συστήματος

---

Τα πρώτα συστήματα Ολοκληρωμένης Διαχείρισης Γεωργικής Παραγωγής (Integrated Crop Management – ICM) προέρχονται από συστήματα ειδικών ερευνών που στοχεύουν να προγραμματίσουν τους υπολογιστές με κανόνες, οι οποίοι θα τους επιτρέπουν να σκεφτούν όπως ένας γεωργός που αναπτύσσει μια καλλιέργεια. Από την πρώτη έρευνα, τα συστήματα μπορούν να είναι χρήσιμα τόσο στη βοήθεια που προσφέρουν στους παραγωγούς σε ότι αφορά τη λήψη αποφάσεων, την υλοποίηση ενεργειών που εξασφαλίζουν την ποιότητα κ.α.

Τα συστήματα σχεδιάστηκαν για να παρέχουν συγκεκριμένη καθοδήγηση για τον παραγωγό. Συγκεκριμένα περιλαμβάνουν τη βάση δεδομένων με ένα σύνολο από πληροφορίες, οδηγίες και δεδομένα και που αφορούν τη καλλιέργεια, τις αλληλεπιδράσεις φυτοφαρμάκων και λιπασμάτων και τις πληροφορίες καθοδήγησης των καλλιεργητών σε ότι αφορά την εξασφάλιση της ποιότητας της τελικής παραγωγής. Αποτελούν λοιπόν ένα πρόγραμμα που συνδυάζει τη γνώση με συγκεκριμένες πληροφορίες καλλιέργειας και ένα μηχανισμό εισόδου στη βάση δεδομένων που αφορούν δεδομένα με πιθανές οδηγίες για λιπάνσεις και ποτίσματα και κατευθυντήριες οδηγίες για τον παραγωγό. Τα συστήματα αυτά μπορούν να εφαρμοστούν χρησιμοποιώντας μια ποικιλία από πλατφόρμες (π.χ. internet, προσωπικός υπολογιστής) καθώς και μια ποικιλία από προσεγγίσεις στον υπολογιστή που βασίζονται στο γεγονός ότι η γνώση είναι διαθέσιμη από ένα κεντρικό σημείο (πιθανώς έξω από την τοπική τοποθεσία και προσβάσιμη και ενσωματωμένη τοπικά όταν χρειάζεται).

Οι τεχνολογικές λύσεις δε χρειάζονται να είναι προσφιλείς στο χρήστη. Αυτό που προέχει είναι τα συστήματα να βελτιώσουν συνολικά την αποτελεσματικότητα, να εξασφαλίσουν την ποιότητα και να βοηθήσουν στην ιχνηλασιμότητα, να διαχειριστούν σωστά τις πληροφορίες που λαμβάνουν και να οριοθετήσουν τον έλεγχο και την πρόσβαση που θα έχει ο χρήστης στις πληροφορίες.

Τα μέχρι συστήματα έως σήμερα επικεντρώνονται στην υποστήριξη του παραγωγού και όχι στη συμμετοχή του καταναλωτή στην όλη διαδικασία. Το προτεινόμενο σύστημα θα υποστηρίξει και τον καταναλωτή παρέχοντας του τις σωστές πληροφορίες και τις υπηρεσίες ιχνηλασιμότητας των προϊόντων. Επίσης το σύστημα θα χρησιμοποιεί τεχνικές βάσεων δεδομένων για να βελτιώσουν την εξόρυξη συμπερασμάτων και την πρόσβαση στο διαδίκτυο.

Το προτεινόμενο σύστημα μπορεί να χαμηλώσει το κόστος παραγωγής, να βελτιώσει την αποτελεσματικότητα και ποιότητα και να μειώσει τις δυσκολίες για τον παραγωγό.

Επίσης, το προτεινόμενο σύστημα θα διευρύνει το είδος των συσκευών συμπεριλαμβανομένων των tablets και smartphones που αξιοποιούνται στη σύγχρονη γεωργική παραγωγή.

## Αρχιτεκτονική Συστήματος και Βασικά Στοιχεία

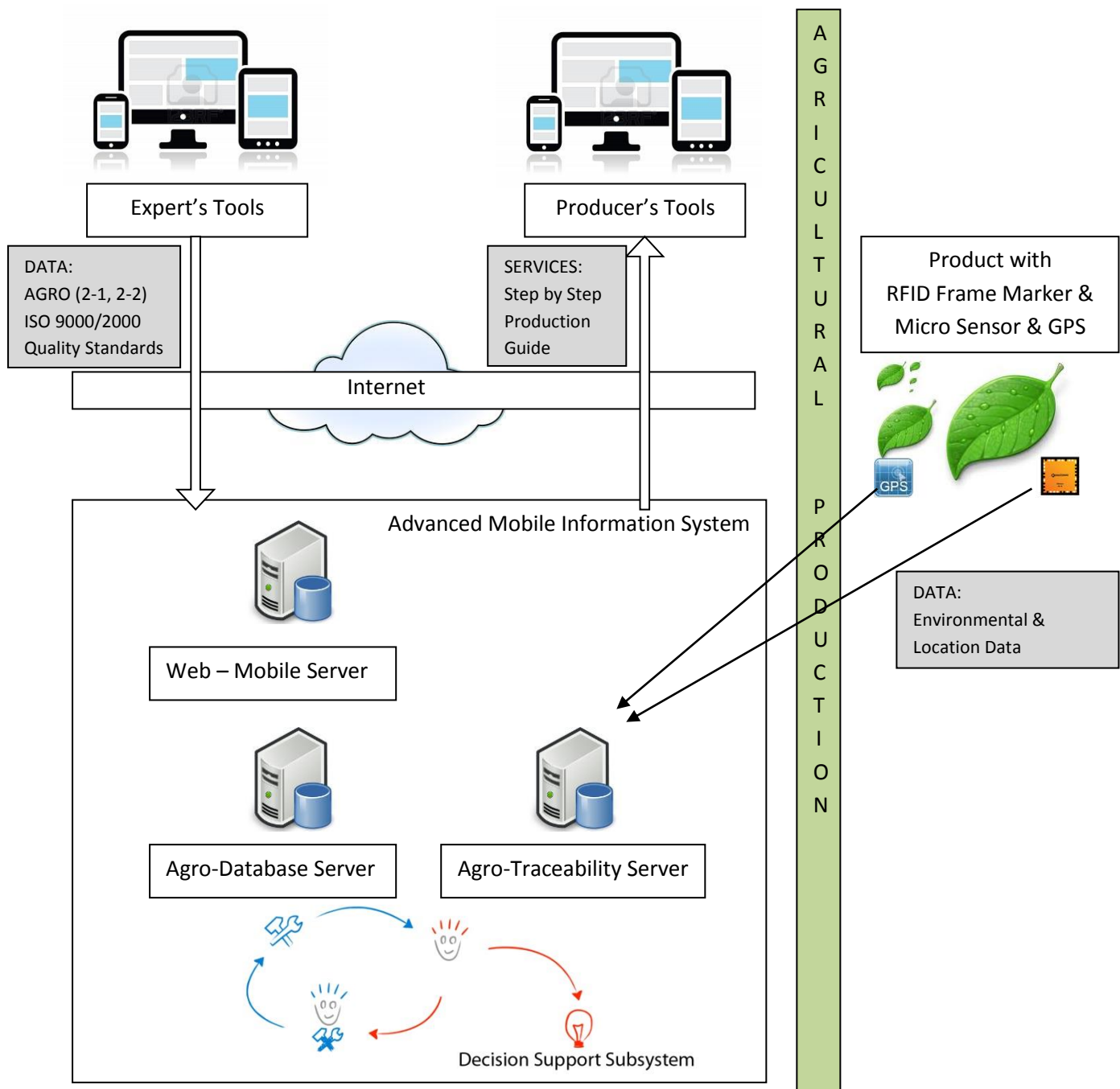
Το προτεινόμενο σύστημα βασίζεται στο τρίπτυχο: Ποιότητα Παραγωγής, Ασφάλεια Καταναλωτή και Προστασία του Περιβάλλοντος. Βάσει του τρίπτυχου ένα ολοκληρωμένο σύστημα υποστήριξης της διαχείρισης παραγωγής και εξασφάλισης της ποιότητας είναι αναγκαίο. Το προτεινόμενο σύστημα αποσκοπεί:

- 1) Στην υποστήριξη των ροών που απαιτούνται για την εξασφάλιση της ποιότητας της παραγωγής βάσει των διεθνών προτύπων AGRO (2-1 / 2-2).
- 2) Παροχή online και κινητών υπηρεσιών στον παραγωγό για:
  - Την επισκόπηση και ανάλυση των απαραίτητων ενεργειών που γίνονται στο χωράφι για την οργάνωση και τη διαχείριση της καλλιέργειας, τη λίπανση κ.α. βάσει της συγκεκριμένης χρονικής στιγμής και του ιστορικού της καλλιέργειας.
  - Διαχείριση της καλλιέργειας και της παραγωγής ανά πάσα στιγμή και από οποιαδήποτε συσκευή.
  - Αποθήκευση των χαρακτηριστικών των προϊόντων με τρόπο που να εξασφαλίζει την μελλοντική ιχνηλασιμότητα του προϊόντος.

Για την επίτευξη των στόχων αυτών προτείνεται η ανάπτυξη ενός κινητού πληροφοριακού συστήματος. Το σύστημα εκτός των άλλων θα περιλαμβάνει και δυνατότητες υποστήριξης λήψης απόφασης για τον παραγωγό έτσι ώστε η Ολοκληρωμένη Διαχείριση της καλλιέργειας να επιτευχθεί.

Το σύστημα θα παρέχει επίσης κινητές υπηρεσίες για την υποστήριξη της διαχείρισης της παραγωγής πολλών ταυτόχρονων και γεωγραφικά απομακρυσμένων προϊόντων.

Η αρχιτεκτονική...



Η κυριότερες οντότητες της αρχιτεκτονικής είναι οι παρακάτω:

Το Προηγμένο Κινητό Πληροφοριακό Σύστημα (Advanced Mobile Information System): Το προηγμένο κινητό πληροφοριακό σύστημα περιλαμβάνει όλα το αναγκαίο λογισμικό και υλικό για να παρέχει πλήρεις και αποδοτικές online και κινητές υπηρεσίες στις ομάδες χρηστών. Τα βασικά του τμήματα είναι τα παρακάτω:

- Η Βάση Δεδομένων Γεωργικής Παραγωγής (Agricultural Database Server): είναι ένα αποθετήριο δεδομένων γεωργικής παραγωγής που επικεντρώνεται στην ορθή παραγωγή βάσει των προτύπων AGRO (2-1, 2-2) και του προτύπου ποιότητας ISO 9000/2000. Η βάση δεδομένων αποθηκεύει μακροπρόθεσμα τα απαραίτητα δεδομένα, τις πληροφορίες, τα κριτήρια, τις εικόνες κ.α. στοιχεία που καθοδηγούν τον παραγωγό στην εξασφάλιση της ποιότητας των παραγόμενων προϊόντων. Οι ειδικοί παραγωγοί καθορίζουν τα απαραίτητα δεδομένα και τους τύπους τους.
- Ο Εξυπηρετητής Βάσης Δεδομένων Ιχνηλασιμότητας (Traceability Database Server): Πρόκειται για ένα αποθετήριο που αποθηκεύει τα δεδομένα που σχετίζονται με τη μελλοντική ιχνηλασιμότητα των προϊόντων, ιδιαίτερα μετά την αποθήκευση και τη μεταφορά τους. Οι πληροφορίες που αποθηκεύονται περιλαμβάνουν την τοποθεσία, τα δεδομένα περιβάλλοντος και το χρόνο αποθήκευσης ενός προϊόντος. Η βάση δεδομένων έχει τη δυνατότητα να αναγνωρίζει μοναδικά κάθε προϊόν και να ανιχνεύει τις παραμέτρους αποθήκευσης και μεταφορά κατά το κύκλο ζωής του.
- Το Σύστημα Υποστήριξης Λήψης Απόφασης (Decision Support System - DSS): Το DSS χρησιμοποιεί δεδομένα από τη βάση δεδομένων γεωργικής παραγωγής και τα συνδυάζει με μετεωρολογικά δεδομένα και δεδομένα περιβάλλοντος για να προτείνει λύσεις ή να στηρίξει την απόφαση του παραγωγού σε κάποια συγκεκριμένη χρονική στιγμή και φάση της καλλιέργειας. Επίσης προτείνει τα βήματα που απαιτούνται για την εξασφάλιση της ποιότητας του τελικού αποτελέσματος. Το DSS επίσης παράγει συναγερμούς σε περιπτώσεις ακραίων καιρικών φαινομένων και περιβαλλοντικών δεδομένων που μπορεί να έχουν αρνητική επίδραση στην καλλιέργεια. Οι συναγερμοί παρουσιάζονται στους παραγωγούς για την υποστήριξή τους καθ' όλη τη διάρκεια της καλλιέργειας.
- Ο Εξυπηρετητής Διαδικτύου και Κινητών Υπηρεσιών (Web and Mobile Server): Ο εξυπηρετητής παρέχει απρόσκοπτη επικοινωνία μέσω ενσύρματης ή ασύρματης επικοινωνίας και συνεχόμενη ανταλλαγή δεδομένων μεταξύ όλων των επιμέρους τμημάτων της αρχιτεκτονικής. Επίσης είναι η πύλη πρόσβασης του συστήματος από τις ομάδες χρηστών.

Τα εργαλεία των ειδικών (Expert's Tools): Πρόκειται για εργαλεία λογισμικού και φιλικές διεπαφές χρήστη, όπως επίσης και εφαρμογές κινητών συσκευών που επιτρέπουν σ' έναν ειδικό να συλλέξει, να προωθήσει και να αποθηκεύσει κρίσιμα δεδομένα για την καλλιέργεια στις βάσεις δεδομένων του συστήματος. Ο ειδικός έχει τη δυνατότητα να χρησιμοποιήσει το desktop / laptop / tablet ή το smart phone για να συλλέξει τη πληροφορία και να συνδεθεί στο υπόλοιπο σύστημα.

Τα εργαλεία των παραγωγών (Producer's Tools): Πρόκειται για εργαλεία λογισμικού και εύχρηστες διεπαφές χρήστη, όπως καθώς και εφαρμογές κινητών συσκευών που υποστηρίζουν τον παραγωγό κατά την καλλιέργειά του. Τα βασικά εργαλεία υποστήριξης είναι ένας step by step οδηγός εξασφάλισης της ποιότητας κατά τη



διεργασία της παραγωγής, συναγερμοί σχετικά με τις καιρικές συνθήκες ή άλλες παραμέτρους που μπορεί να έχουν αρνητική επίδραση στην παραγωγή και συναφείς πληροφορίες. Ο στόχος είναι ο παραγωγός να έχει στο χέρι του και επί τόπου στον τόπο παραγωγής ένα εργαλείο για να σχεδιάσει, να διαχειριστικά και να προστατέψει την παραγωγή του.

Το προϊόν (The Product): Το προϊόν είναι το αποτέλεσμα της διαδικασίας παραγωγής, το οποίο έχει συσκευασθεί και είναι έτοιμο να μεταφερθεί προς τον τελευταίο προορισμό του. Το προϊόν προστατεύεται από έναν frame marker που το αναγνωρίζει μοναδικά και με τη χρήση ενός tablet ή smart phone ένας καταναλωτής θα μπορεί να πληροφορηθεί πότε και που το προϊόν παράχθηκε, τα μέρη στα οποία έχει μεταφερθεί, τις συνθήκες αποθήκευσης κ.α. Επιπρόσθετα ο frame marker χρησιμοποιείται και ως μια RFID λύση για να μπορεί να αξιοποιηθεί εύκολα από smartphones και όχι από δύσχρηστους barcode αναγνώστες κατά τη μεταφορά και αποθήκευση των προϊόντων. Επίσης, το προϊόν προστατεύεται από ένα μικροαισθητήρα και δέκτη GPS οι οποίοι συνδυαστικά συλλέγουν περιβαλλοντικά δεδομένα και χωρικά δεδομένα τα οποία και μεταδίδονται στον εξυπηρετητή και τη βάση δεδομένων ιχνηλασιμότητας. Ο στόχος είναι το πληροφοριακό σύστημα να αποθηκεύει και να διαχειρίζεται πληροφορίες σχετικά με το προϊόν και τις συνθήκες αποθήκευσης και μεταφοράς του. Τα δεδομένα αυτά αυξάνουν την προστιθέμενη αξία του προϊόντος.

Τα δεδομένα που ανταλλάσσονται σε σχέση με τις υπηρεσίες που παρέχονται βάσει της αρχιτεκτονικής αυτής περιλαμβάνουν:

- Δεδομένα ειδικών: τα δεδομένα που συλλέγονται και μεταφέρονται από τα εργαλεία των ειδικών και τις συσκευές τους προς το κυρίως σύστημα.
- Ο Step by Step οδηγός για τον παραγωγό: Δεδομένα για το επόμενο βήμα προς μια ποιοτική και αποδοτική καλλιέργεια.
- Δεδομένα Αισθητήρων και GPS: Δεδομένα περιβάλλοντος και τοποθεσίας που μεταφέρονται από τους μικροαισθητήρες προς τη βάση δεδομένων ιχνηλασιμότητας.
- Δεδομένα Frame Marker: Τα δεδομένα της ταυτότητας κάθε προϊόντος βάσει ενός RFID τα οποία μέσω ενός Frame Marker ανακατευθύνουν των φυλλομετρητή διαδικτύου των κινητών συσκευών προς πληροφορίες σχετικά με το που, πότε και πως παράχθηκε ένα προϊόν, που αποθηκεύθηκε και πως μεταφέρθηκε.

### **Οι τεχνολογίες που αξιοποιούνται**

---

Συνοπτικά οι τεχνολογίες που αξιοποιούνται για την υλοποίηση της αρχιτεκτονικής είναι οι παρακάτω:

Με επικέντρωση την ολοκληρωμένη διαχείριση:

- Βάσεις δεδομένων γεωργικής παραγωγής που συλλέγουν και διαχειρίζονται στοιχεία για τα διεθνή πρότυπα ποιότητας AGRO (2-1, 2-2) και ISO 9000/2000.
- Κινητά δίκτυα για τις συσκευές που αξιοποιούνται από τους παραγωγούς. 3G και 4G δίκτυα για ευρυζωνική πρόσβαση.
- Ανάπτυξη εφαρμογών κινητών συσκευών σε IOS και Android για τα εργαλεία των ειδικών και τα εργαλεία των παραγωγών.
- Υλοποίηση εφαρμογών και υπηρεσιών διαδικτύου Development for Web για την ανταλλαγή δεδομένων μεταξύ των εργαλείων των ειδικών και των εργαλείων των παραγωγών.
- Συστήματα Web GIS Systems για την πληροφορία GPS και την παρουσίασή της σε χαρτογραφικά συστήματα.
- Συστήματα Υποστήριξης Λήψης της Απόφασης. Νευρωνικά δίκτυα και αλγόριθμοι τεχνητής νοημοσύνης αξιοποιούνται για την αποδοτική λειτουργία του συστήματος και για την παραγωγή και παρουσίαση των συναγεμίων.

Με επικέντρωση στην ιχνηλασιμότητα:

- Βάσεις δεδομένων ιχνηλασιμότητας.
- Μοναδικά αναγνωριστικά RFID για την ταυτοποίηση του προϊόντος και την ανακατεύθυνση των φυλλομετρητών σταθερών ή κινητών συσκευών προς πληροφορίες σχετικά με το προϊόν που είναι αποθηκευμένες στη βάση δεδομένων της ιχνηλασιμότητας.
- Μικρο αισθητήρες και μικρό GPS δέκτες για τη συλλογή και την αποστολή δεδομένων περιβάλλοντος και τοποθεσίας.
- Δίκτυα αισθητήρων για την ανταλλαγή δεδομένων μεταξύ των αισθητήρων, των δεκτών και των εξυπηρετητών που υποστηρίζουν την ιχνηλασιμότητα.

Οι τεχνολογίες αυτές περιγράφονται αναλυτικά στα παρακάτω κεφάλαια.

### **Καινοτόμα χαρακτηριστικά**

Το προτεινόμενο σύστημα βασίζεται σε ένα συνδυασμό τεχνολογιών και βασικών αρχών. Συγκεκριμένα περιλαμβάνει ψηφιακές τεχνολογίες που αφορούν τους παραγωγούς και οι οποίες στηρίζονται σε διεθνή τεχνολογικά πρότυπα. Η πρόσβαση στα δεδομένα και η ανταλλαγή αυτών βασίζεται στα διεθνή μεταδεδομένα και στα έμπειρα συστήματα.

Η χαρτογράφηση των μεταδεδομένων στηρίζεται στα διεθνή πρότυπα μεταδεδομένων και εφαρμόζεται στην αγροτική πληροφορική. Η διαδικασία χαρτογράφησης εφαρμόζεται με ένα αυτόνομο τρόπο μέσα στις βάσεις δεδομένων που χρησιμοποιούνται. Η βάση δεδομένων κατασκευάζεται και εφαρμόζεται με έναν τρόπο σύμφωνα με τα διεθνή μεταδεδομένα. Ο αριθμός των εισαγωγών αρχικά είναι αυστηρά περιορισμένος και οι βάσεις δεδομένων είναι προσβάσιμες.

Το σύστημα περιλαμβάνει και ένα έμπειρο (expert) υποσύστημα το οποίο χρησιμοποιεί τεχνητή νοημοσύνη και συνεισφέρει στην εξασφάλιση της ποιότητας. Τα νευρωνικά δίκτυα και οι αλγόριθμοι ασαφούς λογικής υλοποιούνται ως μαύρα κουτιά για την εισαγωγή δεδομένων και για τα αποτελέσματα εξόδου. Τα άνωθεν έχουν μετατραπεί από γλώσσες προγραμματισμού σε βιβλιοθήκες δυναμικής σύνδεσης που μπορούν να ενσωματωθούν στο σύνολο του συστήματος πληροφοριών. Οι διεπαφές του χρήστη που υλοποιούνται σε διαδικτυακές γλώσσες μπορούν να εκτελέσουν τους αλγορίθμους με βάση τα δεδομένα που εισάγονται από τον τελικό χρήστη. Τα αποτελέσματα έχουν επίσης θα ενσωματωθούν στις διαδικτυακές εφαρμογές.

Το ενσωματωμένο σύστημα υποστηρίζει τους παραγωγούς στο να εξασφαλίσουν ποιότητα στην καλλιέργειά τους. Η αξιολόγηση του συστήματος θα πρέπει να επικεντρωθεί στα αλγοριθμικά μέτρα απόδοσης για τη διαδικασία λήψης αποφάσεων, τα στατιστικά στοιχεία και τη μελέτη αξιολόγησης του συνόλου των επιδόσεων του έμπειρου συστήματος. Με βάση τη μελέτη αξιολόγησης του έμπειρου συστήματος θα πρέπει να βελτιστοποιηθεί τελικά.

Επιπλέον, δεδομένου ότι μακροπρόθεσμα το σύστημα συλλέγει ένα τεράστιο ποσό δεδομένων που παράγονται από χιλιάδες προϊόντα, οι τεχνικές εξόρυξης δεδομένων θα χρησιμοποιούνται προς δύο κατευθύνσεις με αποτελεσματική πρόσβαση στα κρίσιμα δεδομένα από τους παραγωγούς και τη χρήση των κρίσιμων δεδομένων στο υποσύστημα υποστήριξης αποφάσεων. Το κλειδί στη χρήση των μεθόδων εξόρυξης δεδομένων έγκειται στην εξαγωγή ειδικών κανόνων από τα γεωργικά δεδομένα που συλλέγονται, τα οποία χρησιμοποιούνται για να υποστηρίξουν περαιτέρω την εξασφάλιση της ποιότητας. Τα γεωργικά στοιχεία περιλαμβάνουν το ιστορικό, τις ασθένειες των προϊόντων, τα συμπτώματα, τα εργαστηριακά αποτελέσματα κ.α. Η ανάλυση αυτών των δεδομένων είναι μια σύνθετη και δύσκολη διαδικασία που απαιτεί μεθόδους εξόρυξης δεδομένων για την εξαγωγή από τις πιο χρήσιμες και κρίσιμες - ανά - υπόθεση πληροφορίες. Στο προτεινόμενο σύστημα υποστήριξης αποφάσεων η βάση δεδομένων χρησιμοποιείται κυρίως για να ερμηνευθούν τα πρότυπα και οι ενώσεις σε σεί των δεδομένων, έτσι ώστε να βελτιωθεί η πρόσβαση σε κρίσιμα δεδομένα ενώ τα πρότυπα της πρόβλεψης θα πρέπει να εφαρμόζονται στην κατασκευή των μοντέλων αποφάσεων, στην υποστήριξη της καλλιέργειας και στον προγραμματισμό των επόμενων βημάτων.

Το προτεινόμενο σύστημα παρουσιάζει σημαντική πρωτοτυπία και παρέχει μια σημαντική καινοτομία σε ότι αφορά τις υπάρχουσες πρακτικές. Οι κυριότερες καινοτομίες είναι οι ακόλουθες:

1. Ανάπτυξη μιας συγκεκριμένης γεωργικής γνώσης βασισμένη σε διεθνή δεδομένα.
2. Ολιστική διαχείριση της διαχείρισης της παραγωγής. Δυνατότητα να χρησιμοποιηθούν παράμετροι που εξασφαλίζουν μια πιο ακριβή και ορθή γεωργική πρακτική, βασισμένη σε μια διαδικασία καταγραφής δεδομένων.
3. Δόμηση της αλληλεπίδρασης παραγωγού – ειδικού ακόμα και εν κινήσει. Συλλογή δεδομένων και χρήση μεθόδων εξόρυξης δεδομένων που μπορούν να χρησιμοποιηθούν περαιτέρω για την πρόσβαση και τη λήψη απόφασης, συνεχή παροχή ενδείξεων καθώς και μια αναλυτική απογραφή του πληθυσμού των

προϊόντων και των χαρακτηριστικών του, προκειμένου να σχεδιαστούν και να καθοριστούν όλες οι παρεμβάσεις εξασφάλισης της ποιότητας.

4. Παραγωγή ολοκληρωμένων υπηρεσιών ώστε να είναι προσβάσιμες από διάφορες συσκευές, ως υποστήριξη ανάμεσα στον παραγωγό, τον ειδικό και τον καταναλωτή.

Το προτεινόμενο σύστημα θα ρυθμιστεί άμεσα από το γεωργικό προσωπικό, μέσω πιλοτικών δοκιμών. Αυτό το χαρακτηριστικό θεωρείται απαραίτητο για την παραγωγή αυστηρών πρωτοκόλλων ώστε να παρέχουν πληροφορίες σε επαγγελματίες γεωργίας που είναι παράλληλα και χρήστες smartphones. Το υποσύστημα υποστήριξης αποφάσεων θα διαμορφωθεί έτσι ώστε να παράγονται ειδοποιήσεις που βασίζονται σε πράσινους, κίτρινους και κόκκινους κώδικες ανάλογα με τις διάφορες περιπτώσεις. Το χαρακτηριστικό αυτό δεν καλύπτεται ακόμη από τις τελευταίες τεχνολογίας συστήματα αγροτικής παραγωγής. Το έργο θα παράγει επίσης μια βάση δεδομένων για τη συγκεκριμένη παραγωγή που θα είναι χρήσιμη σε εθνικό επίπεδο..

Τα στοιχεία που συγκεντρώθηκαν για μια συγκεκριμένη παραγωγή βασίζονται σε διεθνή πρωτόκολλα και πρότυπα μεταδεδομένων, έτσι ώστε τα διαγράμματα ροής δεδομένων να καταρτίζονται προς την επιλεγμένη καλλιέργεια ανά περίπτωση. Η παραγωγή του διαγράμματος ροής δεδομένων είναι απαραίτητη για την αποτελεσματική λειτουργία του συστήματος υποστήριξης αποφάσεων. Το διάγραμμα ροής δεδομένων θεωρείται ως η γραφική παράσταση της "ροής" των δεδομένων που προέρχονται από εξωτερικούς φορείς (π.χ. τα εργαλεία του παραγωγού) και εφαρμόζονται στο σύστημα πληροφοριών. Παρουσιάζει τον τρόπο με τον οποίο τα δεδομένα που εισάγονται επηρεάζουν τις αποφάσεις του συνολικού συστήματος (Ναι ή Όχι, οι τιμές Min / Max σε κρίσιμες παράμετροι, κ.λπ.).

Επιπλέον, δεδομένου ότι συλλέγει και αναλύει με τεχνικές εξόρυξης δεδομένων ένα τεράστιο ποσό δεδομένων, επιτρέπει τη διερεύνηση αιτίας αποτελέσματος ανάμεσα στο τρόπο καλλιέργειας και την τελική ποιότητα του προϊόντος. Από την άλλη πλευρά, σύμφωνα με την παρατήρηση των δεδομένων που προκύπτουν από χιλιάδες προϊόντα, επισημαίνονται φαινόμενα που διαφορετικά δεν θα ήταν αναγνωρίσιμα, όπως για παράδειγμα ο αντίκτυπος ενός βήματος βάσει του προτύπου AGRO σε μια συγκεκριμένη γεωγραφική περιοχή σε σχέση με μια διαφορετική περιοχή, προκειμένου να επισημανθούν ενέργειες βελτίωσης και ως εκ τούτου, να εξελιχθούν και άλλο τα διεθνή πρότυπα.

## Τεχνολογίες Υλοποίησης

---

Για την υλοποίηση του συστήματος αξιοποιήθηκαν τρία δημοφιλή εργαλεία τα οποία ανήκουν στην κατηγορία του λογισμικού ανοικτού κώδικα και συγκεκριμένα η γλώσσα PHP για την κατασκευή των τελικών υπηρεσιών και εφαρμογών του συστήματος, η MySQL για την κατασκευή και διαχείριση των βάσεων δεδομένων και ο εξυπηρετητής διαδικτύου Apache για την εγκατάσταση του εξυπηρετητή στον οποίο λειτουργεί το τελικό σύστημα.

Ο εξυπηρετητής διαδικτύου έχει την ικανότητα να εκτελεί και να παρουσιάζει δυναμικές σελίδες γραμμένες σε γλώσσα προγραμματισμού PHP, ενώ ταυτόχρονα μπορεί να συνδεθεί με μια βάση δεδομένων MySQL για την ανάκτηση δεδομένων καθώς και την αναζήτηση σ' αυτά.

Το βασικό πλεονέκτημα της χρήσης του τεχνολογικού αυτού τρίπτυχου είναι η αποδοτική λειτουργία τους ανεξαρτήτως πλατφόρμας και λειτουργικού συστήματος. Η λειτουργία είναι το ίδιο αποδοτική σε όλα τα υπάρχοντα και δημοφιλή λειτουργικά συστήματα (π.χ. Windows, Linux, IOS ).

## Λογισμικό Ανοικτού Κώδικα

---

### Εισαγωγή

---

Το λογισμικό ανοικτού κώδικα (OpenSource software) είναι ένα μοντέλο διανομής λογισμικού, του οποίου η διανομή γίνεται μαζί με τον πηγαίο κώδικα (source code) και ο οποιοσδήποτε μπορεί να επέμβει σε αυτόν και να προσαρμόσει το λογισμικό στις δικές του ανάγκες. Η διανομή του λογισμικού γίνεται κάτω από τους όρους συγκεκριμένης αδειάς, οι οποίοι ορίζουν τις χρήσεις στις οποίες μπορεί να χρησιμοποιηθεί το λογισμικό, τον τρόπο διανομής, καθώς και εάν θα περιλαμβάνεται στη διανομή ο πηγαίος κώδικας ή όχι. Οι πιο διαδεδομένες άδειες είναι οι BSD και MIT. Η άδεια χρήσης BSD, επιτρέπει τις αλλαγές χωρίς αναδιανομή του πηγαίου κώδικα.

Το λογισμικό ανοικτού κώδικα ξεκινάει συνήθως από κάποιον (φορέα ή ιδιώτη) ο οποίος γράφει την αρχική έκδοση του προγράμματος και στη συνέχεια αποφασίζει να το διαθέσει με το συγκεκριμένο μοντέλο. Υπάρχουν και περιπτώσεις εμπορικού λογισμικού το οποίο στην πορεία διατίθεται από τον κατασκευαστή του με το μοντέλο του λογισμικού ανοικτού κώδικα.

Ένας άλλος όρος που χρησιμοποιείται είναι το "ελεύθερο λογισμικό" που προέρχεται από τον αγγλικό όρο "free software", με τη λέξη "free" να μην έχει την έννοια του "δωρεάν" αλλά την έννοια του "ελεύθερου". Επομένως, το ελεύθερο λογισμικό διανέμεται με τέτοιο τρόπο ώστε ο χρήστης του να μπορεί να χρησιμοποιεί

το πρόγραμμα για κάθε σκοπό, αλλά και να το προσαρμόσει στις ανάγκες του (το τελευταίο υπονοεί ότι ο χρήστης έχει πρόσβαση στον πηγαίο κώδικα). Επίσης, είναι ελεύθερος να αναδιανέμει αντίγραφα του λογισμικού σε τρίτους, με την προϋπόθεση ότι η διανομή γίνεται στα πλαίσια της αδείας. Η αναδιανομή σε τρίτους μπορεί να γίνεται μαζί με τις όποιες βελτιώσεις επιφέρει ένας χρήστης στο λογισμικό.

Οι παραπάνω όροι είναι γενικά παρόμοιοι. Το ελεύθερο λογισμικό δίνει έμφαση στην ελευθερία του λογισμικού, ώστε να παραχθεί λογισμικό που θα λειτουργήσει ως κοινωνικό και δημόσιο αγαθό. Το λογισμικό ανοικτού κώδικα από την πλευρά του, αναζητά ανθρώπους που θα βοηθήσουν στην ανάπτυξη και βελτίωση του λογισμικού και ως ανταμοιβή -πέραν της ηθικής ικανοποίησης- θα έχουν το δικαίωμα της χρήσης του λογισμικού.

### *Συμμετοχή στην ανάπτυξη λογισμικού Ανοικτού κώδικα*

---

Η ανάπτυξη λογισμικού με τη φιλοσοφία του opensource δεν αποσκοπεί γενικά στο κέρδος, χωρίς αυτό να σημαίνει πως οι δύο έννοιες δεν μπορούν να συνυπάρχουν. Υπάρχουν αρκετοί λόγοι για να συμμετάσχει κάποιος αφιλοκερδώς στην ανάπτυξη ελεύθερου λογισμικού ή λογισμικού ανοικτού κώδικα, τους οποίους λόγους κάποιος μπορεί να έχει, είτε μεμονωμένα είτε σε συνδυασμό. Οι βασικότεροι από αυτούς είναι:

- Η ταχύτερη ανάπτυξη ενός λογισμικού το οποίο θα βοηθήσει και τον ίδιο τον συμμετέχοντα όταν αυτό ολοκληρωθεί να το χρησιμοποιήσει και ο ίδιος.
- Η απόκτηση εμπειρίας από την ανάπτυξη ενός λογισμικού, καθώς και η συνεργασία με άλλους προγραμματιστές.
- Η απόκτηση φήμης εάν το έργο πετύχει και γίνει γνωστό.
- Η ηθική ικανοποίηση.
- Το κέρδος, εφόσον είναι επιθυμητό, μπορεί να προκύψει τόσο από την πώληση του λογισμικού σε binary form (έτοιμου προς χρήση δηλαδή) μαζί με τον πηγαίο κώδικα, όσο και από την πώληση υπηρεσιών που σχετίζονται με το λογισμικό αυτό.

### *Πλεονεκτήματα χρήσης λογισμικού Ανοικτού Κώδικα*

---

- Η λογική της ανάπτυξής του είναι τέτοια ώστε επιτρέπει τον ποιοτικό του έλεγχο από πολλούς ερευνητές και προγραμματιστές οι οποίοι ενδέχεται να είναι ικανότατοι προγραμματιστές και πολύ εξειδικευμένοι.
- Υπάρχει τεράστια δυνατότητα προσαρμογής του λογισμικού στις ανάγκες ιδιωτών ή και εταιριών.
- Μπορεί να αποτελέσει σημαντικό εκπαιδευτικό εργαλείο ή εργαλείο για απόκτηση προγραμματιστικής εμπειρίας από αυτούς που αναπτύσσουν κώδικα.

- Το κόστος (χρήση + απόκτηση) του ελεύθερου λογισμικού ή του λογισμικού ανοικτού κώδικα είναι συνήθως σημαντικά μικρότερο από το κόστος αντίστοιχων εμπορικών λύσεων.
- Όταν πρόκειται για δημοφιλή προγράμματα τα οποία χρησιμοποιούνται σε πληθώρα εγκαταστάσεων ανά τον κόσμο, η υποστήριξη σε περίπτωση εμφάνισης προβλημάτων μπορεί να προέλθει άμεσα, με τη χρήση των καναλιών επικοινωνίας του internet (π.χ. newsgroups).
- Η χρήση ελεύθερου λογισμικού ή λογισμικού ανοικτού κώδικα δε δημιουργεί εξαρτήσεις από κάποια συγκεκριμένη εταιρία.
- Ο κώδικας είναι διαθέσιμος και μπορεί να ελεγχθεί η αξιοπιστία του, κάτι που δεν μπορεί να γίνει σε εμπορικά προγράμματα όπου ο κώδικας δεν είναι διαθέσιμος.

### *Μειονεκτήματα χρήσης λογισμικού Ανοικτού Κώδικα*

---

- Δεν αποτελούν στο σύνολό τους λύσεις σταθερές στη λειτουργία τους. Συνήθως το λογισμικό υποδομής, το οποίο χρησιμοποιείται από τους πολλούς και άρα έχουν συμφέρον να είναι ισχυρό, είναι αρκετά σταθερό. Αντίθετα κάποια επιμέρους προγράμματα τα οποία ενδιαφέρουν λίγους, πιθανώς να μην είναι το ίδιο σταθερά με τα αντίστοιχα εμπορικά.
- Η παρεχόμενη τεκμηρίωση είναι συνήθως μικρή, με εξαίρεση τα βασικά προγράμματα υποδομής (π.χ. λειτουργικά συστήματα, δικτυακές υπηρεσίες).
- Η υποστήριξη γίνεται συνήθως από την ίδια την κοινότητα ανάπτυξης και χρήσης του λογισμικού, που σημαίνει ότι δεν υπάρχει κάποια επαγγελματική δέσμευση. Για λογισμικά που χρησιμοποιούνται ευρέως, υπάρχουν και επαγγελματίες που αναλαμβάνουν τη σύναψη συμβολαίων υποστήριξης (π.χ. Linux).
- Χρειάζεται να περάσουν αρκετές εκδόσεις του λογισμικού ώστε να ωριμάσει και να σταθεροποιηθεί, έτσι ώστε αν κάποιος το υιοθετήσει στις πρώτες του εκδόσεις, ενδεχομένως να αντιμετωπίσει αρκετά προβλήματα.
- Δεν είναι ελεγχόμενη η ανάπτυξή του. Ο καθένας προσθέτει τα χαρακτηριστικά που χρειάζεται. Έτσι, εάν κάτι δεν συμπεριλαμβάνεται στο λογισμικό, θα πρέπει να το αναπτύξει ο ίδιος ο ενδιαφερόμενος.

Για αρκετούς το ελεύθερο λογισμικό και το λογισμικό ανοικτού κώδικα θεωρείται ως ένα προϊόν το οποίο προορίζεται για ειδικούς και πεπειραμένους χρήστες και δεν είναι κατάλληλο για το συνηθισμένο χρήστη. Στη θέση αυτή υπάρχουν και αλήθειες και ανακρίβειες. Όπως αναφέρθηκε παραπάνω, το ελεύθερο λογισμικό και το λογισμικό ανοικτού κώδικα στις αρχικές του εκδόσεις δεν είναι αρκετά σταθερό αλλά βελτιώνεται στην πορεία. Επίσης, επειδή η ανάπτυξή του ξεκινά από κάποια συγκεκριμένη ανάγκη που διαμορφώνεται και συμπληρώνεται στην πορεία και καθώς αυξάνεται ο αριθμός των συμμετεχόντων στην ανάπτυξή του, οι αρχικές εκδόσεις μπορεί να μην καλύπτουν το σύνολο των αναγκών, όπως ένα εμπορικό λογισμικό. Αφότου περάσουν τα πρώτα στάδια της ανάπτυξης και δοκιμαστεί το λογισμικό σε διάφορες εγκαταστάσεις από τις οποίες θα προκύψουν παρατηρήσεις και βελτιώσεις, σταδιακά δημιουργούνται σταθερότερες λύσεις. Το λογισμικό υποδομής το οποίο

είναι αρκετά διαδεδομένο και χρησιμοποιείται σε πολλές εγκαταστάσεις όπως λειτουργικά συστήματα (Linux) και λογισμικό για servers (Apache Web Server), είναι εξίσου λειτουργικό και σταθερό και ορισμένες φορές σταθερότερο από αντίστοιχες εμπορικές λύσεις.

Τελευταία μεγάλοι κατασκευαστές υποστηρίζουν πλέον λογισμικό που έχει αναπτυχθεί ως ελεύθερο λογισμικό ή λογισμικό ανοικτού κώδικα, όπως το Linux και διάφορες εκδόσεις για εξυπηρετητές. Ο Βασικότερος λόγος είναι το μικρότερο κόστος σε σχέση με αντίστοιχες εμπορικές λύσεις. Εάν μία επιχείρηση μπορεί να καλύψει συγκεκριμένες της ανάγκες με τέτοιου τύπου λογισμικά, έχει όφελος τουλάχιστον να εξετάσει το ενδεχόμενο να τα χρησιμοποιήσει σε συγκεκριμένες εγκαταστάσεις (π.χ. εγκατάσταση ενός email Server, χρήση μιας βάσης δεδομένων) και αν επιθυμεί να κάνει μία δοκιμαστική δραστηριότητα, με τη χρήση ελεύθερου λογισμικού μπορεί να δοκιμάσει ελεύθερα, χωρίς να χρειαστεί να αγοράσει άδειες.

Τέλος, για το λόγο ότι το ελεύθερο λογισμικό και το λογισμικό ανοικτού κώδικα περιλαμβάνουν και τον πηγαίο κώδικα, ένα οργανισμός, μία εταιρία ή ένας ιδιώτης μπορούν να προσαρμόσουν το λογισμικό στις ιδιαίτερες ανάγκες τους, χωρίς να συμβιβάζονται με λύσεις που δεν ικανοποιούν απολύτως ούτε ξεκινώντας τη δημιουργία λογισμικού από το μηδέν. Όλα τα παραπάνω μπορούν να ωφελήσουν εταιρίες κάθε μεγέθους και ιδιαίτερα τις Μικρομεσαίες Επιχειρήσεις και τους δημόσιους οργανισμούς, όπου η δαπάνη αγοράς λογισμικού είναι σημαντικό πρόβλημα, καθώς δεσμεύει κεφάλαια.

## **Γλώσσες προγραμματισμού**

---

### *PHP*

---

Η PHP αποτελεί μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

### *Επεκτάσεις αρχείων και διακομιστές*

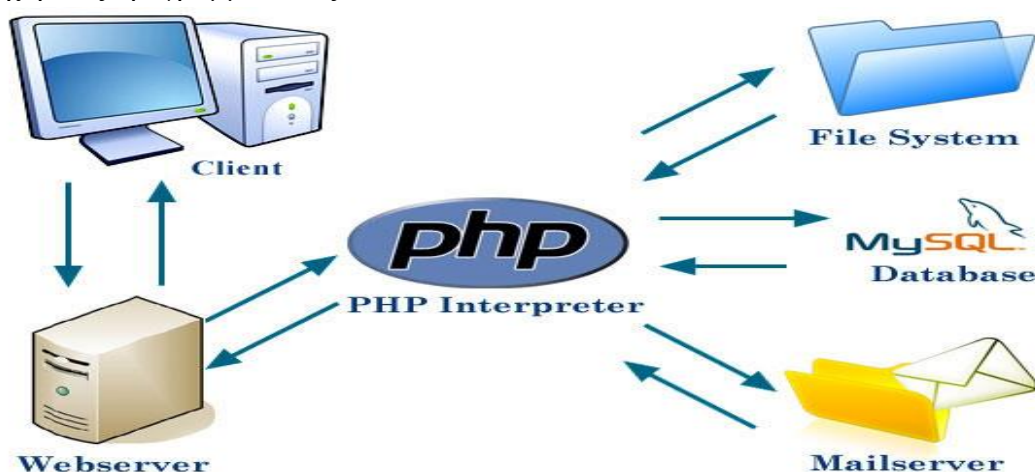
---

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 δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, παρά μόνο εάν έχει γίνει η κατάλληλη ρύθμιση στα MIME (πολλαπλών χρήσεων internet mail) types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται κώδικα PHP. Ο διακομιστής Apache που χρησιμοποιείται σήμερα ευρέως σε συστήματα με



τα λειτουργικά συστήματα GNU/Linux και Microsoft Windows, υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP.

Η PHP είναι μια scripting γλώσσα υψηλού επιπέδου, που σχεδιάστηκε με σκοπό τη δημιουργία δυναμικών ιστοσελίδων. Αν και έχει εξελιχθεί σε μια γλώσσα γενικής χρήσης, η κύρια δύναμη της παραμένει στο χώρο του web development. Πολλά από τα γνωστότερα προγράμματα online forum ή wiki είναι γραμμένα σε PHP και ο συνδυασμός της με τον Apache web server και το σύστημα διαχείρισης βάσης δεδομένων mysql είναι τόσο συχνός, που αναφέρεται συχνά ως LAMP. Η σύνταξη της θυμίζει αρκετά αυτή της C και είναι αρκετά απλή στην εκμάθηση ακόμα και από αρχάριους προγραμματιστές.

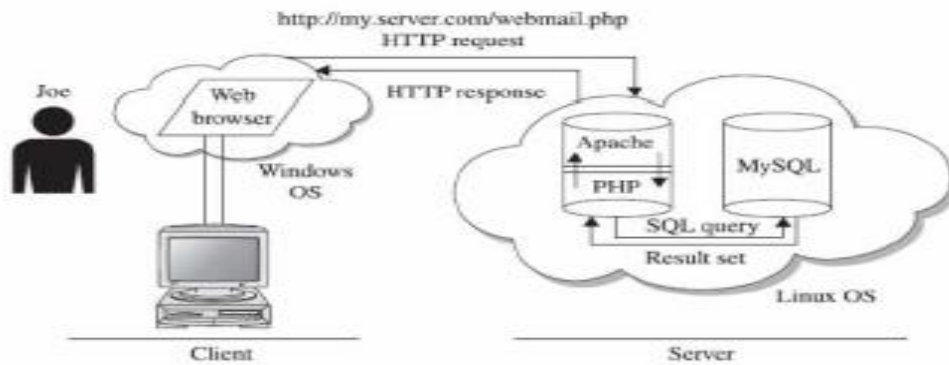


**Εικόνα:** PHP

### *Χρησιμοποίηση PHP*

Σήμερα περισσότερα από 16.000.000 Web Sites, ποσοστό μεγαλύτερο από το 35% των ιστοσελίδων του διαδικτύου, χρησιμοποιούν scripts γραμμένα με τη γλώσσα PHP, ενώ το υπόλοιπο 65% το μοιράζονται στατικές σελίδες HTML και όλες οι άλλες γλώσσες προγραμματισμού. Τα περισσότερα Web Sites επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP. Με τις γλώσσες Perl και C/C++ στις οποίες έχει τις ρίζες της η PHP, έχει εξαιρετική ομοιότητα ως προς τον τρόπο σύνταξης, καθώς και πολλές εντολές της. Η PHP είναι μια γλώσσα "Server-Side". Αυτό σημαίνει ότι ο κώδικας PHP που περιέχει μια σελίδα εκτελείται στον Server (όπου είναι αποθηκευμένη η σελίδα), ενώ τα αποτελέσματα εμφανίζονται με μορφή HTML στον τελικό χρήστη.

Η εικόνα που ακολουθεί παρουσιάζει αναλυτικά τον τρόπο με τον οποίο δουλεύει ένας Web Server (απαραίτητο λογισμικό για την επεξεργασία και τη λειτουργία μιας ιστοσελίδας) στον οποίο υπάρχει εγκατεστημένη η PHP. Ο χρήστης "καλεί" μια σελίδα και ο server κάνει τις αντίστοιχες διεργασίες, για να παρουσιάσει το επιθυμητό αποτέλεσμα πίσω στο χρήστη. Μια απλή σελίδα HTML παρακάμπτει το εγκατεστημένο λογισμικό της PHP στον Web Server και εμφανίζεται όπως ακριβώς είναι στο χρήστη.



**Εικόνα:** Λογισμικό της PHP στον Web Server

### Εκτέλεση αρχείων php

Για να είναι ορατό το αποτέλεσμα μιας εργασίας πρέπει το php αρχείο το οποίο δημιουργήθηκε να αποθηκευθεί στον φάκελο htdocs του Apache και στη συνέχεια στην γραμμή διεύθυνσης του browser πρέπει να δοθεί το όνομα του αρχείου με την απαραίτητη προσθήκη του localhost (π.χ. για ένα αρχείο MyFile.php θα γράψουμε localhost/MyFile.php )

### Παράδειγμα κώδικα PHP

Ο PHP κώδικας περικλείεται από τα tags `<?php` και `?>`  
 Την απεικόνιση κειμένου αναλαμβάνουν οι εντολές `print` ή `echo`.  
 πχ.

```

1 <html>
2 <head>
3   <title>PHP Test</title>
4 </head>
5 <body>
6
7   <?php echo '<p>Hello World</p>'; ?>
8
9 </body>
10 </html>|
  
```

**Εικόνα:** Παράδειγμα κώδικα PHP

### MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) το οποίο μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από

μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, σήμερα θυγατρική της Sun Microsystems. Η βάση δεδομένων MySQL έχει γίνει η πιο δημοφιλής βάση δεδομένων ανοικτού κώδικα διότι παρέχει αυξημένη απόδοση, υψηλή αξιοπιστία και ευκολία στην διαχείριση της. Χρησιμοποιείται σε περισσότερες από 6 εκατομμύρια εγκαταστάσεις κλιμακούμενες από μεγάλες εταιρίες μέχρι εξειδικευμένες εφαρμογές με ενσωματωμένες βάσεις δεδομένων σε όλο τον κόσμο, ακόμα και στην Ανταρκτική.

Η MySQL είναι η πιο δημοφιλής βάση δεδομένων ανοικτού κώδικα και καθίσταται η απόλυτη επιλογή μιας νέας γενιάς εφαρμογών που υλοποιούνται σε περιβάλλον LAMP (Linux, Apache, MySQL, PHP / Perl / Python). Η MySQL λειτουργεί σε περισσότερες από 20 πλατφόρμες συμπεριλαμβανομένων των Linux, Windows, OS/X, HP-UX, AIX, Netware. Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι. Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων. Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιοκτητών συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle. Μια βάση δεδομένων εννοείται μία συλλογή δεδομένων οργανωμένη σε συσχετισμένους πίνακες που παρέχει ταυτόχρονα ένα μηχανισμό για ανάγνωση, εγγραφή, τροποποίηση ή και πιο πολύπλοκες διαδικασίες πάνω στα δεδομένα.

Ο σκοπός μιας βάσης δεδομένων είναι η οργανωμένη αποθήκευση πληροφορίας και η δυνατότητα εξαγωγής της πληροφορίας αυτής, ιδίως σε πιο οργανωμένη μορφή, σύμφωνα με ερωτήματα που τίθενται στη σχεσιακή βάση δεδομένων. Τα δεδομένα είναι δυνατόν να αναδιοργανώνονται με πολλούς διαφορετικούς τρόπους, σε νοητούς πίνακες, χωρίς να είναι απαραίτητη η αναδιοργάνωση των φυσικών πινάκων που τα αποθηκεύουν. Τη σχεσιακή βάση δεδομένων επινόησε ο Έντγκαρ Κοντ το 1970.

Οι ερωτήσεις είτε από το χρήστη είτε από το λογισμικό προς τη βάση δεδομένων, γίνονται συνήθως μέσω της διαδεδομένης διαλογικής γλώσσας SQL (Structured Query Language). Εκτελώντας ερωτήματα ο χρήστης (ή το λογισμικό που εκπροσωπεί το χρήστη) είναι δυνατόν, ανάλογα με τα δικαιώματά του, να δημιουργήσει, να μεταβάλλει και να διαγράψει δεδομένα στη βάση, ή να ανασύρει πληροφορίες με σύνθετα κριτήρια αναζήτησης.

### *Βάσεις Δεδομένων*

---

Μια Βάση Δεδομένων (DB,) είναι ένα σύνολο αρχείων με υψηλό βαθμό οργάνωσης τα οποία είναι συνδεδεμένα μεταξύ τους με λογικές σχέσεις, έτσι ώστε να μπορούν να χρησιμοποιούνται από πολλές εφαρμογές και από πολλούς χρήστες ταυτόχρονα. Υπάρχει ένα ειδικό λογισμικό το οποίο μεσολαβεί ανάμεσα στις αρχεία δεδομένων και τις εφαρμογές που χρησιμοποιούν οι χρήστες και αποκαλείται Σύστημα Διαχείρισης Βάσης Δεδομένων (ΣΔΒΔ) ή DBMS (Data Base Management

System). Το ΣΔΒΔ είναι στην ουσία ένα σύνολο από προγράμματα και υπορουτίνες που έχουν να κάνουν με τον χειρισμό της βάσης δεδομένων, όσον αφορά τη δημιουργία, τροποποίηση, διαγραφή στοιχείων, με ελέγχους ασφαλείας κ.ά.

Οι χρήστες των εφαρμογών αντλούν τα στοιχεία που τους ενδιαφέρουν από τη βάση δεδομένων χωρίς να είναι σε θέση να γνωρίζουν με ποιο τρόπο είναι οργανωμένα τα δεδομένα σ' αυτήν. Το ΣΔΒΔ παίζει τον ρόλο του μεσάζοντα ανάμεσα στον χρήστη και τη βάση δεδομένων και μόνο μέσω του ΣΔΒΔ μπορεί ο χρήστης να αντλήσει πληροφορίες από τη βάση δεδομένων. Ένα ΣΔΒΔ μπορεί να είναι εγκατεστημένο σ' έναν μόνο υπολογιστή ή και σ' ένα δίκτυο υπολογιστών και μπορεί να χρησιμοποιείται από έναν χρήστη ή και από πολλούς χρήστες.

Ένα Σύστημα Βάσης Δεδομένων (ΣΒΔ) ή DBS (Data Base System) αποτελείται από το υλικό, το λογισμικό, τη βάση δεδομένων και τους χρήστες. Είναι δηλαδή ένα σύστημα με το οποίο μπορούμε να αποθηκεύσουμε και να αξιοποιήσουμε δεδομένα με τη βοήθεια ηλεκτρονικού υπολογιστή. Το υλικό (hardware) αποτελείται όπως είναι γνωστό από τους ηλεκτρονικούς υπολογιστές, τα περιφερειακά, τους σκληρούς δίσκους καθώς και τις μαγνητικές ταινίες, όπου είναι αποθηκευμένα τα αρχεία της βάσης δεδομένων αλλά και τα προγράμματα που χρησιμοποιούνται για την επεξεργασία τους. Το λογισμικό (software) είναι τα προγράμματα που χρησιμοποιούνται για την επεξεργασία των δεδομένων (στοιχείων) της βάσης δεδομένων. Η βάση δεδομένων (data base) αποτελείται από το σύνολο των αρχείων όπου είναι αποθηκευμένα τα δεδομένα του συστήματος. Τα στοιχεία αυτά μπορεί να βρίσκονται αποθηκευμένα σ' έναν φυσικό υπολογιστή αλλά και σε περισσότερους. Όμως στον χρήστη δίνεται η εντύπωση ότι βρίσκονται συγκεντρωμένα στον ίδιο υπολογιστή. Τα δεδομένα των αρχείων αυτών είναι ενοποιημένα (data integration), δεν υπάρχει πλεονασμός (άσκοπη επανάληψη) δεδομένων και μερισμένα (data sharing), δηλαδή υπάρχει δυνατότητα ταυτόχρονης προσπέλασης των δεδομένων από πολλούς χρήστες. Ο κάθε χρήστης έχει διαφορετικά δικαιώματα και βλέπει διαφορετικό κομμάτι της βάσης δεδομένων, ανάλογα με τον σκοπό για τον οποίο συνδέεται.

### *Η Σχεσιακή Βάση Δεδομένων MySQL*

---

Η σχεσιακή βάση δεδομένων MySQL είναι η πιο διαδεδομένη σχεσιακή βάση ανοιχτού κώδικα.

- Είναι συμβατή με το μεγαλύτερο μέρος του προτύπου SQL-92.
- Είναι διαθέσιμη σε διάφορα λειτουργικά συστήματα.
- Είναι δωρεάν για χρήση σε προϊόντα ανοιχτού κώδικα.
- Είναι γρήγορη και δεν έχει μεγάλες υπολογιστικές απαιτήσεις.

Το standard interface για να δουλέψουμε με τις βάσεις δεδομένων της MySQL είναι να συνδεθούμε με το λογισμικό του MySQL server και να δίνουμε μία εντολή την φορά. Για να κάνουμε αυτήν την σύνδεση με τον server, θα χρειαστούμε το πρόγραμμα πελάτη (client program) της MySQL.

Στο Linux, το πρόγραμμα αποκαλείται `mysql` και βρίσκεται εξ ορισμού στον κατάλογο `/usr/local/mysql/bin`, ενώ στα Windows, το πρόγραμμα αποκαλείται `mysql.exe` και βρίσκεται εξ ορισμού στον κατάλογο `C:\mysql\bin`.

Υπάρχουν δύο τρόποι για να μπορέσουμε να συνδεθούμε με τον MySQL server.

- Ο πρώτος είναι να χρησιμοποιήσουμε το `telnet` για να συνδεθούμε (`log into`) στον server του Web host που μας φιλοξενεί και να δώσουμε την εντολή `mysql` από εκεί.
- Ο δεύτερος είναι να φορτώσουμε (`download`) και να εγκαταστήσουμε το λογισμικό πελάτη (`client software`) της MySQL από το `sitehttp://www.mysql.com/` στον δικό μας υπολογιστή και να το χρησιμοποιήσουμε για να συνδεθούμε με τον MySQL server.

Όποια μέθοδο κι αν επιλέξουμε και όποιο λειτουργικό σύστημα χρησιμοποιούμε, θα καταλήξουμε σε μια γραμμή εντολών (`command line`), έτοιμοι να εκτελέσουμε το πρόγραμμα πελάτη της MySQL για να συνδεθούμε στον MySQL server. Πρέπει να γράψουμε τα εξής :

```
mysql -h <hostname> -u <username> -p
```

Θα πρέπει να αντικαταστήσουμε το `<hostname>` με το όνομα του host ή την IP διεύθυνση του υπολογιστή στον οποίο εκτελείται ο MySQL server. Αν εκτελούμε το πρόγραμμα πελάτη στον ίδιο υπολογιστή με τον server, μπορούμε να παραλείψουμε το τμήμα `-h <hostname>` της εντολής αντί να γράψουμε `-h localhost`, για παράδειγμα. Το `<username>` πρέπει να είναι το δικό μας όνομα χρήστη στην MySQL.

Αν εγκαταστήσαμε εμείς οι ίδιοι τον MySQL server, αυτό θα είναι το `root`, ενώ αν χρησιμοποιούμε τον MySQL server του Web host που μας φιλοξενεί, αυτό θα πρέπει να είναι το όνομα χρήστη της MySQL που μας έχει δοθεί. Το όρισμα `-p` λέει στο πρόγραμμα να ζητήσει από μας τον κωδικό εισόδου (`password`), το οποίο θα συμβεί μόλις δώσουμε την παραπάνω εντολή. Αν έχουμε εγκαταστήσει εμείς οι ίδιοι τον MySQL, αυτό το `password` θα είναι το `root password` που επιλέξαμε εμείς, ενώ αν χρησιμοποιούμε τον MySQL server του Web host που μας φιλοξενεί, αυτό θα πρέπει να είναι το `password` της MySQL που μας έχει δοθεί.

Τώρα, ο MySQL server είναι σε θέση να παρακολουθεί περισσότερες από μία βάσεις δεδομένων, που αυτό σημαίνει ότι Web host μπορεί να στήσει έναν μόνο MySQL server για να χρησιμοποιηθεί από πολλούς από τους συνδρομητές του. Ο MySQL server χρησιμοποιεί την πρώτη βάση δεδομένων, με όνομα `mysql`, για να μπορεί να παρακολουθεί τους χρήστες, τα συνθηματικά τους (`passwords`) καθώς και το τι επιτρέπεται να κάνουν. Θα αφήσουμε για λίγο αυτή την βάση δεδομένων. Η δεύτερη βάση δεδομένων, με όνομα `test` αποτελεί ένα δείγμα βάσης δεδομένων. Η διαδικασία της διαγραφής στην MySQL αποκαλείται `dropping` (απόρριψη).

## SQL

---

Η γλώσσα SQL (`Structured Query Language`) είναι η πιο διαδεδομένη διαλογική γλώσσα ερωταπαντήσεων που χρησιμοποιείται για την επικοινωνία του χρήστη με σχεσιακές ΒΔ. Πρόκειται για μία μη-διαδικαστική γλώσσα τέταρτης

γενιάς, στην οποία ο χρήστης διατυπώνει διάφορα αιτήματα και το ΣΔΒΔ αναλαμβάνει να τα ικανοποιήσει. Η SQL δίνει τη δυνατότητα στο χρήστη να δημιουργήσει, να τροποποιήσει και να ενημερώσει τους πίνακες της βάσης, καθώς και να αναζητήσει πληροφορίες από τη βάση εφαρμόζοντας σύνθετα κριτήρια αναζήτησης. Η γνώση της SQL είναι απαραίτητη ακόμη κι αν χρησιμοποιούμε ένα ΣΔΒΔ με γραφικό περιβάλλον (όπως η Access), καθώς πολλά συστήματα ΒΔ της μορφής πελάτη εξυπηρετητή (client-server) έχουν SQL server.

Η SQL αποτελείται από δύο υποσύνολα, τη DDL και τη DML.

- DDL (Data Definition Language): Γλώσσα ορισμού δεδομένων που αποτελείται από τις εντολές με τις οποίες καθορίζουμε τη λογική οργάνωση των δεδομένων της βάσης, δηλαδή δημιουργούμε τους πίνακες και τις μεταξύ τους σχέσεις.
- DML (Data Manipulation Language): Γλώσσα χειρισμού δεδομένων που αποτελείται από τις εντολές με τις οποίες ενημερώνουμε τα δεδομένα της βάσης και δημιουργούμε ερωτήματα για ανάκληση πληροφοριών από τη βάση.

### *Πρότυπα της SQL*

---

Τα περισσότερα σχεσιακά ΣΔΒΔ (συμπεριλαμβανομένης και της Microsoft Access 2000) χρησιμοποιούν το πρότυπο ANSI / ISO SQL-92, υπάρχει όμως και το νέο πρότυπο ANSI / ISO SQL-99 που χρησιμοποιείται κυρίως από αντικειμενοστρεφή ΣΔΒΔ.

### *SQL Server*

---

Ο SQL Server είναι μια σχεσιακή βάση δεδομένων, η οποία αναπτύσσεται από τη Microsoft. Οι κύριες γλώσσες που χρησιμοποιούνται είναι η T-SQL και η ANSI SQL. Ο SQL Server βγήκε για πρώτη φορά στην αγορά το 1989 σε συνεργασία με την Sybase. Η κύρια μονάδα αποθήκευσης στοιχείων είναι μια βάση δεδομένων, η οποία αποτελείται από μια συλλογή πινάκων και κώδικα. Αναλύουμε περαιτέρω τα παρακάτω θέματα:

- Αποθήκευση
- Ενδιάμεση μνήμη
- Ταυτοχρονισμός
- Ανάκτηση δεδομένων
- CLR
- Υπηρεσίες
- Παραπομπές
- Αποθήκευση

Η κεντρική βάση δεδομένων του SQL υποστηρίζει διαφορετικούς τύπους, συμπεριλαμβανομένων των ακεραίων αριθμών, αριθμών κινητής υποδιαστολής, δεκαδικών, αλφαριθμητικών, Varchar (σειρές χαρακτήρων μεταβλητού μήκους), δυαδικών αριθμών (για τα μη δομημένα δεδομένα), κειμένων (για κείμενα). Επιτρέπει

επίσης καθορισμένους από το χρήστη σύνθετους τύπους δεδομένων (UDTs), δηλαδή τύπους που βασίζονται στους βασικούς τύπους αλλά μπορούν να τροποποιηθούν. Τα στοιχεία στη βάση δεδομένων αποθηκεύονται σε ένα (ή περισσότερα) αρχεία με επέκταση mdf.

Τα δευτεροβάθμια στοιχεία αποθηκεύονται στο αρχείο με επέκταση ndf. Το αρχείο καταγραφής το οποίο περιέχει όλες τις πρόσφατες αλλαγές στη βάση δεδομένων αποθηκεύεται σε αρχείο με επέκταση ldf. Ο χώρος αποθήκευσης που διατίθεται σε μια βάση δεδομένων διαιρείται σε διαδοχικά αριθμημένες σελίδες, κάθε μία από τις οποίες έχει μέγεθος 8 KB.

### *Ενδιάμεση μνήμη*

---

Οι σελίδες αποθηκεύονται στην ενδιάμεση (buffer) μνήμη RAM για να ελαχιστοποιηθεί η μεταφορά δεδομένων προς και από τον σκληρό δίσκο. Οποιαδήποτε σελίδα 8 KB μπορεί να είναι αποθηκευμένη στη μνήμη και το σύνολο όλων των σελίδων που αποθηκεύονται σε μία περίοδο καλείται λανθάνουσα μνήμη (cache).

Το ποσό μνήμης που είναι διαθέσιμο στον κεντρικό διακομιστή SQL αποφασίζει πόσες σελίδες θα εναποθηκευθούν στη λανθάνουσα μνήμη. Ο SQL Server έχει διαφορετικούς και έξυπνους αλγόριθμους για την καλύτερη απόδοση της λανθάνουσας μνήμης.

Ο κεντρικός διακομιστής SQL εξασφαλίζει ότι οποιαδήποτε αλλαγή στα στοιχεία είναι "ACID", δηλ. χρησιμοποιεί τις "συναλλαγές" (transactions) που εξασφαλίζουν ότι οποιαδήποτε λειτουργία είτε ολοκληρώνεται συνολικά είτε αποτυγχάνει συνολικά, αλλά δεν αφήνει ποτέ τη βάση δεδομένων σε μία ενδιάμεση κατάσταση.

Χρησιμοποιώντας τις συναλλαγές αυτές, μια ακολουθία εντολών μπορεί να αποτελέσει μία ομάδα, με την προϋπόθεση ότι είτε όλες οι εντολές θα πετύχουν ή καμία δεν θα πετύχει. Ο κεντρικός διακομιστής SQL επιτρέπει σε πολλούς πελάτες να χρησιμοποιούν την ίδια βάση δεδομένων ταυτόχρονα.

Υπό αυτήν τη μορφή, θα πρέπει να ελέγξει την ταυτόχρονη πρόσβαση στα κοινά στοιχεία, για να εξασφαλίσει την ακεραιότητα των στοιχείων – Π.χ όταν πολλαπλάσιοι πελάτες ενημερώνουν τα ίδια στοιχεία, ή οι πελάτες προσπαθούν να διαβάσουν ένα στοιχείο το οποίο αλλάζεται από έναν άλλο πελάτη.

### *Ταυτοχρονισμός*

---

Ο κεντρικός διακομιστής SQL παρέχει δύο τρόπους ελέγχου του ταυτοχρονισμού: απαισιόδοξος ταυτοχρονισμός και αισιόδοξος ταυτοχρονισμός. Όταν ο απαισιόδοξος έλεγχος ταυτοχρονισμού χρησιμοποιείται, ο κεντρικός διακομιστής SQL ελέγχει την ταυτόχρονη πρόσβαση με τη χρησιμοποίηση κλειδώματος (locks). Τα κλειδώματα μπορούν είτε να διαμοιράζονται είτε να είναι

αποκλειστικά. Το αποκλειστικό κλειδίωμα χορηγεί στο χρήστη την αποκλειστική πρόσβαση στα στοιχεία (συνήθως κατά την αλλαγή στοιχείων) - κανένας άλλος χρήστης δεν μπορεί να έχει πρόσβαση στα στοιχεία εφόσον υπάρχει κλειδίωμα. Το κοινό κλειδίωμα χρησιμοποιείται όταν διαβάζεται κάποιο στοιχείο - πολλαπλοί χρήστες μπορούν να διαβάσουν από τα στοιχεία που κλειδώνονται με ένα κοινό κλειδίωμα, αλλά να μην αποκτούν αποκλειστικό.

Το κλειδίωμα μπορεί να εφαρμοστεί σε διαφορετικά επίπεδα - σε ολόκληρους πίνακες, σελίδες ή ακόμα και σε διαφορετικές γραμμές. Το επίπεδο που χρησιμοποιείται καθορίζεται σε μια βάση δεδομένων από περίπλοκους αλγόριθμους του SQL Server, οι οποίοι έχουν σκοπό την γρηγορότερη και καλύτερη λειτουργία από χιλιάδες χρήστες η ακόμη και εκατοντάδες χιλιάδες την ίδια στιγμή. Ένας SQL Server είναι δυνατόν να πραγματοποιήσει μέχρι και 1.000.000 transactions το δευτερόλεπτο με το ανάλογο υλικό. Ο SQL Server λειτουργεί σε 32 ή 64 bit και υποστηρίζει μέχρι και 256 πυρήνες (CPU) Τυπική χρήση είναι οι 8-16 επεξεργαστές.

### Ανάκτηση δεδομένων

---

Η ερώτηση (query) είναι ο κύριος τρόπος για την ανάκτηση στοιχείων από μια βάση δεδομένων. Η ερώτηση εκφράζεται χρησιμοποιώντας μια παραλλαγή της αποκαλούμενης SQL T-SQL, είναι μια διάλεκτος SQL που αναπτύχθηκε από την Microsoft και Sybase. Η TSQL είναι πολύ κοντά στα ANSI standards που έχουν καθιερωθεί διεθνώς, σε αντιδιαστολή με άλλες διαλέκτους όπως η PL-SQL της Oracle που διαφέρουν περισσότερο από το ANSI standards.

Η ερώτηση διευκρινίζει επεξηγηματικά αυτό που είναι να ανακτηθεί από την βάση δεδομένων. Η ερώτηση υποβάλλεται σε επεξεργασία από τον επεξεργαστή ερώτησης, ο οποίος υπολογίζει τη σειρά των βημάτων που θα είναι απαραίτητα για να ανακτηθούν τα ζητούμενα στοιχεία. Η ακολουθία ενεργειών απαραίτητων να εκτελέσουν μια ερώτηση καλείται σχέδιο ερώτησης.

Υπάρχουν πολλοί τρόποι για να υποβληθεί σε επεξεργασία η ίδια ερώτηση. Π.χ, μια ερώτηση που ενώνει 2 πίνακες A & B μαζί θα μπορούσε να γίνει ή πηγαίνοντας πρώτα στο A βρίσκοντας τα στοιχεία και μετά ενώνοντας με τα στοιχεία του πίνακα B, ή πρώτα από το B και μετά πίνακα A. Ο SQL Server και όχι ο Προγραμματιστής παίρνει την απόφαση μέσω στατιστικών στοιχείων και άλλων στοιχείων που διαθέτει για τους 2 πίνακες να ακολουθήσει την διαδρομή A-B γιατί αυτός είναι ο γρηγορότερος τρόπος. Στις procedural γλώσσες προγραμματισμού πχ C, Pascal, NET, ο προγραμματιστής θα πρέπει να κάνει την απόφαση το οποίο πολλές φορές σημαίνει λάθος απόφαση. Αυτό καλείται βελτιστοποίηση ερώτησης και εκτελείται από τον επεξεργαστή ερώτησης. Ο SQL παίρνει πολύ περίπλοκες αποφάσεις βασισμένος σε αλγόριθμους που έχουν αναπτυχθεί τα τελευταία 20-30 χρόνια πάνω στο αντικείμενο. Μια παρόμοια ερώτηση σε μια προηγούμενη έκδοση του SQL Server θα μπορούσε να πάρει 10-20 φορές περισσότερο χρόνο λόγω του διαφορετικού αλγόριθμου.



Γενικά κάθε καινούργια έκδοση βελτιώνει την αποτελεσματικότητα και άρα τον χρόνο εκτέλεσης. Εδώ ακριβώς είναι ένα από τα πλεονεκτήματα προγραμματισμού ενός προβλήματος σε μια βάση δεδομένων σε σύγκριση με C, Pascal, C++ κτλ. Βέβαια όλα τα προβλήματα δεν είναι κατάλληλα για εφαρμογή σε βάση δεδομένων.

Ο SQL Server επιτρέπει επίσης αποθηκευμένες ερωτήσεις (stored procedures). Οι αποθηκευμένες ερωτήσεις είναι ερωτήσεις με παραμέτρους T-SQL, οι οποίες αποθηκεύονται στον SQL Server. Μια stored procedure τυπικά είναι από 100-1000 γραμμές κώδικα SQL. Οι αποθηκευμένες ερωτήσεις μπορούν να δεχτούν τιμές που στέλνονται από τον πελάτη ως παράμετροι εισαγωγής και να στείλουν τα αποτελέσματα ως παραμέτρους παραγωγής. Μπορούν να καλέσουν functions, και άλλες stored procedures. Οι stored procedures είναι γρηγορότερες από κατασκευή σε σχέση με απλές ακολουθίες SQL.

### CLR

---

Ο Microsoft SQL Server 2005 – 2008 περιλαμβάνει ένα module που λέγεται SQL CLR μέσω του οποίου ενσωματώνει το NET μέσα στον SQL Server. Με το SQL CLR, οι αποθηκευμένες διαδικασίες μπορούν να γραφτούν σε οποιαδήποτε γλώσσα NET συμπεριλαμβανομένου C# και VB.NET και να δημιουργήσουν μια stored procedure. Αυτό σημαίνει ότι ο SQL Server έχει όλες τις βιβλιοθήκες και πλεονεκτήματα του NET, αυτόχθονα μέσα στο περιβάλλον του, τα οποία μπορεί να τα καλέσει οποιαδήποτε στιγμή.

### 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. Αποτελεί υποσύνολο της γλώσσας SGML (Standard Generalized Markup Language) που επινοήθηκε από την IBM προκειμένου να λυθεί το πρόβλημα της μη τυποποιημένης εμφάνισης κειμένων στα διάφορα υπολογιστικά συστήματα. 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. Η html χρησιμοποιεί τις ειδικές ετικέτες (τα tags) να δώσει τις απαραίτητες οδηγίες στον browser.

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> Οι οδηγίες είναι case insensitive, δεν επηρεάζονται από το αν έχουν γραφτεί με πεζά (μικρά) ή κεφαλαία. Ένα αρχείο HTML πρέπει να έχει κατάληξη htm ή html.

Για να μπορούν οι browser να ερμηνεύουν σχεδόν απόλυτα σωστά την html έχουν θεσπιστεί κάποιοι κανόνες. Αυτοί οι κανόνες είναι γνωστοί ως προδιαγραφές.

Επομένως σχεδόν κάθε είδος υπολογιστή μπορεί να δείξει το ίδιο καλά μια ιστοσελίδα. Οι πρώτες προδιαγραφές ήταν η html 2.0. Πρόβλημα προέκυψε όταν η Microsoft και η Netscape πρόσθεσαν στην html τέτοιες δυνατότητες που στην αρχή τουλάχιστον ήταν συμβατές μόνο με συγκεκριμένους browser. Ακόμη και σήμερα υπάρχουν διαφορές στην απεικόνιση κάποιας σελίδας από διαφορετικούς browsers. Ιδιαίτερο είναι το πρόβλημα όταν η ιστοσελίδα, εκτός από "καθαρή" HTML περιλαμβάνει και εφαρμογές Javascript.

## *Η HTML σήμερα*

---

Το 1990 ο Tim Berners-Lee από το Cern, το εργαστήριο φυσικής της Γενεύης, δημιούργησε ένα νέο πρωτόκολλο με το οποίο θα μπορούσαν να μεταφέρονται κάθε είδος αρχείων και αντικειμένων μέσα από το Internet. Το πρωτόκολλο αυτό ονομάστηκε HTTP (HyperText Transfer Protocol) και σηματοδότησε την αρχή του WWW όπως το ξέρουμε σήμερα. Οι σελίδες που ήταν η βάση του WWW ήταν γραμμένες στην πρώτη έκδοση της γλώσσα HTML. Το 1994 αναπτύσσεται το πρότυπο HTML 2.0 από ένα διεθνή οργανισμό (Internet Engineering Task Force). Η επόμενη έκδοση η 3.0 δεν έγινε αποδεκτή από τις Microsoft και Netscape οπότε γρήγορα αντικαταστάθηκε από την έκδοση 3.2 (1996). Η τελευταία περιελάμβανε πολλές από τις σημάνσεις (tags) που είχαν εισάγει οι δύο εταιρίες. Η έκδοση 4.0 παρουσιάστηκε τον Ιούνιο του 1997. Σήμερα πολλοί είναι εκείνοι που δημιουργούν μια ιστοσελίδα σε κάποιο πρόγραμμα που επιτρέπει την δημιουργία χωρίς την συγγραφή κώδικα. Η κοινή άποψη πάνω στο θέμα όμως είναι ότι κάτι τέτοιο είναι αρνητικό επειδή ο δημιουργός δεν έχει τον απόλυτο έλεγχο του κώδικα με αποτέλεσμα πολλές φορές να υπάρχει οπτικό χάος στην προσπάθεια των browser να εμφανίσουν την ιστοσελίδα. Για το σκοπό αυτό έχει δημιουργηθεί ειδικό λογισμικό, που επιτρέπει το "στήσιμο" της σελίδας οπτικά, χωρίς τη συγγραφή κώδικα, δίνει όμως τη δυνατότητα παρέμβασης και στον κώδικα. Χαρακτηριστικό παράδειγμα το λογισμικό Dreamweaver της Adobe και το FrontPage της Microsoft.

## *Εντολές HTML – Παραδείγματα*

---

`<head>` Όπως έχει προαναφερθεί η εντολή αυτή περιέχει τον τίτλο και άλλες σημαντικές πληροφορίες για το έγγραφο.

`<title>` Μέσα σε αυτήν την εντολή περικλείεται ο τίτλος του εγγράφου. Ο τίτλος εμφανίζεται στην μπάρα του browser (Netscape, Explorer κ.α)

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

`<h1>` Η εντολή αυτή καθορίζει το μέγεθος των γραμμάτων. Ο αριθμός μπορεί να ανέλθει ως το 6 (δηλαδή `<h2>`, `<h3>` ?). Το `<h1>` είναι το μεγαλύτερο μέγεθος και το `<h6>` είναι το μικρότερο.

`<br>` Η εντολή `<br>` αναγκάζει το κείμενο (και όλο το υπόλοιπο υλικό που περιέχει η σελίδα) να συνεχίσει στη επόμενη σειρά. (Το `</br>` δεν έχει νόημα).

<center> Από τη στιγμή που εμφανίζεται το <center> σ' ένα έγγραφο μέχρι να κλείσει με </center> όλα όσα περιέχονται εμφανίζονται κεντραρισμένα

<b> Η εντολή αυτή από την εμφάνισή της και μέχρι την λήξη της (δηλ. </b>) 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). Κάθε HTML έγγραφο πρέπει να περιέχει συγκεκριμένες εντολές. Κάθε έγγραφο αποτελείται από ένα κείμενο επικεφαλίδα (head) και το κυρίως μέρος (body). Η επικεφαλίδα περιέχει τον τίτλο και το κυρίως μέρος το κείμενο που θα φαίνεται στη σελίδα που μπορεί να περιέχει παραγράφους, πίνακες, λίστες κ.α. Ένα μικρό παράδειγμα εγγράφου HTML είναι το ακόλουθο.

### Παράδειγμα 1

```
<html>
<head>
<title>Ένα απλό παράδειγμα</title>
</head>
<body>
<h1> HTML </h1>
<p> Αυτή είναι η πρώτη παράγραφος </p>
<p> Αυτή είναι η δεύτερη παράγραφος</p>
</body>
</html>
```

Ένα οποιοδήποτε κείμενο αποτελείται από επιμέρους στοιχεία όπως πίνακες παραγράφους, λίστες κλπ. Σε ένα HTML κείμενο πρέπει να μαρκάρεις την αρχή και το τέλος κάθε τέτοιου στοιχείου γράφοντας την κατάλληλη εντολή. Κάθε εντολή αρχίζει με το σύμβολο < και τελειώνει με το > δηλαδή μια εντολή είναι κάπως έτσι <εντολή> . Για να δηλώσουμε το τέλος της ισχύς μιας εντολής προσθέτουμε στην εντολή μια κάθετο, δηλαδή </εντολή> .

### Σύνδεσμοι (links)

Ως σύνδεσμο (link) στην HTML ορίζουμε ότι μπορεί να επιλεγεί (κάνοντας click) 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 </a>) ότι περιέχεται γίνεται αυτόματα σύνδεσμος και εμφανίζεται διαφορετικά από τον browser (συνήθως υπογραμμισμένο και με μπλε γράμματα). Σημαντικός παράμετρος είναι η href="ΌνομαΝεουΕγγραφου" όπου καθορίζει σε ποιο έγγραφο είναι δείκτης ο σύνδεσμος.

Παραδείγματα χρήσης είναι τα ακόλουθα :

<a href="katalogos.html">κατάλογος</a>

<a href="photos/photogr.html">φωτογραφία</a>

<a href="http://www.physics.uch.gr/entupo.html">έντυπο</a>

Ακόμα σύνδεσμος μπορεί να είναι και μια εικόνα :

<a href="kat.html"></a> Τοποθέτηση σελίδων σε Server

## CSS

---

Η CSS (Cascading Style Sheets-διαδοχικά Φύλλα Στυλ) ή ( αλληλουχία φύλλων στυλ) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Πιο πρακτικά χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα του υπολογιστή βασισμένη για να αναπτύξει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώσει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει πάρα πολλές δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η γλώσσα CSS είναι απαραίτητη.

Τα CSS (Cascading Style Sheets), στην κατασκευή ιστοσελίδων, είναι ηλεκτρονικά έγγραφα με ένα σύνολο κανόνων για την μορφοποίηση μιας ιστοσελίδας. Ενδεικτικά μπορεί να χρησιμοποιηθεί για να διαμορφώσει:

- Τα χρώματα και το μέγεθος διαφόρων στοιχείων της ιστοσελίδας.
- Την συμπεριφορά τους σε διάφορες ενέργειες.

Η μεγάλη ευκολία που προσφέρει η χρήση ενός CSS είναι ότι οι ενδεχόμενες αλλαγές γίνονται μόνο σε ένα έγγραφο και αυτόματα εφαρμόζονται σε όλες τις σελίδες που το χρησιμοποιούν. Έτσι μία ιστοσελίδα που χρησιμοποιεί CSS, μπορεί πολύ ευκολότερα να αλλάξει όψη.

Για ένα έγγραφο πχ xhtml θα υπάρχουν παραπάνω από ένα φύλλα στυλ τα οποία περιέχουν δηλώσεις για την εμφάνιση ενός συγκεκριμένου στοιχείου. Το φύλλο στυλ που εφαρμόζεται σε ένα έγγραφο μπορεί να προέρχεται από :

- το συγγραφέα μιας ιστοσελίδας.
- το χρήστη του πλοηγού.
- τον ίδιο τον πλοηγό, αν έχει το δικό του προκαθορισμένο φύλλο στυλ

Συνεπώς για ένα xhtml στοιχείο θα υπάρχουν παραπάνω από μια δηλώσεις που πιθανώς να είναι συγκρουόμενες. Το πρότυπο CSS για να επιλύσει παρόμοιες συγκρούσεις έχει καθορίσει μια αλληλουχία-σειρά στην οποία θα μπουν αυτές οι δηλώσεις και με βάση την οποία θα επιλεγεί π.χ. η δήλωση που είναι πρώτη στη σειρά.

Ο αλγόριθμος δημιουργίας αυτής της σειράς-αλληλουχίας είναι ο ακόλουθος: Βρίσκονται όλες οι δηλώσεις που εφαρμόζονται στο στοιχείο που μας ενδιαφέρει. Οι δηλώσεις εφαρμόζονται στο στοιχείο αν ο επιλογέας του το επιλέγει (ταιριάζει με αυτό). Ταξινομούμε με βάση τη σημασία (κανονική ή σημαντική) και προέλευση ( συγγραφέας , χρήστη ή πλοηγός χρήστη). Σε αυξάνουσα σειρά προτεραιότητας:

- Δηλώσεις πλοηγού χρήστη.
- Κανονικές δηλώσεις χρήστη.
- Κανονικές δηλώσεις συγγραφέα.
- Σημαντικές δηλώσεις συγγραφέα.
- Σημαντικές δηλώσεις χρήστη.

Ταξινομούνται οι δηλώσεις τις ίδια σημασίας και προέλευσης με κριτήριο την εξειδίκευση του επιλογέα: οι πιο εξειδικευμένοι επιλογείς υπερισχύουν τον ποιών γενικών. Τα ψευδό-στοιχεία και οι ψευδο-κλάσεις λογαριάζονται σαν κανονικά στοιχεία και κλάσεις αντίστοιχα. Τέλος ταξινομούνται ανάλογα με τη σειρά καθορισμού: αν δύο δηλώσεις έχουν το ίδιο βάρος την ίδια προέλευση και εξειδίκευση, επικρατεί αυτή που προσδιορίστηκε τελευταία. Οι δηλώσεις σε εισαγόμενα φύλλα στυλ θεωρούνται ότι δηλώνονται πριν από τις δηλώσεις στο ίδιο το φύλλο στυλ. Αφού λοιπόν προκύψει μια σειρά-αλληλουχία κανόνων εμφάνισης που αφορούν το ίδιο στοιχείο θα επιλεγεί προς εφαρμογή (για την αποφυγή συγκρούσεων) η δήλωση που θα είναι τελευταία στην σειρά που αναλύσαμε μόλις.

## *Τι είναι η CSS*

---

Κατά την διάρκεια γραφής του HTML κώδικα, χρειάζεται να “ντύνουμε” τον κώδικα με κάποια στοιχεία τα οποία θα βοηθήσουν να δώσουμε στυλ και να μορφοποιήσουμε όπως θέλουμε τον κώδικα χρησιμοποιώντας CSS. Τα στοιχεία αυτά τα οποία είναι γνωστά ως selectors (επιλογείς), χωρίζονται σε δύο κατηγορίες. Στα Ids και στα Classes.

## *Σύνταξη Των Στοιχείων Id*

---

Τα Ids είναι μοναδικά στοιχεία που προσθέτουμε σε ένα div στον HTML κώδικα. Ονομάζονται μοναδικά διότι δηλώνουν στοιχεία που θα εμφανιστούν μια και μοναδική φορά σε μία σελίδα. Για παράδειγμα:

```
<div id="logo">
<a href="/home.html"></a>
</div>
```

Το στοιχείο id=logo θα εμφανίζεται μόνο μια φορά στην σελίδα και αυτό είναι στην κορυφή. Έτσι καλό είναι να του δώσουμε Id και όχι Class. Για να δώσουμε στυλ στους Id selectors όταν γράφουμε CSS, χρησιμοποιούμε την δέση (#). Έτσι για παράδειγμα, για το στοιχείο id=logo θα γράψουμε:

```
#logo { margin-left:10px; position:relative;
```

## *Σύνταξη Των Στοιχείων Class*

---

Τα Classes χρησιμοποιούνται διαφορετικά από τα Ids και συνήθως τα εφαρμόζουμε σε στοιχεία τα οποία θέλουμε να κατηγοριοποιήσουμε και να κατατάξουμε. Επομένως το χρησιμοποιούμε σε στοιχεία στα οποία θέλουμε να έχουν όλα την ίδια μορφοποίηση, ενώ θα εμφανίζονται όσες φορές θέλουμε στην σελίδα μας. Για παράδειγμα έχουμε:

```
<p class="description"> </p>
```

Δίνοντας στο στοιχείο της παραγράφου παραπάνω την κλάση description, ουσιαστικά κατηγοριοποιούμε αυτού του είδους την παράγραφο να έχει συγκεκριμένες ιδιότητες διαφορετικές από τις υπόλοιπες παραγράφους της σελίδας μας. Για να δώσουμε στυλ στους Class selectors όταν γράφουμε CSS, χρησιμοποιούμε την τελεία (.). Για παράδειγμα για το στοιχείο class=description γράφουμε:

```
p.description { font-size:12px; color:#c9c9c9; }
```

### *Σύνδεση CSS Με HTML*

---

Αφού αναλύθηκε ο τρόπος με τον οποίο δίνονται στυλ στα διάφορα στοιχεία στην σελίδα μας, τώρα θα δούμε πως ακριβώς γίνεται η σύνδεση των CSS με το HTML αρχείο μας. Υπάρχουν τρεις τρόποι για να δώσουμε στυλ και μορφοποίηση στα στοιχεία της HTML σελίδας μας.

- Δήλωση Στο Head

Ο πρώτος είναι να γράψουμε όλο τον CSS κώδικα μας στο <head> μέρος της ιστοσελίδας μας.

```
<head>
```

```
<style type="text/css" media="screen, projection">
```

```
</style>
```

- Inline

Ο δεύτερος είναι να γράψουμε τον CSS κώδικα μέσα στο HTML στοιχείο μας. Αυτός ο τρόπος είναι γνωστός ως Inline Styling. Ας δούμε ένα παράδειγμα:

```
<p style="font-size:12px; color:#c9c9c9"> </p>
```

### *Εξωτερικό Αρχείο*

---

Τέλος, ο πιο διαδεδομένος τρόπος είναι να συμπεριλάβουμε στο <head> μέρος της ιστοσελίδας μας ένα νέο εξωτερικό που θα έχει κατάληξη css, στο οποίο θα περιλαμβάνεται όλος ο CSS κώδικας μας. Ας δούμε ένα παράδειγμα:

```
<head>
```

```
<link rel="stylesheet" type="text/css" href="style.css" />
```

```
</head>
```

### *Φύλλα Στυλ*

---

Ο όρος CSS σημαίνει Cascading Style Sheets δηλαδή διαδοχικά Φύλλα Στυλ. Τα στυλ ορίζουν τον τρόπο εμφάνισης των στοιχείων της HTML. Τα Στυλ (Styles) αποθηκεύονται σε Φύλλα Στυλ (Style Sheets). Τα Στυλ προστέθηκαν στην HTML 4.0 για να λύσουν κάποια προβλήματα. Τα Εξωτερικά Φύλλα Στυλ (External Style Sheets) μπορούν να μας γλυτώσουν από πολλή δουλειά.

Τα στυλ στην HTML 4.0 ορίζουν το πώς εμφανίζονται τα HTML στοιχεία, όπως ακριβώς το tag font και το attribute color στην HTML 3.2. Τα στυλ αποθηκεύονται συνήθως σε αρχεία που είναι εξωτερικά (external) στα HTML έγγραφα μας.

Τα εξωτερικά φύλλα στυλ (external style sheets) μάς δίνουν τη δυνατότητα να αλλάξουμε την εμφάνιση και τη διάταξη όλων των σελίδων στο δικό μας Web site, με απλή επεξεργασία ενός μόνου CSS εγγράφου. Αν έχουμε προσπαθήσει να αλλάξουμε τη γραμματοσειρά (font) ή το χρώμα (color) όλων των επικεφαλίδων (headings) σ' όλες τις ιστοσελίδες μας, θα καταλάβουμε γιατί τα CSS μπορούν να μας γλυτώσουν από πολλή δουλειά.

Τα CSS αποτελούν μια μεγάλη επιτυχία στον σχεδιασμό του Web (Web design) επειδή δίνουν τη δυνατότητα στους developers να ελέγξουν το στυλ και τη διάταξη πολλών ιστοσελίδων μονομιάς. Σαν Web developer μπορούμε να ορίσουμε ένα στυλ για κάθε HTML στοιχείο και να το εφαρμόσουμε σ' όσες ιστοσελίδες θέλουμε. Για να κάνουμε μια καθολική (global) αλλαγή, απλά αλλάζουμε το στυλ μία φορά και όλα τα στοιχεία του Web ενημερώνονται αυτόματα.

Τα Φύλλα Στυλ επιτρέπουν τον ορισμό των πληροφοριών στυλ με πολλούς τρόπους. Τα στυλ μπορούν να ορισθούν μέσα σ' ένα μόνο HTML στοιχείο, μέσα στο στοιχείο <head> μιας HTML σελίδας ή σ' ένα εξωτερικό αρχείο CSS. Πολλά εξωτερικά Φύλλα Στυλ μπορούν να χρησιμοποιούνται μέσα απ' ένα μόνο HTML έγγραφο. Σε γενικές γραμμές μπορούμε να πούμε ότι όλα τα στυλ θα καταλήξουν (cascade) σ' ένα νέο εικονικό (virtual) Φύλλο Στυλ σύμφωνα με τους παρακάτω κανόνες, όπου ο μεγαλύτερος αριθμός έχει και την υψηλότερη προτεραιότητα :

- Προεπιλογή του φυλλομετρητή.
- Εξωτερικό Φύλλο Στυλ (External Style Sheet).
- Εσωτερικό Φύλλο Στυλ (Internal Style Sheet), μέσα στο τμήμα header του εγγράφου.
- Inline Style, μέσα στο HTML στοιχείο.

### *Εισαγωγή ενός Φύλλου Στυλ (Style Sheet)*

---

Όταν ένας φυλλομετρητής διαβάζει ένα φύλλο στυλ (style sheet), θα μορφοποιήσει το έγγραφο σύμφωνα μ' αυτό. Υπάρχουν τρεις τρόποι για να εισάγουμε ένα φύλλο στυλ.

### *Εξωτερικά Φύλλα Στυλ (External Style Sheets)*

---

Ένα εξωτερικό φύλλο στυλ (external style sheet) είναι ιδανικό όταν το στυλ εφαρμόζεται σε πολλές σελίδες. Μ' ένα εξωτερικό φύλλο στυλ μπορούμε να αλλάξουμε την εμφάνιση ενός ολόκληρου Web site αλλάζοντας ένα μόνο αρχείο. Η κάθε σελίδα πρέπει να έχει έναν δεσμό (link) προς το φύλλο στυλ που χρησιμοποιεί το tag <link>, ο οποίος βρίσκεται μέσα στο τμήμα head, ως εξής:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

Ο φυλλομετρητής θα διαβάσει τους ορισμούς στυλ από το αρχείο mystyle.css και θα μορφοποιήσει το έγγραφο σύμφωνα μ' αυτό το αρχείο. Ένα εξωτερικό φύλλο στυλ μπορεί να γραφεί σ' έναν οποιονδήποτε text editor. Το αρχείο δεν πρέπει να περιέχει καθόλου html tags και πρέπει να αποθηκευθεί με την επέκταση css.

### *Εσωτερικά Φύλλα Στυλ (Internal Style Sheets)*

---

Ένα εσωτερικό φύλλο στυλ (internal style sheet) πρέπει να χρησιμοποιηθεί όταν ένα έγγραφο έχει ένα μοναδικό στυλ. Ορίζουμε τα εσωτερικά στυλ στο τμήμα head χρησιμοποιώντας το tag <style>, ως εξής :

```
<style type="text/css">
hr {color: sienna}
p {margin-left: 20px}
body {background-image: url("images/back40.gif")}
</style>
```

Ο φυλλομετρητής θα διαβάσει τους ορισμούς των στυλ και θα μορφοποιήσει ανάλογα το έγγραφο. Ένας φυλλομετρητής κανονικά αγνοεί τα άγνωστα tags. Αυτό σημαίνει ότι ένας παλιός φυλλομετρητής που δεν υποστηρίζει στυλ, θα αγνοήσει το tag <style>, αλλά το περιεχόμενο του tag <style> θα εμφανισθεί στη σελίδα.

### *JavaScript*

---

Η JavaScript είναι γλώσσα προγραμματισμού η οποία έχει σαν σκοπό την παραγωγή δυναμικού περιεχομένου και την εκτέλεση κώδικα στην πλευρά του πελάτη (client-side) σε ιστοσελίδες. Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMA script. Η Java είναι μια γλώσσα προγραμματισμού, παρόμοια κατά κάποιο τρόπο με τη C++, αλλά με την πρόθεση να είναι πιο αξιόπιστη για εφαρμογές Internet. Αρκετοί πιστεύουν ότι το JavaScript είναι Java λόγω της ομοιότητας του ονόματος, αλλά αυτό δεν αληθεύει. Η Java στεχεύει στο να υλοποιούνται προγράμματα με ένα λειτουργικό σύστημα έτοιμο να τρέξει. Ο προγραμματιστής μπορεί να δημιουργήσει μία και μόνο έκδοση του προγράμματος που θα τρέχει μέσα από έναν μεταφραστή της Java, ένα πρόγραμμα όπως το HotJava ή ο Netscape Navigator. Υπάρχουν διαφορετικοί διερμηνευτές για διαφορετικούς



υπολογιστές και λειτουργικά συστήματα ( ο Netscape Navigator που τρέχει σε Macintosh, μια έκδοσή του που τρέχει σε Windows 3.1, μια για Windows 95 κ.λ.π. ).

Με αυτό τον τρόπο ένα και μόνο πρόγραμμα Java μπορεί να τρέξει σε πολλά και διαφορετικά λειτουργικά συστήματα, εφόσον ο χρήστης έχει έναν διερμηνευτή που να τρέχει στο σύστημά του. Οι διερμηνευτές της Java δεν είναι αληθινά λειτουργικά συστήματα αν και βασίζονται λίγο-πολύ στις ίδιες αρχές. Ένα λειτουργικό σύστημα είναι η σύζευξη μεταξύ ενός προγράμματος και του υπολογιστή. Αντί για ένα πρόγραμμα που μιλάει κατευθείαν στα περιφερειακά του υπολογιστή, ο προγραμματιστής μπορεί απλά να γράψει ένα πρόγραμμα που μιλάει στο λειτουργικό σύστημα και να βάλει το λειτουργικό σύστημα στον κόπο να λειτουργήσει τα περιφερειακά. Ο διερμηνευτής της Java λειτουργεί με παρόμοιο τρόπο. Βρίσκεται ανάμεσα στο λειτουργικό σύστημα και το Java πρόγραμμα. Αντί να γράφει διαφορετικές εκδοχές του ίδιου προγράμματος, γράφει ένα και μόνο πρόγραμμα που επικοινωνεί με οποιονδήποτε διερμηνευτή της Java σε οποιοδήποτε σύστημα υπολογιστή. Ο διερμηνευτής μεσολαβεί ανάμεσα στο Java πρόγραμμα και το λειτουργικό σύστημα, μεταφράζοντας ό,τι λέει το ένα σε κάτι που μπορεί να καταλάβει το άλλο. Πρόκειται για ριζοσπαστική τεχνολογία υποστηρίζουν οι υπερασπιστές της Java, γιατί γκρεμίζει τους περιορισμούς που μπαίνουν στο λογισμικό εξαιτίας των λειτουργικών συστημάτων. Η Java προσφέρει μια πολυδύναμη γλώσσα προγραμματισμού με την οποία μπορούν να δημιουργηθούν καλομελετημένα προγράμματα που μπορούν να τρέξουν σε πολλούς διαφορετικούς υπολογιστές.

### *Τρόπος σύνταξης*

---

Όπως και η PHP, η Javascript έχει βασιστεί όσον αφορά τον τρόπο σύνταξης του κώδικά της στη γλώσσα προγραμματισμού C, με την οποία παρουσιάζει πολλές ομοιότητες. Όμως ενώ η PHP εκτελείται στο διακομιστή (server-side γλώσσα προγραμματισμού), η Javascript εκτελείται στον πελάτη (client-side γλώσσα προγραμματισμού). Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή αλλά στο πρόγραμμα περιήγησης των επισκεπτών. Αυτή η διαφορά έχει και πλεονεκτήματα και μειονεκτήματα για καθεμιά από τις δύο γλώσσες. Συγκεκριμένα, η Javascript δεν έχει καμία απαίτηση από πλευράς δυνατοτήτων του server για να εκτελεστεί (όπως επεξεργαστική ισχύ ή συμβατό λογισμικό διακομιστή), αλλά βασίζεται στις δυνατότητες του browser των επισκεπτών.

Επίσης μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Παρόλα αυτά, οι δυνατότητές της διαφέρουν σημαντικά από αυτές της PHP (για παράδειγμα δεν παρέχει συνδεσιμότητα με βάσεις δεδομένων που βρίσκονται στο διακομιστή).

### *Δείγμα κώδικα javascript*

---

Ο κώδικας Javascript μιας σελίδας περικλείεται από τις ετικέτες της HTML `<script type="text/javascript">` και `</script>`.

Για παράδειγμα, ο ακόλουθος κώδικας Javascript εμφανίζει ένα πλαίσιο διαλόγου με το κείμενο "Γεια σου, κόσμε!":

```
<script type="text/javascript">
alert("Γεια σου, κόσμε!");
</script>
```

**Εικόνα 63:** Κώδικας Javascript

Αν ο κώδικας Javascript περιέχει περισσότερες από μία εντολές, αυτές θα πρέπει να διαχωριστούν μεταξύ τους με το χαρακτήρα του ελληνικού ερωτηματικού ';' (δηλαδή της λατινικής άνω τελείας). Η χρήση του χαρακτήρα αυτού για την τελευταία εντολή δεν είναι απαραίτητη. Η διαχώριση των εντολών στους νεότερους φυλλομετρητές (browsers) δεν είναι απαραίτητη.

Μια άλλη βασική εντολή, η `window.prompt("μήνυμα προς το χρήστη")`, ζητάει από το χρήστη να συμπληρώσει ένα κομμάτι μιας αίτησης απευθείας ώστε τα δεδομένα να χρησιμοποιηθούν σαν κείμενο:

```
<script>
var Onoma = window.prompt("PLEASE ENTER YOUR NAME")
alert("Your name is " + Onoma + ".")
</script>
```

**Εικόνα 64:** Κώδικας Javascript

### *Οφέλη της JavaScript*

---

Με τη JavaScript μπορούμε να φτιάξουμε σενάρια που να εκτελούν αυτόματες εργασίες, π.χ. όταν μια σελίδα του Web ανοίγει ή κλείνει. Επίσης μπορούμε να κάνουμε την JavaScript να εκτελεί ενέργειες ανταποκρινόμενη σε ένα συγκεκριμένο γεγονός. Για παράδειγμα όταν ο χρήστης επιλέγει ένα κουμπί ή ένα σύνδεσμο, όταν εστιάζει από ένα στοιχείο μιας φόρμας σε ένα άλλο στοιχείο της κ.ο.κ. Οι ενέργειες αυτές μπορεί να είναι απλές. Τα σενάρια μπορεί να ανοίγουν νέα παράθυρα στον browser και να εμφανίζουν συγκεκριμένα HTML έγγραφα ή να παρουσιάζουν μια σελίδα επιλεγμένη από τον κατάλογο ιστορικού του browser. Μπορεί επίσης να είναι και περίπλοκες δηλαδή ένα σενάριο μπορεί να ελέγχει τα περιεχόμενα μιας φόρμας που θέλει να υποβάλει ένας χρήστης και στη συνέχεια να προειδοποιεί τον χρήστη αν τα δεδομένα είναι λάθος. Το σενάριο μπορεί να ψάξει για

πληροφορίες σε μια μικρή βάση δεδομένων ή να κάνει πολύπλοκους υπολογισμούς οικονομικών στοιχείων.

## *Τρέχοντας JavaScript*

---

Για να τρέξουν scripts γραμμένα σε JavaScript χρειαζόμαστε ένα browser που υποστηρίζει JavaScript - παραδείγματος χάριν το Netscape Navigator (από την έκδοση 2.0) ή το Microsoft Internet Explorer (MSIE εω συντομία - από την έκδοση 3.0). Εφόσον αυτοί οι δύο browsers είναι πολύ διαδεδομένοι, πολλοί χρήστες έχουν την δυνατότητα να τρέξουν scripts γραμμένα σε JavaScript. Αυτό είναι βεβαίως ένας σημαντικός λόγος για να διαλέξετε JavaScript για να εμπλουτίσετε τις web-σελίδες σας. Βεβαίως χρειάζεται πρώτα να κατανοείτε βασικές λειτουργίες της HTML προτού διαβάσετε αυτό το tutorial. Μπορείτε να βρείτε πολλές on-line πηγές πληροφοριών για HTML. Το καλύτερο που έχετε να κάνετε είναι να χρησιμοποιήσετε μια μηχανή αναζήτησης για τη λέξη-κλειδί 'html', π.χ. στο Yahoo για να πάρετε πληροφορίες για το HTML.

## *Events - Γεγονότα*

---

Events ονομάζουμε τις ενέργειες που βάζουν σε κίνηση το μηχανισμό εκτέλεσης των Methods. Για παράδειγμα όταν κάνει κλικ ο χρήστης να τρέχει ένα υποπρόγραμμα: `onClick="run_my_function()`. Τα γεγονότα (Events) και οι χειριστές γεγονότων (event handlers) είναι ένα πολύ σημαντικό μέρος στον JavaScript προγραμματισμό. Τα Events προκαλούνται από τις πράξεις του χρήστη. Αν ο χρήστης πατήσει ένα κουμπί, τότε συμβαίνει ένα Click-event. Αν ο δείκτης του mouse κινηθεί πάνω από μια διεύθυνση (link), τότε συμβαίνει ένα MouseOver-event. Υπάρχουν πολλά διαφορετικά events.

Εάν θέλουμε το JavaScript πρόγραμμά να αντιδρά σε συγκεκριμένα events, μπορεί να γίνει με την βοήθεια των event-handlers (χειριστές γεγονότων). Ένα κουμπί μπορεί να εμφανίζει ένα pop-up παράθυρο όταν πατιέται. Αυτό σημαίνει ότι το pop-up παράθυρο πρέπει να εμφανιστεί σαν απάντηση στο Click-event. Ο event-handler που χρειαζόμαστε λέγεται `onClick`. Αυτός λέει στον υπολογιστή τι να γίνει όταν συμβεί το ανάλογο event. Ο επόμενος κώδικας δείχνει ένα απλό παράδειγμα του event-handler `onClick`:

```
<form>
<input type="button" value="Πάτησε με" onClick="alert('Καλημέρα!')">
</form>
```

## *XML*

---

Σε ένα κόσμο όπου οι πληροφορίες παρέχονται μέσω του παγκόσμιου διαδικτύου, τα έγγραφα πρέπει να είναι εύκολα προσβάσιμα, μεταφέρσιμα και ευέλικτα. Πρέπει επίσης να είναι ανεξάρτητα οποιουδήποτε συστήματος και περιεχομένου. Οι γενικευμένες γλώσσες έχουν τέτοια χαρακτηριστικά, παρέχοντας στα έγγραφα αυτά μια δυνατότητα η οποία δεν υπάρχει σε άλλες γλώσσες περιγραφής εγγράφων. Η HTML είναι προβληματική και περιοριστική γλώσσα. Η XML έλυσε πολλά από τα προβλήματα που αντιμετώπισαν οι σχεδιαστές του web και είναι υπεύθυνη για την XHTML, μια ανασχεδιασμένη HTML. Θα χρησιμοποιείται για πολλά χρόνια επειδή προσφέρει αποτελεσματικές και δυναμικές πολυμεσικές λύσεις. Η XML σχεδιάστηκε να ικανοποιήσει πολλές ανάγκες δίνοντας στα έγγραφα ένα μεγαλύτερο επίπεδο προσαρμοστικότητας στο στυλ και τη δομή από αυτό που υπήρχε παλαιότερα στην HTML. Η XML προσφέρει στους σχεδιαστές της HTML τη δυνατότητα να προσθέτουν περισσότερα στοιχεία στη γλώσσα, δεν αναφέρεται μονάχα στους σχεδιαστές του web αλλά σε οποιονδήποτε ασχολείται με εκδόσεις.

Στην πραγματικότητα η XML είναι markup γλώσσα για έγγραφα που περιέχουν δομημένες πληροφορίες. Η Markup γλώσσα είναι ένας μηχανισμός που καθορίζει δομές σε ένα έγγραφο. Οι δομημένες πληροφορίες περιλαμβάνουν περιεχόμενο και κάποιες διευκρινίσεις για το ρόλο που παίζει το περιεχόμενο. Σχεδόν όλα τα έγγραφα έχουν την ίδια δομή.

Η XML είναι κάτι περισσότερο από markup language είναι metalanguage, δηλαδή μια γλώσσα που χρησιμοποιείται για να καθορίσει νέες markup γλώσσες. Η XML συμπληρώνει και δεν αντικαθιστά την HTML. Ενώ η HTML χρησιμοποιείται στη διατύπωση και την εμφάνιση των δεδομένων, η XML αναπαριστά τη συναφή έννοια των δεδομένων. Στην HTML τα tags είναι προκαθορισμένα ενώ η XML παρέχει τη δυνατότητα να καθορίζουν οι χρήστες τα tags και τις δομημένες μεταξύ τους σχέσεις.

Τα XML έγγραφα δεν είναι πολύπλοκα αλλά απλά και πολύ αποτελεσματικά. Το διδακτικό υλικό της well-formed XML αναλύει τη δημιουργία των XML εγγράφων, η οποία είναι κατά κάποιο τρόπο ίδια με την HTML καθώς επιτρέπει τη μη δομημένη δημιουργία εγγράφου.

Η valid XML είναι πιο σύνθετη. Απαιτεί την ύπαρξη ενός Document Type Definition πριν να γραφεί το έγγραφο αλλά παρέχει μια γενική δομή με βάση την οποία τη δημιουργούμε. Η γλώσσα προγραμματισμού XML περιγράφει μια κατηγορία πληροφοριών (data objects) που καλούνται XML έγγραφα (documents) καθώς επίσης περιγράφει τμηματικά τη συμπεριφορά των προγραμμάτων που τα επεξεργάζονται. Τα έγγραφα αυτά αποτελούνται από μονάδες αποθήκευσης που καλούνται entities (οντότητες), οι οποίες περιέχουν πληροφορίες αναλυμένες ή μη. Οι αναλυμένες πληροφορίες αποτελούνται από χαρακτήρες (characters) οι οποίοι συνθέτουν character data και άλλοι οι οποίοι συνθέτουν markup. Η μορφή markup κωδικοποιεί την περιγραφή της τελικής αποθήκευσης του εγγράφου καθώς και τη λογική δομή.

Ένα λογισμικό μοντέλο που καλείται επεξεργαστής XML χρησιμοποιείται να διαβάσει XML έγγραφα και παρέχει πρόσβαση στο περιεχόμενο και τη δομή τους. Υποτίθεται ότι ο επεξεργαστής XML λειτουργεί εκ μέρους ενός άλλου μοντέλου που καλείται application (εφαρμογή). Αυτή η προδιαγραφή περιγράφει την απαιτούμενη συμπεριφορά του επεξεργαστή και συγκεκριμένα πως θα πρέπει να διαβάζει τα XML δεδομένα και ποιες πληροφορίες πρέπει να παρέχει στην εφαρμογή.

### *Προέλευση και στόχοι (Origin and goals)*

---

Η γλώσσα XML αναπτύχθηκε από μια Ομάδα Εργασίας της XML κάτω από την καλή κηδεμονία του διεθνούς οργανισμού World Wide Web Consortium (W3C) το 1996. Εδραιώθηκε από τον John Bosak της Sun Microsystems με την ενεργή συμμετοχή μιας XML Ομάδας Ειδικού Ενδιαφέροντος (που οργανώθηκε από τον οργανισμό W3C).

Οι προσχεδιασμένοι στόχοι της XML είναι:

- Η XML πρέπει να είναι εύχρηστη στο Internet.
- Η XML πρέπει να υποστηρίζει μεγάλη ποικιλία από εφαρμογές.
- Η XML πρέπει να είναι συμβατή με την SGML.
- Θα είναι εύκολο να γράφονται προγράμματα που επεξεργάζονται XML έγγραφα.
- Τα XML έγγραφα θα πρέπει να είναι ευανάγνωστα.
- Ο σχεδιασμός XML θα πρέπει να προετοιμάζεται γρήγορα.
- Ο σχεδιασμός XML θα πρέπει να είναι τυπικός και περιεκτικός.
- Τα XML έγγραφα θα πρέπει να δημιουργούνται εύκολα.
- Η περιεκτικότητα στον XML συμβολισμό είναι μικρής σημασίας

Η XML (Extensible Markup Language) είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Ορίζεται κυρίως στην προδιαγραφή XML 1.0, που δημιούργησε ο διεθνής οργανισμός προτύπων W3C (World Wide Web Consortium), αλλά και σε διάφορες άλλες σχετικές προδιαγραφές ανοιχτών προτύπων. Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες του κόσμου. Αν και η σχεδίαση της XML εστιάζει τα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού.

Υπάρχει μία ποικιλία διεπαφών προγραμματισμού εφαρμογών που μπορούν να χρησιμοποιούν οι προγραμματιστές για να προσπελαίνουν δεδομένα XML, αλλά και διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών που προκύπτουν από την XML.

Έως το 2009 έχουν αναπτυχθεί εκατοντάδες γλώσσες που βασίζονται στην XML, συμπεριλαμβανομένων του RSS, του SOAP και της XHTML. Προεπιλεγμένες κωδικοποιήσεις βασισμένες στην XML, υπάρχουν για τις περισσότερες σουίτες εφαρμογών γραφείου, συμπεριλαμβανομένων του Microsoft Office (Office Open XML), του OpenOffice.org (OpenDocument) και του iWork της εταιρίας Apple.

## *Βασική Ορολογία*

---

Το περιεχόμενο αυτής της ενότητας βασίζεται στην προδιαγραφή XML 1.0, ενώ αποτελεί μία πλήρη λίστα όλων των όρων που υπάρχουν στη γλώσσα XML. Είναι μία εισαγωγή στα βασικά στοιχεία που συναντώνται στην καθημερινή της χρήση.

## *Χαρακτήρας Unicode*

---

Εξ ορισμού, ένα κείμενο XML είναι μία ακολουθία χαρακτήρων. Σχεδόν κάθε χαρακτήρας Unicode μπορεί να εμφανίζεται σε ένα κείμενο XML.

## *Επεξεργαστής και Εφαρμογή*

---

Επεξεργαστής είναι το λογισμικό που επεξεργάζεται ένα κείμενο XML. Είναι αναμενόμενο ότι ένας επεξεργαστής δουλεύει για μία εφαρμογή. Υπάρχουν μερικές πολύ συγκεκριμένες απαιτήσεις, σχετικά με το τι μπορεί και τι δεν μπορεί να κάνει ένας επεξεργαστής XML, αλλά καμία όσον αφορά στη συμπεριφορά της εφαρμογής. Ο επεξεργαστής (όπως ονοματίζεται από την προδιαγραφή), αναφέρεται συχνά με τον αγγλικό όρο XML parser.

## *Σήμανση και Περιεχόμενο*

---

Οι χαρακτήρες που απαρτίζουν ένα κείμενο XML, αποτελούν είτε τη σήμανση είτε το περιεχόμενό του. Η σήμανση και το περιεχόμενο μπορούν να επισημανθούν και να διακριθούν, ύστερα από την εφαρμογή κάποιων απλών συντακτικών κανόνων. Όλα τα αλφαριθμητικά που συνιστούν τη σήμανση, είτε ξεκινούν με το χαρακτήρα "<" και καταλήγουν στο χαρακτήρα ">", είτε ξεκινούν με το χαρακτήρα "&" και καταλήγουν στο χαρακτήρα ";" . Ακολουθίες χαρακτήρων που δε συνιστούν τη σήμανση, αποτελούν το περιεχόμενο ενός κειμένου XML.

## *Ετικέτα*

---

Ένα στοιχείο σήμανσης που ξεκινά με το χαρακτήρα "<" και καταλήγει στο χαρακτήρα ">". Υπάρχουν τρία είδη ετικέτας: ετικέτες-αρχής, για παράδειγμα <section>, ετικέτες-τέλους, για παράδειγμα </section>, και ετικέτες-χωρίς-περιεχόμενο, για παράδειγμα <line-break/>.

## Στοιχεία

---

Ένα λογικό απόσπασμα ενός κειμένου, που είτε ξεκινά με μία ετικέτα-αρχής και καταλήγει σε μία ετικέτα-τέλους, είτε αποτελείται μόνο από μία ετικέτα-χωρίς-περιεχόμενο. Οι χαρακτήρες που υπάρχουν, αν υπάρχουν, μεταξύ μιας ετικέτας-αρχής και μιας ετικέτας-τέλους, συνιστούν το περιεχόμενο του στοιχείου, το οποίο μπορεί να περιέχει σήμανση, συμπεριλαμβανομένων και άλλων στοιχείων που ονομάζονται στοιχεία-παιδιά. Ένα παράδειγμα ενός στοιχείου είναι το `<Greeting>Hello, world.</Greeting>`. Ένα άλλο είναι το `<line-break/>`.

## Χαρακτηριστικά

---

Ένα στοιχείο σήμανσης που αποτελείται από ένα ζευγάρι όνομα/τιμή, το οποίο υπάρχει μέσα σε μία ετικέτα-αρχής ή σε μία ετικέτα-χωρίς-περιεχόμενο. Στο παράδειγμα παρακάτω, το στοιχείο `img` έχει δύο χαρακτηριστικά, τα `src` και `alt`: ``. Ένα άλλο παράδειγμα θα ήταν το `<step number="3">Connect A to B.</step>`, όπου το όνομα του χαρακτηριστικού είναι `"number"` και η τιμή του είναι `"3"`.

## Παράδειγμα

Το παρακάτω είναι ένα μικρό, αλλά πλήρες κείμενο XML, που κάνει χρήση όλων των παραπάνω εννοιών και στοιχείων.

```
<?xml version="1.0" encoding='UTF-8'?>
<painting>

<caption>This is Raphael's "Foligno" Madonna, painted in
<date>1511</date>-<date>1512</date>.</caption>
</painting>
```

Υπάρχουν πέντε στοιχεία σε αυτό το κείμενο του παραδείγματος: τα `painting`, `img`, `caption`, και δύο `date`. Τα στοιχεία `date`, είναι παιδιά του στοιχείου `caption`, το οποίο είναι παιδί του στοιχείου-ρίζας `painting`. Το στοιχείο `img` έχει δύο χαρακτηριστικά, τα `src` και `alt`, στοιχείου-ρίζας `painting`. Το στοιχείο `img` έχει δύο χαρακτηριστικά, τα `src` και `alt`.

## Χαρακτήρες και διαφυγή.

---

Τα κείμενα XML αποτελούνται εξ ολοκλήρου από χαρακτήρες Unicode. Εκτός από ένα μικρό αριθμό, ειδικά εξαιρούμενων χαρακτήρων ελέγχου, κάθε χαρακτήρας που ορίζεται στο Unicode, μπορεί να εμφανίζεται στο περιεχόμενο ενός κειμένου XML. Το σύνολο των χαρακτήρων που μπορούν να εμφανίζονται στη σήμανση, αν και

κάπως περιορισμένο, παραμένει μεγάλο. Η XML παρέχει κάποιες διευκολύνσεις για την ταυτοποίηση της κωδικοποίησης των χαρακτήρων Unicode που απαρτίζουν ένα κείμενο και για την απεικόνιση χαρακτήρων που για τον έναν ή τον άλλο λόγο, δεν μπορούν να χρησιμοποιηθούν ευθέως.

## *Βήματα Υλοποίησης Συστήματος*

---

### *Δημιουργία web server μέσω Xampp*

---

Το XAMPP είναι ένα χρήσιμο βοήθημα που μας δίνει την δυνατότητα να κάνουμε τον υπολογιστή μας ένα web server και όχι μόνο. Προορίζεται κυρίως για όσους θέλουν εύκολα και γρήγορα να εγκαταστήσουν όλα τα απαραίτητα εργαλεία για να αναπτύξουν και να δοκιμάσουν, τοπικά στον υπολογιστή τους, εφαρμογές για το διαδίκτυο.

Το XAMPP περιέχει τα ακόλουθα πακέτα:

- Apache 2.2.2
- MySQL 5.0.21
- PHP 5.1.4 + PHP 4.4.2-pl1 + PEAR
- PHP-Switch win32 1.0 (von ApacheFriends, man nehme die "php-switch.bat")
- XAMPP Control Version 2.3
- XAMPP Security 1.0
- SQLite 2.8.15
- OpenSSL 0.9.8b
- phpMyAdmin 2.8.1
- ADOdb 4.80
- Mercury Mail Transport System v4.01b
- FileZilla FTP Server 0.9.16c
- Webalizer 2.01-10
- Zend Optimizer 3.0.0

Αρχικά εγκαταστάθηκε το XAMPP σε επιλεγμένο υπολογιστικό σύστημα

Εγκατάσταση του Xampp από εδώ <http://www.apachefriends.org/en/xampp.html>

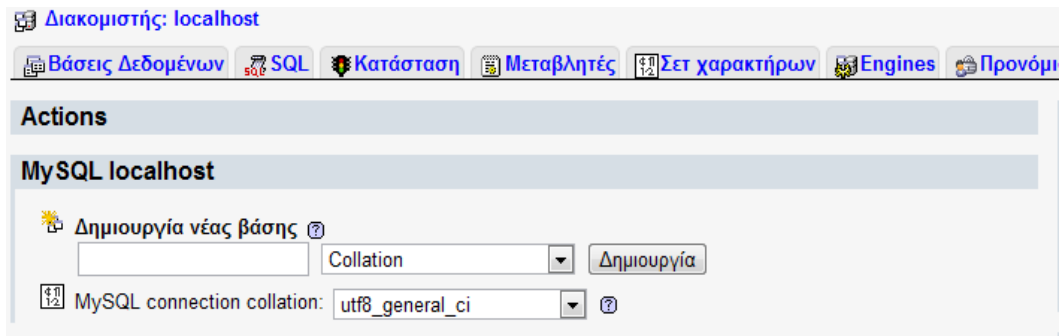
Αυτός ο εικονικός server αποτελείται κυρίως από την Apache HTTP Server, MySQL που επικοινωνεί με την βάση δεδομένων μας.

### *Δημιουργία βάσης*

---

Στη διεύθυνση <http://localhost/phpmyadmin/>





**Εικόνα:** Δημιουργία βάσης

Στην δημιουργία νέας βάσης βάζουμε το όνομα επιλογής. Στο όνομα θα μπορούσαμε να βάλουμε οποιοδήποτε όνομα έχουμε δώσει στο σύστημά μας.

### *Ρυθμίσεις*

---

1. Στο αρχείο config.php αλλάζουμε το path αν έχουμε μετακινήσει το φάκελο σε άλλο path. `define('ABSOLUTE_DIR','C:\xampp\htdocs\');`  
2. Στο αρχείο config.php δηλώνουμε το username, passwords etc  
Συνήθως ισχύουν τα εξής για το xampp αν δεν έχουμε πειράξει τις default ρυθμίσεις κατά την εγκατάσταση.

```
$GLOBALS['config']['host']= 'localhost';  
$GLOBALS['config']['username']= 'root';  
$GLOBALS['config']['password']= '';  
$GLOBALS['config']['database']= 'Deus';
```

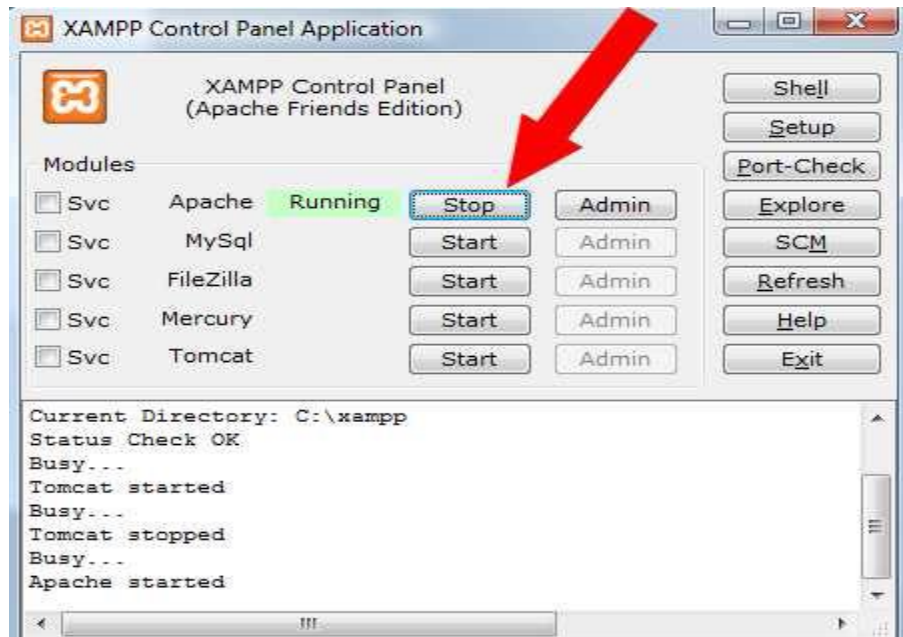
### *Εγκατάσταση διαδικτυακής εφαρμογής*

---

Για να εγκαταστήσουμε χρησιμοποιούμε στην διεύθυνση

`http :// localhost / Deus / install / install . php`

Αφού γίνει η εγκατάσταση των εργασιών μας όταν την τρέχουμε την εργασία μας (με την PHP και το Ajax) ο APACHE πρέπει να είναι ενεργοποιημένος όπως φαίνεται στην παρακάτω εικόνα:



**Εικόνα:** Παράδειγμα XAMMP

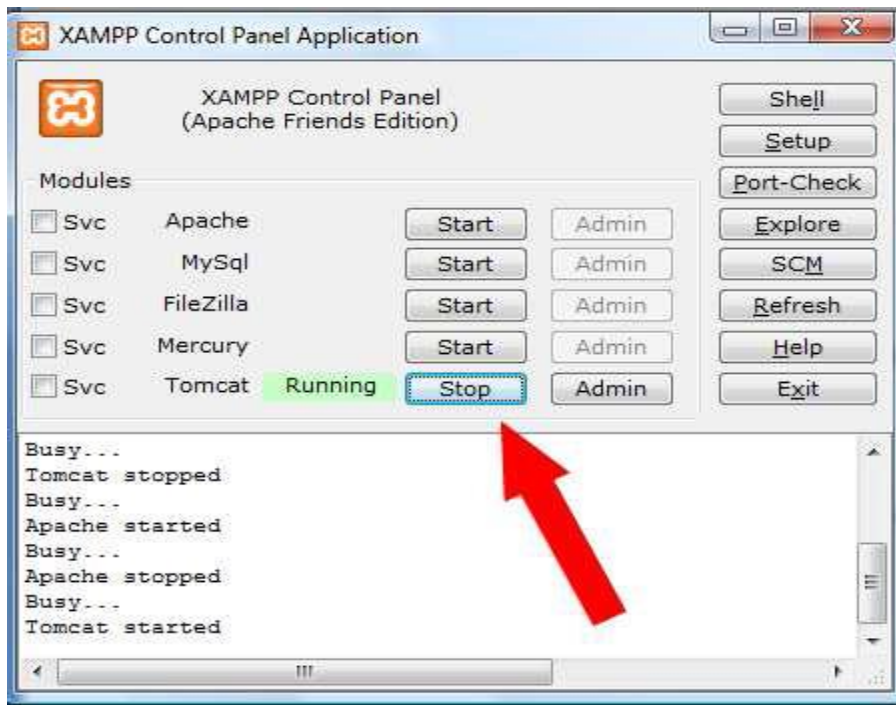
Όλα τα αρχεία μας για την εργασία τα βάζουμε σε φάκελο μέσα στον apache πχ μέσα στο

C:\xampp\htdocs\

Για να δούμε-εκτελέσουμε την άσκηση μας ανοίγουμε στον browser μας την εξής διεύθυνση:

http :// localhost / Deus / .

Όταν τρέχουμε τη διατριβή με τα JSP ο TOMCAT πρέπει να είναι ενεργοποιημένος όπως φαίνεται στην εικόνα.



**Εικόνα:** Παράδειγμα XAMMP

### *Σύστημα λήψης απόφασης*

---

Το υποσύστημα υποστήριξης λήψης έχει εγκατασταθεί και αξιοποιείται ως ένα μαύρο κουτί από το συνολικό σύστημα.

### **Έμπειρα συστήματα**

---

#### *Ορισμός Έμπειρου Συστήματος*

---

Υπάρχουν πολλοί ορισμοί έμπειρων συστημάτων. Άλλοι βασίζονται στη λειτουργία τους, άλλοι στη δομή τους και άλλοι συνδυάζουν στοιχεία και των δυο. Στα παρακάτω θα παρουσιάσουμε σύντομα τα λειτουργικά και δομικά χαρακτηριστικά ενός έμπειρου συστήματος.[60] Τα έμπειρα συστήματα είναι προγράμματα τα οποία επιδεικνύουν νοήμονα συμπεριφορά σε συγκεκριμένους τομείς και διαδικασίες, ανάλογη ενός ανθρώπου εμπειρογνώμονα με ειδικότητα σε κάποιο τομέα. Ένα έμπειρο σύστημα είναι ένα πρόγραμμα υπολογιστή το οποίο κωδικοποιεί και χειρίζεται τη γνώση και τη συλλογιστική που υπάρχει για έναν εξειδικευμένο τομέα με σκοπό την επίλυση προβλημάτων ή/και την παροχή συμβουλών στον εν λόγω τομέα.

Ο πυρήνας του έμπειρου συστήματος αποτελείται από δύο μέρη: τη *βάση γνώσης* (knowledge base) και τον *μηχανισμό εξαγωγής συμπερασμάτων* (inference engine). Αυτός ο πλήρης διαχωρισμός της γνώσης του συστήματος από το μηχανισμό χειρισμού της είναι βασικό χαρακτηριστικό των έμπειρων συστημάτων. Προσφέρει διαφάνεια στον προγραμματισμό του συστήματος και ευκολία προσθήκης, αφαίρεσης και τροποποίησης της γνώσης. Ο πυρήνας του έμπειρου συστήματος αναπτύσσεται συνήθως σε ένα προγραμματιστικό περιβάλλον ανάπτυξης υψηλού επιπέδου, το οποίο μπορεί να είναι είτε μια γλώσσα προγραμματισμού ή κάποιο εργαλείο εξειδικευμένο για την ανάπτυξη έμπειρων συστημάτων.

### Λειτουργικά στοιχεία

---

Τα παρακάτω χαρακτηριστικά περιγράφουν τι (πρέπει να) κάνει ένα έμπειρο σύστημα (και όχι πώς επιτυγχάνεται).

#### Χώρος προβλημάτων

Ένα έμπειρο σύστημα:

- Συμπεριφέρεται σαν ένας έμπειρος (άνθρωπος).
- Εφαρμόζει γνώση εμπείρου για τη λύση δύσκολων προβλημάτων του πραγματικού κόσμου (real world problems).
- Λύνει προβλήματα αποδοτικά και αποτελεσματικά σε μια εξειδικευμένη περιοχή γνώσης/προβλημάτων.
- Επιλύει προβλήματα τα οποία έχουν συμβολική αναπαράσταση.
- Λύνει προβλήματα τα οποία είναι αρκετά δύσκολα και απαιτούν μεγάλη εξειδίκευση για τη λύση τους.

#### Απαιτήσεις αποδοτικότητας.

Ένα έμπειρο σύστημα:

- Είναι ικανό να λειτουργήσει σε επίπεδο ενός εμπειρου.
- Μιμείται τις συμβουλευτικές ικανότητες εμπείρων.

#### Εξήγηση συλλογιστικής

Ένα έμπειρο σύστημα:

- Είναι ικανό να δικαιολογήσει τον τρόπο συλλογιστικής του έτσι ώστε να καταλάβει ο χρήστης το αποτέλεσμα.

Τα παρακάτω χαρακτηριστικά περιγράφουν πώς επιτυγχάνει τους σκοπούς του ένα έμπειρο σύστημα.

### Τεχνικές Τεχνητής Νοημοσύνης

Ένα έμπειρο σύστημα:

Χρησιμοποιεί προγραμματιστικές τεχνικές Τεχνητής Νοημοσύνης και ειδικότερα αυτές της επίλυσης προβλημάτων.

### Γνώση

Ένα έμπειρο σύστημα:

Ενσωματώνει σε ένα πρόγραμμα τη γνώση ενός έμπειρου.

Αποθηκεύει γνώση σε μορφή γεγονότων και ευριστικών. Τα γεγονότα αντιπροσωπεύουν πληροφορίες οι οποίες είναι κοινά γνωστές, διαθέσιμες και αποδεκτές από έμπειρους στο γνωστικό πεδίο.

### Διαχωρισμός γνώσης και ελέγχου

Ένα έμπειρο σύστημα:

Διαχωρίζει τη γνώση από το υπόλοιπο σύστημα το οποίο την επεξεργάζεται.

### Χρήση συλλογιστικών διαδικασιών - αβεβαιότητα

Ένα έμπειρο σύστημα:

- Είναι ένα "έξυπνο" υπολογιστικό πρόγραμμα το οποίο χρησιμοποιεί γνώση και διαδικασίες εξαγωγής συμπερασμάτων.
- Βασίζεται σε κανόνες.
- Διαχωρίζει τη γνώση από το υπόλοιπο σύστημα το οποίο την επεξεργάζεται.
- Μπορεί να χειριστεί αποτελεσματικά την αβεβαιότητα των πληροφοριών.

Ένα έμπειρο σύστημα (expert system) διαθέτει κωδικοποιημένη γνώση από ένα σχετικά περιορισμένο γνωστικό αντικείμενο. Το σύστημα χρησιμοποιεί την γνώση αυτή για τη λύση προβλημάτων τα οποία συνήθως απαιτούν για την λύση τους την εμπειρία ενός εμπειρογνώμονα (ειδικού). Ένας εμπειρογνώμονας είναι σε θέση να

μετατρέψει ένα πρόβλημα που του παρουσιάζεται σε μία μορφή που να επιτρέπει την γρήγορη και αποτελεσματική λύση. Ο εμπειρογνώμονας πρέπει να είναι σε θέση να εξηγεί τα αποτελέσματα, να μαθαίνει για το αντικείμενό του και να αναμορφώνει την γνώση που έχει όποτε αυτό απαιτείται.

Συμβατικά Συστήματα	Έμπειρα Συστήματα
επεξεργασία και γνώση συνδυάζονται σε ένα σειριακό πρόγραμμα	η βάση γνώσης είναι χωριστή από τον μηχανισμό επεξεργασίας
τα προγράμματα δεν κάνουν λάθη	το πρόγραμμα μπορεί να κάνει λάθος
συνήθως δεν εξηγούν γιατί τα δεδομένα εισόδου χρειάζονται ή πώς συνάγονται τα συμπεράσματα	η επεξήγηση είναι μέρος των περισσότερων έμπειρων συστημάτων
η τροποποίηση της γνώσης είναι κοπιώδης	η τροποποίηση των κανόνων είναι εύκολη
η εκτέλεση είναι βήμα-προς-βήμα	η εκτέλεση βασίζεται σε ευριστικούς κανόνες και την λογική
χρειάζονται ολοκληρωμένη πληροφορία	διαχειρίζονται ελλιπή ή ασαφή πληροφορία
αποτελεσματική διαχείριση μεγάλων βάσεων δεδομένων	αποτελεσματική διαχείριση μεγάλων βάσεων γνώσεων
αναπαράσταση και χρήση δεδομένων	αναπαράσταση και χρήση γνώσης

**Πίνακας 6:** Σύγκριση συμβατικών συστημάτων και έμπειρων συστημάτων

## Αρχιτεκτονική

Ένα έμπειρο σύστημα περιλαμβάνει την βάση γνώσης, το μηχανισμό εξαγωγής συμπερασμάτων, τον χώρο εργασίας, το σύστημα επικοινωνίας με τον χρήστη, το σύστημα επεξηγήσεων και το σύστημα βελτίωσης της γνώσης.

### Βάση Γνώσης

Η βάση γνώσης (knowledge base) περιλαμβάνει τις πληροφορίες που απαιτούνται για την κατανόηση, μορφοποίηση και επίλυση του προβλήματος. Περιέχει δύο βασικά στοιχεία:

Τα γεγονότα (όπως για παράδειγμα τα αντικείμενα του προβλήματος και τις σχέσεις ανάμεσα σε αυτά, περιορισμούς, την κατάσταση του προβλήματος).

Την σχετική θεωρία, τους κανόνες που χρησιμοποιούνται για την λύση του προβλήματος.

Η γνώση που εισάγεται στο σύστημα μπορεί να ταξινομηθεί σε τρεις γενικές κατηγορίες, τη «μεταγλωττισμένη» γνώση, την ποιοτική γνώση και την ποσοτική γνώση. Η γνώση που προέρχεται από τους εμπειρογνώμονες του πεδίου, η γνώση που συλλέγεται από έγγραφα, αρχεία, βιβλία, προδιαγραφές κτλ, αποτελεί τη «μεταγλωττισμένη» γνώση. Η ποιοτική γνώση προέρχεται από εμπειρικούς κανόνες, προσεγγιστικές θεωρίες, αιτιολογικά μοντέλα διαδικασιών και κοινή λογική. Η ποσοτική γνώση σχετίζεται με τη χρήση μεθόδων βασισμένων σε μαθηματικές θεωρίες, αριθμητικών τεχνικών κτλ. Η «μεταγλωττισμένη» και η ποιοτική γνώση μπορούν περαιτέρω να ταξινομηθούν σε δύο υποκατηγορίες, τη δηλωτική γνώση και τη διαδικαστική γνώση. Η δηλωτική γνώση προέρχεται από φυσικές ιδιότητες ενώ η διαδικαστική συνδέεται με τεχνικές επίλυσης προβλημάτων. Η ανάπτυξη ενός εμπειρου συστήματος προϋποθέτει τη χρήση όλων των κατηγοριών γνώσης ,αλλά ο όγκος και η χρησιμότητα κάθε κατηγορίας εξαρτάται από το πεδίο εφαρμογής. Ο μηχανικός γνώσης πρέπει να αποφασίσει ποιά γνώση θα αναπαρασταθεί με ποιο σχήμα, ανάλογα με τη φύση της γνώσης και την αποδοτικότητα χρήσης της. Οι πιο κοινές μέθοδοι αναπαράστασης γνώσης είναι:

- η δηλωτική λογική.
- οι κανόνες παραγωγής.
- τα πλαίσια και τα σημασιολογικά δίκτυα.
- τα συμβατικά προγράμματα.

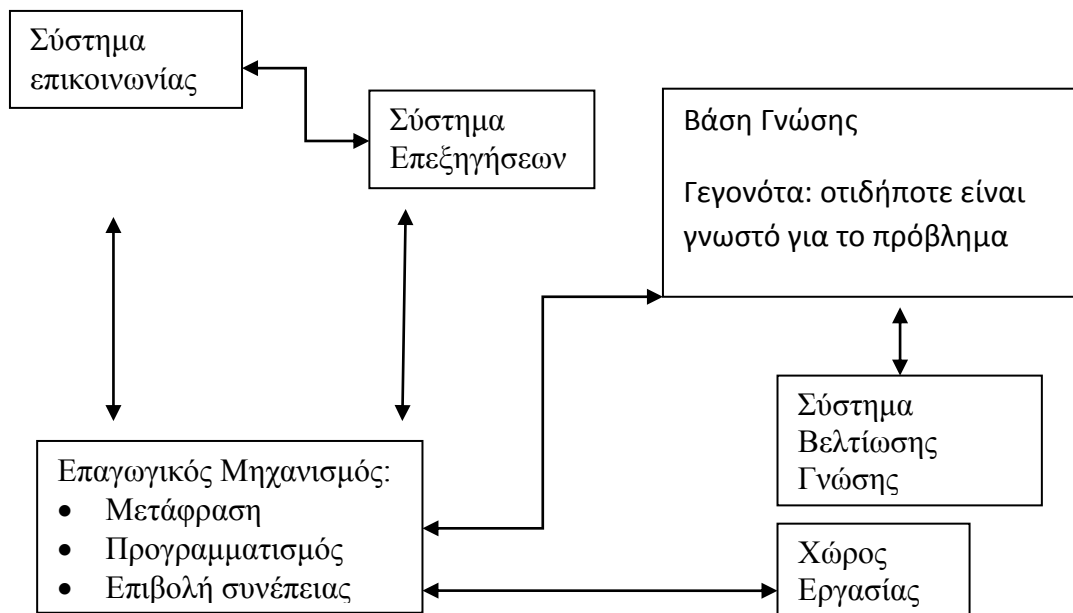
#### Μηχανισμοί εξαγωγής συμπερασμάτων

Οι μηχανισμοί εξαγωγής συμπερασμάτων είναι στρατηγικές ελέγχου, οι οποίες εκτελούν αναζητήσεις στη βάση γνώσης και οδηγούνται σε αποφάσεις. Η γνώση βάσης είναι ο στατικός χώρος και ο μηχανισμός εξαγωγής συμπερασμάτων είναι μια διαδικασία αναζήτησης. Καθώς τα έμπειρα συστήματα επεξεργάζονται σύμβολα, η διαδικασία εξαγωγής συμπερασμάτων μεταχειρίζεται τα σύμβολα επιλέγοντας τους κανόνες, ταιριάζοντας τα σύμβολα των γεγονότων και πυροδοτώντας τους κανόνες, παράγει νέα γεγονότα. Η διαδικασία είναι συνεχής σαν αλυσίδα έως ότου επιτευχθεί ένας καθορισμένος στόχος. Η εξαγωγή συμπερασμάτων μπορεί να πραγματοποιηθεί με δύο τρόπους, την προς τα πίσω και την προς τα εμπρός αλυσίδα εκτέλεσης. Η προς τα πίσω αλυσίδα εκτέλεσης προχωρεί από ένα στόχο πίσω στα γεγονότα για να εξετάσει αν τα γεγονότα αυτά καλύπτουν τις προϋποθέσεις ικανοποίησης του στόχου. Γι' αυτό χαρακτηρίζεται ως «κατευθυνόμενη από το στόχο» (goal-driven). Ο τρόπος αυτός είναι προτιμότερος στην περίπτωση πολλών αρχικών καταστάσεων και λιγότερων καταστάσεων-στόχων. Η προς τα εμπρός αλυσίδα εκτέλεσης προχωρεί από τα δεδομένα προς τους στόχους και για αυτό χαρακτηρίζεται ως «κατευθυνόμενη από τα δεδομένα» (data-driven). Έχοντας δηλαδή μία υπόθεση ως δεδομένο, θεωρείται ότι πραγματοποιείται και το συμπέρασμα της ως στόχος. Ο τρόπος αυτός ενδείκνυται

στις περιπτώσεις που υπάρχουν λίγες αρχικές καταστάσεις και πολλές καταστάσεις-στόχοι. Σε μεγάλα έμπειρα συστήματα, υπάρχουν πολλές αρχικές καταστάσεις και πολλές καταστάσεις-στόχοι. Σε αυτές τις περιπτώσεις, ένας υβριδικός μηχανισμός εξαγωγής συμπερασμάτων είναι κατάλληλος, ξεκινώντας με την προς τα εμπρός εκτέλεση με κενό περιεχόμενο, προσπαθώντας να θέσει τα γεγονότα όπως εμφανίζονται στη βάση γνώσης και στη συνέχεια με προς τα πίσω αλυσίδες να τα πιστοποιήσει ή να τα αναιρέσει.

### Χώρος Εργασίας

Ο χώρος εργασίας (workspace) είναι μία περιοχή της μνήμης του υπολογιστή που χρησιμοποιείται για την περιγραφή του τρέχοντος προβλήματος, καθώς επίσης και για την προσωρινή καταχώριση ενδιάμεσων αποτελεσμάτων. Στο χώρο εργασίας καταγράφονται τα σχέδια που περιγράφουν πώς θα αντιμετωπισθεί το πρόβλημα, η ατζέντα που περιέχει τις μελλοντικές ενέργειες προς εκτέλεση και τις υποθέσεις και εναλλακτικούς τρόπους δράσης που έχει δημιουργήσει το σύστημα μέχρι τη στιγμή αυτή. Για παράδειγμα, σε ένα έμπειρο σύστημα για την διάγνωση και επιδιόρθωση βλαβών αυτοκινήτων, η βάση γνώσης περιέχει τους κανόνες που σχετίζονται με τις βλάβες του αυτοκινήτου. Όταν το αυτοκίνητο παρουσιάσει βλάβη, ο χρήστης εισάγει τα συμπτώματα της βλάβης στον υπολογιστή, τα οποία καταχωρούνται στον χώρο εργασίας. Το σύστημα μπορεί να δημιουργήσει κάποιες υποθέσεις και να ζητήσει από το χρήστη επιπλέον πληροφορίες, οι οποίες και αυτές καταχωρούνται στον χώρο εργασίας.



**Εικόνα:** Αρχιτεκτονική Έμπειρου Συστήματος



## Σύστημα Επικοινωνίας (user interface)

Το σύστημα αυτό χρησιμοποιείται για την επικοινωνία ανάμεσα στον χρήστη και στον υπολογιστή. Το σύστημα βασίζεται σε μία γλώσσα για την φιλική επικοινωνία με τον χρήστη. Η γλώσσα αυτή θα μπορούσε να είναι μία φυσική γλώσσα και σε ορισμένες περιπτώσεις υποστηριζόμενη από μενού επιλογών και γραφικά.

## Σύστημα Επεξηγήσεων

Το σύστημα επεξηγήσεων (explanation system) είναι υπεύθυνο για την παροχή απαντήσεων στις παρακάτω ερωτήσεις:

- Γιατί το έμπειρο σύστημα έθεσε μία ερώτηση προς τον χρήστη;
- Πως το έμπειρο σύστημα συνήγαγε ένα συμπέρασμα;
- Γιατί μία εναλλακτική λύση απορρίφθηκε;
- Ποιο είναι το σχέδιο για την εύρεση της λύσης; (για παράδειγμα τι μένει να αποδειχθεί για να προσδιοριστεί η διάγνωση;)

## Σύστημα Βελτίωσης της Γνώσης

Οι εμπειρογνώμονες είναι σε θέση να εξετάσουν την απόδοσή τους, να μάθουν και να βελτιώσουν το τρόπο αντιμετώπισης των προβλημάτων. Κατά αντιστοιχία, το έμπειρο σύστημα μπορεί να διαθέτει τον μηχανισμό που είναι υπεύθυνος για την ανάλυση των λόγων επιτυχίας ή αποτυχίας του συστήματος. Με τον τρόπο αυτό θα μπορούσαν να προκύψουν βελτιώσεις της βάσης γνώσης. Το σύστημα αυτό προς το παρόν δεν είναι διαθέσιμο στα υπάρχοντα εμπορικά έμπειρα συστήματα.

## Ταξινόμηση

---

Η ταξινόμηση των έμπειρων συστημάτων είναι ανάλογη με την ταξινόμηση των Σ.Υ.Α. Έτσι, διακρίνουμε τρία επίπεδα τεχνολογίας:

Εξειδικευμένα έμπειρα συστήματα χρησιμοποιούνται για την παροχή συμβουλών μίας συγκεκριμένης θεματικής περιοχής.

Κελύφη - ένα κέλυφος είναι ένα πακέτο το οποίο επιτρέπει την σύντομη και εύκολη ανάπτυξη έμπειρων συστημάτων. Ένα κέλυφος παρέχει τα περισσότερα υποσυστήματα ενός έμπειρου συστήματος, εκτός από την βάση γνώσης.

Εργαλεία έμπειρων συστημάτων - αποτελούν το χαμηλότερο επίπεδο. Πρόκειται για εργαλεία που διευκολύνουν την ανάπτυξη είτε εξειδικευμένων είτε κελυφών έμπειρων συστημάτων.

### *Εργαλεία ανάπτυξης έμπειρων συστημάτων*

---

Διάφορες γλώσσες προγραμματισμού έχουν αναπτυχθεί επηρεασμένες από τον τρόπο σκέψης της Τεχνητής Νοημοσύνης. Οι γλώσσες αυτές (που αναφέρονται ως γλώσσες 5ης γενιάς) έχουν χρησιμοποιηθεί για την ανάπτυξη έμπειρων συστημάτων, χωρίς αυτό να σημαίνει ότι οι συμβατικές γλώσσες προγραμματισμού δεν μπορούν να χρησιμοποιηθούν για τον ίδιο λόγο. Παρακάτω περιγράφονται συνοπτικά τα βασικά εργαλεία που μπορούν να χρησιμοποιηθούν για την ανάπτυξη ενός έμπειρου συστήματος.

### *Γλώσσες 5ης γενιάς*

---

Οι γλώσσες 5ης γενιάς παρέχουν έναν αποτελεσματικό τρόπο επεξεργασίας συμβόλων. Τα χαρακτηριστικότερα παραδείγματα αυτής της κατηγορίας γλωσσών είναι η LISP και η PROLOG.

### *Πλεονεκτήματα*

---

Μονιμότητα γνώσης. Τα έμπειρα συστήματα δεν ξεχνούν, σε αντίθεση με τους ανθρώπους.

Αναπαραγωγή. Η δημιουργία πολλαπλών αντιγράφων ενός έμπειρου συστήματος είναι μια εύκολη και φθηνή διαδικασία. Αντίθετα η εκπαίδευση ανθρώπινου προσωπικού για την ανάλογη εξειδίκευση είναι μια χρονοβόρα και ακριβή διαδικασία.

Αποδοτικότητα-Κόστος. Τα έμπειρα συστήματα μπορούν να αυξήσουν την παραγωγικότητα και να μειώσουν το κόστος προσωπικού. Τα έμπειρα συστήματα έχουν ένα υψηλό κόστος παραγωγής αλλά ένα χαμηλό κόστος λειτουργίας. Εξάλλου το συνολικό κόστος παραγωγής και συντήρησης μπορεί να μοιραστεί σε πολλούς χρήστες.

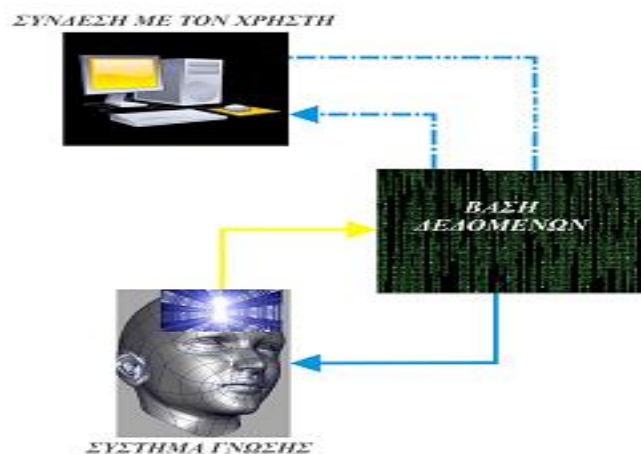
Συνέπεια. Τα έμπειρα συστήματα συμπεριφέρονται με παρόμοιο τρόπο σε παρόμοια προβλήματα. Οι άνθρωποι επηρεάζονται από πολλούς παράγοντες οι οποίοι μειώνουν την αντικειμενικότητά τους.

Τεκμηρίωση γνώσης. Ένα έμπειρο σύστημα παρέχει μια μόνιμη τεκμηρίωση της γνώσης και των κανόνων του γνωστικού πεδίου. Σε πολλές περιπτώσεις άλλωστε χρησιμοποιούνται τα έμπειρα συστήματα για την εκπαίδευση ανθρώπων.

Πληρότητα. Ένα έμπειρο σύστημα μπορεί σε μικρό χρόνο να εξετάσει πολλές πιθανές λύσεις και να λάβει υπόψη του όλα τα δεδομένα εισόδου.

Έγκαιρη πληροφορία. Η έγκαιρη πληροφορία μπορεί να βοηθήσει στην αποφυγή λαθών.

Εύρος γνώσης. Σε ένα έμπειρο σύστημα συνδυάζεται η γνώση και εμπειρία πολλών εμπειρών ανθρώπων. Η γνώση που αποθηκεύεται έχει μεγαλύτερο εύρος από τη γνώση ενός μεμονωμένου έμπειρου.



**Εικόνα:** Σύστημα λήψης απόφασης

Η υλοποίησή του πραγματοποιήθηκε με σκοπό την προσομοίωση της συμπεριφοράς και των ενεργειών των ανθρώπων οι οποίοι έχουν εμπειρία στην παραγωγή. Οι δύο άξονες ανάπτυξης του συστήματος γνώσης είναι:

- η γνωσιακή βάση: χρησιμοποιεί τη συλλογιστική των περιπτώσεων, η οποία μοντελοποιεί τις συνθήκες υπό τις οποίες εξελίχθηκαν οι καλλιέργειες στο παρελθόν και αναπαριστάται με τη μορφή δηλωτικών και διαδικαστικών κανόνων (if συνθήκες then συμπέρασμα/ενέργειες).
- το κέλυφος JESS: είναι μια μηχανή εκτέλεσης των κανόνων βασισμένο στο εργαλείο CLIPS, χρησιμοποιεί μία εξελιγμένη έκδοση του αλγορίθμου Rete ως αλγόριθμο ταυτοποίησης των κανόνων με τα γεγονότα της λειτουργικής μνήμης

και συνεργάζεται άριστα με τη JAVA, μέσω της οποίας επιτυγχάνεται η διασύνδεσή του με τη βάση δεδομένων και με το λοιπό σύστημα.

Οι είσοδοι του συστήματος είναι:

- ✓ Μια σειρά δεδομένων (απαιτείται βελτιστοποίηση).

Οι έξοδοι του συστήματος είναι:

- ✓ Το είδος της ενέργειας.
- ✓ Η προτεινόμενη ενέργεια.

Για το υποσύστημα απαιτεί περαιτέρω βελτιστοποίηση ως προς την επιλογή των εισόδων και επιπρόσθετη αξιολόγηση για να διερευνηθεί αν παράγει τα επιθυμητά αποτελέσματα.

