



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή εργασία

**Τίτλος: Ανάπτυξη διαδικτυακής εφαρμογής βασισμένη
σε τεχνολογία AJAX**

Ανδρέας Παύλου (ΑΜ: 1078)

Ηράκλειο – Ημερομηνία

Επόπτης Καθηγητής: Μαλάμος Αθανάσιος

Υπεύθυνη Δήλωση: Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Εφαρμοσμένης Πληροφορικής και Πολυμέσων του Τ.Ε.Ι. Κρήτης.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για την ψυχολογική υποστήριξη τους καθ' όλη της διάρκειας συγγραφής της εργασίας αυτής.

Abstract

The following paper's objective is to study the ajax programming technology. As you read through the paper, you will see an extensive analysis of ajax programming, as well as a short analysis of the programming languages that consist AJAX.

In the following chapters of this paper you will see step by step the construction of the multimedia lab of the Technical Educational Institution of Crete's web site using the AJAX programming procedures, by analyzing the source code.

In the last chapter the conclusions of the ajax programming technique are mentioned with a few proposals for further improvement.

Σύνοψη

Η παρακάτω πτυχιακή εργασία έχει ως αντικείμενο μελέτης την τεχνική προγραμματισμού AJAX. Στις σελίδες της παρούσας εργασίας περιγράφετε η λειτουργία της τεχνικής AJAX, καθώς και μια σύντομη αναφορά στις επιμέρους γλώσσες προγραμματισμού που χρησιμοποιούνται συνδυαστικά με αποτέλεσμα την AJAX.

Στα τελευταία κεφάλαια της εργασίας πραγματοποιείτε η υλοποίηση της ιστοσελίδας του εργαστηρίου πολυμέσων ΤΕΙ Κρήτης κάνοντας χρήση της τεχνικής AJAX. Παραθέτεται ο πηγαίος κώδικας της ιστοσελίδας, ο οποίος και αναλύετε κάνοντας έτσι ευκολότερη την κατανόηση του. Τέλος παραθέτονται τα συμπεράσματα της Πτυχιακής εργασίας, καθώς και προτάσεις για μελλοντικές εργασίες .

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Ευχαριστίες	5
Abstract	6
Συνοψη	Error! Bookmark not defined.
1. Εισαγωγή.....	11
1.1 Περίληψη	12
1.2 Κινητρο για την διεξαγωγή της εργασίας	13
1.3 Σκοπος και στοχοι εργασίας.....	13
1.4 Δομη της εργασίας	14
2. Ανάλυση των τεχνολογιών της AJAX	15
2.1 Ανάλυση της τεχνολογίας HTML	16
2.1.1 HTML Markup	16
2.1.1.1 HTML Elements	17
2.2 Ανάλυση των CSS.....	18
2.2.1 Σύνταξη των CSS.....	19
2.2.2 Χρήση των CSS	20
2.2.3 Πηγές πληροφοριών στα CSS	20
2.3 JavaScript.....	22
2.3.1 Χαρακτηριστικά της JavaScript	22
2.3.1.1 Αναγκαστική και δομημένη	22
2.3.1.2 Δυναμική.....	23
2.3.1.3 Λειτουργίες	23
2.3.1.4 Βασισμένη στα πρωτότυπα	24
2.3.1.5 Διάφορα.....	24

2.3.2 Χρήση της JavaScript στις ιστοσελίδες.....	25
2.4 XML	27
2.4.1 Ανάλυση των χαρακτηριστικών της XML	27
2.4.2 Χαρακτήρες και διαφυγή	29
2.4.3 XMLHttpRequest.....	30
2.4.3.1 Η αίτηση HTTP.....	31
2.5 Java Servlets	34
2.5.1 Κύκλος ζωής ενός Java Servlet.....	35
2.5.2 Servlet Containers	36
3. Παρουσίαση του site του McLab	38
3.1 το interface του site.....	39
3.2 Λειτουργικότητα του site	40
3.3 Ανάλυση του κώδικα	42
3.3.1 Εισαγωγή	42
3.3.2 Ο κώδικας HTML.....	43
3.3 ο κώδικας JavaScript	47
3.4 Το αρχείο XML	55
4. Συμπεράσματα για την τεχνολογία AJAX	68
4.1 Πλεονεκτήματα.....	68
4.2 Μειονεκτήματα	68
Βιβλιογραφία.....	70

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1 : Η αρχική εμφάνιση της ιστοσελίδας.....	39
Εικόνα 2 : Η ιστοσελίδα αφού πατήσουμε το κουμπί projects.....	40
Εικόνα 3 : Η ιστοσελίδα αφού επιλέξουμε κάποιον από τους τίτλους.....	41

1. Εισαγωγή

Η τεχνολογία Ajax¹, που μερικές φορές γράφεται ως AJAX (συντομογραφία για το Asynchronous JavaScript and XML²), είναι μια ομάδα αλληλένδετων τεχνικών ανάπτυξης ιστοσελίδων, που χρησιμοποιούνται από τον υπολογιστή-πελάτη(client-side), για τη δημιουργία διαδραστικών διαδικτυακών εφαρμογών ή εφαρμογές διαδικτύου. Με τη τεχνική Ajax, οι διαδικτυακές εφαρμογές μπορούν να ανακτούν δεδομένα από το διακομιστή ασύγχρονα στο παρασκήνιο χωρίς να παρεμβαίνουν στην εμφάνιση και τη συμπεριφορά των υφιστάμενων σελίδων. Η χρήση της τεχνικής Ajax έχει οδηγήσει σε αύξηση των διαλογικών ή διαδραστικών διεπαφών στις δυναμικές ιστοσελίδες καθώς και στη βελτίωση της ποιότητας των υπηρεσιών του διαδικτύου, λόγω της ασύγχρονης λειτουργίας του. Τα δεδομένα συνήθως ανακτώνται χρησιμοποιώντας το αντικείμενο XMLHttpRequest³. Παρά την ονομασία, η χρήση του JavaScript και XML δεν απαιτείται στην πραγματικότητα, ούτε τα αιτήματα που κάνουμε πρέπει να είναι ασύγχρονα.

Η τεχνική Ajax δεν είναι μια τεχνολογία από μόνη της, αλλά μια ομάδα τεχνολογιών. Χρησιμοποιεί ένα συνδυασμό:

- HTML και CSS για τη σήμανση και για την εμφάνιση.
- Πρόσβαση στην DOM⁴ με την JavaScript, για να εμφανίζονται και να αλληλεπιδρούν δυναμικά με τις πληροφορίες που παρουσιάζονται.
- Μια μέθοδο για την ασύγχρονη ανταλλαγή δεδομένων μεταξύ browser και server, αποφεύγοντας έτσι τις επαναφορτώσεις των σελίδων. Το αντικείμενο αυτό είναι το XMLHttpRequest (XHR) και χρησιμοποιείται τις περισσότερες φορές, αλλά μερικές φορές ένα αντικείμενο iframe⁵ ή μια δυναμικά προστιθέμενη <script> ετικέτα αντί αυτού.
- Μια μορφή για τα δεδομένα τα οποία αποστέλλονται στον browser. Οι πιο συνηθισμένες μορφές είναι είτε XML, είτε προ-μορφοποιημένη HTML, είτε απλό

¹ Asynchronous JavaScript and XML http://en.wikipedia.org/wiki/AJAX_ajax_programing

² Extensible Markup Language <http://en.wikipedia.org/wiki/XML>

³ Ένα αντικείμενο του DOM <http://en.wikipedia.org/wiki/XMLHttpRequest>

⁴ Document object Model http://en.wikipedia.org/wiki/Document_Object_Model

⁵ Αντικείμενο της HTML http://en.wikipedia.org/wiki/HTML_element#Frames

κείμενο, ή JavaScript Object Notation (JSON)⁶ αλλά θα μπορούσαν να δημιουργούνται δυναμικά από κάποιον υπολογιστή-εξυπηρετητή(server-side scripting).

Οι τεχνικές για την ασύγχρονη φόρτωση του περιεχομένου χρονολογούνται από τα μέσα της δεκαετίας του 1990. Τα Java applets εισήχθησαν στην πρώτη έκδοση της γλώσσας Java, το 1995 και επέτρεπαν σε client-side κώδικα να φορτώνει τα δεδομένα ασύγχρονα από τον web server αφού αυτή η ιστοσελίδα φορτωθεί. Το 1996, ο Internet Explorer παρουσίασε το στοιχείο IFrame στην HTML, το οποίο επέτρεπε, επίσης, να φορτώνει ασύγχρονα τα δεδομένα. Το 1999, η Microsoft δημιούργησε το XMLHTTP ActiveX control στον Internet Explorer 5, το οποίο υποστηρίζεται τώρα από τον Mozilla, το Safari και άλλα προγράμματα περιήγησης, τα οποία υποστηρίζουν και το αντικείμενο XMLHttpRequest. Ωστόσο, η τεχνική AJAX έγινε ευρύτερα γνωστή, αφού χρησιμοποιήθηκε από το Gmail (2004) και το Google Maps (2005).

Ο όρος "Ajax" επινοήθηκε το 2005 από τον Jesse James Garrett, ο οποίος σκέφτηκε τον όρο "Ajax", ενώ ήταν στο ντους, όταν συνειδητοποίησε την ανάγκη για ένα συνοπτικό όνομα που να εκπροσωπεί την ακολουθία των τεχνολογιών που είχε να προτείνει σε έναν πελάτη.

Στις 5 Απριλίου του 2006 το World Wide Web Consortium (W3C)⁷ κυκλοφόρησε το πρώτο σχέδιο προδιαγραφής για το αντικείμενο, σε μια προσπάθεια να δημιουργηθεί ένα πρότυπο επίσημης ιστοσελίδας.

1.1 Περίληψη

Στην εργασία που ακολουθεί θα δούμε την ανάπτυξη μιας διαδικτυακής εφαρμογής βασισμένη στην τεχνολογία AJAX. Πιο συγκεκριμένα θα υλοποιηθεί η κατασκευή της ιστοσελίδας του εργαστήριου πολυμέσων του ΤΕΙ Κρήτης κάνοντας χρήση της τεχνολογίας AJAX. Στα κεφάλαια που ακολουθούν θα αναλυθεί η τεχνική AJAX καθώς και οι επιμέρους γλώσσες προγραμματισμού που θα χρησιμοποιηθούν.

⁶ Τύπος αναπαράστασης δεδομένων http://en.wikipedia.org/wiki/JavaScript_Object_Notation

⁷ Διεθνής οργανισμός προτύπων <http://www.w3.org/>

1.2 Κίνητρο για την διεξαγωγή της εργασίας

Ένα από τα προβλήματα που αντιμετωπίζουν οι χρήστες κατά την πλοήγηση τους στις διάφορες σελίδες του διαδικτύου, είναι η αναμονή επαναφόρτωσης όλης της ιστοσελίδας σε οποιαδήποτε αλλαγή του περιεχομένου αυτής. Με την τεχνολογία AJAX μπορεί να αντιμετωπιστεί αυτό το πρόβλημα καθώς τα δεδομένα και οι πληροφορίες της ιστοσελίδας ανακτούνται από το διακομιστή ασύγχρονα στο παρασκήνιο, χωρίς να επηρεάζετε η εμφάνιση και η συμπεριφορά της ιστοσελίδας. Έτσι οι χρήστες μπορούν να πλοηγούνται και να έχουν πρόσβαση γρήγορα σε όλο το περιεχόμενο της ιστοσελίδας.

1.3 Σκοπός και στόχοι εργασίας

Μέσα από την μελέτη της παρούσας πτυχιακής εργασίας, ο αναγνώστης της θα μπορέσει να κατανοήσει την λειτουργία της τεχνικής AJAX, αλλά και να την εφαρμόσει, μέσω της μελέτης του πηγαίου κώδικα της ιστοσελίδας του εργαστήριου πολυμέσων που παρατίθεται στη παρούσα εργασία αλλά και μέσω των παραδειγμάτων που αναφέρονται και αναλύονται καθ' όλη τη διάρκεια της μελέτης της τεχνικής AJAX.

1.4 Δομή της εργασίας

Το πρώτο κεφάλαιο της εργασίας είναι το κεφάλαιο της εισαγωγής. Εδώ υπάρχει μια περίληψη της εργασίας που ακολουθεί καθώς και άλλα υποκεφάλαια στα οποία αναφέρονται το κίνητρο για την διεξαγωγή της παρούσας εργασίας, αλλά και τον σκοπό και τους στόχους της εργασίας αυτής.

Στο δεύτερο κεφάλαιο, το οποίο είναι και ένα από τα κυριότερα της εργασίας αυτής, αναλύονται οι επιμέρους γλώσσες προγραμματισμού που συνιστούν την τεχνική AJAX. Πιο συγκεκριμένα γίνεται αναφορά στην γλώσσες HTML, CSS, Javascript και XML. Προς το τέλος του δευτέρου κεφαλαίου αναλύετε το αντικείμενο XMLHttpRequest, το οποίο και αποτελεί τον πυρήνα της τεχνικής AJAX.

Στο τρίτο κεφάλαιο γίνεται η παρουσίαση, η παράθεση και ανάλυση του κώδικα της ιστοσελίδας του εργαστηρίου πολυμέσων του ΤΕΙ Κρήτης. Πιο συγκεκριμένα αναλύετε ο κώδικας της γλώσσας HTML, που αποτελεί την βάση της ιστοσελίδας, της JavaScript, που μέσα της υπάρχει ο πυρήνας της AJAX, και τέλος το αρχείο XML, το οποίο περιέχει όλες της πληροφορίες της ιστοσελίδας.

Στο τέταρτο και τελευταίο κεφάλαιο παραθέτονται τα συμπεράσματα για την τεχνική AJAX μαζί με τα πλεονεκτήματα αλλά και τα μειονεκτήματα της.

2. Ανάλυση των τεχνολογιών της AJAX

Ο όρος Ajax έχει εφευρεθεί για να αντιπροσωπεύσει μια ευρεία ομάδα διαδικτυακών τεχνολογιών που μπορούν να χρησιμοποιηθούν για την υλοποίηση μιας εφαρμογής διαδικτύου που επικοινωνεί με ένα διακομιστή στο παρασκήνιο, χωρίς να παρεμβαίνουν στην τρέχουσα κατάσταση της σελίδας. Στον όρο Ajax, ο Jesse James Garrett εξήγησε ότι απαιτούνται οι ακόλουθες τεχνολογίες:

- XHTML⁸ και CSS⁹ για την παρουσίαση
- Το Document Object Model για δυναμική απεικόνιση και αλληλεπίδραση με τα δεδομένα
- Η XML και XSLT για την ανταλλαγή, χειραγώγηση και απεικόνιση, των δεδομένων
- Το αντικείμενο XMLHttpRequest για την ασύγχρονη επικοινωνία
- Τη JavaScript για να φέρουν όλες τις τεχνολογίες αυτές μαζί

Από τότε, ωστόσο, υπήρξαν ορισμένες εξελίξεις στις τεχνολογίες που χρησιμοποιούνται σε μια εφαρμογή Ajax, καθώς και στον ορισμό του όρου Ajax. Ειδικότερα, διαπιστώθηκε ότι:

- Η JavaScript δεν είναι η μόνη client-side scripting γλώσσα που μπορεί να χρησιμοποιηθεί για την εκτέλεση μιας εφαρμογής Ajax. Άλλες γλώσσες, όπως η VBScript είναι επίσης σε θέση να διεκπεραιώσει την απαιτούμενη λειτουργικότητα.
- Η XML δεν απαιτείται για την ανταλλαγή δεδομένων και, συνεπώς, η XSLT δεν απαιτείται για το χειρισμό των δεδομένων. Η JavaScript Object Notation (JSON) συχνά χρησιμοποιείται ως εναλλακτική μορφή για την ανταλλαγή δεδομένων, αν και άλλες

⁸ Extensible Hypertext Markup Language <http://en.wikipedia.org/wiki/XHTML>

⁹ Cascading Style Sheets <http://en.wikipedia.org/wiki/CSS>

μορφές, όπως ήδη διαμορφωμένη HTML ή απλά κείμενα, μπορούν επίσης να χρησιμοποιηθούν.

2.1 Ανάλυση της τεχνολογίας HTML

Ο όρος HTML, που σημαίνει Hyper Text Markup Language, είναι η κυρίαρχη γλώσσα σήμανσης για τις ιστοσελίδες. Παρέχει όλα εκείνα τα μέσα που χρειάζονται για τη δημιουργία δομημένων εγγράφων, δηλώνοντας τις διαρθρωτικές σημασιολογίες στο κείμενο όπως επικεφαλίδες, παραγράφους, καταλόγους κλπ., όπως επίσης και για τις συνδέσεις, τα quotes και τα άλλα στοιχεία. Επιτρέπει σε εικόνες και αντικείμενα που πρέπει να ενταχθούν, να μπορούν να χρησιμοποιηθούν για τη δημιουργία διαδραστικών φορμών. Τα στοιχεία της HTML αποτελούνται από "ετικέτες" που περιβάλλονται από γωνιακές αγκύλες εντός του περιεχομένου της ιστοσελίδας και μπορούν να περιλαμβάνουν ή να φορτώνουν scripts σε γλώσσες όπως η JavaScript, οι οποίες επηρεάζουν τη συμπεριφορά των λειτουργιών της HTML.

2.1.1 HTML Markup

Η HTML markup αποτελείται από διάφορα βασικά κομμάτια όπως, τα HTML elements (και τις ιδιότητές τους), σε τύπους δεδομένων βασισμένων σε χαρακτήρες, σε αναφορές χαρακτήρων και σε αναφορές οντοτήτων. Ένα σημαντικό στοιχείο της, είναι η δήλωση τύπου εγγράφου (document type declaration), η οποία καθορίζει το Document Type Definition. Από την HTML 5 και έπειτα, το Document Type Definition δεν χρειάζεται να προσδιορίζετε.

Το πρόγραμμα Hello World είναι ένα σύνθημα πρόγραμμα ηλεκτρονικού υπολογιστή που χρησιμοποιείται για την εκμάθηση των διαφορών γλωσσών προγραμματισμού, γλωσσών scripting, γλωσσών σήμανσης. Στο παρακάτω παράδειγμα έχει γραφτεί στη γλώσσα HTML και αποτελείται από 9 γραμμές κώδικα :

```
<!doctype html>  
  
<html>  
  
  <head>
```



```
<title>Hello HTML</title>

</head>

<body>

  <p>Hello World!</p>

</body>

</html>
```

Στο παραπάνω παράδειγμα βλέπουμε στη πρώτη γραμμή του εγγράφου την δήλωση `<!doctype html>`. Έτσι το πρόγραμμα περιήγησης αντιλαμβάνεται ότι το έγγραφο που ακολουθεί πρόκειται για ένα HTML έγγραφο έτσι ώστε να μπορέσει να το χειριστεί κατάλληλα. Η δήλωση `<html>` και `</html>` περιγράφει την έναρξη και το τέλος της ιστοσελίδας, ενώ το κείμενο μεταξύ των ετικετών `<body>` και `</body>` είναι το ορατό περιεχόμενο της ιστοσελίδας, εκτός από την ετικέτα `<title> Hello HTML </title>` η οποία περιγράφει τον τίτλο της καρτέλας του προγράμματος περιήγησης.

2.1.1.1 HTML Elements

Τα στοιχεία της HTML (HTML Elements) είναι τα βασικά συστατικά για την HTML markup. Τα στοιχεία της HTML έχουν δύο βασικές ιδιότητες: τα χαρακτηριστικά και το περιεχόμενο. Κάθε χαρακτηριστικό ενός στοιχείου HTML όπως και το περιεχόμενο του κάθε στοιχείου HTML έχει ορισμένους περιορισμούς που πρέπει να τηρούνται σε ένα έγγραφο έτσι ώστε αυτό να θεωρείται έγκυρο. Ένα στοιχείο έχει συνήθως μια ετικέτα έναρξης (π.χ. `<element-name>`) και μια ετικέτα τέλους (π.χ. `</element-name>`). Οι ιδιότητες του στοιχείου περιέχονται στην ετικέτα έναρξης ενώ το περιεχόμενο τους βρίσκεται μεταξύ των ετικετών (π.χ. `<element-name attribute="value">Content</element-name>`). Ορισμένα στοιχεία, όπως `
`, δεν έχουν κανένα περιεχόμενο και δεν πρέπει να έχουν ετικέτα κλεισίματος. Παρακάτω παραθέτονται οι διάφορες κατηγορίες των στοιχείων σήμανσης που χρησιμοποιούνται σε ένα έγγραφο HTML.

- Τα structural markup τα οποία περιγράφουν το σκοπό του κειμένου. Για παράδειγμα, το `<h2>Golf</h2>` καθιερώνει το "Golf" ως ένα δεύτερου επίπεδου τίτλο, ο οποίος θα μεταφραστεί στο πρόγραμμα περιήγησης με τον κατάλληλο τρόπο. Τα structural markup δεν υποδηλώνουν καμία συγκεκριμένη απόδοση, αλλά τα περισσότερα προγράμματα περιήγησης έχουν τυποποιηθεί με ένα προεπιλεγμένο στυλ για τη μορφοποίηση των διαφόρων στοιχείων. Το κείμενο μπορεί διαμορφωθεί περαιτέρω με τα CSS.

- Το Presentation markup το οποίο περιγράφει την εμφάνιση του κειμένου, ανεξάρτητα από τη λειτουργία του. Για παράδειγμα ` boldface </ b>` υποδηλώνει ότι στις οπτικές συσκευές εξόδου θα πρέπει η λέξη "boldface" να εμφανιστεί με έντονο χρώμα, αλλά δεν παρέχει καμία ένδειξη για τις συσκευές που δεν είναι η θέση να το κάνουν αυτό (όπως ακουστικές συσκευές που διαβάζουν το κείμενο φωναχτά) τι θα πρέπει να κάνουν. Στην περίπτωση των ` bold </ b>` και `<i> italic </ i>`, υπάρχουν στοιχεία τα οποία συνήθως έχουν μία ισοδύναμη οπτική απόδοση, αλλά έχουν περισσότερο σημασιολογικό χαρακτήρα, όπως το ` strong emphasis </ strong>` και το ` emphasis</ em>` αντίστοιχα.
- Το Hypertext Markup το οποίο συνδέει τμήματα του εγγράφου σε άλλα έγγραφα. HTML. Για να δημιουργηθεί μια υπέρ-σύνδεση (hyperlink) στη ροή του κειμένου απαιτείται η χρήση ενός βασικού στοιχείου: `<a> Wikipedia </ a>`. Ωστόσο, το χαρακτηριστικό href πρέπει να έχει μια έγκυρη διεύθυνση URL για να λειτουργήσει. Έτσι για παράδειγμα το HTML markup, ` Wikipedia </ a>`, θα καταστήσει τη λέξη "Wikipedia" ως hyperlink. Ένας άλλος τρόπος για τη δημιουργία ενός hyperlink είναι να χρησιμοποιηθεί μια εικόνα. Για να γίνει αυτό θα πρέπει η ετικέτα χρησιμοποιήσει την ακόλουθη σύνταξη: ` </ a>`.

2.2 Ανάλυση των CSS

Τα Cascading Style Sheets ή αλλιώς εν συντομογραφία CSS είναι μια γλώσσα μορφοποίησης των φύλλων, που χρησιμοποιείται για να περιγράψει τη σημασιολογική παρουσίαση (δηλαδή, την εμφάνιση και μορφοποίηση) από ένα έγγραφο γραμμένο σε μια markup language. Η πιο κοινή εφαρμογή της είναι η μορφοποίηση ιστοσελίδων γραμμένες σε HTML και XHTML, αλλά η γλώσσα μπορεί να εφαρμοστεί σε οποιοδήποτε είδος εγγράφου XML.

Η CSS είναι κυρίως σχεδιασμένη ώστε να επιτρέπει το διαχωρισμό του περιεχομένου του εγγράφου (γραμμένο σε HTML ή παρόμοια markup language) από την παρουσίαση του εγγράφου, συμπεριλαμβανομένων των στοιχείων όπως, το χρώμα, τις γραμματοσειρές, και τη διάταξη. Αυτός ο διαχωρισμός μπορεί να βελτιώσει την προσβασιμότητα του περιεχομένου, να παρέχει περισσότερη ευελιξία και έλεγχο στη συγγραφή των χαρακτηριστικών παρουσίασης έτσι ώστε πολλές σελίδες να μπορούν να μοιράζονται μια ενιαία μορφοποίηση, και να μειώσει την πολυπλοκότητα και την επανάληψη του διαρθρωτικού περιεχομένου. Η CSS μπορεί επίσης να επιτρέψει στην ίδια σελίδα σήμανσης να παρουσιάζεται σε διαφορετικά στυλ για τις διαφορετικές μεθόδους επεξεργασίας, όπως στην οθόνη, στην εκτύπωση, με φωνή (όταν

διαβάζεται από μια ομιλία με βάση το πρόγραμμα περιήγησης ή ανάγνωσης οθόνης). Αν και ο συντάκτης του εγγράφου συνδέει τυπικά το έγγραφο με ένα φύλλο στυλ CSS, ο αναγνώστης μπορεί να χρησιμοποιήσει ένα διαφορετικό στυλ, για παράδειγμα στον υπολογιστή του, παρακάμπτοντας έτσι το στυλ που ο συγγραφέας έχει καθορίσει.

Η CSS έχει προκαθορίσει ένα σύστημα προτεραιοτήτων, που καθορίζει πιο στυλ υπερισχύει εάν περισσότερα από ένα ανταγωνίζονται ένα συγκεκριμένο στοιχείο. Σε αυτό το λεγόμενο cascade, οι προτεραιότητες και οι βαρύτητες υπολογίζονται και αποδίδονται σε κανόνες, έτσι ώστε τα αποτελέσματα να είναι προβλέψιμα.

Οι προδιαγραφές των CSS τηρούνται από το World Wide Web Consortium (W3C) ενώ ο τύπος των μέσων ενημέρωσης (τύπος MIME) text / css έχει καταχωρηθεί για χρήση με τα CSS από το RFC 2318 (Μάρτιος 1998).

2.2.1 Σύνταξη των CSS

Η CSS έχει απλή σύνταξη και χρησιμοποιεί μια σειρά αγγλικών λέξεων-κλειδιά για να καθορίσει τις ονομασίες των διαφόρων ιδιοτήτων των στυλ.

Ένα φύλλο στυλ αποτελείται από μια λίστα των κανόνων, όπου κάθε κανόνας ή σεντ κανόνων αποτελείται από έναν ή περισσότερους επιλογείς και μια μπλοκ δήλωση. Μια μπλοκ δήλωση αποτελείται από μια λίστα των δηλώσεων και κάθε δήλωση με τη σειρά της αποτελείται από ένα property, μια άνω και κάτω τελεία (:), μια τιμή, και μετά μια άνω τελεία (;).

Στα CSS, οι επιλογείς (selectors) χρησιμοποιούνται για να δηλώσουν σε ποια από τα στοιχεία markup εφαρμόζεται ένα στυλ, σαν ένα είδος έκφρασης συσχέτισης. Οι επιλογείς μπορούν να ισχύουν για όλα τα στοιχεία ενός συγκεκριμένου τύπου, ή μόνο για εκείνα τα στοιχεία που εμφανίζουν ένα συγκεκριμένο χαρακτηριστικό. Τα στοιχεία μπορούν να συνοδεύονται, ανάλογα με τον τρόπο που τοποθετούνται το ένα σε σχέση με το άλλο στον κώδικα σήμανσης, ή ανάλογα τον τρόπο που βρίσκονται εντός του μοντέλου αντικειμένου εγγράφου (document object model).

Οι ψευδο-κλάσεις είναι μια άλλη μορφή προδιαγραφών που χρησιμοποιούνται στα CSS για να προσδιοριστούν τα στοιχεία σήμανσης, καθώς και σε ορισμένες περιπτώσεις, συγκεκριμένες ενέργειες του χρήστη, ισχύει ένα συγκεκριμένο μπλοκ δήλωσης. Ένα συχνό παράδειγμα που χρησιμοποιείται η ψευδο-κλάση είναι η `: hover` η οποία εφαρμόζει ένα στυλ μόνο όταν ο χρήστης «δείχνει» στο ορατό στοιχείο, συνήθως κρατώντας τον κέρσορα του ποντικιού πάνω από αυτό. Άλλες ψευδο-κλάσεις και ψευδο-στοιχεία είναι, για παράδειγμα, το `:first-line`, το `:visited`, το `:before` κτλ.

Μια ψευδο-κλάση επιλέγει ολόκληρα στοιχεία, όπως το `:link` ή το `:visited`, λαμβάνοντας υπόψη της ότι ένα ψευδο-στοιχείο κάνει μια επιλογή που μπορεί να αποτελείται από επιμέρους στοιχεία, όπως `:first-line` ή `:first-letter`.

Οι επιλογείς μπορούν να συνδυάζονται και με άλλους τρόπους, ιδίως στη CSS 2.1, για να επιτευχθεί μεγαλύτερη ακρίβεια και ευελιξία.

Παρακάτω φαίνεται ένα παράδειγμα το οποίο αθροίζει όλους τους κανόνες που αναφέραμε παραπάνω:

```
selector [, selector2, ...]:pseudo-class {  
property: value;  
}  
/* comment*/
```

2.2.2 Χρήση των CSS

Πριν από τα CSS, σχεδόν όλα τα χαρακτηριστικά παρουσίασης των HTML εγγράφων περιλαμβάνονταν μέσα στο έγγραφο σήμανσης HTML, όπως τα χρώματα του φόντου, τα στυλ του φόντου, οι ευθυγραμμίσεις των στοιχείων, ενώ τα σύνορα και τα μεγέθη έπρεπε να περιγράφονται ρητά, και συχνά κατ'επανάληψη. Με τη χρήση των CSS επιτρέπετε στους συγγραφείς να μεταφέρουν πολλές από αυτές τις πληροφορίες σε ένα χωριστό φύλλο στυλ με αποτέλεσμα η HTML σήμανση να γίνεται πολύ απλούστερη.

Πριν από τα CSS, οι συγγραφείς που ήθελαν να δώσουν τέτοια τυπογραφικά χαρακτηριστικά στο έγγραφο, για παράδειγμα, όλα τα headings h2 έπρεπε να χρησιμοποιήσουν τη γραμματοσειρά HTML και άλλα στοιχεία παρουσίασης, έπρεπε να το κάνουν για κάθε εμφάνιση του heading h2. Έτσι η πρόσθετη σήμανση παρουσίασης στα έγγραφα HTML, τα έκανε πιο πολύπλοκα, και γενικά πιο δύσκολο να διατηρηθούν. Στα CSS, η παρουσίαση χωρίζεται από τη δομή επιτρέποντας έτσι, στην εκτύπωση, το CSS να μπορεί να ορίσει το χρώμα, τη γραμματοσειρά, την στοίχιση του κειμένου, το μέγεθος, τα σύνορα, τις αποστάσεις, τη διάταξη και πολλά άλλα τυπογραφικά χαρακτηριστικά,. Ενώ μπορούν να το κάνουν αυτό ανεξάρτητα για την οθόνη και ανεξάρτητα για τα έντυπα. Το W3C θεωρεί ότι τα πλεονεκτήματα των CSS για τον καθορισμό όλων των πτυχών της παρουσίασης των σελίδων HTML είναι ανώτερη από άλλες μεθόδους παρουσίασης.

2.2.3 Πηγές πληροφοριών στα CSS

Οι πληροφορίες των CSS μπορούν να παρέχονται από διάφορες πηγές και μπορούν να συνδεθούν με το έγγραφο HTML είτε ως χωριστό έγγραφο είτε να ενσωματωθούν σε αυτό. Διαφορετικά στυλ μπορούν να εφαρμοστούν σε ένα έγγραφο HTML ανάλογα με τη συσκευή εξόδου που χρησιμοποιείται. Για παράδειγμα, η έκδοση για την οθόνη μπορεί να είναι αρκετά

διαφορετική από την έντυπη έκδοση, έτσι ώστε οι συντάκτες να μπορούν να προσαρμόσουν την παρουσίαση κατάλληλα για κάθε μέσο.

Οι προτεραιότητες για τις πηγές CSS (από την υψηλότερη στη χαμηλότερη) είναι οι ακόλουθες :

Author styles (είναι πληροφορίες που παρέχονται από το συγγραφέα της ιστοσελίδας), με τη μορφή

- Inline στυλ, δηλαδή πληροφορίες μέσα στο έγγραφο HTML, για το ύψος σε ένα μόνο στοιχείο, που καθορίζονται με το χαρακτηριστικό "style".
- Ενσωματωμένα στυλ, δηλαδή μπλοκ πληροφοριών για τα CSS στο εσωτερικό του έγγραφου HTML
- εξωτερικά φύλλα στυλ, δηλαδή ένα ξεχωριστό CSS-αρχείο που αναφέρεται στο έγγραφο HTML.

User Styles

- Αυτού του είδους τα στυλ είναι ένα τοπικό αρχείο CSS, το οποίο καθορίζεται από τον χρήστη χρησιμοποιώντας τις επιλογές στο web browser, και ενεργεί ως παράκαμψη, που πρέπει να εφαρμόζεται σε όλα τα έγγραφα.

User Agent styles

- Είναι το φύλλο στυλ που ισχύει από τον χρήστη, π.χ. η προεπιλεγμένη παρουσίαση των στοιχείων του browser του χρήστη.

Το φύλλο στυλ με την υψηλότερη προτεραιότητα χρησιμοποιείται ώστε να εμφανίσει το περιεχόμενο. Δηλώσεις που δεν έχουν δηλωθεί ως πηγές υψηλής προτεραιότητας, θα παρακάμπτονται από πηγές χαμηλότερης προτεραιότητας, όπως π.χ. το στυλ πράκτορα χρήστη. Αυτή η διαδικασία ονομάζεται cascading.

Ένας από τους στόχους των CSS είναι, να επιτρέπει στους χρήστες να έχουν μεγαλύτερο βαθμό ελέγχου στην παρουσίαση. Για παράδειγμα εκείνοι που βρίσκουν τους κόκκινους πλάγιους τίτλους δύσκολο να διαβαστούν, μπορούν να εφαρμόσουν άλλα φύλλα στυλ στο έγγραφο. Έτσι ανάλογα με τον browser και το web site, ο χρήστης μπορεί να επιλέξει διάφορα φύλλα στυλ που παρέχονται από τους σχεδιαστές, ώστε να αποσύρει όλα τα προστιθέμενα στυλ και να προβάλει την ιστοσελίδα χρησιμοποιώντας το προεπιλεγμένο στυλ του browser ή μπορεί να εξουδετερώσει μόνο τους κόκκινους πλάγιους τίτλους της κλάσης χωρίς να τροποποιήσει τα άλλα χαρακτηριστικά της σελίδας.

Το αρχείο `highlightheaders.css` περιέχει:

```
h1 { color: white; background: orange !important; }  
h2 { color: white; background: green !important; }
```

Ένα τέτοιο αρχείο είναι αποθηκευμένο σε τοπικό επίπεδο και μπορεί να εφαρμοστεί σε περίπτωση που έχει οριστεί στις επιλογές περιήγησης. Η λέξη "`!important`" στο τέλος σημαίνει ότι το στυλ αυτό υπερισχύει από τα ορίσματα του συγγραφέα.

2.3 JavaScript

Η JavaScript είναι μια γλώσσα σε μια μορφή script που χρησιμοποιείται για να επιτρέψει την πρόσβαση μέσω προγραμματισμού σε αντικείμενα τόσο στην εφαρμογή-πελάτη όσο και άλλες εφαρμογές. Χρησιμοποιείται κυρίως με τη μορφή `client-side JavaScript`, ενσωματωμένη ως ένα ολοκληρωμένο στοιχείο του `web browser`, επιτρέποντας έτσι την ανάπτυξη ενισχυμένων διεπαφών χρήστη και δυναμικών ιστοσελίδων. Η JavaScript είναι μια διάλεκτος του προτύπου `ECMAScript`¹⁰ και χαρακτηρίζεται ως δυναμική και ασθενώς δακτυλογραφημένη. Η JavaScript επηρεάστηκε από πολλές γλώσσες και έχει σχεδιαστεί να μοιάζει με τη `Java`, αλλά να είναι ευκολότερο για τους μη προγραμματιστές να εργαστούν με αυτή.

2.3.1 Χαρακτηριστικά της JavaScript

Τα ακόλουθα χαρακτηριστικά είναι κοινά σε όλες τις εφαρμογές σύμφωνα με το `ECMAScript`, εκτός και αν ορίζεται διαφορετικά.

2.3.1.1 Αναγκαστική και δομημένη

Η JavaScript υποστηρίζει όλη τη δομημένη σύνταξη προγραμματισμού σε `C` (π.χ., δηλώσεις `if`, `while` βρόχοι, δηλώσεις `switch`, κλπ.), με μία μερική εξαίρεση την οριοθέτηση των πεδίων: πχ. το `C-block style-scoping` επίπεδο δεν υποστηρίζεται (αντ' αυτού, η JavaScript έχει ένα `functional-scoping` επίπεδο). Η JavaScript 1.7, ωστόσο, υποστηρίζει το `block-επίπεδο scoping` χρησιμοποιώντας την λέξη `let`. Όπως και η `C`, έτσι και η JavaScript κάνει διάκριση μεταξύ των εκφράσεων (`expressions`) και των δηλώσεων (`statements`).

¹⁰ Πρότυπο γλωσσών scripting <http://en.wikipedia.org/wiki/ECMAScript>

2.3.1.2 Δυναμική

Δυναμική δακτυλογράφηση

Όπως στις περισσότερες γλώσσες προγραμματισμού, οι τύποι σχετίζονται με τις αξίες, και όχι με τις μεταβλητές αυτές καθ' αυτές. Για παράδειγμα, μια μεταβλητή `x` θα μπορούσε να δεσμεύεται σε έναν αριθμό, και αργότερα σε μια λέξη. Η JavaScript υποστηρίζει διάφορους τρόπους για τη δοκιμή του τύπου ενός αντικείμενου.

Βασισμένη στο αντικείμενο

Η JavaScript είναι σχεδόν εξ ολοκλήρου βασισμένη στο αντικείμενο. Τα αντικείμενα της JavaScript είναι συστοιχίες, επαυξημένες με τα πρωτότυπα. Τα ονόματα των ιδιοτήτων των αντικειμένων είναι λέξεις κλειδιά : Για παράδειγμα το `obj.x = 10` και `obj ["x"] = 10` είναι ισοδύναμα, με την τελεία στη σύνταξη να χρησιμοποιείται για να διαβάζεται ή να εκφράζεται ευκολότερα το αντικείμενο. Οι ιδιότητες και οι τιμές των αντικειμένων, μπορούν να προστεθούν, μεταβληθούν ή και να διαγραφούν κατά το χρόνο εκτέλεσης καθώς οι περισσότερες ιδιότητες ενός αντικείμενου (και εκείνων που αφορούν την πρωτότυπη αλυσίδα κληρονομιάς) μπορούν να καταμετρηθούν με τη χρήση ενός βρόχου `for`. Η JavaScript έχει ένα μικρό αριθμό ενσωματωμένων αντικειμένων όπως το `Function` και το `Date`.

Αξιολόγηση του χρόνου εκτέλεσης

Η JavaScript περιλαμβάνει τη λειτουργία `eval` που μπορεί να εκτελέσει τα `statements` που έχουν δοθεί σαν `strings` κατά το χρόνο εκτέλεσης.

2.3.1.3 Λειτουργίες

Πρώτης κλάσης λειτουργίες

Τα `functions` ανήκουν στην πρώτη κλάση και υπάγονται στη κατηγορία των αντικειμένων και ως εκ τούτου, έχουν ιδιότητες και μπορούν να χρησιμοποιηθούν όπως κάθε άλλο αντικείμενο.

Εσωτερικές λειτουργίες(functions) και closures

Τα εσωτερικά `functions` (`functions` που ορίζονται σε άλλα `functions`) δημιουργούνται κάθε φορά που γίνεται επίκληση στην εξωτερική `function`, και οι μεταβλητές των εξωτερικών `functions` εξακολουθούν να υφίστανται στο βαθμό που οι εσωτερικές `functions` εξακολουθούν να υπάρχουν, ακόμη και μετά την ολοκλήρωση της επίκλησης (π.χ. αν η εσωτερική `function`

επιστράφηκε, εξακολουθεί να έχει πρόσβαση στις μεταβλητές της εξωτερικής function). Αυτός είναι ο μηχανισμός πίσω από τα closures εντός της JavaScript.

2.3.1.4 Βασισμένη στα πρωτότυπα

Πρωτότυπα

Η JavaScript χρησιμοποιεί πρωτότυπα αντί των κλάσεων κληρονομικότητας. Είναι δυνατό να προσομοιωθούν πολλά χαρακτηριστικά που χρησιμοποιούν ως βάση τις κλάσεις χρησιμοποιώντας τα πρωτότυπα στη JavaScript.

Λειτουργίες(functions) ως κατασκευαστές αντικειμένων

Οι λειτουργίες διπλασιάζονται σαν κατασκευαστές αντικειμένων μαζί με το τυπικό ρόλο τους. Προκαθορίζοντας μια κλήση συνάρτησης με τη λέξη new δημιουργείτε ένα νέο αντικείμενο και οι κλήσεις του λειτουργούν με την τοπική λέξη-κλειδί this δεσμεύοντας έτσι αυτό το αντικείμενο για την εν λόγω επίκληση. Η ιδιότητα prototype του κατασκευαστή καθορίζει το αντικείμενο που χρησιμοποιείται για το εσωτερικό πρωτότυπο ενός νέου αντικειμένου. Οι ενσωματωμένοι κατασκευαστές που υπάρχουν στην JavaScript, όπως για παράδειγμα το Array, έχουν επίσης πρωτότυπα τα οποία μπορούν να τροποποιηθούν.

Λειτουργίες ως μέθοδοι

Σε αντίθεση με πολλές αντικειμενοστραφείς γλώσσες, δεν υπάρχει διάκριση μεταξύ του ορισμού της λειτουργίας και του ορισμού της μεθόδου. Αντίθετα, η διάκριση γίνεται κατά τη διάρκεια της λειτουργίας, έτσι μια function μπορεί να κληθεί σαν να ήταν μια μέθοδος. Όταν μια function καλείται σαν μέθοδος ενός αντικειμένου, η λειτουργία της τοπικής λέξης this είναι υποχρεωτική για την εν λόγω επίκληση στο αντικείμενο.

2.3.1.5 Διάφορα

Περιβάλλον run-time

Η JavaScript συνήθως βασίζεται στο χρόνο εκτέλεσης του περιβάλλοντος (π.χ. σε ένα web browser) για να παρέχει αντικείμενα και μεθόδους μέσω των οποίων τα οποία scripts μπορούν να αλληλεπιδράσουν με "τον έξω κόσμο". Στην πραγματικότητα, εναπόκειται από το

περιβάλλον, έτσι ώστε να παρέχετε η δυνατότητα να συμπεριλάβει / scripts εισαγωγής (π.χ. HTML `<script>` στοιχεία).

Λειτουργίες variadic

Ένας άοριστος αριθμός παραμέτρων μπορεί να περάσει σε μια λειτουργία. Η λειτουργία αυτή μπορεί να έχει πρόσβαση σε αυτά μέσω των επίσημων παραμέτρων και των αντικειμένων τοπικών επιχειρημάτων.

Πίνακες και αντικείμενα literals

Όπως πολλές γλώσσες προγραμματισμού, έτσι και εδώ ,οι πίνακες και τα αντικείμενα (associative arrays σε άλλες γλώσσες), μπορούν να δημιουργηθούν το καθένα με μια συνοπτική συντόμευση της σύνταξης. Στην πραγματικότητα, αυτές οι literals αποτελούν τη βάση για τη μορφή των δεδομένων JSON.

Σύνηθες εκφράσεις

Η JavaScript υποστηρίζει επίσης, σύνηθες εκφράσεις με παρόμοιο τρόπο με τη Perl, οι οποίες παρέχουν μια πιο συνοπτική και ισχυρή σύνταξη για τη χειραγώγηση του κειμένου που είναι πιο εξελιγμένη από ό, τι οι ενσωματωμένες συναρτήσεις συμβολοσειράς.

2.3.2 Χρήση της JavaScript στις ιστοσελίδες

Η κύρια χρήση της JavaScript είναι να γράφει λειτουργίες που είναι ενσωματωμένες ή περιλαμβάνονται στις HTML σελίδες και αλληλεπιδρούν με το Document Object Model (DOM) της σελίδας. Μερικά απλά παραδείγματα αυτής της χρήσης είναι:

- Το άνοιγμα ενός νέου παραθύρου με προγραμματιστικό έλεγχο του μεγέθους, της θέσης και των χαρακτηριστικών του νέου παραθύρου (δηλαδή εάν το μενού, η γραμμών εργαλείων, κλπ. θα είναι ορατά).
- Επικύρωση των τιμών μιας διαδικτυακής φόρμας για τη βεβαίωση ότι οι τιμές αυτές θα γίνουν δεκτές προτού αυτές υποβληθούν στο διακομιστή.
- Αλλαγή εικόνων καθώς ο κέρσορας του ποντικιού κινείται από πάνω τους: Η τεχνική αυτή χρησιμοποιείται συχνά για να επιστήσει την προσοχή του χρήστη σε σημαντικές συνδέσεις οι οποίες εμφανίζονται ως γραφικά στοιχεία.

Επειδή ο κώδικας της JavaScript μπορεί να λειτουργεί τοπικά σε ένα πρόγραμμα περιήγησης του χρήστη (και όχι σε ένα απομακρυσμένο server) μπορεί να ανταποκρίνεται άμεσα στις ενέργειες χρηστών, κάνοντας τις εφαρμογές πιο διαδραστικές. Επιπλέον, ο κώδικας της JavaScript μπορεί να ανιχνεύσει ενέργειες χρηστών που η HTML δεν μπορεί να ανιχνεύσει από μόνη της, όπως παραδείγματος χάριν ατομικές πληκτρολογήσεις. Εφαρμογές, όπως το Gmail επωφελούνται από αυτό αφού ένα μεγάλο μέρος της λογικής της διεπαφής του χρήστη είναι γραμμένο σε

JavaScript, και η JavaScript αποστέλλει αιτήσεις για παροχή πληροφοριών (όπως το περιεχόμενο ενός μηνύματος ηλεκτρονικού ταχυδρομείου) στο διακομιστή. Η ευρύτερη τάση του προγραμματισμού Ajax εκμεταλλεύεται με τον ίδιο τρόπο αυτή την δύναμη.

Ένας μηχανισμός της JavaScript (επίσης γνωστός και ως διερμηνέας της JavaScript ή JavaScript implementation) είναι ένας διερμηνέας που μεταφράζει τον JavaScript κώδικα και τον εκτελεί αναλόγως. Η πρώτη μηχανή JavaScript δημιουργήθηκε από Brendan Eich στο Netscape Communications Corporation, για την εφαρμογή περιήγησης Netscape Navigator web.

Ένα πρόγραμμα περιήγησης στο διαδίκτυο είναι μακράν το πιο κοινό περιβάλλον υποδοχής για τη JavaScript. Τα προγράμματα περιήγησης στο διαδίκτυο χρησιμοποιούν συνήθως API για τη δημιουργία "αντικειμένων υποδοχής" τα οποία είναι υπεύθυνα για την αντανάκλαση του DOM σε JavaScript. Ο web server είναι μια άλλη κοινή εφαρμογή του μηχανισμού της JavaScript. Ένας JavaScript web Server εκθέτει τα αντικείμενα εκείνα που αντιπροσωπεύουν μια αίτηση HTTP καθώς και αντικείμενα απάντησης, τα οποία ένα πρόγραμμα JavaScript στη συνέχεια χειραγωγεί, δημιουργώντας μια δυναμική ιστοσελίδα.

Ένα παράδειγμα μιας ιστοσελίδας που περιέχει JavaScript είναι το παρακάτω:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head><title>simple page</title></head>
  <body>
    <script type="text/javascript">
      document.write('Hello World!');
    </script>
    <noscript>
      <p>Your browser either does not support JavaScript, or you
have JavaScript turned off.</p>
    </noscript>
  </body>
</html>
```

2.4 XML

Η XML (eXtensible Markup Language) είναι ένα σύνολο κανόνων για την κωδικοποίηση των εγγράφων με ηλεκτρονικά μέσα. Ορίζεται στην προδιαγραφή XML 1.0 που παράγεται από το W3C και πολλές άλλες σχετικές προδιαγραφές.

Οι σχεδιαστικοί στόχοι της XML είναι να τονίσει την απλότητα, τη γενικότητα, και την ευχρηστία της μέσω του διαδικτύου. Πρόκειται για μια μορφή δεδομένων κειμένου, με ισχυρή υποστήριξη μέσω του Unicode για τις διάφορες γλώσσες ανά τον κόσμο. Αν και ο σχεδιασμός της XML επικεντρώνεται σε έγγραφα, χρησιμοποιείται ευρέως για την αναπαράσταση των αυθαίρετων δομών δεδομένων, όπως για παράδειγμα στον τομέα των υπηρεσιών web.

Υπάρχει μια ποικιλία προγραμμάτων που χρησιμοποιούν οι προγραμματιστές λογισμικού ώστε να μπορούν να χρησιμοποιήσουν την XML για πρόσβαση στα δεδομένα, και πολλά συστήματα σχημάτων με στόχο να βοηθήσουν στον ορισμό των γλωσσών που βασίζονται στη XML.

Από το 2009 και μετά, έχουν αναπτυχθεί πολλές γλώσσες που βασίζονται σε XML, συμπεριλαμβανομένου σε αυτές την RSS, την Atom, την SOAP, και την XHTML. Η XML έχει γίνει η προεπιλεγμένη μορφή αρχείου για τα περισσότερα εργαλεία γραφείου παραγωγικότητας, συμπεριλαμβανομένου του Microsoft Office, OpenOffice.org, AbiWord, και iWork της Apple.

2.4.1 Ανάλυση των χαρακτηριστικών της XML

Τα παρακάτω χαρακτηριστικά βασίζονται στις προδιαγραφές της XML, χωρίς όμως να αποτελούν έναν πλήρη κατάλογο όλων των χαρακτηριστικών που εμφανίζονται στην XML αλλά παρέχετε μια εισαγωγή στα βασικά χαρακτηριστικά που συναντάμε συχνότερα.

Χαρακτήρες (Unicode)

Εξ ορισμού, ένα έγγραφο XML είναι μια συμβολοσειρά χαρακτήρων, στον οποίο σχεδόν κάθε νόμιμος χαρακτήρας Unicode, μπορεί να εμφανιστεί .

Επεξεργαστής και εφαρμογές

Αναμένεται ότι ένας επεξεργαστής λειτουργεί στην υπηρεσία της εφαρμογής που χειρίζεται XML. Υπάρχουν ορισμένες πολύ ειδικές απαιτήσεις σχετικά με το τι πρέπει να κάνει και το τι δεν πρέπει να κάνει ο επεξεργαστής XML, αλλά καμία ως το προς τη συμπεριφορά της

εφαρμογής. Ο επεξεργαστής (όπως οι προδιαγραφές τον ονομάζουν) αναφέρεται συχνά στην καθομιλουμένη ως ένας XML parser.

Markup και περιεχόμενο

Οι χαρακτήρες που συνθέτουν ένα έγγραφο XML χωρίζονται στη σήμανση και στο περιεχόμενο. Η σήμανση και το περιεχόμενο μπορούν να διακρίνονται από την εφαρμογή μέσω απλών συντακτικών κανόνων. Όλες οι συμβολοσειρές που αποτελούν σήμανση αρχίζουν με το χαρακτήρα "<" και τελειώνουν με το χαρακτήρα ">", ή αρχίζουν με το χαρακτήρα "&" και τελειώνουν με τον χαρακτήρα ";". Οι συμβολοσειρές χαρακτήρων που δεν αποτελούν μέρος του markup αποτελούν μέρος του περιεχόμενου.

Tag (Ετικέτες)

Είναι μια markup κατασκευή που ξεκινά με "<" και τελειώνει με ">". Οι ετικέτες εμφανίζονται σε τρεις μορφές: Ετικέτες έναρξης, για παράδειγμα <section>, Ετικέτες τέλους, για παράδειγμα </section>, και σε ετικέτες κενών στοιχείων, για παράδειγμα <line-break/>.

Στοιχεία

Είναι μια λογική συνιστώσα του εγγράφου η οποία αρχίζει είτε με την ίδρυση μιας ετικέτας και τελειώνει με μια ετικέτα τέλους, ή αποτελείται μόνο από μια ετικέτα κενών στοιχείων. Οι χαρακτήρες μεταξύ των σημείων των ετικετών έναρξης και λήξης, εάν υπάρχουν, είναι τα περιεχόμενα του στοιχείου, και μπορεί να περιέχουν σήμανση, συμπεριλαμβανομένων και των άλλων στοιχείων, τα οποία ονομάζονται child elements. Ένα παράδειγμα είναι ένα στοιχείο <Greeting> Hello, world. </Greeting>. Ένα άλλο είναι το <line-break/>.

Ιδιότητα

Μια markup κατασκευή αποτελείται από ένα ζεύγος όνομα / τιμή που υπάρχει μέσα σε μία ετικέτα έναρξης ή σε μια ετικέτα κενού στοιχείου. Σε αυτό το παράδειγμα, το όνομα της ιδιότητας είναι "number" και η τιμή της είναι "3": <step number="3"> Connect A to B. </ step> Το ακόλουθο στοιχείο έχει δύο ιδιότητες, src και alt: . Ένα στοιχείο δεν πρέπει να έχει δύο ιδιότητες με το ίδιο όνομα.

Δηλώσεις XML

Τα XML έγγραφα ξεκινούνε δηλώνοντας ορισμένες πληροφορίες για τον εαυτό τους, όπως στο ακόλουθο παράδειγμα.

```
<? xml version = "1.0" encoding = "utf-8"?>
```

Παράδειγμα

Εδώ είναι ένα μικρό, πλήρες έγγραφο XML, το οποίο χρησιμοποιεί όλες τις δομές και τις έννοιες.

```
<? xml version = "1.0" encoding = "utf-8"?>
<painting>
  
  <caption> This is Raphael's "Folighno" Madonna, painted in
  <date> 1511 </ date>-<date> 1512 </ date>. </caption>
</painting>
```

Υπάρχουν πέντε στοιχεία σε αυτό το έγγραφο: `painting`, `img`, `title`, και δύο `date`. Τα στοιχεία `date` είναι παιδιά του `caption`, το οποίο είναι ένα παιδί του στοιχείου `painting`. Το `img` έχει δύο ιδιότητες, `src` και `alt`.

2.4.2 Χαρακτήρες και διαφυγή

Τα XML έγγραφα που αποτελούνται εξ ολοκλήρου από τους χαρακτήρες του Unicode. Εκτός από ένα μικρό αριθμό, ρητά αποκλεισμένων, χαρακτήρων ελέγχου κάθε χαρακτήρας που ορίζεται από το Unicode μπορεί να εμφανιστεί στο περιεχόμενο ενός εγγράφου XML. Η επιλογή των χαρακτήρων που μπορούν να εμφανιστούν εντός σήμανσης είναι κάπως πιο περιορισμένη αλλά εξακολουθεί να είναι μεγάλη.

Η XML περιλαμβάνει ιδιότητες για τον προσδιορισμό της κωδικοποίησης των χαρακτήρων Unicode που συνθέτουν το έγγραφο, και για την έκφραση των χαρακτήρων που, για τον έναν ή τον άλλο λόγο, δεν μπορούν να χρησιμοποιηθούν άμεσα.

Κωδικοποίηση ανίχνευσης

Το σύνολο χαρακτήρων Unicode μπορούν να κωδικοποιηθούν σε bytes για την αποθήκευση ή τη μετάδοση με μια ποικιλία διαφορετικών τρόπων, που ονομάζεται "κωδικοποίηση". Το Unicode από μόνο του ορίζει κωδικοποιήσεις που καλύπτουν όλο το σύνολο, ποιο γνωστές από τις οποίες είναι το UTF-8 και το UTF-16. Υπάρχουν πολλές άλλες κωδικοποιήσεις κείμενου που υπήρχαν πριν από το Unicode, όπως ASCII και ISO / IEC 8859. Τα χαρακτηριστικά τους σχεδόν σε κάθε περίπτωση είναι υποσύνολα των χαρακτήρων Unicode.

Η XML επιτρέπει τη χρήση οποιασδήποτε από τις καθορισμένες Unicode κωδικοποιήσεις, καθώς και κάθε άλλη μορφή κωδικοποίησης χαρακτήρων των οποίων οι χαρακτήρες εμφανίζονται στο Unicode. Η XML επίσης παρέχει ένα μηχανισμό με τον οποίο ένας XML επεξεργαστής μπορεί με αξιοπιστία και χωρίς καμία προηγούμενη γνώση, να καθορίσει τη κωδικοποίηση που χρησιμοποιείται, χωρίς να είναι απαραίτητο ότι κωδικοποιήσεις πέραν της UTF-8 και της UTF-16 θα αναγνωριστούν από κάθε parser.

Χαρακτήρες Διαφυγής

Υπάρχουν διάφοροι λόγοι για τους οποίους μπορεί να είναι δύσκολο ή αδύνατο να συμπεριληφθούν ορισμένοι χαρακτήρες απευθείας σε ένα έγγραφο XML. Για παράδειγμα, οι χαρακτήρες "<" και "&" αποτελούν βασικούς δείκτες σύνταξης και δεν μπορούν να εμφανιστούν στο περιεχόμενο, (π.χ. μπορεί να κωδικοποιηθεί ένα έγγραφο XML σε ASCII, αλλά δεν η ASCII δεν μπορεί να υποστηρίξει πολλούς χαρακτήρες Unicode, όπως για παράδειγμα, το "έ"). Για τους λόγους αυτούς, η XML παρέχει χαρακτηριστικά διαφυγής για τις περιπτώσεις που υπάρχουν προβληματικοί ή ανύπαρκτοι χαρακτήρες. Υπάρχουν πέντε προκαθορισμένες οντότητες: το `<`; το οποίο αναπαριστά τον χαρακτήρα "<", το `>`; το οποίο αναπαριστά τον χαρακτήρα ">", το `&`; το οποίο αναπαριστά τον χαρακτήρα "&", το `'`; το οποίο αναπαριστά τον χαρακτήρα "'", και το `"`; το οποίο αναπαριστά τον χαρακτήρα ". Επίσης, κάθε χαρακτήρας Unicode μπορεί να εκπροσωπείται με έναν αριθμητικό χαρακτήρα αναφοράς. Σκεφτείτε το κινεζικό χαρακτήρα "中", αριθμητικός κωδικός του οποίου σε μορφή Unicode είναι ο 4E2D σε δεκαεξαδική μορφή ή σε δεκαδική ο 20.013. Ένας χρήστης του οποίου το πληκτρολόγιο δεν προσφέρει καμία μέθοδο για την εισαγωγή αυτού χαρακτήρα, θα μπορούσε να τον εισάγει παράλα αυτά σε ένα έγγραφο XML με την κωδικοποίηση του είτε ως `中` ή `中`. Ομοίως, η συμβολοσειρά "I <3 Jörg" θα μπορούσε να είναι κωδικοποιημένη για συμπεριλαμβάνεται σε ένα έγγραφο XML ως `I <3 Jörg`".

2.4.3 XMLHttpRequest

Το XMLHttpRequest (XHR) είναι ένα DOM API που μπορεί να χρησιμοποιηθεί μέσα σε μια web browser scripting language, όπως JavaScript, για να στείλει μία αίτηση HTTP ή HTTPS

απευθείας σε έναν web server και να φορτώσει τα δεδομένα της απάντησης από τον server απευθείας πίσω σε scripting language. Από την στιγμή που τα δεδομένα θα βρίσκονται εντός του scripting language, είναι διαθέσιμα και ως ένα έγγραφο XML, αν η απάντηση ήταν έγκυρης σήμανσης της XML, αλλά και ως απλό κείμενο. Έτσι τα δεδομένα της XML μπορούν να χρησιμοποιηθούν ώστε να χειραγωγήσουν το ενεργό έγγραφο στο παράθυρο του προγράμματος περιήγησης χωρίς να είναι απαραίτητο να φορτωθεί εκ νέου το έγγραφο της ιστοσελίδας από τον χρήστη. Το απλό κείμενο δεδομένων επίσης μπορεί να αξιολογηθεί στο πλαίσιο του scripting language ώστε να χειραγωγηθεί το έγγραφο. Στο παράδειγμα της JavaScript, το απλό κείμενο μπορεί να έχει διαμορφωθεί ως JSON από τον web server και να αξιολογηθεί εντός της JavaScript για να δημιουργήσει ένα αντικείμενο δεδομένων που θα χρησιμοποιηθεί στο τρέχων DOM έγγραφο.

Το αντικείμενο XMLHttpRequest έχει ένα αρκετά σημαντικό ρόλο στην τεχνική AJAX. Χρησιμοποιείται σήμερα από πολλές ιστοσελίδες για να εφαρμόζουν και να ανταποκρίνονται σε δυναμικές web εφαρμογές. Παραδείγματα αυτών των εφαρμογών web περιλαμβάνουν το Gmail, το Google Maps, οι Bing Χάρτες, η δυναμική διεπαφή χάρτη MapQuest, το Facebook, και άλλα.

2.4.3.1 Η αίτηση HTTP

Παρακάτω θα δούμε πώς γίνεται μια αίτηση χρησιμοποιώντας τις λειτουργίες του αντικείμενου XMLHttpRequest σε ένα πράκτορα χρήστη με βάση το σχέδιο εργασίας του W3C. Δεδομένου ότι το πρότυπο του W3C για το αντικείμενο XMLHttpRequest εξακολουθεί να είναι ένα σχέδιο, οι πράκτορες του χρήστη ίσως να μην μπορούν να τηρήσουν όλες τις λειτουργίες του ορισμού του W3C και οποιοδήποτε από τα παρακάτω υπόκεινται σε αλλαγές. Εξαιρετική προσοχή θα πρέπει να λαμβάνεται υπόψη κατά το scripting με το αντικείμενο XMLHttpRequest σε πολλούς πράκτορες χρήστη. Παρακάτω απαριθμούνται οι ασυμβατότητες μεταξύ των κυριότερων πρακτόρων χρήστη.

Η μέθοδος open

Τα HTTP και HTTPS αιτήματα του αντικείμενου XMLHttpRequest πρέπει να αρχικοποιηθούν μέσω της μεθόδου open. Η μέθοδος αυτή πρέπει να κληθεί πριν από την πραγματική αποστολή της αίτησης για να επικυρωθεί και να επιλυθεί η μέθοδος αίτησης, ενώ το URL και URI πληροφορίες του χρήστη θα χρησιμοποιηθούν για την αίτηση. Η μέθοδος αυτή δεν διαβεβαιώνει ότι η διεύθυνση URL υπάρχει ή ότι τα στοιχεία του χρήστη είναι σωστά. Αυτή η μέθοδος μπορεί να δεχτεί μέχρι και πέντε παραμέτρους, ωστόσο δύο μόνο, είναι αρκετοί για να αρχικοποιηθεί μια αίτηση.

Η πρώτη παράμετρος της μεθόδου είναι μια συμβολοσειρά κειμένου που αναφέρει τη μέθοδο-αίτησης HTTP που θα χρησιμοποιηθεί. Οι μέθοδοι αίτησης που πρέπει να υποστηρίζονται από

έναν πράκτορα χρήστη, και έχουν οριστεί από το σχέδιο του W3C για το αντικείμενο XMLHttpRequest, περιλαμβάνονται προς το παρόν τα εξής:

- GET (Υποστηρίζεται από IE7 +, Mozilla 1 +)
- POST (Υποστηρίζεται από IE7 +, Mozilla 1 +)
- HEAD (Υποστηριζόμενο από IE7 +)
- PUT
- DELETE
- OPTIONS (Υποστηριζόμενο από IE7 +)

Ωστόσο, οι μέθοδοι αίτησης δεν περιορίζονται μόνο σε εκείνες που αναγράφονται παραπάνω. Το σχέδιο του W3C δηλώνει ότι ένα πρόγραμμα περιήγησης μπορεί να υποστηρίζει επιπρόσθετες μεθόδους αίτησης κατά την κρίση τους.

Η δεύτερη παράμετρος της μεθόδου είναι άλλη μια συμβολοσειρά κειμένου, που αναφέρει το URL της αίτησης HTTP. Το W3C συνιστά ότι οι περιηγητές θα πρέπει να αναφέρουν ένα λάθος και να μην επιτρέπουν την αίτηση του URL σε μια διαφορετική θύρα ή με ένα διαφορετικό ihost URI από ότι το τρέχον έγγραφο.

Στη τρίτη παράμετρο, υπάρχει μια boolean τιμή οι οποία συμβολίζει εάν η αίτηση θα είναι ή δεν θα είναι ασύγχρονη, βέβαια αυτή δεν είναι μια απαραίτητη παράμετρος που αναφέρεται στο σχέδιο του W3C. Η προεπιλεγμένη τιμή αυτής της παραμέτρου θα πρέπει να θεωρείται ότι είναι αληθής σύμφωνα με το W3C από κάθε πράκτορα χρήστη σε περίπτωση που αυτή δεν παρέχεται. Μια ασύγχρονη αίτηση ("true") δεν θα περιμένει για μια απάντηση από τον server πριν συνεχίσει κανονικά με την εκτέλεση του τρέχοντος script. Θα επικαλεστεί, αντ' αυτού το onreadystatechange event listener του αντικειμένου XMLHttpRequest κατά τα διάφορα στάδια της αίτησης. Αντίθετα μια σύγχρονη αίτηση ("false"), θα παγώσει την εκτέλεση του τρέχοντος script έως ότου η αίτηση ολοκληρωθεί, χωρίς να επικαλεστεί το onreadystatechange event listener.

Η τέταρτη και πέμπτη παράμετροι είναι τα URI, όνομα χρήστη και κωδικό πρόσβασης, αντίστοιχα. Αυτές οι παράμετροι, ή απλά το όνομα χρήστη, παρέχονται για τον έλεγχο της ταυτότητας του χρήστη, εφόσον αυτή απαιτείται από το διακομιστή για το συγκεκριμένο αίτημα.

Η μέθοδος setRequestHeader

Με την επιτυχή αρχικοποίηση μιας αίτησης, η μέθοδος setRequestHeader του αντικειμένου XMLHttpRequest μπορεί να χρησιμοποιηθεί για να αποστείλει κεφαλίδες HTTP μαζί με την αίτηση. Η πρώτη παράμετρος αυτής της μεθόδου είναι η συμβολοσειρά του ονόματος του

κειμένου της κεφαλίδας και η δεύτερη παράμετρος είναι η συμβολοσειρά της τιμής του κειμένου. Στη μέθοδο αυτή πρέπει να γίνει επίκληση για κάθε κεφαλίδα που πρέπει να αποσταλεί με την αίτηση αλλιώς κάθε επιπλέον κεφαλίδα που είναι συνημμένη θα αφαιρείται την επόμενη φορά που η μέθοδος `open` θα επικαλεστεί σε ένα πράκτορα χρήστη.

Η μέθοδος αποστολής (send)

Για να σταλεί μια αίτηση HTTP, πρέπει να επικαλεστεί η μέθοδος αποστολής του `XMLHttpRequest`. Αυτή η μέθοδος δέχεται μία μόνο παράμετρο, που περιέχει το περιεχόμενο που θα αποσταλεί με την αίτηση, η οποία παράμετρος μπορεί να παραληφθεί εάν δεν υπάρχει κάποιο περιεχόμενο για να αποσταλεί. Το σχέδιο του W3C αναφέρει ότι η παράμετρος αυτή μπορεί να είναι οποιοδήποτε είδους που είναι στη διάθεση της `scripting language`, εφόσον αυτό μπορεί να μετατραπεί σε μια συμβολοσειρά κειμένου, με εξαίρεση το αντικείμενο εγγράφου DOM. Εάν ένας πράκτορας χρήστη δεν μπορεί να κάνει συμβολοσειρά την παράμετρο, τότε η παράμετρος αυτή θα αγνοείται.

Εάν η παράμετρος αποτελεί ένα αντικείμενο εγγράφου DOM, τότε ο πράκτορας του χρήστη θα πρέπει να διαβεβαιώσει ότι το έγγραφο έχει μετατραπεί σε καλά μορφοποιημένο XML το οποίο χρησιμοποιεί την κωδικοποίηση που υποδεικνύεται από το `inputEncoding` του αντικειμένου του εγγράφου. Εάν η κεφαλίδα αίτησης `Content-Type` δεν έχει προστεθεί μέσω του `setRequestHeader`, θα πρέπει αυτόματα να προστίθεται από τον πράκτορα του χρήστη ως `"application/xml; charset=charset"` όπου το `charset` είναι η κωδικοποίηση που χρησιμοποιείται για την κωδικοποίηση του εγγράφου.

Ο onreadystatechange event listener

Εάν γίνει επίκληση της μεθόδου `open` του αντικειμένου `XMLHttpRequest`, με την τρίτη παράμετρο να ισχύει για ασύγχρονο αίτημα, θα γίνει αυτόματα επίκληση του `onreadystatechange event listener` για κάθε μία από τις ακόλουθες ενέργειες που αλλάζουν την ιδιότητα `readyState` του αντικειμένου `XMLHttpRequest`.

- Αφού έχει επικληθεί με επιτυχία η μέθοδος `open`, στην ιδιότητα του `readyState` του αντικειμένου `XMLHttpRequest` θα πρέπει να έχει αποδοθεί η τιμή 1.
- Αφού έχει επικληθεί με επιτυχία η μέθοδος `send` και η απάντηση των κεφαλίδων HTTP έχει ληφθεί, στην ιδιότητα `readyState` του αντικειμένου `XMLHttpRequest` θα πρέπει να έχει αποδοθεί η τιμή 2.
- Μόλις το περιεχόμενο της απάντησης HTTP αρχίζει να φορτώνεται, στην ιδιότητα `readyState` του αντικειμένου `XMLHttpRequest` θα πρέπει να έχει αποδοθεί η τιμή 3.

- Μόλις το φόρτωμα του περιεχομένου της απάντησης HTTP έχει ολοκληρωθεί, στην ιδιότητα `readyState` του αντικειμένου `XMLHttpRequest` θα πρέπει να έχει αποδοθεί η τιμή 4.

Η απάντηση HTTP

Μετά από μια επιτυχημένη και ολοκληρωμένη κλήση της μεθόδου `send` του `XMLHttpRequest` αντικειμένου, εάν η απάντηση από τον server ήταν ένα έγκυρο XML και η κεφαλίδα του `content-type` που αποστάληκε από το διακομιστή είναι κατανοητή από τον πράκτορα του χρήστη ως `Internet media type` για XML, η ιδιότητα `responseXML` του αντικειμένου `XMLHttpRequest` θα περιέχει ένα αντικείμενο εγγράφου DOM. Το `responseText` θα περιέχει την απάντηση του server σε μορφή απλού κειμένου από ένα σύμφωνο πράκτορα χρήστη, ανεξάρτητα από το αν ήταν ή δεν ήταν κατανοητό ως XML.

2.5 Java Servlets

Τα Servlets είναι αντικείμενα της γλώσσας προγραμματισμού Java που επεξεργάζονται δυναμικά την αίτηση και την κατασκευή απάντησης. Η Java Servlet API επιτρέπει την ανάπτυξη λογισμικού ώστε να είναι δυνατόν να προσθεθεί περιεχόμενο σε ένα διακομιστή Web που χρησιμοποιεί την πλατφόρμα Java, δυναμικά. Το περιεχόμενο που παράγεται συνήθως είναι HTML, αλλά μπορεί να είναι άλλων μορφών, όπως XML. Τα Servlets είναι τα υποκατάστατα για τις μη δυναμικές τεχνολογίες περιεχομένου Java στον Παγκόσμιο Ιστό, όπως η CGI και ASP.NET. Τα Servlets μπορούν να διατηρήσουν την κατάσταση τους σε πολλές συναλλαγές διακομιστή χρησιμοποιώντας τα HTTP cookies, τις μεταβλητές περιόδου ή την επανεγγραφή του URL.

Το Servlet API, το οποίο περιέχεται στο πακέτο `javax.servlet` της Java, καθορίζει όλες τις αναμενόμενες αλληλεπιδράσεις του web container και του servlet. Ένας web container είναι ουσιαστικά το συστατικό του διακομιστή Web που αλληλεπιδρά με τα servlets και είναι υπεύθυνος για τη διαχείριση του κύκλου ζωής των servlets, τη χαρτογράφηση μιας διεύθυνσης URL σε ένα συγκεκριμένο servlet και την διασφάλιση ότι ο αιτών του URL έχει τα σωστά δικαιώματα πρόσβασης.

Ένα Servlet είναι ένα αντικείμενο που δέχεται μια αίτηση και δημιουργεί μια απάντηση που βασίζεται στην εν λόγω αίτηση. Το βασικό πακέτο του servlet ορίζει τα αντικείμενα της Java τα οποία εκπροσωπούν τις αιτήσεις των servlet αλλά και τις απαντήσεις, καθώς και αντικείμενα τα οποία αντανakλούν τις παραμέτρους διαμόρφωσης του servlet και το περιβάλλον εκτέλεσης τους. Το πακέτο `javax.servlet.http` ορίζει τις ειδικές HTTP υποκατηγορίες των γενικών στοιχείων των servlet, συμπεριλαμβανομένων και εκείνων των αντικειμένων διαχείρισης σύνδεσης που

παρακολουθούν τις πολλαπλές αιτήσεις και απαντήσεις μεταξύ του Web server και ενός πελάτη. Τα Servlets μπορούν να συμπεριλαμβάνονται σε ένα αρχείο WAR ως μια εφαρμογή web.

Τα Servlets μπορούν να παράγονται αυτόματα από τα JavaServer Pages (JSP) compiler, ή εναλλακτικά από πρότυπους μηχανισμούς όπως WebMacro ή το apache velocity ώστε να παράγουν HTML. Συχνά τα Servlets χρησιμοποιούνται σε συνδυασμό με τα JSPs σε ένα μοτίβο που ονομάζεται "Model 2", το οποίο είναι ένα υποσύνολο του μοτίβου model-view-controller.

2.5.1 Κύκλος ζωής ενός Java Servlet

Η διάρκεια του κύκλου ζωής ενός servlet αποτελείται από τα ακόλουθα βήματα:

1. Η κλάση servlet φορτώνεται από τον container κατά τη διάρκεια της εκκίνησης.
2. Ο container καλεί τη μέθοδο `init()` η οποία αρχικοποιεί το servlet και πρέπει να καλείται προτού το servlet μπορεί να είναι σε θέση να εξυπηρετήσει τις όποιες αιτήσεις, και καλείται μόνο μία φορά κατά τη διάρκεια του κύκλου ζωής του.
3. Μετά την αρχικοποίηση, το servlet μπορεί να εξυπηρετήσει τις αιτήσεις των πελατών. Κάθε αίτηση εξυπηρετείται με το δικό της ξεχωριστό αίτημα. Ο container καλεί την μέθοδο `service()` του servlet για κάθε αίτηση η οποία με την σειρά της προσδιορίζει το είδος του αιτήματος και το προωθεί στην κατάλληλη μέθοδο ώστε να το χειριστεί. Ο προγραμματιστής του servlet πρέπει να παρέχει την εφαρμογή αυτών των μεθόδων. Εάν πραγματοποιηθεί αίτημα για μια μέθοδο που δεν έχει εφαρμοστεί από το servlet, η μέθοδος της μητρικής κλάσης καλείται, οδηγώντας συνήθως σε σφάλμα που επιστρέφεται στον αιτούντα.
4. Τέλος, ο container καλεί τη μέθοδο `destroy()` που θέτει το servlet εκτός υπηρεσίας. Η μέθοδος `destroy()`, όπως και η μέθοδος `init()`, καλείται μόνο μία φορά κατά τη διάρκεια του κύκλου ζωής του servlet.

Παρακάτω παραθέτετε ένα απλό servlet που παράγει μόνο HTML. Το `HttpServlet` είναι μια υποκατηγορία των `GenericServlet`. Η μέθοδος `service()` αποστέλλει αιτήσεις στις μεθόδους `doGet()`, `doPost()`, `doPut()`, `doDelete()`, κ.λπ., ανάλογα με το αίτημα HTTP.

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 \" +
            \"Transitional//EN\">\n" +
                "<html>\n" +
                "<head><title>Hello WWW</title></head>\n" +
                "<body>\n" +
                "<h1>Hello WWW</h1>\n" +
                "</body></html>");
    }
}

```

ServletConfig and ServletContext

Σε κάθε εφαρμογή υπάρχει μόνο ένα ServletContext. Αυτό το αντικείμενο μπορεί να χρησιμοποιηθεί από όλα τα servlets για να πάρει πληροφορίες από το επίπεδο της εφαρμογής ή τις λεπτομέρειες του container. Κάθε servlet, από την άλλη πλευρά, παίρνει το δικό του αντικείμενο ServletConfig το οποίο αναλαμβάνει να αρχικοποιήσει τις διάφορες παραμέτρους για ένα servlet. Οι προγραμματιστές μπορούν να λάβουν αναφορές για το ServletContext χρησιμοποιώντας το αντικείμενο ServletConfig.

2.5.2 Servlet Containers

Ένας servlet container είναι ένας εξειδικευμένος web server που υποστηρίζει την εκτέλεση του servlet. Συνδυάζει τις βασικές λειτουργίες ενός web server με συγκεκριμένες Java / servlet βελτιστοποιήσεις και επεκτάσεις, όπως ένα ολοκληρωμένο περιβάλλον χρόνου εκτέλεσης Java,

καθώς και την ικανότητα να μεταφράζει αυτόματα συγκεκριμένες διευθύνσεις URL σε αιτήματα servlet. Μεμονωμένα servlets καταχωρούνται με servlet containers, παρέχοντας στο container τις πληροφορίες για το τι λειτουργίες θα παρέχουν, και ποια διεύθυνση URL ή ποιο άλλο εντοπιστή πόρων θα χρησιμοποιήσουν για να αναγνωρίσουν τη ταυτότητα τους. Έπιτα ο container του servlet είναι σε θέση να αρχικοποιήσει το servlet ανάλογα με τις ανάγκες και να παραδώσει τις αιτήσεις στο servlet, καθώς αυτές καταφθάνουν. Πολλοί containers έχουν την δυνατότητα να προσθέτουν και να αφαιρούν servlets δυναμικά από το σύστημα, επιτρέποντας έτσι στα νέα servlets να εγκαθίστανται ή να αναιρούνται γρήγορα χωρίς να επηρεάζουν τα άλλα servlets που δουλεύουν από τον ίδιο container. Τα Servlet containers αναφέρονται επίσης ως web containers ή web engines.

Όπως και τα άλλα Java API, διαφορετικά vendors παρέχουν τις δικές τους υλοποιήσεις για το πρότυπο του servlet container. Μια λίστα με μερικά από τα ελεύθερα και web containers παρέχεται παρακάτω (Με τον όρο «ελεύθερα» σημαίνει ότι η μη εμπορική χρήση είναι δωρεάν, ενώ ορισμένα από τα εμπορικά containers, όπως το Resin και το Orion, είναι ελεύθερά στο να χρησιμοποιούνται σε ένα περιβάλλον server από μη κερδοσκοπικούς οργανισμούς):

- Apache Tomcat (πρώην Jakarta Tomcat) είναι ένας web container ανοικτής πηγής διαθέσιμος στα πλαίσια του Apache Software License.
- Apache Geronimo είναι μια πλήρης εφαρμογή Java EE από την Apache.
- GlassFish (ανοικτής πηγής κώδικας), από την Sun Microsystems.
- Jetty
- Jaminid περιέχει μεγαλύτερη άντληση από servlets.
- Enhydra
- Winstone στηρίζει προδιαγραφή v2.4, επικεντρώνετε στην ελάχιστη διαμόρφωση και στην ικανότητα να απλοποιεί τον container και να τον αφήνει μόνο με όσα χρειάζεται.
- Tiny Java Web Server (TJWS).

3. Παρουσίαση του site του McLab

Για την εκπόνηση της παρούσας πτυχιακής εργασίας καθώς και για την υλοποίηση της ιστοσελίδας του εργαστηρίου πολυμέσων χρειάστηκε να μελετηθούν διάφορα θεωρητικά μοντέλα , αλλά και γλώσσες προγραμματισμού, των οποίων το αποτέλεσμα συνιστά την τεχνική προγραμματισμού AJAX. Ενδεικτικά αναφέρονται οι γλώσσες προγραμματισμού που χρησιμοποιούνται με την τεχνική AJAX :

- CSS
- HTML
- JAVASCRIPT

Πλέον όμως τα τελευταία χρόνια έχει ξεκινήσει η χρήση της τεχνικής AJAX με την γλωσσά προγραμματισμού PHP, ο οποίος είναι ένας πολλά υποσχόμενος συνδυασμός.

Στην παρακάτω ενότητα που ακολουθεί θα γίνει η παρουσίαση του site του εργαστηρίου πολυμέσων του Τ.Ε.Ι. Κρήτης (McLab), το οποίο υλοποιήθηκε κάνοντας χρήση της τεχνικής Ajax, τόσο σε επίπεδο κώδικα αλλά και παρουσίασης. Πιο συγκεκριμένα θα δούμε τον πηγαίο κώδικα της σελίδας και θα δούμε βήμα προς βήμα πως υλοποιείτε η τεχνική Ajax.

Βεβαία για την ολοκλήρωση της ιστοσελίδας του εργαστηρίου πολυμέσων, χρησιμοποιήθηκαν και κάποια άλλα εργαλεία, τα οποία και παραθέτονται παρακάτω.

- DREAMWEAVER: το Dreamweaver είναι ένα λογισμικό ανάπτυξης ιστοσελίδων τόσο σε επίπεδο κώδικα αλλά και σε επίπεδο σχεδίασης. Το λογισμικό αυτό κυκλοφορεί από την εταιρία Adobe, και είναι από τα πιο διαδεδομένα του είδους του. Με το λογισμικό Dreamweaver αναπτύχτηκε ο HTML και JavaScript κώδικας της ιστοσελίδας του εργαστηρίου πολυμέσων αλλά και ο κώδικας του css.
- Apache Tomcat: Η εφαρμογή Apache Tomcat είναι μια εφαρμογή ανοιχτού λογισμικού, που επιτρέπει την εκτέλεση των java servlet και java server τεχνολογιών. Η εφαρμογή αυτή χρησιμοποιήθηκε για την επικοινωνία της ιστοσελίδας με τον server, έτσι ώστε να είναι δυνατή η ανταλλαγή δεδομένων μεταξύ τους.

3.1 το interface του site

Το περιεχόμενο του site εμφανίζετε χρησιμοποιώντας τα λειτουργικά κουμπιά τα οποία είναι:

- Το κουμπί Τα νέα μάς
- Το κουμπί Projects
- Το κουμπί επικοινωνία
- Το κουμπί Τελευταία νέα
- Το κουμπί About us
- Το κουμπί Πτυχιακές

Σε κάθε ένα από τα κουμπιά αυτά όταν πατιέται εμφανίζετε το σχετικό με τον τίτλο περιεχόμενο το οποίο αντλείτε από ένα αρχείο XML το οποίο βρίσκετε αποθηκευμένο στον server .

Παρακάτω παραθέτετε μια εικόνα με το site στην οποία φαίνονται όλα τα προαναφερθέντα στοιχεία.



Εικόνα 1: Η αρχική εμφάνιση της ιστοσελίδας

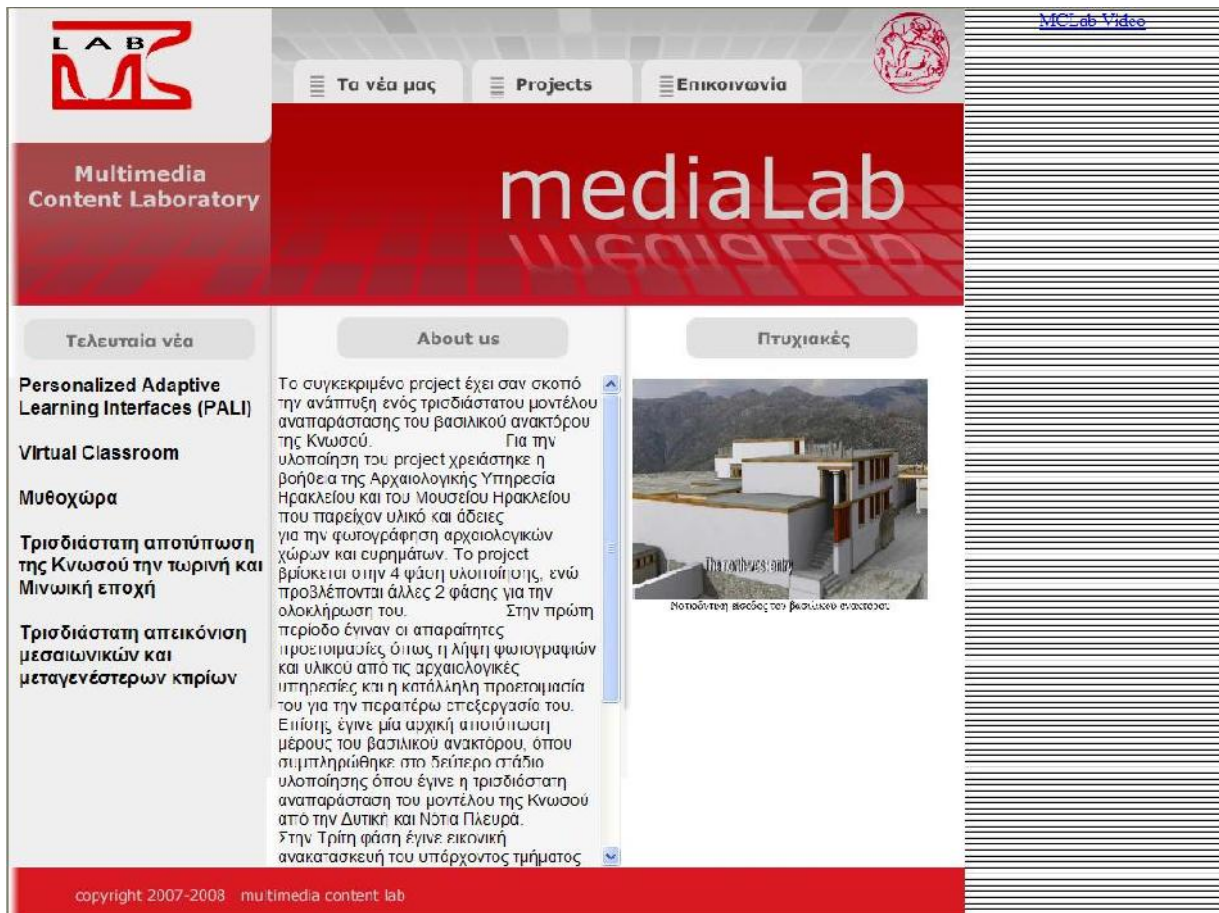
3.2 Λειτουργικότητα του site

Κάθε φορά που κάνουμε κλικ σε κάποιο από τα κουμπιά στη πρώτη, από τα αριστερά και κάτω από το κουμπί Τελευταία νέα, στήλη εμφανίζονται σε τίτλους τα περιεχόμενα της κάθε κατηγορίας.



Εικόνα 2 :Η ιστοσελίδα αφού πατήσουμε το κουμπί projects

Έπειτα ο χρήστης επιλέγει κάποιον από τους τίτλους που θέλει, κάνοντας κλικ και έτσι εμφανίζονται στη δεύτερη, από τα αριστερά και κάτω από το κουμπί About us, στήλη οι πληροφορίες που αφορούν το συγκεκριμένο τίτλο που έχει επιλέξει ο χρήστης καθώς και μια εικόνα που αφορά τον συγκεκριμένο τίτλο που έχει επιλεγεί.



Εικόνα 3: Η ιστοσελίδα αφού επιλέξουμε κάποιον από τους τίτλους

3.3 Ανάλυση του κώδικα

3.3.1 Εισαγωγή

Στην ενότητα αυτή θα δούμε-αναλύσουμε τον κώδικα της ιστοσελίδας. Ο κώδικας μας αποτελείται από τρία διαφορετικά κομμάτια τα οποία όλα μαζί συνθέτουν και κάνουν την ιστοσελίδα αυτή λειτουργική.

Το πρώτο κομμάτι αφορά τον κώδικα HTML το οποίο με τη σειρά του καλεί-κάνει χρήση των δυο άλλων κομματιών κώδικα που είναι οι συναρτήσεις JavaScript αλλά και ο κώδικας του CSS.

3.3.2 Ο κώδικας HTML

Παρακάτω παραθέτετε ο κώδικας HTML της ιστοσελίδας του McLab

```
<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<title>Untitled Document</title>

</head>

<body>

<link rel="StyleSheet" type="text/css" href="mLabCss.css" />

<script type="text/javascript" src="McLab.js"></script>

<div id="Layer27"></div>

<div id="Layer4"></div>

<div id="Layer5"></div>

<div id="Layer6"></div>

<div id="Layer7"></div>

<div id="Layer8"></div>

<div id="Layer9"></div>
```

```

<div id="Layer10"></div>

<div id="Layer11"></div>

<div id="Layer12"></div>

<div id="Layer13"></div>

<div width="100%" border="0" id="allResults" name="allResults" >
    <div width="790" border="0" id="Layer14" name="Layer14" >
        <span id="resultsBody" > </span>
    </div>

    <div id="Layer31"></div>

    <p><p><p>

    <div id="Layer23"></div>

    <div id="Layer32"></div>

    <div width="790" border="0" id="Layer21" name="Layer21">
        <span id="resultsBody1"> </span>
    </div>

    <div id="Layer33"></div>

    <div id="Layer24"></div>

    <div width="790" border="0" id="Layer22" name="Layer22">
        <span id="resultsBody2"> </span>
    </div>

```

```
</div>

<div id="Layer17"></div>

<div id="Layer19"></div>

<div id="Layer20"></div>

<div id="Layer25"></div>

<div id="Layer3"></div>

<div id="Layer26"></div>

<div id="Layer1"></div>

<div id="Layer29"></div>

<div id="Layer2">

  <div align="center"><a href="videos/medialab_showreel.mp4"
target="_blank">MCLab Video</a></div>

</div>

<div id="Layer16"></div>

<div id="Layer28"></div>

</body>

</html>
```

Εξετάζοντας τον παραπάνω κώδικα βλέπουμε ότι στο HTML έγγραφο, συναντάμε στη αρχή την ετικέτα link. Η ετικέτα link χρησιμοποιείτε για να συνδέσει ένα εξωτερικό αρχείο με το HTML έγγραφο, και παίρνει σαν ορίσματα διάφορες παραμέτρους. Μερικές από αυτές παρατίθενται παρακάτω:

- href : Στην ιδιότητα αυτή ορίζετε η διαδρομή στην οποία είναι αποθηκεμένο το αρχείο το οποίο συνδέουμε.
- Rel : Στην ιδιότητα αυτή ορίζετε η σχέση του εγγράφου με το συνδεόμενο έγγραφο.
- Type : Στην ιδιότητα αυτή ορίζετε ο τύπος του συνδεόμενου αρχείου.

Στο κώδικα μας ο τύπος είναι stylesheet και ο τύπος του είναι text/css.

Προχωρώντας συναντάμε την ετικέτα script, η οποία χρησιμοποιείτε για να οριστεί ένα script, σε διάφορες γλώσσες, το οποίο θα τρέχει τοπικά στον περιηγητή του χρήστη. Το script μπορεί είτε να οριστεί μέσα στην ετικέτα είτε να αντληθεί από ένα εξωτερικό αρχείο. Σε αυτή την ετικέτα είναι υποχρεωτικό να οριστεί η ιδιότητα type η οποία καθορίζει τον τύπο-γλώσσα που είναι γραμμένο το script. Μερικές από τις υπόλοιπες ιδιότητες της ετικέτας script οι οποίες είναι προαιρετικό στο να της δηλώσεις είναι οι ακόλουθες :

- Src : Στην ιδιότητα αυτή ορίζετε η διαδρομή στην οποία είναι αποθηκεμένο το script το οποίο συνδεεται με το έγγραφο.
- Charset: Στην ιδιότητα αυτή ορίζετε η κωδικοποίηση του κειμένου που χρησιμοποιείτε.

Στο κώδικα μας το script είναι γραμμένο σε γλώσσα JavaScript και αντλείτε εξωτερικά από ένα αρχείο (McLab.js).

Συνεχίζοντας την ανάγνωση του κώδικα συναντάμε την ετικέτα div. Η ετικέτα div χρησιμοποιείτε για να ορίσει ένα τμήμα ή ένα κομμάτι του εγγράφου HTML, ενώ συχνά χρησιμοποιείτε για να μορφοποιήσει τα διάφορα elements με ένα συγκεκριμένο στυλ css. Στην ετικέτα div συναντάμε τις ακόλουθες ιδιότητες :

- Id : Εδώ προσδιορίζεται μια μοναδική id για το αντικείμενο.
- Lang : Εδώ προσδιορίζεται η γλώσσα του περιεχομένου που χρησιμοποιείτε στο αντικείμενο
- Title : Εδώ προσδιορίζεται ο τίτλος του αντικειμένου.

Εκτός από τις ιδιότητες αυτές στην ετικέτα div υπάρχουν και action events τα οποία ενεργοποιούνται ανάλογα με την ενεργεία του χρήστη πάνω στο αντικείμενο. Τέτοια events είναι:

- **OnClick** : Το event αυτό ενεργοποιείτε όταν γίνει ένα κλικ του ποντικιού στο αντικείμενο
- **Onmousedown** : Το event αυτό ενεργοποιείτε όταν πατηθεί ένα κουμπί του mouse στο αντικείμενο
- **Onmouseup** : Το event αυτό ενεργοποιείτε όταν απελευθερωθεί το κουμπί του mouse στο αντικείμενο
- **Onkeydown** : Το event αυτό ενεργοποιείτε όταν πατηθεί ένα πλήκτρο στο αντικείμενο
- **Onkeyup** : Το event αυτό ενεργοποιείτε όταν απελευθερωθεί το πλήκτρο στο αντικείμενο
- **Onmouseover** : Το event αυτό ενεργοποιείτε όταν ο κέρσορας του ποντικιού περάσει πάνω από το αντικείμενο.

Στο div με id layer6 παρατηρούμε ότι στο event onclick καλείτε η συνάρτηση clearAllResults() η οποία περιγράφεται μέσα στο αρχείο McLab.js

3.3 ο κώδικας JavaScript

Σε αυτό το κεφαλαίο θα δούμε το και αναλύσουμε το κώδικα JavaScript που περιέχει το αρχείο McLab.js και το οποίο είναι απαραίτητο προκειμένου να μπορεί να λειτουργήσει το site του McLab χρησιμοποιώντας τη τεχνική Ajax.

```
var xmlhttp;

    var details="";

    var row;

    var xtag;

    var y;

    var target;

    var temp;

    function createXMLHttpRequest() {

        if (window.ActiveXObject) {

            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

        }

        else if (window.XMLHttpRequest) {
```

```

        xmlHttp = new XMLHttpRequest();
    }
}
function clearAllResults(y,temp){
    xtag=y;
    target=temp;
    var tableBody = document.getElementById("resultsBody");
    if (tableBody !=null){
        while(tableBody.childNodes.length > 0) {
            tableBody.removeChild(tableBody.childNodes[0]);
        }
        var tableBody =
document.getElementById("resultsBody1");
        while(tableBody.childNodes.length > 0) {
            tableBody.removeChild(tableBody.childNodes[0]);
        }
        var tableBody =
document.getElementById("resultsBody2");
        while(tableBody.childNodes.length > 0) {
            tableBody.removeChild(tableBody.childNodes[0]);
        }
    }
    bringData();
}

function bringData(){

```



```

        createXMLHttpRequest();

        var url = "test1";

        xmlHttp.onreadystatechange = handleStateChange;

        xmlHttp.open("GET", url, true);

        xmlHttp.send(null);
    }

function handleStateChange() {
    if(xmlHttp.readyState == 4) {
        if(xmlHttp.status == 200) {
            var x=null;

            var curneo=null;

            var newRes=xmlHttp.responseXML;

            var nea=newRes.getElementsByTagName(xtag);

            for (var i=0; i<nea.length; i++){
                curneo=nea[i];

                var
name=curneo.getElementsByTagName("name")[0].firstChild.nodeValue;

                var
details=curneo.getElementsByTagName("details");

                var
image=curneo.getElementsByTagName("url")[0].firstChild.nodeValue;

                var out=out + "\n" +
curneo.childNodes[0].nodeValue;

                addTableRow(name,details,image,target);
            }
        }
    }
}

```

```

        }
    }
}

function addTableRow(name, details, image, target) {

    row = document.createElement("span");
    row.setAttribute("id", "nea");
    var cell = createCellWithText(name, row);
    cell.setAttribute("width", "220");
    row.appendChild(cell);
    document.getElementById(target).appendChild(row);
    cell.onclick=function() {clearPrevRes(details, row);

        var k=0;
        while(k<details.length)
        {
            row = document.createElement("span");
            var
cell=createCellWithText(details[k].firstChild.nodeValue, row);
            row.appendChild(cell);

            document.getElementById("resultsBody1").appendChild(row);

            k++;
        }
        createCellWithImage(image);

```

```

};

}

function createCellWithImage(text) {
var text=text;
var cell1= document.createElement("a");
var cell = document.createElement("img");
cell1.align="center";
cell.src=text;
cell.width="250";
cell.height="200";
cell.setAttribute("id", "text");
cell1.appendChild(cell);
document.getElementById("resultsBody2").appendChild(cell1);
cell.onclick=function(){cell1.href=text;
cell1.target="_blank";}
}

function createCellWithText(text,grammh) {
var text=text;
var cell = document.createElement("p");
var textNode = document.createTextNode(text);
cell.appendChild(textNode);
cell.setAttribute("width", "500");
cell.setAttribute("id", "text");
grammh.appendChild(cell);
return cell;
}

```

```

}

function clearPrevRes (details, row) {

    var line=document.getElementById("resultsBody");

    if (line !=null){

        var line = document.getElementById("resultsBody1");

        while(line.childNodes.length > 0) {

            line.removeChild(line.childNodes[0]);

        }

        var line1 = document.getElementById("resultsBody2");

        while(line1.childNodes.length > 0) {

            line1.removeChild(line1.childNodes[0]);

        }

    }

}

```

Η πρώτη συνάρτηση που καλείτε από την HTML είναι η `ClearAllResults()` και καλείτε όταν ενεργοποιείτε το event `onclick` της εικόνας δηλαδή όταν ο χρήστης κάνει `click` σε κάποιο από τα κουμπιά που υπάρχουν στο site. Η συνάρτηση `ClearAllResults()` παίρνει σαν όρισμα μια λέξη η οποία αντιστοιχεί στο όνομα της κατηγορίας που υπάρχει μέσα στο αρχείο XML, έτσι ώστε να αντληθούν τα κατάλληλα δεδομένα για κάθε κουμπί. Το δεύτερο όρισμα που περιέχετε αντιστοιχεί στο όνομα του αντικειμένου της HTML στο οποίο θα προβληθούν οι πληροφορίες που αντλούνται από το XML.

Στη συνέχεια η συνάρτηση βάζει στη μεταβλητή `tableBody` όλα τα στοιχεία που έχουν `id` το `resultsBody`. Έπειτα εξετάζετε αν η μεταβλητή αυτή είναι κενή ή όχι. Αν δεν είναι κενή τότε

ενεργοποιείτε ένας βρόχος while ο οποίος αφαιρεί όλα τα περιεχόμενα-children του resultsBody εωσότου μείνει κενό. Ομοίως γίνεται και για τα resultsBody1 και resultsBody2. Αυτή η διαδικασία γίνεται έτσι ώστε να εμφανίζονται σε κάθε κουμπί τα περιεχόμενα που αφορούν μόνο τη συγκεκριμένη κατηγορία και να μην υπάρχουν τυχόν προηγούμενα αποτελέσματα. Τέλος καλείτε η συνάρτηση bringData().

Στη συνάρτηση bringData() πρώτα πρώτα καλείτε η συνάρτηση createXMLHttpRequest() η οποία δημιουργεί ένα νέο αντικείμενο τύπου XMLHttpRequest, εάν το πρόγραμμα περιήγησης μας είναι ο Internet Explorer, ενώ σε κάθε άλλη περίπτωση δημιουργείτε ένα XMLHttpRequest αντικείμενο. Και τα δυο αυτά αντικείμενα αποτελούν τη «καρδιά» της τεχνικής Ajax.

Αφού δημιουργηθεί ένα από τα δυο αντικείμενα ορίζετε στη μεταβλητή url το όνομα του servlet που χρησιμοποιείτε και ορίζουμε την συνάρτηση που θα κληθεί σε οποία αλλαγή του onreadystatechange του XMLHttpRequest αντικειμένου. Στη περίπτωση μας αυτή είναι η handleStateChange().

Όταν καλείτε η handleStateChange() γίνεται αρχικά ένας έλεγχος για να διαπιστωθεί η κατάσταση του server μέσω της ιδιότητας readyState του XMLHttpRequest. Εάν η ιδιότητα readyState του XMLHttpRequest αντικειμένου έχει την τιμή 4 τότε η αίτηση μας προς το server έχει ολοκληρωθεί. Έπειτα εξετάζετε το status του XMLHttpRequest αντικειμένου. Εάν το status έχει την τιμή 200 τότε σημαίνει ότι όλα έγιναν όπως έπρεπε κατά την ανταλλαγή των δεδομένων.

Έπειτα αποθηκεύεται σε μια μεταβλητή η απάντηση του server στο XMLHttpRequest αντικείμενο σε μορφή XML, και στη συνέχεια επιλέγουμε τα elements της κατηγορίας που έχουμε επιλέξει (πατώντας το κατάλληλο κουμπί) από το XML. Αφού έχουμε πάρει τα στοιχεία της κατηγορίας, αποθηκεύουμε στις μεταβλητές name, details και image τα child nodes της κατηγορίας αυτής και καλούμε την συνάρτηση addTableRow(name,details,image,target).

Η συνάρτηση addTableRow(name,details,image,target) παίρνει σαν ορίσματα της μεταβλητές name,details,image και target οι οποίες αντιστοιχούν στους τίτλους της κατηγορίας, στις πληροφορίες για κάθε τίτλο , στις εικόνες για κάθε τίτλο και στο πλαίσιο στο οποίο θα εμφανιστούν οι τίτλοι της κατηγορίας. Στη συνέχεια δημιουργείτε ένα αντικείμενο τύπου span με την εντολή document.createElement("span") και ορίζετε το id του αντικειμένου που δημιουργήθηκε με την εντολή setAttribute("id", "nea"). Έπειτα δημιουργούμε cells μέσα στο span κάνοντας κλήση στη συνάρτηση createCellWithText().

Η συνάρτηση createCellWithText() παίρνει σαν ορίσματα το κείμενο το οποίο θα περιέχετε στο cell που θα δημιουργηθεί καθώς και τη γραμμή στην οποία θα δημιουργηθεί το cell. Έπειτα δημιουργείτε ένα αντικείμενο τύπου παραγράφου και ένα text node με το κείμενο που έχουμε περάσει σαν ορίσμα. Τελειώνοντας βάζουμε το text node μέσα στο cell που δημιουργήσαμε, έπειτα προσδιορίζουμε τις ιδιότητες width που είναι το πλάτος και id του cell και βάζουμε το

cell στη γραμμή που πήραμε σαν όρισμα. Τέλος επιστρέφετε πίσω στη συνάρτηση `addTableRow()` το cell που δημιουργήθηκε.

Στη συνάρτηση `addTableRow()` συνεχίζοντας ορίζουμε σαν child το cell που δημιουργήσαμε και μετά ορίζουμε σαν child τη γραμμή που δημιουργήσαμε, στο πλαίσιο που είχαμε περάσει σαν όρισμα στη συνάρτηση. Το οπτικό αποτέλεσμα που έχουμε πάρει μέχρι στιγμής φαίνεται στην εικόνα 2 σελ.41.

Για να μπορέσουμε να έχουμε τις πληροφορίες που θέλουμε καθώς και την εικόνα για κάθε τίτλο στα διπλανά από τους τίτλους πλαίσια, χρησιμοποιούμε το event του κάθε cell που περιέχει τίτλο, `onclick`. Στο event αυτό ορίζεται μια συνάρτηση (function) η οποία αρχικά κάνει κλήση στη συνάρτηση `clearPrevRes()`

Η συνάρτηση `clearPrevRes()` χρησιμοποιείται για να αφαιρέσει τυχόν προηγούμενα αποτελέσματα που υπάρχουν στο πλαίσιο τα οποία δεν θέλουμε να υπάρχουν εκεί αφότου επιλέξουμε άλλο τίτλο. Σε επίπεδο κώδικα αλλά και λειτουργίας η συνάρτηση `clearPrevRes()` με την `clearAllResults()` είναι ίδιες με μοναδική διαφορά ότι η `clearPrevRes()` δεν αφαιρεί τους τίτλους που υπάρχουν στο πλαίσιο `resultsBody` αλλά μονό τα περιεχόμενα των τίτλων και τις εικόνες.

Στη συνέχεια συναντάμε ένα βρόχο `while` ο οποίος αναλαμβάνει να δημιουργήσει γραμμές με αντικείμενα `span` και `cell` με `text`, κάνοντας χρήση της συνάρτησης `createCellWithText()` που περιγράψαμε παραπάνω. Μοναδική διαφορά με πριν, είναι ότι αυτή τη φορά τα `clid nodes` προστίθενται στο πλαίσιο `resultsBody1`. Η διαδικασία αυτή επαναλαμβάνεται σε περιπτώσεις όπου οι ετικέτες `details` στο XML αρχείο είναι περισσότερες από μια. Τέλος καλείτε και η συνάρτηση `createCellWithImage()`.

Η συνάρτηση `createCellWithImage(image)` παίρνει σαν όρισμα μια μεταβλητή που ονομάζετε `image` και περιέχει τη διαδρομή στην οποία είναι υποθηκευμένες οι διάφορες εικόνες που αφορούν κάθε τίτλο. Αρχικά δημιουργείτε ένα αντικείμενο τύπου `<a>` και ένα αντικείμενο τύπου ``. Διαμορφώνετε η ιδιότητα `align` της ετικέτας `<a>` με τη τιμή `center`, και οι ιδιότητες `height`, `width` και `src` που ορίζει τη διαδρομή που είναι αποθηκευμένο το αρχείο της εικόνας. Έπειτα προστίθεται σαν child η ετικέτα `` στην ετικέτα `<a>`, και αυτή με τη σειρά της προστίθεται σαν child στο πλαίσιο `resultsBody2`. Ο λόγος για τον οποίο προστίθεται η ετικέτα `` μέσα στην ετικέτα `<a>` είναι για να μπορεί να λειτουργεί η εικόνα σαν link και να ανοίγει σε νέο παράθυρο όταν ενεργοποιείτε το event `onclick`. Αυτό επιτυγχάνετε με την ακόλουθη εντολή `cell.onclick=function(){cell1.href=text; cell1.target="_blank";}` όπου `cell` είναι η ετικέτα ``, `cell1` η ετικέτα `<a>` και `text` η διαδρομή του αρχείου εικόνας. Με τη δήλωση της ιδιότητας `cell1.target="_blank"` επιτυγχάνουμε να ανοίγεται ένα νέο παράθυρο με την εικόνα.

Έτσι με την ολοκλήρωση της συνάρτησης `createCellWithImage()` ολοκληρώνετε και η συνάρτηση `addTableRow()` και επιστρέφουμε στην συνάρτηση `handleStateChange()` όπου συναντάμε δυο ακόμη εντολές. Η εντολή `xmlHttp.open("GET", url, true)` είναι αυτή που

δημιουργεί την σύνδεση με το servlet. Με το πρώτο ορισμα προσδιορίζετε η ενέργεια που θα εκτελεστεί, στο δεύτερο ορισμα προσδιορίζετε η διεύθυνση του servlet ενώ τέλος με το τρίτο ορισμα ελέγχετε για το αν η επικοινωνία θα είναι ασύγχρονη ή όχι. Η τελευταία εντολή της συνάρτησης bringData() είναι η xmlhttp.send(null) η οποία είναι υπεύθυνη για να αποστείλει δεδομένα στο servlet, αλλά εφόσον στο site δεν χρειαζόμαστε να στέλνουμε δεδομένα αλλά μόνο να παίρνουμε βάζουμε σαν ορισμα τη λέξη null δηλαδή δεν στέλνουμε τίποτα.

Με την ολοκλήρωση της συνάρτησης bringData(), ολοκληρώνετε και η εκτέλεση της συνάρτησης clearAllResults(), έχοντας πάρει και εμφανίσει όλα τα δεδομένα που θέλαμε. Το οπτικό αποτέλεσμα της ιστοσελίδας μας είναι αυτό που φαίνεται στην εικόνα 3 στη σελίδα 42.

3.4 Το αρχείο XML

Οι πληροφορίες όλες που εμφανίζονται πατώντας τα διάφορα κουμπιά της ιστοσελίδας είναι αποθηκευμένες μέσα σε ένα αρχείο XML, από το οποίο και αντλούνται. Η δομή του αρχείου XML που χρησιμοποιείτε στην ιστοσελίδα McLab φαίνεται παρακάτω :

```
<nea>
  <neo>
    <name>Νέο site</name>
    <details>Εναρξη λειτουργίας του νέου site</details>
    <url>images/mclab_logo.jpg</url>
  </neo>

  <ptixiakes>
    <name>Ανάπτυξη συστήματος διαγραμματειακής
υποστήριξης</name>
    <details>Σκοπός της πτυχιακής εργασίας ήταν η ανάπτυξη
      ενός διαδικτυακού συστήματος που θα συμπλήρωνε
      το σύστημα γραμματειακής διαχείρισης ΠΑΝΔΩΡΑ.
      Το σύστημα αναπτύχθηκε χρησιμοποιώντας την γλώσσα
      προγραμματισμού ASP .NET, για την δημιουργία
      των ιστοσελίδων και η σχεσιακή βάση δεδομένων
```

Microsoft SQLServer 2000. Το ολοκληρωμένο σύστημα
έδινε την δυνατότητα στους χρήστες του συστήματος
να ολοκληρώσουν οποιαδήποτε διαδικασία που
απαιτούσε

την παρουσία τους στην γραμματεία της σχολής,
μέσα από το Διαδίκτυο.</details>

<url>images/2.jpg</url>

</ptixiakes>

<ptixiakes>

<name>Ανάπτυξη ευφυούς αποκεντρωμένου εξυπηρετητή</name>

<details>Σκοπός της πτυχιακής εργασίας ήταν η μελέτη των
agent

τεχνολογιών και η υλοποίηση μίας εφαρμογής που θα
εκμεταλλεύονταν

τις agent τεχνολογίες. Η εφαρμογή αναπτύχθηκε
χρησιμοποιώντας την

γλώσσα προγραμματισμού Visual Basic 6.0 και το πρωτόκολλο
επικοινωνίας

ανάμεσα στους υπολογιστές την γλώσσα αναπαράστασης
δεδομένων XML.

Η ολοκληρωμένη εφαρμογή ήταν ένας file-sharing agent
βασισμένος

σε αρχές πρωτοκόλλου Gnutella, και ένα πρότυπο πρωτόκολλο
επικοινωνίας.</details>

<url>images/2.jpg</url>

</ptixiakes>

<ptixiakes>

<name>Ανάπτυξη εφαρμογής τηλεφωνικής υποστήριξης</name>

<details>Σκοπός της πτυχιακής εργασίας ήταν η μελέτη και
ενσωμάτωση

των τεχνολογιών SAPI και TAPI σε μία εφαρμογή που θα παρείχε on-line

υπηρεσίες μέσω τηλεφώνου. Η εφαρμογή αναπτύχθηκε χρησιμοποιώντας

την γλώσσα προγραμματισμού Visual Basic 6.0. Η ολοκληρωμένη εφαρμογή

ήταν μία εφαρμογή τύπου τηλεφωνικού κέντρου, που αναλάμβανε να παρέχει

οδηγίες σχετικά με την ολοκλήρωση ορισμένων διαδικασιών που απαιτούσε

ο χρήστης μέσω τηλεφώνου.</details>

<url>images/2.jpg</url>

</ptixiakes>

<projects>

<name>Personalized Adaptive Learning Interfaces (PALI)</name>

<details> Το PALI είναι ένα πολυεπίπεδο προσωποποιημένο σύστημα που ενσωματώνει έναν αριθμό από τεχνολογίες με σκοπό την προσωποποιημένη παροχή υπηρεσιών στον τελικό χρήστη.

Το σύστημα αποτελείται από μία εφαρμογή που εγκαθίσταται σε φορητές συσκευές μέσω τις οποίας ο τελικός χρήστης αποκτά πρόσβαση στις υπηρεσίες που παρέχει το σύστημα.

Το κυρίως σύστημα είναι ένας web server με δυνατότητα εκτέλεσης web services που αναλαμβάνει μέσω μίας ενδιάμεσης εφαρμογής να δρομολογεί τα μηνύματα των χρηστών

στις κατάλληλες web services μέσω ανταλλαγής SOAP μηνυμάτων. Οι web services αναλαμβάνουν να προωθήσουν το τελικό μήνυμα μέσω sockets, στην end-user εφαρμογή που εκτελείται

σε απομακρυσμένους υπολογιστές που έχει πρόσβαση ο χρήστης για την ολοκλήρωση των αιτήσεων. Οι παρεχόμενες υπηρεσίες είναι μεταξύ άλλων:

</details><details>

1) Απομακρυσμένη πρόσβαση σε personal information manager εφαρμογές όπως το MS Outlook.

</details><details>

2) Απομακρυσμένη πρόσβαση σε πόρους τοπικών δικτύων όπως εκτυπωτές και fax.

</details><details>

3) Απομακρυσμένη πρόσβαση σε εταιρική πληροφορία (έγγραφα, βάσεις δεδομένων).

</details><details>

4) Απομακρυσμένη κράτηση ταξιδιωτικών εισιτηρίων και on-line αγορών.

</details><details>

5) Απομακρυσμένη προσωποποιημένη διαδικτυακή αναζήτηση σε μηχανές αναζήτησης.

</details><details>

6) Απομακρυσμένη ανταλλαγή αρχείων.

</details><details>

Η διαδικασία προσωποποίησης των υπηρεσιών περιλαμβάνει την ύπαρξη ενός παραμετροποιήσιμου προφίλ χρηστών. Το προφίλ αυτό είναι αποθηκευμένο σε δομημένη μορφή (XML), σε έναν

profile database server (eXist). Το προφίλ ανανεώνεται μέσω μίας κατάλληλης διεπαφής που παρακολουθεί την επικοινωνία ανάμεσα στην ενδιαμέση εφαρμογή και τις web services και ανανεώνει

χρησιμοποιώντας Xqueries το προφίλ του κάθε χρήστη με τις αλλαγές που συνέβησαν. Μέχρι το στάδιο ολοκλήρωσης του έργου προβλέπεται η ενσωμάτωση τεχνολογιών για την

σημασιολογική αναζήτηση των δεδομένων του χρήστη καθώς και τεχνικές τεχνητής νοημοσύνης στα στάδια προσωποποίησης των υπηρεσιών.</details>

<url>images/pali.jpg</url>

</projects>

<projects>

<name>Virtual Classroom</name>

<details>Το έργο έχει ολοκληρωθεί ως work package στα πλαίσια του έργου που ανέλαβε το ΤΕΙ Κρήτης από την προκήρυξη 98 της Κοινωνίας της Πληροφορίας.

Πρόκειται για εφαρμογή εικονικής αίθουσας κατασκευασμένη εξολοκλήρου στο εργαστήριο με τεχνολογία JAVA και JMF και αποτελεί βασική πλατφόρμα σύγχρονη διδασκαλίας μέσω δικτύου με ανταλλαγή πολυμεσικής πληροφορίας.

</details>

<url>images/virtualclass.jpg</url>

</projects>

<projects>

<name>Μυθοχώρα</name>

<details>Το project Μυθοχώρα αναπτύχθηκε σε συνεργασία με την εταιρεία Mentoring ΑΕ, για λογαριασμό του ΕΔΙΑΜΜΕ, του παιδαγωγικού τμήματος του Πανεπιστημίου Κρήτης και είναι

μία πολυμεσική εφαρμογή ανεπτυγμένη με την εφαρμογή Macromedia Director. Σκοπός της εφαρμογής είναι να παρέχει συμπληρωματικές πληροφορίες σχετικά με την ελληνική μυθολογία

στο βιβλίο του ΕΔΙΑΜΜΕ που διανέμεται σε σχολεία του απόδημου ελληνισμού. </details>

<url>images/mithoxora.jpg</url>

</projects>

<projects>

<name>Τρισδιάστατη αποτύπωση της Κνωσού την τωρινή και Μινωική εποχή</name>

<details>Το συγκεκριμένο project έχει σαν σκοπό την ανάπτυξη ενός τρισδιάστατου μοντέλου αναπαράστασης του βασιλικού ανακτόρου της Κνωσού.

Για την υλοποίηση του project χρειάστηκε η βοήθεια της Αρχαιολογικής Υπηρεσία Ηρακλείου και του Μουσείου Ηρακλείου που παρείχαν υλικό και άδειες

για την φωτογράφιση αρχαιολογικών χώρων και ευρημάτων. Το project βρίσκεται στην 4 φάση υλοποίησης, ενώ προβλέπονται άλλες 2 φάσεις για την ολοκλήρωση του.

Στην πρώτη περίοδο έγιναν οι απαραίτητες προετοιμασίες όπως η λήψη φωτογραφιών και υλικού από τις αρχαιολογικές υπηρεσίες και η κατάλληλη προετοιμασία του για την περαιτέρω επεξεργασία του.

Επίσης έγινε μία αρχική αποτύπωση μέρους του βασιλικού ανακτόρου, όπου συμπληρώθηκε στο δεύτερο στάδιο υλοποίησης όπου έγινε η τρισδιάστατη αναπαράσταση του μοντέλου της Κνωσού από την Δυτική και Νότια Πλευρά.

Στην Τρίτη φάση έγινε εικονική ανακατασκευή του υπάρχοντος τμήματος του Παλατιού, ενώ παράλληλα έγινε εκκίνηση της τέταρτης φάσης του παλατιού της Κνωσού που αποτελείται

από την αποτύπωση και τρισδιάστατη αναπαράσταση του Βόρειου και Ανατολικού τμήματος της Κνωσού. Στις δύο τελευταίες φάσεις της δημιουργίας του βασιλικού ανακτόρου της Κνωσού προβλέπεται

η αναστήλωση του παλατιού όπως εικάζεται ότι ήταν την μινωική εποχή και η δημιουργία εικονικής περιήγησης με χρήση προκαθορισμένων και μη διαδρομών.

</details>

<url>images/knossos.jpg</url>

</projects>

<projects>

<name>Τρισδιάστατη απεικόνιση μεσαιωνικών και μεταγενέστερων κτιρίων</name>

<details>Το συγκεκριμένο project έχει σαν στόχο την τρισδιάστατη μοντελοποίηση και απεικόνιση μεσαιωνικών και μεταγενέστερων κάστρων. Τα κτίρια τα οποία αυτήν την στιγμή

βρίσκονται υπό ανάπτυξη είναι το κάστρο των Άπτερα στην περιοχή των Χανίων, το Φραγκοκάστελο στην περιοχή των Χανίων, ενώ γίνονται σκέψεις για την τρισδιάστατη απεικόνιση

του φρουρίου του Κούλε, στην περιοχή του Ηρακλείου. Για την ανάπτυξη των τρισδιάστατων μοντέλων χρησιμοποιήθηκαν στοιχεία από αρχαιολογικές υπηρεσίες και σχέδια από πολιτικούς μηχανικούς.

Επίσης, στοιχεία προήλθαν από φωτογραφήσεις στον χώρο των κτιρίων

</details>

```
<url>images/aptera.jpg</url>

</projects>

<staff>

  <name>Υπεύθυνος Εργαστηρίου</name>

  <details>Δρ. Μαλάμος Αθανάσιος  Επίκουρος Καθηγητής του
Τμήματος Εφαρμοσμένης Πληροφορικής και Πολυμέσων του ΤΕΙ
Κρήτης</details>

  <url>images/1.jpg</url>

</staff>

<staff>

  <name>Μέλη</name>

  <details>α) Δρ. Ιωάννης Καλιακάτσος, Καθηγητής Τ.Ε.Ι. Κρήτης
</details><details>

    β) Δρ. Δημήτριος Καραγιαννάκης, Καθηγητής Τ.Ε.Ι.
Κρήτης</details><details>

    γ) Δρ. Ιωάννης Παχουλάκης, επικ. Καθηγητής Τ.Ε.Ι.
Κρήτης</details><details>

</details>

  <url>images/1.jpg</url>

</staff>

<staff>

  <name>Συνεργάτες</name>

  <details>Μαμάκης Γεώργιος, Συντονιστής έργων,
Προγραμματιστής</details><details>

    Μαυραγάνης Κωνσταντίνος,
Προγραμματιστής</details><details>

    Φραγκούλης Εμμανουήλ, Προγραμματιστής,
Databases</details><details>

    Κωσταντουράκης Ιωσήφ, 3D</details><details>

    Σύμπα Παρασκευή, 3D</details><details>
```

Τσιράκης Μιχαήλ, 3D και Multimedia</details><details>

Δαμιανίδου Χαρίκλεια, Graphics</details><details>

</details>

<url>images/1.jpg</url>

</staff>

<staff>

<name>Δημοσιεύσεις</name>

<details># Α.Γ. Μαλάμος, Γ. Μαμάκης, Ι. Καλιακάτσος, Δια βίου εκπαίδευση, μία μελέτη πάνω στις τεχνολογικές απαιτήσεις που δημιουργεί και στις λύσεις που προσφέρει η σύγχρονη τεχνολογία, «Προσεγγίσεις της Εκπαίδευσης Από Απόσταση Στην Τριτοβάθμια Εκπαίδευση», Καλογιαννάκης Ε, Βασιλάκης Κ., ISBN 978-960-6622-96-0

</details><details>

A.G. Malamos, Chapter "Quality of Service Adaptive Control in Multimedia Telecommunication Services", in book "Quality and Reliability of Large Scale Telecommunication Services", Wiley Publishers, USA

</details><details>

JOURNALS

</details><details>

G.A. Rovithakis, A.G. Malamos T.A. Varvarigou and M.A. Christodoulou, Quality of Service Adaptive Control of Multimedia Services in Short-Term Resource Reservation Networks, Cybernetics and Systems Journal, Volume 36, Number 3, pp. 251-281, April-May 2005

</details><details>

A.G. Malamos, T.A. Varvarigou, A.G. Axaridou, Scheduling Jobs with Critical and Non-critical Tasks in Shops with No-Wait Multipurpose Machines and Deadlines, under review in Nova Publishers Journal of Applied Mathematics

</details><details>

CONFERENCES

</details><details>

Malamos A. G., Sympa P. V., Mamakis G. , Kaliakatsos Y.,EVIE-M (Educational Virtual Environment Mathematics). An Alternative Approach For An Educational Strategy Game For Mathematics., Proceedings of CBLIS 07 Conference, Herakleion Crete, Greece, 30 June - 6 July 2007

</details><details>

Malamos A.G., Mamakis G., Sympa P., Tsirakis M., Piperidis G., Karakechagias J., Mavraganis K., Kaliakatsos Y.,VCLASS-3D: A Multimedia Educational Collaboration Platform With 3D Virtual Workspace Support, to be presented at WBE 06, Puerto Vallarta, 23-25 January 2006

</details><details>

Malamos, A. G. Mamakis, G. Anastasopoulos, T. Drikoudis, J. Fragoulis, E. Mavraganis, K. Moutou, A. Tesseris, G. Kaliakatsos, Y.,P.A.L.I. - A Platform for Secretarial Services Support to Business Users, Proceedings of Information and Communications Technology, 2006. ICICT '06. ITI 4th International Conference, Cairo, Egypt

</details><details>

A. G. Malamos, G. Mamakis, Y. Kaliakatsos, A.Axaridou, JA Ware,Meeting Lifelong education requirements in handling unstructured information through an automatic content handling system, Proceedings of the 5th Panhellenic Conference, Information and Communication Technologies in Education, October 2006, Thessaloniki, Greece

</details><details>

Sympa P.V. Malamos A.G., Mamakis G., Kaliakatsos Y.,Virtual Reality Application in Education, Proceedings of the 5th Global Congress on Engineering Education, 17-21 July 2006, New York, USA

</details><details>

A. G. Malamos, G. Mamakis, J. A. Ware,Applying Statistic-based Algorithms for Automated Content Summarization in Greek language, Proceedings of TEMU 2006 Conference, Herakleion Crete, 5-7 July 2006

</details><details>

Mamakis G., Malamos A.G., Kaliakatsos Y., Axaridou A., Ware A., An algorithm for automatic content summarization in modern greek language, in Proceedings of ICICT 05, Cairo, Egypt, 5-6 December 2005

</details><details>

Malamos A.G., Mamakis G., Anastasopoulos T., Kaliakatsos Y., Axaridou A., Enhancing Access in Knowledge for Adults by applying WWW (whenever-whenever-whatever) capabilities in we-based education. A case study of virtual workspaces technology adoption in learning environment, in Proceedings of NHIBE 05, Corfu, Greece, 25-26 August 2005

</details><details>

Mamakis G., Malamos A.G., A presentation of project Personalized Adaptive Learning Interfaces (PALI): Goals and Process of Development, in Proceedings of TEMU 05 Conference, Ammoudara, Herakleion Crete , 23-26 June 2005

</details><details>

Tesseris G., Mamakis G., Malamos A.G., Remote access to calendar applications using open-source technologies, in Proceedings of TEMU 05 Conference, Ammoudara, Herakleion Crete , 23-26 June 2005

</details><details>

Fragkoulis E., Mamakis G., Malamos A.G., Manipulating dynamic user profiles with the use of XML databases, in Proceedings of TEMU 05 Conference, Ammoudara, Herakleion Crete , 23-26 June 2005

</details><details>

Moutou., Mamakis G., Malamos A.G., Enhancing distributed platforms with eCommerce by using web-services and cXML technology, in Proceedings of TEMU 05 Conference, Ammoudara, Herakleion Crete , 23-26 June 2005

</details><details>

Mavraganis K., Malamos A.G., Remote access between heterogeneous systems over internet, a cross-technology integration solution, in Proceedings of TEMU 05 Conference, Ammoudara, Herakleion Crete , 23-26 June 2005

</details><details>

Elkarouni K., Malamos A.G., Gerard F., Remote access in Local Network Resources with the use of Web Services, in Proceedings of TEMU 05 Conference, Ammoudara, Herakleion Crete , 23-26 June 2005

</details><details>

Urganci G., Malamos A.G., Vilain D., Remote access to calendar applications using open-source technologies, in Proceedings of TEMU 05 Conference, Ammoudara, Herakleion Crete , 23-26 June 2005

</details><details>

Malamos A.G., Mamakis G., Anastasopoulos T., Kaliakatsos Y., Axaridou A., Enhancing Access in Knowledge for Adults by Applying WWW (whenever- wherever- whatever) Capabilities in Computer Based Education. A Case Study of Mobile and Web Service Technology Adoption in For Life, in Proceedings of "New Technologies in for-life education" Panhellenic Conference, Lamia, Greece, 15-16 April 2005

</details><details>

Mamakis G., Malamos A.G., A novel peer-to-peer application with dedicated extensions for teleworking applications, in Proceedings of the 9th International Telework Workshop, Hersonnisos, Crete, 6-9 August 2004

</details><details>

Malamos A.G., Kaliakatsos Y., Axaridou A., Improving training by applying WWW (whenever- wherever- whatever) Capabilities to learning platforms, in Proceedings of NHIE 03, pp 201-206, Santorini, Greece, 28-29 August 2003

</details>

<url>images/1.jpg</url>

</staff>

<contact>

<name>Athanasios G. Malamos</name>

<details>Associate Professor, Department of Applied Informatics and Multimedia, Technological Educational Institute of Crete

e-mail: amalamos@epp.teicrete.gr</details>

```
<url>images/1.jpg</url>
</contact>
<contact>
  <name>Mamakis Georgios</name>
  <details>Programmer
  Postgraduate Student, Project Coordinator, University of Glamorgan
  e-mail: gmamakis@epp.teicrete.gr </details>
  <url>images/1.jpg</url>
</contact>
<contact>
  <name>Paraskevi Sympa</name>
  <details>Graphics
  Postgraduate Student, School of Computing, University of Glamorgan,
  Wales
  e-mail: esympa@teiher.gr</details>
  <url>images/1.jpg</url>
</contact>
<contact>
  <name>Dimitrakopoulos Dimitris</name>
  <details>
  Student, Department of Applied Informatics and Multimedia, TEI Crete
  e-mail: dimitrid@tele2.be</details>
  <url>images/1.jpg</url>
</contact>
</nea>
```

Στο αρχείο XML παρατηρούμε ότι τα δεδομένα είναι μορφοποιημένα σε δενδροειδή διάταξη. Πρώτα έχουμε τη κατηγορία nea η οποία έχει ως childs τις κατηγορίες neo, Ptixiakes, Projects, staff και contact καθεμιά από τις οποίες έχει σαν childs τις κατηγορίες name, details, url. Το name αντιστοιχεί στους τίτλους που εμφανίζονται μόλις πατάμε κάθε κουμπί ενώ τα details και url αφορούν της λεπτομέρειες και τη διαδρομή της εικόνας όπου είναι αποθηκευμένη, αντίστοιχα.

4. Συμπεράσματα για την τεχνολογία AJAX

Δεν μπορούμε να είμαστε απόλυτοι και κατηγορηματικοί για το αν η τεχνολογία Ajax είναι καλό να χρησιμοποιείτε ή όχι, και σε αυτό το κεφάλαιο δεν θα υποστηρίξουμε κάποιο από τα δυο. Ωστόσο όμως θα παραθέσουμε όλα τα στοιχεία εκείνα που θεωρούμε σαν πλεονεκτήματα αλλά και σαν μειονεκτήματα.

4.1 Πλεονεκτήματα

Τα πλεονεκτήματα της τεχνολογίας Ajax εντοπίζονται στα εξής σημεία :

- Σε πολλές περιπτώσεις, οι σελίδες που είναι σχετικές μεταξύ τους σε έναν δικτυακό τόπο ένα μεγάλο μέρος του περιεχομένου τους είναι κοινό μεταξύ τους. Με παραδοσιακές μεθόδους, το περιεχόμενο θα πρέπει να ξαναφορτώνεται σε κάθε αίτηση. Ωστόσο, χρησιμοποιώντας την τεχνολογία Ajax, μια διαδικτυακή εφαρμογή μπορεί να ζητήσει μόνο το περιεχόμενο που πρέπει να ενημερωθεί, μειώνοντας έτσι δραστικά χρήση του εύρους ζώνης και του χρόνου φόρτωσης.
- Η χρήση των ασύγχρονων αιτήσεων επιτρέπει στο UI του Web browser του πελάτη να είναι πιο διαδραστικό και να ανταποκρίνεται γρήγορα στις εισόδους, και τα τμήματα των σελίδων που μπορούν επίσης να επαναφορτωθούν ατομικά. Οι χρήστες μπορούν να αντιλαμβάνονται η αίτηση να είναι ταχύτερη και πιο ευαίσθητη, ακόμη και αν η αίτηση δεν έχει αλλάξει από την πλευρά του διακομιστή.
- Η χρήση του Ajax μπορεί να μειώσει τις συνδέσεις στο διακομιστή, αφού scripts και style sheets πρέπει να ζητηθούν μία φορά μόνο.
- Η κατάσταση μπορεί να διατηρηθεί σε ολόκληρη μια τοποθεσία στο Web. Οι μεταβλητές της JavaScript θα εξακολουθήσουν να υπάρχουν, διότι η κύρια σελίδα container δεν χρειάζεται να επαναφορτωθεί.

4.2 Μειονεκτήματα

Στον αντίποδα τα μειονεκτήματα της τεχνολογίας Ajax εντοπίζονται στα παρακάτω σημεία :

- Η ευκολία της χρήσης των διεπαφών που χρησιμοποιούν Ajax συχνά αυξάνουν δραματικά τον αριθμό των χρηστών που δημιουργούν αιτήσεις στους web servers και τα πίσω τους άκρα (βάσεις δεδομένων, ή άλλα). Αυτό μπορεί να οδηγήσει σε βραδύτερους

χρόνους απόκρισης και πρόσθετο υλικό (hardware) για να μπορέσει να υποστηρίξει διασυνδέσεις που χρησιμοποιούν Ajax.

- Οι δυναμικές σελίδες που δημιουργήθηκαν με τη χρήση διαδοχικών αιτήσεων Άγιαξ δεν καταγράφονται αυτόματα στο ιστορικό του browser, κι έτσι κάνοντας κλικ στο πρόγραμμα περιήγησης στο "πίσω" να μην μπορεί να επιστρέψει το χρήστη σε μια προηγούμενη κατάσταση της σελίδας που χρησιμοποιεί Ajax, αλλά μπορεί να τα επιστρέψει, στην τελευταία πλήρη σελίδα που επισκέφτηκε πριν. Μια λύση περιλαμβάνει την χρήση αόρατων IFrames να προκαλέσουν αλλαγές στο ιστορικό του browser και αλλάζοντας το τμήμα του URL (μετά από #) όταν τρέχει η Ajax και η παρακολούθησή της για αλλαγές.
- Η δυναμική ενημέρωση ιστοσελίδας επίσης καθιστά δύσκολο για ένα χρήστη να βάλει σαν σελιδοδείκτη μια ιδιαίτερη κατάσταση της εφαρμογής. Λύσεις για το πρόβλημα αυτό υπάρχουν, πολλές από τις οποίες χρησιμοποιούν το αναγνωριστικό κομμάτι της διεύθυνσης URL (το μέρος του URL μετά το "#") να το παρακολουθούν, και επιτρέπουν στους χρήστες να επιστρέψουν στην εφαρμογή σε μία δεδομένη κατάσταση.
- Επειδή τα περισσότερα προγράμματα ανίχνευσης του ιστού δεν εκτελούν κώδικα JavaScript, οι δημόσιες web εφαρμογές θα πρέπει να παρέχουν ένα εναλλακτικό μέσο πρόσβασης στο περιεχόμενο που θα έπρεπε κανονικά να ανακτηθεί με Άγιαξ, για να επιτρέψει τις μηχανές αναζήτησης να τις καταχωρήσουν στο ευρετήριο.
- Κάθε χρήστης του οποίου το πρόγραμμα περιήγησης δεν υποστηρίζει Ajax ή JavaScript, ή απλώς έχει το JavaScript απενεργοποιημένο, δεν θα είναι σε θέση να χρησιμοποιεί τη λειτουργικότητά του. Ομοίως, συσκευές, όπως κινητά τηλέφωνα, PDAs, και οι αναγνώστες οθόνης μπορεί να μην έχουν υποστήριξη για JavaScript ή το αντικείμενο XMLHttpRequest. Επίσης, οι αναγνώστες οθόνης που είναι σε θέση να χρησιμοποιούν Ajax δεν μπορούν ακόμη να είναι σε θέση να διαβάσουν σωστά το περιεχόμενο που δημιουργείται δυναμικά. Ο μόνος τρόπος για να αφήσει ο χρήστης την πραγματοποίηση της λειτουργίας είναι να καταφύγει σε μη JavaScript μεθόδους. Αυτό μπορεί να επιτευχθεί με τη διασφάλιση συνδέσεων και εντύπων μπορούν να χρησιμοποιηθούν σωστά και δεν βασίζονται αποκλειστικά σε Ajax. Στην JavaScript, η υποβολή μορφής θα μπορούσε τότε να σταματήσει με το "return false".
- Η ίδια πολιτική καταγωγή εμποδίζει κάποιες τεχνικές της Ajax από το να χρησιμοποιηθούν μεταξύ τομέων (domains), αν και το W3C έχει ένα σχέδιο του αντικειμένου XMLHttpRequest που θα επιτρέψει αυτή τη λειτουργία.
- Η Ajax ανοίγει έναν άλλο φορέα της επίθεσης για κακόβουλο κώδικα που προγραμματιστές μπορεί να μην έχουν δοκιμάσει πλήρως.

4.3 Μελλοντική εργασία και επεκτάσεις

Στην παρούσα εργασία μελετήσαμε και αναλύσαμε την τη χρήση της τεχνικής AJAX και σε θεωρητικό αλλά και σε πρακτικό επίπεδο. Η ιστοσελίδα που αναπτύξαμε στο πλαίσιο της πτυχιακής εργασίας έκανε χρήση της γλώσσας HTML, της JavaScript και της XML καθώς και της JAVA για την επικοινωνία της ιστοσελίδας με τον διακομιστή. Μια ενδιαφέρουσα πρόταση για μελλοντική εργασία θα ήταν η παρουσίαση και μελέτη της τεχνικής AJAX, χρησιμοποιώντας την γλώσσα PHP για την ανάπτυξη μιας ιστοσελίδας, αναλύοντας ταυτόχρονα όλες τις δυνατότητες και τα πλεονεκτήματα που μπορεί να μας δώσει αυτός ο συνδυασμός.

Βιβλιογραφία

- Steve Holzner, P. (2006). *Ajax for Dummies*. Indianapolis, Indiana: Wiley Publishing, Inc.
- Charles Wyke-Smith (2007). *Scriptin' with JavaScript and Ajax: A Designer's Guide*
- Ullman Chris (2007). *Beginning Ajax*
- [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))
- Tom Negrino and Dori Smith (2007) *JavaScript and Ajax for the Web: Visual QuickStart Guide (7th Edition)*
- Jason Cranford Teague (2006). *CSS, DHTML, and Ajax, Fourth Edition*
- Jerry Lee Ford (2008). *Ajax Programming for the Absolute Beginner*
- <http://www.w3.org/standards/webdesign/htmlcss>
- <http://en.wikipedia.org/wiki/HTML>
- <http://www.w3schools.com/html/default.asp>
- <http://en.wikipedia.org/wiki/JavaScript>
- <http://en.wikipedia.org/wiki/Css>
- http://en.wikipedia.org/wiki/Java_Servlet#Life_cycle_of_a_servlet
- http://en.wikipedia.org/wiki/Java_servlets
- <http://www.w3schools.com/xml/default.asp>
- <http://en.wikipedia.org/wiki/Xml>

- <http://en.wikipedia.org/wiki/XMLHttpRequest>
- <http://tomcat.apache.org/>
- <http://www.adobe.com/products/dreamweaver.html>