



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ**  
**ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων  
Πληροφορικής

**Συστήματα Διαχείρισης Περιεχομένου**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Άγγελος Χ. Μιχάλας

**Επιβλέπων :** Βασίλειος Λούμος  
Καθηγητής Ε.Μ.Π

Αθήνα, Φεβρουάριος 2009





**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ**  
**ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων  
Πληροφορικής

**Συστήματα Διαχείρισης Περιεχομένου**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Άγγελος Χ. Μιχάλας

**Επιβλέπων :** Βασίλειος Λούμος  
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12η Ιανουαρίου 2009.  
Αθήνα, Φεβρουάριος 2009

.....  
Β. Λούμος  
Καθηγητής Ε.Μ.Π

.....  
Ε. Καγιάφας  
Καθηγητής Ε.Μ.Π

.....  
Θ. Βαρβαρίγου  
Καθηγητής Ε.Μ.Π

Αθήνα, Φεβρουάριος 2009

.....  
Άγγελος Χ. Μιχάλας

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Άγγελος Χ. Μιχάλας, 2009

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Περίληψη

Στην παρούσα εργασία περιγράφεται η αρχιτεκτονική ενός συστήματος διαχείρισης περιεχομένου. Τα δομικά μέρη του συστήματος θα περιγραφούν καθώς και οι βασικές λειτουργικές οντότητες που αποτελούν το συνολικό σύστημα. Η ανάλυση περιέχει τόσο την περιγραφή της αρχιτεκτονικής του συστήματος, των βασικών αντικειμένων που συμμετέχουν στην λειτουργία του συστήματος καθώς και τον τρόπο που συνεργάζονται για την λειτουργία του συστήματος.

Επιπλέον περιγράφεται η αρχιτεκτονική των workflows καθώς και η αρχιτεκτονική κρυφής μνήμης του υποσυστήματος και ο τρόπος λειτουργίας της. Η ενσωμάτωση Υπηρεσιών Ιστού (Web Services) αναλύεται επίσης και δείχνει ότι η χρησιμότητά τους αποτελεί ένα σημαντικό κομμάτι, χρήσιμο στην διανομή του περιεχομένου.

Στο τελευταίο μέρος της εργασίας προτείνονται και υλοποιούνται επεκτάσεις του συστήματος για την ενσωμάτωση σε αυτό λειτουργιών όπως η διαχείριση δομημένου περιεχομένου, καθώς και η ανάπτυξη οδηγών (templates) για εύκολη δημιουργία δικτυακών τόπων.

## Λέξεις Κλειδιά

Σύστημα Διαχείρισης Περιεχομένου, Ανάλυση Συστημάτων, Υπηρεσίες Ιστού, Διαδίκτυο, Βάσεις Δεδομένων, Workflows, Κρυφή Μνήμη, Αντικειμενοστραφής Προγραμματισμός.

## **Abstract**

The present work describes the architecture of a content management system. The structural parts of the system are presented as well as the basic functional entities that constitute the total system. The analysis contains the description the system architecture, the basic objects that participate in the operation of the system as well as the way they interact during system operation.

Moreover the architecture of workflows as well as the cache memory sub system and its operation is described. The incorporation of Web Services is analyzed and its usefulness is presented for the distribution of content.

The last part of the work proposes and implements extensions of the system for the incorporation of operations such as the management of structured content, as well as the creation of templates which help the development of web sites.

## **Keywords**

Content Management System, System Analysis, Web Services, Internet, Data Bases, Workflows, Cache Memory, Object-Oriented Programming.

# ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή .....	9
2. Βασικά δομικά μέρη .....	12
2.1. Αρχιτεκτονική Συστήματος .....	13
2.2. Πρόσβαση στους πόρους του συστήματος .....	15
2.3. Πόροι του συστήματος.....	17
2.4. Το εικονικό σύστημα αρχείων (Virtual File System – VFS) του OpenCMS 18	
2.5. Δομή των κλάσεων του OpenCMS.....	20
3. Τεχνολογίες υποστήριξης του CMS .....	22
3.1. Γενικές ενέργειες συστήματος .....	22
3.1.1. Manage Users, Groups, Access Control Rights - ACRs.....	23
3.1.2. Επικύρωσης - Authentication .....	26
3.1.3. Manage Delivery Channels.....	26
3.2. Workflows.....	27
3.3. Ο μηχανισμός FlexCache.....	34
3.3.1. Επιλογές διαμόρφωσης της FlexCache.....	35
3.3.2. Διαχείριση της FlexCache.....	35
3.3.3. Ορισμός οδηγιών cache (cache directives) σε κάποιο στοιχείο.....	38
3.3.4. Οδηγίες FlexCache .....	39
3.3.5. Ανάλυση του κώδικα της FlexCache .....	40
4. Ορισμός νέων καναλιών διανομής περιεχομένου σε ένα Σύστημα Διαχείρισης Περιεχομένου (CMS).....	46
4.1. Γλώσσα XML .....	46
4.2. Document Object Model (DOM).....	47
4.3. Java Document Object Model (JDOM).....	48
4.4. Simple API for XML (SAX).....	49
4.5. Υπηρεσίες Ιστού (Web Services).....	51
4.5.1. SOAP (Simple Object Access Protocol).....	51
4.5.2. WSDL (Web Services Description Language) .....	52
4.5.3. UDDI (Universal Description, Discovery and Integration).....	53
4.6. Web Services στα πλαίσια του OpenCMS .....	54

4.7.	Υποστήριξη των Web Services.....	55
5.	Επεκτάσεις του OpenCMS .....	58
5.1.	Εισαγωγή νέου περιεχομένου με την μορφή των modules.....	58
5.1.1.	Ορισμός νέου τύπου περιεχομένου.....	58
5.1.2.	Υποστήριξη νέου τύπου περιεχομένου στην διαχείριση .....	61
5.2.	Επέκταση της πλατφόρμας OpenCMS για την αποθήκευση και διαχείριση δομημένου περιεχόμενου .....	61
5.3.	Δυναμικό φόρτωμα δομημένου περιεχόμενου και προβολή του με την χρήση XSL.....	67
5.4.	Χρήση οδηγών (templates) εμφάνισης στο επίπεδο διαχείρισης περιεχομένου.....	68
5.5.	Σύστημα διαχείρισης αρχείων PDF .....	68
5.6.	Προσθήκη δυνατότητας εισαγωγής σχολίων σε ένα περιεχόμενο .....	69
5.7.	Πρότυπα (Templates).....	71
6.	Αναφορές .....	73
7.	Παράρτημα Α – Κώδικας .....	74



# 1. Εισαγωγή

Ένα Σύστημα Διαχείρισης Περιεχομένου [1] (Content Management System – CMS) χρησιμοποιείται προκειμένου να οργανώνει να διαχειρίζεται και να διανέμει πληροφορία. Η πληροφορία αυτή είναι αποθηκευμένη σε δομημένη μορφή και την ονομάζουμε περιεχόμενο.

Ένα CMS σύστημα επιτρέπει στο περιεχόμενο να αποθηκευτεί, ανακτηθεί, να επεξεργασθεί, να ενημερωθεί, να ελεγχθεί, και να προκύψει μία έξοδος με πλήθος τρόπων και μέσων. Τέτοιου είδους λύσεις εμπλέκουν και την ενσωμάτωση βάσεων δεδομένων, workflows, και εργαλείων επεξεργασίας κειμένων. Πολλά από τα προϊόντα τείνουν να εφαρμοστούν σε ειδικές εκδόσεις και όχι σαν γενικές, ευρείας εφαρμογής, λύσεις.

Τα σημαντικότερα μέρη ενός συστήματος CMS είναι οι αποθηκευτικοί χώροι των δεδομένων (data repository), η διεπαφή με το χρήστη, ένα σχήμα επίλυσης γεγονότων (workflows), εργαλεία επεξεργασίας κειμένου, και διάφοροι συντελεστές εξόδου. Τα παραπάνω επιτρέπουν στους συγγραφείς ή χρήστες να εργάζονται με ένα τρόπο και σε ένα χώρο (ή σε πολλούς), δημιουργώντας ή ανανεώνοντας το περιεχόμενο. Οι επιμελητές της εκάστοτε έκδοσης μπορούν να χρησιμοποιούν διαφορετικά εργαλεία προκειμένου να αλληλεπιδρούν με ό,τι οι συγγραφείς παραδίδουν και να παρατηρούν τις ενέργειες των συγγραφέων.

Τα συστατικά στοιχεία ενός συστήματος διαχείρισης περιεχομένου αναλύονται παρακάτω:

**Αποθηκευτικός χώρος δεδομένων:** Τέτοιες είναι οι βάσεις δεδομένων, και ουσιαστικά αποτελούν το χώρο όπου επιτρέπεται η πρόσβαση η ενημέρωση και επανακατανομή. Ο αποθηκευτικός χώρος μπορεί να είναι μία από τις σχεσιακές βάσεις δεδομένων μεταβλητού μεγέθους και ισχύος,. Η μορφοποίηση της πληροφορίας μπορεί να είναι σε SGML (Standard Generalized Markup Language) αλλά ακόμα και απλό ASCII κείμενο. Μπορεί να είναι προσβάσιμο από ένα τοπικό δίκτυο, ένα intranet, ή το Internet, ενώ η ασφάλεια για τον έλεγχο εξουσιοδοτημένης πρόσβασης πρέπει να λαμβάνεται υπόψη.

**Διεπαφή Χρήστη:** Εφόσον ένα CMS σύστημα, τυπικά, αποτελεί ενσωμάτωση πολλών λειτουργιών, είναι λογικό να χρησιμοποιεί πολλές αλλά σχετικές μεταξύ τους

διεπαφές, κυρίως Internet browsers, και επεξεργαστές κειμένων. Τα παραπάνω χρησιμοποιούνται σε συνδυασμό με ορισμένες διεπαφές σχεδιασμένες να ταιριάζουν σε συγκεκριμένες ανάγκες διεργασιών ενός οργανισμού.

**Εργαλεία επεξεργασίας κειμένου:** Οι επεξεργαστές κειμένου και οι SGML editors είναι εργαλεία που διαθέτουν τα περισσότερα CMS συστήματα. Προσφέρουν τη δυνατότητα για δημιουργία και επεξεργασία περιεχομένου, καθώς και φόρμες αρχείων που διευκολύνουν την συνεχή επεξεργασία των περιεχομένων. Τα εργαλεία επεξεργασίας κειμένου επιτρέπουν τους συγγραφείς να εργάζονται σε ένα φιλικό περιβάλλον λογισμικού, όπως οι επεξεργαστές κειμένου και οι text editors, ενώ ακολουθούν συγκεκριμένους βασικούς κανόνες.

**Σχήμα ροής εργασίας (Workflow scheme):** ένα CMS σύστημα δίνει την δυνατότητα να γνωρίζουμε πάντα τι συμβαίνει σε κάθε δεδομένη συνιστώσα του περιεχομένου. Το workflow σχήμα μπορεί να παρακολουθεί κάθε στοιχείο δεδομένου, τον έλεγχο του (check in, check out), καθώς και την εξέλιξη των εκδόσεων (version history) που υπάρχουν. Αυτό επιτρέπει σε αυτούς που ελέγχουν την κατάσταση ενός άρθρου, να γνωρίζουν αν το άρθρο έχει προσπελαστεί, βρίσκεται σε κατάσταση επεξεργασίας, έχει παραδοθεί για δημιουργία αντιγράφου έκδοσης, έχει επιστρέψει στο συγγραφέα το δικαίωμα για περαιτέρω εργασία, ή έχει γίνει αποδεκτό στην τελευταία του μορφή. Επιπροσθέτως, μέσω της δυνατότητας παρατήρησης παρελθοντικών εκδόσεων (version history), ο χρήστης ή ο διαχειριστής, παρακολουθεί ποια έκδοση του άρθρου έχει εμφανιστεί και σε ποια δημοσίευση. Διαφορετικές εκδόσεις παραμένουν προσβάσιμες για όσο καιρό αποφασίζεται. Το workflow σχήμα δημιουργεί και custom αναφορές προκειμένου να προσφέρει στο χρήστη την εκάστοτε κατάσταση της πληροφορίας σε μία ποικιλία μορφοποιήσεων.

**Συντελεστές εξόδου (output utilities):** είναι φίλτρα που λαμβάνουν την πληροφορία από το χώρο αποθήκευσης δεδομένων την μορφοποιούν για τα διάφορα μέσα έκδοσης.

Στην παρούσα εργασία περιγράφεται η αρχιτεκτονική ενός συστήματος διαχείρισης περιεχομένου. Τα δομικά μέρη του συστήματος θα περιγραφούν καθώς και οι βασικές λειτουργικές οντότητες που αποτελούν το συνολικό σύστημα. Η ανάλυση περιέχει τόσο την περιγραφή της αρχιτεκτονικής του συστήματος, των βασικών αντικειμένων που συμμετέχουν στην λειτουργία του συστήματος καθώς και τον τρόπο που συνεργάζονται για την λειτουργία του συστήματος.

Επιπλέον περιγράφεται η αρχιτεκτονική των workflows καθώς και η αρχιτεκτονική κρυφής μνήμης του υποσυστήματος και ο τρόπος λειτουργίας της. Η ενσωμάτωση Υπηρεσιών Ιστού (Web Services) αναλύεται επίσης και δείχνει ότι η χρησιμότητά τους αποτελεί ένα σημαντικό κομμάτι, χρήσιμο στην διανομή του περιεχομένου.

Στο τελευταίο μέρος της εργασίας προτείνονται και υλοποιούνται επεκτάσεις του συστήματος για την ενσωμάτωση σε αυτό λειτουργιών όπως η διαχείριση δομημένου περιεχομένου, καθώς και η ανάπτυξη οδηγών (templates) για εύκολη δημιουργία δικτυακών τόπων.

## 2. Βασικά δομικά μέρη

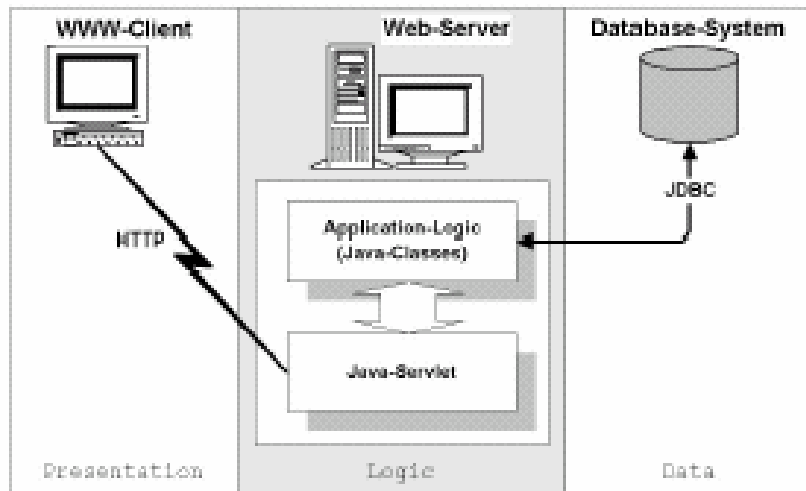
Το σύστημα διαχείρισης περιεχομένου [1] που θα χρησιμοποιήσουμε και θα αναλύσουμε, υπάρχει ως ανοικτός κώδικας στο διαδίκτυο ως OpenCMS [2]. Το OpenCMS είναι ένα open source σύστημα διαχείρισης περιεχομένου. Ο σκοπός του είναι να επιτύχει τη δημιουργία και την διαχείριση πολύπλοκων ιστοσελίδων, χωρίς τη γνώση html. Ένας ενσωματωμένος editor γνωστός ως WYSIWYG με μία διεπαφή χρήστη υποστηρίζουν τους χρήστες να δημιουργήσουν το περιεχόμενο, ενώ μία σύγχρονη template μηχανή, επιβάλλει ένα ευρύ corporate layout. Το OpenCMS βασίζεται στη Java και στην τεχνολογία XML. Για το λόγο αυτό ταιριάζει απόλυτα σε όλες σχεδόν τις υπάρχουσες και μοντέρνες IT κατασκευές. Το OpenCMS μπορεί και λειτουργεί σε ένα περιβάλλον που υποστηρίζει, για παράδειγμα, Linux, Apache, Tomcat, MySQL [3] καθώς και σε εμπορικά συνιστώμενα μέρη όπως IIS, BEA WebLogic, Oracle [4] DB.

Το OpenCMS είναι εξολοκλήρου ανεπτυγμένο σε java. Ο WebServer που χρησιμοποιεί είναι ο Apache. Επίσης η χρήση του Microsoft IIS για συστήματα windows έχει επιτυχώς εξακριβωθεί. Επίσης, όσο αφορά την servlet runtime engine, το Java Servlet API (JSDK) 2.3 και ένα servlet engine είναι απαραίτητα από τον WebServer προκειμένου για την αρχικοποίηση των Java προγραμμάτων σε κλήσεις http. Η χρήση του TOMCAT είναι αυτή που χρησιμοποιείται ως αναφορά για Java Servlets. Όσο αφορά την Java η Java VM έκδοση 1.4 η νεότερη είναι αυτή που χρησιμοποιείται. Όσο αφορά τη χρήση βάσεως δεδομένων, η αρχιτεκτονική του OpenCMS επιτρέπει την χρήση οποιας SQL βάσης δεδομένων που προσφέρει JDBC σύνδεση [5]. Επίσης το OpenCMS χρησιμοποιεί ιδιαίτερα την XML προκειμένου για την αποθήκευση περιεχομένου και δεδομένων για templates. Όσο αφορά την χρήση XML Parsers, το OpenCMS κάνει χρήση του Xerces που είναι ανεπτυγμένο από την εταιρεία Apache.

Αρκετές αλλαγές και παρεμβάσεις έχουν γίνει σε σχεδιαστικό επίπεδο και σε επίπεδο υλοποίησης του OpenCMS, προκειμένου να το προσαρμόσουμε στις τρέχουσες ανάγκες.

## 2.1. Αρχιτεκτονική Συστήματος

Το OpenCMS είναι ένα σύστημα πελάτη-εξυπηρετητή που μπορεί να χρησιμοποιηθεί σε HTTP περιβάλλοντα όπως το Διαδίκτυο. Αποτελεί μια κλασική εφαρμογή Web αρχιτεκτονικής τριών επιπέδων (Σχήμα 2-1).



Σχήμα 2-1: Η 3-επιπέδων αρχιτεκτονική του OpenCMS συστήματος

Το στρώμα παρουσίασης αποτελείται από έναν φυλομετρητή Web (Internet Explorer ή Mozilla Firefox κλπ) που χρησιμοποιείται στην εμφάνιση και πλοήγηση των δυναμικών HTML σελίδων της εφαρμογής (πχ JavaScript[6] κώδικας).

Το λογικό επίπεδο περιλαμβάνει τον web server, ο οποίος επεκτείνεται από ένα περιβάλλον εκτέλεσης για servlets της Java [7] [8]. Ο web server μπορεί να είναι ο Apache Web Server [9] (έκδοσης 1.3.6 ή ψηλότερης), διαμορφωμένος κατάλληλα έτσι ώστε να μπορεί να εκτελεί τα servlets της εφαρμογής. Όλες οι Java κλάσεις εγκαθίστανται στο servlet περιβάλλον του server. Το OpenCMS servlet (που είναι μια java κλάση) παρέχει τη διεπαφή στο στρώμα παρουσίασης καθώς επίσης και τη διεπαφή στο στρώμα της βάσης δεδομένων. Χρησιμοποιεί το πρωτόκολλο HTTP για την επικοινωνία μεταξύ του χρήστη και της εφαρμογής που εκτελείται στον server (σε αυτήν την περίπτωση το OpenCMS). Η βάση δεδομένων προσεγγίζεται μέσω της διεπαφής JDBC. Η βάση δεδομένων περιέχει τους πίνακες των πόρων, των χρηστών και των χαρακτηριστικά των δεδομένων.

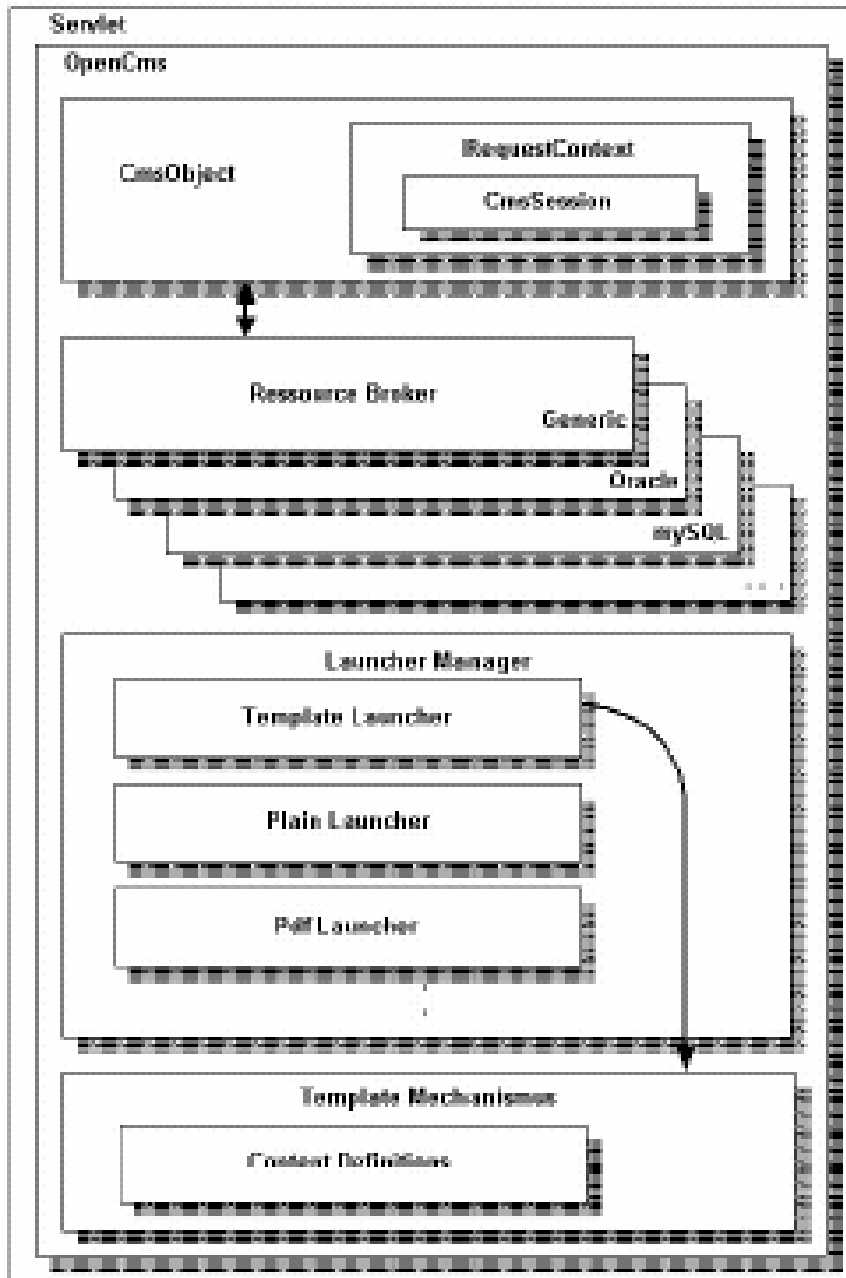
Στο Σχήμα 2-2 παρουσιάζονται τα πιο σημαντικά υποσυστήματα του OpenCMS.

**CmsObject:** Η πρόσβαση στους πόρους του συστήματος γίνεται μέσω του CmsObject. Αποτελεί την διεπαφή του OpenCMS συστήματος για τους υπεύθυνους ανάπτυξης των Modules. Το CmsObject αρχικοποιείται με τα στοιχεία του χρήστη.

**Resource Broker:** Ο resource broker ελέγχει τα αιτήματα για ανάκτηση των πόρων (αρχείων) του συστήματος και τα εκτελεί εφόσον ο χρήστης έχει τις κατάλληλες άδειες πρόσβασης. Η διαμόρφωσή του εξαρτάται από τη βάση δεδομένων που χρησιμοποιείται.

**Launcher Manager:** Ο Launcher Manager καθορίζει τον launcher που θα παραγάγει το ζητούμενο περιεχόμενο βάσει του τύπου ή/και του περιεχόμενου του ζητούμενου πόρου. Εάν επιλεγεί ο template launcher, τότε εκτελείται ο μηχανισμός Template.

**Template Mechanism:** Ο μηχανισμός Template δημιουργεί HTML σελίδες με βάση τη δομή του πόρου. Τα Content Definitions αντικείμενα επιτρέπουν την πρόσβαση σε περιεχόμενο που προέρχονται από διαφορετικές πηγές (π.χ. το εικονικό σύστημα αρχείων (VFS), τη βάση δεδομένων, το σύστημα αρχείων του λειτουργικού συστήματος).

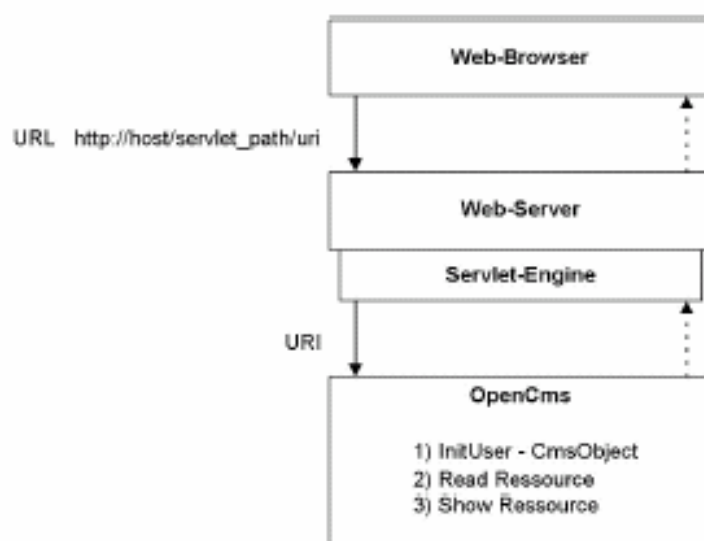


Σχήμα 2-2: Υποσυστήματα του OpenCMS

## 2.2. Πρόσβαση στους πόρους του συστήματος

Η γενική διαδικασία μετά από αίτημα ενός χρήστη στο OpenCMS για την πρόσβαση σε μια ιστοσελίδα παρουσιάζεται στο Σχήμα 2-3. Αρχικά, το αίτημα στέλνεται από τον χρήστη μέσω ενός φυλομετρητή Web. Το URL περιέχει τον WEB server, το path του servlet και το URI της ιστοσελίδας. Ο server αρχίζει τον servlet μηχανισμό, ο οποίος θα περάσει στο OpenCMS το URI.

Η πρόσβαση στους πόρους του συστήματος επιτυγχάνεται μέσω της CmsObject κλάσης, η οποία παρέχει τη public διεπαφή που χρησιμοποιείται για να είναι εφικτή η πρόσβαση στο σύστημα και να δημιουργήσει το αντίστοιχο περιεχόμενο. Από το CmsObject, τα αιτήματα διαβιβάζονται στα μη-public μέρη του OpenCMS. Κατά συνέπεια, αρχικά δημιουργείται ένα αντικείμενο της κλάσης CmsObject και αρχικοποιείται με πληροφορίες που αφορούν τον χρήστη. Το CmsObject περνά το αίτημα των πόρων στο μεσίτη των πόρων (resource broker). Ο μεσίτης των πόρων αποτελεί τη διεπαφή στη βάση δεδομένων. Εκτελεί το αίτημα μέσω της ενότητας πρόσβασης στη βάση δεδομένων (database access module) και επιστρέφει τον πόρο εάν ο χρήστης έχει δικαίωμα πρόσβασης σε αυτόν. Η ενότητα πρόσβασης στη βάση δεδομένων αποτελεί τη διασύνδεση μεταξύ του λογικού επιπέδου (OpenCMS σύστημα) και του επιπέδου της βάσης δεδομένων. Όταν ένας πόρος ζητείται, ο μεσίτης των πόρων διαβιβάζει το αίτημα στην ενότητα πρόσβασης. Η ενότητα πρόσβασης παράγει μια δήλωση SQL για να πάρει τα στοιχεία από τη βάση δεδομένων. Στη συνέχεια παράγει ένα CmsResourceObject με τα στοιχεία που ανακτήθηκαν από τη βάση δεδομένων και το επιστρέφει στο μεσίτη των πόρων. Ο μεσίτης των πόρων μπορεί τώρα να ελέγξει τις άδειες πρόσβασης του αρχείου και να επιστρέψει το αρχείο στο CmsObject εφόσον ο χρήστης έχει τις κατάλληλες άδειες πρόσβασης.

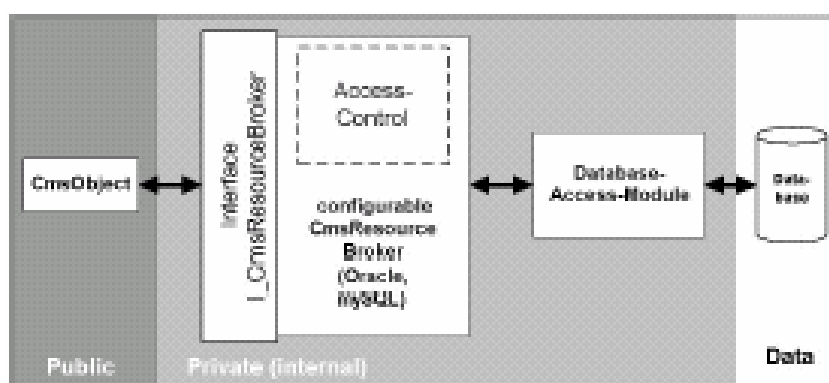


Σχήμα 2-3: Πρόσβαση στους πόρους του συστήματος



Επειδή το OpenCMS μπορεί να λειτουργεί με διαφορετικά είδη βάσεων δεδομένων, ο μεσίτης των πόρων και η ενότητα πρόσβασης διαμορφώνονται σύμφωνα με τη βάση δεδομένων που χρησιμοποιείται (Oracle, MySQL, κ.λπ.). Το Σχήμα 2-4 παρουσιάζει την αλληλεπίδραση μεταξύ των ενοτήτων του OpenCMS συστήματος για την λειτουργία της πρόσβασης σε πόρους που διαθέτει.

Εάν η λειτουργία είναι επιτυχής (δηλ. ο χρήστης έχει την άδεια να διαβάσει το αρχείο), ο πόρος εμφανίζεται (στέλνεται στον web browser του χρήστη).



**Σχήμα 2-4: Πρόσβαση σε πόρους του OpenCMS**

Η αποστολή κάποιου πόρου στον χρήστη γίνεται από τον Launcher Manager. Ο Launcher Manager εκτελεί έναν καθορισμένο launcher σύμφωνα με τον τύπο του ζητούμενου πόρου. Για τα απλά αρχεία κειμένων, εικόνες κ.λπ, χρησιμοποιείται ένας απλός launcher, ο οποίος προωθεί το περιεχόμενο του πόρου άμεσα. Ο template launcher χρησιμοποιείται για να εκτελέσει το μηχανισμό template όταν ζητείται κάποιο template. Ο μηχανισμός Template δημιουργεί σελίδες HTML βασισμένες στο περιεχόμενο με δομημένη μορφή.

### **2.3. Πόροι του συστήματος**

Όλα τα αντικείμενα που διαχειρίζεται το OpenCMS καλούνται πόροι. Προς το παρόν, υπάρχουν οκτώ τύποι πόρων. Με εξαίρεση το φάκελο τύπων των πόρων (resource type folder), όλοι οι άλλοι τύποι είναι αρχεία που το περιεχόμενό τους μπορεί να επιδειχθεί μέσω κάποιου web browser. Ο τύπος του πόρου χρησιμοποιείται για να επιλεγθεί ο κατάλληλος launcher - που αρχίζει τη διαδικασία επεξεργασίας και επιδεικνύει στη συνέχεια τα δεδομένα του πόρου. Ο ακόλουθος πίνακας (Πίνακας 1) παρουσιάζει τους τύπους των πόρων και τον launcher που αντιστοιχεί στον καθένα.

**Πίνακας 2.1:** Πόροι του OpenCMS

Τύπος	ID	Περιγραφή	Launcher
folder	0	Φάκελος	None
plain	1	Αρχεία που δεν επεξεργάζεται το OpenCMS (text files, JavaScript libraries etc.)	Dump launcher
XMLTemplate	2	XML κωδικοποιημένο αρχείο template.	XML launcher
page	3	Αναπαριστά κάποιο site (“clickable file” or “page file”).	XML launcher
binary	4	Διαδικά αρχεία (πχ zip αρχεία και PDF αρχεία, αλλά όχι εικόνες).	Dump launcher
image	5	Διαδικά αρχεία εικόνων (πχ GIF, JPEG)	Dump launcher
script	6	JavaScript στον server	Dump launcher
newspage	7	Αναπαράσταση άρθρου τύπου “news”, ειδική περίπτωση του “page file”	XML launcher
JSP	8	Java Server Page σελίδα	JSP loader

#### **2.4. Το εικονικό σύστημα αρχείων (Virtual File System – VFS) του OpenCMS**

Όλοι οι αρχεία και φάκελοι των διαφορετικών έργων που εμφανίζονται στον explorer view του OpenCMS, δεν αποθηκεύονται στο σύστημα αρχείων του λειτουργικού συστήματος. Το OpenCMS έχει ένα εικονικό σύστημα αρχείων που κρατά τα αρχεία και τους φακέλους σε μια βάση δεδομένων και επιπλέον ενσωματώνει σε αυτά άδειες πρόσβασης. Οι άδειες πρόσβασης στο OpenCMS είναι παρόμοιες με εκείνες του συστήματος αρχείων του Unix. Εντούτοις, με τη χρησιμοποίηση ενός VFS και μιας βάσης δεδομένων, το OpenCMS είναι ανεξάρτητο από το λειτουργικό σύστημα που έχει εγκατασταθεί στον server υπολογιστή. Κατ' αυτό τον τρόπο OpenCMS μπορεί να τρέξει σε λειτουργικά συστήματα, που δεν

παρέχουν άδειες πρόσβασης στο σύστημα των αρχείων τους. Μια σύντομη επισκόπηση των φακέλων και των αρχείων στο VFS του OpenCMS δίνεται στον πίνακα 2.

**Πίνακας 2.2:** VFS του OpenCMS

Φάκελος	Αρχεία του φακέλου
/system	Αρχικός φάκελος που περιέχει τους φακέλους των workplace templates και τα αρχεία διαμόρφωσης του συστήματος.
/system/bodies	Τα body templates αποθηκεύονται σε αυτόν τον φάκελο ή σε υποφακέλους του. Αποθηκεύονται σε κάποιον φάκελο που έχει το ίδιο όνομα με το όνομα του φακέλου που περιέχει το web site.
/system/modules	Τα modules του OpenCMS τοποθετούνται αυτόματα σε αυτόν τον φάκελο.
/system/modules/default	Τοποθετείται το default module που περιέχει ένα βασικό άδειο template και κάποιους πόρους.
/system/modules/default/templates	Τα default master templates που ορίζουν τη βασική δομή των web sites.
/system/galleries	Όλα τα galleries που διαθέτει ο editor.
/system/galleries/download	Gallery αρχείων που μπορούν να γίνουν download.
/system/galleries/pics	Gallery εικόνων που χρησιμοποιούνται σε web sites.
/system/login	Περιέχει την login εικόνα του παραθύρου εργασίας του OpenCMS
/system/workplace	Τα αρχεία εφαρμογών του παραθύρου εργασίας του OpenCMS.
/system/workplace/action	Εκτελέσιμα αρχεία όπως login, copy κλπ.
/system/workplace/templates	Τα templates για την δημιουργία του παραθύρου εργασίας του OpenCMS.
/system/workplace/scripts	Javascript αρχεία του παραθύρου εργασίας του OpenCMS.

/system/workplace/administration	Υποφάκελοι για κάθε στοιχείο του Administration view.
/system/workplace/locales	Τοπικές ρυθμίσεις του παραθύρου εργασίας, πχ. αρχεία γλώσσας. Αποθηκεύονται σε ξεχωριστό φάκελο για κάθε γλώσσα.
/system/workplace/resources	Στατικοί πόροι του παραθύρου εργασίας. Πχ style sheets, εικόνες κουμπιών κλπ.
/system/workplace/help	Το σύστημα βοήθειας.

## 2.5. Δομή των κλάσεων του OpenCMS

Η λειτουργία του OpenCMS υλοποιείται σε πολλές διαφορετικές κλάσεις της Java. Οι κλάσεις κατηγοριοποιούνται σε packages ανάλογα με τις λειτουργίες που επιτελούν. Αυτήν την περίοδο, υπάρχουν 10 packages [2] που καλύπτουν ένα ορισμένο μέρος της λειτουργίας του συστήματος (Πίνακας 3).

**Πίνακας 2.3:** Δομή των κλάσεων

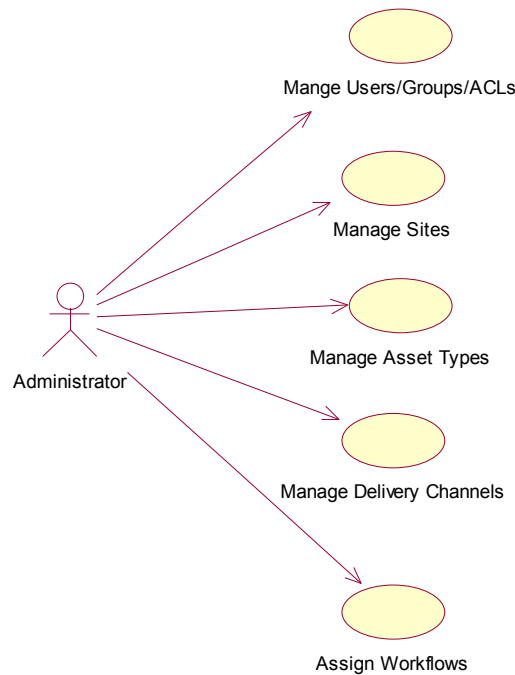
com.OpenCMS.core	Κλάσεις που αποτελούν τον πυρήνα του συστήματος. Περιλαμβάνει επίσης το servlet του OpenCMS.
com.OpenCMS.file	Ορισμοί όλων των αντικειμένων.
com.OpenCMS.file.	Μεσίτης πόρων για generic SQL βάσεις δεδομένων.
com.OpenCMS.file.mySql	Ειδικός μεσίτης πόρων για την mySQL.
com.OpenCMS.file.oraclep lsql	Ειδικός μεσίτης πόρων για την ORACLE.
com.OpenCMS.file.utils	File package.
com.OpenCMS.launcher	Όλοι οι launchers που χρησιμοποιούνται για εμφάνιση αρχείων ή templates και ο Launcher Manager.
com.OpenCMS.template	Όλες οι κλάσεις των template για

	την δημιουργία αρχείων δυναμικά και ο Template Manager.
com.OpenCMS.util	Γενικές λειτουργίες (encoder, decoder etc.)
com.OpenCMS.workplace	Όλα τα στοιχεία του GUI του παραθύρου εργασίας (e.g. buttons, context sensitive project).

### 3. Τεχνολογίες υποστήριξης του CMS

#### 3.1. Γενικές ενέργειες συστήματος

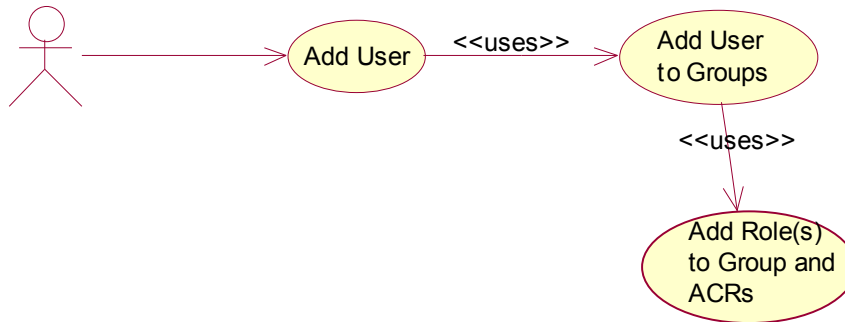
Ο διαχειριστής είναι σε θέση να διαχειρίζεται χρήστες και ομάδες χρηστών, να καθορίζει τα δικαιώματά τους από την άποψη των αδειών (permissions), να διαχειρίζεται σύνολα ιστοσελίδων, τύπους asset, κανάλια παράδοσης (delivery channels) καθώς και να καθορίζει workflows. Αυτές οι περιπτώσεις χρήσης αποτελούν γενικές υπηρεσίες τις οποίες ο διαχειριστής είναι σε θέση να προσδιορίζει και να διαχειρίζεται μέσω του CMS συστήματος. Στις υποπαραγράφους που ακολουθούν αναλύονται επιλεγμένες περιπτώσεις χρήσης



Σχήμα 3-1: Διάγραμμα χρήσης διαχειριστή

### 3.1.1. Manage Users, Groups, Access Control Rights - ACRs

#### Προσθήκη νέου χρήστη



Σχήμα 3-2: Προσθήκη χρήστη – use case

**Ο δράστης - actor** είναι ο διαχειριστής.

**“Add User”**: Είναι μια περίπτωση χρήσης που περιγράφει μια συγκεκριμένη λειτουργία που προσφέρεται από το σύστημα στο διαχειριστή. Ο διαχειριστής είναι σε θέση να προσθέσει έναν νέο χρήστη στο σύστημα. Μπορεί να παρέχει ένα όνομα και έναν κωδικό πρόσβασης για το νέο χρήστη.

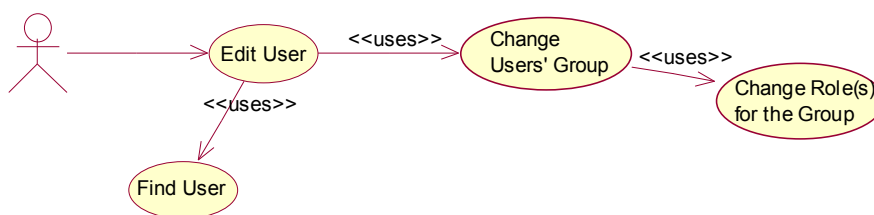
**“Add User to Groups”**: Είναι μια περίπτωση χρήσης που περιγράφει τη συγκεκριμένη λειτουργία που προσφέρεται από το σύστημα στο διαχειριστή μέσω της use case "Add User to Groups". Ο διαχειριστής είναι σε θέση να προσθέσει έναν νέο χρήστη σε μια ομάδα. Η ομάδα μπορεί να επιλεγεί από έναν προκαθορισμένο κατάλογο.

**“Add Roles to Groups and ACRs”**: Είναι μια περίπτωση χρήσης που περιγράφει τη συγκεκριμένη λειτουργία που προσφέρεται από το σύστημα στο διαχειριστή μέσω της use case "Add Roles to Groups and ACRs ". Ως εκ τούτου ο διαχειριστής είναι σε θέση να προσθέσει κάποιον(ους) ρόλο(ους) σε μια ομάδα και να προσθέσει άδειες σε αυτούς τους ρόλους.

**Σημείωση:** Η "Add Roles to Groups and ACRs" περίπτωση χρήσης μπορεί να είναι προαιρετική σε αυτήν την περίπτωση χρήσης. Μπορεί να θεωρηθεί ως

αυτόνομη περίπτωση χρήσης που συνδέεται άμεσα με έναν δράστη. Οι περιπτώσεις χρήσης που ακολουθούν υιοθετούν αυτή την αρχιτεκτονική. Περιγράφουμε περιπτώσεις όπου αρχικά καθορίζουμε ρόλους και ομάδες για τους χρήστες, ενώ ο καθορισμός των αδειών για κάθε ρόλο παρουσιάζεται σε διαφορετική use case. (βλέπε παράγραφος "ορισμός αδειών σε ρόλους")

### Ενημέρωση υπάρχοντος χρήστη



Σχήμα 3-3: Ενημέρωση χρήστη – use case

**Ο δράστης - actor** είναι ο διαχειριστής.

**“Edit User”**: Είναι μια περίπτωση χρήσης που περιγράφει τη συγκεκριμένη λειτουργία που προσφέρεται από το σύστημα στο διαχειριστή. Ο διαχειριστής είναι σε θέση να ενημερώσει έναν υπάρχοντα χρήστη μέσω του συστήματος.

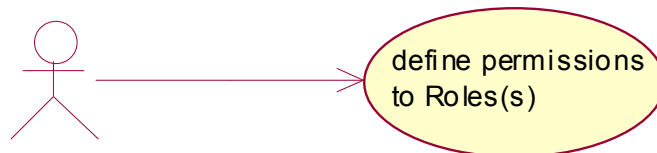
**“Find User”**: Είναι μια περίπτωση χρήσης που περιγράφει τη συγκεκριμένη λειτουργία που προσφέρεται από το σύστημα στο διαχειριστή μέσω της "Edit User" use case. Ο διαχειριστής, για την ενημέρωση ενός υπάρχοντος χρήστη, πρέπει πρώτα να τον προσδιορίσει μέσω της "Find User" use case.

**“Change Users’ Group”**: Ο διαχειριστής, αφού προσδιορίσει έναν υπάρχοντα χρήστη μέσω της "Find User" use case, είναι σε θέση να αλλάξει την ομάδα αυτού του χρήστη μέσω της λειτουργίας που προσφέρεται από την “Change Users’ Group” use case.



**“Change Role(s) for the Group”**: Ο διαχειριστής, είναι σε θέση να αλλάξει το ρόλο(ους) του συγκεκριμένου χρήστη μέσω της λειτουργίας που προσφέρεται από την “Change Role(s) for the Group” use case.

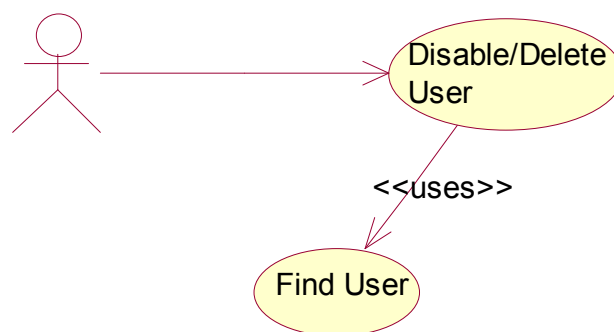
### Ορισμός αδειών σε ρόλους - Define Permissions to Roles



Σχήμα 3-4: Ορισμός αδειών σε ρόλους – use case

**“define permissions to Role(s)”**: Ο διαχειριστής, είναι σε θέση να αλλάξει τις άδειες (ACRs) κάθε συγκεκριμένου ρόλου. Αυτή η περίπτωση χρήσης συνδέεται με τις δύο προηγούμενες περιπτώσεις χρήσης.

### Απενεργοποίηση χρήστη - Disable a User



Σχήμα 3-5: Απενεργοποίηση / διαγραφή χρήστη – use case

**“disable/delete user”**: Ο διοικητής, είναι σε θέση να απενεργοποιήσει έναν υπάρχοντα χρήστη μέσω αυτής της περίπτωσης χρήσης.

**“Find User”**: Είναι η περίπτωση χρήσης που περιγράφει τη λειτουργία που προσφέρεται από το σύστημα στο διαχειριστή μέσω της "disable/delete user" use case. Ο διαχειριστής, για τη διαγραφή ενός υπάρχοντος χρήστη από το σύστημα, πρέπει αρχικά να τον προσδιορίσει μέσω της “Find User” use case.

### 3.1.2. *Επικύρωσης - Authentication*

Η ακόλουθη περίπτωση χρήσης αναφέρεται στη λειτουργία επικύρωσης του συστήματος:



Σχήμα 3-6: Πρόσβαση στο σύστημα – use case

**“Access”**: Αυτή η περίπτωση χρήσης προσφέρει τη λειτουργία επικύρωσης ενός χρήστη. Όταν ο χρήστης χρειαστεί να έχει πρόσβαση στο σύστημα ή οποιοδήποτε προστατευμένο πόρο, θα του ζητηθεί να πληκτρολογήσει έναν κωδικό πρόσβασης για λόγους επικύρωσης.

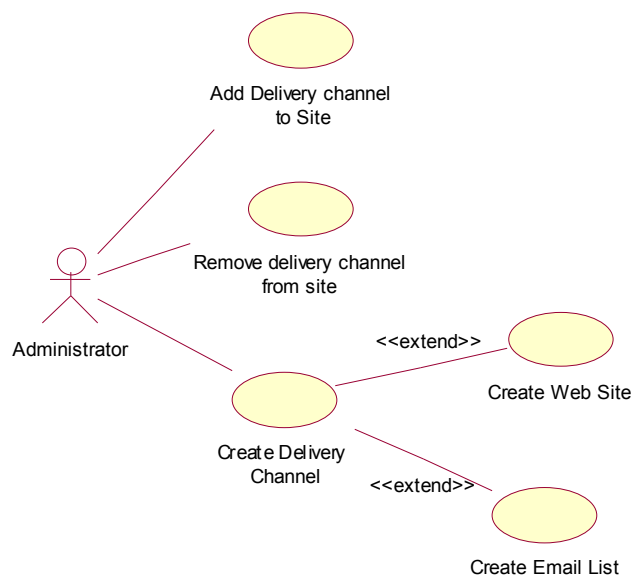
### 3.1.3. *Manage Delivery Channels*

Το σύστημα είναι σε θέση να δημοσιεύσει σε διάφορα κανάλια παράδοσης, τα assets που διαθέτει στην αποθήκη του. Τα κανάλια παράδοσης πρέπει να είναι επεκτάσιμα μέσω διεπαφών που καθορίζουν το σχήμα που απαιτείται για κάθε τύπο δημοσίευσης. Τα πρότυπα (templates) για κάθε asset μπορούν να καθορίσουν τη μορφή (format) με την οποία θα παραδοθεί το asset.

Ο διαχειριστής του συστήματος είναι σε θέση να δημιουργεί κανάλια παράδοσης και να τα προσθέτει σε ομάδες ιστοσελίδων (sites). Σε κάθε asset πρέπει

να παρασχεθεί ένα συγκεκριμένο πρότυπο, για να μπορεί να παραδοθεί από κάποιο κανάλι, σύμφωνα με τις οδηγίες που ορίζονται στο συγκεκριμένο κανάλι.

Ως κανάλια παράδοσης μπορούν να θεωρηθούν, ένας ιστοχώρος, ένας κατάλογος ηλεκτρονικού ταχυδρομείου ή μια υπηρεσία εκτύπωσης.



Σχήμα 3-7: Διαχείριση καναλιών παράδοσης – use case

### 3.2. Workflows

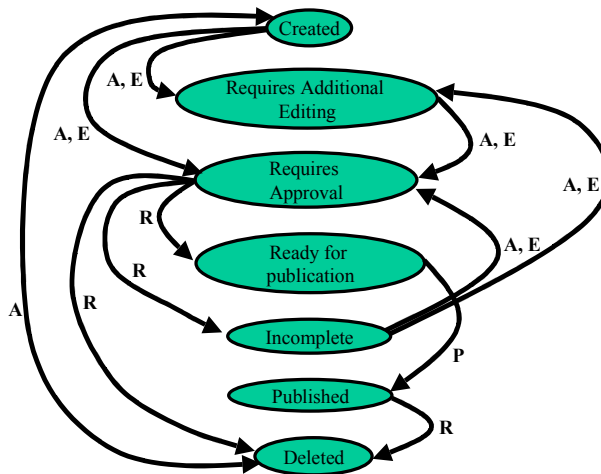
Ένα CMS σύστημα δίνει την δυνατότητα να γνωρίζουμε πάντα τι συμβαίνει σε κάθε δεδομένη συνιστώσα του περιεχομένου. Το workflow σχήμα μπορεί να παρακολουθεί κάθε στοιχείο δεδομένου, τον έλεγχο του (check in, check out), καθώς και την εξέλιξη των εκδόσεων (version history) που υπάρχουν. Αυτό επιτρέπει σε αυτούς που ελέγχουν την κατάσταση ενός άρθρου, να γνωρίζουν αν το άρθρο έχει προσπελαστεί, βρίσκεται σε κατάσταση επεξεργασίας, έχει παραδοθεί για δημιουργία αντιγράφου έκδοσης, έχει επιστρέψει στο συγγραφέα το δικαίωμα για περαιτέρω εργασία, ή έχει γίνει αποδεκτό στην τελευταία του μορφή. Επιπροσθέτως, μέσω της δυνατότητας παρατήρησης παλαιότερων εκδόσεων (version history), ο χρήστης ή ο διαχειριστής, παρακολουθεί ποια έκδοση του άρθρου έχει εμφανιστεί και σε ποια δημοσίευση. Διαφορετικές εκδόσεις παραμένουν προσβάσιμες για όσο καιρό αποφασίζεται. Το workflow σχήμα δημιουργεί και custom αναφορές προκειμένου να

προσφέρει στο χρήστη την εκάστοτε κατάσταση της πληροφορίας σε μία ποικιλία μορφοποιήσεων

Το OpenCMS συντονίζει την έκδοση και τον έλεγχο ομάδων ιστοσελίδων (sites). Το σύστημα διαχειρίζεται workflows, μέσω των οποίων η διαχείριση των sites ενσωματώνει, τον συντονισμό των αλλαγών που πραγματοποιούνται σε αυτά, σύμφωνα με τις πολιτικές που ορίζουν οι διαχειριστές. Έγιναν παρεμβάσεις στον κώδικα του OpenCMS ώστε να υποστηρίζονται τα ακόλουθα:

Ειδικευμένοι χρήστες του CMS μπορούν να δημιουργήσουν, να τροποποιήσουν, να ψάξουν, και να δημοσιεύσουν αντικείμενα στο WEB με έναν τρόπο που ελέγχεται από προκαθορισμένα workflows. Ένα αντικείμενο προς δημοσίευση μπορεί να είναι ένα asset ή ένα site. Τα workflows καθορίζονται από τον διαχειριστή για τα **δημοσιεύσιμα αντικείμενα** και αποτελούνται από:

- Καταστάσεις (states), οι οποίες ορίζουν το στάδιο στο οποίο βρίσκεται το κάθε workflow. Ένα site μπορεί να βρίσκεται σε κάποια από τις παρακάτω καταστάσεις: δημιουργημένο (created), απαιτεί επιπρόσθετη ενημέρωση (requires additional editing), απαιτεί έγκριση (requires approval), έτοιμο για δημοσίευση (ready for publication), ελλιπές (incomplete), δημοσιευμένο (published) και διαγραμμένο (deleted). Πληροφορίες όσον αφορά τους χρήστες καθώς και χρονικές πληροφορίες πρέπει επίσης να συμπεριλαμβάνονται στα workflows οι οποίες θα καθορίζουν από ποιους και πότε αλλάχθηκε η κατάσταση ενός δημοσιεύσιμου αντικειμένου.
- Βήματα (steps), τα οποία είναι προκαθορισμένες μεταβάσεις μεταξύ των states. Το ακόλουθο διάγραμμα επεξηγεί όλες τις δυνατές μεταβάσεις από ένα state σε άλλο. Τα βέλη του σχήματος δείχνουν τις δυνατές μεταβάσεις από των states. Δίπλα σε κάθε βέλος είναι τα αρχικά του ρόλου που πρέπει να έχει ο χρήστης προκειμένου να μπορεί να αλλάξει το state του workflow στο επόμενο.



Σχήμα 3-8: Διάγραμμα μετάβασης των states

- Οι ρόλοι (roles), έχουν οριστεί προηγουμένως στους χρήστες μέσω της λειτουργίας “Add a new user” του διαχειριστή. Ο ρόλος για τον χρήστη ενός workflow, καθορίζει τις άδειες (permissions) που αυτός διαθέτει για κάποιο δημοσιεύσιμο αντικείμενο. Ο ακόλουθος πίνακας επεξηγεί τις άδειες χρηστών σύμφωνα με τον ρόλο που έχουν.

Πίνακας 3-1: Ρόλοι και άδειες

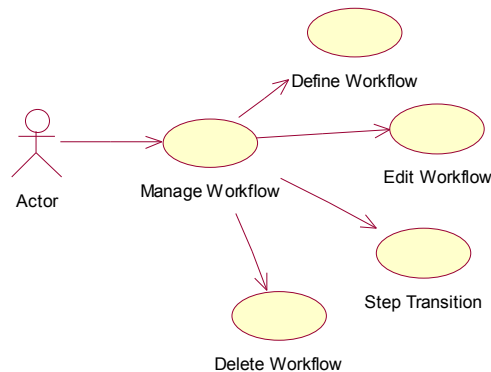
Role	Permissions
Author	Μπορεί να δημιουργήσει ένα αντικείμενο, να το τροποποιήσει, ή να το διαγράψει
Editor	Μπορεί να τροποποιήσει ένα υπάρχον αντικείμενο που δεν είναι σε ένα από τα ακόλουθα state: “published”, “ready for publication”, “requires approval”, “deleted”.
Publishers	Μπορεί να δημοσιεύσει ένα υπάρχον αντικείμενο, το οποίο είναι “ready for publication” ώστε να το καταστήσει προσβάσιμο μέσω του Web.

Reviewer	Μπορεί να αλλάξει την κατάσταση ενός υπάρχοντος αντικειμένου από το state: "requires approval" στα states: "ready for publication", "incomplete", or "deleted". Στην περίπτωση που το αντικείμενο είναι μέρος ενός άλλου δημοσιεύσιμου αντικειμένου ο reviewer του τελευταίου ειδοποιείται.
----------	---

- Μέθοδος ανακοίνωσης: Ο τρόπος με τον οποίο ανακοινώνεται η αλλαγή του state ενός workflow στους χρήστες του μέσω ηλεκτρονικού ταχυδρομείου.

Παρακάτω παρουσιάζονται οι περιπτώσεις χρήσης, που περιγράφουν την χρήση των workflows.

Όσο αφορά την διαχείριση των Workflows ισχύει το παρακάτω μοντέλο:

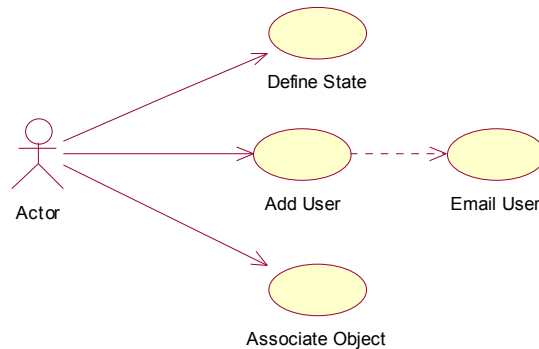


**Σχήμα 3-9: Διαχείριση των workflows**

Αυτή η περίπτωση χρήσης περιγράφει τη λειτουργία που προσφέρεται από το σύστημα για τη διαχείριση ενός workflow. Ο δράστης μπορεί να καθορίσει (define), να ενημερώσει (edit), να αλλάξει το state και να διαγράψει (delete) ένα workflow.

Κάθε μια από αυτές τις λειτουργίες περιγράφεται λεπτομερέστερα στις ακόλουθες περιπτώσεις χρήσης.

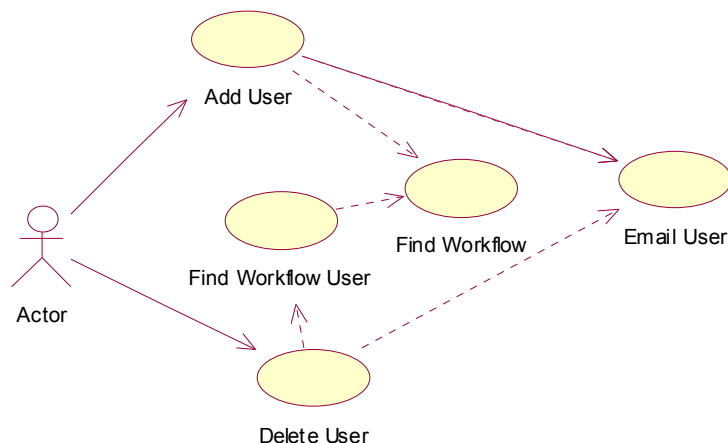
Στην περίπτωση του καθορισμού του workflow έχουμε το εξής παρακάτω μοντέλο λειτουργίας:



**Σχήμα 3-10: Καθορισμός workflow – use case**

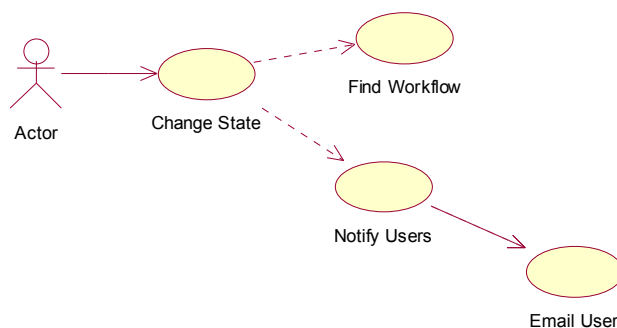
Ο δράστης είναι ο διαχειριστής του CMS συστήματος. Μέσω αυτής της λειτουργίας ο χρήστης μπορεί να καθορίσει ένα νέο workflow, να διευκρινίσει το state του, να ορίσει τους χρήστες του workflow, και να συνδέσει τελικά το νέο workflow με ένα δημοσιεύσιμο αντικείμενο. Ένας χρήστης έχει συνδεθεί με έναν ρόλο μέσω της λειτουργίας “Add a new user” του διαχειριστή. Όταν ένας χρήστης του συστήματος ορίζεται και χρήστης σε ένα workflow, ενημερώνεται μέσω ηλεκτρονικού ταχυδρομείου.

Αντιστοίχως για την ενημέρωση ενός workflow έχουμε το παρακάτω μοντέλο:



**Σχήμα 3-11: Ενημέρωση workflow – use case**

Ο δράστης είναι ο διαχειριστής του CMS. Μέσω αυτής της λειτουργίας ο δράστης μπορεί να προσθέσει / διαγράψει τους χρήστες ενός workflow. Όταν ένας χρήστης προστίθεται ή διαγράφεται ειδοποιείται μέσω ηλεκτρονικού ταχυδρομείου. Περαιτέρω για την μετάβαση των states έχουμε :

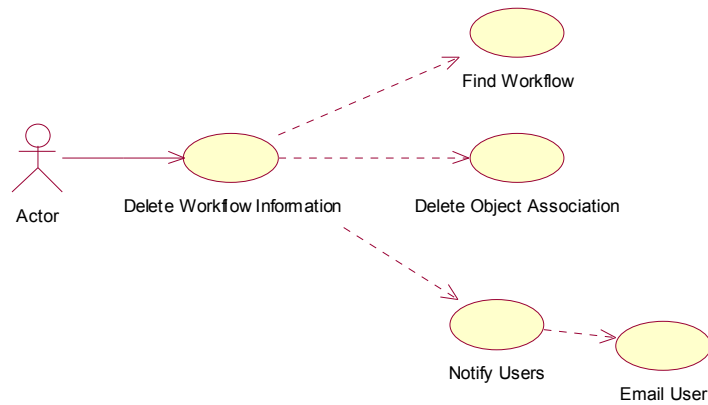


**Σχήμα 3-12: Μετάβαση του workflow από κάποιο state σε άλλο**

Ο δράστης είναι ο διαχειριστής του CMS ή ένας χρήστης του workflow με τις κατάλληλες άδειες (βλ. Πίνακα 3-1). Μέσω αυτής της λειτουργίας ο δράστης μπορεί να αλλάξει το state ενός workflow σε κάποιο άλλο state. Οι δυνατές μεταβάσεις μεταξύ των states καθορίζονται στο Σχήμα 3-8. Όλοι οι χρήστες ενός workflow ειδοποιούνται όταν αλλάζει το state ενός workflow.

Για την διαγραφή ενός workflow οι λειτουργίες του συστήματος ορίζονται ως εξής:





**Σχήμα 3-13: Διαγραφή workflow**

Ο δράστης είναι ο διαχειριστής του CMS ή κάποιος χρήστης που ορίζεται ως reviewer σε αυτό το workflow. Μέσω αυτής της λειτουργίας ο δράστης διαγράφει ένα workflow. Οι χρήστες και όλες οι σχετικές πληροφορίες του workflow διαγράφονται καθώς επίσης και η σύνδεσή του με το δημοσιεύσιμο αντικείμενο. Ένα νέο workflow μπορεί να καθοριστεί μετά από αυτήν την λειτουργία για δημοσιεύσιμο αντικείμενο.

### 3.3. Ο μηχανισμός FlexCache

Η FlexCache είναι μια νέα εφαρμογή cache για τις σελίδες OpenCMS. Προς το παρόν χρησιμοποιείται μόνο για να αποθηκεύσει JSP και στοιχεία που περιλαμβάνονται σε JSP, αλλά είναι πιθανό ότι θα επεκταθεί σε άλλους τύπους στο μέλλον.

Ο μηχανισμός FlexCache είναι σε θέση να αποθηκεύσει μια σελίδα καθώς και όλα τα υποστοιχεία μιας σελίδας. Όλα τα στοιχεία αποθηκεύονται ως χωριστές εγγραφές στην cache. Οι επιγραφές (headers) και επανακατευθύνσεις (redirects) εναποθηκεύονται επίσης. Αυτό σημαίνει ότι στην περίπτωση που σε μια σελίδα A περιλαμβάνεται ένα υποστοιχείο B, τότε και η A σελίδα και το υποστοιχείο B θα αποθηκευθούν ως ξεχωριστά στοιχεία, αντί του ενός συνδυασμένου στοιχείου A+B. Κάθε στοιχείο έχει τις δικές του οδηγίες cache (cache directives) που μπορούν να οριστούν στην cache ιδιότητα του στοιχείου χρησιμοποιώντας το OpenCMS Workplace μενού. Οι οδηγίες cache προσδιορίζουν κάτω από ποιες περιστάσεις ένα στοιχείο θα αποθηκευτεί στη μνήμη cache ή όχι. Παραδείγματος χάριν εάν η σελίδα A έχει οριστεί να αποθηκεύεται στην cache και το υποστοιχείο της B όχι, στην cache αποθηκεύεται μια εγγραφή (entry) για το A και το B εισάγεται δυναμικά στην σελίδα A όταν η τελευταία αναζητηθεί από τον χρήστη. Επιπλέον, οι οδηγίες cache επιτρέπουν "τις παραλλαγές εγγραφών (variation entries)" που σημαίνει ότι μπορούν να υπάρξουν περισσότερες από μια εγγραφές ενός στοιχείου στην cache. Αυτό είναι ιδιαίτερα χρήσιμο σε διάφορες περιπτώσεις.

Ένα απλό παράδειγμα του πώς χρησιμοποιούνται οι οδηγίες cache ακολουθεί: Θεωρούμε ότι υπάρχει ένας ιστοχώρος προσιτός για όλους τους χρήστες με προαιρετική εισαγωγή κωδικού έναρξης συνόδου (login). Σε περίπτωση που ένας χρήστης δεν εισάγει κωδικό έναρξης, μια μικρή περιοχή της κεντρικής σελίδας πρέπει να εμφανίζει μια απλή HTML φόρμα με το λογότυπο "παρακαλώ δώστε κωδικό έναρξης εδώ". Εάν ο χρήστης έχει συνδεθεί επιτυχώς, αυτή η μικρή περιοχή πρέπει να αντικατασταθεί από το μήνυμα "έχετε συνδεθεί ως χρήστης xxxx". Με την FlexCache, μπορεί να τοποθετηθεί ολόκληρη η περιοχή του κωδικού έναρξης ως ένα υποστοιχείο που περιλαμβάνεται στην κεντρική σελίδα. Κατόπιν, θα μπορούσε να οριστεί στις οδηγίες cache της κεντρικής σελίδας η οδηγία always (που δηλώνει ότι αυτή η σελίδα είναι πάντα στην cache) και στις οδηγίες cache του υποστοιχείου το

groups= (Guests), που δηλώνει ότι μη εξουσιοδοτημένες επισκέψεις μπορούν να αποθηκευθούν στην cache. Έτσι εάν η σελίδα ζητηθεί από έναν χρήστη που δεν έχει εισάγει κωδικό έναρξης, και τα δύο στοιχεία θα προέλθουν από την cache, με συνέπεια έναν πολύ γρήγορο χρόνο παράδοσης σελίδων. Εάν ο χρήστης συνδεθεί δίνοντας κωδικό έναρξης, το κύριο μέρος της σελίδας προέρχεται ακόμα από την cache, και μόνο η μικρή HTML φόρμα σύνδεσης κατασκευάζεται δυναμικά κατά τον χρόνο εκτέλεσης λαμβάνοντας το όνομα του χρήστη, γεγονός που οδηγεί σε μια επίσης γρηγορότερη απόδοση από ότι ανακτώντας ολόκληρη τη σελίδα δυναμικά.

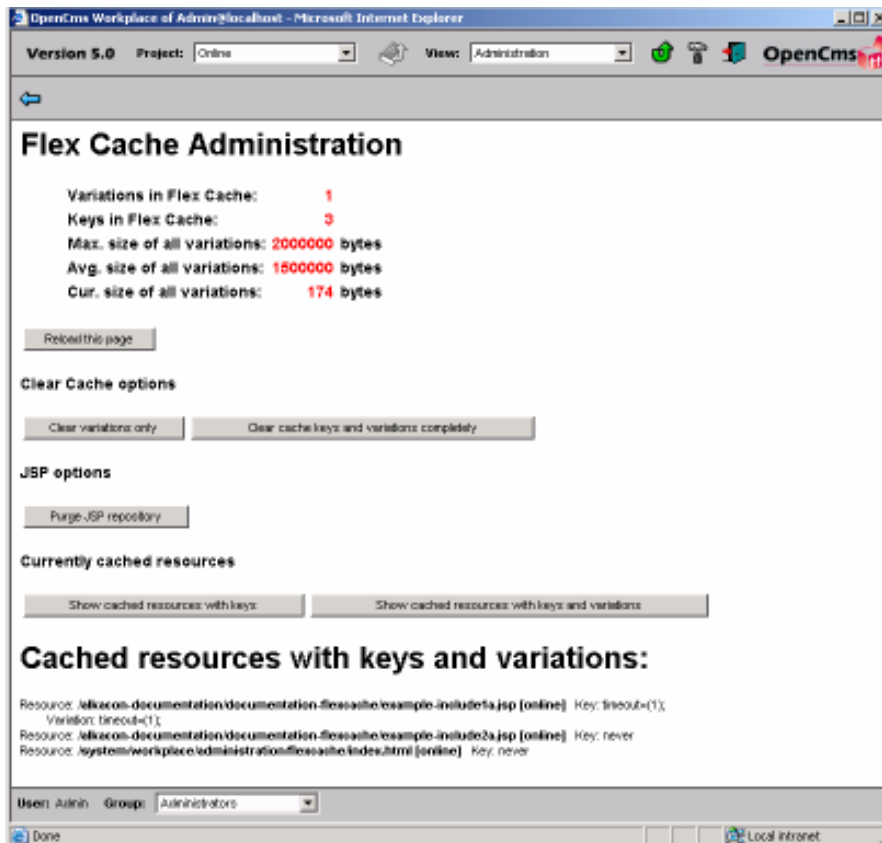
### **3.3.1. Επιλογές διαμόρφωσης της FlexCache**

Ο μηχανισμός FlexCache ενεργοποιείται ή απενεργοποιείται μέσω του αρχείου διαμόρφωσης /WEB-INF/config/OpenCMS.properties του OpenCMS. Υπάρχουν δύο επιλογές για την FlexCache:

- Επιτρέψτε ή θέστε εκτός λειτουργίας την cache καθώς και
- Επιτρέψτε ή θέστε εκτός λειτουργίας την αποθήκευση στην cache πόρων για τα offline έργα (projects) (δηλ. πόροι που δεν έχουν ακόμα δημοσιευτεί).

### **3.3.2. Διαχείριση της FlexCache**

Η διαχείριση του μηχανισμού FlexCache γίνεται μέσα από το περιβάλλον διαχείρισης του OpenCMS. Ο χρήστης θα πρέπει να ανήκει στην ομάδα χρηστών “διαχειριστής” για να διαχειριστεί την FlexCache (σχήμα 3-14).



Σχήμα 3-14: Διαχείριση μηχανισμού FlexCache

Η σελίδα διαχείρισης της FlexCache παρουσιάζει κάποιες πληροφορίες για την τρέχουσα κατάσταση της cache καθώς και διάφορες επιλογές. Στην πάνω περιοχή της σελίδας εμφανίζονται πόσα κλειδιά (keys) και πόσες εγγραφές (entries) είναι αυτήν την περίοδο στην cache.

Τα κλειδιά αντιπροσωπεύουν οδηγίες cache που έχουν αναλυθεί. Για κάθε πόρο URI που ζητείται στο OpenCMS, η οδηγία cache που περιέχει, θα αναλυθεί μόνο μια φορά και θα σωθεί έπειτα στην cache για να αποφευχθεί η περαιτέρω περιττή ανάλυση ή η πρόσβαση στη βάση δεδομένων. Κατά συνέπεια, ο αριθμός κλειδιών στην cache δεν μπορεί να είναι υψηλότερος από τον αριθμό πόρων του VFS του OpenCMS.

Ο αριθμός καταχωρήσεων στην cache παρουσιάζει πόσες παραλλαγές εγγραφών (variation entries), ενός ή περισσότερων πόρων, έχουν παραχθεί αυτήν την περίοδο. Αυτός ο αριθμός εξαρτάται από τις οδηγίες cache που έχουν οριστεί, δεδομένου ότι για κάθε ενιαίο πόρο του OpenCMS μπορεί θεωρητικά να υπάρξει ένας απεριόριστος αριθμός παραλλαγών εγγραφών στην cache.

Στο κάτω μέρος της σελίδας εμφανίζονται κουμπιά που επιτρέπουν στον χρήστη είτε να καθαρίσουν την cache είτε για να δει το περιεχόμενο της cache. Τα αποτελέσματα αυτών των κουμπιών περιγράφονται παρακάτω:

#### **Clear variations only**

Αυτή η επιλογή καθαρίζει όλες τις παραλλαγές εγγραφών στην cache, αλλά κρατά τα αποθηκευμένα κλειδιά. Έτσι σε περίπτωση που έχει αλλαχθεί κάποιο στοιχείο, χωρίς να έχουν αλλαχθεί οι οδηγίες cache, μπορεί να αποφευχθεί η περιττή ανάλυση των οδηγιών καθαρίζοντας με αυτήν την επιλογή μόνο τις παραλλαγές εγγραφών.

#### **Clear cache keys and variations completely**

Αυτή η επιλογή καθαρίζει όλη την cache. Σε περίπτωση που δημοσιευτεί κάτι, η cache καθαρίζει αυτόματα.

#### **Purge JSP Repository**

Η επιλογή αυτή καθαρίζει όλες τις εγγραφές των JSP στοιχείων.

#### **Clear only online/offline variations**

Αυτή η επιλογή είναι μόνο διαθέσιμη εάν επιτρέπεται η αποθήκευση στην cache online/offline πόρων. Καθαρίζει όλες τις online/offline παραλλαγές εγγραφών.

#### **Clear only online/offline keys and variations**

Αυτή η επιλογή είναι μόνο διαθέσιμη εάν επιτρέπεται η αποθήκευση στην cache online/offline πόρων. Καθαρίζει όλα τα στοιχεία (κλειδιά και παραλλαγές εγγραφών τους) για τα online/offline έργα που έχουν αποθηκευτεί στην cache.

#### **Show cached resources with keys**

Αυτή η επιλογή παρουσιάζει έναν κατάλογο όλων των πόρων της cache καθώς και τις οδηγίες cache (δηλ. κλειδιά cache) που έχουν αποθηκευτεί για αυτούς.

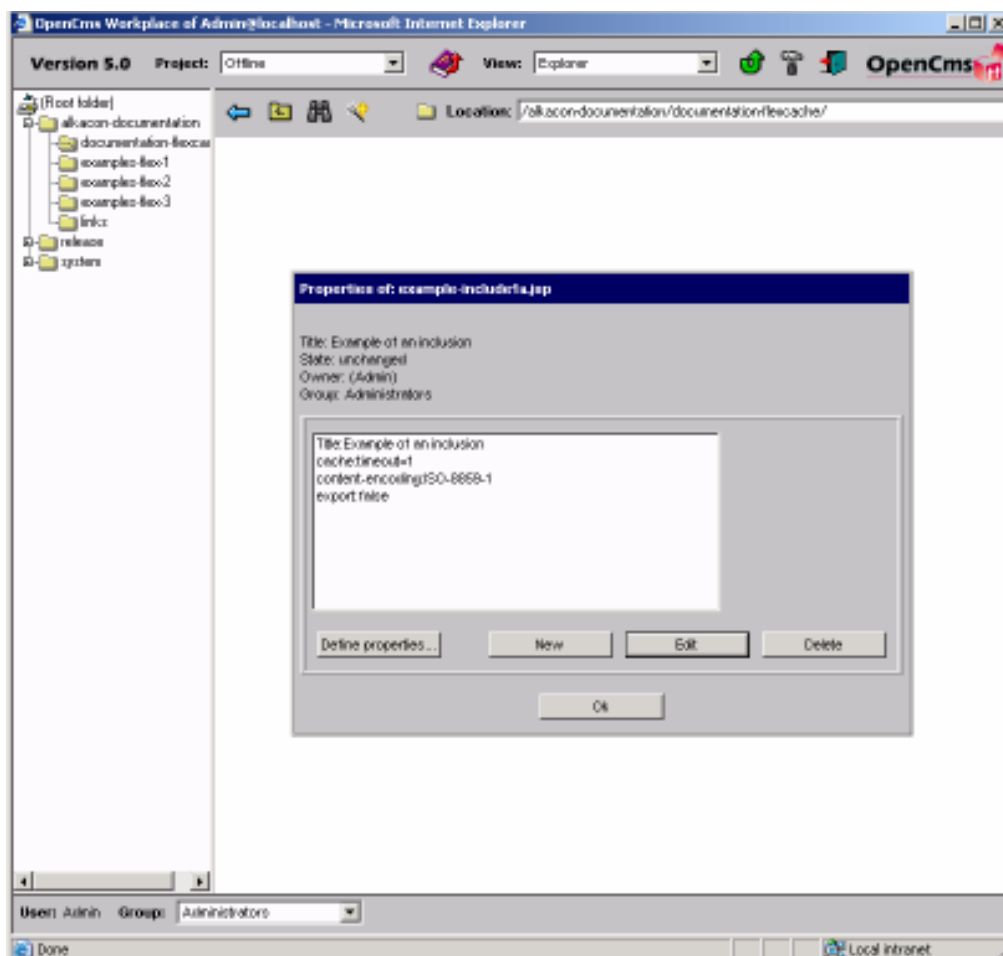
#### **Show cached resources with keys and variations**

Με αυτήν την επιλογή εμφανίζεται ένας κατάλογος των αποθηκευμένων πόρων (όπως περιγράφεται ανωτέρω). Επιπρόσθετα για κάθε πόρο παρατίθενται οι παραλλαγές εγγραφών του.

Εκτός από τη σελίδα διαχείρισης, η FlexCache μπορεί να ελεγχθεί με τη χρησιμοποίηση των παραμέτρων αιτήματος. Παραδείγματος χάριν, εάν χρειάζεται να δουν οι χρήστες πώς εμφανίζεται μια σελίδα όταν δεν προέρχεται από την cache, μπορούν να επισυνάψουν την παράμετρο `_flex=no-cache` στην ερώτηση της σελίδας. Οι επιλογές των παραμέτρων αιτήματος είναι ίδιες με εκείνους που διατίθενται στη σελίδα διαχείρισης για την cache και τα λεκτικά τους ονόματα είναι: `clearcache`, `clearvariations`, `online`, `offline`, `recompile`, και `purge`.

### 3.3.3. Ορισμός οδηγιών cache (cache directives) σε κάποιο στοιχείο

Όλες οι οδηγίες cache μπορούν να τεθούν σε κάθε στοιχείο-πόρο (πχ. σελίδα) του OpenCMS. Για να τεθούν οδηγίες cache σε κάποιον πόρο, αρχικά επιλέγεται η εικόνα του πόρου σε κάποια offline εργασία που τον περιλαμβάνει και από την λίστα που εμφανίζεται η επιλογή “properties”. Στη συνέχεια ένα πλαίσιο διαλόγου παρουσιάζει όλες τις ιδιότητες που έχουν ορισθεί για τον συγκεκριμένο πόρο (σχήμα 3-15).



Σχήμα 3-15: Πλαίσιο διαλόγου ιδιοτήτων του πόρου example-include1a.jsp

Επιλέγεται σε αυτό το πλαίσιο το κουμπί "new" και κατόπιν "cache" και ακολούθως εισάγονται οι απαραίτητες οδηγίες cache. Σε περίπτωση αλλαγής των οδηγιών cache σε κάποιον πόρο πρέπει να καθαριστεί η cache ώστε να ενεργοποιηθούν σε αυτή οι νέες αλλαγές.

### 3.3.4. Οδηγίες FlexCache

Ο ακόλουθος πίνακας παραθέτει όλες τις οδηγίες cache. Αυτές οι οδηγίες πρέπει να συνδεθούν με μια συγκεκριμένη σελίδα χρησιμοποιώντας την ιδιότητα cache. Σε περίπτωση που χρειάζεται να ορισθούν περισσότερες από μια οδηγίες, χρησιμοποιείται το ερωτηματικό ";" για να τις χωρίσει.

**Πίνακας 3-2: Οδηγίες FlexCache**

Never (= false) (default)	Αυτός ο πόρος δεν είναι γράφεται στην cache. Τίποτα δεν θα αποθηκευθεί σε αυτή κάτω από οποιαδήποτε περίπτωση. Η οδηγία αυτή αγνοεί όλες τις άλλες οδηγίες που επίσης θα δοθούν.
Always (= true)	Αυτός ο πόρος γράφεται στην cache και θα υπάρξει πάντα το ίδιο αποτέλεσμα για όλα τα αιτήματα ανάκτησής του, δηλ. μόνο μια παραλλαγή εγγραφής θα σωθεί στην cache για την σελίδα. Η οδηγία αυτή αγνοεί όλες τις άλλες οδηγίες που επίσης θα δοθούν (εκτός της never και no params).
user	Μια παραλλαγή εγγραφής αυτού του πόρου παράγεται για κάθε διαφορετικό χρήστη.
groups groups=(group1, group2, ...)	Η οδηγία groups θα δημιουργήσει μια παραλλαγή εγγραφής οριζόμενου πόρου για κάθε ομάδα που ζητά τη σελίδα. Εάν χρησιμοποιηθεί η οδηγία groups=(group1, group2...) θα υπάρξει μια παραλλαγή εγγραφής μόνο για κάθε group που απαριθμείται.
uri	Μια παραλλαγή εγγραφής του πόρου παράγεται για κάθε "κορυφαίου επίπεδου" URI που ζητείται.
Timeout={integer}	Αυτή η οδηγία θέτει ένα χρονικό διάστημα μετά από το οποίο

	ο αποθηκευμένος πόρος θα ακυρωθεί. Ο δεδομένος ακέραιος αριθμός αντιπροσωπεύει χρονικό διάστημα σε λεπτά.
params params=(param1, param2, ...)	οr Θα δημιουργηθεί μια παραλλαγή εγγραφής για κάθε τιμή των παραμέτρων ή για κάθε συνδυασμό τιμών.
no-params no-params=(no- param1, no-param2, ...)	οr Αυτή η οδηγία αποτρέπει την αποθήκευση ενός πόρου στην cache. Εάν χρησιμοποιηθεί το no-params η παρουσία οποιασδήποτε παραμέτρου στο αίτημα θα αποτρέψει την αποθήκευση του πόρου. Εάν χρησιμοποιηθεί το no-params=(no-param1, no-param2...), η εμφάνιση οποιασδήποτε από τις απαριθμημένες παραμέτρους θα αποτρέψει την αποθήκευση του πόρου.
Schemes schemes=(scheme1, scheme2, ...)	οr Αυτή η οδηγία προκαλεί την δημιουργία μιας παραλλαγής εγγραφής για κάθε σχήμα αιτήματος (ή για κάθε σχήμα αιτήματος που απαριθμείται) πχ "http", "https" κλπ.
ports ports=(port1, port2, ...)	οr Αυτή η οδηγία προκαλεί την δημιουργία μιας παραλλαγής εγγραφής για κάθε (ή για το κάθε απαριθμημένο) TCP port του αιτήματος.

### 3.3.5. Ανάλυση του κώδικα της FlexCache

Ο μηχανισμός της FlexCache υποστηρίζεται από το package com.OpenCMS.flex που περιλαμβάνει τα ακόλουθα sub-packages:

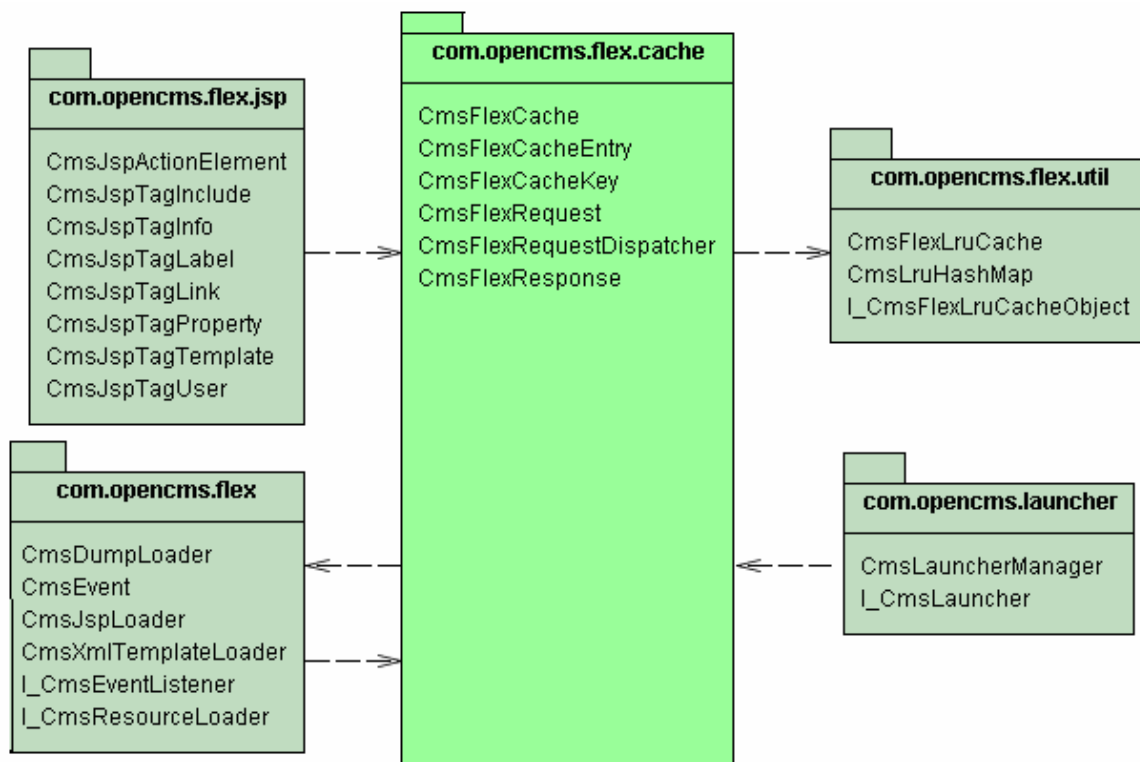
com.OpenCMS.flex.cache

com.OpenCMS.flex.jsp

com.OpenCMS.flex.util

Το διάγραμμα των κλάσεων που ακολουθεί (σχήμα 3-16) παρουσιάζει την σχέση των παραπάνω πακέτων μεταξύ τους. Επιπλέον παρουσιάζονται οι μέθοδοι που αυτά περιέχουν.





Σχήμα 3-16: Διάγραμμα κλάσεων της FlexCache.

Συνοπτικά η λειτουργία της FlexCache έχει ως εξής: Όταν ζητηθεί κάποιο στοιχείο κατά την αίτηση ενός χρήστη καλείται η **OpenCMS.showResource** η οποία αν το στοιχείο είναι JSP καλεί την **CmsJspLoader.initlaunch()** που με την σειρά της καλεί την **CmsJspLoader.load()**. Η τελευταία ελέγχει αν υπάρχουν οδηγίες cache για το στοιχείο και αν ναι καλεί την **CmsFlexRequestDispatcher.include()** η οποία στο εσωτερικό της καλεί τις ακόλουθες βασικές μεθόδους:

**CmsFlexCache.get()** η οποία ελέγχει αν το στοιχείο είναι ήδη στην cache

Αν είναι:

Καλείται η **CmsFlexCacheEntry.service()** για να το φέρει από την cache και να το εμφανίσει.

Διαφορετικά:

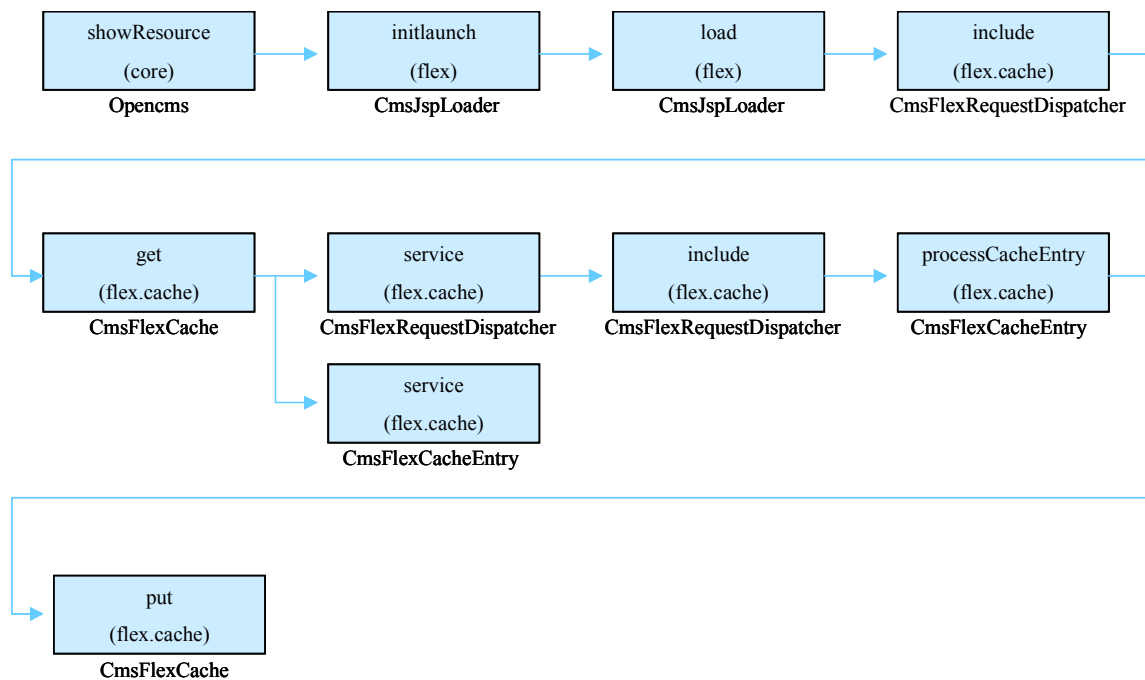
Καλείται η **CmsFlexRequestDispatcher.service()** η οποία καλεί την:

**CmsFlexRequestDispatcher .include()** που εμφανίζει το στοιχείο από το VFS.

Καλείται η **CmsFlexCacheEntry.processCacheEntry()** που δημιουργεί ένα **CmsFlexCacheEntry** αντικείμενο.

Καλείται η **CmsFlexCache.put()** που εισάγει το **CmsFlexCacheEntry** αντικείμενο στην cache.

Διαγραμματικά η λειτουργία της FlexCache παρουσιάζεται στο σχήμα που ακολουθεί (Σχήμα 3-17). Εσωτερικά σε κάθε πλαίσιο αναφέρεται το package (μέσα σε παρένθεση) και η μέθοδος που καλείται, ενώ έξω από το πλαίσιο, η κλάση στην οποία βρίσκεται η μέθοδος αυτή.



Σχήμα 3-17: Κλάσεις που καλούνται κατά την λειτουργία της FlexCache.

Στις παρακάτω παραγράφους ακολουθεί η ανάλυση των βασικότερων κλάσεων του μηχανισμού FlexCache. Η ανάλυση θα γίνει ακολουθώντας την ιεραρχία των packages της FlexCache.

### Package com.OpenCMS.flex

Είναι το package το οποίο περιέχει τα packages και κλάσεις του μηχανισμού FlexCache. Οι βασικές κλάσεις που έχουν αναπτυχθεί σε αυτό είναι οι εξής:

Η κλάση **CmsJspLoader** υλοποιεί τις διεπαφές **I\_CmsLauncher** και **I\_CmsResourceLoader**. Η κλάση αυτή αποτελεί τον loader που επιτρέπει την εκτέλεση των JSP στο OpenCMS σύστημα. Δεν κληρονομεί τον

**com.OpenCMS.launcher.A\_CmsLauncher**, διότι τα JSP δεν σχετίζονται με τον μηχανισμό των templates. Ωστόσο υλοποιεί τη διεπαφή **I\_CmsLauncher** ώστε τα JSP να μπορούν να εισαχθούν ως υποστοιχεία σε **XMLTemplate** σελίδες. Η διεπαφή **I\_CmsResourceLoader** περιγράφει ένα loader του OpenCMS, μέσω του οποίου κάποιος πόρος μπορεί να φορτωθεί από το VFS, να γίνει η επεξεργασία του και να παραδοθεί το αποτέλεσμα της επεξεργασίας μέσω του OpenCMS στον χρήστη. Οι βασικές μέθοδοι που αποτελούν την κλάση είναι:

**initlaunch:** Καλείται από το OpenCMS σύστημα ύστερα από μια αίτηση για την εμφάνιση κάποιου πόρου. Επεξεργάζεται το αίτημα για την εμφάνιση του πόρου και επιστρέφει το αποτέλεσμα. Όλες οι αιτήσεις προωθούνται στην **load()** μέθοδο της κλάσης.

**load:** Πρόκειται για την βασικότερη μέθοδο της **I\_CmsResourceLoader** διεπαφής. Καλείται από την **initlaunch()** αν ζητηθεί από τον χρήστη η εμφάνιση κάποιου JSP. Αν υπάρχουν οδηγίες cache για το στοιχείο καλεί την μέθοδο **CmsFlexRequestDispatcher.include()**.

**service:** Καλείται από την **CmsFlexRequestDispatcher** και παραδίδει τα περιεχόμενα του JSP πόρου που ζητήθηκε. Για τον σκοπό αυτό καλεί την μέθοδο **CmsFlexRequestDispatcher.include()**.

#### **Package com.OpenCMS.flex.cache**

Είναι το package που υλοποιεί τον βασικό μηχανισμό της FlexCache. Οι κλάσεις που περιέχει είναι οι: **CmsFlexRequestDispatcher**, **CmsFlexCache**, **CmsFlexCacheEntry**, **CmsFlexCacheKey**, **CmsFlexRequest**, **CmsFlexResponse**.

Η κλάση **CmsFlexRequestDispatcher** υλοποιεί τη διεπαφή **javax.servlet.RequestDispatcher** και επιτρέπει σε JSP πόρους να ανακτώνται από το OpenCMS σύστημα. Ο dispatcher μπορεί να εμφανίζει δεδομένα είτε από το πραγματικό σύστημα αρχείων του υπολογιστή (πχ στην περίπτωση JSP σελίδων) είτε από το σύστημα αρχείων VFS του OpenCMS είτε από την cache. Οι βασικότερη μέθοδος της κλάσης είναι **include** που λειτουργεί ως εξής:

Εφόσον είναι ενεργοποιημένη η cache για τον πόρο που έχει ζητηθεί να εμφανιστεί, ελέγχεται μέσω της **CmsFlexCache.get()** αν υπάρχει έγκυρη εγγραφή του πόρου στην cache και στην περίπτωση που υπάρχει, καλείται η **CmsFlexCacheEntry.service()** που θα το φέρει από την cache και θα το εμφανίσει. Στην αντίθετη περίπτωση, κατασκευάζεται το κατάλληλο κλειδί για την οδηγία cache του πόρου και αποθηκεύεται στην cache. Κατόπιν καλείται η **CmsFlexRequestDispatcher.service()** η οποία καλεί ξανά την **CmsFlexRequestDispatcher .include()**. Η τελευταία εμφανίζει το στοιχείο από το VFS. Τέλος καλούνται διαδοχικά η μέθοδος **CmsFlexCacheEntry.processCacheEntry()** που δημιουργεί ένα **CmsFlexCacheEntry** αντικείμενο και στη συνέχεια η μέθοδος **CmsFlexCache.put()** που εισάγει το αντικείμενο στην cache.

Η κλάση **CmsFlexCache** υλοποιεί την cache μνήμη. Η δομή δεδομένων που χρησιμοποιεί είναι ένα hashtable δύο επιπέδων. Στο πρώτο επίπεδο υπολογίζεται από το όνομα του πόρου, δηλ. το όνομα του πόρου όπως αναφέρεται στο VFS του OpenCMS μαζί με ένα επίθημα [ on-line ] ή [ off-line ] που επισυνάπτεται στο όνομα του πόρου ώστε να υπάρχει διάκριση μεταξύ των on-line και σε μη off-line έργων του OpenCMS. Το δεύτερο επίπεδο υπολογίζεται από το κλειδί-cache του πόρου, το οποίο είναι string που αντιπροσωπεύει την συγκεκριμένη παραλλαγή εγγραφής στην cache η οποία δημιουργήθηκε για τον πόρο. Οι εγγραφές στο πρώτο επίπεδο της cache είναι τύπου **CmsFlexCacheVariation**, η οποία είναι υποκλάση της **CmsFlexCache**. Αυτή η κλάση είναι ένας απλός τύπος δεδομένων που περιέχει μια λίστα από παραλλαγές εγγραφών τύπου **CmsFlexCacheEntry**.

Η κλάση **CmsFlexCacheEntry** περιέχει το περιεχόμενο ενός αποθηκευμένου πόρου της cache. Περιλαμβάνει μια λίστα που περιλαμβάνει το προ-παραγμένο περιεχόμενο του πόρου, κλήσεις τύπου include() σε άλλους πόρους (μαζί με τις παραμέτρους αιτήματος) καθώς και επιγραφές HTTP που αυτός ο πόρος απαιτεί να ορισθούν.

Το προ-παραγμένο περιεχόμενο σώζεται σε πίνακες τύπου byte[]. Η κλήσεις τύπου include() σώζονται ως strings και οι παράμετροί τους σε δομή HashMap. Οι HTTP επιγραφές σώζονται επίσης σε δομή HashMap. Σε περίπτωση ύπαρξης επανακατεύθυνσης (redirect), ο πόρος της επανακατεύθυνσης αποθηκεύεται ως string.

Ένα **CmsFlexCacheEntry** αντικείμενο μπορεί επίσης να περιέχει μια χρονική τιμή, η οποία δείχνει το χρονικό διάστημα πέρα από το οποίο το αντικείμενο θα είναι άκυρο και θα πρέπει να καθαριστεί από την cache.

Η κλάση **CmsFlexCacheKey** υλοποιεί το κλειδί που περιγράφει την οδηγία cache ενός συγκεκριμένου πόρου.

Η κλάση **CmsFlexRequest** είναι wrapper κλάση της **HttpServletRequest**. Ενθυλακώνει την στάνταρ **HttpServletRequest** αίτηση ώστε το αποτέλεσμα αυτής να παραδίδεται στην κλάση **CmsFlexCache**.

Η κλάση **CmsFlexResponse** είναι wrapper κλάση της **HttpServletResponse**. Ενθυλακώνει την στάνταρ **HttpServletResponse** αίτηση ώστε το αποτέλεσμα αυτής να παραδίδεται στην κλάση **CmsFlexCache**.

Package `com.OpenCMS.flex.util`

Περιέχει κλάσεις υποστήριξης του μηχανισμού **FlexCache**. Η κυριότερη από αυτές είναι η **CmsFlexLruCache**.

Η **CmsFlexLruCache** υλοποιεί την πολιτική βάση της οποίας τα αντικείμενα αποθηκεύονται στην cache. Η πολιτική αυτή ακολουθεί τον αλγόριθμο του τελευταίου-πρόσφατα-χρησιμοποιημένου αντικειμένου (Last Recently Used – LRU) για να βγάλει κάποιο αντικείμενο από την cache σε περίπτωση που χρειάζεται να αποθηκευτεί στην cache κάποιο νέο αντικείμενο και έχει εξαντληθεί το μέγεθος της cache. Η **CmsFlexLruCache** οργανώνεται ως διπλή συνδεδεμένη λίστα. Τα αντικείμενα που αποθηκεύονται στην cache (τύπου **CmsFlexCacheEntry**) πρέπει να υλοποιούν τις μεθόδους που ορίζει η **I\_CmsFlexLruCacheObject** διεπαφή ώστε να ενημερώνονται όταν προστίθονται ή διαγράφονται από την **CmsFlexLruCache**.

## 4. Ορισμός νέων καναλιών διανομής περιεχομένου σε ένα Σύστημα Διαχείρισης Περιεχομένου (CMS)

Σημαντικό μέρος ενός συστήματος διαχείρισης περιεχομένου αποτελεί η δυνατότητα διανομής των διαχειριζόμενων δεδομένων σε διαφορετικά κανάλια με διαφορετικές μορφοποιήσεις. Κάθε κανάλι συνήθως έχει τον δικό του τρόπο επικοινωνίας με το σύστημα και ένα συγκεκριμένο και τυποποιημένο τρόπο μεταφοράς της πληροφορίας. Στόχος ενός συστήματος διαχείρισης περιεχομένου είναι το αποθηκευμένα δεδομένα να μπορούν να είναι προσπελάσιμα με περισσότερους από ένα τρόπους μέσα από το σύστημα και ιδιαίτερα η υποστήριξη ανοικτών και ευέλικτων προτύπων ώστε να εξασφαλίζεται η διαλειτουργικότητα μεταξύ τέτοιων συστημάτων.

### 4.1. Γλώσσα XML

Η XML (eXtensible Markup Language) [10][11][12] αναπτύχθηκε από το World Wide Web Consortium (W3C). Αποτελεί μία απλοποιημένη μορφή της SGML (Standard Generalized Markup Language) και σχεδιάστηκε για να ικανοποιήσει την ανάγκη της περιγραφής και της δόμησης των δεδομένων ενός εγγράφου, με απλό και επεκτάσιμο τρόπο. Η XML δεν είναι μόνο μία γλώσσα σήμανσης και μορφοποίησης (markup language), όπως για παράδειγμα η HTML, αλλά μία μεταγλώσσα βάση της οποίας μπορούν να δημιουργηθούν νέες γλώσσες. Η XML δεν έχει προκαθορισμένες ετικέτες, όπως η HTML, και εστιάζεται στην περιγραφή των δεδομένων και όχι στον τρόπο με τον οποίο αυτά εμφανίζονται. Η XML είναι επεκτάσιμη και ευέλικτη, καθώς επιτρέπει την προσθήκη νέων ετικετών και στοιχείων για την περιγραφή των δεδομένων. Η περιγραφή και ο ορισμός των στοιχείων που χρησιμοποιούνται σε κάποιο XML αρχείο, γίνεται με τη χρήση ενός ορισμού τύπου εγγράφου (DTD: Document Type Definition) ή ενός σχήματος XML (XML Schema) κυρίως σε νεότερες υλοποιήσεις. Από την άλλη, ο καθορισμός της μορφοποίησης και του τρόπου εμφάνισης τους μπορεί να γίνει μέσω Cascading Style Sheets, μέσω της γλώσσας XSL ή μέσω της χρήσης κάποιας γλώσσας προγραμματισμού που διαθέτει τις κατάλληλες δυνατότητες, όπως είναι για παράδειγμα η Java. Μέσα από τη δυνατότητα της XML να δημιουργεί νέες γλώσσες βασισμένες στους κανόνες και τις

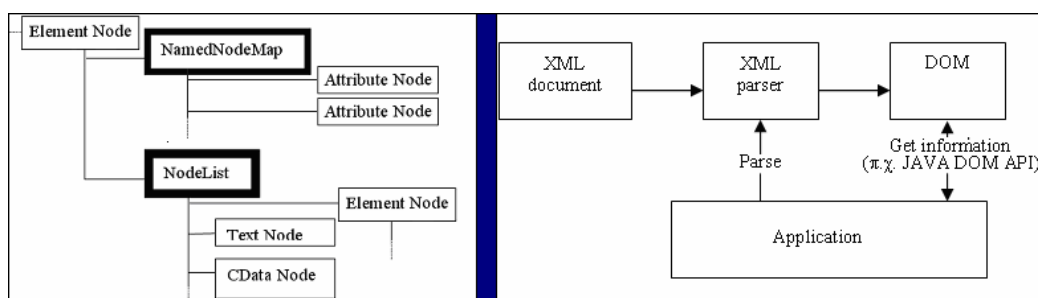
αρχές που τη διέπουν, έχουν δημιουργηθεί διάφορες γλώσσες εξειδικευμένες σε συγκεκριμένα επιστημονικά πεδία. Παράδειγμα αποτελεί η γλώσσα WML (Wireless Markup Language), που χρησιμοποιείται για τη μορφοποίηση και σήμανση εφαρμογών Διαδικτύου για φορητές συσκευές, όπως τα κινητά τηλέφωνα.

Το βασικό πλεονέκτημα της XML είναι ότι μπορεί να χρησιμοποιηθεί σε κάθε πλατφόρμα, επειδή αποτελείται από κώδικα σε μορφή κειμένου (ASCII). Ένα άλλο πλεονέκτημα της XML είναι ότι είναι επεκτάσιμη και προσφέρεται δωρεάν. Οι προγραμματιστές μπορούν να ορίσουν ότι ετικέτα θέλουν για την περιγραφή και τη δόμηση των δεδομένων που επιθυμούν. Λόγω των προαναφερθέντων πλεονεκτημάτων το πρότυπο της XML εξαπλώθηκε πάρα πολύ γρήγορα και χρησιμοποιείται από πολλές εταιρείες ανάπτυξης λογισμικού. Με την XML μπορεί να επιτευχθεί η ανταλλαγή δεδομένων μεταξύ διαφορετικών εφαρμογών ή/και συστημάτων. Τα υπολογιστικά συστήματα και οι βάσεις δεδομένων συχνά περιέχουν δεδομένα σε μη συμβατές μορφές. Μία από τις πιο χρονοβόρες προκλήσεις είναι η ανταλλαγή δεδομένων μεταξύ τέτοιων συστημάτων μέσω Διαδικτύου. Μετατρέποντας τα δεδομένα σε XML, μειώνεται σημαντικά η πολυπλοκότητα και δημιουργούνται δεδομένα που μπορούν να χρησιμοποιηθούν από πολλές διαφορετικές εφαρμογές, ή πλατφόρμες γενικότερα. Η XML έχει επίσης δώσει λύσεις σε εφαρμογές ηλεκτρονικού εμπορίου. Στο ηλεκτρονικό εμπόριο τύπου “πελάτη-προς-επιχείρηση” (C2B), η XML παρέχει έναν ευέλικτο τρόπο για τη δόμηση των πληροφοριών που παρουσιάζονται στον πελάτη, όπως οι πληροφορίες που αφορούν ένα προϊόν, ή οι πληροφορίες που αφορούν το καλάθι αγορών του καταναλωτή. Τα δεδομένα μπορούν να διαχωριστούν από τη σελίδα HTML και με τη χρήση CSS ή XSL να εμφανιστούν σε ένα φυλλομετρητή. Η XML χρησιμοποιείται επίσης στο ηλεκτρονικό εμπόριο τύπου “επιχείρηση-προς-επιχείρηση” (B2B), όπου εξελίσσεται ως η βασική γλώσσα για την ανταλλαγή οικονομικής πληροφορίας μέσω Διαδικτύου. Η έρευνα για την ανάπτυξη προτύπων XML ικανών να υποστηρίξουν ανταλλαγή πληροφορίας, όπως ηλεκτρονικές παραγγελίες, ηλεκτρονικά τιμολόγια κτλ., βρίσκεται σε εξέλιξη.

## **4.2. Document Object Model (DOM)**

Το Document Object Model (DOM) μας βοηθά να θεωρήσουμε τη δενδρική μορφή του αρχείου XML με αντικειμενοστραφή τρόπο και να επέμβουμε σε αυτό με

συγκεκριμένα interfaces. Με το XML DOM μπορούμε να δημιουργήσουμε ένα αρχείο XML, να πλοηγηθούμε μέσα σε αυτό και να προσθέσουμε, μεταβάλουμε και αφαιρέσουμε στοιχεία του. Ένα XML αρχείο περνά από έναν parser, ο οποίος δημιουργεί και τα σχετικά αντικείμενα σύμφωνα με τις προδιαγραφές που ορίζει το DOM. Χρησιμοποιώντας τα διαθέσιμα interfaces (μέσω JavaScript, VBScript, Java κλπ.) μπορούμε να κάνουμε τις επιθυμητές ενέργειες. Κεντρική ιδέα του DOM είναι το Node Object με το αντίστοιχό του, Node Interface. Όμως εδώ ως Node (κόμβος) νοείται όχι μόνο το κάθε στοιχείο (element) του XML, αλλά πέραν του Element Node, έχουμε το Attribute Node, το Text Node, το CDATA Node και άλλα περιφερειακής σημασίας. Η όλη δενδρική δομή, στην οποία εντάσσεται κατά DOM κάθε αρχείο XML, φαίνεται στο Σχήμα 4-1. Όλα τα μη μαυρισμένα τετράγωνα είναι Node Objects. Σημαντικό χαρακτηριστικό του DOM είναι ότι η διαδικασία του parsing γίνεται αυτόματα και χωρίς τη δική μας επέμβαση κατά την εξέλιξή της. Όπως φαίνεται και στο Σχήμα 4-1, ο parser μετατρέπει το κείμενο XML στο αντίστοιχο δένδρο το οποίο διατίθεται στο προγραμματιστικό μας περιβάλλον.



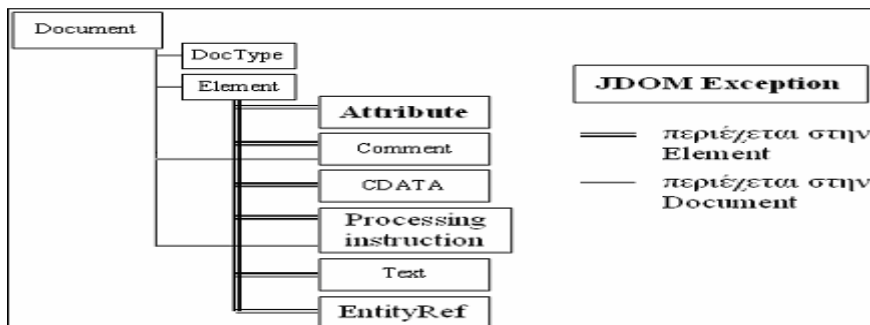
Σχήμα 4-1: Αριστερά η ιεραρχία κατά το DOM. Δεξιά η γενική αρχή DOM

### 4.3. *Java Document Object Model (JDOM)*

Το Java Document Object Model (JDOM) μας προσφέρει πρόσβαση σε ένα αρχείο XML μέσω μίας δενδρικής δομής (όπως και το DOM), αλλά εδώ αποκλειστικά μέσω ανοικτού Java API. Οι σχετικές κλάσεις σε UML μορφή φαίνονται στο Σχήμα 4-2. Το JDOM είναι διαφορετικό από το Java API του DOM, που παρουσιάζεται παραπάνω. Με τις μεθόδους του JDOM επεξεργαζόμαστε τη δομή που δημιουργείται ως αναπαράσταση του XML αρχείου. Ένα αντικείμενο “Document” είναι η αναπαράσταση όλου του αρχείου XML και συγχρόνως περιέχει τα άλλα JDOM αντικείμενα που αντιστοιχούνται σύμφωνα με τις ονομασίες τους άμεσα στις αντίστοιχες του XML. Το (XML) Element του JDOM δεν περιέχει τα



μέρη του ομαδοποιημένα σε NodeList και NamedNodeMap όπως στο DOM. Επιπλέον, μπορούμε να επηρεάσουμε (θέσουμε, αλλάξουμε, σβήσουμε) το περιεχόμενο ενός Element, χωρίς να μεταφερόμαστε και να δουλεύουμε σε επιμέρους κόμβους, όπως θα κάναμε στο DOM (Attribute, Text, CDATA Node, κλπ.) ξεχωριστά. Για παράδειγμα η μέθοδος getAttributes() της Element class επιστρέφει μία (Java) List. Επιπλέον, στο JDOM δεν υπάρχει TextNode. Το κείμενο (text) ενός Element απλά επιστρέφεται από τη μέθοδο getText() του Element. Κατά συνέπεια, αντικείμενα όπως Element, Attribute, ProcessingInstruction, Comment κλπ. μπορούν να δημιουργηθούν με το new. Αυτό χρησιμοποιείται για την κατασκευή ενός XML κειμένου εκ του μηδενός. Η ουσιαστική διαφορά έγκειται ότι ενώ το JDOM δεν νοείται έξω από την Java, οι δομές του DOM είναι ανεξάρτητες και πέραν από συγκεκριμένη γλώσσα προγραμματισμού, κάτι το οποίο δεν θα μας επηρεάσει στη συγκεκριμένη εργασία, λόγω ότι εμείς θα χειριζόμαστε τα XML δεδομένα μας μέσα από τη Java. Το JDOM παρέχει τον τρόπο πρόσβασης στα στοιχεία του αρχείου XML, αλλά για την εισαγωγή/εξαγωγή του XML αναγκαστικά βασίζεται σε κάποιον parser (συγκεκριμένα στον SAX parser που παρουσιάζεται παρακάτω) ή στην δενδρική δομή της ιεραρχίας του DOM (αν διατίθενται τέτοια).

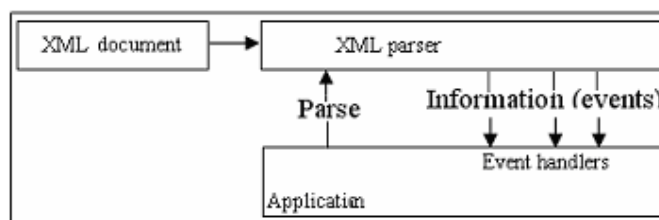


Σχήμα 4-2: Τα αντικείμενα (object model) του JDOM

#### 4.4. Simple API for XML (SAX)

Στη βάση όλων των παραπάνω βρίσκεται το SAX (Simple API for XML), το οποίο προσφέρεται από τον SAX parser, τον οποίο μπορούμε να φανταστούμε σαν έναν σειριακό αναγνώστη χαρακτήρα προς χαρακτήρα (SAX reader), του κειμένου XML. Ο SAX parser αναγνωρίζοντας τα διάφορα μέρη του αρχείου XML, δημιουργεί αντίστοιχα συμβάντα (events), στα οποία το δικό μας πρόγραμμα οφείλει να ανταποκριθεί σύμφωνα με τις επιθυμίες μας. Ο SAX parser διαβάζει τον “συρμό

χαρακτήρων” του κειμένου XML και ανιχνεύει-σταματά σε συγκεκριμένα σημεία. Σε κάθε σταμάτημα δημιουργείται ένα συμβάν (event) προς τον δικό μας event handler. Μόλις ο event handler αντιμετωπίσει το event, ο parser συνεχίζει μέχρι το επόμενο σημείο. Στο Σχήμα 4-3 φαίνεται η σειριακή είσοδος του κειμένου XML στον SAX parser. Αυτό ελέγχεται από την εφαρμογή στην οποία δηλώνει πίσω (callbacks) συμβάντα, δηλαδή την ανίχνευση των επί μέρους μερών του υπό ανάγνωση αρχείου XML.



Σχήμα 4-3: Γενική αρχή SAX parser

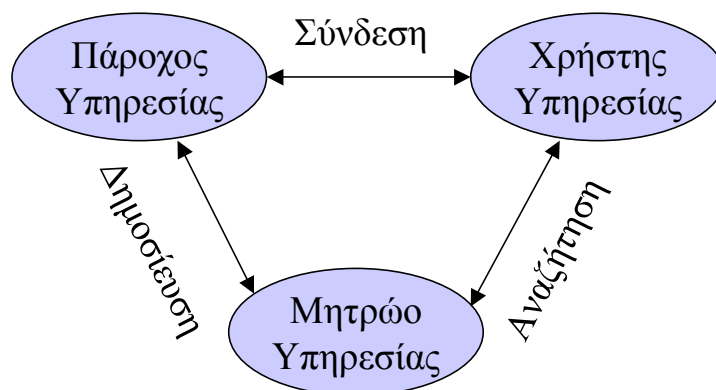
Σε μορφή interface μέσα στον SAX parser υπάρχει ενσωματωμένος ο ContentHandler, ο οποίος είναι ένας event handler κατά τα γνωστά της Java, ο οποίος δημιουργεί κλήσεις πίσω στην εφαρμογή κάθε φορά που ένα “αξιοπρόσεκτο” γεγονός (parsing event) διαπιστώνεται κατά την σειριακή ανάγνωση του αρχείου XML. Επιπλέον έχουμε τον XMLEventsPresentor, ο οποίος οφείλει να υλοποιήσει όλες τις σχετικές μεθόδους, έστω και με τετριμμένο τρόπο (άμεση επιστροφή χωρίς καμία ενέργεια). Παρόμοια με τον ContentHandler, υπάρχουν στον SAX και άλλα τρία παρόμοια interfaces (ErrorHandler, DTDHandler και EntityResolver) με αντίστοιχη φιλοσοφία χρήσης. Με τον SAX parser βρισκόμαστε στο κατώτερο προγραμματιστικά περιβάλλον, το οποίο μας δίνει πλήρη πρόσβαση στο XML. Η ένταξη των δικών μας επιθυμιών (εφαρμογή μας) είναι πιο επίπονη από ότι με το DOM ή JDOM, αλλά οι διαθέσιμες δυνατότητες και η ταχύτητα εδώ υπερτερούν. Δεν περιμένουμε μέχρι να δημιουργηθεί και να μας προσφερθεί η πλήρης δομή του αρχείου XML (όπως με το DOM ή το JDOM), αλλά ειδοποιούμαστε και μπορούμε να ανταποκριθούμε στα σχετικά συμβάντα on the fly, καθώς διαβάζεται το XML κείμενο και χωρίς να περιμένουμε το τέλος αυτής της ανάγνωσης.

## 4.5. Υπηρεσίες Ιστού (Web Services)

Παράλληλα με την μεγάλη επιτυχία και εξάπλωση της χρήσης της γλώσσας σήμανσης XML για τον χαρακτηρισμό, την αποθήκευση και την ανταλλαγή εγγράφων, διεθνείς οργανισμοί και εταιρίες προσπάθησαν να προδιαγράψουν ένα κοινώς αποδεκτό τρόπο για την μεταφορά τέτοιου περιεχομένου πάνω από δίκτυα IP. Σκοπός ήταν η ελάχιστη δυνατή μεταβολή της ήδη υπάρχουσας υποδομής, αλλά και η εκμετάλλευση της γνώσης που είχε συσσωρευτεί για την περίπτωση επιλογής και σχεδίασης middleware αλλά και διαδικτυακών εφαρμογών-πρωτοκόλλων.

Αποτέλεσμα της συνεργασίας αρκετών φορέων ήταν ο σχεδιασμός του μοντέλου των Υπηρεσιών Ιστού [13][14], όπου στηρίχθηκε στην επιτυχημένη ιδέα της XML αλλά και στην γνώση και ευρεία εξάπλωση του πρωτοκόλλου HTTP.

Η ιδέα που οι υπηρεσίες Ιστού προσπάθησαν να δώσουν στην κοινότητα του Διαδικτύου ήταν ο σχεδιασμός ενός κατανεμημένου περιβάλλοντος πάνω στο οποίο ένα σύνολο από εφαρμογές θα μπορούν να αλληλεπιδρούν ανεπηρέαστα από την υποκείμενη πλατφόρμα υλοποίησης, και βασιζόμενες σε τεχνολογίες και πρωτόκολλα του Διαδικτύου.



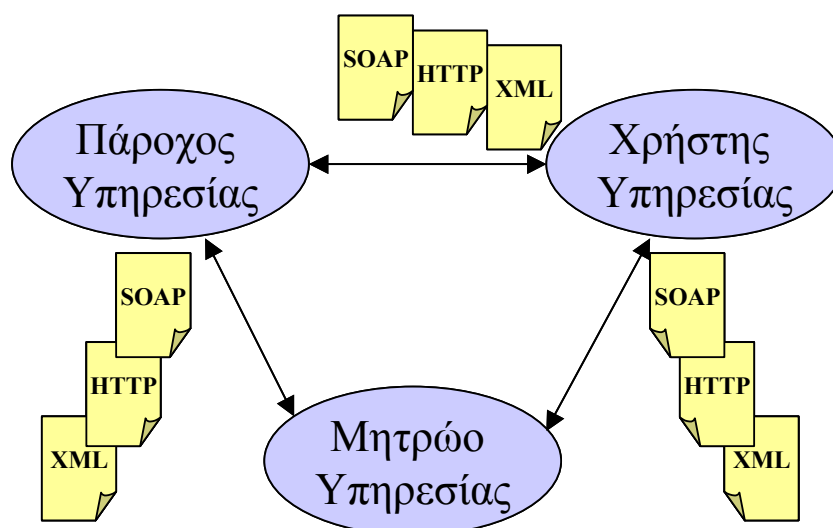
Σχήμα 4-4 Ρόλοι και λειτουργίες των Υπηρεσιών Ιστού

Οι Υπηρεσίες Ιστού στηρίζονται στις ακόλουθες τεχνολογίες:

### 4.5.1. SOAP (Simple Object Access Protocol)

Το SOAP [13][14] είναι ένα πρωτόκολλο μηνυμάτων ανεξάρτητο από το τρόπο μεταφοράς. Ορίζει μονόδρομα μηνύματα τα οποία μπορούν να συνδυαστούν σε ακολουθίες αιτήματος-απόκρισης ώστε να κατασκευαστεί ένας αμφίδρομος τρόπος επικοινωνίας. Κάθε μήνυμα του SOAP είναι ένα έγγραφο XML και η προδιαγραφή καθορίζει το σχήμα του μηνύματος-εγγράφου XML αλλά όχι το είδος

και την μορφή του περιεχομένου και πώς στέλνεται αυτό πραγματικά. Το SOAP, εντούτοις, διευκρινίζει πώς τα μηνύματα δρομολογούνται πάνω από το πρωτόκολλο HTTP ( δίνοντας όμως την δυνατότητα να χρησιμοποιηθεί και πάνω από άλλα πρωτόκολλα του επιπέδου εφαρμογής όπως το SMTP).



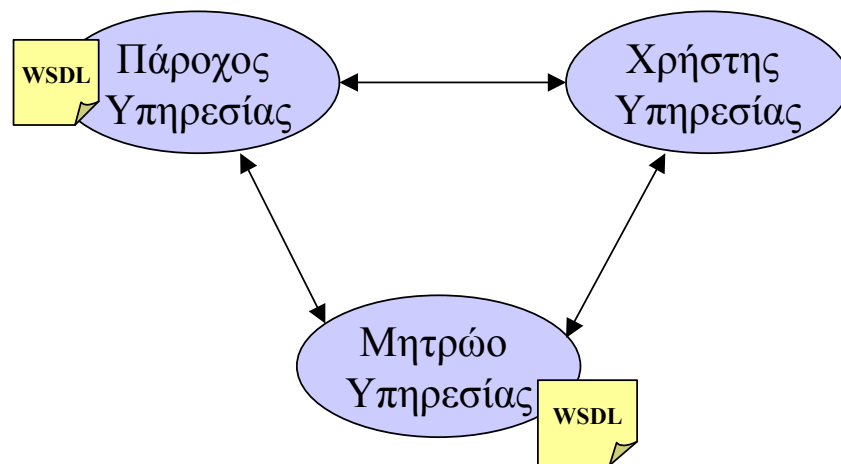
Σχήμα 4-5 Η χρήση του πρωτοκόλλου SOAP

Κάθε μήνυμα του SOAP έχει ως ρίζα του εγγράφου XML ένα στοιχείο με όνομα <Envelope>. Το στοιχείο αυτό περιλαμβάνει όλα τα άλλα στοιχεία του εγγράφου. Μέσα στο "Envelope" (φάκελος) υπάρχουν δύο μέρη: μια επικεφαλίδα και ένα σώμα. Η επικεφαλίδα περιέχει τα δεδομένα δρομολόγησης ή εννοιολογική πληροφορία και μπορεί να είναι κενό. Το σώμα περιέχει το πραγματικό μήνυμα το οποίο μπορεί επίσης να είναι κενό.

#### 4.5.2. WSDL (Web Services Description Language)

Μια Υπηρεσία Ιστού είναι στην ουσία άχρηστη αν άλλοι δεν μπορούν να ανακαλύψουν τι κάνει και πώς ώστε να την χρησιμοποιήσουν. Για να μπορέσει ένα πρόγραμμα πελάτης να επικοινωνήσει και να χρησιμοποιήσει τις υπηρεσίες μίας Υπηρεσίας Ιστού χρησιμοποιείται ένας τρόπος περιγραφής που ονομάζεται WSDL [14][15]. Η WSDL είναι μία γλώσσα σήμανσης βασισμένη στην XML και χρησιμοποιείται για αυτό το σκοπό, να καθορίσει τις υπηρεσίες και να περιγράψει τον τρόπο πρόσβασης στις μεθόδους που παρέχει ένα web service. Πιο συγκεκριμένα περιγράφει τα δεδομένα και τις μεθόδους που μπορεί κανείς να ανταλλάξει με ένα

web service. Έχοντας το έγγραφο WSDL ενός web service το πρόγραμμα πελάτης γνωρίζει τι μπορεί να καλέσει και με ποιες παραμέτρους.

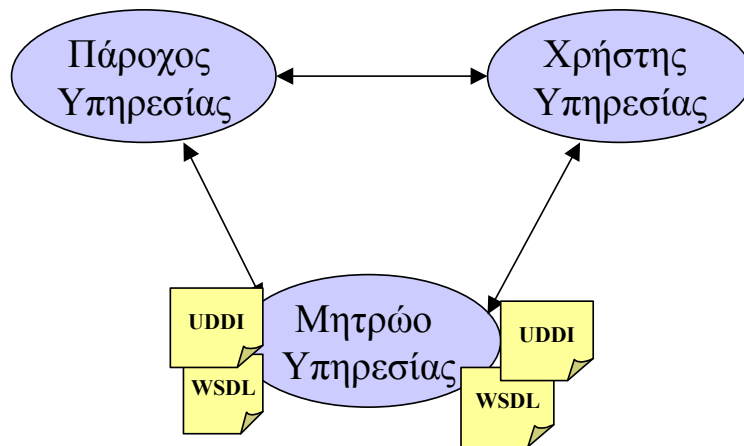


Σχήμα 4-6 Η χρήση του WSDL

#### 4.5.3. UDDI (Universal Description, Discovery and Integration)

Το UDDI [14][16] είναι ένα πρότυπο για την περιγραφή, την δημοσίευση και την αναζήτηση Υπηρεσιών Ιστού που παρέχει μία επιχείρηση. Χρησιμοποιείται για να κατασκευάζεται ένα κατακευματισμένο μητρώο πληροφορίας για διάφορες Υπηρεσίες Ιστού. Μετά την ανάπτυξη ενός web service και την περιγραφή του με WSDL, το UDDI παρέχει τον τρόπο με τον οποίο αυτή η πληροφορία μπορεί να φτάσει στους τελικούς χρήστες του web service. Στην ουσία το web service δημοσιεύεται σε ένα μητρώο UDDI όπου πιθανοί χρήστες θα μπορέσουν να το εντοπίσουν και να το χρησιμοποιήσουν.

Το περιεχόμενο ενός μητρώου UDDI μοιάζει με την ιδέα του τηλεφωνικού καταλόγου, όπου υπάρχει η δυνατότητα να βρεθεί μία επιχείρηση αλλά και οι υπηρεσίες που παρέχει.



Σχήμα 4-7 Η περιγραφή στο μητρώο με UDDI

#### 4.6. **Web Services στα πλαίσια του OpenCMS**

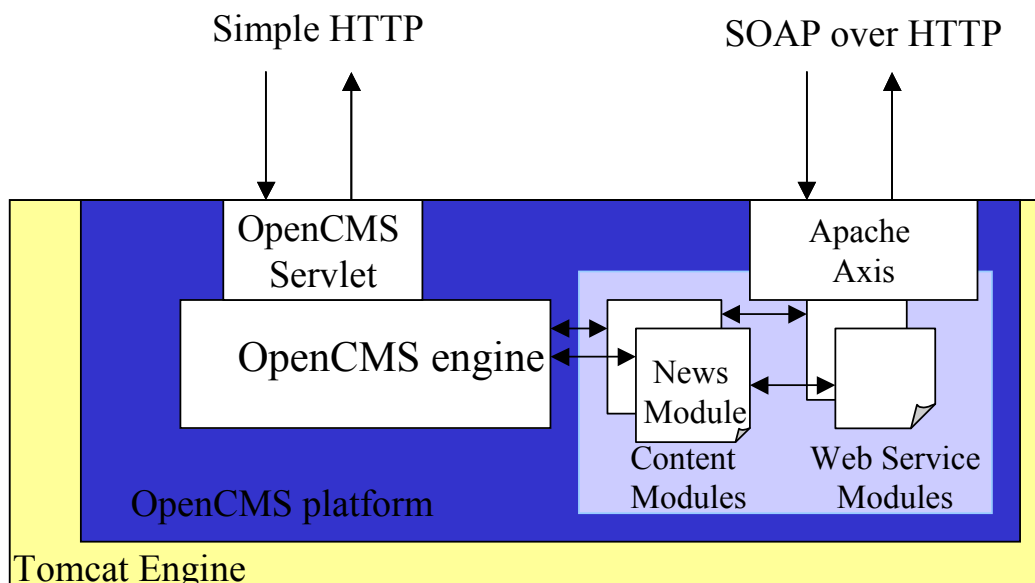
Η πλατφόρμα OpenCMS προσφέρει την δυνατότητα εισαγωγής νέας λειτουργικότητας μέσα από τον μηχανισμό των modules. Τα modules μπορούν να υποστηρίζονται από κώδικα και να διαθέτουν διεπαφές ελέγχου και διαχείρισης από τον διαχειριστή του συστήματος, αλλά και δημιουργίας από τον τελικό χρήστη.

Σκοπός είναι να εισαγάγουμε την δυνατότητα δημοσίευσης περιεχομένου που υπάρχει και κατασκευάζεται μέσα στο OpenCMS σε ένα κανάλι Υπηρεσιών Ιστού.

Για την πραγματοποίηση αυτής της λειτουργίας θα στηριχθούμε σε βιβλιοθήκες ανοικτού κώδικα του πρωτοκόλλου SOAP και JAX-RPC που έχουν υλοποιηθεί από το Apache Software Foundation, το έργο ονομάζεται Axis (v.1.1).

Η υλοποίηση θα παρέχει πρόσβαση στα άρθρα που αποθηκεύονται στην πλατφόρμα OpenCMS και θα δίνει μια περιγραφή WSDL ώστε να μπορούν τα προγράμματα πελάτες να παίρνουν το περιεχόμενο.

Η λογική είναι να κατασκευαστεί ένα αντικείμενο περίβλημα (wrapper) στο `com.OpenCMS.modules.homepage.news.NewsContentDefinition` ώστε να μπορούν να χρησιμοποιηθούν οι μέθοδοι που παρέχει και να επεκταθεί ως web service, παρέχοντας τις κατάλληλες μεθόδους για αναζήτηση νέων στοιχείων από την βάση. Έχοντας το NewsBean αντικείμενο, θα κατασκευαστεί ένα web service έτσι ώστε να είναι προσπελάσιμη η πληροφορία μέσα από το Apache Axis framework.



Σχήμα 4-8 Η Χρήση Web Services στο OpenCMS

Η χρήση του web service framework θα επιτρέψει την δυνατότητα επέκτασης του συστήματος σε πιο δυναμικές και κατανεμημένες αρχιτεκτονικές. Ιδιαίτερα στα πλαίσια της ανάπτυξης ενός κατανεμημένου συστήματος αναζήτησης πόρων στηριζόμενο στην πλατφόρμα J2EE, είναι δυνατό να δοθεί μέρος της αναζήτησης που ενδιαφέρει τον τελικό χρήστη (και όχι τον χρήστη της πλατφόρμας ο οποίος θα έχει δικαιώματα χρήσης της κατανεμημένης αναζήτησης) ως υπηρεσία ιστού, με αποτέλεσμα να μπορεί να εκμεταλλευτεί τους κατανεμημένους πόρους και ένας χρήστης που δεν ανήκει στο ιδιωτικό δίκτυο του OpenCMS. Προφανώς στην συγκεκριμένη περίπτωση θα εφαρμόζονται περιορισμοί ασφάλειας, όπως ακριβώς ορίζει το OpenCMS, για τους εξωτερικούς χρήστες της πλατφόρμας.

#### 4.7. Υποστήριξη των Web Services

Για την υποστήριξη των web services στην πλατφόρμα OpenCMS, χρειάζεται να κατασκευαστεί ένα αντικείμενο `WebServiceBackoffice` που επεκτείνει την λειτουργικότητα του `A_CmsBackoffice`, το οποίο αναλαμβάνει τις ακόλουθες λειτουργίες :

1. Να καταχωρεί τα modules που είναι διαθέσιμα ως web service στην βάση δεδομένων
2. Να παρέχει πρόσβαση στις wrapper κλάσεις αυτών των modules, ώστε να είναι δυνατή η επικοινωνία του web service με το OpenCMS framework

3. Να δίνει στην πλατφόρμα τα κατάλληλα URLs ώστε να μπορεί να γίνεται προσθήκη, διόρθωση και διαγραφή των παραμέτρων ενός web service
4. Να παρέχονται τα αρχεία WSDL και UDDI στο σύστημα, και να δημοσιεύονται σε προκαθορισμένα μητρώα.

Επειδή κάθε web service μοντελοποιείται ως περιεχόμενο, χρειάζεται να κατασκευαστεί η κλάση `WebServiceContent`, επέκταση της `A_CmsContentDefinition` και η κατάλληλη υποστήριξη στην βάση δεδομένων, ώστε να μπορούν να αποθηκεύονται οι επιμέρους παράμετροι ενός web service στην βάση και να είναι προσπελάσιμες από το σύστημα.

Στην βάση δεδομένων χρειάζεται να κατασκευαστεί πίνακας με όνομα `WEB_SERVICE_MODULE` σύμφωνα με την ακόλουθη δομή

Πεδίο	Τύπος	Μέγεθος	Περιγραφή
id	int	32	Το πρωτεύον κλειδί
web_srv_name	varchar	200	Το όνομα της υπηρεσίας
web_srv_class	varchar	200	Η κλάση με την οποία καλείται η υπηρεσία
web_srv_desc	varchar	255	Μικρή περιγραφή
web_srv_req	int	32	Πλήθος αιτήσεων για την υπηρεσία
web_srv_wsdl	blob		Το WSDL αρχείο που περιγράφει την υπηρεσία
web_srv_uddi	blob		Το UDDI για την εισαγωγή σε κάποιο



			μητρώο
--	--	--	--------

## 5. Επεκτάσεις του OpenCMS

### 5.1. *Εισαγωγή νέου περιεχομένου με την μορφή των modules*

Για την εισαγωγή νέου τύπου περιεχομένου και την διαχείριση αυτού η πλατφόρμα OpenCMS διαθέτει την λογική των modules. Modules μπορούν να χαρακτηριστούν οποιαδήποτε τύπου νέα λειτουργικότητα που φορτώνεται στο σύστημα δυναμικά. Τα modules μπορούν να παρέχουν υποστήριξη τόσο για την εμφάνιση όσο και για την αποθήκευση και διαχείριση (δημιουργία, διόρθωση, διαγραφή) ενός στοιχείου του συστήματος.

Για την δημιουργία νέου τύπου περιεχομένου τρεις είναι οι βασικές λειτουργίες που πρέπει να αναπτυχθούν

- Ορισμός νέου τύπου περιεχομένου
- Κατασκευή αντικειμένων για την υποστήριξη του νέου τύπου ως προς την διαχείριση του από τα παράθυρα ελέγχου της πλατφόρμας OpenCMS.
- Δημιουργία σελίδων διαχείρισης που να επιτρέπουν την δημιουργία, διόρθωση και διαγραφή του περιεχομένου από τους χρήστες του συστήματος.

#### 5.1.1. *Ορισμός νέου τύπου περιεχομένου*

Για τον ορισμό νέου τύπου περιεχομένου, η πλατφόρμα OpenCMS, παρέχει μία αφηρημένη κλάση με όνομα `A_CmsContentDefinition` την οποία ο χρήστης πρέπει να επεκτείνει και να προσθέσει τα πεδία εκείνα που χαρακτηρίζουν το νέο περιεχόμενο που θέλει να ορίσει. Μέσα σε ένα αντικείμενο του τύπου `A_CmsContentDefinition` ενθυλακώνονται τα διάφορα πεδία του

περιεχομένου με συγκεκριμένο τρόπο, ώστε η πλατφόρμα να γνωρίζει τον τρόπο διαχείρισης αυτών.

Αρχικά παρέχουμε κατασκευαστές (constructors) της νέας κλάσης που επιτρέπουν την προσθήκη ή την ανάγνωση του συγκεκριμένου τύπου περιεχομένου από την πηγή δεδομένων.

Για τον ορισμό των πεδίων που χαρακτηρίζουν το περιεχόμενο, στην επέκταση της κλάσης, προσθέτουμε μεθόδους `get` και `set` για όλα τα πεδία, όπως ακριβώς κάνει κανείς όταν ορίζει JavaBeans. Για παράδειγμα αν θελήσουμε να ορίσουμε ένα πεδία με όνομα `author` τότε πρέπει να προστεθούν οι μέθοδοι `getAuthor` και `setAuthor`.

Στην συνέχεια κάνουμε `override` τις `abstract` μεθόδους `write()` και `delete()` ώστε να επιτρέπουμε την ενημέρωση και την διαγραφή στην πηγή περιεχομένου αντίστοιχα. Τέλος προστίθενται στατικές μέθοδοι για να επιστρέφονται ομάδες από τέτοιου τύπου αντικείμενα, στις οποίες μπορούμε να θέτουμε και φίλτρα.

Η αφηρημένη κλάση `A_CmsContentDefinition` παρέχει ακόμα υλοποιημένες μεθόδους `isReadable()` και `isWritable()` για τον έλεγχο πρόσβασης κάποιου χρήστη στον συγκεκριμένο τύπο περιεχομένου. Η εξ' ορισμού υλοποίηση τους επιτρέπει πάντα την πρόσβαση και την εγγραφή δεδομένων, αλλά ο προγραμματιστής μπορεί να υπερκεράσει (`override`) τις μεθόδους αυτές ώστε να προσφέρει διαφορετικού τύπου δικαιώματα στους χρήστες. Το σχήμα που χρησιμοποιείται για τον έλεγχο πρόσβασης είναι αντίστοιχο με αυτό του Unix όπου τα δικαιώματα πρόσβασης ορίζονται για τον ιδιοκτήτη, την ομάδα και τους υπολοίπους.

Η υλοποίηση ενός νέου τύπου περιεχομένου προϋποθέτει την υποστήριξη για αποθήκευση και ανάκτηση από τις πηγές περιεχομένου που διαθέτει η πλατφόρμα. Κατά την υλοποίηση λοιπόν των μεθόδων εγγραφής και ανάγνωσης ο προγραμματιστής πρέπει να αναλάβει να εκτελέσει τα σωστά ερωτήματα SQL στην βάση.

Όταν είναι επιθυμητό να μπορεί ένας χρήστης να διαχειρίζεται το νέο περιεχόμενο από τα μενού διαχείρισης που παρέχει το OpenCMS, τότε πρέπει να προστεθούν στον νέο τύπο περιεχομένου μια σειρά από μεθόδους οι οποίες θα χρησιμοποιηθούν από την πλατφόρμα για εύκολη και άμεση ενσωμάτωση και χρήση

του περιεχομένου από τα μενού διαχείρισης. Οι μέθοδοι που πρέπει να υλοποιηθούν είναι οι ακόλουθες:

- `getFilterMethods()`
- `getFieldNames()`
- `getFieldMethods()`
- `getUniqueId()`
- `isLockable()`

Αυτές οι μέθοδοι χρησιμοποιούνται από τα αντικείμενα του λεγόμενου `BackOffice` (το οποίο παρουσιάζεται σε επόμενη παράγραφο), ώστε να μπορεί να δείχνει στον τελικό χρήστη την πληροφορία.

Για να υπάρχει ενιαίος και ομοιόμορφος τρόπος κατασκευής περιεχομένου, χωρίς να χρειάζεται ο προγραμματιστής να ορίζει κάθε φορά νέα στοιχεία στην βάση δεδομένων, η πλατφόρμα `OpenCMS`, διαθέτει ένα γενικό πλαίσιο εργασίας (framework) για την κατασκευή των modules, ονομαζόμενο `Mastermodule`. Σε αυτό το πλαίσιο εργασίας ο ορισμός ενός νέου περιεχομένου απλοποιείται σε σχέση με την αποθήκευση και την πρόσβαση στις βάσεις δεδομένων. Η πλατφόρμα ορίζει γενικούς πίνακες στην βάση δεδομένων οι οποίοι μπορούν να χρησιμοποιηθούν ανάλογα με τον τύπο περιεχομένου που θέλουμε να ορίσουμε. Το αντικείμενο που πρέπει να επεκταθεί αντί του `A_CmsContentDefinition` είναι το `CmsMasterContent` το οποίο στα πλαίσια του framework παρέχει βελτιώσεις όπως

- Γενικού τύπου πίνακες με πεδία για αποθήκευση ακεραίων, κειμένου και δυαδικών δεδομένων
- Πρόσβαση στην βάση (οι μέθοδοι `write` και `read` του `A_CmsContentDefinition` είναι ήδη υλοποιημένες).
- Τα διάφορα πεδία είναι ενσωματωμένα με τον μηχανισμό των projects που έχει το `OpenCMS`
- Διαχείριση δικαιωμάτων πρόσβασης για τα πεδία
- Τα πεδία μπορούν να οργανωθούν μέσα από κανάλια

Μάλιστα η δυνατότητα χρήσης διαφορετικών βάσεων δεδομένων έχει οργανωθεί ώστε να μπορεί ο προγραμματιστής να γράφει ξεχωριστά αντικείμενα για

πρόσβαση στην βάση σε περίπτωση που επιθυμεί να εκμεταλλευτεί ιδιότητες-χαρακτηριστικά μιας βάσης που δεν υπάρχουν σε κάποια άλλη.

### *5.1.2. Υποστήριξη νέου τύπου περιεχομένου στην διαχείριση*

Για την υποστήριξη ενός νέου τύπου περιεχομένου από την πλατφόρμα διαχείρισης του OpenCMS, ο προγραμματιστής πρέπει να ορίσει ένα νέο BackOffice module που να αναλαμβάνει να επικοινωνεί με το σύστημα και να στέλνει τις κατάλληλες αιτήσεις στην βάση δεδομένων. Επιπλέον του περιεχομένου, ένα αντικείμενο BackOffice παρέχει στο σύστημα εκείνα τα URL που πρέπει να κληθούν με σκοπό να παρουσιαστούν στο χρήστη του συστήματος οι κατάλληλες οθόνες για την διαχείριση του περιεχομένου.

## **5.2. Επέκταση της πλατφόρμας OpenCMS για την αποθήκευση και διαχείριση δομημένου περιεχόμενου**

Η πλατφόρμα OpenCMS βασικά στηρίζεται σε περιεχόμενο αποθηκευμένο σε ένα μόνο πεδίο ενός πίνακα με την μορφή μεγάλου δυαδικού αντικειμένου (BLOB). Το περιεχόμενο αυτό ορίζεται ως αποθηκευμένο περιεχόμενο (stored content) και όπως είναι φανερό δεν επιτρέπει στον χρήστη-προγραμματιστή να αλλάζει εύκολα και αποδοτικά συγκεκριμένες τιμές και χαρακτηριστικά, παρά μόνο να ανακτά ολόκληρο το αποθηκευμένο περιεχόμενο και μετά να το επεξεργάζεται. Παράδειγμα χρησιμότητας ενός τέτοιου είδους αποθήκευσης είναι τα αρχεία html. Στην περίπτωση αυτή είναι επιθυμητό να διαχειριζόμαστε το αρχείο ως ένα ολοκληρωμένο και πλήρες περιεχόμενο. Κάθε φορά που ο χρήστης επιθυμεί να μεταβάλλει το αρχείο, τότε πρέπει να το ανακτήσει ολόκληρο και να κάνει τις αλλαγές που επιθυμεί. Στην περίπτωση όμως που επιθυμούμε να αποθηκεύσουμε περιεχόμενο διαφορετικής μορφής, όπως π.χ. τα στοιχεία ενός πελάτη, τότε η αποθήκευσή τους σε ένα μόνο πεδίο του πίνακα με μορφή κειμένου (ακόμα και δομημένου κειμένου όπως XML) οδηγεί σε προβλήματα και χαμηλής ποιότητας εκμετάλλευση των δυνατοτήτων μιας σχεσιακής βάσης δεδομένων.

Για την κατασκευή δομημένου περιεχομένου η πλατφόρμα OpenCMS παρέχει στον προγραμματιστή την δυνατότητα να χρησιμοποιήσει το στοιχείο `A_CmsContentDefinition` του πακέτου `com.OpenCMS.defaults`, το οποίο υλοποιεί την διεπαφή `I_CmsContent` του πακέτου `com.OpenCMS.template`. Το στοιχείο αυτό, στην ουσία, αποτελεί μια προσέγγιση παρόμοια με τα Java Data Objects για ανάκτηση σε μορφή αντικειμένων, πεδίων από μια σχεσιακή βάση δεδομένων. Με το `A_CmsContentDefinition` παρέχεται ένας απλουστευμένος τρόπος για αντιστοίχιση αντικειμένων σε σχεσιακές βάσεις (Object to Relational, O-R mapping) και αντίστροφα.

Στα πλαίσια της εργασίας, και σύμφωνα με τις απαιτήσεις των προδιαγραφών για δυνατότητα κατασκευής και διαχείρισης δομημένου περιεχομένου οδηγηθήκαμε στο να υλοποιήσουμε ένα μηχανισμό επέκτασης της υπάρχουσας υποδομής.

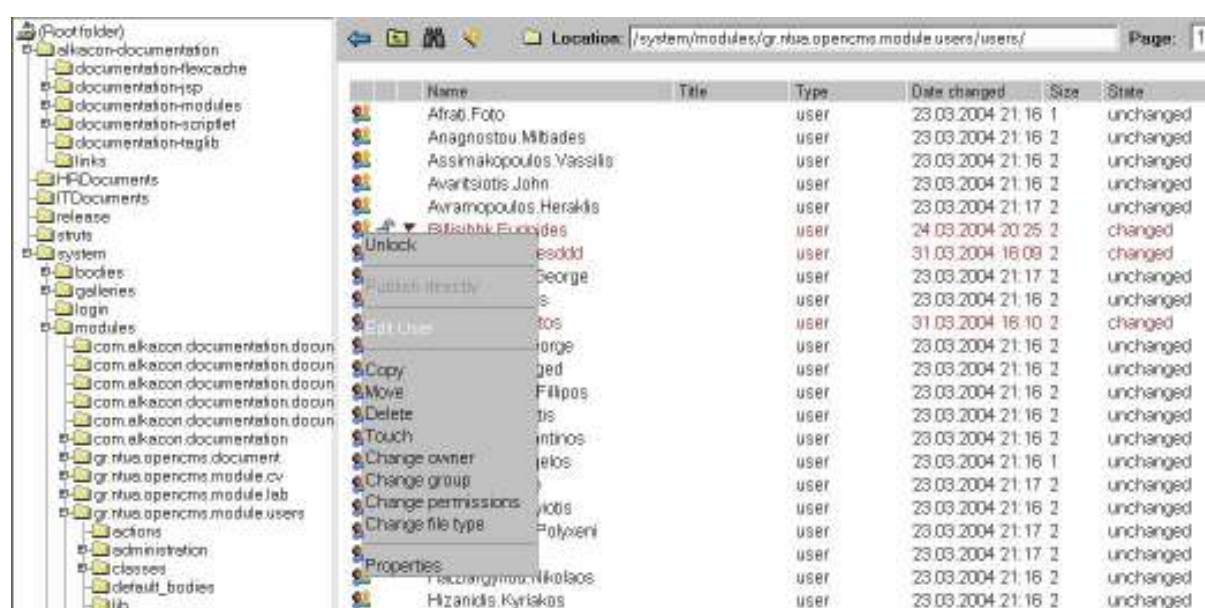
Συνδυάσαμε την λειτουργία για την εισαγωγή νέων τύπων πόρων του συστήματος (`ResourceType`) με αυτή του ορισμού νέου περιεχομένου. Για τον καθορισμό ενός νέου περιεχομένου αρχικά σχεδιάζουμε τους αναγκαίους νέους πίνακες στην βάση. Ένας από τους πίνακες πρέπει να διαθέτει ένα μοναδικό αναγνωριστικό κλειδί ώστε να μπορούμε να αναφερόμαστε και να ανατρέχουμε στις εγγραφές του περιεχομένου με μοναδικό τρόπο. Στην συνέχεια, για κάθε νέο είδος περιεχομένου, ορίζεται μια νέα `NewTypeCmsContentDefinition` κλάση που επεκτείνει την `A_CmsContentDefinition` με εσωτερικές μεταβλητές τα στοιχεία των πινάκων που αντιστοιχούν και χαρακτηρίζουν το συγκεκριμένο περιεχόμενο. Στην `NewTypeCmsContentDefinition` υλοποιούνται μέθοδοι `set/get` για πρόσβαση στα πεδία του. Κάθε `NewTypeCmsContentDefinition` έχει υποχρεωτικά μία μέθοδο `write` ώστε να αποθηκεύει τα περιεχόμενα του στην βάση και έναν κατασκευαστή που δέχεται ως όρισμα το αναγνωριστικό κλειδί του πόρου. Ο κατασκευαστής αναλαμβάνει κατά την αρχικοποίηση να ενημερώνει τα εσωτερικά πεδία του πόρου με τιμές από τους πίνακες της βάσης .

Για να μπορούμε να ενημερώνουμε τα πεδία του περιεχομένου καθορίζουμε φόρμες διαχείρισης της κάθε `NewTypeCmsContentDefinition` μέσα από το περιβάλλον πλοήγησης του OpenCMS. Οι φόρμες διαχείρισης υλοποιούνται σε αρχεία JSP (Java Server Pages) τα οποία αναλαμβάνουν να επικοινωνήσουν με τις μεθόδους `get/set` της `NewTypeCmsContentDefinition` ώστε να ανανεώσουν το περιεχόμενό τους. Όταν ο χρήστης ζητήσει την αποθήκευση των αλλαγών, τα πεδία των πινάκων ανανεώνονται μέσω της `NewTypeCmsContentDefinition` και η

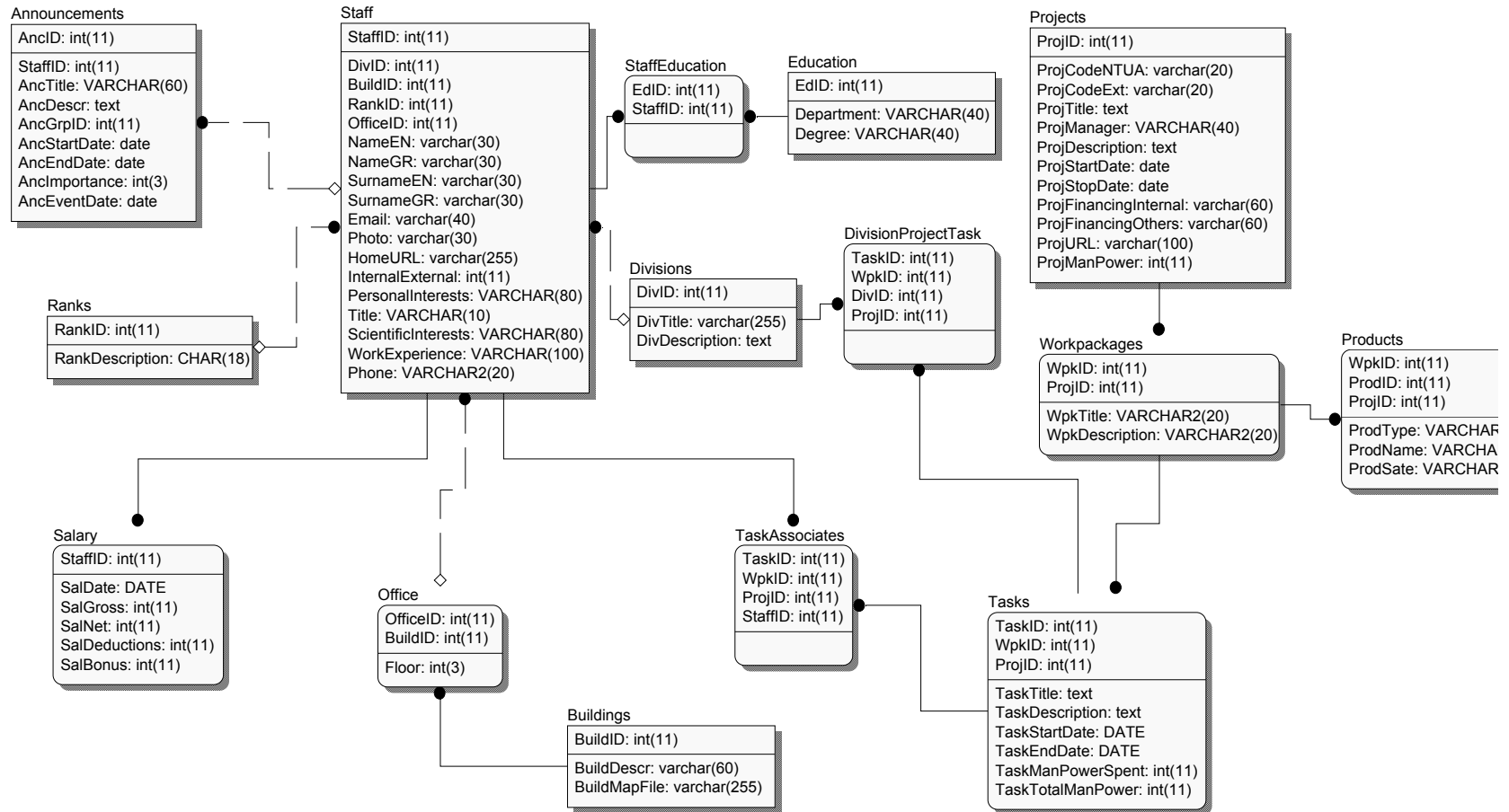
κατάσταση (state) του συγκεκριμένου πόρου τίθεται ως αλλαγμένη. Η φόρμες διαχείρισης μπορούν να ποικίλουν ανάλογα με τις ανάγκες που υπάρχουν (ή μπορούν να προκύψουν) για κάθε νέο περιεχόμενο.

Με την παραπάνω μεθοδολογία το περιβάλλον χρησιμοποιεί το κάθε νέο είδος δομημένου περιεχόμενου ως ένα νέο τύπο του συστήματος. Κάθε τύπος του συστήματος αποθηκεύει μία εγγραφή ως πόρο (resource) στην βάση δεδομένων σε μορφή BLOB. Κατά τον σχεδιασμό επιλέξαμε σε αυτόν το χώρο της βάσης να αποθηκεύεται το μοναδικό αναγνωριστικό που χαρακτηρίζει την συγκεκριμένη εγγραφή. Με αυτόν τον τρόπο το δομημένο περιεχόμενο εμφανίζεται ως ένας νέος τύπος στο περιβάλλον πλοήγησης του OpenCMS, έχει δικές του φόρμες για την αλλαγή του, αλλά κληρονομεί και όλες τις λειτουργίες που έχουν και οι υπόλοιποι τύποι του συστήματος (αλλαγή από διαφορετικούς χρήστες ανάλογα με το δικαίωμα πρόσβασης, διαγραφή, αντιγραφή ή μετακίνηση).

Ακολουθώντας τον παραπάνω μηχανισμό στα πλαίσια της εργασίας, έχουν υλοποιηθεί σε μορφή OpenCMS modules δύο βασικά είδη περιεχόμενου, αυτό του υπαλλήλου μιας εταιρίας και το αυτό των τμημάτων της. Έχει σχεδιαστεί και κατασκευαστεί ένα ολοκληρωμένο module για διαχείριση οργανογράμματος μιας εταιρίας μέσα από την πλατφόρμα του OpenCMS. Στο σημείο αυτό παραθέτουμε αντιπροσωπευτικό δείγμα της διεπαφής με το χρήστη και το διάγραμμα οντοτήτων – συσχετίσεων (E-R) της βάσης για τα modules των υπαλλήλων και των τμημάτων μιας εταιρίας.



Σχήμα 5-1 Ρόλοι και λειτουργίες των Υπηρεσιών Ιστού

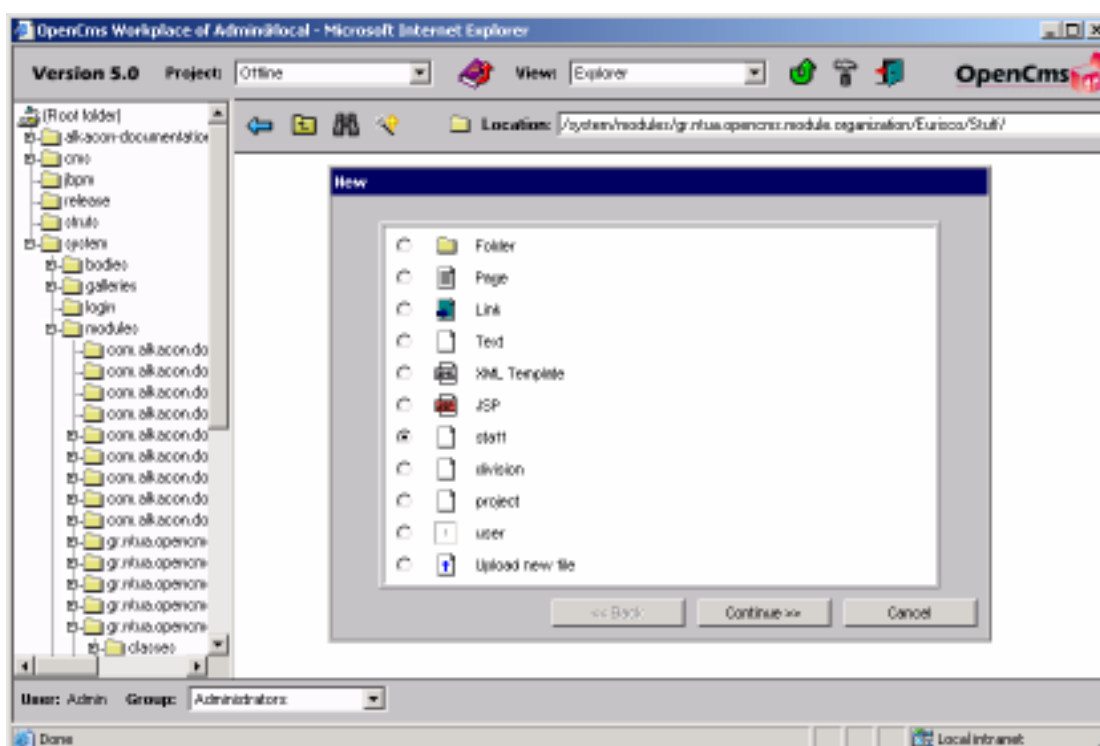


Σχήμα 5-2 E-R διάγραμμα υπαλλήλων / τμημάτων εταιρείας



Στα πλαίσια του έργου, έχει υλοποιηθεί το organization module που δίνει τη δυνατότητα δημιουργίας και διαχείρισης τριών νέων τύπων δομημένου περιεχομένου που αφορούν μια επιχείρηση: των υπαλλήλων (Stuff), των τμημάτων (Divisions) και των έργων (Projects) που έχει αναλάβει η επιχείρηση.

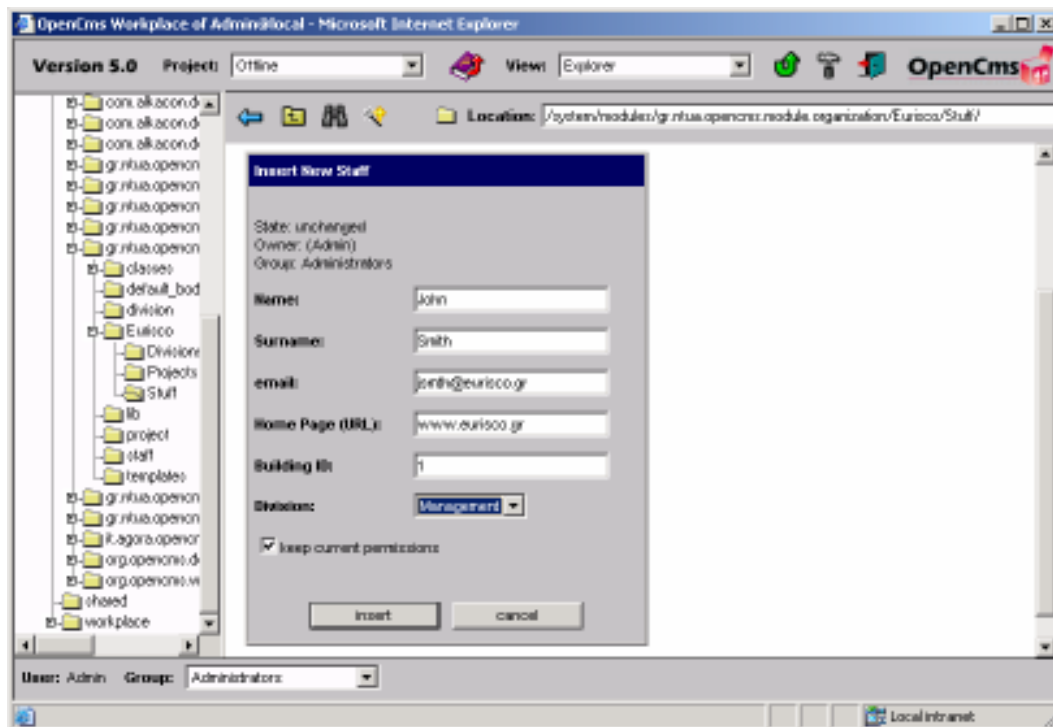
Μέσω του εικονιδίου new βλέπουμε στο αναδυόμενο παράθυρο (Σχήμα 5-3) τους νέους τύπους δομημένου περιεχομένου που παρέχει το organization module. Μέσο του παραθύρου αυτού μπορούμε να εισάγουμε νέο υπάλληλο, ή τμήμα, ή έργο για κάποιον οργανισμό.



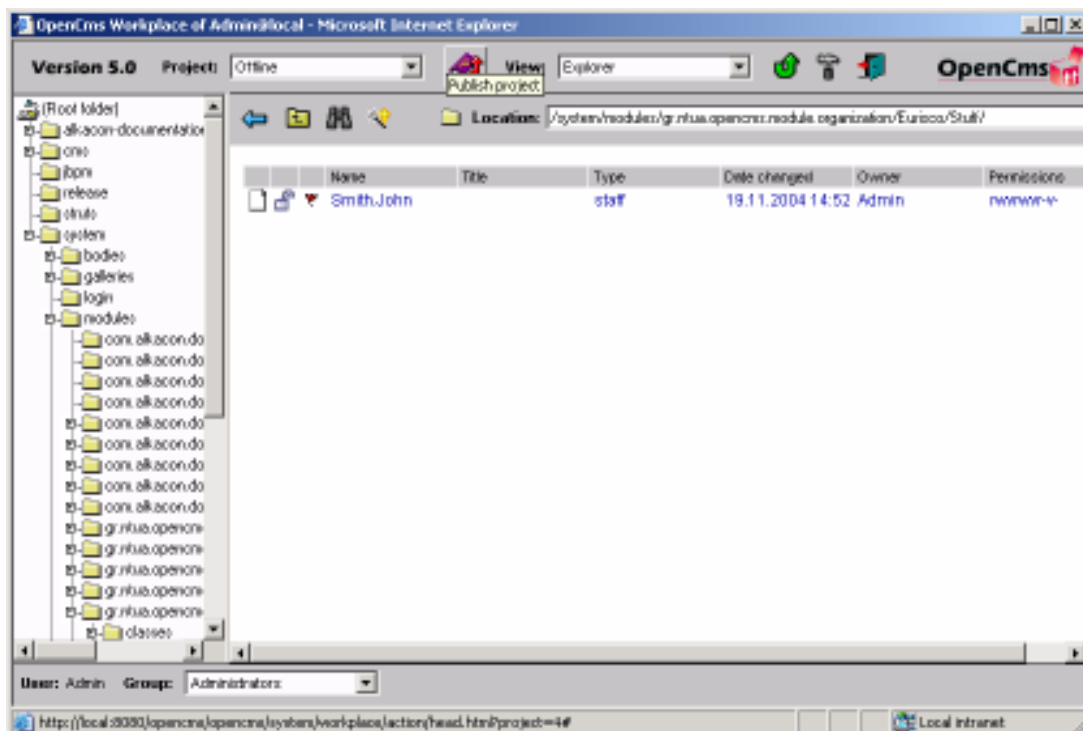
Σχήμα 5-3: Νέοι τύποι δομημένου περιεχομένου που παρέχει το organization module μέσω αναδυόμενου παραθύρου

Στο παρακάτω σχήματα φαίνεται η διαδικασία που ακολουθούμε για την εισαγωγή νέου υπαλλήλου. Παρόμοια είναι και η διαδικασία που ακολουθείται για την εισαγωγή νέου τμήματος ή νέου έργου. Το αναδυόμενο παράθυρο καθορίζει τα χαρακτηριστικά του νέου χρήστη όπως αυτά συμπληρώνονται από τον υπεύθυνο. Έγιναν αρκετές εγγραφές προκειμένου να ελεγχθεί η εγκυρότητα της διαδικασίας και οι επιδόσεις του συστήματος χρησιμοποιώντας διαφορετικά δεδομένα καταχώρησης.

Στο Σχήμα 5-4 και Σχήμα 5-5 παρουσιάζεται ένα παράδειγμα από το σύνολο αυτών που εκτελέστηκαν.



Σχήμα 5-4:: Συμπλήρωση της φόρμας με τα στοιχεία του υπάλληλου



Σχήμα 5-5:: Εμφάνιση του νέου υπάλληλου στον φάκελο που δημιουργήθηκε

Πρέπει να τονιστεί σε αυτό το σημείο ότι ο νέος υπάλληλος που δημιουργήθηκε (Smith John) έχει δικές του φόρμες για την αλλαγή του, αλλά κληρονομεί και όλες τις λειτουργίες που έχουν και οι υπόλοιποι τύποι του συστήματος όπως αλλαγή από διαφορετικούς χρήστες ανάλογα με το δικαίωμα πρόσβασης, δημοσίευση, διαγραφή, αντιγραφή ή μετακίνηση. Επιπλέον μπορεί να συνδεθεί με αντικείμενα άλλων τύπων όπως για παράδειγμα το τμήμα της επιχείρησης στο οποίο ανήκει ο υπάλληλος. Στο παράρτημα Α παρουσιάζεται το **registry.xml** αρχείο του OpenCMS καθώς και το αρχείο **Stuff** του organization module.

### **5.3. Δυναμικό φόρτωμα δομημένου περιεχομένου και προβολή του με την χρήση XSL**

Για να μπορέσουμε να απεικονίσουμε τα περιεχόμενα της εγγραφής σχεδιάσαμε ένα μηχανισμό ανάκτησης τους με μορφή εγγράφων XML. Κάθε νέος τύπος ορίζει ένα διαφορετικό έγγραφο XML το οποίο κατασκευάζεται δυναμικά. Για να μπορέσει αυτό το έγγραφο να απεικονιστεί στον τελικό χρήστη εφαρμόζουμε επάνω του φίλτρα XSL ώστε να του δώσουμε την εμφάνιση που επιθυμούμε. Με αυτό τον τρόπο κάθε νέος τύπος μπορεί να έχει ένα ή περισσότερα XSL φίλτρα για την απεικόνισή του, και μάλιστα μπορούμε να εφαρμόζουμε διαφορετικά φίλτρα XSL σε κάθε εγγραφή αν αυτό κρίνεται απαραίτητο.

Για την εφαρμογή των φίλτρων χρησιμοποιούμε τις βιβλιοθήκες xalan και xerces του Apache Software Foundation. Η οντότητα εκείνη που καλείται να εφαρμόσει τα φίλτρα πάνω στον περιεχόμενο είναι ένα αντικείμενο που υλοποιεί τις διεπαφές *I\_CmsLauncher* και *I\_CmsResourceLoader*. Για κάθε νέο τύπο που ορίζουμε πρέπει να ορισθεί και ένας νέος Loader που να αναλαμβάνει να πάρει το XSL φίλτρο από το σύστημα και να το εφαρμόσει στο δυναμικά κατασκευασμένο έγγραφο XML.

Ακόμα και τα XSL φίλτρα είναι οντότητες διαχειριζόμενες μέσα από την πλατφόρμα του OpenCMS ως διαφορετικοί πόροι του συστήματος.

#### **5.4. Χρήση οδηγών (templates) εμφάνισης στο επίπεδο διαχείρισης περιεχομένου**

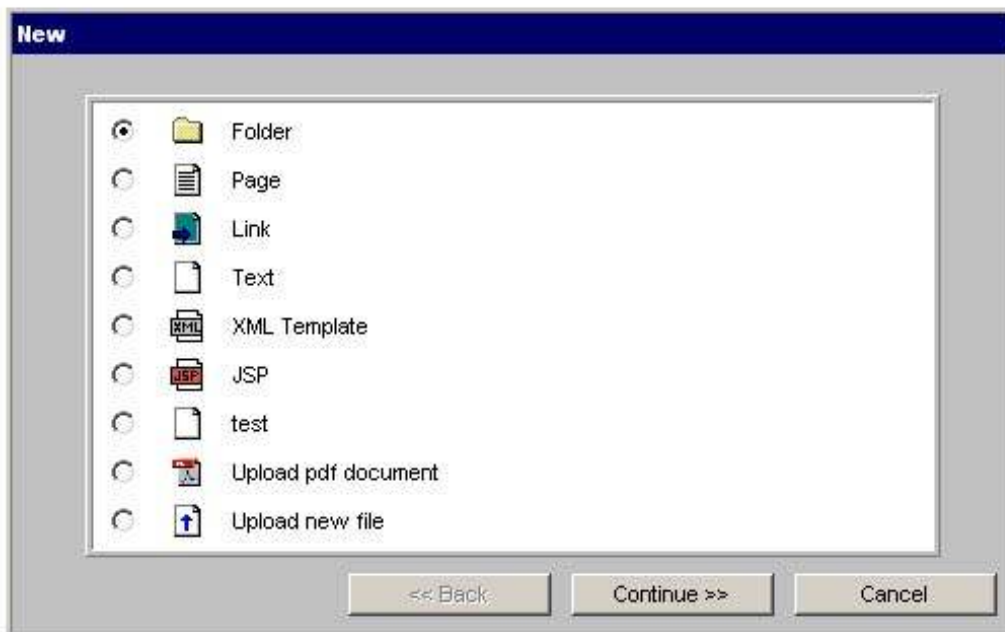
Για την εμφάνιση του περιβάλλοντος διαχείρισης το OpenCMS καθορίζει έναν δικό του μηχανισμό από οδηγούς (templates) με την χρήση XML. Η δομή αυτής της XML έχει καθοριστεί από τους σχεδιαστές του συστήματος και για να χρησιμοποιηθεί πρέπει να υποστηρίζεται από συγκεκριμένα αντικείμενα που αναλαμβάνουν να διαβάσουν συντακτικά (parsing) τα XML αρχεία και ανάλογα με την δομή τους και τις εντολές που περιέχουν να κατασκευάσουν τις σελίδες διαχείρισης.

Ο παραπάνω τρόπος δεν αποτελεί μία αρκετά ευέλικτη λύση μια και αφενός πρέπει κανείς να μάθει μια εντελώς νέα και διαφορετική σύνταξη XML εγγράφων, με ετικέτες και λειτουργίες που ορίζονται μόνο στο OpenCMS χωρίς να ακολουθούν κανένα διεθνές πρότυπο, και αφετέρου για να τις επεκτείνει απαιτείται η συγγραφή κλάσεων που πάλι έχουν μια συγκεκριμένη μεθοδολογία σχεδίασης η οποία περιορίζεται στα όρια της πλατφόρμας. Μάλιστα αξίζει να αναφερθεί ότι από την ίδια την ομάδα σχεδιασμού του OpenCMS έχει εγκαταλειφθεί αυτή τη μέθοδος ανάπτυξης και στην νέα έκδοση βιβλιοθηκών η κατασκευή οδηγών για φόρμες διαχείρισης θα ακολουθεί πρότυπα ανοικτών αρχιτεκτονικών (jsp, TagLibs).

Στα πλαίσια της εργασίας χρησιμοποιήσαμε σελίδες jsp ως οδηγούς οι οποίες αναλαμβάνουν να φορτώσουν άλλες jsp σελίδες για την κάθε μία επιπλέον ενέργεια που θέλει να προσθέσει ο χρήστης. Οι βασικοί οδηγοί, για να είναι εύκολο να επεκταθούν από τους χρήστες του συστήματος, στηρίζονται σε βιβλιοθήκες ετικετών (TagLibs) οι οποίες αναλαμβάνουν να φορτώσουν τα διάφορα στοιχεία.

#### **5.5. Σύστημα διαχείρισης αρχείων PDF**

Για να επιτευχθεί αυτό επεκτείναμε την πλατφόρμα και ορίσαμε ένα νέο τύπο πόρου με την ονομασία pdf. Για τον νέο τύπο πόρου ορίσαμε νέες φόρμες εισαγωγής και ανανέωσης, οι φόρμες δίνουν την δυνατότητα ο χρήστης να μπορεί να εισάγει νέα έγγραφα pdf και να μπορεί να ανανεώνει τα ήδη υπάρχοντα. Οποίες λειτουργίες είχε από πριν το OpenCMS τις κληρονομεί και ο νέος τύπος περιεχομένου, με μόνη εξαίρεση την ενέργεια διόρθωσης ενός pdf εγγράφου, ενέργεια η οποία δεν υλοποιήθηκε διότι κάτι τέτοιο δεν είναι εφικτό.



Σχήμα 5-6 Φόρμα εισαγωγής νέου περιεχομένου (Pdf)

## 5.6. Προσθήκη δυνατότητας εισαγωγής σχολίων σε ένα περιεχόμενο

Για να προσθέσουμε αυτή την δυνατότητα στην πλατφόρμα του OpenCMS υλοποιήσαμε ένα επιπρόσθετο module για σχόλια. Το module αυτό στηρίζει την λειτουργία του σε ένα νέο πίνακα στην βάση δεδομένων όπου αποθηκεύονται τα σχόλια του κάθε χρήστη, το μοναδικό κλειδί της εγγραφής, η ημερομηνία εισαγωγής του σχολίου και το σχόλιο αυτό καθεαυτό.

Για να μπορέσει ένας χρήστης να εισάγει, να διαβάσει και τέλος να σβήσει σχόλια σε μια συγκεκριμένη εγγραφή, σχεδιάσαμε τις απαραίτητες φόρμες για αυτές τις λειτουργίες. Οι φόρμες κατασκευάστηκαν με σελίδες jsp οι οποίες χρησιμοποιούν ένα Java Bean για να υποστηρίζεται η επικοινωνία με την βάση δεδομένων και η ανάκτηση της πληροφορίας με συγκεκριμένες αναζητήσεις.



Σχήμα 5-7 Φόρμες εισαγωγής σχολίων

 The image shows a web interface with a breadcrumb trail: 'Location: /HRDocuments/'. Below it, the text 'Comments about /HRDocuments/DesignMobileCDN.pdf' is displayed. A table lists the comments:
 

User	Comment	Date
Admin	This document is not ready	Wed Apr 21 13:34:09 GMT+02:00 2004 d
hr1	Comment	Wed Mar 31 16:02:50 GMT+02:00 2004 d

Σχήμα 5-8 Φόρμα εμφάνισης σχολίων

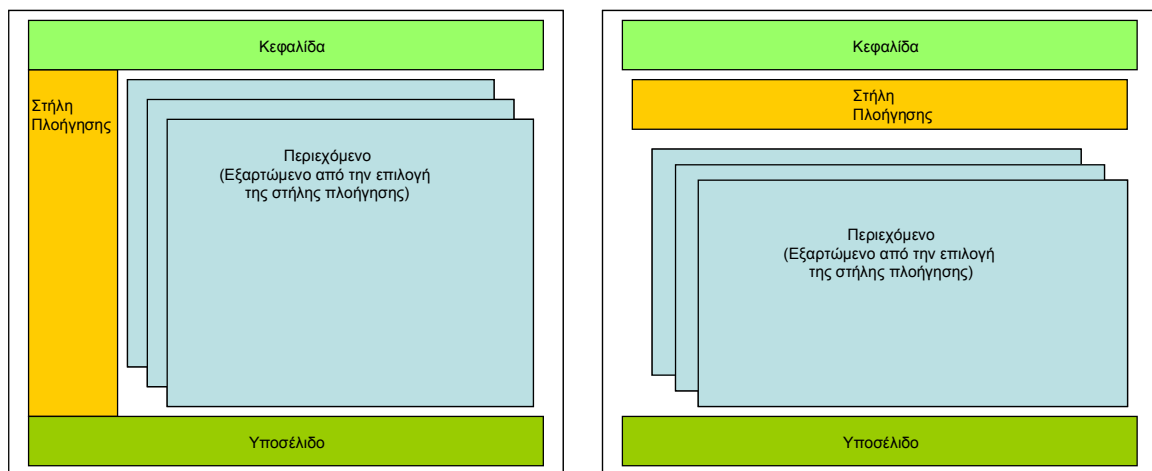
Η δυνατότητα εισαγωγής σχολίων σχεδιάστηκε με τέτοιο τρόπο ώστε να μπορεί να χρησιμοποιηθεί ως επέκταση ακόμα και στα υπάρχοντα βασικά ήδη περιεχομένου με τα οποία διανέμεται η πλατφόρμα OpenCMS.

Στο module έχει σχεδιαστεί η εισαγωγή ενός περιβάλλοντος διαχείρισης ώστε να μπορεί ο διαχειριστής να επιλέγει σε ποιους από τους προϋπάρχοντες τύπους θα μπορεί να ενσωματωθεί αυτή η υπηρεσία.

## 5.7. Πρότυπα (Templates)

Μια βασική λειτουργία που προσδίνει μεγαλύτερη ευελιξία στον σχεδιασμό και ανάπτυξη ενός δικτυακού τόπου που υποστηρίζεται από συστήματα διαχείρισης περιεχομένου είναι η ύπαρξη προτύπων (templates) εμφάνισης και λειτουργιών.

Για την ταχύτερη κατασκευή ενός δικτυακού τόπου που υποστηρίζεται από την πλατφόρμα μας δημιουργήθηκαν δύο βασικά πρότυπα εμφάνισης. Τα πρότυπα αυτά υποστηρίζονται από την τεχνολογία Java JSP και επιτρέπουν την δυναμική εισαγωγή νέου περιεχομένου σε ένα δικτυακό τόπο, χωρίς ο χρήστης να λαμβάνει υπόψη του τον τρόπο εμφάνισης. Τα δύο πρότυπα αυτά στηρίζονται στην εμφάνιση μίας στήλης πλοήγησης (navigation bar) η οποία κατασκευάζεται δυναμικά και επιτρέπει στον διαχειριστή του δικτυακού τόπου να εμφανίζει ή όχι περιεχόμενο που διαχειρίζεται το σύστημα διαχείρισης. Το ένα πρότυπο ακολουθεί την λογική της εμφάνισης οριζόντιας στήλης πλοήγησης και το άλλο κάθετης. Όπως φαίνεται και στο παρακάτω σχήμα τα πρότυπα εμφάνισης έχουν σχεδιαστεί έτσι ώστε να μπορούν να ενσωματώνουν διαφορετικές κεφαλίδες και υποσέλιδα, τα οποία προέρχονται από ξεχωριστά αρχεία.



Σχήμα 5-9 Μοντέλο σχεδιασμού προτύπων

Το αποτέλεσμα είναι ο διαχειριστής ενός δικτυακού τόπου να μπορεί να επιβάλει σε ένα σύνολο από στοιχεία περιεχομένου τον τρόπο εμφάνισής τους ανεξάρτητα.

Παράλληλα η κεφαλίδα και το υποσέλιδο ενσωματώνονται στην τελική εμφάνιση από διαφορετικά αρχεία, πράγμα που καθιστά ακόμα πιο ευέλικτα τα πρότυπα.

Πέρα από την δυνατότητα αυτόματης επιβολής της εμφάνισης, τα συγκεκριμένα πρότυπα υποστηρίζουν και την λειτουργία του ελέγχου πρόσβασης του χρήστη σε συγκεκριμένο περιεχόμενο. Αν ο χρήστης που προσπελαίνει τον δικτυακό τόπο είναι ένας απλός επισκέπτης λαμβάνει το περιεχόμενο που προορίζεται για όλους τους χρήστες. Στην περίπτωση όμως που διαθέτει λογαριασμό πρόσβασης σε εσωτερικές πληροφορίες του δικτυακού τόπου, μπορεί να επιλέξει να πιστοποιηθεί και τότε έχει δυνατότητα πρόσβασης σε περιεχόμενο που χαρακτηρίζεται (από τον διαχειριστή) ιδιωτικό. Με αυτό τον τρόπο μπορεί ένας υπάλληλος της εταιρίας να αποκτήσει πρόσβαση στον εσωτερικό δίκτυο πληροφοριών (intranet), αλλά ακόμα περισσότερο μπορεί ένας δικτυακός τόπος να διαχωρίζει τους χρήστες του και να τους παρέχει διαφορετικό περιεχόμενο ανάλογα με τον λογαριασμό πρόσβασης που διαθέτουν.

Στο Παράρτημα Α φαίνεται η βασική έκδοση του προτύπου για κάθετη γραμμή πλοήγησης (**main\_vertical.jsp**) και ακολουθούν τα αρχεία που ενσωματώνει



## 6. Αναφορές

- [1] Content Management Systems, Phil Suh, Dave Addey, David Thiemecke, James Ellis
- [2] <http://www.OpenCMS.org/en/index.html>.
- [3] <http://www.mysql.com/>
- [4] <http://www.oracle.com/index.html>
- [5] Database Programming with JDBC & Java, Second Edition, George Reese, O'Reilly
- [6] www.w3schools.com tutorials για JavaScript, DOM, XSLT.
- [7] <http://java.sun.com>
- [8] Core Java 2, Volume I Fundamentals Cay Horstman, Gary Cornell, Prentice Hall, the Sun Microsystems Press, Java Series.
- [9] <http://jakarta.apache.org>
- [10] [www.w3.org/xml](http://www.w3.org/xml) τεχνολογίες XML
- [11] Beginning XML, David Hunter, Wrox
- [12] Java & XML, Brett Mc Laughlin, O'Reilly
- [13] Beginning Java Web Services, H.Bequet, M.M. Kunnumpurath, S. Rhody, A. Tost, Wrox (**XML, SOAP, WSDL, UDDI**)
- [14] «Αρχιτεκτονική πλατφόρμας υπηρεσιών ιστού SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging», Συλλογικό έργο, Curbera Francisco, Ferguson Donald F., Leymann Frank, Storey Tony, Weerawarana Sanjiva, εκδόσεις Κλειδάριθμος
- [15] Java and XSLT, Eric Burke, O'Reilly
- [16] XSLT, Doug Tidwell, O'Reilly

## 7. Παράρτημα Α – Κώδικας

### Αρχείο registry.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<registry>
  <system>
    <restype>
      <name>folder</name>
      <launcherType>0</launcherType>
      <launcherClass/>

<resourceClass>com.opencms.file.CmsResourceTypeFolder</resourceClass>
    </restype>
    <restype>
      <name>page</name>
      <launcherType>3</launcherType>
      <launcherClass/>

<resourceClass>com.opencms.file.CmsResourceTypePage</resourceClass>
    </restype>
    <restype>
      <name>link</name>
      <launcherType>4</launcherType>
      <launcherClass/>

<resourceClass>com.opencms.file.CmsResourceTypeLink</resourceClass>
    </restype>
    <restype>
      <name>plain</name>
      <launcherType>1</launcherType>
      <launcherClass/>

<resourceClass>com.opencms.file.CmsResourceTypePlain</resourceClass>
    </restype>
    <restype>
      <name>XMLTemplate</name>
      <launcherType>3</launcherType>
      <launcherClass/>

<resourceClass>com.opencms.file.CmsResourceTypeXMLTemplate</resourceClass>
    </restype>
    <restype>
      <name>binary</name>
      <launcherType>1</launcherType>
      <launcherClass/>

<resourceClass>com.opencms.file.CmsResourceTypeBinary</resourceClass>
```

```

        </restype>
        <restype>
            <name>image</name>
            <launcherType>1</launcherType>
            <launcherClass/>

<resourceClass>com.opencms.file.CmsResourceTypeImage</resource
Class>
        </restype>
        <restype>
            <name>compatiblePlain</name>
            <launcherType>1</launcherType>
            <launcherClass/>

<resourceClass>com.opencms.file.CmsResourceTypeCompatiblePlain
</resourceClass>
        </restype>
        <restype>
            <name>jsp</name>
            <launcherType>6</launcherType>
            <launcherClass/>

<resourceClass>com.opencms.file.CmsResourceTypeJsp</resourceCl
ass>
        </restype>
        <restype>
            <name>staff</name>
            <launcherType>1</launcherType>
            <launcherClass/>

<resourceClass>com.opencms.file.CmsResourceTypeStaff</resource
Class>
        </restype>
        <restype>
            <name>division</name>
            <launcherType>1</launcherType>
            <launcherClass/>

<resourceClass>com.opencms.file.CmsResourceTypeDivision</resou
rceClass>
        </restype>
        <restype>
            <name>project</name>
            <launcherType>1</launcherType>
            <launcherClass/>

<resourceClass>com.opencms.file.CmsResourceTypeProject</resour
ceClass>
        </restype>
        <restype>
            <name>user</name>
            <launcherType>69</launcherType>

<launcherClass>gr.ntua.opencms.module.users.UserLoader</launch
erClass>

```

```

<resourceClass>com.opencms.file.CmsResourceTypeUser</resourceC
lass>
    </restype>
    <view>
        <name>select.explorer</name>

<url>/system/workplace/action/explorer_new.html</url>
    </view>
    <view>
        <name>select.admin</name>

<url>/system/workplace/action/administration.html</url>
    </view>
    <view>
        <name>select.tasks</name>
        <url>/system/workplace/action/tasks.html</url>
    </view>
    <exportpoint>
        <source>/system/workplace/resources/</source>
        <destination>resources/</destination>
    </exportpoint>
    <exportpoint>
        <source>/system/modules/default/classes/</source>
        <destination>WEB-INF/classes/</destination>
    </exportpoint>
    <smtpserver>my.smtp.server</smtpserver>
    <smtpserver2>alternative.smtp.server</smtpserver2>

<defaultmailsender>nobody@nowhere.com</defaultmailsender>
    <checklink>
        <mailfrom>nobody@nowhere.com</mailfrom>
        <mailto>nobody@nowhere.com</mailto>

<mailtemplate>/system/workplace/templates/urlcheckform</mailte
mplate>
    </checklink>
    <launchers>

<launcher1>com.opencms.flex.CmsXmlTemplateLoader</launcher1>

<launcher2>com.opencms.flex.CmsDumpLoader</launcher2>

<launcher3>com.opencms.launcher.CmsLinkLauncher</launcher3>

<launcher4>com.opencms.flex.CmsJspLoader</launcher4>

<launcher5>gr.ntua.opencms.module.users.UserLoader</launcher5>
    </launchers>
    <UseWpPicturesFromVFS>>false</UseWpPicturesFromVFS>
    <searchbylucene>off</searchbylucene>
    <history>
        <enabled>>true</enabled>
        <weeks/>
        <deleteversions>>false</deleteversions>
    </history>

```

```

    <tempfileproject>3</tempfileproject>
  </system>
  <modules>
    <module>
      <type>simple</type>
      <name>org.opencms.default</name>
      <nickname>OpenCms 5.0 Default module</nickname>
      <version>3.0</version>
      <description><![CDATA[This is the traditional
"default" module. It contains the "empty" body and the "empty"
template. In case you import content from an older OpenCms
version, the templates from the 4.x "/content" directory will
be stored in this module.

(c) 2003 by Alkacon Software (http://www.alkacon.com).
]]></description>
      <author>Alkacon Software</author>
      <email>info@alkacon.com</email>
      <creationdate>04.23.2003</creationdate>
      <exportpoint>

<source>/system/modules/org.opencms.default/classes/</source>
      <destination>WEB-INF/classes/</destination>
    </exportpoint>
    <exportpoint>

<source>/system/modules/org.opencms.default/lib/</source>
      <destination>WEB-INF/lib/</destination>
    </exportpoint>
    <view/>
    <publishclass>
      <name/>
    </publishclass>
    <documentation/>
    <dependencies/>
    <maintenance_class/>
    <parameters/>
    <repository/>
    <uploaddate>05.20.2004</uploaddate>
    <uploadedby>Admin</uploadedby>
  </module>
  <module>
    <type>simple</type>
    <name>org.opencms.welcome</name>
    <nickname>OpenCms 5.0 Welcome page and release
notes</nickname>
    <version>3.0</version>
    <description><![CDATA[This module contains the
OpenCms welcome page and release notes. Deleting this module
will remove the "/release" directory and all pages there. It
will also remove the "default" template.

(c) 2003 by Alkacon Software
(http://www.alkacon.com).]]></description>
    <author>Alkacon Software</author>
    <email>info@alkacon.com</email>

```

```

    <creationdate>04.23.2003</creationdate>
    <view/>
    <publishclass>
      <name/>
    </publishclass>
    <documentation/>
    <dependencies/>
    <maintenance_class/>
    <parameters>
      <para>
        <name>additionalresources</name>
        <description><![CDATA[Additional resources
that belong to this module. Please note: If you delete this
module, all resources listed here will be deleted (even if
they are directories).]]></description>
        <type>string</type>
        <value>/release;/index.jsp</value>
      </para>
    </parameters>
    <repository/>
    <uploaddate>05.20.2004</uploaddate>
    <uploadedby>Admin</uploadedby>
  </module>
  <module>
    <type>simple</type>
    <name>gr.ntua.opencms.module.lab</name>
    <nickname>ntua lab</nickname>
    <version>1.0</version>
    <description><![CDATA[ ntua lab]]></description>
    <author>AM</author>
    <email/>
    <creationdate>02.24.2004</creationdate>
    <exportpoint>

<source>/system/modules/gr.ntua.opencms.module.lab/classes/</s
ource>
      <destination>WEB-INF/classes/</destination>
    </exportpoint>
    <exportpoint>

<source>/system/modules/gr.ntua.opencms.module.lab/lib/</sourc
e>
      <destination>WEB-INF/lib/</destination>
    </exportpoint>
    <view/>
    <publishclass>
      <name/>
    </publishclass>
    <documentation/>
    <dependencies/>
    <maintenance_class/>
    <parameters/>
    <repository/>
    <uploaddate>05.20.2004</uploaddate>
    <uploadedby>Admin</uploadedby>
  </module>

```

```

    <module>
      <type>simple</type>
      <name>com.alkacon.documentation</name>
      <nickname>OpenCms 5.0 Documentation
base</nickname>
      <version>1.0</version>
      <description><![CDATA[A set of templates and JSP
elements for the OpenCms interactive documentation and
examples.

(c) 2003 by Alkacon Software
(http://www.alkacon.com).]]></description>
      <author>Alkacon Software</author>
      <email>info@alkacon.com</email>
      <creationdate>04.23.2003</creationdate>
      <exportpoint>

<source>/system/modules/com.alkacon.documentation/classes/</so
urce>
      <destination>WEB-INF/classes/</destination>
    </exportpoint>
    <exportpoint>

<source>/system/modules/com.alkacon.documentation/lib/</source
>
      <destination>WEB-INF/lib/</destination>
    </exportpoint>
    <view/>
    <publishclass>
      <name/>
    </publishclass>
    <documentation/>
    <dependencies/>
    <maintenance_class/>
    <parameters>
      <para>
        <name>additionalresources</name>
        <description><![CDATA[Additional resources
that belong to this module. Please note: If you delete this
module, all resources listed here will be deleted (even if
they are directories).]]></description>
        <type>string</type>
        <value>/alkacon-documentation/index.html;-
;/alkacon-documentation/showsource.jsp;-;/alkacon-
documentation/index_content.html;-;/alkacon-
documentation/license.html;-;/alkacon-
documentation/contribute.html;-;/alkacon-
documentation/links/;-;/alkacon-
documentation/alkacon_services.html;-;/alkacon-
documentation/sponsoring.html;-
;/system/galleries/pics/alkacon-documentation/</value>
      </para>
    </parameters>
    <repository/>
    <uploaddate>05.20.2004</uploaddate>
    <uploadedby>Admin</uploadedby>

```

```

    </module>
    <module>
      <type>simple</type>
      <name>gr.ntua.opencms.module.organization</name>

    <nickname>gr.ntua.opencms.module.organization</nickname>
      <version>1.0</version>
      <description><![CDATA[ ]]></description>
      <author/>
      <email/>
      <creationdate>05.20.2004</creationdate>
      <exportpoint>

    <source>/system/modules/gr.ntua.opencms.module.organization/cl
asses/</source>
      <destination>WEB-INF/classes/</destination>
    </exportpoint>
    <exportpoint>

    <source>/system/modules/gr.ntua.opencms.module.organization/li
b/</source>
      <destination>WEB-INF/lib/</destination>
    </exportpoint>
    <view/>
    <publishclass>
      <name/>
    </publishclass>
    <documentation/>
    <dependencies/>
    <maintenance_class/>
    <parameters/>
    <repository/>
  </module>
  <module>
    <type>simple</type>
    <name>com.alkacon.documentation.documentation-
jsp</name>
    <nickname>OpenCms 5.0 JSP basic
documentation</nickname>
    <version>1.0</version>
    <description><![CDATA[The OpenCms 5.0 JSP
integration basic documentation and examples.

(c) 2003 by Alkacon Software
(http://www.alkacon.com).]]></description>
    <author>Alkacon Software</author>
    <email>info@alkacon.com</email>
    <creationdate>04.23.2003</creationdate>
    <view/>
    <publishclass>
      <name/>
    </publishclass>
    <documentation/>
    <dependencies>
      <dependency>
        <name>com.alkacon.documentation</name>

```



```

        <minversion>1.0</minversion>
        <maxversion>-1.0</maxversion>
    </dependency>
</dependencies>
<maintenance_class/>
<parameters>
    <para>
        <name>additionalresources</name>
        <description><![CDATA[Additional resources
that belong to this module. Please note: If you delete this
module, all resources listed here will be deleted (even if
they are directories).]]></description>
        <type>string</type>
        <value>/alkacon-
documentation/documentation-jsp</value>
    </para>
</parameters>
<repository/>
<uploaddate>05.26.2004</uploaddate>
<uploadedby>Admin</uploadedby>
</module>
<module>
    <type>simple</type>
    <name>com.alkacon.documentation.documentation-
modules</name>
    <nickname>OpenCms 5.0 Module management
documentation</nickname>
    <version>1.0</version>
    <description><![CDATA[The OpenCms 5.0 module
management documentation.
(c) 2003 by Alkacon Software
(http://www.alkacon.com).]]></description>
    <author>Alkacon Software</author>
    <email>info@alkacon.com</email>
    <creationdate>04.23.2003</creationdate>
    <view/>
    <publishclass>
        <name/>
    </publishclass>
    <documentation/>
    <dependencies>
        <dependency>
            <name>com.alkacon.documentation</name>
            <minversion>1.0</minversion>
            <maxversion>-1.0</maxversion>
        </dependency>
    </dependencies>
    <maintenance_class/>
    <parameters>
        <para>
            <name>additionalresources</name>
            <description><![CDATA[Additional resources
that belong to this module. Please note: If you delete this
module, all resources listed here will be deleted (even if
they are directories).]]></description>
            <type>string</type>

```

```

                <value>/alkacon-
documentation/documentation-scriptlet/</value>
            </para>
        </parameters>
        <repository/>
        <uploaddate>05.26.2004</uploaddate>
        <uploadedby>Admin</uploadedby>
    </module>
    <module>
        <type>simple</type>
        <name>com.alkacon.documentation.documentation-
taglib</name>
        <nickname>OpenCms 5.0 JSP taglib
documentation</nickname>
        <version>1.0</version>
        <description><![CDATA[The OpenCms 5.0 <cms:>
taglib documentation and examples.
(c) 2003 by Alkacon Software (http://www.alkacon.com).
]]></description>
        <author>Alkacon Software</author>
        <email>info@alkacon.com</email>
        <creationdate>04.23.2003</creationdate>
        <exportpoint>

<source>/system/modules/com.alkacon.documentation.examples-
jstl/lib/</source>
        <destination>WEB-INF/lib/</destination>
    </exportpoint>
    <view/>
    <publishclass>
        <name/>
    </publishclass>
    <documentation/>
    <dependencies>
        <dependency>
            <name>com.alkacon.documentation</name>
            <minversion>1.0</minversion>
            <maxversion>-1.0</maxversion>
        </dependency>
    </dependencies>
    <maintenance_class/>
    <parameters>
        <para>
            <name>additionalresources</name>
            <description><![CDATA[Additional resources
that belong to this module. Please note: If you delete this
module, all resources listed here will be deleted (even if
they are directories).]]></description>
            <type>string</type>
            <value>/alkacon-documentation/examples-
jstl/</value>
        </para>
    </parameters>
    <repository/>
    <uploaddate>05.26.2004</uploaddate>
    <uploadedby>Admin</uploadedby>

```

```

        </module>
        <module>
            <type>simple</type>
            <name>com.alkacon.documentation.howto-
template</name>
            <nickname>OpenCms 5.0 JSP template
howto</nickname>
            <version>1.0</version>
            <description><![CDATA[The OpenCms 5.0 simple JSP
template development documentation.

<source>/system/modules/it.agora.opencms.module.vtab/lib/</sou
rce>
                <destination>WEB-INF/lib/</destination>
            </exportpoint>
            <exportpoint>

<source>/system/modules/gr.ntua.opencms.module.jbpm/lib/</sour
ce>
                <destination>WEB-INF/lib/</destination>
            </exportpoint>
            <exportpoint>

<source>/system/modules/gr.ntua.opencms.module.jbpm/classes/</
source>
                <destination>WEB-INF/classes/</destination>
            </exportpoint>
            <view/>
            <publishclass>
                <name/>
            </publishclass>
            <documentation/>
            <dependencies/>
            <maintenance_class/>
            <parameters>
                <para>
                    <name>additionalresources</name>
                    <description><![CDATA[A folder with simple
test jsp pages]]></description>
                    <type>string</type>
                    <value>/jbpm/</value>
                </para>
            </parameters>
            <repository/>
            <uploaddate>07.08.2004</uploaddate>
            <uploadedby>Admin</uploadedby>
        </module>
        <module>
            <type>simple</type>
            <name>gr.ntua.opencms.module.struts</name>
            <nickname>Struts in Opencms</nickname>
            <version>1.0</version>
            <description><![CDATA[
]]></description>
            <author/>
            <email/>

```

```

        <creationdate>06.08.2004</creationdate>
        <exportpoint>

<source>/system/modules/gr.ntua.opencms.module.users/lib/</source>
        <destination>WEB-INF/lib/</destination>
    </exportpoint>
    <exportpoint>

<source>/system/modules/gr.ntua.opencms.module.users/classes/</source>
        <destination>WEB-INF/classes/</destination>
    </exportpoint>
    <view/>
    <publishclass>

<source>/system/modules/gr.ntua.opencms.module.cms_site/classes/</source>
        <destination>WEB-INF/classes/</destination>
    </exportpoint>
    <exportpoint>

<source>/system/modules/gr.ntua.opencms.module.cms_site/lib/</source>
        <destination>WEB-INF/lib/</destination>
    </exportpoint>
    <view/>
    <publishclass>
        <name/>
    </publishclass>
    <documentation/>
    <dependencies>
        <dependency>
            <name>gr.ntua.opencms.module.jbpm</name>
            <minversion>1.0</minversion>
            <maxversion>-1.0</maxversion>
        </dependency>
        <dependency>
            <name>gr.ntua.opencms.module.struts</name>
            <minversion>1.0</minversion>
            <maxversion>-1.0</maxversion>
        </dependency>
    </dependencies>
    <maintenance_class/>
    <parameters>
        <para>
            <name>additionalresources</name>
            <description/>
            <type>string</type>
            <value>/cms/</value>
        </para>
    </parameters>
    <repository/>
    <uploaddate>10.01.2004</uploaddate>
    <uploadedby>Admin</uploadedby>
</module>

```

```
</modules>
</registry>
```

## Αρχαίο Stuff

```
vi.resource[resource_id]=new
res("staff","language_key(fileicon.staff)",vi.iconPath+"ic_file_plain.gif",
"/opencms/opencms/system/modules/gr.ntua.opencms.module.organization/newStaff.jsp?init=true&type=staff",true);

addMenuEntry(resource_id,
"language_key(explorer.context.lock)",
"lock.html?initial=true", "", "rules_key(d d
aaai dddd dddd dddd dddd)");
addMenuEntry(resource_id,
"language_key(explorer.context.override.lock)",
"lockchange.html?initial=true", "", "rules_key(d d
dddd dddd aaai aaai aaai)");
addMenuEntry(resource_id,
"language_key(explorer.context.unlock)",
"unlock.html?initial=true", "", "rules_key(d d
dddd aaai dddd dddd dddd)");
addMenuEntry(resource_id, "-",
" ", "", "rules_key(d d
aaaa aaaa aaaa aaaa aaaa)");

addMenuEntry(resource_id,
"language_key(explorer.context.project)",
"resource_to_project.html?initial=true", "", "rules_key(a d
dddd dddd dddd dddd dddd)");
addMenuEntry(resource_id,
"language_key(explorer.context.copyp_topproject)",
"copy_to_project.html?initial=true", "", "rules_key(d a
dddd dddd dddd dddd dddd)");
addMenuEntry(resource_id,
"language_key(explorer.context.publish)",
"publish_resource.html?initial=true", "", "rules_key(d d
iaaa iiii dddd dddd dddd)");
addMenuEntry(resource_id, "-",
" ", "", "rules_key(a a
aaaa aaaa dddd dddd dddd)");

//addMenuEntry(resource_id,
"language_key(explorer.context.htmledit)",
"edit_html.html?init=true", "'_top'", "rules_key(d
d iiii aaai dddd dddd dddd)");
//addMenuEntry(resource_id,
"language_key(explorer.context.edit)",
"/opencms/opencms/system/modules/gr.ntua.opencms.module.organi
zation/editStaff.jsp?init=true", "", "rules_key(d
d iiii aaai dddd dddd dddd)");
addMenuEntry(resource_id, "Edit Staff",
"/opencms/opencms/system/modules/gr.ntua.opencms.module.organi
```

```

zation/editStaff.jsp?init=true",          """, "rules_key(d
d iiii aaai dddd dddd dddd)");

    addMenuEntry(resource_id, "-",
" ",          "", "rules_key(d d
aaaa aaaa dddd dddd dddd)");

    addMenuEntry(resource_id,
"language_key(explorer.context.copy)",
"copy.html?initial=true",          "", "rules_key(d d
aaai aaai dddd dddd dddd)");
    addMenuEntry(resource_id,
"language_key(explorer.context.rename)",
"rename.html?initial=true",          "", "rules_key(d d
iiii aaai dddd dddd dddd)");
    addMenuEntry(resource_id,
"language_key(explorer.context.move)",
"move.html?initial=true",          "", "rules_key(d d
iiii aaai dddd dddd dddd)");
    //addMenuEntry(resource_id,
"language_key(explorer.context.delete)",
"delete.html?initial=true",          "", "rules_key(d d
iiii aaai dddd dddd dddd)");
    addMenuEntry(resource_id,
"language_key(explorer.context.delete)",
"/opencms/opencms/system/modules/gr.ntua.opencms.module.organi
zation/deleteStaff.jsp?init=true",          "",
"rules_key(d d iiii aaai dddd dddd dddd)");

    addMenuEntry(resource_id,
"language_key(explorer.context.touch)",
"touch.html?initial=true",          "", "rules_key(d d
iiii aaai dddd dddd dddd)");
    addMenuEntry(resource_id,
"language_key(explorer.context.undochanges)",
"undochanges.html?initial=true",          "", "rules_key(d d
iiid iaia dddd dddd dddd)");
    addMenuEntry(resource_id,
"language_key(explorer.context.undelete)",
"undelete.html?initial=true",          "", "rules_key(d d
ddda dddi dddd dddd dddd)");
    addMenuEntry(resource_id, "-",
" ",          "", "rules_key(d d
aaaa aaaa dddd dddd dddd)");

    addMenuEntry(resource_id,
"language_key(explorer.context.showvfslinks)",
"explorer_files.html?initial=true&mode=vfslink",          "",
"rules_key(d d aaai aaai dddd dddd dddd)");
    addMenuEntry(resource_id, "-",
" ",          "", "rules_key(d d
aaaa aaaa dddd dddd dddd)");

    addMenuEntry(resource_id,
"language_key(explorer.context.chown)",

```

```

"chown.html?initial=true",          "",      "rules_key(d d
iiii aaai dddd dddd dddd)");
    addMenuEntry(resource_id,
"language_key(explorer.context.chgrp)",
"chgrp.html?initial=true",          "",      "rules_key(d d
iiii aaai dddd dddd dddd)");
    addMenuEntry(resource_id,
"language_key(explorer.context.access)",
"chmod.html?initial=true",          "",      "rules_key(d d
iiii aaai dddd dddd dddd)");
    addMenuEntry(resource_id,
"language_key(explorer.context.type)",
"chtype.html?initial=true",          "",      "rules_key(d d
iiii aaai dddd dddd dddd)");
    addMenuEntry(resource_id, "-",          "",      "rules_key(d d
aaaa aaaa dddd dddd dddd)");

    addMenuEntry(resource_id,
"language_key(explorer.context.history)",
"history.html?initial=true",          "",      "rules_key(a a
aaaa aaaa aaaa aaaa aaaa)");
    addMenuEntry(resource_id,
"language_key(explorer.context.property)",
"property.html?initial=true",          "",      "rules_key(a a
aaai aaai aaai aaai aaai)");

```

## Αρχείο main\_vertical.jsp

```

<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="http://www.OpenCMS.org/taglib/cms" prefix="cms" %>
<%@ page contentType="text/html; charset=ISO-8859-7" %>
<%@ page import="com.OpenCMS.flex.jsp.*" %>
<%
com.OpenCMS.flex.jsp.CmsJspActionElement cms =
    new CmsJspActionElement(pageContext, request, response);
%>

<cms:template element="head">
<html>
<head>

```

```

<title>Test Site - <cms:property name="title" escapeHtml="true"
/></title>
<link title="Stylesheet"
      href = "<cms:link>../resources/css/alter.css</cms:link>"
      type="text/css"
      rel="STYLESHEET">
</head>

<body>
<table border="0" width="800" height="600" id="small">
<tr height="50">
<td valign="top" height="20" colspan="2" id="head">
      <cms:include file="header.jsp" />
</td>
</tr>
<tr>
<td valign="top" width="50">
      <cms:include file="vertical_nav.jsp" />
</td>
<td valign="top" width="700">
</cms:template>

<cms:template element="body">
      <cms:include element="body" />
</cms:template>

<cms:template element="foot">
</td>
</tr>
<tr height="30">
<td align="right" colspan="2" valign="bottom" width="100%"
id="foot">
<cms:include file="footer.jsp" />
</td>
</tr>
</table>
</body>
</html>
</cms:template>

```

### **Αρχείο vertical\_nav.jsp**



```

<%@ taglib prefix="cms" uri="http://www.OpenCMS.org/taglib/cms" %>
<%@page
import="com.OpenCMS.flex.jsp.*,com.OpenCMS.file.*,java.util.*" %>

<%!

private static final String NAVDIRECTIONKEY = "Title";
public CmsJspNavElement getFolderLink(CmsJspActionElement cms,
String folderUri)
{
ArrayList navList =
cms.getNavigation().getNavigationForFolder(folderUri);
CmsJspNavElement nav;
Iterator navItem;
CmsJspNavElement link;
// Check direction
if (cms.property(NAVDIRECTIONKEY, folderUri, "down").equals("up"))
{
Collections.reverse(navList);
}
if (navList.size() > 0)
{
// First check files
navItem = navList.iterator();
while (navItem.hasNext()) {
nav = (CmsJspNavElement) navItem.next();
if (nav != null && !nav.isFolderLink())
{
return nav;
}
}
// then check folders
navItem = navList.iterator();
while (navItem.hasNext())
{
nav = (CmsJspNavElement) navItem.next();
if (nav != null && nav.isFolderLink())
{
link = getFolderLink(cms, nav.getResourceName());
if (link != null)
{

```

```

return link;
    }
}
}
// No valid link found ... return null
return null;
}

%>
<%
CmsJspActionElement cms = new CmsJspActionElement (pageContext,
request,
response);

String username = (String)session.getAttribute ("CMS_USER_NAME");
String navText = "";
String navText2 = "";
String href = "";
cms.getRequestContext ().getFolderUri ();
String currentFolder = "/cms/";
String currentPage = cms.getRequestContext ().getUri ();
ArrayList navList =
cms.getNavigation ().getNavigationForFolder (currentFolder);    %>
<table>
<tr><td id="small">    <%= currentFolder %> <%=username%>
</td></tr></table>
<table id="smalltable" cellpadding='0' cellspacing='5' border='0'
width="95%" align="center">
<%
if (navList.size() > 0)
{
    Iterator navItem = navList.iterator();
    CmsJspNavElement nav, link;
    String target;
    String spacer;
    while (navItem.hasNext())
    {
        nav = (CmsJspNavElement) navItem.next();
        if(nav != null)

```

```

        {
            if(nav.isFolderLink()){
                navText = nav.getNavText();
                href = cms.link(nav.getResourceName());
            }
        }
    }
    <tr>
    <td align="left" valign="top" id="navfolder">
        <a href="<%= href %>"><%= navText %></a>
    </td>
    </tr>
    <%
        String tmpFolder = nav.getResourceName();
        //System.out.println(tmpFolder);
        ArrayList navList2 =
cms.getNavigation().getNavigationForFolder(tmpFolder);
        Iterator navItem2 = navList2.iterator();
        CmsJspNavElement nav2, link2;
        while (navItem2.hasNext())
        {
            nav2 = (CmsJspNavElement) navItem2.next();
            if(nav2 != null && !nav2.isFolderLink())
            {
                navText2 = nav2.getNavText();
                href = cms.link(nav2.getResourceName());
                if(tmpFolder.endsWith("private/") && username==null)
                    continue;
            }
        }
    <%>
    <tr>
    <td align="left" valign="top" id="navitem">
        <a href="<%= href %>"><%= navText2 %></a>
    </td>
    </tr>
    <% }
    }
    }
    }
    <%>
</table>

```

## Αρχείο header.jsp

```
<a href="http://www.ntua.gr">

</a>
<a href="http://www.eurisko.gr">

</a>
```

## Αρχείο footer.jsp

```
<p id="small">
&copy; 2032-2004
<a href="http://www.ntua.gr">NTUA</a>
<br>All Rights Reserved</p>
```