



# **ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΜΕΘΟΔΟΛΟΓΙΕΣ ΚΑΙ ΠΡΟΤΥΠΑ ΣΥΛΛΟΓΗΣ ΚΑΙ  
ΔΗΜΟΣΙΕΥΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ ΣΕ ΣΥΣΤΗΜΑΤΑ  
ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ**

## **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΤΟΥ**

**ΣΤΕΦΑΝΟΥ Χ. ΕΥΑΓΓΕΛΟΥ**

Επιβλέπων : **Τιμόλεων Σελλής**  
**Καθηγητής Ε.Μ.Π.**

**Αθήνα, Μάρτιος 2005**





**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ**  
**ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**  
**ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΜΕΘΟΔΟΛΟΓΙΕΣ ΚΑΙ ΠΡΟΤΥΠΑ ΣΥΛΛΟΓΗΣ ΚΑΙ**  
**ΔΗΜΟΣΙΕΥΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ ΣΕ ΣΥΣΤΗΜΑΤΑ**  
**ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΤΟΥ**

**ΣΤΕΦΑΝΟΥ Χ. ΕΥΑΓΓΕΛΟΥ**

Επιβλέπων : **Τιμολέων Σελλής**  
**Καθηγητής Ε.Μ.Π.**

**Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 3<sup>η</sup> Μαρτίου 2005.**

.....  
**Τιμολέων Σελλής**  
**Καθηγητής Ε.Μ.Π.**

.....  
**Παναγιώτης Τσανάκας**  
**Καθηγητής Ε.Μ.Π.**

.....  
**Νεκτάριος Κοζύρης**  
**Επ. Καθηγητής Ε.Μ.Π.**

**Αθήνα, Μάρτιος 2005**

.....

**ΣΤΕΦΑΝΟΣ Χ. ΕΥΑΓΓΕΛΟΥ**

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

*Copyright © 2005 Στέφανος Χ. Ευαγγέλου*

*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 στη 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.*

# ***Πρόλογος***

Η διπλωματική εργασία υλοποιήθηκε στα πλαίσια της πρωτοβουλίας για την δημιουργία επιστημονικού οδηγού από το Εργαστήριο Βάσεων Γνώσεων και Δεδομένων της Σχολής ΗΜΜΥ του ΕΜΠ. Κατά πρώτο λόγο, κατά τη διάρκεια της εργασίας πραγματοποιήθηκε εκτενής βιβλιογραφική καταγραφή για μια σειρά ζητημάτων που αφορούν στην διαχείριση περιεχομένου και ειδικότερα στην συλλογή και δημοσίευση περιεχομένου. Κατά δεύτερο λόγο, εξετάστηκαν μια σειρά από περισσότερο πρακτικά ζητήματα με βάση το σύστημα διαχείρισης περιεχομένου COMMET του Εργαστηρίου Συστημάτων Βάσεων Γνώσεων και Δεδομένων του ΕΜΠ. Οι πρακτικές υλοποιήσεις αφορούν στην συλλογή και δημοσίευση περιεχομένου από το COMMET και θέτουν την βάση για μια σειρά νέων εφαρμογών για την περαιτέρω εξέλιξη και αναβάθμιση του συστήματος.

Θέλω να εκφράσω τις θερμότερες ευχαριστίες μου στον επιστημονικό υπεύθυνο καθηγητή κ.Τίμο Σελλή για την υποστήριξή του σε όλες τις φάσεις διεκπεραίωσης της διπλωματικής εργασίας. Επίσης θέλω να ευχαριστήσω τον επιβλέποντα της εργασίας κ.Κυριάκο Διακονικολάου για την συνολική καθοδήγησή του και συμβολή του στην ολοκλήρωση της εργασίας.

***Αθήνα, Δευτέρα 31 Ιανουαρίου 2005***  
***Στέφανος Χ. Ευαγγέλου***

## *Περίληψη*

Το αντικείμενο αυτής της διπλωματικής είναι η θεωρητική μελέτη των μεθοδολογιών και προτύπων για εισαγωγή (aggregation) και δημοσίευση (syndication) περιεχομένου στα συστήματα διαχείρισης περιεχομένου (Content Management Systems) και η πρακτική υλοποίηση επιλεγμένων παραδειγμάτων. Τελικός στόχος είναι η μελέτη της εφικτότητας για ενσωμάτωση των μηχανισμών για εισαγωγή και δημοσίευση περιεχομένου στο περιβάλλον COMMET (Content Management Methodologies Environments & Tools) του Ε.Σ.Β.Γ.Δ. του Ε.Μ.Π.

Στα πλαίσια του πρώτου μέρους της διπλωματικής μελετήθηκαν schemas ανταλλαγής περιεχομένου, όπως το RSS και Atom, πρωτόκολλα ανταλλαγής όπως το SOAP και άλλες τεχνολογίες συναφείς με Web Services. Στο δεύτερο μέρος της εργασίας υλοποιήθηκαν οι μετασχηματισμοί (XSLT) για την δημοσίευση του περιεχομένου των εφαρμογών του COMMET σε schemas όπως το RSS και Atom. Τέλος προτάθηκαν αρχιτεκτονικές για την υλοποίηση των Web Services μέσω των οποίων οι εφαρμογές του COMMET θα δημοσιεύουν το περιεχόμενό τους.

## *Abstract*

The scope of the thesis was the theoretical study of methodologies and standards in the fields of content aggregation and syndication in Content Management Systems, as well as the practical implementation of specific examples. The final goal was to study the feasibility to integrate mechanisms of aggregation and syndication into the environment of the COMMET system (Content Management Methodologies Environments & Tools) of the K.D.B.S. Laboratory of N.T.U.A.

In the first part of the thesis we studied various content exchange schemas, such as RSS and Atom, various content exchange protocols, such as SOAP, as well as other various technologies relevant to Web Services. In the second part of the thesis we implemented the transformations (XSLT) in order to syndicate the content of COMMET's applications into schemas, such as RSS and Atom. Finally we proposed architectures in order to implement Web Services with which to syndicate the content of COMMET's applications.

# Πίνακας Περιεχομένων

<b>1</b>	<b>Εισαγωγή.....</b>	<b>14</b>
1.1	Περιγραφή της εργασίας.....	14
1.2	Οργάνωση του τόμου .....	17
<b>2</b>	<b>Διαχείριση Περιεχομένου.....</b>	<b>18</b>
2.1	Τι είναι περιεχόμενο .....	18
2.2	Τι είναι σύστημα διαχείρισης περιεχομένου .....	20
2.3	Υποσυστήματα σε ένα CMS .....	22
2.3.1	Το υποσύστημα collection.....	23
2.3.2	Υποσύστημα (core) management.....	26
2.3.3	Υποσύστημα publishing.....	28
2.4	Οντότητες (entities) ενός CMS.....	31
2.5	Λειτουργικά χαρακτηριστικά CMS.....	32
2.6	Πρακτικές εφαρμογές των CMS.....	40
2.6.1	Δυναμικά Web Sites με ισχυρές δυνατότητες προσωποποίησης (personalization).....	40
2.6.2	Επιχειρηματικές εφαρμογές των CMS.....	40
2.6.3	Διαδικτυακές κοινότητες (Online communities).....	41
2.6.4	Ειδησεογραφικά πρακτορεία (news agencies) και Τύπος (Εντυπος και Ηλεκτρονικός).....	42
<b>3</b>	<b>Κατηγοριοποίηση ΣΔΠ.....</b>	<b>43</b>
3.1	Διάκριση με βάση το εξειδικευμένο περιεχόμενο .....	44
3.1.1	Συστήματα Διαχείρισης Εταιρικού Περιεχομένου - Enterprise Content Management System (ECMS).....	44
3.1.2	Customer Relationship Management Systems .....	44
3.1.3	Ολοκληρωμένα συστήματα διαχείρισης εγγράφων (Integrated Document Management Systems) .....	45
3.1.4	Συστήματα Διαχείρισης Εγγραφών - Records management systems (RMS) 45	
3.1.5	Enterprise Resource Planning (ERP) – Human Resources Management systems (HRM) .....	45
3.1.6	Συστήματα Διαχείρισης Ψηφιακών Πόρων - Digital asset management (DAM) system .....	46
3.1.7	Συστήματα Διαχείρισης Τίτλων - Brand management system.....	46
3.1.8	Συστήματα Διαχείρισης Ψηφιακών Δικαιωμάτων - Digital Rights Management (DRM) system.....	46
3.1.9	Portals - Vortals.....	47
3.1.10	Library management system (LMS) .....	47
3.1.11	Digital imaging system (DIS).....	47
3.1.12	Learning content management system (LCMS).....	47
3.1.13	Διαχείριση γνώσης (Knowledge management - KM).....	48
3.1.14	Geographic information system (GIS) .....	48
3.1.15	Software Configuration Management (SCM) .....	48



3.2	Με βάση το κόστος απόκτησης και συντήρησης.....	49
3.2.1	Εισαγωγή.....	49
3.2.2	Πακέτα Enterprise platforms.....	50
3.2.3	Πακέτα Upper Tier .....	52
3.2.4	Πακέτα Mid-Market .....	53
3.2.5	Πακέτα Lower-priced .....	54
3.2.6	Πακέτα ASP.....	54
3.2.7	Πακέτα Delivery-Oriented .....	54
3.2.8	Πακέτα Open Source .....	55
4	Το σύστημα COMMET .....	58
4.1	Τύποι Περιεχομένου του COMMET .....	59
4.1.1	Κλάσεις κόμβων .....	60
4.1.2	Κλάσεις Πόρων.....	62
4.2	Λειτουργίες .....	66
4.2.1	Λειτουργίες σε κλάσεις κόμβων .....	66
4.2.2	Λειτουργίες σε κλάσεις πόρων .....	81
4.2.3	Γενικές λειτουργίες .....	90
5	Η γλώσσα σήμανσης XML .....	97
5.1.1	Γενικά .....	97
5.1.2	DTD .....	98
5.1.3	Namespaces .....	100
5.1.4	Schemas.....	103
5.1.5	XSLT / XSLFO .....	106
5.1.6	DOM/SAX.....	110
6	Web Services.....	113
6.1	Εισαγωγή στις Web Services.....	113
6.1.1	Τι είναι μια διαδικτυακή υπηρεσία (Web Service).....	113
6.1.2	Σύντομη ιστορική αναδρομή των υπηρεσιών λογισμικού, εξέλιξη των οποίων αποτελούν οι Διαδικτυακές Υπηρεσίες. ....	113
6.2	Τεχνολογίες των Web Services.....	115
6.2.1	Πρωτόκολλο SOAP .....	116
6.2.2	Γλώσσα περιγραφής Διαδικτυακών Υπηρεσιών WSDL .....	121
6.2.3	Πρωτόκολλο UDDI .....	122
6.2.4	Ασφάλεια των Διαδικτυακών Υπηρεσιών .....	123
6.3	Εργαλεία για υλοποίηση των Web Services.....	125
6.3.1	Apache Axis .....	125
6.3.2	IBM Web Services Toolkit .....	125
6.3.3	Microsoft SOAP Toolkit 2 .....	125
6.3.4	Οι Web Services στο περιβάλλον .NET .....	126
6.3.5	Παράδειγμα μιας Web Service.....	128
6.3.6	Παραδείγματα υλοποιημένων Web Services στο Διαδίκτυο .....	130
6.4	Το μέλλον των Web Services.....	131
7	Content aggregation & syndication .....	132
7.1	Εισαγωγή στα πρωτόκολλα syndication .....	132

7.1.1	RSS.....	134
7.1.2	OPML (Outline Processor Markup Language).....	142
7.1.3	Atom .....	145
7.1.4	Άλλα πρωτόκολλα .....	148
7.2	Εφαρμογές του content syndication.....	154
7.2.1	Newsletters - spam.....	154
7.2.2	Αναζήτηση περιεχομένου.....	154
7.2.3	Blogs.....	154
7.2.4	Dynamic Bookmarks.....	154
7.2.5	Mobile Devices .....	155
7.2.6	Επιχειρηματικότητα.....	155
7.3	Syndication περιεχομένου από το COMMET .....	156
7.3.1	Γενικά .....	156
7.3.2	Αντιστοίχιση οντοτήτων COMMET με RSS 2.0 και OPML 1.0. ....	159
7.3.3	Σχεδίαση XSLT stylesheet.....	165
7.4	Το μέλλον για τις τεχνολογίες aggregation-syndication.....	170
7.4.1	Μειονεκτήματα – παράμετροι υπό βελτίωση .....	170
7.4.2	Συμπερασματικά .....	171
8	Επεκτάσεις της παρούσας εργασίας .....	172
8.1	Επέκταση του syndication στο COMMET με τα υπόλοιπα γνωστά schemas	172
8.2	Aggregation περιεχομένου από το COMMET .....	172
8.3	Application syndication .....	173
<b>ΠΑΡΑΡΤΗΜΑ Α Πρωτόκολλα για syndication .....</b>		<b>177</b>
	Σύγκριση μεταξύ των διαφορετικών εκδόσεων του RSS .....	177
	Σύγκριση των OPML Directories για κάθε ένα από τα παραπάνω RSS1, RSS2, Atom .....	198
	Παραδείγματα από Web site feeds.....	204
<b>ΠΑΡΑΡΤΗΜΑ Β .....</b>		<b>210</b>
	Web logs (Blogs) .....	210
	Τί είναι τα Blog.....	210
	Τί είναι τα Wiki .....	210
	Παραδείγματα Blog.....	212
<b>ΠΑΡΑΡΤΗΜΑ Γ .....</b>		<b>215</b>
	<i>Περί αδειών λογισμικού .....</i>	<i>215</i>
	Ελεύθερο Λογισμικό / Λογισμικό Ανοικτού Κώδικα.....	215
	Ελεύθερο Λογισμικό Copylefted .....	216
	Ελεύθερο Λογισμικό Non-copylefted.....	216
	Λογισμικό Public domain .....	216
	Λογισμικό άδειας GPL.....	216
	Μη ελεύθερο Λογισμικό.....	216
	Ημιελεύθερο Λογισμικό .....	217
	Λογισμικό Proprietary .....	217
	Λογισμικό Freeware.....	217
	Λογισμικό Shareware.....	217

Ιδιωτικό Λογισμικό .....	217
Εμπορικό Λογισμικό .....	217
<i>ΠΑΡΑΡΤΗΜΑ Δ</i> .....	218
<i>Πρότυπα στην Πληροφορική</i> .....	218
Γενικά περί προτύπων.....	218
Πρότυπα για το Διαδίκτυο.....	218
ΠΑΡΑΡΤΗΜΑ Ε: XSLT stylesheets .....	221
ΑΝΑΦΟΡΕΣ .....	227
Βιβλία και άρθρα στο Web.....	227
Διάφορα URL.....	229

## *Πίνακας σχημάτων και πινάκων*

Σχήμα 1: Υποσυστήματα σε ένα CMS [BOI02].....	22
Σχήμα 2: Υποσύστημα collection [BOI02] .....	25
Σχήμα 3: CMS Repository [BOI02].....	26
Σχήμα 4: Διάφορες μορφές publications .....	28
Σχήμα 5 Σχηματική αναπαράσταση υποσυστήματος δημοσίευσης περιεχομένου [BOI02].....	30
Σχήμα 6: Διαδικασία δημιουργίας CMS εφαρμογών.....	53
Σχήμα 7 : Classification των URI .....	102
Σχήμα 8: Παράδειγμα XML Schema.....	104
Σχήμα 9: Σύγκριση των XSLT και CSS.....	109
Σχήμα 10: Επεξεργασία DOM με και χωρίς XSLT.....	111
Σχήμα 11: Επεξεργασία DOM στην πλευρά του Microsoft IIS server.....	112
Σχήμα 12: Αρχιτεκτονική middleware.....	114
Σχήμα 13: Κλήση των Web Services.....	117
Σχήμα 14: Αναζήτηση σε UDDI.....	117
Σχήμα 15 : XML-RPC .....	119
Σχήμα 16 Ασύμβατα CORBA/DCOM δίκτυα που απαιτούν γέφυρα για να επικοινωνήσουν .....	119
Σχήμα 17 Η χρήση του SOAP εξασφαλίζει μια κοινή γλώσσα επικοινωνίας.....	120
Σχήμα 18: Schema για το RSS 2.0 .....	135
Σχήμα 19: DTD για το RSS 0.91 .....	140
Σχήμα 20: Schema OPML .....	144
Σχήμα 21: Schema Atom .....	146
Σχήμα 22: NewsML format example.....	150
Σχήμα 23: NITF format example.....	152
Σχήμα 24: Αρχιτεκτονική για syndication περιεχομένου από το COMMET .....	157
Σχήμα 25 Δημοσίευση και διάθεση Web Services από το COMMET .....	157
Σχήμα 26: XSLT stylesheets για το COMMET.....	165
Σχήμα 27 Aggregation περιεχομένου από εξωτερικές πηγές.....	172
Σχήμα 28 Ανεξάρτητη εφαρμογή συλλογής προσωπικών στοιχείων χρήστη [COC04].....	174
Σχήμα 29 Η παραπάνω εφαρμογή ενσωματωμένη σε ένα website [COC04].....	174
Σχήμα 30 Application Syndication .....	175
Σχήμα 31: RSS 0.90 .....	178
Σχήμα 32: RSS 0.91 Netscape.....	179
Σχήμα 33: RSS 0.91 Userland .....	180
Σχήμα 34: RSS 1.0 .....	181
Σχήμα 35: RSS 0.92 .....	184
Σχήμα 36: RSS 0.93 .....	185
Σχήμα 37: RSS 0.94 .....	186
Σχήμα 38: RSS 2.0 .....	187
Σχήμα 39: RSS 2.0.1 .....	188
Σχήμα 40: RSS 2.0.2 .....	189
Σχήμα 41: RSS 1.0 .....	194
Σχήμα 42: RSS 2.0 .....	195
Σχήμα 43: Συνέχεια από το προηγούμενο σχήμα.....	197
Σχήμα 44: Atom.....	197
Σχήμα 45: OPML για RSS 1.....	199

Σχήμα 46: OPML για RSS 2.....	201
Σχήμα 47: OPML για Atom .....	203
Σχήμα 48 Περιεχόμενο website που προσφέρεται με μορφή RSS .....	204
Σχήμα 49: Blog για την παρουσίαση των επίσημων προδιαγραφών του RSS 2.....	204
Σχήμα 50 BBC News Feeds.....	205
Σχήμα 51: Τα νέα από την εφημερίδα Ναυτεμπορική .....	205
Σχήμα 52 Reuters News Feeds .....	206
Σχήμα 53 Γραφικό περιβάλλον Radio News Aggregator.....	207
Σχήμα 54: Mozilla Thunderbird RSS Reader .....	208
Σχήμα 55 Web-based RSS Validator.....	209
Σχήμα 56 Προσωπικό Blog.....	212
Σχήμα 57 Ακόμα ένα προσωπικό Blog .....	212
Σχήμα 58: Παράδειγμα Filter Blog για software developers.....	213
Σχήμα 59 MSDN Knowledge Blog.....	213
Σχήμα 60: Το μελλοντικό Web.....	220

Πίνακας 1: Λειτουργικά χαρακτηριστικά CMS - ομαδοποίηση.....	34
Πίνακας 2 Συνδυασμοί αδειών για ένα πακέτο CMS.....	49
Πίνακας 3: Αντιστοίχιση COMMET – OPML για κάθε κόμβο του COMMET.....	161
Πίνακας 4: Τελική αντιστοίχιση COMMET-OPML για την λειτουργία PrintMySubtreeNodes .....	161
Πίνακας 5: Αντιστοίχιση RSS <channel> - COMMET .....	163
Πίνακας 6: Αντιστοίχιση RSS <item> - COMMET .....	163
Πίνακας 7: Τελική αντιστοίχιση μεταξύ COMMET και RSS2 για την λειτουργία PrintMyResources .....	164
Πίνακας 8: Σύγκριση των δημοφιλέστερων εκδόσεων του RSS .....	190
Πίνακας 9: Αντιστοίχιση RSS 0.9x – RSS 1.0.....	192

# 1 Εισαγωγή

## 1.1 Περιγραφή της εργασίας

Αντικείμενο της παρούσας εργασίας είναι τα συστήματα διαχείρισης περιεχομένου (Content Management Systems – CMS). Ο όρος περιεχόμενο ισοδυναμεί με πληροφορία που διαθέτει μια συγκεκριμένη λειτουργικότητα και προορίζεται για συγκεκριμένο κοινό. Παραδείγματα εφαρμογών διαχείρισης περιεχομένου είναι τα portals, τα intranet και extranet sites και τα web sites μεγάλων ειδησεογραφικών πρακτορείων, επιχειρήσεων και οργανισμών. Όλα αυτά τα συστήματα διαθέτουν ένα back-end σύστημα διαχείρισης της συλλογής, επεξεργασίας και δημοσίευσης του περιεχομένου τους.

Διαχείριση περιεχομένου είναι μια διαδικασία που περιλαμβάνει επιμέρους εργασίες, όπως η συλλογή, η εισαγωγή, η οργάνωση, η κατηγοριοποίηση, η δόμηση, η επεξεργασία και η δημοσίευση μιας μεγάλης κλίμακας τύπων περιεχομένου. Κάθε σύστημα διαχείρισης περιεχομένου (content management system, CMS) είναι ένα σύστημα που αναλαμβάνει να διεκπεραιώσει τις εργασίες αυτές. Σε λογικό επίπεδο, κάθε CMS διαθέτει μια οντολογία και ένα σύνολο από taxonomies για την κατηγοριοποίηση του περιεχομένου του σε κλάσεις αντικειμένων. Η αποθήκευση του περιεχομένου σε φυσικό επίπεδο γίνεται σε συστατικά στοιχεία που επιλέγονται από το εκάστοτε CMS, όπως είναι συνήθως πίνακες σε μια σχεσιακή βάση δεδομένων.

Στην παρούσα εργασία θα μας απασχολήσουν ειδικά τα συστήματα διαχείρισης διαδικτυακού περιεχομένου (Web Content Management Systems - WCMS), τα οποία βρίσκουν ποικίλες εφαρμογές και αξιοποιούνται σε μεγάλο βαθμό για την διαχείριση διαδικτυακού περιεχομένου (υπό μορφή δημόσιου Web site ή Intranet – Extranet) από εταιρείες, οργανισμούς και άλλους φορείς.

Κάθε WCMS μπορεί να διαρθρωθεί σε τρία βασικά υποσυστήματα:

- το υποσύστημα συλλογής (collection)
- το υποσύστημα αποθήκευσης - επεξεργασίας – ελέγχου ροής του περιεχομένου (core management) και
- το υποσύστημα δημοσίευσης (publishing)

Με τον όρο **λειτουργικά χαρακτηριστικά** εννοούμε το σύνολο της λειτουργικότητας που προσφέρει μια εφαρμογή διαχείρισης περιεχομένου, τόσο στον τελικό χρήστη όσο και στον διαχειριστή. Στην παρούσα εργασία επιχειρήθηκε μια κατηγοριοποίηση των λειτουργικών αυτών χαρακτηριστικών. Παράλληλα μελετήθηκαν οι κυριότερες πρακτικές εφαρμογές των συστημάτων διαχείρισης περιεχομένου σε αρκετούς τομείς.

Επίσης σε ξεχωριστό κεφάλαιο επιχειρήθηκε μια κατηγοριοποίηση των διαθέσιμων συστημάτων CMS. Η κατηγοριοποίηση των συστημάτων έγινε με βάση δύο άξονες και πιο συγκεκριμένα:

- Με βάση το εξειδικευμένο είδος του περιεχομένου που διαχειρίζεται το σύστημα (π.χ. διακρίνουμε τα συστήματα διαχείρισης γνώσης ή τα συστήματα διαχείρισης εγγράφων)
- Με βάση το κόστος κτήσης και συντήρησης των αδειών του πακέτου λογισμικού. Εδώ έχουμε δύο μεγάλες κατηγορίες. Αφ' ενός τα open – source συστήματα και αφ' ετέρου τα εμπορικά συστήματα που με τη σειρά τους κατηγοριοποιούνται ανάλογα με το κόστος, τις δυνατότητες που προσφέρουν και την αγορά στην οποία απευθύνονται.

Το Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων του ΕΜΠ έχει αναπτύξει και συντηρεί ένα ολοκληρωμένο σύστημα διαχείρισης περιεχομένου, το οποίο έχει σαν στόχο να συμβάλει στην ανάπτυξη πολλαπλών εφαρμογών διαχείρισης περιεχομένου, όπως επιστημονικοί οδηγοί, κυβερνητικά

και εκπαιδευτικά portals και άλλες εφαρμογές συνεργασίας και διαχείρισης γνώσης. Το σύστημα αυτό ονομάζεται COMMET (Content Management Methodologies, Environments and Tools). Στην εργασία αφιερώνεται ένα ολόκληρο κεφάλαιο στην αναλυτική παρουσίαση της αρχιτεκτονικής του COMMET. Περισσότερα για το COMMET μπορούν να αναζητηθούν στο URL <http://commet.dbnet.ntua.gr>.

Ένα σημείο στο οποίο το περιεχόμενο διαφέρει από την στατική πληροφορία είναι ότι το περιεχόμενο είναι σχεδιασμένο ώστε να επιτρέπει την άμεση επαναδημοσίευση και ενσωμάτωσή του σε κάθε πληροφοριακό σύστημα το οποίο διαθέτει την κατάλληλη υποδομή. Η διαδικασία της συλλογής και ενσωμάτωσης περιεχομένου από πολλές διαφορετικές πηγές σε ένα σύστημα ονομάζεται **aggregation**. Παράλληλα η διαδικασία μαζικής δημοσίευσης περιεχομένου με στόχο την άμεση επαναχρησιμοποίηση και επαναδημοσίευσή του ονομάζεται **syndication**. Στην εργασία αφιερώνεται ένα ξεχωριστό κεφάλαιο σχετικά με το aggregation, το syndication και τις σημαντικότερες εφαρμογές τους. Μια από τις σημαντικότερες εφαρμογές είναι τα Weblogs και τα Wikis, στα οποία αφιερώνεται ένα ξεχωριστό παράρτημα. Τελικός στόχος της θεωρητικής μελέτης είναι η υλοποίηση μεθόδων για aggregation και syndication περιεχομένου από το σύστημα COMMET του DBLAB.

Οι διαδικασίες του content aggregation και syndication είναι δύο συμπληρωματικές διαδικασίες που έχουν να κάνουν με την συλλογή και δημοσίευση περιεχομένου αντίστοιχα. Ένα σύστημα CMS διαθέτει μηχανισμούς με τους οποίους συγχωνεύει περιεχόμενο από ετερογενείς εξωτερικές πηγές και τις ενσωματώνει στο δικό του μοντέλο δεδομένων. Παράλληλα το CMS δίνει την δυνατότητα μαζικής δημοσίευσης του περιεχομένου που περιέχει με την μορφή μικρών δομημένων τμημάτων περιεχομένου, τα λεγόμενα "feeds", ώστε να τροφοδοτήσει άλλα συστήματα διαχείρισης περιεχομένου, εφαρμογές και χρήστες που ενδιαφέρονται να συγκεντρώσουν και να αξιοποιήσουν το περιεχόμενο αυτό.

Κάθε φορέας περιεχομένου διαθέτει το δικό του XML schema αναπαράστασης της οντολογίας του περιεχομένου του. Για τον λόγο αυτό όταν επιχειρείται μετάδοση περιεχομένου με έναν φορέα που υποστηρίζει διαφορετικό XML schema, πρέπει να εξασφαλίζεται κάποιος τρόπος μετατροπής από το ένα schema στο άλλο. Τα τελευταία χρόνια ωστόσο έχουν επικρατήσει διεθνώς κάποια schemas τα οποία αποτελούν εφαρμόσιμα de facto πρότυπα, παρέχοντας μια κοινή γλώσσα επικοινωνίας μεταξύ των παρόχων περιεχομένου. Τα πιο γνωστά από τα XML schemas αυτά είναι το RSS 2.0 (Really Simple Syndication), το RSS 1.0 (RDF Site Summary) και το Atom 0.3. Με βάση το δημοφιλέστερο και πιο πρόσφατο RSS 2.0 παρουσιάζονται αναλυτικά όλα τα διαθέσιμα elements και attributes για το format αυτό. Σε ξεχωριστό παράρτημα γίνεται αναλυτική σύγκριση και παράλληλη παρουσίαση παραδειγμάτων από τα διαφορετικά syndication format.

Στην παρούσα εργασία μας ενδιαφέρει κυρίως η υλοποίηση Web Services για syndication περιεχομένου από το COMMET. Έτσι κάθε εξωτερικό σύστημα ή εφαρμογή, ανεξαρτήτως XML Schema που χρησιμοποιεί, να μπορεί να αξιοποιεί το διαθέσιμο περιεχόμενο. Υπάρχουν δύο σχετικά πιθανά σενάρια:

- Το COMMET παρέχει τα δεδομένα του σε μορφή RSS και Atom, υλοποιώντας εσωτερικά μετασχηματισμούς XSLT για την μετατροπή των δεδομένων από το δικό του schema στα δύο γνωστά syndication format. Έτσι τα δεδομένα μπορούν να εισαχθούν αμέσως σε οποιαδήποτε μηχανή υποστηρίζει κάποιο από τα format αυτά.
- Το COMMET παρέχει τα δεδομένα του σε μορφή XML και επίσης παρέχει το XML Schema (μορφή αρχείου .xsd) που χρησιμοποιεί. Έτσι ο κάθε χρήστης μπορεί να αξιοποιήσει τα δεδομένα του COMMET, αρκεί να κατασκευάσει τον κατάλληλο μετασχηματισμό από το schema του COMMET στο schema που αυτός χρησιμοποιεί.

Παράλληλα μελετάται η δυνατότητα aggregation από το COMMET, δηλαδή ενσωμάτωσης και παρουσίασης από εφαρμογές του COMMET πολλών διαφορετικών πηγών περιεχομένου που διαθέτουν το περιεχόμενό τους σε ένα από τα δημοφιλή schemas RSS και Atom.

Έτσι ο τελικός στόχος με βάση τα όσα προαναφέρθηκαν είναι να μπορούμε να πετύχουμε:

1. Παρουσίαση του περιεχομένου του COMMET σε μορφή RSS για τον τελικό χρήστη των εφαρμογών του COMMET
2. Δημιουργία δυνατοτήτων για aggregation περιεχομένου από τις εφαρμογές του COMMET
3. Παρουσίαση του περιεχομένου του COMMET σε μορφή COMMET schema κάνοντας χρήση Web Services ώστε να είναι αξιοποιήσιμο από άλλες εφαρμογές

Τέλος, λόγω της μεγάλης εξάρτησης των CMS και ιδιαίτερα των διαδικασιών aggregation και syndication από την γλώσσα XML, προηγείται ένα ξεχωριστό εισαγωγικό κεφάλαιο στην μελέτη της γλώσσας. Η XML είναι markup γλώσσα για έγγραφα που περιέχουν δεδομένα με ιεραρχική δόμηση. Οι δομές αυτές περιλαμβάνουν δεδομένα και μεταδεδομένα (metadata) που περιγράφουν την σημασιολογία των δεδομένων αυτών. Εκμεταλλευόμενοι την ιεραρχία που υποστηρίζεται εγγενώς από την XML, μπορούμε να περιγράψουμε μια σειρά από taxonomies σε μια οντολογία δεδομένων με χρήση XML εγγράφων. Τα taxonomies είναι ιεραρχικά οργανωμένες δομές οντοτήτων για ένα δεδομένο σύνολο αντικειμένων. Αυτή η δυνατότητα είναι ιδιαίτερα χρήσιμη σε ένα σύστημα διαχείρισης περιεχομένου. Σε ένα CMS οι οντότητες μπορούν να οργανωθούν με βάση τα taxonomies. Επιπρόσθετα, γνωρίζοντας τα εργαλεία επεξεργασίας των XML εγγράφων μπορούμε να μετασχηματίζουμε κατάλληλα το περιεχόμενο με στόχο την συμβατότητα της παρουσιάσής του σε διαφορετικά ετερογενή συστήματα και περιβάλλοντα. Για τον λόγο αυτό παρουσιάζονται έννοιες όπως τα DTD, XML Schemas, namespaces, XSLT και το DOM XML API.

Ένα πολύ σημαντικό θέμα για τα συστήματα διαχείρισης περιεχομένου είναι η κοινή χρήση περιεχομένου μεταξύ των εφαρμογών. Παράλληλα με την XML και τη χρήση των μεταδεδομένων (metadata), πολύ χρήσιμες είναι οι Διαδικτυακές Υπηρεσίες (Web Services) για τις οποίες αφιερώνουμε, αμέσως μετά την μελέτη της XML, ένα ξεχωριστό κεφάλαιο. Οι Διαδικτυακές Υπηρεσίες βασίζονται στις τεχνολογίες SOAP (επίσημα Simple Object Access Protocol, WSDL (Web Services Description Language) και UDDI (Universal Description, Discovery and Integration). Αναλύεται το πώς γίνεται η υλοποίηση, δημοσίευση, αναζήτηση και κλήση μιας Web Service.

Η εργασία κλείνει με προτάσεις για μελλοντικές υλοποιήσεις με βάση το σύστημα COMMET.



## 1.2 Οργάνωση του τόμου

Στο Κεφάλαιο 2 γίνεται μια εισαγωγή στις έννοιες περιεχόμενο και σύστημα διαχείρισης περιεχομένου. Περιγράφεται η δομή και η λειτουργία ενός συστήματος διαχείρισης διαδικτυακού περιεχομένου (Web Content management System). Ιδιαίτερα τονίζονται τα λειτουργικά χαρακτηριστικά που πρέπει να διαθέτει μια εφαρμογή portal ενός WCMS μαζί με τα διαχειριστικά εργαλεία που την συνοδεύουν. Γίνεται αναφορά στην γλώσσα σήμανσης XML και τα μεταδεδομένα και πώς αυτά εφαρμόζονται στην λειτουργία των επιμέρους υποσυστημάτων ενός συστήματος διαχείρισης περιεχομένου. Τέλος αναλύονται οι κυριότερες πρακτικές εφαρμογές των WCMS μαζί με κάποια χαρακτηριστικά παραδείγματα.

Στο Κεφάλαιο 3 επιχειρείται μια καταγραφή των σημερινών σημαντικότερων συστημάτων διαχείρισης περιεχομένου. Η ομαδοποίησή τους γίνεται με βάση δύο κεντρικούς άξονες: το είδος περιεχομένου που διαχειρίζονται και το κόστος των συνοδευτικών αδειών κτήσης-συντήρησης.

Το Κεφάλαιο 4 παρουσιάζει μια ανάλυση του συστήματος διαχείρισης διαδικτυακού περιεχομένου COMMET (Content Management Methodologies, Environments and Tools) το οποίο αναπτύχθηκε και συντηρείται από ομάδα του Εργαστηρίου Συστημάτων Βάσεων Γνώσεων και Δεδομένων του ΕΜΠ.

Στο Κεφάλαιο 5 συζητούμε θέματα σχετικά με την γλώσσα XML και συναφείς τεχνολογίες, όπως τα DTD, οι χώροι ονοματοδοσίας, τα schemas, η γλώσσα XSL και το DOM API.

Στο Κεφάλαιο 6 γίνεται μια εισαγωγή στις τεχνολογίες που σχετίζονται με τις Διαδικτυακές Υπηρεσίες (Web Services), όπως είναι το SOAP, η γλώσσα WSDL και το UDDI. Στο τέλος του κεφαλαίου παρατίθεται μια λίστα με τα γνωστότερα προϊόντα που υποστηρίζουν Web Services, ενώ δίνονται και παραδείγματα URL με υλοποιημένες Web Services.

Στο Κεφάλαιο 7 γίνεται ανάλυση των διαδικασιών aggregation και syndication για ανταλλαγή περιεχομένου μεταξύ συστημάτων με βάση τα πρωτόκολλα του Web και την XML. Ειδικότερα μελετώνται οι προδιαγραφές που αφορούν στην μορφοποίηση των εγγράφων RSS, OPML και Atom, με την μελέτη των αντίστοιχων XML Schemas και κάποιων παραδειγμάτων. Τέλος παραθέτουμε την μελέτη που έγινε σχετικά με το syndication περιεχομένου από το COMMET με χρήση του RSS 2.0

Στο Κεφάλαιο 8 προτείνονται μελλοντικές επεκτάσεις με βάση την παρούσα εργασία.

Στο τέλος των κεφαλαίων παρατίθενται παραρτήματα. Στο παράρτημα Α δίνονται αναλυτικά παραδείγματα σύγκρισης των εγγράφων RSS και Atom. Στο παράρτημα Β παρουσιάζονται αναλυτικότερα τα Web logs (Weblogs). Στο παράρτημα Γ γίνεται μια επισκόπηση των κατηγοριών αδειών λογισμικού, όπως είναι η GNU GPL, και καταγράφονται τα διαφορετικά είδη λογισμικού. Στο παράρτημα Δ συζητούμε κάποια θέματα σχετικά με τα πρότυπα στην Πληροφορική. Τέλος στο Παράρτημα Ε παρέχουμε τον κώδικα για τους απαιτούμενους XSL μετασχηματισμούς για syndication περιεχομένου από το COMMET.

### **Λέξεις κλειδιά**

Περιεχόμενο, σύστημα διαχείρισης περιεχομένου, διαδικτυακές υπηρεσίες, κανάλι περιεχομένου, πάροχος, Content, Content Management System, COMMET, Internet, content entities, Web Services, RSS, Atom, OPML, Prism, ICE, SOAP, WSDL, UDDI, .NET, XML, XSL, XSLFO, CSS, DTD, Schema, entry, blog, Weblog, channel, feed, content provider, aggregator, parser, validator, RPC, middleware, taxonomies, websites, intranet, extranet, syndication, publication, aggregation, collection, template, security, RDF, semantic web, attribute, element

## 2 Διαχείριση Περιεχομένου

### 2.1 Τι είναι περιεχόμενο

Οι ηλεκτρονικοί υπολογιστές ξεκίνησαν ως απλά υπολογιστικά εργαλεία που επεξεργάζονταν δεδομένα και είχαν τη δυνατότητα για επανάληψη σύνθετων υπολογισμών. Αρχικά χρησιμοποιήθηκαν ως λογιστικά εργαλεία στις επιχειρήσεις ή ως εργαλεία ελέγχου της παραγωγής στις βιομηχανίες. Εν συνεχεία πολλές χρήσεις των υπολογιστών προστέθηκαν στον τομέα του αυτοματισμού γραφείου, όπως για παράδειγμα τα προγράμματα επεξεργασίας κειμένου και επιτραπέζιας τυπογραφίας (desktop publishing). Η διαδίκτυωση των συστημάτων αύξησε τις δυνατότητες επικοινωνίας και διευκόλυνε τις επιχειρηματικές συναλλαγές. Ευρεία υπήρξε και η χρήση των υπολογιστών στην ανερχόμενη βιομηχανία των πολυμέσων. Με την εισαγωγή των πολυμέσων στον κόσμο των υπολογιστών, σε συνδυασμό με τη χρήση του Διαδικτύου, φτάσαμε στο σημείο οι υπολογιστές να μπορούν να προσφέρουν στον χρήστη υπηρεσίες και πληροφορίες σε πεδία που μέχρι τότε μονοπωλούνταν από μέσα όπως η τηλεόραση, το ραδιόφωνο και το βιβλίο.

Κοινός παρονομαστής όλων των παραπάνω χρήσεων του υπολογιστή είναι η συλλογή, επεξεργασία και δημοσίευση της πληροφορίας. Η πληροφορία (information), σε αντίθεση με τη λέξη δεδομένο (data), αποτελεί μια δυναμική έννοια. Αφορά στο υλικό που προκύπτει από την κατάλληλη επεξεργασία των δεδομένων. Τα αποτελέσματα της επεξεργασίας αυτής πρέπει να έχουν νόημα και να είναι πρακτικά εφαρμόσιμα. Υπάρχουν πολλά είδη πληροφορίας που είναι απτά από τον άνθρωπο, με βασικότερες μορφές της το κείμενο, τον ήχο, την σταθερή και την κινούμενη εικόνα και διάφορα είδη αρχείων.

Οι Η/Υ διαθέτουν αρχιτεκτονικές που βασίζονται στην επεξεργασία των δεδομένων. Η λογική που εφαρμόζουν οι υπολογιστές για την διαχείριση της πληροφορίας διαφέρει κατά πολύ από την ανθρώπινη λογική. Τα μέχρι τώρα συστήματα διαθέτουν σε πολύ μικρό βαθμό δυνατότητες τεχνητής νοημοσύνης, μάθησης και επεξεργασίας φυσικής γλώσσας. Για τον άνθρωπο τα δεδομένα από μόνα τους δεν έχουν κάποια ιδιαίτερη σημασία. Επιπρόσθετα δεν είναι σε θέση, λόγω του φόρτου των στοιχειωδών πράξεων που απαιτούνται, να επεξεργαστεί κατευθείαν δεδομένα για να εξάγει πληροφορία.

Προχωρώντας ένα επίπεδο αφαίρεσης πιο πάνω από την πληροφορία, μας ενδιαφέρει ο τρόπος με τον οποίο η πληροφορία γίνεται αξιοποιήσιμη από τον άνθρωπο. Αυτό έχει να κάνει με τα συμφραζόμενα (context) της πληροφορίας και με το πώς ο καθένας μας αντιλαμβάνεται και μεταφράζει τα δεδομένα που αυτή μεταφέρει. Είναι πιθανό για ένα πλαίσιο πληροφοριών να υπάρχουν πολλές διαφορετικές δυνατότητες παρουσίασής του, ανάλογα με το υπόβαθρο και τα ενδιαφέροντα του αποδέκτη. Με την λογική αυτή η πληροφορία αποκτά ένα συγκεκριμένο σκοπό προς επίτευξη με βάση τις ανάγκες που καθορίζονται τόσο από τον αποδέκτη όσο και από τον δημιουργό της. Η πληροφορία, όταν την εκλαμβάνουμε υπό το προαναφερθέν πρίσμα, ονομάζεται και περιεχόμενο (content). Το πιο σημαντικό χαρακτηριστικό της έννοιας «περιεχόμενο» είναι ο ανθρώπινος παράγοντας. Η φύση του περιεχομένου είναι τέτοια που το καθιστά προσιτό και άμεσα κατανοητό από τον άνθρωπο.

Λόγω των αρχιτεκτονικών που επικρατούν σήμερα στην σχεδίαση των υπολογιστικών συστημάτων, δεν διαθέτουμε επαρκή εργαλεία για την απευθείας διαχείριση του περιεχομένου. Διαθέτουμε μόνο μηχανές με δυνατότητα επεξεργασίας ακατέργαστων δεδομένων (raw data). Για την διαχείριση επομένως του περιεχομένου πρέπει να αξιοποιήσουμε κατάλληλα τα διαθέσιμα εργαλεία. Ελπίζουμε ότι στο άμεσο μέλλον θα υλοποιηθεί μια νέα φιλοσοφία δομής και λειτουργίας των υπολογιστικών συστημάτων, που θα φέρει τις μηχανές πιο κοντά στον ανθρώπινο τρόπο σκέψης και θα τους δώσει την δυνατότητα της απευθείας διαχείρισης του περιεχομένου.

Με τα υπάρχοντα εργαλεία διαχείρισης δεδομένων , αυτό που πρακτικά γίνεται είναι να περιβάλλουμε την πληροφορία με ειδικά δεδομένα που ονομάζουμε μεταδεδομένα (metadata) και τα οποία ουσιαστικά συντελούν στην απόδοση στην πληροφορία ειδικού νοήματος και μιας σειράς διακριτικών ιδιοτήτων που διαθέτει. Στόχος είναι να μπορούν τα δεδομένα μαζί με την περιγραφή τους να αποθηκευτούν στον υπολογιστή και να μπορούν να αποτελέσουν αντικείμενο επεξεργασίας. Τα μεταδεδομένα λοιπόν είναι «δεδομένα που περιγράφουν δεδομένα» και προσδίδουν σε αυτά την απαιτούμενη σημασιολογία.

Επανερχόμενοι στην έννοια περιεχόμενο, εξετάζουμε τα δύο βασικά διακριτικά χαρακτηριστικά του. Τα χαρακτηριστικά αυτά είναι η μορφοποίησή (format) του και η δομή (structure).

Η μορφοποίηση αφορά τόσο τον τρόπο αποθήκευσης του περιεχομένου στο εκάστοτε φυσικό μέσο (binary format) , όσο και την μορφή παρουσίασης του περιεχομένου προς τα έξω (rendering format). Η μορφοποίηση μπορεί να εφαρμόζεται σε διάφορα επίπεδα εμβέλειας, όπως για παράδειγμα σε επίπεδο χαρακτήρων , παραγράφου , σελίδας ή ολόκληρης έκδοσης. Η παρουσίαση του περιεχομένου μπορεί να είναι εντελώς ανεξάρτητη από το ίδιο το περιεχόμενο για τον δημιουργό και τον χειριστή περιεχομένου , στην περίπτωση όμως του τελικού αποδέκτη ο τρόπος παρουσίασης του περιεχομένου προσδίδει στο περιεχόμενο μια ειδική ταυτότητα. Η ταυτότητα αυτή μπορεί να σχετίζεται με την κουλτούρα του κοινού , με την αναγνωσιμότητα, το συνολικό βαθμό ενδιαφέροντος και την δυνατότητα που υπάρχει για διαχωρισμό των ουσιαστικών από τα επουσιώδη τμήματα μιας ενότητας περιεχομένου. Γι' αυτό και ο τρόπος μορφοποίησης για την παρουσίαση του περιεχομένου κρίνεται ιδιαίτερα σημαντικός .

Η δομή του περιεχομένου τώρα αφορά στο μοντέλο δεδομένων που υιοθετείται για την εσωτερική αναπαράσταση του περιεχομένου σε ένα σύστημα. Το μοντέλο αυτό μπορεί να βασίζεται στον αντικειμενοστρεφή προγραμματισμό (object oriented programming). Το περιεχόμενο μπορεί να διαιρεθεί σε «κλάσεις περιεχομένου», οι οποίες με τη σειρά τους διαθέτουν μια σειρά από υλοποιήσεις, που είναι τα «αντικείμενα περιεχομένου». Τα αντικείμενα περιεχομένου περιέχουν εσωτερικές ιδιότητες και δυνατότητες επικοινωνίας με άλλα αντικείμενα εντός του περιεχομένου ή με εξωτερικά αντικείμενα που επικοινωνούν με το περιεχόμενο. Για την αξιοποίηση της λειτουργικότητας που προσφέρουν τα «αντικείμενα» αυτά δεν είναι αναγκαία η γνώση της εσωτερικής τους υλοποίησης , αλλά μόνο ο τρόπος που αυτά καλούνται και το είδος του περιεχομένου που επιστρέφουν. Με την αλληλεπίδραση μεταξύ των αντικειμένων περιεχομένου διαπιστώνουμε ότι το περιεχόμενο ως έννοια εμπεριέχει και μια συγκεκριμένη προσφερόμενη λειτουργικότητα. Κάθε αντικείμενο περιεχομένου λοιπόν περιέχει πληροφορία και προσφέρει λειτουργικότητα. Στο παρόν κεφάλαιο εξετάζονται αναλυτικότερα τα μοντέλα που ακολουθούνται στην πράξη για την δόμηση του περιεχομένου.

## 2.2 Τι είναι σύστημα διαχείρισης περιεχομένου

Διαχείριση περιεχομένου είναι η διαδικασία που περιλαμβάνει επιμέρους εργασίες, όπως η συλλογή, η εισαγωγή, η οργάνωση, η κατηγοριοποίηση, η δόμηση, η επεξεργασία και η δημοσίευση μιας μεγάλης κλίμακας τύπων περιεχομένου. Παραδείγματα τύπων περιεχομένου είναι το απλό κείμενο, η σταθερή και κινούμενη εικόνα και τα γραφικά.

Κάθε σύστημα διαχείρισης περιεχομένου (content management system, CMS) είναι ένα σύστημα που στοχεύει στην συλλογή, διαχείριση και δημοσίευση του περιεχομένου. Η αποθήκευση του περιεχομένου γίνεται σε φυσικό επίπεδο σε συστατικά στοιχεία που επιλέγονται από το εκάστοτε CMS, όπως είναι για παράδειγμα ένα σύστημα με αρχεία. Το CMS είναι υπεύθυνο για την δημιουργία και διαχείριση δυναμικών δεσμών μεταξύ αυτών των συστατικών, με στόχο να καθίσταται η πρόσβαση στο περιεχόμενο πιο εύκολη και αποδοτική.

Ένα σύστημα διαχείρισης περιεχομένου χρησιμοποιεί τα μεταδεδομένα για να περιγράψει την πληροφορία και έτσι να δημιουργήσει το περιεχόμενο και χρησιμοποιώντας τις ήδη υπάρχουσες τεχνολογίες διαχείρισης δεδομένων μπορεί να συλλέγει, να διαχειρίζεται και να παρουσιάζει περιεχόμενο. Η χρησιμότητα ενός συστήματος διαχείρισης περιεχομένου έγκειται στο ότι προσφέρει τη δυνατότητα για έλεγχο πάνω στην δημιουργία και την διανομή πληροφορίας και λειτουργικότητας. Με την αξιοποίηση ενός CMS μπορούμε να έχουμε καταγεγραμμένο το συνολικό περιεχόμενο που διαθέτουμε και να γνωρίζουμε σε ποιους αποδέκτες και με ποιο τρόπο πρέπει να δημοσιεύσουμε κάποιο τμήμα αυτού του περιεχομένου την κατάλληλη στιγμή. Παράλληλα με τις δυνατότητες συλλογής, διαχείρισης και δημοσίευσης του περιεχομένου, κάθε σύστημα διαχείρισης περιεχομένου συνοδεύεται απαραίτητα από εργαλεία ελέγχου και εποπτείας της όλης διαδικασίας.

Με βάση τα προαναφερθέντα σχετικά με τα χαρακτηριστικά του περιεχομένου (μορφοποίηση και δομή), κάθε σύστημα διαχείρισης περιεχομένου θα πρέπει να ικανοποιεί ορισμένες προδιαγραφές. Ένα αποδοτικό σύστημα διαχείρισης περιεχομένου πρέπει να μεριμνεί ώστε να υπάρχει η δυνατότητα για να δέχεται και να παράγει πολλά διαφορετικά είδη μορφοποίησης σε φυσικό μέσο (binary formats), να μπορεί να αποσυνδέει το rendering format από το ίδιο το περιεχόμενο, αλλά να χρησιμοποιεί για το rendering format κάποιους κανόνες αναγνωσιμότητας και αναγνωρισιμότητας. Ένα επίσης σημαντικό θέμα που πρέπει να λαμβάνεται υπ' όψη κατά τον σχεδιασμό ενός CMS είναι η λειτουργικότητα που αυτό πρέπει να προσφέρει σύμφωνα με τις προδιαγραφές που έχουν τεθεί και η οποία υλοποιεί την επιχειρηματική λογική (business logic) της εκάστοτε εφαρμογής. Περισσότερες πληροφορίες για τα λειτουργικά χαρακτηριστικά ενός CMS μπορεί να βρει ο αναγνώστης σε ακόλουθη παράγραφο του παρόντος κεφαλαίου.

Στην παρούσα εργασία θα μας απασχολήσουν μόνο τα συστήματα διαχείρισης διαδικτυακού περιεχομένου (Web Content Management Systems), τα οποία βρίσκουν ποικίλες εφαρμογές και αξιοποιούνται σε μεγάλο βαθμό για την διαχείριση διαδικτυακού περιεχομένου (υπό μορφή δημόσιου Web site ή Intranet – Extranet) από εταιρείες, οργανισμούς και άλλους φορείς. Συνεπώς σε όποιο σημείο από εδώ και πέρα αναφέρεται ο όρος «Διαχείριση Περιεχομένου» χωρίς να διευκρινίζεται διαφορετικά, θα εννοούμε «Διαχείριση Διαδικτυακού Περιεχομένου» και επίσης με τον όρο «Συστήματα Διαχείρισης Περιεχομένου» θα εννοούμε «Συστήματα Διαχείρισης Διαδικτυακού Περιεχομένου».

Για την εργασία μας θα κάνουμε την ακόλουθη σύμβαση: Ως σύστημα για την διαχείριση διαδικτυακού περιεχομένου (WCMS) θα θεωρήσουμε μια εγκατάσταση η οποία αποτελείται από την εφαρμογή portal (εφαρμογή που βλέπουν και αξιοποιούν οι τελικοί χρήστες ενός web site ή ενός intranet από τα συστήματα φυλλομέτρησης του Ιστού - Web browsers) μαζί με τις πρόσθετες εφαρμογές διαχείρισης που παρέχονται από τον κατασκευαστή, ανάλογα με τις ανάγκες του εκάστοτε τελικού χρήστη. Κάθε πρόσθετη εφαρμογή διαχείρισης απευθύνεται σε κάποια κατηγορία χρηστών του WCMS, όπως τον διαχειριστή περιεχομένου, τον συγγραφέα περιεχομένου, κ.ά. Επίσης να γίνει η επισήμανση ότι κάποια συστήματα μπορεί να προσφέρουν τις εφαρμογές διαχείρισης μέσα από την ίδια την εφαρμογή portal ή σαν ξεχωριστή Web-based εφαρμογή. Αυτό που συνήθως όμως συμβαίνει

στα πραγματικά συστήματα είναι η επιλογή αποδέσμευσης της βασικής εφαρμογής τελικού χρήστη (portal) από τα υπόλοιπα διαχειριστικά modules, γεγονός που συμβάλλει στην ευκολότερη παραμετροποίηση (configuration) και αναβάθμιση (upgrade) του συνολικού συστήματος.

Ακολουθεί η ανάλυση ενός θεωρητικού μοντέλου CMS, ως ένα θεμελιώδες σύστημα διαχείρισης περιεχομένου. Κάθε πραγματικό σύστημα διαχείρισης περιεχομένου και οι εφαρμογές που αναπτύσσονται με αυτό θα πρέπει να συγκλίνουν, σε ότι αφορά στην σχεδίασή τους, με το μοντέλο που περιγράφουμε στην συνέχεια και το οποίο προκύπτει τόσο από την μελέτη των βασικότερων standard σχεδιαστικών αρχών που επικρατούν στην διεθνή βιβλιογραφία και σε αναφορές από το Διαδίκτυο, όσο και από την σύγκριση των επιμέρους πραγματικών CMS, με στόχο την εύρεση των κοινών χαρακτηριστικών τους.

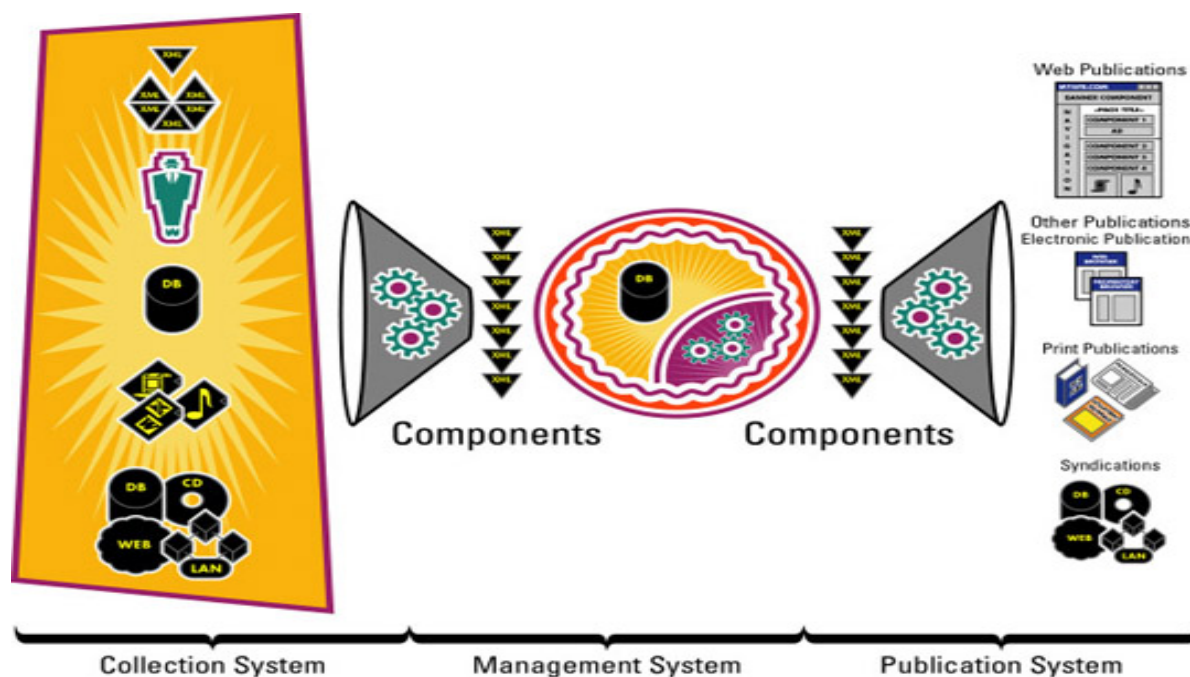
## 2.3 Υποσυστήματα σε ένα CMS

Ένα σύστημα διαχείρισης περιεχομένου είναι ένα ολοκληρωμένο σύστημα υλικού και λογισμικού, το οποίο είναι υπεύθυνο για την συλλογή, διαχείριση και δημοσίευση περιεχομένου. Με βάση αυτή τη λογική μπορούμε να θεωρήσουμε και την βασική διάρθρωση ενός CMS στα τρία αυτά βασικά του υποσυστήματα, δηλαδή:

- το υποσύστημα συλλογής (collection)
- το υποσύστημα αποθήκευσης - επεξεργασίας - ελέγχου ροής του περιεχομένου (core management) και
- το υποσύστημα δημοσίευσης (publishing)

Σε κάθε ένα από τα βασικά αυτά τα υποσυστήματα βρίσκονται λειτουργικές μονάδες ή οντότητες, οι οποίες αλληλεπιδρούν και κάθε φορά παράγουν το ζητούμενο αποτέλεσμα. Να σημειωθεί ότι πολλές φορές μεταξύ των τριών αυτών τμημάτων ενδέχεται να υπάρχουν αλληλοκαλύψεις όσον αφορά στην φυσική τους υλοποίηση, π.χ. το υποσύστημα δημοσίευσης σε ένα web site παράγει φόρμες για να συμπληρωθούν από τους χρήστες με κατάλληλο περιεχόμενο, το οποίο όμως αποτελεί είσοδο για το υποσύστημα συλλογής του CMS. Μερικά μάλιστα από τα προϊόντα που κυκλοφορούν στην αγορά παρέχουν τη δυνατότητα μόνο για αποθήκευση σε repository και επεξεργασία του περιεχομένου, θεωρώντας δεδομένες τις λειτουργίες συλλογής και δημοσίευσης. Ωστόσο ένα πλήρες CMS σύστημα περιλαμβάνει την υλοποίηση σε ένα πακέτο και των τριών ανωτέρω υποσυστημάτων.

Στο παρακάτω σχήμα φαίνεται ένα ενδεικτικό σχήμα του πως μπορεί κανείς να θεωρήσει την συνολική δομή ενός τυπικού CMS.



Σχήμα 1: Υποσυστήματα σε ένα CMS [BOI02]

### 2.3.1 Το υποσύστημα collection

Αυτό είναι το τμήμα που παρέχει όλες τις δυνατότητες μετατροπής των δεδομένων που εισέρχονται στο σύστημα σε περιεχόμενο κατάλληλης μορφής για δημοσίευση, μέσω υπηρεσιών λογισμικού που ονομάζονται υπηρεσίες συλλογής (collection services). Οι πηγές περιεχομένου ποικίλλουν.

- Κατά πρώτο λόγο μπορεί να έχουμε συγγραφείς/δημιουργούς (authors) περιεχομένου που παράγουν το περιεχόμενο με οργανωμένο τρόπο αλλά σε πολλές διαφορετικές μορφές και απευθύνονται συνήθως στις ανάγκες ενός συγκεκριμένου συστήματος CM . Η δημιουργία του περιεχομένου από τους δημιουργούς ξεκινά από το μηδέν. Οι δημιουργοί τροφοδοτούν το CMS με περιεχόμενο μέσα από περιβάλλον που τους παρέχεται από το υποσύστημα συλλογής περιεχομένου. Μέσα από το κατάλληλο κάθε φορά περιβάλλον , ο δημιουργός καθοδηγείται με την παροχή των κατάλληλων templates, την αυτόματη συμπλήρωση χρήσιμων πληροφοριών από το σύστημα, όπως ημερομηνία παραγωγής ή το όνομα του δημιουργού, κτλ. Επίσης το σύστημα βοηθά τον δημιουργό με το να του παρέχει με οπτικά εργαλεία έλεγχο των εκδόσεων του περιεχομένου που παράγει (version control) και έλεγχο ροής της εργασίας του (workflow). Ωστόσο η δημιουργία περιεχομένου με αυτόν τον τρόπο για την εισαγωγή του σε ένα CMS είναι κατά κύριο λόγο χειρωνακτική και όχι αυτόματη διαδικασία. Η παραγόμενη πληροφορία είναι μικρή σε όγκο αλλά έχει υψηλή αξιοπιστία και ποιότητα.
- Κατά δεύτερο λόγο και παράλληλα με τους δημιουργούς μπορεί να έχουμε από το σύστημα και συλλογή ήδη έτοιμου περιεχομένου από υπάρχουσες εξωτερικές πηγές (διαδικασία acquisition). Το περιεχόμενο αυτό σε αντίθεση με το περιεχόμενο που προέρχεται από τους δημιουργούς δεν έχει δημιουργηθεί ειδικά για το συγκεκριμένο CMS που μας ενδιαφέρει. Οι υπάρχουσες πηγές εδώ είναι συνήθως τα syndications και τα διάφορα ήδη αρχείων υπολογιστών. Η παραγόμενη πληροφορία εδώ είναι μεγάλη σε όγκο , αλλά συνήθως χαμηλή σε ποιότητα.

Τα syndications είναι δημοσιεύσεις με πληροφορίες που μεταδίδονται μαζικά και οργανωμένα με τη χρήση κάποιου αναγνωρισμένου πρωτοκόλλου. Στις πληροφορίες αυτές έχουν εισαχθεί μεταδεδομένα (metadata) για τον καλύτερο χαρακτηρισμό τους και την πιο αποδοτική επεξεργασία τους. Περισσότερες πληροφορίες για την λειτουργία του syndication μπορούν να αναζητηθούν στο Κεφάλαιο 5 της παρούσας εργασίας.

Τα αρχεία υπολογιστών μπορούν να προκύψουν είτε από εργασία με εφαρμογές είτε από ψηφιοποίηση αναλογικών δεδομένων, όπως για παράδειγμα εικόνα, κείμενο, video και ήχος.

Μετά την καθ' αυτή συλλογή του περιεχομένου από τις πηγές (sources), το υποσύστημα συλλογής διαθέτει δυνατότητες conversion , δηλαδή απαλοιφής μη απαραίτητων τμημάτων πληροφορίας καθώς και κατάλληλης μετατροπής της μορφής και της δομής του περιεχομένου, που όπως είδαμε είναι τα δύο βασικά χαρακτηριστικά κάθε είδους περιεχομένου. Η μετατροπή αυτή είναι αναγκαία για λόγους συμβατότητας με την υπάρχουσα δομή ή μορφοποίηση του εκάστοτε CMS. Ολοένα και περισσότερα συστήματα διαχείρισης περιεχομένου υποστηρίζουν την δυνατότητα μετατροπής πολλών διαφορετικών τύπων περιεχομένου, όπως για παράδειγμα διαφορετικά είδη αρχείων .doc , .xls , .pdf κ.ά. , σε μορφή XML πριν την εισαγωγή στο σύστημα.

Η επεξεργασία του εισερχόμενου περιεχομένου καταλήγει στην διαδικασία του aggregation, κατά την οποία το συλλεχθέν περιεχόμενο περνά από μια σειρά από τροποποιήσεις που έχουν να κάνουν:

- Με την το στυλ και την εμφάνισή του  
Οι κανόνες στυλ και εμφάνισης αφορούν θέματα σχετικά με τη μορφή που πρέπει να λάβει το περιεχόμενο πριν από την έκδοσή του , για παράδειγμα την σωστή γραμματική , τη σύνταξη , τον τονισμό. Άλλο σχετικό παράδειγμα είναι κανόνες σχετικοί με το κοινό στο οποίο απευθύνεται το περιεχόμενο και αφορούν το επίπεδο και το λεξιλόγιο της γλώσσας, το

πρόσωπο στο οποίο γράφονται οι προτάσεις (ενικός-πληθυντικός). Τέλος άλλο παράδειγμα αφορά τον συσχετισμό του περιεχομένου με το περιβάλλον δημοσίευσής του, δηλαδή όλα τα στοιχεία εκείνα της εκάστοτε παρουσίας (χρώματα, στοίχιση, τονισμός παραγράφων, εικόνες, πλαίσια διαλόγου σε μια ιστοσελίδα) που πρέπει να ληφθούν υπ' όψη ώστε το περιεχόμενο να είναι δοσμένο με εύληπτο τρόπο. Φαίνεται δηλαδή ότι με τους κανόνες αυτούς εξασφαλίζεται η ομοιομορφία της εμφάνισης και του στυλ του περιεχομένου σε όλη την έκτασή του και είναι μια διαδικασία γενικά μη αυτόματη, που απαιτεί αρκετή πείρα και ανθρώπινη ικανότητα.

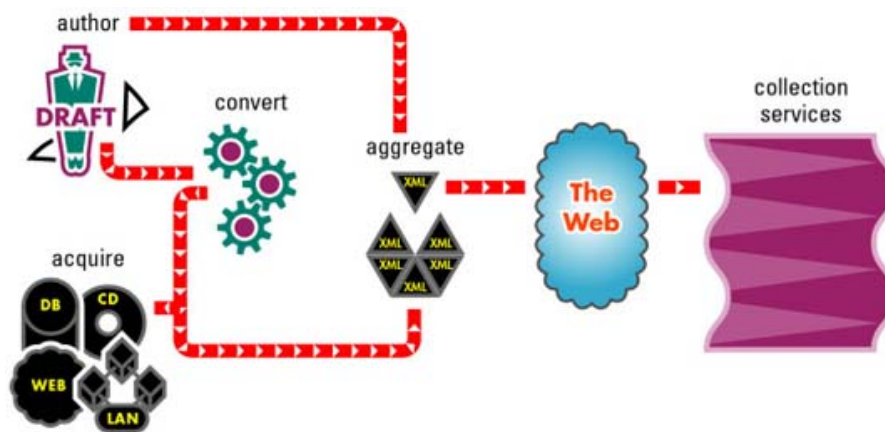
- Την διαίρεσή του σε κατάλληλα διαχειρίσιμα επιμέρους συστατικά στοιχεία (components)  
Η διαίρεση του περιεχομένου σε επιμέρους συστατικά τμήματα, γίνεται ανάλογα με την μορφή στην οποία βρίσκεται το περιεχόμενο πριν παραληφθεί με την διαδικασία της συλλογής (collection) . Οι συνήθεις μορφές είναι τα binary αρχεία ή unicode αρχεία κειμένου, οι βάσεις δεδομένων (σχεσιακές και αντικειμενοστρεφείς) και τα xml αρχεία (στα οποία υπάρχει ήδη μια οργάνωση με βάση τα tags). Στην περίπτωση των unicode και binary αρχείων μια οντότητα που θα τοποθετηθεί στο CMS μπορεί να είναι ήδη διαχωρισμένη σε ένα ξεχωριστό αρχείο ή μπορεί να βρίσκεται στο ίδιο αρχείο μαζί με άλλες οντότητες. Μπορούμε για παράδειγμα να φανταστούμε ένα αρχείο που αποτελεί τον τιμοκατάλογο των προϊόντων μιας εταιρείας, στον οποίο τα διάφορα προϊόντα ξεχωρίζουν από τον τονισμένο τίτλο του ονόματος κάθε προϊόντος ή από την τοποθέτηση κάθε εγγραφής προϊόντος σε ξεχωριστή σελίδα.
- Την προσθήκη των κατάλληλων μεταδεδομένων (metadata) που θα το κάνουν λειτουργικό με βάση τους κανόνες του CMS. Με την εισαγωγή περιεχομένου στο σύστημα πρέπει να παρέχονται λειτουργίες προσάρτησης των κατάλληλων κάθε φορά μεταδεδομένων (metadata) στο περιεχόμενο με κύριους στόχους:
  - Την πιο αποδοτική μετέπειτα αναζήτηση του περιεχομένου
  - Την διαχείριση του περιεχομένου με βάση κάποιους συγκεκριμένους κανόνες για την διευκόλυνση λειτουργιών σε μια εφαρμογή , π.χ. έλεγχος πρόσβασης
  - Την διαχείριση του περιεχομένου με βάση πληροφορίες κατάστασης (π.χ. που χαρακτηρίζουν την εξέλιξη του στον κύκλο ζωής του)Τα μεταδομένα που προσαρτώνται στο περιεχόμενο μπορούν να ανήκουν είτε σε ένα σύνολο προκαθορισμένων από την εφαρμογή μεταδεδομένων είτε να ορίζονται επιτόπου από τον δημιουργό του. Η κατάλληλη επιλογή των μεταδεδομένων συμβάλλει έτσι και στην επέκταση της συνολικής λειτουργικότητας και των χαρακτηριστικών μιας εφαρμογής διαχείρισης.

Στο τέλος της διαδικασίας του aggregation γίνεται εισαγωγή ενός component περιεχομένου στο repository, το οποίο είναι η «αποθήκη» περιεχομένου που αποτελεί και την βάση αποθήκευσης για το σύστημα. Η εισαγωγή ενός component στο repository μπορεί να επιτευχθεί μέσω:

- των Web forms. Με τις ηλεκτρονικές αυτές φόρμες εισάγονται από τον δημιουργό περιεχομένου απευθείας δεδομένα, όπως κείμενο, εικόνες, ήχος, video καθώς και σχετικά κάθε φορά μεταδεδομένα. Κάθε component περιεχομένου πρέπει να ανήκει σε μια κατηγορία (component class) με βάση την ιεράρχηση που έχει γίνει από το CMS (βλέπει παρακάτω: “λειτουργικά χαρακτηριστικά ενός CMS”). Ο σχεδιασμός της μορφής και της λειτουργικότητας των Web forms είναι μια δυνατότητα που παρέχουν τα περισσότερα CMS. Ο σχεδιασμός αυτός μπορεί να γίνεται είτε χειρωνακτικά είτε και αυτόματα, με την ανάγνωση ενός σχήματος ΒΔ ή ενός DTD ή CSS (για τα DTD και CSS βλέπε Κεφάλαιο 4 παρούσας εργασίας) . Μετά την εισαγωγή των στοιχείων στην Web form από τον δημιουργό , το CMS δημιουργεί μια εγγραφή ΒΔ ή ένα στοιχείο XML στο repository , ώστε να αναπαραστήσει το εκάστοτε component.
- Διαφόρων εξωτερικών βοηθητικών προγραμμάτων editors, όπως για παράδειγμα το MS Word , MS Excel, MS Access, Paint Shop Pro, που βοηθούν τον δημιουργό περιεχομένου να παράγει περιεχόμενο, βασισμένος σε συμβάσεις και αυστηρούς κανόνες δομής και εμφάνισης που έχουν τεθεί από τον διαχειριστή του CMS.



Σχηματικά η δομή ενός τυπικού υποσυστήματος συλλογής περιεχομένου φαίνεται παρακάτω στο σχήμα 2.



Σχήμα 2: Υποσύστημα collection [BOI02]

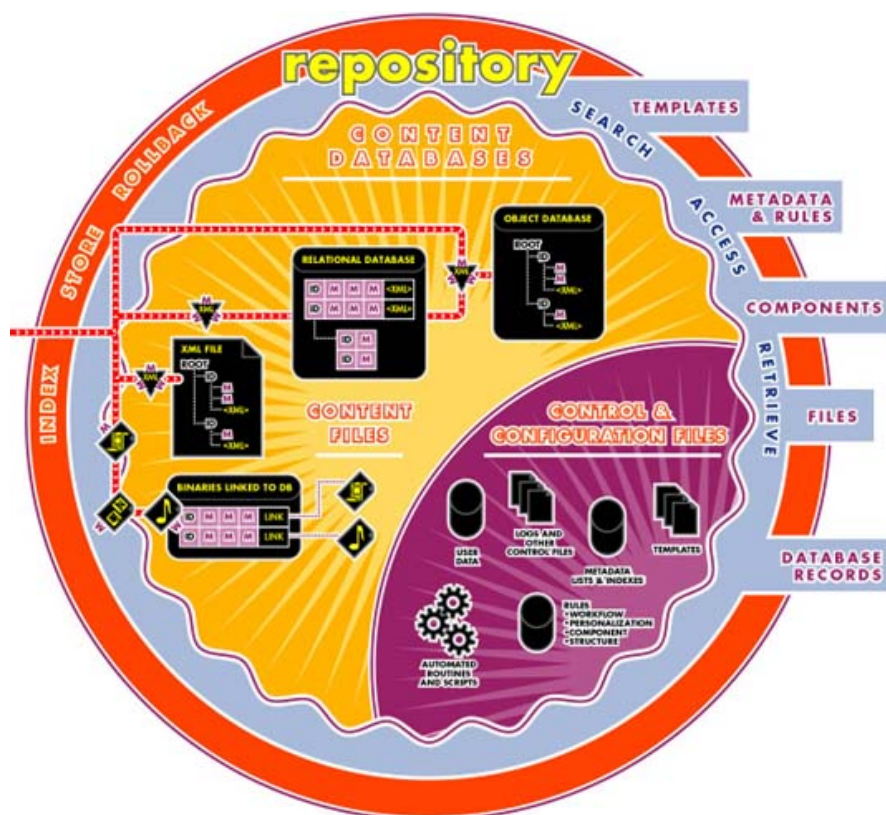
Ένας από τους στόχους της παρούσας εργασίας είναι να διερευνήσει τις δυνατότητες εφαρμογής στο πεδίο του aggregation της τεχνολογίας των Web Services. Θέλουμε να διαπιστώσουμε κατά πόσον μπορεί να υπάρξει ένα κοινό αποδεκτό πρότυπο και να ενισχυθεί η κοινότητα των μαζικών ηλεκτρονικών εκδόσεων για την συλλογή περιεχομένου. Για τα θέματα αυτά μπορεί ο αναγνώστης να ανατρέξει στα κεφάλαιο 5 και 6 της παρούσας εργασίας.

### 2.3.2 Υποσύστημα (core) management

Αυτό το υποσύστημα παρέχει μηχανισμούς για την αποθήκευση περιεχομένου στο repository. Μέσα στο υποσύστημα αυτό εμπεριέχεται και η δυνατότητα συνολικής διαχείρισης όλων των υποσυστημάτων και του περιεχομένου ενός CMS.

Το περιεχόμενο στην φυσική μορφή του είναι πιθανώς ένα σύνολο από βάσεις δεδομένων, καταλόγους αρχείων και άλλες δομές αποθήκευσης, στις οποίες αποθηκεύεται τόσο το περιεχόμενο του CMS με τα μεταδεδομένα του, όσο και άλλα δεδομένα που σχετίζονται με την λειτουργία του CMS, όπως αρχεία συστήματος (system control and configuration files). Τέλος το repository πρέπει να μπορεί να επικοινωνεί με τα με τα υποσυστήματα collection και publishing. Αυτό επιτυγχάνεται με ένα σύνολο από υπηρεσίες συλλογής και δημοσίευσης (collection και publishing services αντίστοιχα). Οι υπηρεσίες αυτές είναι μέθοδοι υλοποιημένες σε κάποια γλώσσα προγραμματισμού και οι οποίες δρουν ως ο συνδετικός κρίκος μεταξύ των υποσυστημάτων.

Μπορούμε να φανταστούμε ένα repository σχηματικά όπως φαίνεται παρακάτω:



Σχήμα 3: CMS Repository [BOI02]

#### Σχετικά με τον λογικό σχεδιασμό του repository

Ένα σύστημα διαχείρισης περιεχομένου οφείλει να προσφέρει ένα όσο το δυνατόν πιο αυτόνομο μοντέλο δεδομένων, το οποίο να μπορεί να προσαρμόζεται στις ανάγκες της εφαρμογής, στην οποία ενσωματώνεται κάθε φορά το CMS.

Το πιο εύληπτο ιεραρχικό μοντέλο αναπαράστασης του περιεχομένου είναι αυτό μιας δενδρικής δομής, όπου σε κάθε κόμβο τοποθετείται μια κλάση συστατικών περιεχομένου. Το πραγματικά διαχειρίσιμο περιεχόμενο βρίσκεται με την μορφή συστατικών περιεχομένου (components) στα φύλλα του δέντρου. Κάθε component κληρονομεί την λειτουργικότητα όλων των προγόνων – υπερκλάσεων. Αυτός ο τρόπος ιεραρχικής οργάνωσης του συνολικού περιεχομένου σε μικρότερες λογικά

σχετιζόμενες μεταξύ λειτουργικές μονάδες αποτελεί την υλοποίηση μιας ταξινόμιας (taxonomy). Κάθε CMS υλοποιεί το δικό του taxonomy.

Στο εσωτερικό τώρα κάθε component βρίσκονται στοιχεία περιεχομένου (elements) . Είναι τα στοιχειώδη τμήματα περιεχομένου στα οποία μπορεί να διαιρεθεί ένα component. Για παράδειγμα, αν το component είναι ένας κατάλογος προϊόντος μιας εταιρείας, τότε elements μπορούν να θεωρηθούν το όνομά του, η περιγραφή του, μια πιθανή φωτογραφία του, κτλ. Τα στοιχεία (elements) σε ένα component μπορούν να διακριθούν σε αυτά τα οποία προβάλλονται από τους μηχανισμούς παρουσίασης του CMS (body elements) και σε αυτά τα οποία υπάρχουν μόνο για λόγους διαχείρισης του ίδιου του περιεχομένου (management elements).

### *Σχετικά με την φυσική υλοποίηση του repository*

Μπορεί να ανήκει σε μια από τις παρακάτω τρεις μεγάλες κατηγορίες:

- Αντικειμενοστρεφές repository (object oriented)  
Το περιεχόμενο εδώ είναι οργανωμένο σε αρχεία XML που αντιπροσωπεύουν τα αντικείμενα περιεχομένου. Για την ανάκτηση και επεξεργασία του περιεχομένου μπορούν να χρησιμοποιηθούν XML πρότυπα από το W3C, όπως είναι η γλώσσα XSL. Για την XSL βλέπε στο Κεφάλαιο 4.
- Repository με σχεσιακή βάση δεδομένων (relational database)  
Σε αυτή τη μορφή έχουμε αποθήκευση μεταδεδομένων XML ως πεδία (fields) στις εγγραφές (records) των πινάκων της βάσης. Η διαχείριση του περιεχομένου που είναι αποθηκευμένο σε πίνακες γίνεται συνήθως μέσω του DOM (Document Object Model) API. Για περισσότερες πληροφορίες για το DOM, ο αναγνώστης μπορεί να ανατρέξει στο Κεφάλαιο 4 της παρούσας εργασίας.
- Repository με σύστημα αρχείων (file system)  
Σε αυτή τη μορφή το περιεχόμενο είναι αποθηκευμένο υπό την αρχική μορφή των αρχείων, όπως είναι .doc, .xls, .pdf, .txt . Κάποια CMS δημιουργούν ένα αντίγραφο σε XML για κάθε αρχείο. Κάποια άλλα CMS διατηρούν μόνο την αρχική μορφή των αρχείων και περιβάλλουν κάθε αρχείο με μεταδεδομένα (metadata wrappers). Η πρόσβαση στο περιεχόμενο μπορεί στις περιπτώσεις αυτές να γίνει με την βοήθεια της XML και του DOM API.

### *Σύγκριση των 3 φυσικών υλοποιήσεων ενός CMS repository*

Η υλοποίηση του αντικειμενοστρεφούς repository θεωρείται η βέλτιστη από πλευράς ενσωμάτωσης της XML. Αυτή η ενσωμάτωση συνεπάγεται πρόσβαση σε μια πληθώρα προγραμματιστικών εργαλείων για πρόσβαση στο περιεχόμενο και εύκολη διαχείρισή του. Ωστόσο η αντικειμενοστρεφής λύση υστερεί σε επιδόσεις και αξιοπιστία. Στον τομέα αυτό υπερτερούν οι άλλες δύο φυσικές υλοποιήσεις, οι οποίες έχουν και μεγαλύτερη εμπορική απήχηση. Η πλειοψηφία των CMS σήμερα βασίζεται σε κάποιο Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS) για την φυσική αποθήκευση του περιεχομένου. Προβλέπεται ότι μελλοντικά θα επικρατήσουν οι υβριδικές λύσεις, οι οποίες θα ενσωματώνουν θετικά χαρακτηριστικά και από τις τρεις υπάρχουσες υλοποιήσεις.

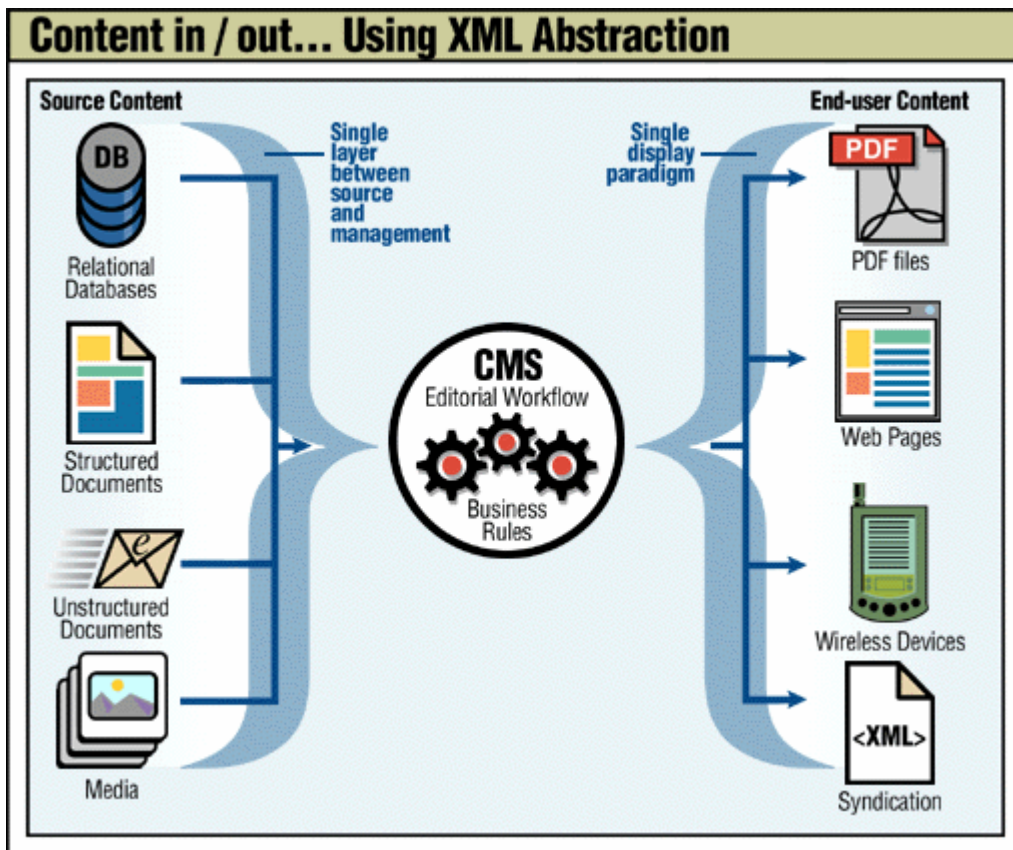
### 2.3.3 Υποσύστημα publishing

Αυτό είναι υπεύθυνο για την μαζική δημοσίευση και παρουσίαση του περιεχομένου σε διαφορετικές μορφές και ακολουθώντας κάθε φορά το κατάλληλο template , ανάλογα με τις ανάγκες της δημοσίευσης. Τα templates είναι ειδικά προγράμματα , βασισμένα σε κάποια γλώσσα προγραμματισμού, όπως για παράδειγμα τα πακέτα ASP/COM, JSP/J2EE, XSL ή άλλες proprietary λύσεις). Τα templates στοχεύουν στην δημιουργία και διαχείριση των ειδικών αρχείων templates , με βάση τα οποία γίνεται κάθε φορά η δημοσίευση του περιεχομένου.

Το κάθε αρχείο template περιλαμβάνει τα εξής στοιχεία:

- Τα στατικά στοιχεία , όπως κείμενο , multimedia υλικό και scripts, τα οποία περνούν από το repository κατευθείαν στην δημοσίευση, χωρίς περαιτέρω επεξεργασία
- Κλήσεις σε υπηρεσίες δημοσίευσης (publishing services) που είναι ενσωματωμένες στο CMS. Οι υπηρεσίες αυτές αφορούν για παράδειγμα στην προσωποποίηση (personalization) και στην προσθήκη δυνατοτήτων πλοήγησης μεταξύ των επιμέρους συστατικών του περιεχομένου (navigation) σε ένα website, με χρήση scripts.
- Κλήσεις σε υπηρεσίες δημοσίευσης που δεν είναι ενσωματωμένες στο τρέχον CMS , αλλά προσφέρονται από άλλους φορείς. Τέτοιες υπηρεσίες μπορεί να είναι μέθοδοι που καλούνται με Remote Procedure Call (RPC) ή Web Services που καλούνται με SOAP. Σχετικά παραδείγματα συναντάμε συχνά σε Enterprise Resource Planning (ERP) και Data Mining εφαρμογές.

Τα αρχεία templates ενεργοποιούνται κάθε φορά κατάλληλα , ανάλογα με το είδος της δημοσίευσης που θέλουμε να παράξουμε. Στις υπηρεσίες δημοσίευσης (publishing services) ενσωματώνεται ένα σημαντικό μέρος από την επιχειρηματική λογική (business logic – application logic) της κάθε εφαρμογής.

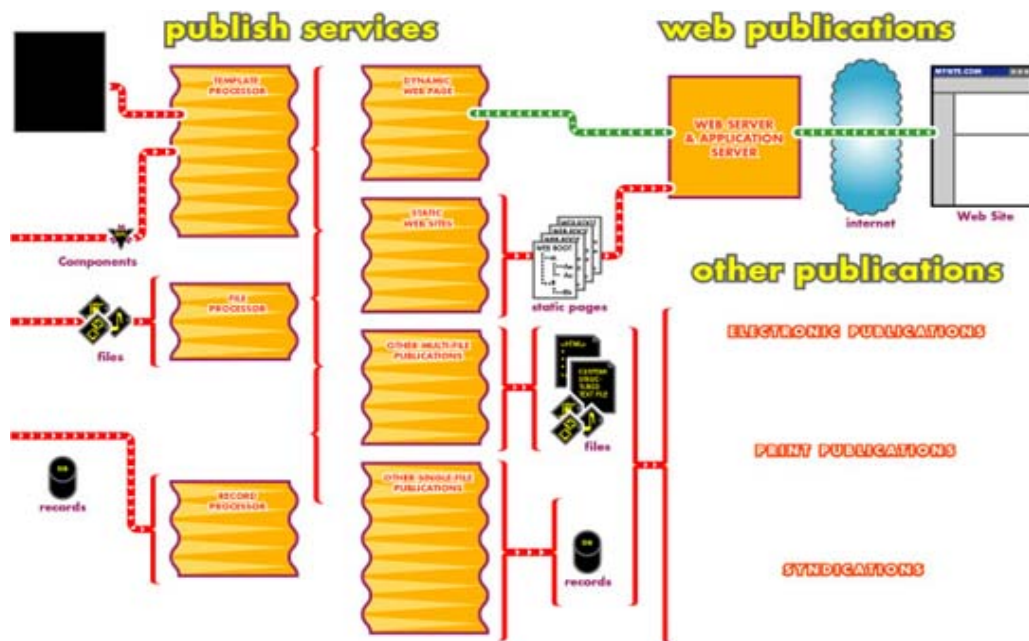


Σχήμα 4: Διάφορες μορφές publications

Όπως φαίνεται και από το σχήμα 4, οι πιο συνηθισμένες κατηγορίες δημοσιεύσεων που συναντάμε στην πράξη είναι:

- **Web εκδόσεις**  
Είναι η πιο συνηθισμένη μορφή εκδόσεων και αυτή που υποστηρίζει η πλειοψηφία των εμπορικών και open-source προϊόντων διαχείρισης περιεχομένου. Αυτές αφορούν την δημιουργία στατικών ή δυναμικών Internet, intranet και extranet sites. Στην περίπτωση των στατικών sites, οι σελίδες παράγονται όλες μαζί αυτόματα από τα publishing services, με βάση κάποιους κανόνες παρουσίασης (triggers) που μπορεί να ορίσει ο διαχειριστής του CMS. Στην περίπτωση όμως των δυναμικών sites, οι υπηρεσίες δημοσίευσης (publishing services) του CMS εκτελούνται συνήθως στο ίδιο μηχάνημα με τον Web Server. Λαμβάνοντας ένα HTTP Request σήμα από τον χρήστη, περνούν στις υπηρεσίες δημοσίευσης οι κατάλληλες παράμετροι και έτσι το περιεχόμενο που βρίσκεται στο repository στέλνεται στον browser του χρήστη με την κατάλληλη μορφοποίηση και δομή. Συνεπώς οι σελίδες στην περίπτωση αυτή παράγονται δυναμικά.
- **Έντυπες εκδόσεις**  
Εκδόσεις που εκτυπώνονται μαζικά από επιτραπέζια (desktop publishing) ή βιομηχανικής κλίμακας συστήματα εκτύπωσης, αφού όμως πρώτα περάσουν μια ενδιάμεση μορφή ως εξειδικευμένα αρχεία. Παραδείγματα είναι τα αρχεία QuarkXPress, FrameMaker και Adobe Acrobat (.pdf) που αποτελούν de facto πρότυπα της βιομηχανίας των εκδόσεων.
- **Ηλεκτρονικές εκδόσεις**  
Εδώ εντάσσονται όλες οι εκδόσεις που κυκλοφορούν με την μορφή οπτικών δίσκων (CD-DVD) ή άλλων μέσων μαζικής αποθήκευσης. Σε μια τέτοια δημοσίευση μπορεί να ενσωματωθούν διαφημιστικό περιεχόμενο και πιθανώς mirrors από web sites για προωθητικούς λόγους. Παραδείγματα αυτής της μορφής εκδόσεων περιλαμβάνουν αρχεία e-mail, τα Microsoft Help Files (.chm) καθώς και διάφορα format εκδόσεων (WAP, WML) που σχεδιάζονται για χρήση με φορητές και ασύρματες συσκευές (PDAs)
- **Syndications**  
Είναι η μαζική δημοσίευση περιεχομένου με στόχο αυτό να επαναχρησιμοποιηθεί από άλλα μεγάλα συστήματα CM. Η δημοσιοποίηση γίνεται με χρήση κατάλληλων πρωτοκόλλων, τα οποία βασίζονται κυρίως στο πρότυπο της γλώσσας XML. Σε μερικά συστήματα μάλιστα παρέχεται η δυνατότητα να μετατρέπουν το περιεχόμενο που έχουν σε μορφή XML με κάποιο συνοδευτικό DTD σε περιεχόμενο πάλι μορφής XML αλλά διαφορετικού DTD. Περισσότερες πληροφορίες για τα XML syndications μπορούν να βρεθούν στο Κεφάλαιο 5, ενώ για τα DTD στο Κεφάλαιο 4.

Συνοψίζοντας, η δομή ενός τυπικού υποσυστήματος δημοσίευσης περιεχομένου φαίνεται στο σχήμα 5.



Σχήμα 5 Σχηματική αναπαράσταση υποσυστήματος δημοσίευσης περιεχομένου [BOI02]

## 2.4 Οντότητες (entities) ενός CMS

Σκεπτόμενοι αφαιρετικά, μπορούμε να αντιληφθούμε ένα σύστημα διαχείρισης περιεχομένου όχι μόνο με βάση το κλειστό σύστημα του σχήματος 1 (με βάση τα υποσυστήματα collection -> core management -> publishing), αλλά και ως ένα σύνολο αλληλεπιδρώντων οντοτήτων που σχετίζονται με τον κύκλο ζωής (lifecycle) ενός CMS, με βάση την φιλοσοφία στο προγραμματιστικό αντικειμενοστρεφές μοντέλο.

Οι οντότητες που συναντάμε σε μια εφαρμογή WCMS είναι οι ακόλουθες:

1. Χρήστες και ομάδες χρηστών
2. Ρόλοι οριζόμενοι για χρήστες / ομάδες χρηστών.
3. Στοιχεία περιεχομένου (πόροι)
4. Δομές οργάνωσης περιεχομένου (κόμβοι)
5. Δομές παρουσίασης περιεχομένου, όπως για παράδειγμα web σελίδες και φόρμες
6. Δομές εισαγωγής περιεχομένου στην εφαρμογή, όπως για παράδειγμα φόρμες
7. Ροές διαχείρισης εργασιών στο σύστημα διαχείρισης περιεχομένου (workflows)
8. Αντικείμενα προβολής, όπως για παράδειγμα τα banners
9. Αντικείμενα διαδικτυακών κοινοτήτων (online communities)

Η παραπάνω κατηγοριοποίηση προκύπτει από την μελέτη των χαρακτηριστικών μιας σειράς συστημάτων διαχείρισης περιεχομένου τόσο εμπορικών όσο και open-source. Η μελέτη των τεχνικών προδιαγραφών (whitpapers, datasheets) μας οδήγησε στην κατηγοριοποίηση αυτή ως το πιο αντιπροσωπευτικό σχήμα παρουσίασης οντοτήτων στο μοντέλο δεδομένων.

Κάθε μια από τις οντότητες αυτές αντιστοιχεί σε μια ένα σύνολο υλοποιημένων κλάσεων στην εφαρμογή διαχείρισης περιεχομένου, καθεμιά με τις δικές τις προσφερόμενες δημόσιες (public) μεθόδους. Για παράδειγμα η οντότητα χρήστες/ομάδες χρηστών μπορεί να αντιστοιχεί στο σύνολο των κλάσεων

{ Δημιουργός περιεχομένου, Διαχειριστής περιεχομένου, Διαχειριστής χρηστών, Διαχειριστής εφαρμογής διαχείρισης περιεχομένου, Τελικός χρήστης }. Η επέκταση μιας εφαρμογής έγκειται στην τροποποίηση και επαναχρησιμοποίηση ορισμένων από τις κλάσεις αυτές για κάθε οντότητα ή ακόμα και στην δημιουργία νέων υποκλάσεων τους από την αρχή, ανάλογα με τις απαιτήσεις.

Ένα ακόμα θέμα που πρέπει να ληφθεί υπ' όψη και το οποίο αποτελεί μια ανεξάρτητη λειτουργία μέσα σε ένα σύστημα διαχείρισης περιεχομένου είναι η απόδοση συγκεκριμένων ρόλων σε συγκεκριμένους χρήστες ή ομάδες και για συγκεκριμένους πόρους περιεχομένου. Αυτό ονομάζεται διαχείριση της πρόσβασης στο περιεχόμενο.

Ένας από τους στόχους της εργασίας αυτής είναι να γίνει μια καλή προσέγγιση και καταγραφή των περισσότερων από τα λειτουργικά χαρακτηριστικά που είναι κοινά ανάμεσα σε όλα τα συστήματα διαχείρισης περιεχομένου. Στην επόμενη παράγραφο θα γίνει, με κατάλληλη ομαδοποίηση, μια παρουσίαση της λίστας των βασικών, επιθυμητών και προχωρημένων λειτουργικών χαρακτηριστικών που συναντάμε στις περισσότερες εφαρμογές WCMS.

## 2.5 Λειτουργικά χαρακτηριστικά CMS

Με τον όρο λειτουργικά χαρακτηριστικά εννοούμε το σύνολο των δυνατοτήτων που προσφέρει μια πλατφόρμα CMS για την διαχείριση των οντοτήτων της, όπως αυτές παρουσιάστηκαν στην προηγούμενη ενότητα.

Οι δυνατότητες αυτές μπορεί να απευθύνονται τόσο στον διαχειριστή του περιεχομένου όσο και σε άλλους χρήστες της πλατφόρμας. Για κάθε CMS μπορούμε να θεωρήσουμε ένα ελάχιστο σύνολο απαιτούμενων λειτουργικών χαρακτηριστικών, ώστε να μπορεί το σύστημα να χαρακτηρίζεται στοιχειωδώς CMS. Κάθε σοβαρή υλοποίηση οφείλει να διαθέτει τα χαρακτηριστικά αυτά και να προσθέτει σε αυτά και άλλα χαρακτηριστικά που τα αναβαθμίζουν και διευκολύνουν τα έργα του διαχειριστή περιεχομένου και επίσης αυξάνουν τις συνολικές δυνατότητες του συστήματος. Τα πρόσθετα αυτά χαρακτηριστικά είναι βεβαίως επιθυμητά. Τέλος σε περισσότερο ακριβά και εξελιγμένα CMS διατίθενται προηγμένα χαρακτηριστικά, τα οποία διευκολύνουν μάλλον συγκεκριμένες ενέργειες διαχείρισης ή και πρόσβασης / χρήσης του συστήματος από όλο το εύρος των πιθανών χρηστών του, εξυπηρετώντας τις ανάγκες σχετικών επιχειρήσεων και οργανισμών. Σε τέτοια εξελιγμένα συστήματα υπάρχουν δυνατότητες που δεν περιορίζονται μόνο σε λειτουργίες συλλογής, διαχείρισης και δημοσίευσης περιεχομένου και λειτουργικότητας, αλλά προσφέρουν και πιο προχωρημένες υπηρεσίες που σχετίζονται με τις αυξημένες επιχειρηματικές ανάγκες ενός οργανισμού, όπως για παράδειγμα δυνατότητα δημιουργίας και διαχείρισης συστημάτων ηλεκτρονικών δημοσκοπήσεων για εταιρείες στατιστικής ή εξειδικευμένες δυνατότητες δημιουργίας και διαχείρισης μεγάλων ιστοτόπων (web sites).

Ωστόσο υπάρχουν και πολλά πακέτα τα οποία προσφέρουν ένα μέρος μόνο των συνολικών δυνατοτήτων που διαθέτει ένα θεωρητικά ολοκληρωμένο CMS, με στόχο την πληρέστερη κάλυψη μιας συγκεκριμένης κατηγορίας αναγκών των πελατών που απευθύνονται στα προϊόντα αυτά, με στόχο τόσο την απλότητα όσο και το χαμηλότερο κόστος, ειδικά αν λάβουμε υπ' όψη τον μεγάλο ανταγωνισμό που υπάρχει στην αγορά όπου οι μεγάλοι παίκτες προωθούν και υποστηρίζουν ολοκληρωμένα προϊόντα. Κυκλοφορούν επομένως προϊόντα τα οποία υπό τον γενικό τίτλο CMS είναι στην ουσία προϊόντα π.χ. μόνο για διαχείριση αρχείων ή μόνο για δημιουργία δυναμικών sites ή μόνο για δημιουργία e-shop κτλ.

Κάθε εταιρεία και φορέας θέτει τους δικούς τους περιορισμούς και ανάγκες για την επιλογή μιας πλατφόρμας CMS. Οι περιορισμοί αυτοί εξαρτώνται από πολλούς παράγοντες όπως το μέγεθος της επιχείρησης, το είδος των λειτουργιών της και τον επιχειρηματικό χώρο στον οποίο δραστηριοποιείται, τα είδη του περιεχομένου που κυρίως διαχειρίζεται καθώς και το προφίλ του κοινού που ανήκει στο πελατολόγιό της. Έτσι στην πράξη είναι αρκετά δύσκολο να βρεθεί ένα προϊόν που να συνδυάζει όλα τα επιθυμητά διαθέσιμα χαρακτηριστικά προσφερόμενων λειτουργιών, ώστε να ικανοποιεί όλες τις ανάγκες.

Για τους λόγους αυτούς και λαμβάνοντας υπ' όψη ότι η αγορά των CMS είναι μια ταχέως αναπτυσσόμενη και αχανής σχετικά αγορά, καθίσταται σημαντική η ανάγκη για όσο το δυνατόν κατηγοριοποίηση και ιεράρχηση των λειτουργικών χαρακτηριστικών των συστημάτων διαχείρισης περιεχομένου. Μια τέτοια ιεράρχηση βασίζεται τόσο στην θεωρητική θεμελίωση των εννοιών content και content management, όσο και στην πρακτική εξέταση των κυριότερων πακέτων της αγοράς και στην σύγκριση των μεταξύ τους ομοιοτήτων και διαφορών. Με βάση την ιεράρχηση αυτή θα μπορούμε να είμαστε σε θέση, είτε σε περίπτωση που αγοράζουμε και αξιολογούμε ένα σύστημα είτε σε περίπτωση που αναπτύσσουμε εμείς ένα, να μπορούμε να αναγνωρίζουμε τα διακριτικά εκείνα χαρακτηριστικά, τα οποία μας οδηγούν σε αποδοτικό σχεδιασμό και στις σωστές αποφάσεις ή επενδύσεις.

Με δεδομένη την πληθώρα πακέτων CMS που κυκλοφορούν στην αγορά και λόγω της προσπάθειας κάθε κατασκευαστή να διαφοροποιηθεί από τον ανταγωνισμό, προκύπτει από μελέτες αγοράς που διεξάγουν εξειδικευμένες εταιρείες και που έχουν δημοσιευθεί στην βιβλιογραφία ότι μόνο το 30% των λειτουργικών χαρακτηριστικών που προσφέρεται σε μια εφαρμογή διαχείρισης περιεχομένου είναι κοινό ανάμεσα στις εφαρμογές που παράγονται με βάση τα διαφορετικά πακέτα. Σύμφωνα με τις μελέτες αυτές της αγοράς (META Group report) προκύπτει ότι την τρέχουσα περίοδο δεν υπάρχει



τάση για σύγκλιση των τεχνολογιών που εμφανίζει κάθε κατασκευαστής CMS σε ένα κοινό πρότυπο (μοντέλο). Το επιχειρηματικό κλίμα και οι δυνατότητες για βελτίωση / ανάπτυξη περαιτέρω των σχετικών τεχνολογιών οδηγεί τον ανταγωνισμό και προβλέπεται ότι αυτή η κατάσταση θα συνεχιστεί τα επόμενα 1-2 χρόνια. Είναι πολύ πιθανό πάντως κάποια στιγμή να υπάρξει προσπάθεια σύγκλισης των standards στην κατασκευή ενός συστήματος CMS, η οποία και θα βασίζεται σε ένα απολύτως ορισμένο κοινό πρότυπο αρχιτεκτονικής, μοντέλου δεδομένων και προσφερόμενης λειτουργικότητας των παραγόμενων εφαρμογών. Μια κοινή διαπίστωση πάντως είναι ότι ολοένα και περισσότερες εταιρείες αναζητούν web-based λύσεις για την διαχείριση του περιεχομένου τους, με στόχο την αποτελεσματική και απρόσκοπτη συνεργασία μεταξύ των εργαζομένων στην επιχείρηση (που είναι οι δημιουργοί, διαχειριστές και αποδέκτες του περιεχομένου) καθώς και την εξοικονόμηση χρόνου με χρήση ενός ενιαίου και εύκολου στην χρήση γραφικού περιβάλλοντος.

Για την ώρα αυτό που είναι σίγουρο είναι ότι κάθε πακέτο CMS έχει διαφορετική προσέγγιση σχετικά με τον σχεδιασμό του μοντέλου δεδομένων του και την αρχιτεκτονική των εφαρμογών. Κάθε κατασκευαστής στοχεύει πιθανώς περισσότερο σε συγκεκριμένες αγορές και διαφορετικές κατηγορίες χρηστών, με διαφορετικό όραμα για την ανάπτυξη και προώθηση του προϊόντος του. Αυτό το γεγονός έχει αντίκτυπο και στην προσφερόμενη λειτουργικότητα (λειτουργικά χαρακτηριστικά) των εφαρμογών που παράγονται από το εκάστοτε CMS πακέτο. Αναφέρουμε μερικά παραδείγματα για να γίνει πιο κατανοητή η διαφοροποίηση μεταξύ των σημερινών πραγματικών συστημάτων WCMS. Μερικά προϊόντα έχουν σχεδιαστεί ώστε να υποστηρίζουν Web sites που έχουν «βάθος» τριών επιπέδων, ενώ άλλα να υποστηρίζουν την δημιουργία sites με οποιοδήποτε βάθος. Η πρώτη περίπτωση προσφέρει απλότητα και ταχύτητα, η δεύτερη περίπτωση είναι πιο σύνθετη στην υλοποίηση, προσφέρει όμως μεγαλύτερες δυνατότητες.

Σε κάθε πακέτο CMS προσφέρονται δυνατότητες προσάρτησης των κατάλληλων metadata στο περιεχόμενο. Ωστόσο ένα σημείο διαφοράς είναι τα διαφορετικά GUIs και οι διαφορετικοί τρόποι που προσφέρονται για διαχείριση και τροποποίηση του περιεχομένου.

Το θέμα του workflow μπορεί να καλύπτεται από ένα απλό checkbox που επιβεβαιώνει ότι η σελίδα είναι ενεργή και να φθάνει μέχρι και την ύπαρξη σύνθετων GUIs όπου επιτρέπεται το drag and drop βημάτων workflow.

Είναι προφανές βέβαια ότι από τη στιγμή που ένα σύστημα καλύπτει κάποιο ελάχιστο σύνολο σχεδιαστικών απαιτήσεων, τότε είναι στην κρίση των ομάδων σχεδίασης και υλοποίησης να κάνουν αναγκαίες παρεμβάσεις, οδηγούμενοι από άλλους πιθανούς περιορισμούς. Αυτό θα έχει σαν αποτέλεσμα όλα τα προϊόντα CMS να πληρούν το minimum των προδιαγραφών, κάθε ένα όμως από αυτά να διαθέτει πιο εξελιγμένους μηχανισμούς σε κάποιον / κάποιους από τους λειτουργικούς τομείς της διαχείρισης περιεχομένου. Σε κάθε περίπτωση πάντως η λειτουργικότητα του συνολικού συστήματος διαχείρισης περιεχομένου είναι δυνατή χάρη στην επίτευξη «ισορροπίας» κατά την αλληλεπίδραση των επιμέρους οντοτήτων (entities) του συστήματος.

Η μορφή αλληλεπίδρασης μεταξύ των entities εξ' άλλου (και η οποία καθορίζεται σε μεγάλο βαθμό από τις λειτουργίες που αναφέρουμε παρακάτω) καθώς και η σημασία που δίνεται στον βαθμό πολυπλοκότητας των αντικειμένων της κάθε οντότητας είναι παράγοντες καθοριστικοί για την τελική μορφή που θα λάβει το CMS.

Για την ιεράρχηση των λειτουργικών χαρακτηριστικών εξετάστηκαν διάφορα μονοδιάστατα και πολυδιάστατα σχήματα ομαδοποίησης και αξιολόγησης των δυνατοτήτων ενός CMS. [BYR03], [CMW04], [STE04], [OMN04], [WEB04], [UWA04], [MSD04], [NAK02].

Ένα διδιάστατο σχήμα που επιλέχθηκε στην παρούσα εργασία φαίνεται στον παρακάτω πίνακα. Με βάση τον πίνακα αυτό ακολουθεί μια παράθεση όλων των λειτουργικών χαρακτηριστικών με κατάλληλη ομαδοποίηση. Δεξιά από κάθε λειτουργικό χαρακτηριστικό βρίσκεται εντός παρένθεσης η σήμανση Y / E / Π, που αντιστοιχεί στα υποχρεωτικά, τα επιθυμητά και τα προχωρημένα χαρακτηριστικά ενός συστήματος αντίστοιχα.

**Πίνακας 1: Λειτουργικά χαρακτηριστικά CMS - ομαδοποίηση**

	Υποσύστημα		
	Συλλογή	Διαχείριση	Δημοσίευση
Οντότητα			
Χρήστες και ομάδες χρηστών			
Ρόλοι οριζόμενοι για χρήστες / ομάδες χρηστών			
Στοιχεία περιεχομένου (πόροι)			
Δομές οργάνωσης περιεχομένου (κόμβοι)			
Δομές παρουσίασης περιεχομένου (π.χ. σελίδες , φόρμες)			
Δομές εισαγωγής περιεχομένου (π.χ. φόρμες)			
Ροές διαχείρισης (workflows)			
Αντικείμενα προβολής (π.χ. banners)			
Οντότητες για κοινότητες (community entities)			

### **Χρήστες και ομάδες χρηστών**

#### *Συλλογή*

- Δυνατότητα ασφαλούς εισόδου και εξόδου (login-logout) στο / από το σύστημα και αναγνώρισης από αυτό όλων των χρηστών με βάση ένα username και password (Y)
- Ορισμός νέου χρήστη (είτε με διαδικασία εγγραφής που εκτελεί ο ίδιος ο χρήστης είτε με ενέργεια του διαχειριστή χρηστών) με αρχικοποίηση ιδιοτήτων όπως ένα μοναδικό user ID, name, type (τύπος ομάδας στην οποία ανήκει) (Y)
- Εύρεση / ανανέωση στοιχείων χρηστών από ένα LDAP directory (Π)

#### *Διαχείριση*

- Παρουσίαση των μετα-δεδομένων (π.χ. στοιχεία επικοινωνίας) ενός χρήστη (Y)
- Δυνατότητα απομακρυσμένης εισόδου στο σύστημα ενός χρήστη με δικαιώματα διαχείρισης περιεχομένου ή χρηστών, ώστε να προβεί σε αναγκαίες αλλαγές. Η απομακρυσμένη πρόσβαση να μπορεί να γίνεται βεβαίως με ασφαλή τρόπο , χρησιμοποιώντας κατάλληλα πρωτόκολλα κρυπτογράφησης και αναγνώρισης της ταυτότητας του χρήστη (Π)
- Ενεργοποίηση και απενεργοποίηση υπάρχοντος χρήστη (Y)
- Τοποθέτηση χρήστη σε μία ή περισσότερες ομάδες χρηστών (Y)
- Εύρεση και παρουσίαση των χρηστών που έχουν δικαιώματα πρόσβασης σε έναν κόμβο της ιεραρχίας ή σε ένα στοιχείο περιεχομένου της εφαρμογής (Y)
- Ορισμός νέας υβριδικής ομάδας χρηστών με βάση μια από τις υπάρχουσες δυνατές κατηγορίες (E)
- Δυνατότητα σύνδεσης χρηστών σαν guests με περιορισμένα ένα permissions (Y)
- Υποστήριξη για το σύστημα NIS (Network Information System) του Unix. Το NIS παλιότερα ονομαζόταν yp (Yellow Pages) και η βασική του λειτουργία είναι η κοινή χρήση, από πολλά μηχανήματα , πληροφοριών ρυθμίσεων (configuration info). Στα περισσότερα συστήματα το NIS αντικαταστάθηκε από το πρωτόκολλο LDAP. (Π)

- Δυνατότητα ενσωμάτωσης με το ευρετήριο δικτύου των Windows (Active Directory) για έλεγχο και ταυτοποίηση χρηστών (Π)
- Υποστήριξη της προδιαγραφής PAM (Pluggable Authentication Modules) του Linux. Το PAM είναι μια τεχνολογία που επιτρέπει την συγγραφή προγραμμάτων τα οποία μπορούν να χρησιμοποιούν προχωρημένες μεθόδους ταυτοποίησης χρήστη (user authentication) αλλά είναι ανεξάρτητα από το σχήμα ταυτοποίησης που χρησιμοποιείται κάθε φορά, όπως για παράδειγμα ένα απλό σχήμα “Username-password” ή κάτι πιο εξελιγμένο όπως μια smart card. Αυτό επιτυγχάνεται με το «φόρτωμα» κάποιων συστατικών κατά το χρόνο εκτέλεσης με την μέριμνα του διαχειριστή συστήματος. (Π)

#### *Δημοσίευση*

- Αυτόματη ειδοποίηση των χρηστών που έχουν δικαίωμα πρόσβασης σε ένα στοιχείο περιεχομένου, κατά τη δημοσίευση ή την τροποποίησή του (E)

#### ***Ρόλοι οριζόμενοι για χρήστες / ομάδες χρηστών***

##### *Διαχείριση*

- Απόδοση ή αφαίρεση ρόλων σε μία ομάδα χρηστών, από τους ρόλους που έχουν οριστεί στη συγκεκριμένη εφαρμογή (Y)
- Ορισμός νέων υβριδικών ρόλων με βάση τους υπάρχοντες (E)
- Διαχείριση της πρόσβασης στο περιεχόμενο με βάση το σχήμα Owner/Group/Others του Unix (Π)

#### ***Στοιχεία περιεχομένου (πόροι)***

##### *Συλλογή*

- Δημιουργία agents και crawlers για την ανίχνευση σχετικού περιεχομένου στο Web , το οποίο κάνουν αυτόματα download και το εισάγουν στο repository (E)
- Δυνατότητα ανάγνωσης από μια εξωτερική σχεσιακή ή μη ΒΔ και αυτόματης εισαγωγής των records ως ξεχωριστών components (E)
- Δυνατότητα σύνδεσης με ένα OPML directory για αυτόματη ανάγνωση και εισαγωγή XML Syndications (Π)
- Φιλτράρισμα του εισερχόμενου περιεχομένου και της λειτουργικότητας και εισαγωγή τους στο repository μόνο αν πληρούν συγκεκριμένα κριτήρια δομής και μορφοποίησης (E)
- Δυνατότητα εισαγωγής περιεχομένου από οπτικά μέσα αποθήκευσης, στα οποία περιλαμβάνεται περιεχόμενο με την μορφή έτοιμων binary αρχείων ή αρχείων ΒΔ (αναγνώριση των γνωστών format .mdb .odbc .jdbc .csv .txt ) (E)
- Διάβασμα SQL query από αρχείο και εκτέλεσή του μέσα στο περιεχόμενο του repository. Τα αποτελέσματα παρουσιάζονται σε κατάλληλο πίνακα (E)
- Παροχή ενός περιβάλλοντος εξειδικευμένου editor (είτε word editor είτε spreadsheet) με την δυνατότητα εισαγωγής style sheets ή εφαρμογής κάποιου template (όπως σε ένα .dot αρχείο στο Word) με ενεργά πεδία , μακροεντολές και συνδέσμους σε εξωτερικές λίστες και αρχεία με metadata (Π)
- Δυνατότητα ψηφιοποίησης εγγράφων και άλλων hardcopy εκδόσεων πόρων περιεχομένου , με διασύνδεση interfaces σχετικών συσκευών (scanners, video cards, microphones) (Π)
- Δυνατότητα να υπάρχει διαθέσιμο ένα καλά τεκμηριωμένο API με μεθόδους για εισαγωγή και αποθήκευση εξωτερικών δεδομένων και ενσωμάτωση (integration) του CMS με εξωτερικές εφαρμογές (Π)

### *Διαχείριση*

- Δημιουργία αναφορών που αφορούν τη δημοσίευση πληροφοριακού υλικού (E)
- Δημιουργία συνδέσμου προς δικτυακούς τόπους. Ο συγγραφέας ενός συνδέσμου πρέπει να μπορεί να καθορίσει τουλάχιστον ένα τίτλο, μία περιγραφή και το σχετικό σύνδεσμο (URL) (Y)
- Δημιουργία αριθμού έκδοσης για κάθε πόρο περιεχομένου (Asset versioning) (E)
- Δυνατότητα για λήψη αντιγράφων ασφαλείας (backup) μέρους ή του συνόλου του αποθηκευμένου περιεχομένου (E)
- Σύστημα scheduling για περιοδική αυτόματη εκτέλεση κάποιων εσωτερικών triggers (π.χ. ανανέωση ή διαγραφή περιεχομένου με βάση συγκεκριμένα κριτήρια) (Π)
- Ύπαρξη συστήματος caching για ταχύτερη φόρτωση του περιεχομένου των σελίδων σε ένα web site, επιταχύνοντας την διαδικασία αναζήτησης στο repository (Π)
- Χρήση έτοιμων templates για δημιουργία νέων πόρων, μέσα από web browser based περιβάλλον (Y)
- Ύπαρξη WYSIWYG editor παράλληλα με code editor (E)
- Εύκολη διαχείριση πολυμεσικού περιεχομένου (εικόνες, video, αρχεία ήχου) με χρήση media gallery για γρήγορη ομαδοποίηση / αντίστοιχα για έγγραφα ένα document gallery (Π)
- Διαχείριση των διαγραμμένων πόρων περιεχομένου (trash handling system), με δυνατότητα αρχειοθέτησης και επαναφοράς (E)
- Υποστήριξη μακροεντολών για αυτόματη εκτέλεση μιας σειράς ενεργειών (εκτέλεση script σε κάποια γνωστή ή custom γλώσσα scripting) (Π)
- Δυνατότητα επαναφοράς σε προηγούμενη κατάσταση, ακυρώνοντας τις τελευταίες ενέργειες στο σύστημα (undo last actions) (E)
- Υποστήριξη clustering (για κατανεμημένα περιβάλλοντα) και imaging του περιεχομένου και των εφαρμογών (Π)
- Υποστήριξη του NFS (Network File System) για αποθήκευση των πόρων του περιεχομένου (E)

### *Δημοσίευση*

- Δυνατότητα αποστολής επιλεγμένου πληροφοριακού υλικού της εφαρμογής, με λειτουργία της ίδιας της εφαρμογής (π.χ. αποστολή ενός άρθρου) (Y)
- Εξαγωγή επιλεγμένου περιεχομένου από το repository σε κάποια μεταφέρσιμη μορφή, π.χ. XML αρχεία (E)
- Επικοινωνία με εξωτερικές Web Services και άλλες εφαρμογές μέσω XML-RPC , SOAP, HTTP (Π)
- Δημοσίευση περιεχομένου σε εξωτερικό web site απευθείας μέσω WebDAV (Π)
- Αυτόματη δημιουργία αρχείου RSS, Atom ή κάποιου άλλου syndication format για διάθεση του περιεχομένου ή μέρους αυτού ως feeds (E)
- Εξαγωγή σε data warehouse για data mining περιεχομένου (Π)
- Δυνατότητα προβολής των πόρων μόνο μέσα από web based ασφαλείς συνδέσεις HTTPS (E)
- Πλήρης υποστήριξη Unicode στα μενού και στις φόρμες (E)
- Internationalization του interface του portal και των βοηθητικών εφαρμογών (E)
- Δυνατότητα σύνδεσης με εξωτερική φορητή συσκευή και αποστολή περιεχομένου σε XML format. Χρήση ασύρματων πρωτοκόλλων μεταφοράς όπως το SMS (Short Messaging System) και το WAP (Wireless Application Protocol) με την WML (Wireless Markup Language) (Π)

### *Δομές οργάνωσης περιεχομένου (κόμβοι)*

#### *Διαχείριση*

- Αναζήτηση κόμβων περιεχομένου με βάση το όνομα ή metadata (Y)

#### *Δημοσίευση*

- Προβολή της πλήρους διαδρομής στην λογική ιεραρχία των κόμβων που βρισκόμαστε ανά πάσα στιγμή (Y)
- Οπτικοποίηση μιας αντιστοίχισης μεταξύ κόμβων πλοήγησης και πόρων που περιλαμβάνονται στον συγκεκριμένο κόμβο (E)

### ***Δομές παρουσίασης περιεχομένου (π.χ. σελίδες , φόρμες)***

#### *Διαχείριση*

- Δημιουργία και ενεργοποίηση / απενεργοποίηση νέας φόρμας αποστολής στοιχείων από τους χρήστες (Y)

#### *Δημοσίευση*

- Σύστημα online help με αναδύμενες φόρμες και μενού σχετικά με την λειτουργία κάθε σελίδας (Y)
- Σύστημα Frequently Asked Questions (FAQ) (E)

### ***Δομές εισαγωγής περιεχομένου (π.χ. φόρμες)***

#### *Συλλογή*

- Δημιουργία Web Forms (φορμών εισαγωγής περιεχομένου) με βάση ένα DTD ή το σχήμα μιας σχεσιακής βάσης δεδομένων (Π)
- Ενσωματωμένος news aggregator (Π)
- Φόρμα αναζήτησης στο web και εισαγωγής των αποτελεσμάτων από τις δημοφιλέστερες μηχανές (Π)
- Φόρμα εισαγωγής ιδιοτήτων (properties) για τους πόρους περιεχομένου, με στόχο την παραμετροποίηση της εμφάνισης του portal (Y)
- Δυνατότητα εισαγωγής ερώτησης σχετικά με την λειτουργία του συστήματος και το πώς διεκπεραιώνεται μια εργασία σε φυσική γλώσσα ή με λέξεις κλειδιά (Π)
- Δυνατότητα αλλαγής της εμφάνισης του περιβάλλοντος της εφαρμογής (αλλαγή theme) με κατάλληλη επιλογή του χρήστη μέσω φόρμας (Π)
- Πρόληψη για άτομα με ειδικές ανάγκες (Accessibility features) / Χρήση εναλλακτικών μορφών και μεθόδων εισαγωγής περιεχομένου (Π)

### ***Ροές διαχείρισης (workflows)***

#### *Συλλογή*

- Σύστημα έγκρισης από τους διαχειριστές (approval) για εισαγωγή νέου περιεχομένου στο repository

#### *Διαχείριση*

- Καθορισμός εάν ένα στοιχείο περιεχομένου (σύνδεσμος, κείμενο, έγγραφο κοκ) είναι δημοσιεύσιμο ή βρίσκεται ακόμη σε κατάσταση συγγραφής (Y)
- Αυτόματη καταγραφή του συντάκτη, της ημερομηνίας δημιουργίας και της ημερομηνίας δημοσίευσης ενός στοιχείου περιεχομένου (Y)
- Αυτόματη καταγραφή του ιστορικού κάθε στοιχείου περιεχομένου, σχετικά με τις ενέργειες τροποποίησης που γίνονται σε αυτό από εξουσιοδοτημένους χρήστες της εφαρμογής (E)
- Δυνατότητα όλων των χρηστών authors (δημιουργοί περιεχομένου) για λειτουργία remove edit permissions. Δηλαδή κάθε συγγραφέας μπορεί να απενεργοποιεί προσωρινά την δυνατότητα που έχουν άλλοι χρήστες να επεξεργάζονται ένα πόρο , όταν αυτός δουλεύει σε αυτόν τον πόρο, δηλαδή κλείδωμα του πόρου για consistency (Y)
- Δυνατότητα αποστολής εσωτερικά μηνυμάτων μεταξύ των εγγεγραμμένων χρηστών (Π)

### *Δημοσίευση*

- Αποστολή email στους διαχειριστές και στους χρήστες με σχετική πρόσβαση όταν ένας πόρος περιεχομένου έχει αλλάξει (E)
- Δημιουργία προσωπικής λίστας επικείμενων εργασιών επεξεργασίας πόρων περιεχομένου από έναν χρήστη (to do list) και δημοσίευσή της σε όλους τους χρήστες που έχουν δυνατότητα επεξεργασίας των ίδιων πόρων (E)

### **Αντικείμενα προβολής (π.χ. banners)**

#### *Συλλογή*

- Εισαγωγή νέων εξωτερικών banners και δυνατότητα σχεδιασμού νέων (Y)

#### *Διαχείριση*

- Στα banners υπάρχουν συνήθως τουλάχιστον 2 taxonomies : το εταιρικό-εμπορικό και το θεματικό. Να προσφέρεται δυνατότητα να ορίζονται και νέα taxonomies (E)

#### *Δημοσίευση*

- Δυνατότητα αυτόματης εναλλαγής των banners με βάση κάποιο συμβόλαιο χρονοχρέωσης. Εφαρμογή μεθόδων marketing και διαφήμισης μιας εταιρείας (Π)

### **Οντότητες για κοινότητες (communities' entities)**

#### *Συλλογή*

- Φόρμα εισαγωγής email για εγγραφή σε newsletter της κοινότητας (E)
- Συστήματα ηλεκτρονικής ψηφοφορίας (voting systems) (E)
- Συστήματα δημοσκοπήσεων / σφυγμομετρήσεων (polls)
- Δυνατότητα παραμετροποίησης της εμφάνισης των σελίδων στο portal (ποια link και ποιες επιλογές θα είναι διαθέσιμες) (E)
- Δυνατότητα σήμανσης / σύνδεσης από τον χρήστη μιας λέξης ή ενός τμήματος κειμένου με ένα URL (Hotwords) (E)
- Φόρμες που συνδέονται με το τοπικό file system για άνοιγμα / upload τοπικών εγγράφων (E)
- Εισαγωγή δεδομένων τοποθεσίας στην οποία βρίσκεται ο χρήστης ώστε να επιλέγεται η βέλτιστη τοποθεσία όταν γίνεται download ένα αρχείο ψηφιακού περιεχομένου (E)
- Δυνατότητα download manager και παροχής online δίσκου για αποθήκευση τοπικών και downloaded εγγράφων (Π)
- Δυνατότητα για voice browsing (φωνητικές εντολές). Γενικότερα μέριμνα για accessibility features (Π)

#### *Διαχείριση*

- Δημιουργία νέου audience που διαθέτει ένα μοναδικό audience ID, name, έναν τουλάχιστο στόχο (goal) που πρέπει να ακολουθεί καθώς και πιο σύνθετες δομές για την αποθήκευση στοιχείων που σχετίζονται με το προφίλ του συγκεκριμένου audience, δηλαδή στατιστικά στοιχεία σχετικά με τις τάσεις, δημογραφία, επιθυμίες, νοοτροπία των μελών του audience αυτού. Συνεπώς όταν ορίζεται μέσα στο CMS ένας χρήστης που ανήκει στο συγκεκριμένο audience , αυτόματα κληρονομεί όλες τις ιδιότητες του audience αυτού. (Y)
- Ενεργοποίηση / απενεργοποίηση audience ή διαγραφή του από το σύστημα (Y)
- Κάθε audience ενδέχεται να κάνει συγκρίσεις μιας δημοσίευσης (publishing) του συστήματός μας με κάποια άλλη δημοσίευση που γνωρίζει και του αρέσει. Μια χρήσιμη λειτουργία λοιπόν είναι η καταγραφή των συγκρίσεων αυτών για κάθε audience, ώστε να γνωρίζουμε ανά πάσα

στιγμή τον ανταγωνισμό και πώς να κινηθούμε ώστε να βελτιώσουμε τις δημοσιεύσεις που απευθύνονται σε ένα συγκεκριμένο audience. (E)

- Σύνδεση ενός audience με κάποιον / κάποιους δημιουργό περιεχομένου, μετά από σύγκριση του profile του audience με το profile του δημιουργού. (E)
- Καταγραφή των τάσεων και προτιμήσεων μιας ομάδας ή ενός μέλους μιας κοινότητας, με στόχο την αντιστοίχιση τον χρήστη/ομάδα μιας ή περισσότερων κατηγοριών περιεχομένου (E)
- Δυνατότητα προσθήκης plug-in και ρύθμισης εξωτερικών προγραμμάτων και εφαρμογών ώστε να συνεργάζονται με τις σελίδες και το περιεχόμενο της online κοινότητας (II)
- Δυνατότητα χρήσης cookies για προηγμένες υπηρεσίες στους χρήστες και προσωποποίηση του portal (E)

### *Δημοσίευση*

- Δυνατότητα αποστολής e-mail σε ένα χρήστη ή σε ένα σύνολο χρηστών (π.χ. σε όλους τους χρήστες που έχουν πρόσβαση σε ένα στοιχείο περιεχομένου) ανεξάρτητα από το λογισμικό e-mail που έχει εγκατεστημένο ο χρήστης στον προσωπικό του υπολογιστή (E)
- Πίνακας ανακοινώσεων με ελεγχόμενη πρόσβαση (για συγγραφή ή ανάγνωση) (Y)
- Δημόσιες συζητήσεις με ελεγχόμενη πρόσβαση (για συγγραφή ή ανάγνωση) (E)
- Ημερολόγιο γεγονότων (events calendar) (E)
- Προβολή σελίδας με στατιστικά στοιχεία σχετικά με την κίνηση και το προφίλ του χρήστη (Y)
- Φόρμα για πρόσβαση μέσω web στους web-based POP3 email λογαριασμούς (II)
- Δυνατότητα δημιουργίας προσωπικού account για web mail από τους χρήστες της κοινότητας (E)
- Παρουσίαση , σε κάθε σελίδα, μιας λίστας με μερικούς προτεινόμενους σχετικούς με το περιεχόμενο της σελίδας εξωτερικούς δεσμούς Web (URL). Η λειτουργία αυτή μπορεί να βασίζεται και στην συλλογή προσωπικών στοιχείων προφίλ του χρήστη (II)
- Δημιουργία από τον χρήστη κάποιων custom menus που θα εμφανίζονται σε κάθε σελίδα (αγαπημένα links ή σύνδεσμοι με περιεχόμενο) (E)
- Εμφάνιση νέων (news) σε μια περιοχή του παραθύρου κάθε σελίδας ή μόνο στην homepage (E)
- Δυνατότητα για real time chatting (με κάποια Java-based εφαρμογή) με άλλους χρήστες της κοινότητα που βρίσκονται online εκείνη τη στιγμή (II)
- Δυνατότητα δημιουργίας και συντήρησης προσωπικού Web log (blog) για κάθε χρήστη και δημοσίευση σε αυτό μέσω syndication formats (II)
- Δυνατότητα δημιουργίας προσωπικής ιστοσελίδας και συντήρησής της από τον ίδιο το χρήστη με web-based WYSIWYG editors (II)
- Δυνατότητα σε χρήστες να αφήνουν σχόλια σε σελίδες άλλων χρηστών / χρήση guestbook (II)
- Εμφάνιση ημερολογίου με καθημερινό to do list, διαχείριση συναντήσεων και εορτολόγιο (E)
- Δυνατότητα ηλεκτρονικής ατζέντας και διαχείρισης προσωπικών επαφών (E)
- Αποστολή feedback στον διαχειριστή του portal (Y)

## **2.6 Πρακτικές εφαρμογές των CMS**

Τα CMS βρίσκουν στις μέρες μας ολοένα και μεγαλύτερη αποδοχή σε πολλούς τομείς δραστηριοτήτων. Ακριβώς όπως ένα λειτουργικό σύστημα παρέχει την πλατφόρμα πάνω στην οποία στηρίζονται οι εφαρμογές για να εκτελεστούν, έτσι και ένα CMS είναι η βάση που χρειάζονται όλα τα συστατικά μιας εφαρμογής (δικτυακής ή τοπικής) για να επιτελέσουν χρήσιμες λειτουργίες, επικοινωνώντας με το back-end σύστημα. Μέσα από τις εφαρμογές επιδιώκεται η προβολή της κατάλληλης πληροφορίας στο κατάλληλο κοινό και την κατάλληλη στιγμή μέσω του κατάλληλου καναλιού. Τα παραπάνω μπορούν να επιτευχθούν ταυτόχρονα μόνο αν υπάρχει ένα ολοκληρωμένο back-end σύστημα διαχείρισης περιεχομένου. Ακολούθως αναφέρονται μερικές από τις σημαντικότερες εφαρμογές των CMS:

### **2.6.1 Δυναμικά Web Sites με ισχυρές δυνατότητες προσωποποίησης (personalization)**

Επιτυγχάνουν εξατομικευμένη διάθεση και προβολή περιεχομένου, η οποία γίνεται με την τροποποίηση κάθε φορά της μορφοποίησης και της δομής του περιεχομένου ανάλογα με τις προτιμήσεις του τελικού χρήστη του συστήματος. Η προσωποποίηση επιδιώκεται κυρίως σε μεγάλα web sites και portals, και αυτό με στόχο να επιτευχθεί μια όσο το δυνατόν ενδιαφέρουσα και χρήσιμη εμπειρία για τον επισκέπτη και χρήστη του site. Η προσωποποίηση πετυχαίνεται με την συλλογή προσωπικών προτιμήσεων και δεδομένων με την συγκατάθεση του χρήστη. Εν συνεχεία τα δεδομένα αυτά συνδέονται με κατάλληλο τρόπο με το περιεχόμενο που κρίνεται ότι ενδιαφέρει περισσότερο τον συγκεκριμένο χρήστη. Όλα τα στοιχεία της μορφοποίησης και του προβαλλόμενου καθ'αυτού περιεχομένου διαμορφώνονται την στιγμή εκείνη με βάση το τι έχει δηλωθεί.

Μπορούμε να διακρίνουμε τρία βασικά είδη προσωποποίησης. Το πρώτο είδος είναι προσωποποίηση με βάση εξατομικευμένα ή ομαδικά προφίλ, δηλαδή ένα σύνολο από χαρακτηριστικά που διαθέτει ο εκάστοτε χρήστης. Με βάση αυτά γίνεται η αντιστοίχιση περιεχομένου – προφίλ. Το δεύτερο είδος είναι αυτό που εξάγει τις γενικότερες προτιμήσεις του χρήστη με βάση την προηγούμενη συμπεριφορά του στο περιβάλλον της εφαρμογής. Κρίνοντας δηλαδή από τα αντικείμενα περιεχομένου που έχει ήδη ζητήσει, το σύστημα συνδέει τις προτιμήσεις αυτές με τα τμήματα περιεχομένου αντίστοιχου ενδιαφέροντος. Το τρίτο τέλος είδος είναι αυτό που βασίζεται στα templates μιας συγκεκριμένης διαφημιστικής εκστρατείας ή ενός επιχειρηματικού πλάνου γενικότερα. Όπως είναι προφανές, τόσο τα στοιχεία της μορφοποίησης όσο και της δομής είναι δεδομένα και υπαγορεύονται από την εκάστοτε πολιτική. Απευθύνονται εξ'άλλου σε δεδομένο κοινό.

Σε καθεμιά από τις τρεις αυτές περιπτώσεις ο κοινός παρανομαστής είναι το σύστημα διαχείρισης περιεχομένου, το οποίο συλλέγει ή εξάγει προσωπικές προτιμήσεις, πετυχαίνει στην συνέχεια το κατάλληλο “ταίριασμα” των προτιμήσεων αυτών με το αντίστοιχο περιεχόμενο και εν συνεχεία διαθέτει το περιεχόμενο αυτό στο κοινό με την κατάλληλη μορφοποίηση. Κάποια ωστόσο στοιχεία της παρουσίασης θα πρέπει να είναι σταθερά και ανεξάρτητα του προφίλ του χρήστη. Οι περιοχές των σταθερών και υπό διαμόρφωση τμημάτων μιας έκδοσης περιεχομένου μπορεί να καθοριστεί σε ένα πρότυπο έκδοσης (publishing template)

### **2.6.2 Επιχειρηματικές εφαρμογές των CMS**

Αρκετά προϊόντα CMS έχουν ως κύριο στόχο την εξυπηρέτηση των αναγκών των μεγάλων web sites και των intranets - extranets. Για τα sites αυτά είναι αναγκαία η ύπαρξη ενός συστήματος που θα δρα συντονιστικά στον πυρήνα όλων των εφαρμογών λογισμικού με τέτοιο τρόπο ώστε οι διαδικασίες να αυτοματοποιούνται. Ένα CMS βοηθά στον συντονισμό των δραστηριοτήτων όλων των εμπλεκόμενων στην διαδικασία της δημιουργίας και συντήρησης ενός μεγάλου εταιρικού web site, όπως είναι για παράδειγμα οι συγγραφείς περιεχομένου, οι προγραμματιστές, οι διαχειριστές περιεχομένου, οι webmasters. Επίσης ένα CMS προσφέρει τη δυνατότητα το περιεχόμενο να απεμπλέκεται από την μορφοποίησή του και την εμφάνισή του στις διαφορετικές κατηγορίες



χρηστών. Βοηθά τέλος στην κατάλληλη ιεράρχηση του περιεχομένου που προέρχεται από πολλές (ανομοιογενείς πολλές φορές) πηγές, με στόχο την ενιαία διαχείρισή του.

Αναφέρουμε ένα παράδειγμα intranet που κάνει εσωτερικά εκτεταμένη χρήση συστήματος διαχείρισης περιεχομένου. Πρόκειται για το intranet Info 2004, το οποίο σχεδιάστηκε και υλοποιήθηκε από την εταιρεία Atos Origin για τις ανάγκες των Ολυμπιακών Αγώνων της Αθήνας. Το συγκεκριμένο σύστημα εκτελούνταν σε συνολικά 2000 τερματικά – πελάτες που βρίσκονταν διασκορπισμένα σε όλους τους χώρους πρόσβασης των δημοσιογράφων και των μελών της Ολυμπιακής οικογένειας. Στόχος ήταν η γρήγορη, απρόσκοπτη και ολοκληρωμένη πληροφόρηση των παραπάνω ατόμων και η παροχή περιεχομένου σχετικού με τα Ολυμπιακά αγωνίσματα.

Πιο συγκεκριμένα το σύστημα παρείχε περιεχόμενο για τα αποτελέσματα των αγώνων, αθλητικά νέα, βιογραφικά αθλητών, στατιστικά μεταλλίων, παγκόσμια και τοπικά ρεκόρ, προγράμματα αγώνων, πληροφορίες για τον καιρό και τις μεταφορές στις ολυμπιακές πόλεις. Τέλος προσφερόταν και η δυνατότητα επικοινωνίας μεταξύ των χρηστών του συστήματος με χρήση εσωτερικού email. Η παρουσίαση του περιεχομένου των Αγώνων γινόταν με διαίρεση της οθόνης σε 4 βασικές υποπεριοχές:

Το πλαίσιο πλοήγησης για την επιλογή της θεματικής κατηγορίας περιεχομένου

Την επικεφαλίδα για την απεικόνιση των εκάστοτε τίτλων και την παροχή υπερσυνδέσμων προς συναφές περιεχόμενο

Το κυρίως μέρος της οθόνης για την απεικόνιση των πόρων του περιεχομένου

Το κυλιόμενο τμήμα στο κάτω μέρος της οθόνης για την εμφάνιση νέων ειδήσεων, μηνυμάτων γενικού ενδιαφέροντος και ειδικών μηνυμάτων σχετιζόμενων με το εκάστοτε παρουσιαζόμενο περιεχόμενο

Για την υλοποίηση του έργου αυτού εργάστηκαν τόσο το τεχνικό προσωπικό της Atos Origin όσο και μια πλειάδα δημοσιογραφικού και τεχνικού προσωπικού από την «Αθήνα 2004» σε θέσεις που σχετίζονταν με την συλλογή, επεξεργασία, μετάφραση και δημοσίευση του περιεχομένου. Πέραν δηλαδή των αυτοματοποιημένων διαδικασιών που περιλαμβάνει ένα σύστημα διαχείρισης περιεχομένου, πάντα είναι απαραίτητος ο ανθρώπινος παράγοντας, ιδιαίτερα στο στάδιο της συλλογής του περιεχομένου και της εισαγωγής του στο σύστημα.

Επιπρόσθετα παραδείγματα και case studies ανάπτυξης επιχειρήσεων και οργανισμών, τα οποία μπορεί κανείς να βρει άφθονα στο Διαδίκτυο, καταδεικνύουν την μεγάλη πρακτική σημασία των CMS στον κόσμο των μεγάλων επιχειρήσεων και οργανισμών. Πολλοί φορείς ξεκίνησαν να χρησιμοποιούν μια απλή λύση Web Content Management για την καλύτερη οργάνωση των εταιρικών πληροφοριών και κατέληξαν να αναπτύσσουν πολυάριθμα intranet, extranet και δημόσια εμπορικά web sites, όπως επίσης και συστήματα διαχείρισης εταιρικού περιεχομένου (Enterprise Content Management - ECM). Όλες οι επιχειρηματικές λειτουργίες μπορούν να πραγματοποιούνται με την βοήθεια ενός ολοκληρωμένου συστήματος διαχείρισης περιεχομένου. Αφού εγκατασταθεί και ρυθμιστεί για να λειτουργεί σύμφωνα με τις ανάγκες της επιχείρησης, απαιτεί από λίγο έως καθόλου την επίβλεψη από τεχνικό προσωπικό. Ένας αποδοτικός σχεδιασμός του business workflow, που λαμβάνει υπ' όψη και τα πιο απαισιόδοξα σενάρια, όταν ενσωματωθεί στο σύστημα διαχείρισης περιεχομένου μπορεί να έχει εκπληκτικά αποτελέσματα.

Τα πλεονεκτήματα για τις λειτουργίες της επιχείρησης είναι πάρα πολλά συγκριτικά με το κόστος της επένδυσης για την εγκατάσταση ενός CMS. Επιτυγχάνεται πιο αποδοτική συνεργασία μεταξύ των τμημάτων της επιχείρησης και χωρίς την παρέμβαση εξειδικευμένου προσωπικού όλοι οι χρήστες και οι υπάλληλοι μπορούν να αναλαμβάνουν ενεργούς ρόλους στην αλυσίδα των διαδικασιών δημιουργίας – διαχείρισης – δημοσίευσης περιεχομένου.

### **2.6.3 Διαδικτυακές κοινότητες (Online communities)**

Είναι οργανωμένα sites που στόχο έχουν να δημιουργήσουν μια ηλεκτρονική κοινότητα, έναν τόπο συνάντησης για άτομα με κοινά ενδιαφέροντα και στόχους. Παρέχονται σε αυτά τα μέλη υπηρεσίες όπως συνομιλία σε πραγματικό χρόνο και μέσω forum, δυνατότητα παρουσιάσεων, αποθήκευσης και διαμοιρασμού αρχείων, αναζήτησης πληροφοριών σχετικά με τα θέματα που συζητούνται και αναζήτησης ατόμων από τη βάση των μελών που είναι εγγεγραμμένοι χρήστες της κοινότητας. Οι

υπηρεσίες αυτές παράγονται και προσφέρονται είτε από εμπορικούς φορείς με τελικό στόχο το κέρδος είτε από ιδιώτες, οι οποίοι από μεράκι επενδύουν στην δημιουργία μιας κοινότητας για το αγαπημένο τους θέμα, με στόχο να φέρουν κοντά ανθρώπους που μοιράζονται κοινές επιδιώξεις.

Στην περίπτωση των online κοινοτήτων, η συλλογή, διαχείριση και προβολή του περιεχομένου γίνεται τόσο από τους διαχειριστές περιεχομένου (ειδικά στις πρώτες φάσεις λειτουργίας) όσο και από τα ενεργά μέλη της κοινότητας. Παράλληλα με τα διάφορα μέσα με τα οποία συλλέγεται το περιεχόμενο, όπως για παράδειγμα φόρμες εισαγωγής στοιχείων υπό μορφή κειμένου ή και ολόκληρων αρχείων, δυνατότητα αποστολής email από τα μέλη, εισαγωγή δεδομένων μέσω bulletin boards και chat rooms, υπάρχει και η πολύ σημαντική δυνατότητα για συλλογή δεδομένων σχετικών με το αντικείμενο της online κοινότητας με χρήση του πρωτοκόλλου RSS, κάνοντας syndication (μια από τις βασικές λειτουργικές περιοχές ενός CMS όπως αναφέραμε παραπάνω) από εμπορικά sites που είναι άροχοι μεγάλων όγκων περιεχομένου.

#### **2.6.4 Ειδησεογραφικά πρακτορεία (news agencies) και Τύπος (Έντυπος και Ηλεκτρονικός)**

Οι διαφορετικές μορφές στις οποίες μπορεί να δημοσιευθεί το περιεχόμενο των φορέων αυτών περιλαμβάνουν τις ηλεκτρονικές εκδόσεις (Internet, intranets, extranets, τηλεφωνικές συσκευές WAP και συσκευές UMTS, PDAs), τις έντυπες εκδόσεις (print publications) και τα syndications.

Μερικά από τα είδη φυσικής μορφοποίησης που μπορεί να λάβει το περιεχόμενο στις ηλεκτρονικές εκδόσεις είναι οι τα αρχεία τύπου HTML, XHTML, DHTML, WinHelp, HTML Help, Java Help, PDF, DOC, XML. Με τη χρήση του CMS το περιεχόμενο οργανώνεται ιεραρχικά και αποδεσμεύεται η δομή του από την μορφοποίησή του, ώστε αυτό να είναι ανεξάρτητο από το είδος της έκδοσης (publishing) που θα χρησιμοποιηθεί. Τέλος υπεισέρχονται από το σύστημα διαχείρισης περιεχομένου συγκεκριμένοι κανόνες για το πώς θα γίνει η αντιστοίχιση περιεχομένου με την εκάστοτε έκδοση (ηλεκτρονικής, έντυπης ή syndication) και ποια θα είναι η τελική μορφή που θα λάβει κάθε έκδοση. Οι εκδόσεις μπορούν να εκδίδονται σε παράλληλο χρόνο και αυτό χάρη στο CMS.

Το XML Syndication είναι μια μορφή μαζικών ηλεκτρονικών εκδόσεων που απευθύνονται ταυτόχρονα σε ευρύ κοινό. Οι δημοσιεύσεις, ανάλογα με το πρωτόκολλο στο οποίο στηρίζονται έχουν και διαφορετική εσωτερική υλοποίηση. Τα syndications είναι οργανωμένα σε κανάλια περιεχομένου και αυτά με την σειρά τους περιέχουν μια σειρά από άρθρα περιεχομένου, τα λεγόμενα «feeds».

Μερικοί από τους πιο γνωστούς παρόχους περιεχομένου διεθνώς (content providers) που χρησιμοποιούν το XML syndication είναι οι ακόλουθοι:

- ABC News
- AOL – Time Warner
- BBC
- Reuters
- CNN
- Forbes Magazine
- MSN
- NASA
- Oracle
- The New York Times
- Yahoo News

Η λειτουργία της δημοσίευσης περιεχομένου με τη χρήση του XML syndication είναι πλέον αρκετά διαδεδομένη. Πολλά sites που θέλουν να δημοσιεύσουν περιεχόμενο προς τρίτους χρησιμοποιούν τη λειτουργία αυτή. Για την παρακολούθηση και ανάγνωση των «feeds» απαιτείται στον τελικό χρήστη μια κατάλληλη εφαρμογή client reader. Περισσότερα για το XML Syndication στο κεφάλαιο 6 και στο παράρτημα Α της παρούσας εργασίας.

Επίσης, ένας από τους στόχους της εργασίας είναι να διερευνήσει τις δυνατότητες εφαρμογής των Web Services στο πεδίο του syndication. Στόχος είναι η διανομή του περιεχομένου να είναι ανεξάρτητη από proprietary πρωτόκολλα και να βασίζεται σε ανοικτά πρότυπα επικοινωνίας, όπως είναι για παράδειγμα οι Web Services.

## 3 Κατηγοριοποίηση ΣΔΠ

Η κατηγοριοποίηση των συστημάτων διαχείρισης διαδικτυακού περιεχομένου (Web Content Management Systems) θα γίνει με βάση δύο άξονες και πιο συγκεκριμένα:

Με βάση το εξειδικευμένο είδος του περιεχομένου που διαχειρίζονται (π.χ. διακρίνουμε τα συστήματα διαχείρισης γνώσης ή τα συστήματα διαχείρισης εγγράφων)

Με βάση το κόστος κτήσης και συντήρησης των αδειών του πακέτου λογισμικού. Εδώ έχουμε δύο μεγάλες κατηγορίες. Αφ' ενός τα open – source συστήματα και αφ' ετέρου τα εμπορικά συστήματα που με τη σειρά τους κατηγοριοποιούνται ανάλογα με το κόστος, τις δυνατότητες που προσφέρουν και την αγορά στην οποία απευθύνονται.

### **3.1 Διάκριση με βάση το εξειδικευμένο περιεχόμενο**

Παρακάτω αναφέρονται εν συντομία οι διάφορες κατηγορίες πληροφοριακών συστημάτων που σχετίζονται με την διαχείριση μιας συγκεκριμένης κατηγορίας περιεχομένου. Μερικά από τα συστήματα αυτά αποτελούν απλά ένα υποσύνολο των συστημάτων CMS, μερικά άλλα είναι απολύτως συμβατά με τις έννοιες που πρέπει να ένα CMS, οι λειτουργίες τους όμως έχουν να κάνουν με την συλλογή, αποθήκευση, επεξεργασία και διανομή πληροφορίας.

#### **3.1.1 Συστήματα Διαχείρισης Εταιρικού Περιεχομένου - Enterprise Content Management System (ECMS)**

Αποτελείται από ένα βασικό σύστημα διαχείρισης περιεχομένου ως πυρήνα του μαζί με πρόσθετες δυνατότητες διαχείρισης εταιρικής πληροφορίας και γνώσης. Μερικές εξειδικεύσεις συστημάτων ECM αναφέρουμε παρακάτω.

Με κύριο εκπρόσωπό το ηλεκτρονικό εμπόριο (e-commerce) και τις ηλεκτρονικές συναλλαγές, η επιχειρηματική δραστηριότητα και ειδικότερα οι ενδοεπιχειρησιακές διεργασίες και οι διεπιχειρησιακές σχέσεις, όπως και οι σχέσεις της επιχείρησης με άλλους παράγοντες (συνεργάτες, μεταπωλητές, προμηθευτές, πελάτες) κρίνονται σε μεγάλο βαθμό από την αυτοματοποίηση της διαχείρισης του εταιρικού περιεχομένου.

Αυτό που πρέπει να τονισθεί για τα CMS είναι ότι δεν ευθύνονται αυτά για την τεχνική υλοποίηση των ηλεκτρονικών συναλλαγών, αλλά προσφέρουν τη δυνατότητα να παρέχεται μια συγκεκριμένη λειτουργικότητα από συγκεκριμένα σημεία την κατάλληλη στιγμή και για τον κατάλληλο φορέα. Η υλοποίηση της δυνατότητας αυτή καθ' αυτή είναι δυνατή χάρη σε άλλες υποστηρικτικές τεχνολογίες (π.χ. CGI Scripts ή δυναμικές σελίδες JSP ή επικοινωνία μέσω Web Services και μηνυμάτων XML). Ας αναφέρουμε ένα παράδειγμα για να γίνουν πιο κατανοητά τα παραπάνω.

Έστω λοιπόν ότι έχουμε ένα κατάστημα που προσφέρει τη δυνατότητα ηλεκτρονικών συναλλαγών, όπως για παράδειγμα ένα ηλεκτρονικό βιβλιοπωλείο. Το site «τρέχει» σε web server της εταιρείας. Στον ίδιο server βρίσκεται αποθηκευμένη η βάση δεδομένων όλων των τιμοκαταλόγων των διαθέσιμων προϊόντων, καθώς και στοιχεία υπαρχόντων πελατών. Το site είναι μια εφαρμογή που έχει υλοποιηθεί με ενσωμάτωση χαρακτηριστικών CMS για την καλύτερη προσωποποίηση και διαχείριση του περιεχομένου. Όταν κάποιος χρήστης κάνει login στο ασφαλές τμήμα του καταστήματος, τότε αναγνωρίζεται από το σύστημα και του εμφανίζονται οι εκκρεμείς παραγγελίες του, καθώς και ενημέρωση για προϊόντα που ίσως τον ενδιαφέρουν με βάση τις προτιμήσεις του. Η δυνατότητα πληρωμής με πιστωτική κάρτα ενέχει την εισαγωγή των στοιχείων της κάρτας, την επικοινωνία με τον τραπεζικό φορέα του πελάτη, την έκδοση ηλεκτρονικής απόδειξης και την διεκπεραίωση της συναλλαγής. Αυτές και διάφορες άλλες ενέργειες δεν συντελούνται από το CMS. Το CMS είναι υπεύθυνο για την τοποθέτηση του κατάλληλου HTML κώδικα στις κατάλληλες σελίδες ώστε να υλοποιηθούν οι παραπάνω ενέργειες με τρόπο που συμφωνεί τόσο με τους κανόνες της εταιρείας όσο και με τις ιδιαιτερότητες του χρήστη.

#### **3.1.2 Customer Relationship Management Systems**

Εκτός από τα συστήματα συναλλαγών, ένα CMS είναι χρήσιμο και στην περιοχή διαχείρισης πελατών (Customer Relationship Management CRM), το οποίο συλλέγει και συνθέτει πληροφορίες σχετικές με το πελατολόγιο μιας επιχείρησης με στόχο η επιχείρηση αυτή να προσαρμόζει τις δραστηριότητές τις, τις υπηρεσίες και τα προϊόντα της με βάση τις ανάγκες και τις προτιμήσεις των εκάστοτε πελατών.

Τα συστήματα CRM εμφανίστηκαν σχετικά πρόσφατα στην αγορά και έχουν σχεδιαστεί ώστε να αποθηκεύουν πολλές διαφορετικές εταιρικές πληροφορίες και να τις παρουσιάζουν στους front-end υπαλλήλους μιας εταιρείας. Οι πληροφορίες που αποθηκεύει ένα CRM σύστημα μπορούν να υπάγονται στις εξής κατηγορίες:

- Στοιχεία πελατών
- Ιστορικό συναλλαγών της εταιρείας

- Δυνατότητες πωλήσεων
- Διεπιχειρησιακές πληροφορίες (B2B Information)
- Feedback πελατών

Ο στόχος της διαχείρισης όλων των παραπάνω πληροφοριών είναι η εξυπηρέτηση των πελατών και η αύξηση των πωλήσεων στην εταιρεία. Για την καλύτερη εξυπηρέτηση του παραπάνω στόχου προστίθενται συνήθως πληροφορίες που σχετίζονται με την διαχείριση της εταιρικής γνώσης και που αφορούν σε:

- Λεπτομέρειες ανταγωνιστικών προϊόντων και υπηρεσιών
- Υλικό για sales support
- Σημαντικές διαδικασίες, κανόνες και πολιτικές εντός της επιχείρησης
- Απαντήσεις σε συχνές ερωτήσεις πελατών

Η παραπάνω λειτουργικότητα για την διαχείριση γνώσης, ανάλογα με την πολυπλοκότητα του CRM, είτε ενσωματώνεται στο CRM είτε διεκπεραιώνεται από κάποιο εξωτερικό CMS.

Μερικά από τα συστήματα που αναφέρονται σε αυτές τις παραγράφους, όπως για παράδειγμα τα συστήματα CRM ή τα συστήματα διαχείρισης γνώσης και τα συστήματα διαχείρισης εγγράφων (βλ. παρακάτω), εντάσσονται όπως είπαμε σε ένα γενικότερο πλαίσιο συστημάτων διαχείρισης που συνήθως καλύπτουν τις επιμέρους ανάγκες τμημάτων σε μεγάλες επιχειρήσεις, ένα πλαίσιο εφαρμογών που έχει επικρατήσει να ονομάζεται Διαχείριση Εταιρικού Περιεχομένου ή Enterprise Content Management (ECM). Άλλα σχετικά συστήματα που αποτελούν ECM συστήματα είναι και τα ακόλουθα:

### **3.1.3 Ολοκληρωμένα συστήματα διαχείρισης εγγράφων (Integrated Document Management Systems)**

Αρχικά τα συστήματα αυτά ξεκίνησαν ως συστήματα που διαχειρίζονταν μόνο έγγραφα, με την έννοια ότι το περιεχόμενο ήταν ένα ολόκληρο έγγραφο και δεν υποδιαιρούνταν σε μικρότερες διαχειρίσιμες μονάδες. Αυτό ωστόσο άλλαξε με την πάροδο των χρόνων και τα συστήματα διαχείρισης εγγράφων υιοθέτησαν πολλά από τα βασικά χαρακτηριστικά των CMS, ένα από τα οποία είναι βέβαια και η διαίρεση του περιεχομένου σε «αντικείμενα περιεχομένου» για ευκολότερη και πιο αποδοτική διαχείριση.

Βοηθούν τις επιχειρήσεις να ελέγχουν την δημιουργία και ροή εγγράφων μέσα στο περιβάλλον τους με την χρήση ενός κεντρικού repository και κανόνων workflow που ενσωματώνουν τους εσωτερικούς κανόνες και την λογική λειτουργίας της επιχείρησης, με την χρήση των κατάλληλων metadata. Στοχεύουν κυρίως στην αποθήκευση (με μετατροπή όλων των ειδών εγγράφων και εισαγωγή τους στο repository) και στην μετέπειτα εξαγωγή του συνόλου των ηλεκτρονικών εγγράφων της επιχείρησης σε διαφορετικές μορφοποιήσεις (π. χ. Στην μορφοποίηση που είχαν αρχικά παραχθεί)

### **3.1.4 Συστήματα Διαχείρισης Εγγραφών - Records management systems (RMS)**

Το αυστραλιανό πρότυπο διαχείρισης εγγραφών (AS 4390) ορίζει τα RMS ως πληροφοριακά συστήματα τα οποία αποθηκεύουν, επεξεργάζονται και προσφέρουν πρόσβαση σε εγγραφές δεδομένων. Οι εγγραφές μπορεί να είναι τόσο ηλεκτρονικά όσο και τυπωμένα ή χειρόγραφα κείμενα.

### **3.1.5 Enterprise Resource Planning (ERP) – Human Resources Management systems (HRM)**

Ίσως μια από τις σημαντικότερες κατηγορίες επιχειρηματικού λογισμικού που σχεδιάζεται και υλοποιείται σήμερα για χρήση από επιχειρήσεις και οργανισμούς. Αφορά στην καταγραφή και διαχείριση όλων των υλικών και ανθρώπινων πόρων μιας επιχείρησης ή οργανισμού. Γίνεται χρήση στατιστικών μοντέλων για data mining και δημιουργία προβλέψεων, ούτως ώστε να τροφοδοτείται με χρήσιμες πληροφορίες ο επενδυτικός και στρατηγικός σχεδιασμός κάθε επιχείρησης, αλλά παράλληλα να βελτιστοποιούνται οι καθημερινές και βραχυπρόθεσμες επιλογές για την συνεχή αναδιάρθρωση εξοπλισμού και προσωπικού. Τελικός στόχος η μεγιστοποίηση του κέρδους και η βελτιστοποίηση της λειτουργίας στο εσωτερικό μιας επιχείρησης.

### **3.1.6 Συστήματα Διαχείρισης Ψηφιακών Πόρων - Digital asset management (DAM) system**

Υποστηρίζουν την αποθήκευση, εξαγωγή και επαναχρησιμοποίηση ψηφιακών αντικειμένων μέσα σε έναν οργανισμό. Η διαφορά των συστημάτων αυτών από ένα CMS είναι ότι υποστηρίζουν την επεξεργασία multimedia περιεχομένου, όπως εικόνες, video και ήχος. Μερικές φορές προσφέρουν και δυνατότητες διαχείρισης πνευματικών δικαιωμάτων.

Η τεχνολογία αυτή ασχολείται με την ψηφιοποίηση, αποθήκευση, αποδοτική διαχείριση και διανομή multimedia περιεχομένου (δηλαδή εικόνες, video, ήχος) και όχι του περιεχομένου που βασίζεται σε κείμενο, όπως είναι οι πιο συνηθισμένοι πόροι περιεχομένου που ήδη γνωρίζουμε. Μερικές από τις ιδιαιτερότητες που παρουσιάζει το multimedia περιεχόμενο είναι:

- Αρκετά μεγάλα μεγέθη αρχείων και υψηλός απαιτούμενος αποθηκευτικός χώρος
- Διαφορετικές παρεχόμενες αναλύσεις και επίπεδα ποιότητας του περιεχομένου
- Η διάθεσή του είναι δυνατή με την μορφή streaming audio και streaming video

Τα παραπάνω χαρακτηριστικά είναι πολύ σημαντικό να ληφθούν υπ' όψη όταν σχεδιάζεται το μοντέλο δεδομένων ενός συστήματος περιεχομένου που διαχειρίζεται online multimedia περιεχόμενο. Το πολυμεσικό περιεχόμενο χαρακτηρίζεται από ιδιαίτερα μορφή και διαχείριση. Για τον λόγο αυτό μπορούμε να θεωρήσουμε τους πόρους του πολυμεσικού περιεχομένου ως ξεχωριστή υποκατηγορία πόρων. Κάθε μια από αυτές τις υπο-οντότητες περιέχει και πληροφορία για την αποδοτική διαχείρισή της. Πιο συγκεκριμένα σε κάθε οντότητα μπορεί να υπάρχει ένα link σε κατάλληλο αντικείμενο στο οποίο υπάρχει περιγραφή για το πού βρίσκεται το πραγματικό αρχείο ή το τμήμα του αρχείου.

Επιπρόσθετα, είναι επιθυμητή η δυνατότητα αναλυτικής αναζήτησης με βάση το multimedia περιεχόμενο (περιγραφή μιας εικόνας, τμημάτων ενός αρχείου ήχου, ή εύρεση frames με κάποια χαρακτηριστικά μέσα σε ένα video). Οι αλγόριθμοι που πρέπει να χρησιμοποιηθούν για κάτι τέτοιο διαφέρουν βέβαια από την αναζήτηση περιεχομένου που βασίζεται σε κείμενο και απαιτούν συνήθως την ενσωμάτωση στο περιεχόμενο και χρήση των κατάλληλων metadata. Σχετικά με την αναζήτηση περιεχομένου σε multimedia περιεχόμενο (π.χ. σε μια εικόνα) μέσω ενός DAM συστήματος μπορεί να γίνει αναζήτηση εννοιών όπως χρώμα, υφές, αποχρώσεις, συγκεκριμένων οντοτήτων για το αν βρίσκονται ή όχι στην εικόνα (π.χ. σπίτι ή καλύβια ή ουρανοξύστης) και τέλος ακόμα και πιο αφηρημένων εννοιών που χαρακτηρίζουν το περιεχόμενο της εικόνας (π.χ. παραγωγικότητα, μελαγχολία, ικανοποίηση). Για την υλοποίηση της αποδοτικής αναζήτησης σε multimedia περιεχόμενο αλλά και την αποδοτική συμπίεσή του για δημοσίευση στο Διαδίκτυο, υπάρχει ο αλγόριθμος MPEG 7.

### **3.1.7 Συστήματα Διαχείρισης Τίτλων - Brand management system**

Δημιουργώντας μια εξειδίκευση στα συστήματα DAM, τα Brand management systems είναι συγκεκριμένες εφαρμογές για την διαχείριση του multimedia διαφημιστικού και προωθητικού υλικού μιας εταιρείας, που απευθύνονται κυρίως στα τμήματα διαφήμισης και marketing μιας εταιρείας.

### **3.1.8 Συστήματα Διαχείρισης Ψηφιακών Δικαιωμάτων - Digital Rights Management (DRM) system**

Αφορούν όπως προκύπτει και από το όνομά τους σε συστήματα διαχείρισης των πνευματικών δικαιωμάτων δημιουργών multimedia κυρίως περιεχομένου. Τέτοιοι δημιουργοί-κάτοχοι πνευματικής ιδιοκτησίας μπορεί να είναι συνθέτες, στιχουργοί, ποιητές, συγγραφείς βιβλίων και παραγωγοί κινηματογραφικών ταινιών. Προσφέρουν δυνατότητα διανομής / παρουσίασης του περιεχομένου στους χρήστες εκείνους που έχουν νομικά το δικαίωμα να το χρησιμοποιήσουν. Για παράδειγμα αυτό ισχύει στην περίπτωση της διανομής αρχείων μουσικής ή video με ένα Web CMS μέσω του Διαδικτύου. Τα συστήματα αποτελούν συνήθως στην πράξη μια πρόσθετη λειτουργικότητα που προσφέρεται για τα συστήματα διαχείρισης ψηφιακών πόρων (digital assets), αφού συνήθως τέτοιου είδους περιεχόμενο συνοδεύεται και από κατοχυρωμένες πατέντες και πνευματικά δικαιώματα.

### **3.1.9 Portals - Vortals**

Παραδείγματα portals είναι τα Yahoo, MSN, In.gr, Flash.gr. Μέσα από τα portal προσφέρεται πρόσβαση σε ένα μεγάλο εύρος διαδικτυακού περιεχομένου, το οποίο είναι αποθηκευμένο σε πολλά διαφορετικά σημεία του Ιστού. Αυτό επιτυγχάνεται με την δυνατότητα σύνθετης αναζήτησης στον Ιστό και επιστροφή αποτελεσμάτων από πολλές πηγές (εξελιγμένες μηχανές αναζήτησης) όσο και από την κατάλληλη ευρετηριοποίηση του Ιστού σε θεματικές κατηγορίες. Τα portals προσφέρουν την δυνατότητα προσωποποίησης (personalization) της παρουσίασης του είδους και της μορφής του περιεχομένου στον χρήστη, με βάση τις προσωπικές του προτιμήσεις. Εξειδίκευση των portals αποτελούν τα vortals (Vertical Portals), τα οποία παρέχουν παρόμοιες υπηρεσίες και λειτουργικότητα στους χρήστες τους, με τη διαφορά ότι διαθέτουν περιεχόμενο για μια συγκεκριμένη θεματική ενότητα (π.χ. για τεχνολογία ή υγεία, κ.ά.)

Μερικά από τα πιο σημαντικά κοινά χαρακτηριστικά που διαθέτουν τα portals είναι τα εξής:

- Ευκολία στην χρήση και προσωποποίηση
- Προσφορά δυνατοτήτων όπως File sharing, Discussion Forum, Email, ως εργαλεία αύξησης της παραγωγικότητας των χρηστών των portals
- Εύκολη πρόσβαση στο αποθηκευμένο περιεχόμενο, ανάλογα με το επίπεδο των δικαιωμάτων πρόσβασης του κάθε χρήστη
- Ενσωμάτωση διαφορετικών ετερογενών πηγών περιεχομένου στο ευρετήριο του portal και επίσης δυνατότητα δημιουργίας δεσμών μεταξύ τους
- Δυνατότητα εκτέλεσης εργασιών συντήρησης και γενικής διαχείρισης του portal, ώστε να μπορεί διαρκώς να ανανεώνεται και εξελίσσεται τόσο σε δομή όσο και σε εμφάνιση.

Τέλος μερικές ακόμα κατηγορίες πληροφοριακών συστημάτων διαχείρισης περιεχομένου είναι οι εξής:

### **3.1.10 Library management system (LMS)**

Τα συστήματα διαχείρισης περιεχομένου βιβλιοθηκών (LMS) παρέχουν μια ολοκληρωμένη λύση για την διαχείριση όλων των τεχνικών λειτουργιών και υπηρεσιών μιας μεγάλης βιβλιοθήκης προς το ευρύ κοινό. Αυτές οι υπηρεσίες μπορεί να σχετίζονται με την καταγραφή, αποθήκευση και διαχείριση όλων των ειδών περιεχομένου που ανήκει στην βιβλιοθήκη. Επίσης έχουν να κάνουν με τον έλεγχο του δανεισμού ψηφιακού και μη περιεχομένου καθώς και όλες τις υποστηρικτικές και διοικητικές υπηρεσίες της βιβλιοθήκης.

### **3.1.11 Digital imaging system (DIS)**

Τα συστήματα ψηφιοποίησης εγγράφων (DIS) αυτοματοποιούν την δημιουργία ηλεκτρονικών εκδόσεων περιεχομένου που βρίσκεται σε έντυπη μορφή, όπως για παράδειγμα δημιουργία αρχείων PDF και TIFF για φωτογραφίες και εικόνες. Το αποτέλεσμα της εργασίας των DIS χρησιμεύει συνήθως ως είσοδος στα συστήματα Records management systems (RMS).

### **3.1.12 Learning content management system (LCMS)**

Υπάρχουν αρκετά CMS που ασχολούνται με την διαχείριση ακαδημαϊκού και εκπαιδευτικού υλικού με στόχο την οργάνωσή του και την διάθεσή του στα κατάλληλα κανάλια στα μέλη των ακαδημαϊκών ιδρυμάτων, όπως φοιτητές, μαθητές, δασκάλους, καθηγητές και ερευνητές. Στόχος των συστημάτων αυτών είναι η ιεραρχική οργάνωση και ενιαία παρουσίαση υλικού σχετικού με κάποια θεματική ακαδημαϊκή περιοχή. Το υλικό αυτό δεν στοχεύει στην πλήρη αντικατάσταση της ζωντανής διδακτικής διαδικασίας, αλλά δρα επικουρικά για κάθε ενδιαφερόμενο σπουδαστή ή καθηγητή. Ως παράδειγμα μπορούμε να αναφέρουμε την προσπάθεια ενοποίησης των web sites για τα διδασκόμενα μαθήματα του Εργαστηρίου Συστημάτων Βάσεων Γνώσεων και Δεδομένων του Εθνικού Μετσόβιου Πολυτεχνείου και η οποία υλοποιήθηκε με τη χρήση της πλατφόρμας COMMET.

Ως μια εξέλιξη των παραπάνω συστημάτων, δημιουργήθηκαν κάποια περισσότερο δυναμικά συστήματα, τα οποία δεν στοχεύουν μόνο στην διαχείριση και προβολή βοηθητικού περιεχομένου, αλλά και στην παροχή ενός ολοκληρωμένου συστήματος εκπαίδευσης που υποκαθιστά τη

εκπαιδευτική διαδικασία στην τάξη. Τα συστήματα αυτά προσφέρουν την δυνατότητα για ενοποιημένη διαχείριση τόσο του εκπαιδευτικού περιεχομένου όσο και ενός ολοκληρωμένου προγράμματος εκπαίδευσης. Παραδείγματα αυτού του είδους μπορεί να μην είναι ιδιαίτερα διαδεδομένα στην Ελλάδα, έχουν όμως μεγαλύτερη απήχηση στο εξωτερικό και λειτουργούν υποστηρικτικά για τις διαδικασίες της απομακρυσμένης εκπαίδευσης και μάθησης (e-learning) από εκπαιδευτικούς οργανισμούς και Πανεπιστήμια.

### **3.1.13 Διαχείριση γνώσης (Knowledge management - KM)**

Αφορά στην συλλογή περιεχομένου και στην κατάλληλη επεξεργασία του με στόχο την σύνθεση των πληροφοριών σε μια ανώτερη έννοια, την γνώση. Η γνώση είναι λοιπόν η αξιοποίηση της συσσωρευμένης πληροφορίας. Ένα σύστημα διαχείρισης γνώσης βοηθά σημαντικά έναν οργανισμό να “γνωρίζει τι γνωρίζει”. Αυτό το πετυχαίνει με συστηματική καταγραφή και κατάλληλη επεξεργασία (π.χ. data mining και αναφορές πάνω στα αποτελέσματα του data mining) των δομημένων και ημιδομημένων πληροφοριών (υπό μορφή είτε ποσοτικών είτε ποιοτικών στοιχείων) που διαθέτει η εταιρεία από διάφορες πηγές και με διάφορες μορφές. Ως παραδείγματα τέτοιου περιεχομένου θα μπορούσαν να αναφερθούν emails, αρχεία, βάσεις δεδομένων, υπομνήματα, σημειώσεις, αναφορές, απολογισμοί και προϋπολογισμοί, πρακτικά συνεδριάσεων, κ.ά. Τελικός στόχος συνήθως είναι η δυνατότητα λήψης αποφάσεων σχετικά με μελλοντικές επιχειρηματικές ενέργειες και με τη χρήση αρκετά συχνά βοηθητικών εφαρμογών υποστήριξης λήψης αποφάσεων (decision support systems DSS). Στην πράξη πολλά εξελιγμένα πακέτα CMS, πέραν των βασικών λειτουργικών τους χαρακτηριστικών, διαθέτουν προχωρημένα χαρακτηριστικά που σχετίζονται με το knowledge management.

Η διαχείριση γνώσης σαν θεματική περιοχή μπορεί να μην αφορά μόνον στο πεδίο των επιχειρήσεων, στοχεύει όμως σε κάθε περίπτωση κυρίως στην λήψη αποφάσεων βασισμένων στην υπάρχουσα πληροφορία και την δομημένη οργάνωση της υπάρχουσας πληροφορίας.

### **3.1.14 Geographic information system (GIS)**

Τα γεωγραφικά συστήματα πληροφοριών (GIS) είναι πληροφοριακά συστήματα ειδικού σκοπού για την σύλληψη, αποθήκευση, ανάκτηση, επεξεργασία και παρουσίαση μιας σειράς χωρικών δεδομένων και περιεχομένου.

### **3.1.15 Software Configuration Management (SCM)**

Διαχείριση των εγκαταστάσεων (συνήθως μαζικές εγκαταστάσεις μέσω των τοπικών δικτύων της επιχείρησης) και παραμέτρων λειτουργίας οποιουδήποτε είδους λογισμικού έχει εγκατασταθεί και λειτουργεί σε μια επιχείρηση. Συστήματα αυτού του είδους καλύπτουν όλον τον κύκλο ζωής ενός προϊόντος λογισμικού που εγκαθίστανται στα μηχανήματα μιας εταιρείας, από την εγκατάσταση και την αναβάθμιση του εκάστοτε λογισμικού μέχρι τον οριστικό τερματισμό της λειτουργίας και την αντικατάστασή του.



## 3.2 Με βάση το κόστος απόκτησης και συντήρησης.

### 3.2.1 Εισαγωγή

Η ομαδοποίηση των προϊόντων CMS μπορεί να γίνει σύμφωνα με την άδεια και τους όρους χρήσης που συνοδεύουν κάθε προϊόν. Τα πακέτα CMS που διατίθενται για εμπορική χρήση εντάσσονται στις παρακάτω κατηγορίες:

- Διατίθενται τα αρχεία εγκατάστασης και μαζί κάποιο API για την επέκταση του CMS, αλλά δεν παρέχεται ο πηγαίος κώδικας του συστήματος. Προσφέρεται τεχνική υποστήριξη.
- Διατίθενται τα αρχεία εγκατάστασης μαζί με τον πηγαίο κώδικα είτε όλου του συστήματος είτε μερικών από τις συνοδευτικές εφαρμογές, κατ'επιλογή του τελικού χρήστη. Προσφέρεται τεχνική υποστήριξη.
- Δίνονται τα αρχεία εγκατάστασης και ο πηγαίος κώδικας δωρεάν, αλλά δεν προσφέρεται τεχνική υποστήριξη. Για την παροχή τεχνικής υποστήριξης ο χρήστης πρέπει να καταβάλλει συνδρομή.

Εκτός από τα παραπάνω πρέπει να ληφθεί ακόμα υπ' όψη ότι κάθε προϊόν CMS είναι σχεδιασμένο για να λειτουργεί σε κάποιο λειτουργικό σύστημα και απαιτεί την ύπαρξη πρόσθετων συστατικών λογισμικού. Τόσο το λειτουργικό σύστημα όσο και τα πρόσθετο απαιτούμενο λογισμικό (για παράδειγμα ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων) ονομάζονται Λογισμικό Υποδομής. Κάποια CMS είναι σχεδιασμένα ώστε να λειτουργούν σε open source περιβάλλοντα. Κάποια άλλα σχεδιάζονται ώστε να συνδυαστούν με proprietary λογισμικό υποδομής. Συνεπώς η άδεια χρήσης ενός CMS συμπεριλαμβάνει σιωπηρά και την προϋπόθεση της απόκτησης των κατάλληλων αδειών για χρήση του λογισμικού υποδομής. Ο πίνακας που ακολουθεί επεξηγεί όλους τους δυνατούς συνδυασμούς αδειών που μπορεί να προσφέρει ένα πακέτο CMS.

**Πίνακας 2 Συνδυασμοί αδειών για ένα πακέτο CMS**

	Αρχεία εγκατάστασης + API	Πηγαίος κώδικας διαθέσιμος με τεχνική υποστήριξη	Πηγαίος κώδικας διαθέσιμος χωρίς τεχνική υποστήριξη
Open Source λογισμικό υποδομής			
Proprietary λογισμικό υποδομής			

Για περισσότερα στοιχεία σχετικά με τις άδειες λογισμικού που αφορούν το λογισμικό υποδομής των Συστημάτων Διαχείρισης Περιεχομένου, ο αναγνώστης μπορεί να ανατρέξει στο παράρτημα Γ.

Ανεξαρτήτως τώρα της άδειας χρήσης που συνοδεύει κάθε πακέτο CMS, διακρίνουμε την παρακάτω κατηγοριοποίηση. Η κατηγοριοποίηση αυτή προκύπτει με βάση το κόστος κτήσης και συντήρησης ενός πακέτου, το οποίο είναι άμεση απόρροια των δυνατοτήτων του πακέτου και πιο συγκεκριμένα του αγοραστικού κοινού στο οποίο απευθύνεται.

### 3.2.2 Πακέτα Enterprise platforms

Πλατφόρμες Web CMS μεγάλης κλίμακας οι οποίες πωλούνται ως συστατικά ενός ευρύτερου πακέτου εργαλείων CMS. Απευθύνονται σε πολυεθνικές επιχειρήσεις και οργανισμούς μεγάλης κλίμακας. Οι τιμές τους ξεκινούν από τα 200-250 χιλιάδες δολάρια ΗΠΑ.

#### *Stellent Universal Content Management Suite*

Το πακέτο στοχεύει στην καταγραφή και οργάνωση όλων των ειδών περιεχομένου που κατέχει μια μεγάλης κλίμακας επιχείρηση, όπως για παράδειγμα λογιστικά φύλλα, έγγραφα, καταλόγους προϊόντων, νομικά έγγραφα και προωθητικό υλικό. Μέσα από το περιβάλλον της πλατφόρμας μπορεί η επιχείρηση να εφαρμόσει τεχνικές Web Content Management, Document Management, Collaboration Management, Records management και Digital assets management μέσα από ένα ενιαίο γραφικό περιβάλλον, απλά χρησιμοποιώντας κάθε φορά τα κατάλληλα εργαλεία.

Η βάση όλων είναι η μηχανή Stellent Content Server, η οποία αποτελείται από ένα web-based repository και μια σειρά από βοηθητικά services. Οι υπηρεσίες αυτές είναι διαθέσιμες σε όλα τα είδη περιεχομένου που αποθηκεύονται στο repository και περιλαμβάνουν υπηρεσίες βιβλιοθήκης, ασφάλειας, μετατροπών στην δομή και μορφοποίηση του περιεχομένου, διαχείρισης της ροής εργασιών (workflow), προσωποποίησης, δεικτοδότησης-αναζήτησης δεδομένων καθώς και πολλές υπηρεσίες διαχείρισης του ίδιου του συστήματος. Με βάση αυτές τις βοηθητικές υπηρεσίες έχουν δημιουργηθεί τα εξής modules εφαρμογών. Όλα τα modules αυτά βασίζονται στις θεμελιώδεις υπηρεσίες που αναφέρθηκαν, αλλά προσφέρουν στην συνέχεια με την σειρά τους κάποιες πρόσθετες υπηρεσίες ανώτερου επιπέδου που έχουν να κάνουν με την διαχείριση του εξειδικευμένου περιεχομένου με το οποίο κάθε application module καλείται να εργαστεί. Τα application modules είναι τα ακόλουθα:

- Stellent Web Content Management
- Stellent Document Management
- Stellent Collaboration Management
- Stellent Records Management
- Stellent Digital Asset Management

#### *Vignette - V7 Content Management Suite*

Περιλαμβάνει τις παρακάτω βασικές υπηρεσίες, τις οποίες η εταιρεία εμπορεύεται ανεξάρτητα ή και ως συνολικό πακέτο [VIG04]:

- Content Services Μια σειρά από υπηρεσίες για την συλλογή περιεχομένου από τους authors και άλλες εξωτερικές πηγές καθώς και την δυνατότητα δημοσίευσης του περιεχομένου αυτού στους κατάλληλους αποδέκτες.
- Portal Services Web-based εργαλεία που προσφέρουν ασφαλή και προσωποποιημένη πρόσβαση σε εφαρμογές, υπηρεσίες και στο ίδιο το περιεχόμενο που αντλείται από διάφορες πηγές.
- Integration Services Με την χρήση των λεγόμενων pre-packaged integration modules, δίνει την δυνατότητα για ενσωμάτωση δομημένων και μη δομημένων πληροφοριών σε ένα repository για ενιαία διαχείριση.
- Collaboration Services βοηθούν στην διάχυση της γνώσης και του περιεχομένου μεταξύ των εργαζομένων ενός μεγάλου οργανισμού ή μιας εταιρείας, με στόχο την αποτελεσματικότερη συνεργασία. Αυτά τα modules δίνουν την δυνατότητα ενσωμάτωσής τους με προγράμματα επικοινωνίας και συνεργασίας όπως είναι οι πελάτες ηλεκτρονικού ταχυδρομείου και Digital Dashboards, που είναι εφαρμογές για την απομακρυσμένη από κοινού χρήση μιας εφαρμογής ή την διαχείριση ενός project.

- Process Services Βοηθούν τόσο το τεχνικό όσο και το διοικητικό προσωπικό μιας εταιρείας να εμπλακεί στην διαδικασία δημιουργίας, επεξεργασίας και δημοσίευσης του περιεχομένου της εταιρείας, κάνοντας χρήση εξελιγμένων business workflows.
- Analysis Services επιτρέπουν στα διοικητικά στελέχη να μετρούν και να ελέγχουν την αποτελεσματικότητα των Web sites , των portals καθώς και των διεργασιών workflow.

Πρόσφατο παράδειγμα εφαρμογής της σουίτας εργαλείων διαχείρισης περιεχομένου V7 είναι το site των Ολυμπιακών Αγώνων Athens2004.com που “έτρεχε” Vignette.

### *Documentum (EMC) - Web Publisher 5.2*

Τα συστατικά του πακέτου της Documentum είναι τα εξής:

- Content Rendition Services  
Αυτοματοποιημένη διαδικασία μετατροπής παραδοσιακών μορφών εγγράφων σε web oriented formats, όπως είναι το HTML και το PDF ή XML και αποθήκευσης των τελικών μορφών σε ένα κατάλληλο repository.
- Content Server  
Παρέχει την δυνατότητα για ενοποιημένη διαχείριση όλων των μορφών περιεχομένου, όπως XML/HTML, αλλά και γραφικά, πολυμέσα και παραδοσιακά formats κειμένου.
- FTP Services  
Διασφαλίζει την ασφαλή μεταφορά από και προς το repository κάθε τύπου περιεχομένου
- Site Deployment Services  
Εξάγει περιεχόμενο από ένα repository και το δημοσιεύει σε μορφή web σελίδων, εύκολα και γρήγορα, ακόμα και για μεγάλης κλίμακας web sites
- Web Publisher  
Εύκολο στη χρήση γραφικό web browser based εργαλείο, που επιτρέπει στους μη τεχνικούς χρήστες να δημιουργούν, να επεξεργάζονται και να δημοσιεύουν περιεχόμενο σε οποιοδήποτε Web site τους έχει αποδοθεί πρόσβαση.

### *FileNET - P8 Web Content Manager*

Μερικά από τα βασικά χαρακτηριστικά του πακέτου είναι τα ακόλουθα:

- Το περιεχόμενο δημοσιεύεται σε πολλά sites, formats, κανάλια και μεταφράζεται σε πολλές γλώσσες
- Εύκολη δημιουργία, παραμετροποίηση και διαχείριση μεγάλων δικτυακών τόπων
- Μείωση του κόστους επανεκπαίδευσης των στελεχών της εταιρείας στην χρήση ενός νέου συστήματος διαχείρισης περιεχομένου
- Επαναχρησιμοποίηση πόρων και συστατικών περιεχομένου και έτοιμα templates
- FileNet P8 μπορεί εύκολα να ενσωματωθεί με υπάρχουσες μεγάλης κλίμακας υλοποιημένες επιχειρηματικές εφαρμογές, οι οποίες ήδη λειτουργούν σε μια επιχείρηση. Έτσι επιτυγχάνεται η μέγιστη απόδοση στο σύστημα και οικονομικά οφέλη για την επιχείρηση

### *Interwoven – Live Site Web Content Management Suite*

Ορισμένα από τα κύρια χαρακτηριστικά του προϊόντος είναι τα παρακάτω:

- Δημοσίευση περιεχομένου με βάση την λογική του WYSIWYG  
Με την χρήση έτοιμων και χρηστικών layouts και με γραφικό περιβάλλον και εργαλεία web browser based , ακόμα και το διοικητικό προσωπικό μιας επιχείρησης χωρίς τεχνικές γνώσεις μπορεί να δημιουργήσει και να διαχειριστεί περιεχόμενο σε πολύ σύντομο χρονικό διάστημα.
- Χρήση συλλογών περιεχομένου βασισμένων σε συστατικά (components)

- Διαχείριση και επισκόπηση των αλλαγών στο περιεχόμενο μέσα από το γραφικό Web browser based περιβάλλον
- Μετατροπές και αλλαγή στο περιεχόμενο κατευθείαν μέσα από μια ιστοσελίδα, χωρίς να απαιτείται να μεταβεί ο χρήστης σε ξεχωριστές σελίδες παραμετροποίησης
- Κεντρικό σημείο διαχείρισης των web sites και Intranets και αυτοματοποιημένος έλεγχος του συστήματος διαχείρισης περιεχομένου

### 3.2.3 Πακέτα Upper Tier

Τα πακέτα αυτά στοχεύουν σε τμήματα και διευθύνσεις μεγάλων επιχειρήσεων και οι περισσότερες προσφερόμενες λειτουργίες εστιάζουν αποκλειστικά σε διαχείριση διαδικτυακού περιεχομένου. Το μέσο κόστος κτήσης ξεκινά από τα 125-175 χιλιάδες δολάρια ΗΠΑ.

#### *FatWire - Content Server 5.5*

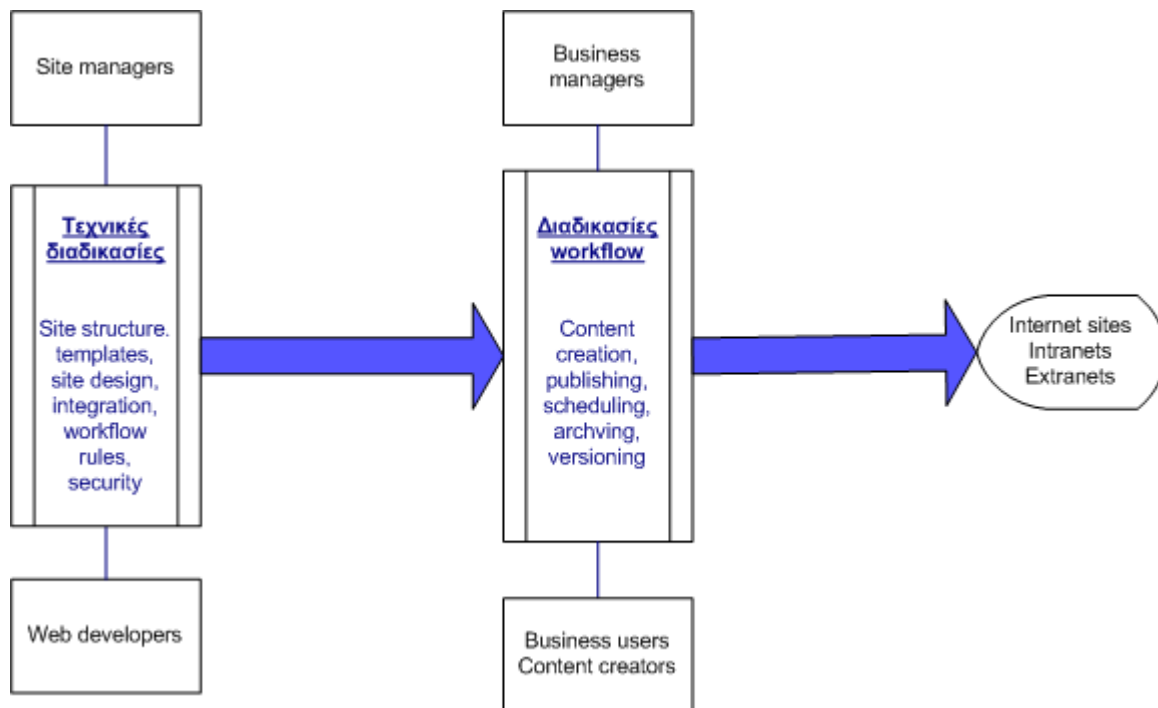
Μερικά από τα βασικότερα χαρακτηριστικά του πακέτου είναι:

- Διαχείριση εταιρικού περιεχομένου  
Προσφέρεται ευκολία στην δημιουργία και διαχείριση κάθε είδους εταιρικού περιεχομένου συμπεριλαμβανομένων των εγγράφων, προϊόντων καταλόγων καθώς και του πολυμεσικού περιεχομένου (digital assets) για όλους τους χρήστες του πακέτου, με βάση βέβαια κάποιον έλεγχο της πρόσβασης. Επίσης υπάρχει η δυνατότητα διασύνδεσης σε πραγματικό χρόνο με προγράμματα back-office και ελέγχου του business workflow.
- Ευκολία χρήσης και συντήρησης  
Με τα εύκολα web browser based εργαλεία και το φιλικό γραφικό περιβάλλον της η πλατφόρμας διευκολύνει όλους τους συμμετέχοντες στις διαδικασίες του content management να συνεισφέρουν και να διαχειρίζονται περιεχόμενο.
- Υψηλή διαθεσιμότητα και επιδόσεις  
Χρησιμοποιεί τεχνολογίες που βασίζονται σε Java(όπως το JSR 168 specification για ενσωμάτωση με portal) και J2EE application servers, όπως είναι οι BEA, IBM, Sun, και JBoss.

#### *Microsoft - Content Management Server 2002*

Βασίζεται στο μοντέλο δεδομένων του .NET και προσφέρει δυνατότητες διαχείρισης και δημοσίευσης περιεχομένου σε web sites, intranets και extranets, ακολουθώντας την εξής φιλοσοφία σχεδίασης για μια υποθετική εταιρεία που το χρησιμοποιεί:

Αρχικά αναλαμβάνει δράση το τεχνικό προσωπικό της εταιρείας, το οποίο δημιουργεί την βασική δομή και την λειτουργικότητα του site. Εν συνεχεία τα διοικητικά στελέχη και οι δημιουργοί και διαχειριστές περιεχομένου μπορούν να δημιουργούν, να επεξεργάζονται και να δημοσιεύουν περιεχόμενο χωρίς την ανάγκη για παροχή τεχνικής υποστήριξης σε κάποια από τα στάδια του workflow. Όλα τα παραπάνω αντικατοπτρίζονται καλύτερα στο παρακάτω σχήμα:



**Σχήμα 6: Διαδικασία δημιουργίας CMS εφαρμογών**

Το παραπάνω σχήμα είναι πολύ αντιπροσωπευτικό, διότι αναπαριστά την διαδικασία που ακολουθείται από τα περισσότερα WCMS για τον σχεδιασμό και την συντήρηση ενός site περιεχομένου. Τα τρία βασικά συστατικά του CM Server 2002 είναι τα εξής:

- Web Author Client, για την διαχείριση και δημοσίευση περιεχομένου απευθείας σε κάποιο site και προεπισκόπησή του μέσα από κάποιον web browser
- Authoring Connector for Microsoft Office, που προσφέρει την δυνατότητα διαχείρισης του περιεχομένου μέσα από τις εφαρμογές του Office και συνεπώς δίνει την δυνατότητα να εκμεταλλευτούν οι απλοί χρήστες όλα τα πρόσθετα χαρακτηριστικά διαχείρισης που προσφέρουν οι εφαρμογές αυτές.
- Connector for SharePoint Technologies, που προσφέρουν την δυνατότητα ενσωμάτωσης τεχνολογιών που διαθέτουν τα προϊόντα Office SharePoint Portal Server και Windows SharePoint Services, και οι οποίες σχετίζονται κυρίως με υπηρεσίες συνεργασίας και επικοινωνίας.

### 3.2.4 Πακέτα Mid-Market

Αυτά τα πακέτα στοχεύουν στις επιχειρήσεις και οργανισμούς μεσαίας κλίμακας ή σε επιμέρους τμήματα μεγάλων πολυεθνικών. Το μέσο κόστος κυμαίνεται στα 40-100 χιλιάδες δολάρια ΗΠΑ. Κοινό χαρακτηριστικό τους η ευκολία εγκατάστασης και διαχείρισης αλλά παράλληλα ως τίμημα διαθέτουν μικρά περιθώρια παραμετροποίησης.

#### *Serena - (Merant) Collage 4.5*

Αποτελεί μια λύση διαχείρισης διαδικτυακού περιεχομένου για ανάπτυξη και συντήρηση σχετικά μεγάλων web sites . Είναι υλοποιημένο σε αρχιτεκτονική J2EE και έτσι είναι εύκολα μεταφέρσιμο σε οποιαδήποτε πλατφόρμα υποστηρίζει Java. Το περιβάλλον και τα εργαλεία βασίζονται σε παράθυρα browser και γραφικά περιβάλλοντα χρήστη. Παρέχονται λειτουργίες διαχείρισης εργασιών (task management), διαχείρισης ροής (workflow) και προσφέρεται επίσης η δυνατότητα εύκολης ενσωμάτωσης με τις εφαρμογές Microsoft Office και Macromedia Dreamweaver. Τέλος είναι πολύ

απλουστευμένη και η διαδικασία εισαγωγής και διαχείρισης υπαρχόντων web sites που έχουν υλοποιηθεί με οποιαδήποτε άλλη πλατφόρμα.

### **3.2.5 Πακέτα Lower-priced**

Αυτά τα πακέτα είναι τυπικά συστήματα για διαχείριση διαδικτυακού περιεχομένου με μέσο κόστος κτήσης τα 5 – 25 χιλιάδες δολάρια ΗΠΑ

#### *Ektron - CMS100/200/300/400*

Το πακέτο υλοποιεί κυρίως ανάπτυξη και διαχείριση websites και επίσης βοηθά στην καλύτερη εσωτερική οργάνωση των επιχειρήσεων με βελτίωση του business workflow. Θετικά αποτελέσματα της χρήσης ενός τέτοιου πακέτου από μικρές και μικρομεσαίες επιχειρήσεις είναι η ικανοποίηση του πελάτη και μια αποδοτική απόσβεση κεφαλαίου (Return of Investment – ROI) . Το CMS400.NET είναι υλοποιημένο με βάση το .NET framework και είναι η βάση για την ανάπτυξη διαδικτυακών εφαρμογών και διαχείρισης περιεχομένου με χρήση web browser based εργαλείων.

### **3.2.6 Πακέτα ASP**

Τα πακέτα αυτά είναι συστήματα CMS τα οποία τρέχουν σε εξωτερικούς servers των εταιριών που τα προσφέρουν, είτε για πώληση είτε για ενοικίαση. Από άποψη κόστους βρίσκονται περίπου στην κατηγορία mid-market.

#### *iUpload - iUpload*

Σε ένα τυπικό e-business portal του iUpload το περιεχόμενο οργανώνεται στις εξής τρεις κατηγορίες:

- Περιεχόμενο δημιουργούμενο από το προσωπικό της επιχείρησης  
Κατάλογοι , έγγραφα και projects από τα διάφορα τμήματα και της διευθύνσεις μια επιχείρησης που πρέπει να γρήγορα διαθέσιμο σε διάφορα κανάλια και κυρίως στο Web.
- Περιεχόμενο των εφαρμογών  
Δημιουργείται από εφαρμογές και διεργασίες που τρέχουν στο εκάστοτε site, όπως μηχανές αναζήτησης, φόρμες συλλογής στοιχείων , κ.ά.
- Περιεχόμενο πελατών ή επισκεπτών της επιχείρησης  
Προέρχεται από τους επισκέπτες των sites μέσα από μερικά εργαλεία όπως φόρμες, polls, chat rooms και εγγραφές σε newsletters.

Η πλατφόρμα διαθέτει τα παρακάτω χαρακτηριστικά, η λειτουργία των οποίων είναι προφανής από τους τίτλους τους:

- Content Manager
- eMarketing Manager
- Content Search
- Survey Manager
- Collaboration Manager

### **3.2.7 Πακέτα Delivery-Oriented**

Αυτά τα πακέτα, τα οποία είναι συνήθως portal και application servers στοχεύουν κυρίως στην παροχή των ακόλουθων χαρακτηριστικών: δημιουργία δυναμικών web σελίδων, προσωποποίηση, caching και μια σειρά από άλλες υπηρεσίες που παρέχονται κατά τον χρόνο εκτέλεσης των συνοδευτικών εφαρμογών. Τα πακέτα αυτά αποτελούν τμήμα μιας μεγαλύτερης πλατφόρμας υπηρεσιών, όπως φαίνεται από τα παραδείγματα που ακολουθούν.

### *Oracle – Oracle10g*

Το νέο προϊόν-πλατφόρμα της Oracle, το Application Server 10g παρέχει μια ολοκληρωμένη λύση για ανάπτυξη, ενσωμάτωση με υπάρχουσες τεχνολογίες και διάθεση μεγάλης κλίμακας επιχειρηματικών δικτυακών εφαρμογών, portals, και Web sites. Όλες οι παραπάνω εφαρμογές αναπτύσσονται στον Application Server με βάση έναν J2EE server, λογισμικό για portals και μια σειρά από βοηθητικές εφαρμογές ενσωμάτωσης της επιχειρηματικής λογικής σε κάθε εφαρμογή. Το προϊόν έχει σχεδιαστεί για το κατανεμημένο υπολογιστικό μοντέλο grid computing και βασίζεται σε αρχιτεκτονική βασισμένη σε υπηρεσίες (Service Oriented Architecture - SOA).

### *IBM - Websphere Portal*

Η πρόταση της IBM για μια πλατφόρμα ανάπτυξης λύσεων διαχείρισης περιεχομένου και internet portals και intranet sites ακούει στο όνομα Websphere Portal. Το προϊόν αυτό τρέχει κάτω από Lotus Domino και IBM Websphere και ενσωματώνει δυνατότητες του IBM DB2 Content manager. Το Websphere Portal έρχεται σε δύο εκδόσεις:

- IBM Workplace Web Content Management  
Κατάλληλη για μεγάλες και πολύ μεγάλες επιχειρήσεις και οργανισμούς.
- IBM Workplace Web Content Management, Standard Edition  
Κατάλληλη για μικρές και μικρομεσαίες επιχειρήσεις.

### **3.2.8 Πακέτα Open Source**

#### *Zope - Content Management Framework / Plone*

Είναι μια από τις γνωστότερες online κοινότητες ανάπτυξης συστημάτων διαχείρισης περιεχομένου και σύνθετων δικτυακών εφαρμογών. Βασίζεται στην γλώσσα προγραμματισμού Python και διαθέτει μια ενσωματωμένη αντικειμενοστρεφή βάση δεδομένων και μια εξελιγμένη μηχανή αναζήτησης που μπορεί να χρησιμοποιηθεί κατά την υλοποίηση των projects. Σχεδόν όλα τα εργαλεία ανάπτυξης και διαχείρισης είναι web based και παρέχεται έτσι η δυνατότητα σε όλους όσους αναπτύσσουν δικτυακές εφαρμογές με βάση το Zope να μπορούν να ελέγχουν τα project τους από οπουδήποτε. Η ευελιξία αυτή ενισχύεται ακόμα περισσότερο από ένα πολύ εξελιγμένο μοντέλο ασφάλειας και διαχείρισης της πρόσβασης στο περιεχόμενο των εφαρμογών. Το Zope διαθέτει πολυάριθμα plug-in, τα οποία επεκτείνουν τις βασικές δυνατότητες της πλατφόρμας. Μερικά από τα πιο γνωστά plug-in περιλαμβάνουν καινούργια αντικείμενα περιεχομένου, συνδέσεις για σχεσιακές βάσεις δεδομένων, ακόμα και έτοιμες πλήρεις εφαρμογές e-business και εργαλεία διαχείρισης περιεχομένου. Το Zope λειτουργεί με βάση τα πρωτόκολλα HTTP, FTP, WebDAV, και XML-RPC και μπορεί να λειτουργήσει και κάτω από άλλους διάφορους web servers.

Με βάση το Zope έχουν αναπτυχθεί μια σειρά από πλατφόρμες διαχείρισης περιεχομένου με προηγμένα χαρακτηριστικά, οι οποίες δίνουν την δυνατότητα στο χρήστη δημιουργίας και διαχείρισης εξελιγμένων web sites και portals, χωρίς να υπάρχουν ιδιαίτερες τεχνικές γνώσεις. Παραδείγματα υλοποιήσεων του Zope είναι το Content Management Framework (CMF) και το Plone, το οποίο επεκτείνει τις δυνατότητες του CMF με μια μεγάλη γκάμα έτοιμων templates και περιεχομένου που προσφέρεται από τα πολυάριθμα μέλη της κοινότητας Plone. Άλλη υλοποίηση είναι το Icoya που είναι με την σειρά του μια επέκταση του Plone. Το σύστημα Silva είναι μια web based υλοποίηση βασισμένη στο Zope, η οποία στοχεύει κυρίως στην διαχείριση σύνθετων εγγράφων. Τέλος, το Bizar Shop είναι ένα έτοιμο εμπορικό προϊόν που βασίζεται στην πλατφόρμα του Zope και προσφέρει την δυνατότητα δημιουργίας online καταστημάτων και εφαρμογών e-business.

### *OpenCms - OpenCms*

Είναι ένα επαγγελματικό open-source σύστημα διαχείρισης διαδικτυακού περιεχομένου. Χωρίς γνώση HTML και με την χρήση ενός WYSIWYG editor και φιλικό προς τον χρήστη GUI με έτοιμα templates μπορεί ο καθένας να δημιουργήσει και να διαχειριστεί περιεχόμενο με τελικό στόχο την ανάπτυξη και συντήρηση σύνθετων και μεγάλων σε κλίμακα web sites. Το OpenCms βασίζεται σε τεχνολογίες Java (JSP, Servlets) και XML. Εγκαθίσταται και τρέχει σε περιβάλλοντα open-source, όπως είναι ο συνδυασμός Linux, Apache, Tomcat, MySQL αλλά επίσης και σε πλατφόρμες με εμπορικό λογισμικό, όπως είναι ο συνδυασμός Windows NT, IIS, BEA Weblogic, Oracle DB. Για ευκολότερη εγκατάσταση το OpenCms διαθέτει έναν ενσωματωμένο HTML οδηγό εγκατάστασης (setup wizard). Απαιτείται ένα περιβάλλον με servlet container , όπως είναι ο Apache Tomcat μαζί με ένα DBMS. Με την αξιοποίηση του Web server στον οποίο εγκαθίσταται, το OpenCms προβάλλει όλες τις λειτουργίες διαχείρισης και δημιουργίας περιεχομένου μέσα από το περιβάλλον ενός browser.

### *Τυρο3 - Τυρο3*

Αποτελεί ένα μια πλατφόρμα ανάπτυξης λύσεων για διαχείριση περιεχομένου, η οποία μπορεί να εγκατασταθεί και να λειτουργήσει τόσο σε περιβάλλοντα Unix όσο και σε Windows. Ωστόσο η εγκατάσταση και παραμετροποίησή του απαιτεί αρκετές τεχνικές γνώσεις και μόνο χρήστες με τεχνικό υπόβαθρο θα είναι σε θέση να εκμεταλλευτούν πλήρως τις δυνατότητές του για να δημιουργήσουν λύσεις content management. Υποστηρίζει εξελιγμένο workflow, μια ποικιλία διαφορετικών formats και είναι πολύ επεκτάσιμο με μια σειρά από extension modules. Αυτό το καθιστά πολύ πλούσιο σε χαρακτηριστικά και κατάλληλο για χρήση από φορείς μεγάλης κλίμακας.

### *Apache Project - Cocoon Framework*

Πρόκειται για ένα πλαίσιο με open-source εργαλεία και τεχνολογίες για την ανάπτυξη web εφαρμογών. Η λογική της ανάπτυξης των προγραμμάτων είναι βασισμένη σε συστατικά (components) ενώ υπάρχει ένα εκτεταμένο δίκτυο προγραμματιστών και πολλά εργαλεία που διασφαλίζουν την επικοινωνία και συνεργασία για την ανταλλαγή τεχνογνωσίας μεταξύ τους. Μια προγραμματιστική έννοια που είναι πολύ διαδεδομένη και χρησιμοποιείται από το Cocoon είναι αυτή των component pipelines (χρήση σωληνώσεων για τα συστατικά λογισμικού). Κάθε συστατικό σε μια σωλήνωση εξειδικεύεται σε μια συγκεκριμένη λειτουργία. Με την κατάλληλη ενσωμάτωση μιας σειράς components σε κάθε pipeline επιτυγχάνεται κάθε φορά το επιθυμητό αποτέλεσμα, χωρίς να απαιτείται η συγγραφή μεγάλων τμημάτων κώδικα. Μια επίσης πολύ χρήσιμη προγραμματιστική ευκολία είναι ότι το Cocoon δίνει ένα πολύ εύχρηστο API για ενσωμάτωσή σε αυτό εφαρμογών και συστημάτων διαχείρισης περιεχομένου που υλοποιούνται από κάθε ανεξάρτητο φορέα, ο οποίος θέλει να αξιοποιήσει τις δυνατότητες του συστήματος Cocoon. Τέλος το Cocoon επιτρέπει την εισαγωγή και μετατροπή ενός μεγάλου αριθμού διαφορετικών τύπων αρχείων και formats σε XML.

### *Mambo - Mambo Server*

Άλλο ένα open-source εργαλείο για την δημιουργία τόσο σύνθετων web sites και portals , όσο και διαδικτυακών εφαρμογών κάθε κλίμακας. Διαθέτει και αυτό μια σειρά από πολλά βοηθητικά εργαλεία διαχείρισης και συνεργασίας και προσφέρει ένα οργανωμένο σύστημα βοήθειας για τα μέλη της κοινότητας του Mambo server.



### *TikiWiki - Tiki*

Βασισμένο στην γλώσσα PHP και την XML το TikiWiki ανήκει σε μια σειρά εφαρμογών, οι οποίες για εύκολη αναγνώριση διαθέτουν στο όνομά τους σαν πρώτο ή δεύτερο συνθετικό τη μνημονική λέξη Wiki. Αυτές οι εφαρμογές, με πρώτη και κύρια την ελεύθερη εγκυκλοπαίδεια Wikipedia, βασίζονται σε ελεύθερο λογισμικό και ανοικτά πρότυπα και στοχεύουν στην δημιουργία μιας ενιαίας πλατφόρμας διάχυσης και αξιοποίησης του λεγόμενου ελεύθερου περιεχομένου. Μερικά παραδείγματα εφαρμογών Wiki περιλαμβάνουν εφαρμογές Web, web sites, portals, intranets και extranets.

Συγκεκριμένα τώρα το TikiWiki είναι ένα εξελιγμένων χαρακτηριστικών CMS. Τα λειτουργικά χαρακτηριστικά που προσφέρει το TikiWiki μπορούν να διαιρεθούν στις ακόλουθες κατηγορίες:

- Δημιουργία περιεχομένου και εργαλεία ελέγχου  
Για να μπορούν οι διαχειριστές και οι χρήστες να συντηρούν περιεχόμενο διαφόρων μορφών, όπως κείμενο, δυαδικά αρχεία, εικόνες, δεσμούς, κ.ά.
- Εργαλεία διαχείρισης περιεχομένου και δημοσίευσης περιεχομένου  
Για την εσωτερική οργάνωση, δόμηση και παρουσίαση του περιεχομένου κατά τον επιθυμητό τρόπο.
- Εργαλεία επικοινωνίας  
Για την επικοινωνία μεταξύ των χρηστών αλλά και μεταξύ διαχειριστών και χρηστών.
- Εργαλεία συντήρησης  
Εργαλεία γενικής συντήρησης ενός site που έχει δημιουργηθεί με το TikiWiki.

### *PostNuke*

Είναι ένα πολύ δημοφιλές open-source εργαλείο και σύμφωνα με τους δημιουργούς του δεν αποτελεί απλώς ένα σύστημα διαχείρισης περιεχομένου, αλλά ένα ολοκληρωμένο σύστημα συνεργασίας και διαχείρισης περιεχομένου, βασισμένο στην λογική της ηλεκτρονικής κοινότητας (community). Του έχει αποδοθεί και ο τίτλος C3MS, από τις λέξεις Community, Content, Collaborative Management System. Ένα από τα ισχυρά σημεία του PostNuke είναι η αρκετά εκτεταμένη υποστήριξη που προσφέρει στα μέλη της κοινότητας, ειδικά διαμέσου του σχετικού forum <http://forums.postnuke.com/>. Η εγκατάσταση, παραμετροποίηση και διαχείριση του συστήματος για τις ανάγκες ενός site, intranet ή άλλης εφαρμογής που διαχειρίζεται περιεχόμενο είναι μια ιδιαίτερα απλή διαδικασία και δεν απαιτεί εξειδικευμένες γνώσεις. Όλες οι λειτουργίες εκτελούνται μέσα από περιβάλλον web browser.

## **4 Το σύστημα *COMMET***

Το Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων του ΕΜΠ έχει αναπτύξει και συντηρεί ένα ολοκληρωμένο σύστημα διαχείρισης περιεχομένου, το οποίο έχει σαν στόχο να συμβάλλει στην ανάπτυξη πολλαπλών εφαρμογών διαχείρισης περιεχομένου, όπως επιστημονικοί οδηγοί, κυβερνητικά και εκπαιδευτικά portals και άλλες εφαρμογές συνεργασίας και διαχείρισης γνώσης.

Το σύστημα αυτό ονομάζεται *COMMET* (Content Management Methodologies, Environments and Tools). Ακολουθεί ανάλυση των βασικότερων τεχνικών προδιαγραφών του συστήματος *COMMET*.

## ***4.1 Τύποι Περιεχομένου του COMMET***

Περιγράφεται ακολούθως με τυπικό τρόπο η οντολογία του COMMET. Με βάση την βασική αρχή του object orientation, που ακολουθεί παντού το περιβάλλον COMMET, κάθε τύπος περιεχομένου αντιστοιχεί σε μία κλάση της οντολογίας. Φυσικά, μπορούν να οριστούν νέοι τύποι περιεχομένου ή υπο-τύποι των υπαρχόντων. Στη συνέχεια θα χρησιμοποιούμε τον όρο κλάση αντικειμένου ως συνώνυμο του όρου «τύπος περιεχομένου».

Στο παρόν κεφάλαιο δίνεται το σχήμα των κλάσεων και για κάθε κλάση περιγράφονται τα ακόλουθα:

- Τα μετα-δεδομένα της κλάσης.
- Τις υποστηριζόμενες λειτουργίες παρουσίασης που σχετίζονται με αυτήν την κλάση.
- Παραδείγματα αντικειμένων της κλάσης αυτής.

Στο λογισμικό του COMMET, οι κλάσεις περιγράφονται με XML schemata και τα αντικείμενα κάθε κλάσης υλοποιούνται με XML αρχεία που ακολουθούν το XML schema της κλάσης. Στην περιγραφή που ακολουθεί, το XML schema κάθε κλάσης παριστάνεται με απεικόνιση grid, για να είναι πιο κατανοητό στους αναγνώστες.

### 4.1.1 Κλάσεις κόμβων

Όλες οι κλάσεις κόμβων έχουν τα εξής μετα-δεδομένα συστήματος:

Κωδικό Όνομα	Τίτλος	Είδος	Τύπος	Περιγραφή
nid	Κωδικός	protected	Integer	Μοναδικός Αναγνωριστικός Κωδικός του κόμβου
ntype	Κλάση	protected, final	String	Κλάση, που ανήκει ο κόμβος
nsubtype	Τύπος	protected	String	Υποτύπος της κλάσης, που ανήκει ο κόμβος

Φάκελος (folder)

Μετα-δεδομένα (Properties)

Κωδικό Όνομα	Τίτλος	public	Inherited	Τύπος	Περιγραφή
nname	Όνομα			String	Όνομα του φακέλου
ndescr	Περιγραφή			String	Περιγραφή του φακέλου
nsavepath	Save Path			String	Μονοπάτι αποθήκευσης για το περιεχόμενο (πόρους) που τοποθετείται στο Φάκελο
ncolor	Χρώμα			String	Χαρακτηριστικό χρώμα του φακέλου
ncreator	Δημιουργός			Integer	Χρήστης που δημιούργησε το Φάκελο (Αναγνωριστικός Κωδικός του χρήστη)
ndatecreated	Ημερομηνία Δημιουργίας			Date	Ημερομηνία δημιουργίας του φακέλου

Μέθοδοι (Methods)

Κωδικό Όνομα	Τίτλος	Είδος	Επιστρέφει	Περιγραφή
getAttributes		protected	Hashtable	Επιστρέφει δυναμικό πίνακα με ζεύγη μεταδεδομένο – τιμή του
getLink		public	String	Επιστρέφει σύνδεσμο προς σελίδα του Portal
getColor		public	String	Επιστρέφει το Αναγνωριστικό Χρώμα του Φακέλου
getSavePath		public	String	Επιστρέφει το Μονοπάτι αποθήκευσης περιεχομένου
getDescendants		protected	Vector	Επιστρέφει δυναμική λίστα με τους Αναγνωριστικούς Κωδικούς των υποφακέλων
printMyAttributes	Μεταδεδομένα Φακέλου	public	XML	Επιστρέφει τα μετα-δεδομένα του Φακέλου και σχετικές πληροφορίες
printMySubtreeNodes	Απόγονοι Φακέλου	public	XML	Επιστρέφει ολόκληρο το δέντρο με τους απογόνους του Φακέλου
-- printMyChildren	Υπο-φάκελοι Φακέλου	public	XML	Επιστρέφει μόνο τους Υπο-Φακέλους
print MySubtreeResources	Πόροι Φακέλου και Απογόνων	public	XML	Επιστρέφει τους πόρους ολόκληρου του Υποδέντρου
-- prinMyResources	Πόροι Φακέλου	public	XML	Επιστρέφει τους πόρους μόνο του Φακέλου
-- print MyChildrenResources	Πόροι Υπο-Φακέλων	public	XML	Επιστρέφει τους πόρους μόνο των Υπο-φακέλων του Φακέλου

Θεματική κατηγορία (category)  
Υποκλάση της κλάσης: Φάκελος

### Μετα-δεδομένα (Properties)

Κωδικό Όνομα	Τίτλος	public	Inherited	Τύπος	Περιγραφή
nname	Όνομα		√	String	Όνομα της Θεματικής Κατηγορίας
ndescr	Περιγραφή		√	String	Περιγραφή της Θεματικής Κατηγορίας
nsavepath	Save Path		√	String	Μονοπάτι αποθήκευσης για το περιεχόμενο (πόρους) που τοποθετείται στην Κατηγορία
ncolor	Χρώμα		√	String	Χαρακτηριστικό χρώμα της Κατηγορίας
ncreator	Δημιουργός		√	Integer	Χρήστης που δημιούργησε το Φάκελο (Αναγνωριστικός Κωδικός του χρήστη)
ncreated	Ημερομηνία Δημιουργίας		√	Date	Ημερομηνία δημιουργίας της Κατηγορίας

### Μέθοδοι (Methods)

Κληρονομεί όλες τις μεθόδους της κλάσης Φάκελος

### Πίνακας Ανακοινώσεων (blackboard)

Υποκλάση της κλάσης: Φάκελος

### Μετα-δεδομένα (Properties)

Κωδικό Όνομα	Τίτλος	public	Inherited	Τύπος	Περιγραφή
nname	Όνομα		√	String	Όνομα του Πίνακα Ανακοινώσεων
ndescr	Περιγραφή		√	String	Περιγραφή του Πίνακα Ανακοινώσεων
nsavepath	Save Path		√	String	Μονοπάτι αποθήκευσης για το περιεχόμενο (πόρους) που τοποθετείται στον Πίνακα
ncolor	Χρώμα		√	String	Χαρακτηριστικό χρώμα του Πίνακα
ncreator	Δημιουργός		√	Integer	Χρήστης που δημιούργησε το Φάκελο (Αναγνωριστικός Κωδικός του χρήστη)
ncreated	Ημερομηνία Δημιουργίας		√	Date	Ημερομηνία δημιουργίας του Πίνακα

### Μέθοδοι (Methods)

Κληρονομεί όλες τις μεθόδους της κλάσης Φάκελος

### Γεγονός (meeting)

Υποκλάση της κλάσης: Φάκελος

### Ιδιότητες (Properties)

Κωδικό Όνομα	Τίτλος	public	Inherited	Τύπος	Περιγραφή
nname	Όνομα		√	String	Όνομα
ndescr	Περιγραφή		√	String	Περιγραφή
nsavepath	Save Path		√	String	Μονοπάτι αποθήκευσης για το περιεχόμενο (πόρους) που τοποθετείται στο Γεγονός
ncolor	Χρώμα		√	String	Χαρακτηριστικό χρώμα του Γεγονότος
ndate	Ημερομηνία		-	Date	Ημερομηνία Διεξαγωγής του Γεγονότος
nstatus	Κατάσταση		-	String	Δηλώνει εάν το γεγονός έγινε ή εκκρεμεί
nhost	Τόπος		-	String	Τόπος Διεξαγωγής του γεγονότος
nrespons	Υπεύθυνος		-	String	Ατομο υπεύθυνο για το γεγονός
ncreator	Δημιουργός		√	Integer	Χρήστης που δημιούργησε το Φάκελο (Αναγνωριστικός Κωδικός του χρήστη)
ncreated	Ημερομηνία Δημιουργίας		√	Date	Ημερομηνία δημιουργίας του Γεγονότος

### Μέθοδοι (Methods)

Κληρονομεί όλες τις μεθόδους της κλάσης Φάκελος

## 4.1.2 Κλάσεις Πόρων

Όλες οι κλάσεις κόμβων έχουν τα εξής μετα-δεδομένα συστήματος:

Κωδικό Όνομα	Τίτλος	Είδος	Τύπος	Περιγραφή
rid	Κωδικός	protected	Integer	Μοναδικός Αναγνωριστικός Κωδικός του πόρου
rtype	Κλάση	protected, <i>final</i>	String	Κλάση, που ανήκει ο πόρος

### Πόρος (file)

Η κλάση file είναι abstract, δηλαδή δεν επιτρέπεται να δημιουργηθούν αντικείμενα αυτής της κλάσης, αλλά επιτρέπεται να δημιουργηθούν αντικείμενα των κλάσεων που κληρονομούν αυτή.

### Ιδιότητες (Properties)

Κωδικό Όνομα	Τίτλος	Τύπος Δεδομένου	Περιγραφή
rowner	Δημιουργός	Integer	Χρήστης που δημιούργησε τον πόρο (Αναγνωριστικός Κωδικός του χρήστη)
rdate	Ημ/νία Δημιουργίας	Date	Ημερομηνία Δημιουργίας του πόρου
rname	Τίτλος	String	Τίτλος του πόρου
rdescr	Περιγραφή	String	Σύντομη περιγραφή του πόρου
rpublished	Δημοσιεύσιμο	Boolean	Εάν ο πόρος είναι δημοσιεύσιμος
rposted	Ημερ Έκδοσης	Date	Ημερομηνία έκδοσης (για δημοσίευση)
rsmalltext	Αναλυτική Περιγραφή	String	Το HTML έγγραφο που περιέχει αναλυτική περιγραφή του πόρου

### Μέθοδοι (Methods)

Όνομα	Είδος	Επιστρέφει	Περιγραφή
getAttributes	protected	Hashtable	Επιστρέφει δυναμικό πίνακα με τα ζεύγη χαρακτηριστικό-τιμή
getLink	public	String	Επιστρέφει αλφαριθμητικό με τον σύνδεσμο προς σελίδα του Portal για το συγκεκριμένο αντικείμενο
getSpecificLink	public	String	Επιστρέφει αλφαριθμητικό που περιέχει το σύνδεσμο για την προβολή του αντικειμένου σε νέο παράθυρο σε εκτυπώσιμη μορφή
printMyAttributes	public	XML	Συγκεντρώνει σε XML δομή, τα μεταδεδομένα του πόρου, με κατάλληλες συνοδευτικές πληροφορίες
printMyHistory	public	XML	Συγκεντρώνει σε XML δομή, το ιστορικό διαχείρισης του πόρου (δημιουργία, τροποποίηση, προβολή του πόρου)
printMyRelatedFiles	public	XML	Συγκεντρώνει σε XML δομή, τους πόρους που έχουν συσχετιστεί με τον συγκεκριμένο πόρο
printMyParents	public	XML	Συγκεντρώνει σε XML δομή, τους φακέλους όπου έχει τοποθετηθεί ο συγκεκριμένος πόρος

### Εγγραφή Καταλόγου (catalogitem)

#### Ιδιότητες (Properties)

Κωδικό Όνομα	Τίτλος	Τύπος Δεδομένου	Περιγραφή
rowner	Δημιουργός	Integer	<inherited>
rdate	Ημ/νία Δημιουργίας	Date	<inherited>
rname	Τίτλος	String	<inherited>
rdescr	Περιγραφή	String	<inherited>
rpublished	Δημοσιεύσιμο	Boolean	<inherited>
rposted	Ημερ Έκδοσης	Date	<inherited>
rsmalltext	Αναλυτική Περιγραφή	String	<inherited>

#### Μέθοδοι (Methods)

Κληρονομεί όλες τις μεθόδους της κλάσης **πόρος**

### Εγγραφο (document)

#### Ιδιότητες (Properties)

Κωδικό Όνομα	Τίτλος	Τύπος Δεδομένου	Περιγραφή
<b>rowner</b>	Δημιουργός	<b>Integer</b>	<inherited>
<b>rdate</b>	Ημ/νία Δημιουργίας	<b>Date</b>	<inherited>
<b>rname</b>	Τίτλος	<b>String</b>	<inherited>
<b>rdescr</b>	Περιγραφή	<b>String</b>	<inherited>
<b>rpublished</b>	Δημοσιεύσιμο	<b>Boolean</b>	<inherited>
<b>rposted</b>	Ημερ Έκδοσης	<b>Date</b>	<inherited>
<b>rsmalltext</b>	Αναλυτική Περιγραφή	<b>String</b>	<inherited>
<b>rfile</b>	Έγγραφο για Download	<b>String</b>	Το αρχείο του εγγράφου προς download
<b>rauthor</b>	Συγγραφέας του εγγράφου	<b>String</b>	Συγγραφέας του εγγράφου
<b>rfile_type</b>	Τύπος του εγγράφου	<b>String</b>	Τύπος του download αρχείου
<b>rfile_size</b>	Μέγεθος του εγγράφου	<b>String</b>	Μέγεθος του download αρχείου (σε bytes)
<b>rfile_modified</b>	Ημ/νία τελευταίας μεταβολή	<b>String</b>	Ημερομηνία τελευταίας μεταβολής του download
<b>reventdate</b>	Ημερομηνία γεγονότος	<b>Date</b>	Ημερομηνία γεγονότος

### Μέθοδοι (Methods)

Κληρονομεί όλες τις μεθόδους της κλάσης **πόρος**

### Κείμενο (usertext)

#### Ιδιότητες (Properties)

Κωδικό Όνομα	Τίτλος	Τύπος Δεδομένου	Περιγραφή
<b>rowner</b>	Δημιουργός	<b>Integer</b>	<inherited>
<b>rdate</b>	Ημ/νία Δημιουργίας	<b>Date</b>	<inherited>
<b>rname</b>	Τίτλος	<b>String</b>	<inherited>
<b>rdescr</b>	Περιγραφή	<b>String</b>	<inherited>
<b>rpublished</b>	Δημοσιεύσιμο	<b>Boolean</b>	<inherited>
<b>rposted</b>	Ημερ Έκδοσης	<b>Date</b>	<inherited>
<b>rsmalltext</b>	Αναλυτική Περιγραφή	<b>String</b>	<inherited>
<b>rtxt</b>	Έγγραφο HTML	<b>String</b>	Το αρχείο του HTML εγγράφου
<b>rauthor</b>	Συγγραφέας του εγγράφου	<b>String</b>	Συγγραφέας
<b>reventdate</b>	Ημερομηνία γεγονότος	<b>Date</b>	Ημερομηνία γεγονότος

### Μέθοδοι (Methods)

Κληρονομεί όλες τις μεθόδους της κλάσης **πόρος**

### Σύνδεσμος (weblink)

#### Ιδιότητες (Properties)

Κωδικό Όνομα	Τίτλος	Τύπος Δεδομένου	Περιγραφή
<b>rowner</b>	Δημιουργός	<b>Integer</b>	<inherited>
<b>rdate</b>	Ημ/νία Δημιουργίας	<b>Date</b>	<inherited>
<b>rname</b>	Τίτλος	<b>String</b>	<inherited>
<b>rdescr</b>	Περιγραφή	<b>String</b>	<inherited>
<b>rpublished</b>	Δημοσιεύσιμο	<b>Boolean</b>	<inherited>
<b>rposted</b>	Ημερ Έκδοσης	<b>Date</b>	<inherited>
<b>rsmalltext</b>	Αναλυτική Περιγραφή	<b>String</b>	<inherited>
<b>rlink</b>	Σύνδεσμος (web link)	<b>String</b>	Το URL του συνδέσμου

### Μέθοδοι (Methods)

Κληρονομεί όλες τις μεθόδους της κλάσης **πόρος**

### Εσωτερικός Σύνδεσμος (internallink)

#### Ιδιότητες (Properties)

Κωδικό Όνομα	Τίτλος	Τύπος Δεδομένου	Περιγραφή
<b>rowner</b>	Δημιουργός	<b>Integer</b>	<inherited>
<b>rdate</b>	Ημ/νία Δημιουργίας	<b>Date</b>	<inherited>

<b>rname</b>	Τίτλος	String	<inherited>
<b>rdescr</b>	Περιγραφή	String	<inherited>
<b>rpublished</b>	Δημοσιεύσιμο	Boolean	<inherited>
<b>rposted</b>	Ημερ Έκδοσης	Date	<inherited>
<b>rsmalltext</b>	Αναλυτική Περιγραφή	String	<inherited>
<b>rlink</b>	Σύνδεσμος (εσωτερικό link)	String	<inherited>

Μέθοδοι (*Methods*)

Κληρονομεί όλες τις μεθόδους της κλάσης [σύνδεσμος](#)

Εικόνα (*image*)

### Ιδιότητες (Properties)

Κωδικό Όνομα	Τίτλος	Τύπος Δεδομένου	Περιγραφή
<b>rowner</b>	Δημιουργός	Integer	<inherited>
<b>rdate</b>	Ημ/νία Δημιουργίας	Date	<inherited>
<b>rname</b>	Τίτλος	String	<inherited>
<b>rdescr</b>	Περιγραφή	String	<inherited>
<b>rpublished</b>	Δημοσιεύσιμο	Boolean	<inherited>
<b>rposted</b>	Ημερ Έκδοσης	Date	<inherited>
<b>rsmalltext</b>	Αναλυτική Περιγραφή	String	<inherited>
<b>rimage</b>	Αρχείο Εικόνας	String	Αρχείο της εικόνας (gif, jpg, png, κλπ)
<b>rlink</b>	Link Εικόνας	String	Το URL του συνδέσμου από την εικόνα

Μέθοδοι (*Methods*)

Κληρονομεί όλες τις μεθόδους της κλάσης [πόρος](#)

Πρόσωπο (*person*)

### Ιδιότητες (Properties)

Κωδικό Όνομα	Τίτλος	Τύπος Δεδομένου	Περιγραφή
<b>rowner</b>	Δημιουργός	Integer	<inherited>
<b>rdate</b>	Ημ/νία Δημιουργίας	Date	<inherited>
<b>rname</b>	Όνοματεπώνυμο	String	Όνοματεπώνυμο του προσώπου
<b>rdescr</b>	Περιγραφή	String	<inherited>
<b>rpublished</b>	Δημοσιεύσιμο	Boolean	<inherited>
<b>rposted</b>	Ημερ Έκδοσης	Date	<inherited>
<b>rsmalltext</b>	Βιογραφικό	String	Αρχείο HTML με το Βιογραφικό του προσώπου

Μέθοδοι (*Methods*)

Κληρονομεί όλες τις μεθόδους της κλάσης [πόρος](#)

Ανακοίνωση (*announcement*)

### Ιδιότητες (Properties)

Κωδικό Όνομα	Τίτλος	Τύπος Δεδομένου	Περιγραφή
<b>rowner</b>	Δημιουργός	Integer	<inherited>
<b>rdate</b>	Ημ/νία Δημιουργίας	Date	<inherited>
<b>rname</b>	Τίτλος	String	<inherited>
<b>rdescr</b>	Περιγραφή	String	<inherited>
<b>rpublished</b>	Δημοσιεύσιμο	Boolean	<inherited>
<b>rposted</b>	Ημερ Έκδοσης	Date	<inherited>
<b>rsmalltext</b>	Αναλυτική Περιγραφή	String	<inherited>

Μέθοδοι (*Methods*)

Κληρονομεί όλες τις μεθόδους της κλάσης [πόρος](#)



### Ανακοίνωση - Έγγραφο (anndocument)

#### Ιδιότητες (Properties)

Κωδικό Όνομα	Τίτλος	Τύπος Δεδομένου	Περιγραφή
rowner	Δημιουργός	Integer	<inherited>
rdate	Ημ/νία Δημιουργίας	Date	<inherited>
rname	Τίτλος	String	<inherited>
rdescr	Περιγραφή	String	<inherited>
rpublished	Δημοσιεύσιμο	Boolean	<inherited>
rposted	Ημερ Έκδοσης	Date	<inherited>
rsmalltext	Αναλυτική Περιγραφή	String	<inherited>
rfile	Έγγραφο της ανακοίνωσης	String	<a href="#">Το αρχείο του εγγράφου της ανακοίνωσης</a>
rauthor	Συγγραφέας του εγγράφου	String	<a href="#">Συγγραφέας του εγγράφου</a>
rfile_type	Τύπος του εγγράφου	String	<a href="#">Τύπος του εγγράφου της ανακοίνωσης (doc, pdf)</a>
rfile_size	Μέγεθος του εγγράφου	String	<a href="#">Μέγεθος του εγγράφου της ανακοίνωσης (bytes)</a>
rfile_modified	Ημ/νία τελευταίας μεταβολής	String	<a href="#">Ημερομηνία τελευταίας μεταβολής του εγγράφου</a>
reventdate	Ημερομηνία γεγονότος	Date	<a href="#">Ημερομηνία γεγονότος</a>

#### Μέθοδοι (Methods)

Κληρονομεί όλες τις μεθόδους της κλάσης [ανακοίνωση](#)

### Ανακοίνωση - Κείμενο (annusertext)

#### Ιδιότητες (Properties)

Κωδικό Όνομα	Τίτλος	Τύπος Δεδομένου	Περιγραφή
rowner	Δημιουργός	Integer	<inherited>
rdate	Ημ/νία Δημιουργίας	Date	<inherited>
rname	Τίτλος	String	<inherited>
rdescr	Περιγραφή	String	<inherited>
rpublished	Δημοσιεύσιμο	Boolean	<inherited>
rposted	Ημερ Έκδοσης	Date	<inherited>
rsmalltext	Αναλυτική Περιγραφή	String	<inherited>
rtext	Έγγραφο HTML	String	<a href="#">Το αρχείο του HTML εγγράφου της ανακοίνωσης</a>
rauthor	Συγγραφέας του εγγράφου	String	<a href="#">Συγγραφέας</a>
reventdate	Ημερομηνία γεγονότος	Date	<a href="#">Ημερομηνία γεγονότος</a>

#### Μέθοδοι (Methods)

Κληρονομεί όλες τις μεθόδους της κλάσης [ανακοίνωση](#)

### Ανακοίνωση - Σύνδεσμος (annweblink)

#### Ιδιότητες (Properties)

Κωδικό Όνομα	Τίτλος	Τύπος Δεδομένου	Περιγραφή
rowner	Δημιουργός	Integer	<inherited>
rdate	Ημ/νία Δημιουργίας	Date	<inherited>
rname	Τίτλος	String	<inherited>
rdescr	Περιγραφή	String	<inherited>
rpublished	Δημοσιεύσιμο	Boolean	<inherited>
rposted	Ημερ Έκδοσης	Date	<inherited>
rsmalltext	Αναλυτική Περιγραφή	String	<inherited>
rlink	Σύνδεσμος (web link)	String	<a href="#">Το URL του συνδέσμου</a>

#### Μέθοδοι (Methods)

Κληρονομεί όλες τις μεθόδους της κλάσης [ανακοίνωση](#)

## 4.2 Λειτουργίες

Για κάθε λειτουργία περιγράφονται τα εξής στοιχεία:

- Τίτλος και σύντομη περιγραφή της λειτουργίας
- Παράμετροι κλήσης της λειτουργίας
- Το XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία.
- Ενδεικτικό παράδειγμα περιεχομένου που επιστρέφει η λειτουργία (σε XML μορφή)
- Ενδεικτικά παραδείγματα δομών εμφάνισης που παράγονται από τη λειτουργία (HTML format)

### 4.2.1 Λειτουργίες σε κλάσεις κόμβων

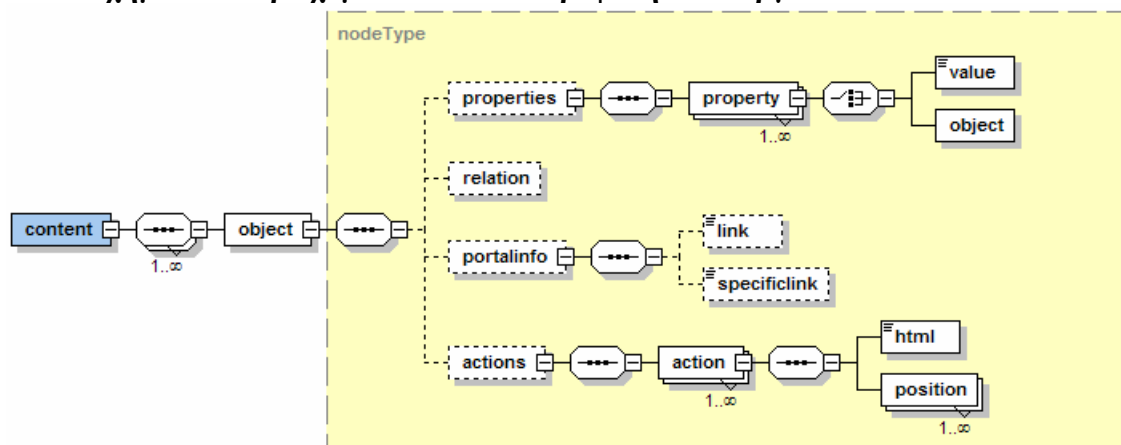
#### Περιγραφή Φακέλου

Η λειτουργία αυτή συγκεντρώνει και παρουσιάζει τα μετα-δεδομένα (τίτλο, περιγραφή, κλπ) ενός αντικειμένου της κλάσης Φάκελος.

#### Παράμετροι κλήσης της λειτουργίας

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Περιγραφή Παραμέτρου
lngFolderId	Κωδικός	Κωδικός του Φακέλου. - Default τιμή: <a href="#">Τρέχον Φάκελος (curmode)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να εφαρμοστεί σε άλλο φακέλο από τον τρέχοντα
lngLanguageId	Γλώσσα	Γλώσσα των προς εμφάνιση πληροφοριών - Default τιμή: <a href="#">Τρέχουσα Γλώσσα (curlang)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να δείξει το περιεχόμενό της σε γλώσσα διαφορετική από την τρέχουσα
lngUserId	Χρήστης	Χρήστης που προσπελαύνει τις πληροφορίες του Φακέλου - Default τιμή: <a href="#">Τρέχον Χρήστης (lngUserId)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη της τα δικαιώματα πρόσβασης διαφορετικού από τον τρέχοντα χρήστη
blnShowButtons	Παραπομπές	Αν η λειτουργία θα περιλάβει και τις απαραίτητες πληροφορίες για τις παραπομπές
blnIncludePresentationInfo	Τίτλοι	Αν η λειτουργία θα περιλάβει και τους τίτλους (labels) των μετα-δεδομένων

#### XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία



## Παράδειγμα περιεχομένου που επιστρέφει η λειτουργία

content																
object																
classification	node															
class	root															
visible	true															
properties																
property (9)																
name	value	presentation														
1 nid	4	presentation visible=false														
2 ntype	root	presentation visible=false														
3 nsubtype	main	presentation visible=false group=0 required=false														
4 nname	COMMET	<table border="1"> <tr> <td>visible</td> <td>true</td> </tr> <tr> <td>order</td> <td>1</td> </tr> <tr> <td>group</td> <td>0</td> </tr> <tr> <td colspan="2">description (2)</td> </tr> <tr> <td>language</td> <td>Abc Text</td> </tr> <tr> <td>1 0</td> <td>Τίτλος</td> </tr> <tr> <td>2 1</td> <td>Title</td> </tr> </table>	visible	true	order	1	group	0	description (2)		language	Abc Text	1 0	Τίτλος	2 1	Title
visible	true															
order	1															
group	0															
description (2)																
language	Abc Text															
1 0	Τίτλος															
2 1	Title															
5 ndescr	Χώρος συνεργασίας της ομάδας ανάπτυξης του COMMET	<table border="1"> <tr> <td>visible</td> <td>true</td> </tr> <tr> <td>order</td> <td>3</td> </tr> <tr> <td>group</td> <td>0</td> </tr> <tr> <td colspan="2">description (2)</td> </tr> <tr> <td>language</td> <td>Abc Text</td> </tr> <tr> <td>1 0</td> <td>Περιγραφή</td> </tr> <tr> <td>2 1</td> <td>Description</td> </tr> </table>	visible	true	order	3	group	0	description (2)		language	Abc Text	1 0	Περιγραφή	2 1	Description
visible	true															
order	3															
group	0															
description (2)																
language	Abc Text															
1 0	Περιγραφή															
2 1	Description															
6 nsavepath	main/	presentation visible=false group=0 required=false														
7 ncolor	#FF0033	presentation visible=false														
8 ncreator		presentation visible=false order=2 group=0 required=false														
9 ndatecreated		presentation visible=false order=4 group=0 required=false														
relation	level=0															
portalinfo																
link																
CDData	/core/portal.asp?cpage=NODE&cnode=4&clang=0&cuser=5B1A2680-1406-4B1A-82E6-66AAB517030C															

## Παράδειγματα εμφάνισης του περιεχομένου που επιστρέφει η λειτουργία

<b>COMMET</b>	
▶ Τίτλος	: <b>COMMET</b>
▶ Περιγραφή	: Χώρος συνεργασίας της ομάδας ανάπτυξης του COMMET

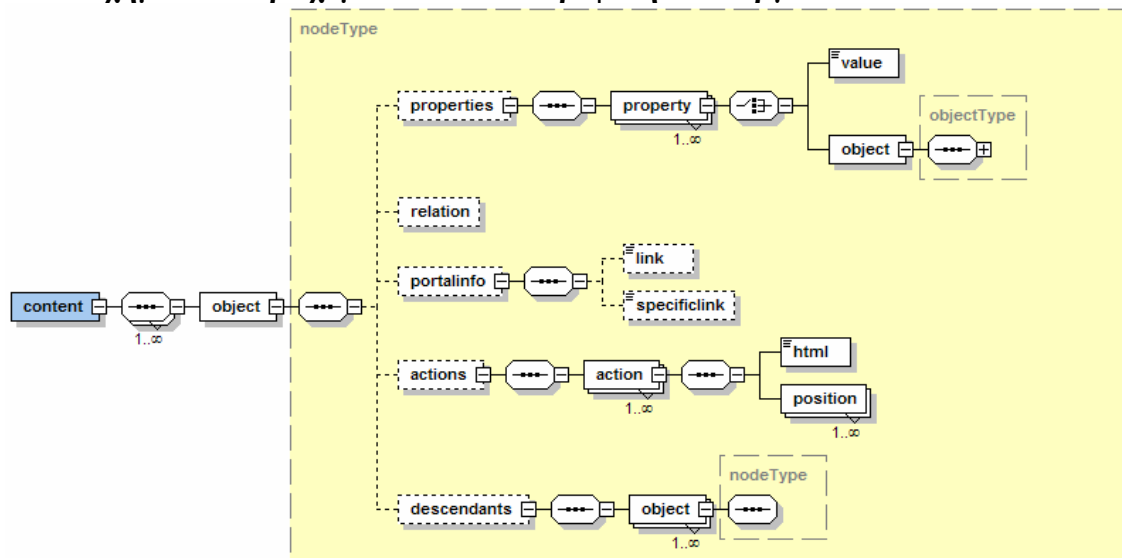
## Απόγονοι Φακέλου

Η λειτουργία «Απόγονοι Φακέλου» συγκεντρώνει και παρουσιάζει τους απογόνους ενός φακέλου. Κάθε φάκελος ενός Portal που βασίζεται στο COMMET ανήκει σε μία ιεραρχική (δενδρική) δομή φακέλων. Κατά συνέπεια, η εφαρμογή της λειτουργίας αυτής σε ένα φάκελο ενός Portal, επιστρέφει ολόκληρο το υποδέντρο της ιεραρχικής δομής του Portal που έχει ρίζα τον φάκελο αυτό

## Παράμετροι κλήσης της λειτουργίας

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Περιγραφή Παραμέτρου
lngFolderId	Κωδικός	Κωδικός του Φακέλου. - Default τιμή: <a href="#">Τρέχον Φάκελος</a> (curmode) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να εφαρμοστεί σε άλλο φάκελο από τον τρέχοντα.
lngLanguageId	Γλώσσα	Γλώσσα των προς εμφάνιση πληροφοριών - Default τιμή: <a href="#">Τρέχουσα Γλώσσα</a> (curlang) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να δείξει το περιεχόμενο της σε γλώσσα διαφορετική από την τρέχουσα
lngUserId	Χρήστης	Χρήστης που προσπελαύνει τις πληροφορίες του Φακέλου - Default τιμή: <a href="#">Τρέχον Χρήστης</a> (lngUserId) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη της τα δικαιώματα πρόσβασης διαφορετικού από τον τρέχοντα χρήστη
blnShowButtons	Παραπομπές	Αν η λειτουργία θα περιλάβει και τις απαραίτητες πληροφορίες για τις παραπομπές
blnIncludePresentationInfo	Τίτλοι	Αν η λειτουργία θα περιλάβει και τους τίτλους (labels) των μετα-δεδομένων
arrAllowedTypes	Συμπεριλαμβανόμενοι τύποι Φακέλων	Κλάσεις φακέλων (σύνολο ζευγών <u>κλάση/τύπος</u> ) που θα περιληφθούν στους απογόνους. Οι απόγονοι που δεν ανήκουν στις κλάσεις αυτές δεν θα περιληφθούν στα αποτελέσματα της λειτουργίας.
arrHiddenTypes	Εξαιρούμενοι τύποι Φάκελοι	Κλάσεις φακέλων (σύνολο ζευγών <u>κλάση/τύπος</u> ) που θα εξαιρεθούν από τους απογόνους. Η παράμετρος αυτή λαμβάνεται υπόψη, μόνο εάν η παράμετρος <a href="#">arrAllowedTypes</a> είναι κενή.
lngDepth	Βάθος απογόνων	Το βάθος μέχρι το οποίο θα εντοπιστούν απόγονοι. Αν έχει τιμή 1 τότε θα συμπεριληφθούν μόνο τα παιδιά του φακέλου, αν έχει τιμή 2 τότε θα συμπεριληφθούν τα «παιδιά» και τα «εγγόνια» του φακέλου, κ.ο.κ.

## XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία



## Παράδειγμα περιεχομένου που επιστρέφει η λειτουργία

content

- object
  - classification node
  - class team
  - visible false
  - descendants
    - object
      - classification node
      - class folder
      - visible true
      - properties
        - property (8)
 

#	name	sort	value	object
1	nid		2460	
2	ntype		folder	
3	nsubtype		themeContainer	
4	nname		Θέματα Εργασίας	
5	ncreator			object
6	nsavepath		main/	
7	ncolor		#6633FF	
8	ndatecreated	20041207	7/12/2004	
      - relation level=1
      - actions
      - descendants
        - object
          - classification node
          - class theme
          - visible true
          - properties
            - property (8)
 


#	name	sort	value	object
1	nid		2591	
2	ntype		theme	
3	nname		XSL Configuration	
4	ndescr		Κατασκευή εργαλείου για την διαχείριση των XSL Templates μας εφαρμογής.	
5	ncreator			object classificati...
6	nsavepath		main/	
7	ncolor		#66FF66	
8	ndatecreated	20041215	15/12/2004	
          - relation level=2
          - actions
  - object
    - classification node
    - class folder
    - visible true
    - properties
      - property name=nid
      - property
        - name ntype
        - value folder
      - property name=nsubtype
      - property
        - name nname
        - value Ειδικές Συναντήσεις
      - property name=ndescr
      - property name=ncreator
      - property name=nsavepath
      - property name=ncolor
      - property name=ndatecreated sort=20041207
    - relation level=1
    - actions
    - descendants
      - object
        - classification node
        - class meeting
        - visible true
        - properties
          - property (12)
 

#	name	sort	value	object
1	nid		2621	
2	ntype		meeting	
3	nname		Διαχείριση αρχείου Φακέλων	
4	ndescr		Συζήτηση για τη διαχείριση του αρχείου ενός φακέλου σε περίπτωση που οι πόροι στο αρχείο είναι πολλοί	
5	ncreator			object classificati...
6	nsavepath		main/	
7	ncolor		#6633FF	
8	ndatecreated	20050202	2/2/2005	
9	ndate	20041222	22/12/2004	
10	nstatus		Εγινε	
11	nhost		Γραφείο	
12	nrespons		Κυριάκος Διακονικαίου	
        - relation level=2
        - actions
    - object
      - classification node
      - class meeting
      - visible true
      - properties
        - property (12)
 



#	name	sort	value	object
1	nid		2625	
2	ntype		meeting	
3	nname		Προβολή πόρων φακέλου σε πολλαπλές σελίδες	
4	ndescr		Όταν ένας φάκελος περιέχει πολλούς πόρους είναι επιθυμητό να εμφανίζονται χωρισμένοι σε σελίδες	
5	ncreator			object classificati...
6	nsavepath		main/	
7	ncolor		#6633FF	
8	ndatecreated	20050202	2/2/2005	
9	ndate	20050111	11/1/2005	
10	nstatus		Εγινε	
11	nhost		Γραφείο	
12	nrespons		Αλέξανδρος Λιάσκης	
      - relation level=2
      - actions

## Παραδείγματα εμφάνισης του περιεχομένου που επιστρέφει η λειτουργία

### Θέματα Εργασίας

 XSL Configuration	Κατασκευή εργαλείου για την διαχείριση των XSL Templates μιας εφαρμογής.
---	--

### Ειδικές Συναντήσεις

 22/12/2004 Έγινε	<b>Διαχείριση αρχείου Φακέλων</b>	Συζήτηση για τη διαχείριση του αρχείου ενός φακέλου σε περίπτωση που οι πόροι στο αρχείο είναι πολλοί
 11/1/2005 Έγινε	<b>Προβολή πόρων φακέλου σε πολλαπλές σελίδες</b>	Όταν ένας φάκελος περιέχει πολλούς πόρους είναι επιθυμητό να εμφανίζονται χωρισμένοι σε σελίδες

## Υπο-φάκελοι Φακέλου

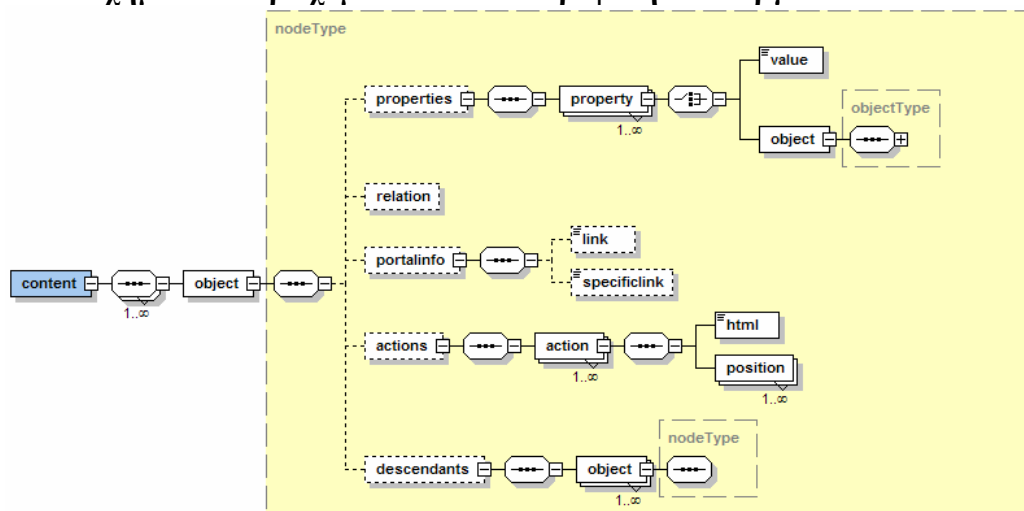
Η λειτουργία «Υπο-Φάκελοι Φακέλου» συγκεντρώνει και παρουσιάζει τους υποφακέλους ενός φακέλου.

Αποτελεί μία εξειδίκευση της λειτουργίας «Απόγονοι Φακέλου», η οποία περιορίζει την αναζήτηση των απογόνων μόνο στα παιδιά ενός Φακέλου (βάθος απογόνων 1).

### Παράμετροι κλήσης της λειτουργίας

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Περιγραφή Παραμέτρου
lngFolderId	Κωδικός	Κωδικός του Φακέλου. - Default τιμή: <a href="#">Τρέχον Φάκελος</a> (curnode) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να εφαρμοστεί σε άλλο φακέλο από τον τρέχοντα.
lngLanguageId	Γλώσσα	Γλώσσα των προς εμφάνιση πληροφοριών - Default τιμή: <a href="#">Τρέχουσα Γλώσσα</a> (curlang) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να δείξει το περιεχόμενο της σε γλώσσα διαφορετική από την τρέχουσα
lngUserId	Χρήστης	Χρήστης που προσπελάει τις πληροφορίες του Φακέλου - Default τιμή: <a href="#">Τρέχον Χρήστης</a> (lngUserId) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη της τα δικαιώματα πρόσβασης διαφορετικού από τον τρέχοντα χρήστη
blnShowButtons	Παραπομπές	Αν η λειτουργία θα περιλάβει και τις απαραίτητες πληροφορίες για τις παραπομπές
blnIncludePresentationInfo	Τίτλοι	Αν η λειτουργία θα περιλάβει και τους τίτλους (labels) των μετα-δεδομένων
arrAllowedTypes	Συμπεριλαμβανόμενοι τύποι Φακέλων	Κλάσεις φακέλων (σύνολο ζευγών <u>κλάση/τύπος</u> ) που θα περιληφθούν στους απογόνους. Οι απόγονοι που δεν ανήκουν στις κλάσεις αυτές δεν θα περιληφθούν στα αποτελέσματα της λειτουργίας.
arrHiddenTypes	Εξαιρούμενοι τύποι Φάκελοι	Κλάσεις φακέλων (σύνολο ζευγών <u>κλάση/τύπος</u> ) που θα εξαιρεθούν από τους απογόνους. Η παράμετρος αυτή λαμβάνεται υπόψη, μόνο εάν η παράμετρος <a href="#">arrAllowedTypes</a> είναι κενή.

### XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία





## Παράδειγμα περιεχομένου που επιστρέφει η λειτουργία

The screenshot displays a content management system interface. On the left, a tree view shows the hierarchy of content. The main area shows a detailed view of a selected node (nid=2442). The node is of class 'node' and type 'team'. It has several properties, including 'name' (Αντικείμενα Εργασιών), 'ndescr' (Λογισμικό της εφαρμογής Δικτυακής Πύλης & της εφαρμογής Διαχείρισης Περιεχομένου σε περιβάλλον MS ASP), 'ncolor' (#FF3333), and 'ndatecreated' (7/12/2004). The interface also shows relations and actions for the node.

id	class	class	v	properties	relation	actions																																										
1	node	team	true	<table border="1"> <thead> <tr> <th>name</th> <th>sort</th> <th>value</th> <th>object</th> </tr> </thead> <tbody> <tr> <td>1 nid</td> <td></td> <td>2442</td> <td></td> </tr> <tr> <td>2 ntype</td> <td></td> <td>team</td> <td></td> </tr> <tr> <td>3 nname</td> <td></td> <td>COMMET ASP</td> <td></td> </tr> <tr> <td>4 ndescr</td> <td></td> <td>Λογισμικό της εφαρμογής Δικτυακής Πύλης &amp; της εφαρμογής Διαχείρισης Περιεχομένου σε περιβάλλον MS ASP</td> <td></td> </tr> <tr> <td>5 ncreator</td> <td></td> <td></td> <td>object</td> </tr> <tr> <td>6 nsavepath</td> <td></td> <td>main/</td> <td></td> </tr> <tr> <td>7 ncolor</td> <td></td> <td>#FF3333</td> <td></td> </tr> <tr> <td>8 ndatecreated</td> <td>20041207</td> <td>7/12/2004</td> <td></td> </tr> </tbody> </table>	name	sort	value	object	1 nid		2442		2 ntype		team		3 nname		COMMET ASP		4 ndescr		Λογισμικό της εφαρμογής Δικτυακής Πύλης & της εφαρμογής Διαχείρισης Περιεχομένου σε περιβάλλον MS ASP		5 ncreator			object	6 nsavepath		main/		7 ncolor		#FF3333		8 ndatecreated	20041207	7/12/2004		relation level=1	<table border="1"> <thead> <tr> <th>name</th> <th>html</th> <th>position</th> </tr> </thead> <tbody> <tr> <td>1 cmn_subtree</td> <td>html</td> <td>position</td> </tr> </tbody> </table>	name	html	position	1 cmn_subtree	html	position
name	sort	value	object																																													
1 nid		2442																																														
2 ntype		team																																														
3 nname		COMMET ASP																																														
4 ndescr		Λογισμικό της εφαρμογής Δικτυακής Πύλης & της εφαρμογής Διαχείρισης Περιεχομένου σε περιβάλλον MS ASP																																														
5 ncreator			object																																													
6 nsavepath		main/																																														
7 ncolor		#FF3333																																														
8 ndatecreated	20041207	7/12/2004																																														
name	html	position																																														
1 cmn_subtree	html	position																																														
2	node	team	true	properties	relation level=1	actions																																										
3	node	team	true	properties	relation level=1	actions																																										
4	node	team	true	properties	relation level=1	actions																																										
5	node	team	true	properties	relation level=1	actions																																										

## Παραδείγματα εμφάνισης του περιεχομένου που επιστρέφει η λειτουργία

Αντικείμενα Εργασιών		
	COMMET ASP	Λογισμικό της εφαρμογής Δικτυακής Πύλης & της εφαρμογής Διαχείρισης Περιεχομένου σε περιβάλλον MS ASP
	COMMET NET	Λογισμικό της εφαρμογής Δικτυακής Πύλης & της εφαρμογής Διαχείρισης Περιεχομένου σε περιβάλλον MS .NET
	COMMET Tools	Τα εργαλεία διαχείρισης του COMMET (διαχείριση οντολογίας, διαχείριση παραπομπών, κλπ). Χρησιμοποιούνται και από την ASP και από την .NET έκδοση της Δικτυακής Πύλης. Έχουν αναπτυχθεί σε περιβάλλον MS .NET
	COMMET Access	Τα εργαλεία διαχείρισης των χρηστών και εξουσιοδότησης της πρόσβασης στο περιεχόμενο. Χρησιμοποιούνται τόσο από την ASP όσο και από τη .NET έκδοση της Δικτυακής Πύλης
	Installations	Αρχείο για τις εφαρμογές που έχουν αναπτυχθεί με το COMMET. Status κάθε εφαρμογής. Link με το αντίστοιχο έργο-συμβόλαιο.

## Πόροι Φακέλου & Απογόνων

Η λειτουργία «Πόροι Φακέλου & Απογόνων» συγκεντρώνει και παρουσιάζει τους πόρους που έχουν τοποθετηθεί σε ένα φάκελο και στους απογόνους του.

Η εφαρμογή της λειτουργίας αυτής σε ένα φάκελο ενός Portal:

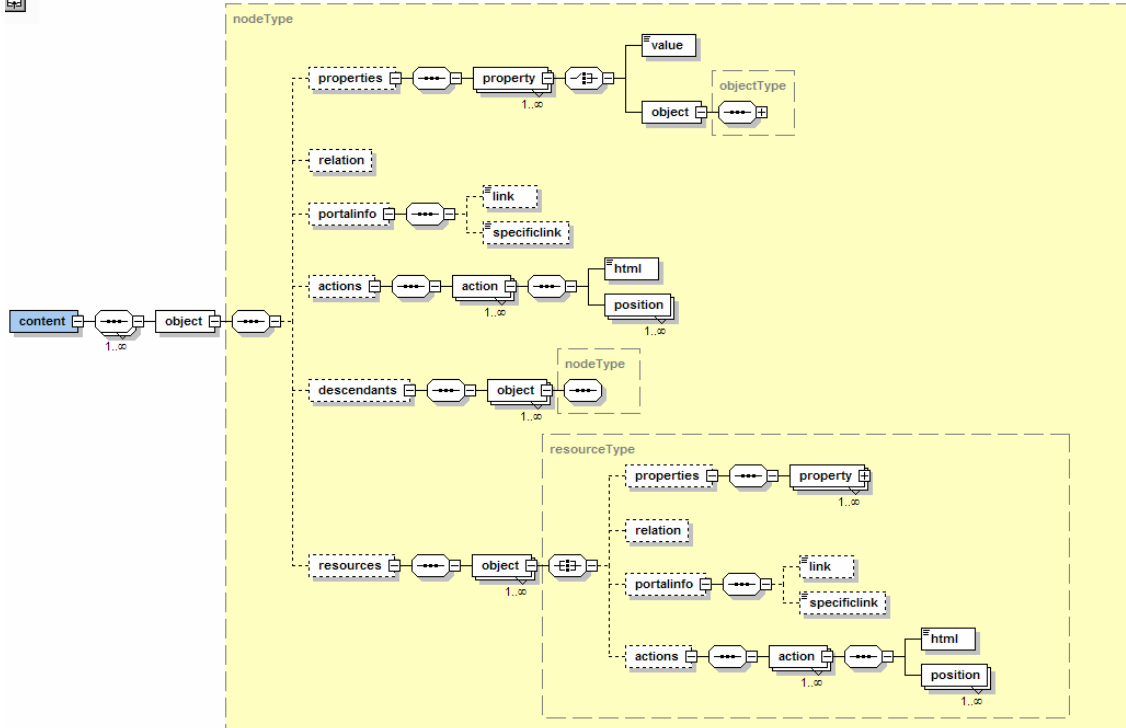
Επιστρέφει ολόκληρο το υποδέντρο της ιεραρχικής δομής του Portal που έχει ρίζα τον φάκελο.

Σε κάθε φάκελο του υποδέντρου, περιλαμβάνει τους πόρους που έχουν τοποθετηθεί σε αυτόν.

## Παράμετροι κλήσης της λειτουργίας

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Περιγραφή Παραμέτρου
lngFolderId	Κωδικός	Κωδικός του Φακέλου. - Default τιμή: <a href="#">Τρέχον Φάκελος</a> (curmode) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να εφαρμοστεί σε άλλο φάκελο από τον τρέχοντα.
lngLanguageId	Γλώσσα	Γλώσσα των προς εμφάνιση πληροφοριών - Default τιμή: <a href="#">Τρέχουσα Γλώσσα</a> (curlang) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να δείξει το περιεχόμενο της σε γλώσσα διαφορετική από την τρέχουσα.
lngUserId	Χρήστης	Χρήστης που προσπελαύνει τις πληροφορίες του Φακέλου - Default τιμή: <a href="#">Τρέχον Χρήστης</a> (lngUserId) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη της τα δικαιώματα πρόσβασης διαφορετικού από τον τρέχοντα χρήστη.
blnShowButtons	Παραπομπές	Αν η λειτουργία θα περιλάβει και τις απαραίτητες πληροφορίες για τις παραπομπές.
blnIncludePresentationInfo	Τίτλοι	Αν η λειτουργία θα περιλάβει και τους τίτλους (labels) των μετα-δεδομένων.
arrAllowedTypes	Συμπεριλαμβανόμενοι τύποι Φακέλων	Κλάσεις φακέλων (σύνολο ζευγών <a href="#">κλάση/τύπος</a> ) που θα περιληφθούν στους απογόνους. Οι απόγονοι που δεν ανήκουν στις κλάσεις αυτές δεν θα περιληφθούν στα αποτελέσματα της λειτουργίας.
arrHiddenTypes	Εξαιρούμενοι τύποι Φάκελοι	Κλάσεις φακέλων (σύνολο ζευγών <a href="#">κλάση/τύπος</a> ) που θα εξαιρεθούν από τους απογόνους. Η παράμετρος αυτή λαμβάνεται υπόψη, μόνο εάν η παράμετρος <a href="#">arrAllowedTypes</a> είναι κενή.
lngDepth	Βάθος απογόνων	Το βάθος μέχρι το οποίο θα εντοπιστούν απόγονοι. Αν έχει τιμή 1 τότε θα συμπεριληφθούν μόνο τα παιδιά του φακέλου, αν έχει τιμή 2 τότε θα συμπεριληφθούν τα «παιδιά» και τα «εγγόνια» του φακέλου, κ.ο.κ.
strState	Πλάνο εμφάνισης	Περιορίζει τους πόρους που θα παρουσιαστούν, με βάση το status της τοποθέτησης (δηλ. εάν έχει δηλωθεί ότι ένας πόρος θα ανήκει στο αρχείο του φακέλου, κοκ). Παίρνει τιμές: *, <b>main</b> , <b>archive</b> , <b>personal</b> Αν έχει τιμή "*" τότε θα εμφανιστούν όλοι οι πόροι του φακέλου, ειδάλλως θα εμφανιστούν μόνο οι πόροι με την επιλεγμένη κατάσταση.
strPresentationOverride	Υποχρεωτικός Τρόπος Παρουσίασης	Καθορίζει τον τρόπο εντοπισμού των χαρακτηριστικών παρουσίασης ενός πόρου στα πλαίσια του Φακέλου όπου έχει τοποθετηθεί. Εάν έχει τιμή, τότε η τιμή αυτή είναι ο επιβαλλόμενος τρόπος παρουσίασης για όλους τους πόρους του Φακέλου. Αλλιώς, κάθε πόρος θα εμφανιστεί με τον ξεχωριστό τρόπο που έχει επιλεγεί κατά την τοποθέτησή του στο Φάκελο.

## XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία




## Παράδειγμα περιεχομένου που επιστρέφει η λειτουργία


content									
object									
classification	node								
class	folder								
visible	true								
properties									
property	name=nid								
property	name=nstype								
property	name=nsubtype								
property									
name	nname								
value	MS ASP, .NET								
property	name=ndescr								
property	name=ncreator								
property	name=nsavepath								
property	name=ncolor								
relation	level=0								
portalinfo									
actions									
resources									
object (1)									
classification	resource	class	document	properties	relation	portalinfo	actions		
1				properties	relation	portalinfo	actions		
descendants									
object (2)									
classification	node	class	folder	visible	properties	relation	portalinfo	actions	resources
1				properties	relation	portalinfo	actions	resources	object (2)
properties									
property	name=nid								
property	name=nstype								
property									
name	nname								
value	Ερευνες και Γενικό Πληροφοριακό Υλικό								
property	name=ncreator								
property	name=nsavepath								
property	name=ncolor								
relation									
portalinfo									
actions									
resources	object (2)								
2				properties	relation	portalinfo	actions	resources	object (2)
properties									
property	name=nid								
property	name=nstype								
property									
name	nname								
value	Σχετικά Links								
property	name=ncreator								
property	name=nsavepath								
property	name=ncolor								

## Παραδείγματα εμφάνισης του περιεχομένου που επιστρέφει η λειτουργία

### MS ASP, .NET

 Περιγραφικό κείμενο	Σύντομη περιγραφή για τις τεχνολογίες ASP και ASP.NET
---	---

### Έρευνες και Γενικό Πληροφοριακό Υλικό

 Τεχνολογικές Κατευθύνσεις της Microsoft	Κείμενο εργασίας που περιγράφει τις τεχνολογίες προς τις οποίες θα κινηθεί η Microsoft τα επόμενα έτη
 XML και Microsoft	Περιγράφει την ενσωμάτωση της "XML" σε όλα τα νέα περιβάλλοντα της Microsoft

### Σχετικά Links

 Microsoft	Σύνδεσμος προς το site της Microsoft
 Λίστα σχετικών συνδέσμων	

## Πόροι Φακέλου

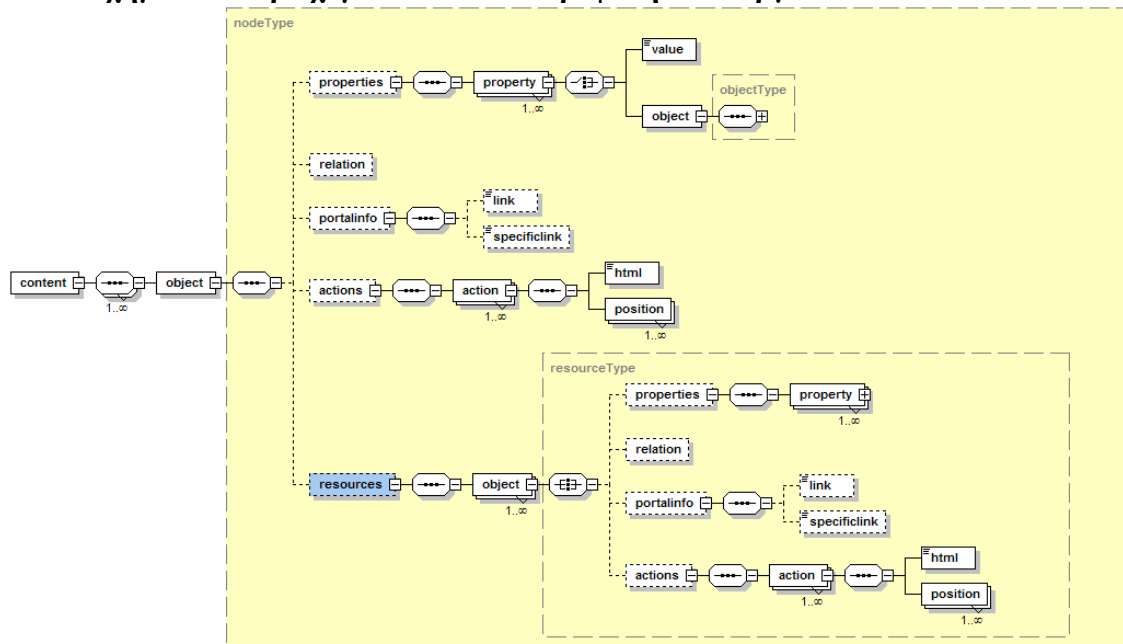
Η λειτουργία «Πόροι Φακέλου» συγκεντρώνει και παρουσιάζει τους πόρους που έχουν τοποθετηθεί σε ένα φάκελο.

Η λειτουργία αυτή είναι μία εξειδίκευση της λειτουργίας «Πόροι Φακέλου & Απογόνων», η οποία περιορίζει την αναζήτηση μόνο στους πόρους του Φακέλου.

## Παράμετροι κλήσης της λειτουργίας

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Περιγραφή Παραμέτρου
lngFolderId	Κωδικός	Κωδικός του Φακέλου. - Default τιμή: <a href="#">Τρέχον Φάκελος</a> (curmode) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να εφαρμοστεί σε άλλο φακέλο από τον τρέχοντα.
lngLanguageId	Γλώσσα	Γλώσσα των προς εμφάνιση πληροφοριών - Default τιμή: <a href="#">Τρέχουσα Γλώσσα</a> (curlang) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να δείξει το περιεχόμενο της σε γλώσσα διαφορετική από την τρέχουσα
lngUserId	Χρήστης	Χρήστης που προσπελαύνει τις πληροφορίες του Φακέλου - Default τιμή: <a href="#">Τρέχον Χρήστης</a> (lngUserId) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη της τα δικαιώματα πρόσβασης διαφορετικού από τον τρέχοντα χρήστη
blnShowButtons	Παραπομπές	Αν η λειτουργία θα περιλάβει και τις απαραίτητες πληροφορίες για τις παραπομπές
blnIncludePresentationInfo	Τίτλοι	Αν η λειτουργία θα περιλάβει και τους τίτλους (labels) των μετα-δεδομένων
strState	Πλάνο εμφάνισης	Περιορίζει τους πόρους που θα παρουσιαστούν, με βάση το status της τοποθέτησης (δηλ. εάν έχει δηλωθεί ότι ένας πόρος θα ανήκει στο αρχείο του φακέλου, κοκ). Παίρνει τιμές: *, main , archive , personal Αν έχει τιμή "*" τότε θα εμφανιστούν όλοι οι πόροι του φακέλου, ειδάλως θα εμφανιστούν μόνο οι πόροι με την επιλεγμένη κατάσταση
strPresentationOverride	Υποχρεωτικός Τρόπος Παρουσίασης	Καθορίζει τον τρόπο εντοπισμού των χαρακτηριστικών παρουσίασης ενός πόρου στα πλαίσια του Φακέλου όπου έχει τοποθετηθεί. Εάν έχει τιμή, τότε η τιμή αυτή είναι ο επιβαλλόμενος τρόπος παρουσίασης για όλους τους πόρους του Φακέλου. Αλλιώς, κάθε πόρος θα εμφανιστεί με τον ξεχωριστό τρόπο που έχει επιλεγεί κατά την τοποθέτησή του στο Φάκελο,

## XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία



## Παράδειγμα περιεχομένου που επιστρέφει η λειτουργία

content

object

- classification: node
- class: blackboard
- visible: true

properties

#	name	sort	value	object
1	nid		1970	
2	rntype		blackboard	
3	rname		Γενικές Ανακοινώσεις	
4	ncreator			object cla...
5	nsavepath		main/	
6	ncolor		#33FFCC	
7	ndatecreated	20040614	14/6/2004	

relation level=0

portalinfo

actions

action

- name: cmn\_showarch
- html
- position (1)
 

#	group	presentation	subsche...
1	100	header-resources	NodeResources
- action name=cmn\_path

resources

object

- classification: resource
- class: anndocument

properties

#	name	sort	value	object
1	rid		845	
2	rntype		anndocument	
3	rpublished		-1	
4	rname		Οδηγός Χρήσης της Δικτυακής Πύλης	
5	rowner			object classificati...
6	rauthor		Υπεύθυνος Εφαρμογής	
7	rsmaltext		/appdata/main/htmls/f_243.html	
8	rsmaltext_html		value	
9	rfile_type		zip	
10	rfile		/appdata/main/docs/commet-Committees-USER GUIDE-Dec-03.zip	
11	rfile_size		1794009	
12	rdate	20031219	19/12/2003	
13	rposted	20031219	19/12/2003	
14	reventdate	20040407	7/4/2004	
15	rdescr		Ο Οδηγός Χρήσης της παρούσης Δικτυακής Πύλης σε ZIP αρχείο	

relation state=main order=9 visibility=-1

portalinfo

- link
- specificlink

actions

- object classification=resource class=annusertext
- object classification=resource class=annusertext
- object classification=resource class=annusertext

## Παράδειγμα εμφάνισης του περιεχομένου που επιστρέφει η λειτουργία

### Γενικές Ανακοινώσεις

		<b>22/12/2004</b>	<b>Άρχισε η πιλοτική χρήση του site για συνεργασία μεταξύ ημών</b>	Αλέξανδρος Λάσκαρης	
		<b>19/12/2003</b>	<b>Οδηγός Χρήσης της Δικτυακής Πύλης</b>	Υπεύθυνος Εφαρμογής	
		<b>18/12/2003</b>	<b>Γενική Περιγραφή της Δικτυακής Πύλης</b>	Υπεύθυνος Εφαρμογής	
		<b>8/6/2003</b>	<b>Επικοινωνία με τους υπεύθυνους του Portal</b>	Υπεύθυνος Εφαρμογής	

**Δείτε το αρχείο του φακέλου >>>**

## Πόροι Υπο-Φακέλων

Η λειτουργία «Πόροι Υπο-Φακέλων» συγκεντρώνει και παρουσιάζει τους πόρους που έχουν τοποθετηθεί στους υποφακέλους ενός φακέλου και όχι στον ίδιο το φάκελο.

Η λειτουργία αυτή είναι μία ειδική περίπτωση της λειτουργίας «Πόροι Φακέλου & Απογόνων», η οποία περιορίζει την αναζήτηση πόρων μόνο στους Υπο-Φακέλους του Φακέλου, εξαιρώντας τους πόρους του ίδιου του Φακέλου.

## Παράμετροι κλήσης της λειτουργίας

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Περιγραφή Παραμέτρου
lngFolderId	Κωδικός	Κωδικός του Φακέλου. - Default τιμή: <a href="#">Τρέχον Φάκελος</a> (curmode) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να εφαρμοστεί σε άλλο φακέλο από τον τρέχοντα.
lngLanguageId	Γλώσσα	Γλώσσα των προς εμφάνιση πληροφοριών - Default τιμή: <a href="#">Τρέχουσα Γλώσσα</a> (curlang) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να δείξει το περιεχόμενο της σε γλώσσα διαφορετική από την τρέχουσα.
lngUserId	Χρήστης	Χρήστης που προσπελάει τις πληροφορίες του Φακέλου - Default τιμή: <a href="#">Τρέχον Χρήστης</a> (lngUserId) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη της τα δικαιώματα πρόσβασης διαφορετικού από τον τρέχοντα χρήστη.
blnShowButtons	Παραπομπές	Αν η λειτουργία θα περιλάβει και τις απαραίτητες πληροφορίες για τις παραπομπές.
blnIncludePresentationInfo	Τίτλοι	Αν η λειτουργία θα περιλάβει και τους τίτλους (labels) των μετα-δεδομένων.
arrAllowedTypes	Συμπεριλαμβανόμενοι τύποι Φακέλων	Κλάσεις φακέλων (σύνολο ζευγών <a href="#">κλάση/τύπος</a> ) που θα περιληφθούν στους απογόνους. Οι απόγονοι που δεν ανήκουν στις κλάσεις αυτές δεν θα περιληφθούν στα αποτελέσματα της λειτουργίας.
arrHiddenTypes	Εξαιρούμενοι τύποι Φάκελοι	Κλάσεις φακέλων (σύνολο ζευγών <a href="#">κλάση/τύπος</a> ) που θα εξαιρεθούν από τους απογόνους. Η παράμετρος αυτή λαμβάνεται υπόψη, μόνο εάν η παράμετρος <a href="#">arrAllowedTypes</a> είναι κενή.
strState	Πλάνο εμφάνισης	Περιορίζει τους πόρους που θα παρουσιαστούν, με βάση το status της τοποθέτησης (δηλ. εάν έχει δηλωθεί ότι ένας πόρος θα ανήκει στο αρχείο του φακέλου, κοκ). Παίρνει τιμές: *, main , archive , personal Αν έχει τιμή "*" τότε θα εμφανιστούν όλοι οι πόροι του φακέλου, ειδάλως θα εμφανιστούν μόνο οι πόροι με την επιλεγμένη κατάσταση.
strPresentationOverride	Υποχρεωτικός Τρόπος Παρουσίασης	Καθορίζει τον τρόπο εντοπισμού των χαρακτηριστικών παρουσίασης ενός πόρου στα πλαίσια του Φακέλου όπου έχει τοποθετηθεί. Εάν έχει τιμή, τότε η τιμή αυτή είναι ο επιβαλλόμενος τρόπος παρουσίασης για όλους τους πόρους του Φακέλου. Αλλιώς, κάθε πόρος θα εμφανιστεί με τον ξεχωριστό τρόπο που έχει επιλεγεί κατά την τοποθέτησή του στο Φάκελο.

## XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία

Η λειτουργία δεν έχει δικό της σχήμα περιεχομένου, καθώς δεν παράγει μονάχα ένα XML αρχείο περιεχομένου, αλλά τόσο όσα και οι Υπο-Φάκελοι του φακέλου. Κάθε ένα από αυτά τα XML ακολουθεί το σχήμα που παρουσιάστηκε στη λειτουργία «Πόροι Φακέλου».



## 4.2.2 Λειτουργίες σε κλάσεις πόρων

### Μετα-δεδομένα Πόρου

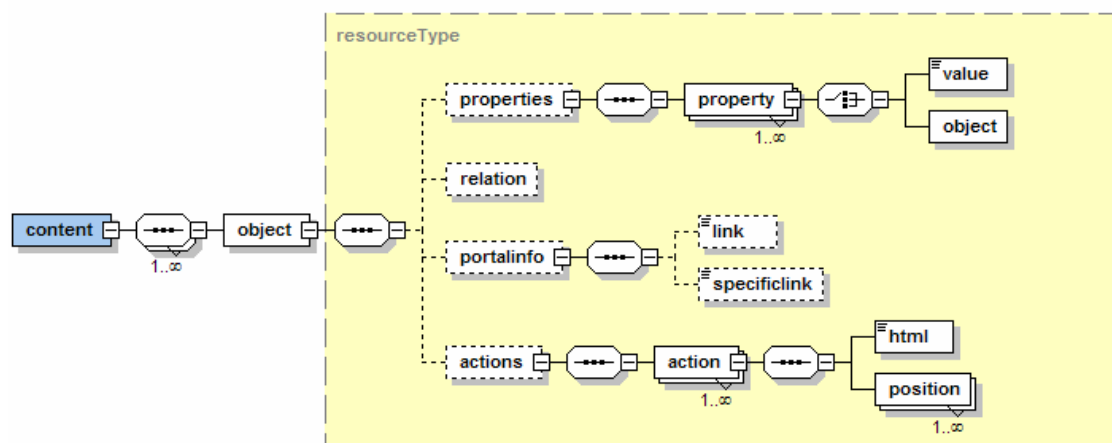
Η λειτουργία αυτή συγκεντρώνει και παρουσιάζει τα μετα-δεδομένα (τίτλο, περιγραφή, δημιουργό, ημερομηνία δημιουργίας, κλπ) ενός πόρου (δηλαδή, ενός αντικειμένου της κλάσης Πόρος).

Εκτός από τα βασικά μετα-δεδομένα ενός πόρου, η μπορεί ενδέχεται να παρουσιάζει και εξειδικευμένα στοιχεία, όπως το κείμενο ενός HTML ή το αρχείο ενός εγγράφου, κτλ, ανάλογα με την κλάση ενός πόρου («έγγραφο», «κείμενο», κλπ).

### Τρόπος κλήσης και παράμετροι κλήσης της λειτουργίας

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Περιγραφή Παραμέτρου
lngFileId	Κωδικός	Κωδικός του Πόρου. - Default τιμή: <a href="#">Τρέχον Πόρος (curresrc)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να εφαρμοστεί σε άλλο πόρο από τον τρέχοντα
lngContainerNodeId	Φάκελος Πόρου	Φάκελος στα πλαίσια του οποίου παρουσιάζεται ο Πόρος. - Default τιμή: <a href="#">Τρέχον Φάκελος (cumode)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη άλλο Φάκελο, όπου έχει τοποθετηθεί ο Πόρος
lngLanguageId	Γλώσσα	Γλώσσα των προς εμφάνιση πληροφοριών - Default τιμή: <a href="#">Τρέχουσα Γλώσσα (curlang)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να δείξει το περιεχόμενό της σε γλώσσα διαφορετική από την τρέχουσα
lngUserId	Χρήστης	Χρήστης που προσπελαύνει τις πληροφορίες του Πόρου - Default τιμή: <a href="#">Τρέχον Χρήστης (lngUserId)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη της τα δικαιώματα πρόσβασης διαφορετικού από τον τρέχοντα χρήστη
blnShowButtons	Παραπομπές	Αν η λειτουργία θα περιλάβει και τις απαραίτητες πληροφορίες για τις παραπομπές
blnIncludePresentationInfo	Τίτλοι	Αν η λειτουργία θα περιλάβει και τους τίτλους (labels) των μετα-δεδομένων του πόρου

### XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία







## Παράδειγμα περιεχομένου που επιστρέφει η λειτουργία

content																				
object																				
classification	resource																			
class	annusertext																			
properties																				
property (13)																				
name	sort	value	object	presentation																
1 rid		1089		presentation visible=false																
2 rtype		annusertext		presentation visible=false																
3 rpublished		-1		presentation visible=false group=0 required=false																
4 rname		Αλλαγή διαχείρισης του αρχείου ενός φακέλου		presentation <table border="1"> <tr> <td>visible</td> <td>true</td> </tr> <tr> <td>order</td> <td>0</td> </tr> <tr> <td>group</td> <td>0</td> </tr> <tr> <td>required</td> <td>false</td> </tr> <tr> <th colspan="2">description (2)</th> </tr> <tr> <th>language</th> <th>value</th> </tr> <tr> <td>1 0</td> <td>Τίτλος</td> </tr> <tr> <td>2 1</td> <td>Title</td> </tr> </table>	visible	true	order	0	group	0	required	false	description (2)		language	value	1 0	Τίτλος	2 1	Title
visible	true																			
order	0																			
group	0																			
required	false																			
description (2)																				
language	value																			
1 0	Τίτλος																			
2 1	Title																			
5 rowner			object	presentation visible=false order=1 group=0 required=false																
6 rauthor		Κωνσταντίνος Καράμπελας		presentation <table border="1"> <tr> <td>visible</td> <td>true</td> </tr> <tr> <td>order</td> <td>1</td> </tr> <tr> <td>group</td> <td>0</td> </tr> <tr> <td>required</td> <td>false</td> </tr> <tr> <th colspan="2">description (2)</th> </tr> <tr> <th>language</th> <th>value</th> </tr> <tr> <td>1 0</td> <td>Συγγραφέας</td> </tr> <tr> <td>2 1</td> <td>Author</td> </tr> </table>	visible	true	order	1	group	0	required	false	description (2)		language	value	1 0	Συγγραφέας	2 1	Author
visible	true																			
order	1																			
group	0																			
required	false																			
description (2)																				
language	value																			
1 0	Συγγραφέας																			
2 1	Author																			
7 rtext		/appdata/main/htmls/1_301.html		presentation <table border="1"> <tr> <td>visible</td> <td>true</td> </tr> <tr> <td>required</td> <td>true</td> </tr> <tr> <td>group</td> <td>1</td> </tr> </table>	visible	true	required	true	group	1										
visible	true																			
required	true																			
group	1																			
8 rtext_html		value																		
9 rdate	20050103	3/1/2005		presentation visible=false order=2 group=0 required=false																
10 rposted	20050103	3/1/2005		presentation visible=false order=3 group=0 required=false																
11 rsmalltext				presentation visible=false group=1 required=false																
12 reventdate				presentation visible=false order=1 group=0 required=false																
13 rdescr				presentation visible=false order=4 group=0 required=false																
portalinfo																				
link																				
specificlink																				
actions																				
action (6)																				
name	html	position																		
1 cmn_usrsc_rsc	html	position (4)																		
2 cmn_history	html	position (3)																		
3 cmn_toPersonal	html	position (2)																		
4 cmn_printsc	html	position (1)																		
5 cmn_mailrsc	html	position (1)																		
6 cmn_bck	html	position (1)																		

## Παράδειγμα εμφάνισης του περιεχομένου που επιστρέφει η λειτουργία

### Αλλαγή διαχείρισης του αρχείου ενός φακέλου






 Επιστροφή

**Τίτλος** : Αλλαγή διαχείρισης του αρχείου ενός φακέλου  
**Συγγραφέας** : Κωνσταντίνος Καράμπελας

Καλό θα ήταν να αλλάξει η διαχείριση της προβολής του αρχείου ενός κόμβου ώστε να είναι αυστηρώς καθορισμένη. Μια λύση θα περιλάμβανε τα εξής :

- Ξεχωριστού τύπου σελίδα (π.χ. `cpage=NODEARCH`)
- Σε εκείνη τη σελίδα να εμφανίζονται τα περιεχόμενα του κόμβου με επιλεγμένο state και με pagination

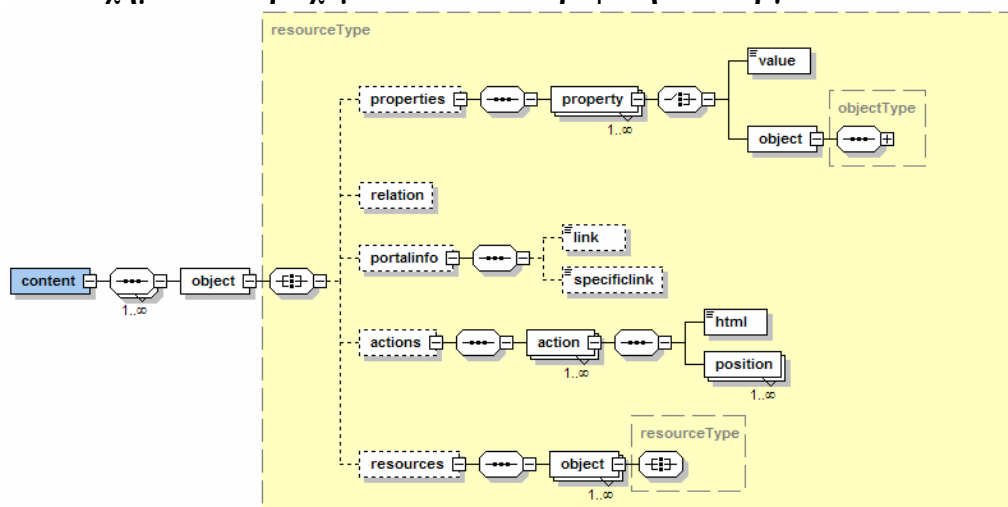
## Συσχετισμένοι Πόροι

Η λειτουργία αυτή συγκεντρώνει και παρουσιάζει πληροφορίες όλων των Πόρων ενός Portal που έχουν συσχετιστεί με ένα συγκεκριμένο Πόρο.

### Τρόπος κλήσης και παράμετροι κλήσης της λειτουργίας

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Περιγραφή Παραμέτρου
lngFileId	Κωδικός	Κωδικός του Πόρου. - Default τιμή: <a href="#">Τρέχον Πόρος (curresrc)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να εφαρμοστεί σε άλλο πόρο από τον τρέχοντα
lngContainerNodeId	Φάκελος Πόρου	Φάκελος στα πλαίσια του οποίου παρουσιάζεται ο Πόρος. - Default τιμή: <a href="#">Τρέχον Φάκελος (curnode)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη άλλο Φάκελο, όπου έχει τοποθετηθεί ο Πόρος
lngLanguageId	Γλώσσα	Γλώσσα των προς εμφάνιση πληροφοριών - Default τιμή: <a href="#">Τρέχουσα Γλώσσα (curlang)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να δείξει το περιεχόμενό της σε γλώσσα διαφορετική από την τρέχουσα
lngUserId	Χρήστης	Χρήστης που προσπελαύνει τις πληροφορίες του Πόρου - Default τιμή: <a href="#">Τρέχον Χρήστης (lngUserId)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη της τα δικαιώματα πρόσβασης διαφορετικού από τον τρέχοντα χρήστη
blnShowButtons	Παραπομπές	Αν η λειτουργία θα περιλάβει και τις απαραίτητες πληροφορίες για τις παραπομπές
blnIncludePresentationInfo	Τίτλοι	Αν η λειτουργία θα περιλάβει και τους τίτλους (labels) των μετα-δεδομένων του πόρου
strRelationType	Είδος Συσχέτισης	Το είδος της συσχέτισης που θα αναζητήσει η λειτουργία. Αν έχει τιμή "*" τότε θα εμφανιστούν όλοι οι πόροι που έχουν συσχετιστεί με το συγκεκριμένο πόρο, ανεξάρτητα από το είδος της συσχέτισής τους. Αλλιώς, θα εμφανιστούν μόνο οι πόροι με το επιλεγμένο είδος συσχέτισης




### XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία



## Παράδειγμα περιεχομένου που επιστρέφει η λειτουργία

content											
object											
classification		resource									
class		annusertext									
properties											
property		name=rid									
property		name=rtype									
property		name=rpublished									
property		name=rname									
		value: Αλλαγή διαχείρισης του αρχείου ενός φακέλου									
property		name=rowner									
property		name=rauthor									
property		name=rtext									
property		name=rtext_html									
property		name=rdate sort=20050103									
property		name=rposted sort=20050103									
portalinfo											
resources											
object (3)											
classification		class		properties				relation		actions	
1	resource	annusertext	properties (10)				relation	relative	actions		
				name				order			
1	rid	sort	value	object							
1	rid	1088									
2	rtype	annusertext									
3	rpublished	-1									
4	rname	Λύση για εύκολο Pagination πόρων									
5	rowner	object									
6	rauthor	Κωνσταντίνος Καράμπτελας									
7	rtext	/appdata/main/htmls/f_300.html									
8	rtext_html	value									
9	rdate	20050103	3/1/2005								
10	rposted	20050103	3/1/2005								
2	resource	annusertext	properties (9)				relation	next	actions		
				name				order			
1	rid	sort	value	object							
1	rid	1103									
2	rtype	annusertext									
3	rpublished	-1									
4	rname	Να αλλάξει ο τρόπος του Login για τον εξειδικευμένο χρήστη Anonymous									
5	rowner	object									
6	rauthor	Κωνσταντίνος Καράμπτελας									
7	rdate	20050124	24/1/2005								
8	rposted	20050124	24/1/2005								
9	rdescr	Καλό θα ήταν να γίνεται login ο ανώνυμος χωρίς να υπάρχει cuser και χωρίς να χρησιμοποιείται cookie									
3	resource	annusertext	properties (9)				relation	next	actions		
				name				order			
1	rid	sort	value	object							
1	rid	1104									
2	rtype	annusertext									
3	rpublished	-1									
4	rname	Βελτίωση της Login σελίδας									
5	rowner	object									
6	rauthor	Κωνσταντίνος Καράμπτελας									
7	rdate	20050124	24/1/2005								
8	rposted	20050124	24/1/2005								
9	rdescr	Πρέπει επιτέλους όπ εμφανίζεται στην Login σελίδα να μετατραπεί σε μερικά components για να είναι ρυθμίσιμα τα πάντα εκεί μέσα...									

## Παραδείγματα εμφάνισης του περιεχομένου που επιστρέφει η λειτουργία

Σχετικό Υλικό	
	Αύση για εύκολο Pagination πόρων
Επόμενο Υλικό	
	Να αλλάξει ο τρόπος του Login για τον εξειδικευμένο χρήστη Anonymous
	Βελτίωση της Login σελίδας
	Καλό θα ήταν να γίνεται login ο ανώνυμος χωρίς να υπάρχει cuser και χωρίς να χρησιμοποιείται cookie
	Πρέπει επιτέλους όπ εμφανίζεται στην Login σελίδα να μετατραπεί σε μερικά components για να είναι ρυθμίσιμα τα πάντα εκεί μέσα...

## Εμπεριέχοντες Φάκελοι

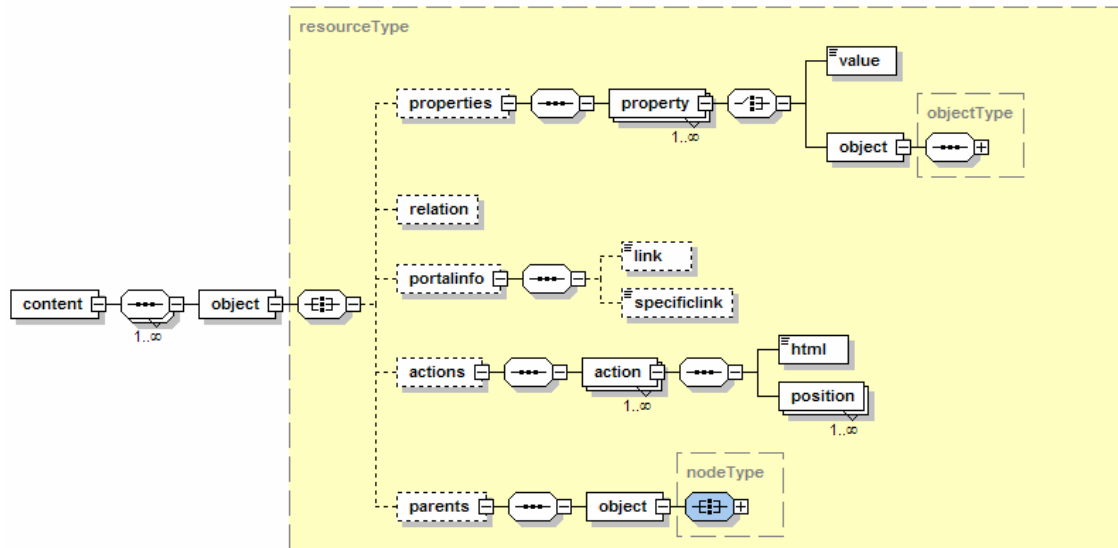
Η λειτουργία «Εμπεριέχοντες Φάκελοι» συγκεντρώνει και παρουσιάζει για ένα πόρο ενός Portal, όλους τους Φάκελους του Portal όπου έχει τοποθετηθεί ο πόρος αυτός.

Όπως αναφέρθηκε στη περιγραφή του μοντέλου δεδομένων του COMMET, ένας πόρος μπορεί να τοποθετηθεί σε πολλούς Φάκελους ενός Portal.

## Τρόπος κλήσης και παράμετροι κλήσης της λειτουργίας

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Περιγραφή Παραμέτρου
lngFileId	Κωδικός	Κωδικός του Πόρου. - Default τιμή: <a href="#">Τρέχον Πόρος (currsrc)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να εφαρμοστεί σε άλλο πόρο από τον τρέχοντα
lngContainerNodeId	Φάκελος Πόρου	Φάκελος στα πλαίσια του οποίου παρουσιάζεται ο Πόρος. - Default τιμή: <a href="#">Τρέχον Φάκελος (currnode)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη άλλο Φάκελο, όπου έχει τοποθετηθεί ο Πόρος
lngLanguageId	Γλώσσα	Γλώσσα των προς εμφάνιση πληροφοριών - Default τιμή: <a href="#">Τρέχουσα Γλώσσα (curlang)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να δείξει το περιεχόμενό της σε γλώσσα διαφορετική από την τρέχουσα
lngUserId	Χρήστης	Χρήστης που προσπελάει τις πληροφορίες του Πόρου - Default τιμή: <a href="#">Τρέχον Χρήστης (lngUserId)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη της τα δικαιώματα πρόσβασης διαφορετικού από τον τρέχοντα χρήστη
blnShowButtons	Παραπομπές	Αν η λειτουργία θα περιλάβει και τις απαραίτητες πληροφορίες για τις παραπομπές
blnIncludePresentationInfo	Τίτλοι	Αν η λειτουργία θα περιλάβει και τους τίτλους (labels) των μετα-δεδομένων του πόρου



## XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία



## Παράδειγμα περιεχομένου που επιστρέφει η λειτουργία

content			
object			
classification	resource		
class	annusertext		
properties			
property (10)			
name	sort	value	obj..
1 rid		1089	
2 rtype		annusertext	
3 rpublished		-1	
4 rname		Αλλαγή διαχείρισης του αρχείου ενός φακέλου	
5 rowner			obj..
6 rauthor		Κωνσταντίνος Καράμπτελας	
7 rtext		/appdata/main/htmls/f_301.html	
8 rtext_html		value	
9 rdate	20050103	3/1/2005	
10 rposted	20050103	3/1/2005	
portalinfo			
parents			
object			
classification	node		
class	blackboard		
visible	true		
properties			
property (9)			
name	sort	value	object
1 nid		2444	
2 ntype		blackboard	
3 nsubtype		pendingjobs	
4 nname		Εκκρεμότητες - Λύσεις	
5 ndescr		Λογισμικό της εφαρμογής Δικτυακής Πύλης &amp; της εφαρμογής Διαχείρισης Περιεχομένου σε περιβάλλον MS ASP	
6 ncreator			object
7 nsavepath		main/	
8 ncolor		#FF3333	
9 ndatecreated	20041207	7/12/2004	
relation			
actions			
action (3)			
name	html	position	
1 cmn_showarch	html	position (1)	
2 cmn_showmain	html	position (1)	
3 cmn_path	html	position (2)	

## Παραδείγματα εμφάνισης του περιεχομένου που επιστρέφει η λειτουργία

Περιέχεται σε	
 Εκκρεμότητες - Λύσεις	Λογισμικό της εφαρμογής Δικτυακής Πύλης & της εφαρμογής Διαχείρισης Περιεχομένου σε περιβάλλον MS ASP 

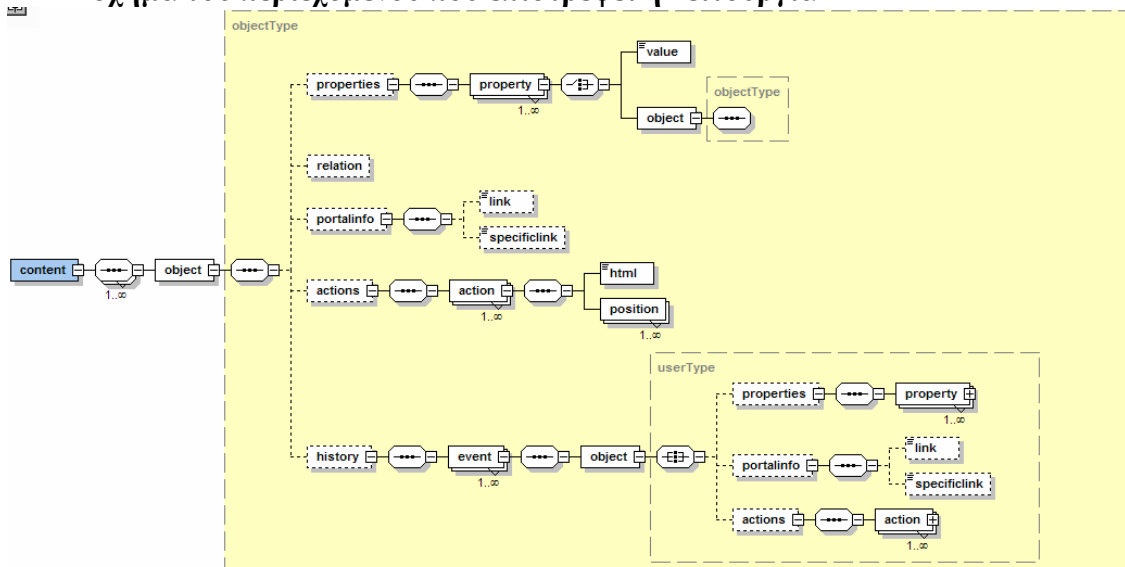
## Ιστορικό Διαχείρισης Πόρου

Η λειτουργία αυτή συγκεντρώνει και παρουσιάζει για ένα πόρο ενός Portal, στοιχεία που αφορούν τις ενέργειες διαχείρισης που έχουν εκτελεστεί σε αυτόν (εισαγωγή, τροποποίηση, εμφάνιση).

### Τρόπος κλήσης και παράμετροι κλήσης της λειτουργίας

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Περιγραφή Παραμέτρου
lngFileId	Κωδικός	Κωδικός του Πόρου. - Default τιμή: <a href="#">Τρέχον Πόρος (curresrc)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να εφαρμοστεί σε άλλο πόρο από τον τρέχοντα
lngContainerNodeid	Φάκελος Πόρου	Φάκελος στα πλαίσια του οποίου παρουσιάζεται ο Πόρος. - Default τιμή: <a href="#">Τρέχον Φάκελος (curnode)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη άλλο Φάκελο, όπου έχει τοποθετηθεί ο Πόρος
lngLanguageId	Γλώσσα	Γλώσσα των προς εμφάνιση πληροφοριών - Default τιμή: <a href="#">Τρέχουσα Γλώσσα (curlang)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να δείξει το περιεχόμενό της σε γλώσσα διαφορετική από την τρέχουσα
lngUserId	Χρήστης	Χρήστης που προσπελάζει τις πληροφορίες του Πόρου - Default τιμή: <a href="#">Τρέχον Χρήστης (lngUserId)</a> Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη της τα δικαιώματα πρόσβασης διαφορετικού από τον τρέχοντα χρήστη
blnShowButtons	Παραπομπές	Αν η λειτουργία θα περιλάβει και τις απαραίτητες πληροφορίες για τις παραπομπές
blnIncludePresentationInfo	Τίτλοι	Αν η λειτουργία θα περιλάβει και τους τίτλους (labels) των μετα-δεδομένων του πόρου

### XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία





## Παράδειγμα περιεχομένου που επιστρέφει η λειτουργία

content

- object
  - classification resource
  - class annusertext
  - properties
    - property name=rid
    - property name=rtype
    - property name=rpublished
    - property
      - name rname
      - value Αλλαγή διαχείρισης του αρχείου ενός φακέλου
    - property name=rowner
    - property name=rauthor
    - property name=rtext
    - property name=rtext\_html
    - property name=rdate sort=20050103
    - property name=rposted sort=20050103
  - portalinfo
  - history
 

event (5)	type	date	object
1	Created	3/1/2005 6:04:20 μμ	<ul style="list-style-type: none"> <li>object               <ul style="list-style-type: none"> <li>classification user</li> <li>class user</li> <li>properties                   <ul style="list-style-type: none"> <li>property name=uid</li> <li>property name=username</li> <li>property name=upassword</li> <li>property                       <ul style="list-style-type: none"> <li>name ufirstname</li> <li>value Κωνσταντίνος</li> </ul> </li> <li>property                       <ul style="list-style-type: none"> <li>name ulastname</li> <li>value Καράμπελας</li> </ul> </li> <li>property name=unotes</li> <li>property name=uemail</li> <li>property name=ucv</li> </ul> </li> <li>portalinfo</li> </ul> </li> </ul>
2	Viewed	2/2/2005 12:50:29 μμ	object classification=user class=user
3	Modified	2/2/2005 12:51:33 μμ	object classification=user class=user
4	Viewed	2/2/2005 1:03:54 μμ	object classification=user class=user
5	Viewed	2/2/2005 1:06:50 μμ	object classification=user class=user

## Παραδείγματα εμφάνισης του περιεχομένου που επιστρέφει η λειτουργία

Ιστορικό του "Αλλαγή διαχείρισης του αρχείου ενός φακέλου"		
3/1/2005 6:04:20 μμ	Created	Κωνσταντίνος Καράμπελας
2/2/2005 12:50:29 μμ	Viewed	Δημήτρης Ντζίδος
2/2/2005 12:51:33 μμ	Modified	Κωνσταντίνος Καράμπελας
2/2/2005 1:03:54 μμ	Viewed	Κωνσταντίνος Καράμπελας
2/2/2005 1:06:50 μμ	Viewed	Κωνσταντίνος Καράμπελας

### 4.2.3 Γενικές λειτουργίες

#### Μενού

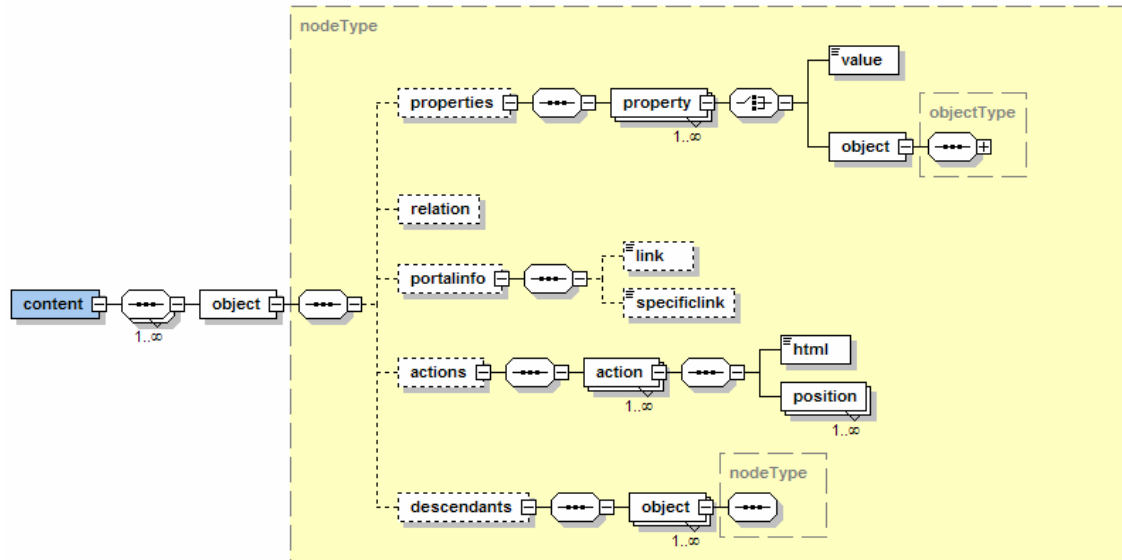
Η λειτουργία αυτή συγκεντρώνει και παρουσιάζει τους απογόνους ενός φακέλου, με σκοπό την παρουσίασή τους σε μορφή μενού πλοήγησης.

Το περιεχόμενο που επιστρέφει λειτουργίας είναι παρόμοιο με αυτό που περιγράφεται για την λειτουργία «Απόγονοι Φακέλου». Ο τρόπος, όμως, με τον οποίο μορφοποιείται το περιεχόμενο είναι ο τυπικός τρόπος των μενού πλοήγησης.

#### Παράμετροι κλήσης της λειτουργίας

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Περιγραφή Παραμέτρου
lngRootId	Κωδικός Ρίζας Μενού	Ο Φάκελος που θα αποτελεί τη ρίζα του Μενού. - Default τιμή: <a href="#">Ρίζα του τρέχοντος Portal</a> (lngPortalRoot) Χρησιμοποιείται μόνο εάν το μενού πρέπει να ξεκινά από φάκελο διαφορετικό από τη ρίζα του Portal
lngLanguageId	Γλώσσα	Γλώσσα των προς εμφάνιση πληροφοριών - Default τιμή: <a href="#">Τρέχουσα Γλώσσα</a> (curlang) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να δείξει το περιεχόμενό της σε γλώσσα διαφορετική από την τρέχουσα
lngUserId	Χρήστης	Χρήστης που προσπελάζει τις πληροφορίες του Φακέλου - Default τιμή: <a href="#">Τρέχον Χρήστης</a> (lngUserId) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη της τα δικαιώματα πρόσβασης διαφορετικού από τον τρέχοντα χρήστη
blnShowButtons	Παραπομπές	Αν η λειτουργία θα περιλάβει και τις απαραίτητες πληροφορίες για τις παραπομπές
blnIncludePresentationInfo	Τίτλοι	Αν η λειτουργία θα περιλάβει και τους τίτλους (labels) των μετα-δεδομένων
arrAllowedTypes	Συμπεριλαμβανόμενοι τύποι Φακέλων	Κλάσεις φακέλων (σύνολο ζευγών <a href="#">κλάση/τύπος</a> ) που θα περιληφθούν στους απογόνους. Οι απόγονοι που δεν ανήκουν στις κλάσεις αυτές δεν θα περιληφθούν στα αποτελέσματα της λειτουργίας.
arrHiddenTypes	Εξαιρούμενοι τύποι Φακέλοι	Κλάσεις φακέλων (σύνολο ζευγών <a href="#">κλάση/τύπος</a> ) που θα εξαιρεθούν από τους απογόνους. Η παράμετρος αυτή λαμβάνεται υπόψη, μόνο εάν η παράμετρος <a href="#">arrAllowedTypes</a> είναι κενή.
lngDepth	Βάθος απογόνων	Το βάθος μέχρι το οποίο θα φτάσει το μενού. Αν έχει τιμή 1 τότε το μενού θα έχει βάθος 1, αν έχει τιμή 2 τότε το μενού θα έχει βάθος 2, κ.ο.κ.

## XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία



## Παράδειγμα περιεχομένου που επιστρέφει η λειτουργία

content							
object							
<input checked="" type="checkbox"/>	classification	node					
<input checked="" type="checkbox"/>	class	root					
<input checked="" type="checkbox"/>	visible	true					
properties							
<input checked="" type="checkbox"/>	property	name=nid					
property							
<input checked="" type="checkbox"/>	name	ntype					
<input checked="" type="checkbox"/>	value	root					
<input checked="" type="checkbox"/>	property	name=nsubtype					
property							
<input checked="" type="checkbox"/>	name	nname					
<input checked="" type="checkbox"/>	value	COMMENT					
<input checked="" type="checkbox"/>	property	name=ndescr					
<input checked="" type="checkbox"/>	property	name=nsavepath					
<input checked="" type="checkbox"/>	property	name=ncolor					
<input checked="" type="checkbox"/>	relation	level=0					
<input checked="" type="checkbox"/>	portalinfo						
descendants							
object (5)							
<input checked="" type="checkbox"/>	clas...	class	visible	properties	relation	portalinfo	
<input checked="" type="checkbox"/>	1	node	folder	true	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
properties							
property (8)							
<input checked="" type="checkbox"/>	1	nid	1968				
<input checked="" type="checkbox"/>	2	ntype	folder				
<input checked="" type="checkbox"/>	3	nsubtype	teamContainer				
<input checked="" type="checkbox"/>	4	nname	Αντικείμενα Εργασιών				
<input checked="" type="checkbox"/>	5	ndescr	Αντικείμενα εργασιών ανάπτυξης λογισμικού στο περιβάλλον COMMENT				
<input checked="" type="checkbox"/>	6	ncreator		<input checked="" type="checkbox"/>			
<input checked="" type="checkbox"/>	7	nsavepath	main/				
<input checked="" type="checkbox"/>	8	ncolor	#FF3333				
<input checked="" type="checkbox"/>	2	node	folder	true	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
properties							
property (7)							
<input checked="" type="checkbox"/>	3	node	folder	true	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
properties							
property (9)							
<input checked="" type="checkbox"/>	4	node	blackboard	true	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
properties							
property (7)							
<input checked="" type="checkbox"/>	5	node	folder	true	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
properties							
property (8)							

## Παραδείγματα εμφάνισης του περιεχομένου που επιστρέφει η λειτουργία

<b>COMMET</b>
> Αντικείμενα Εργασιών
> COMMET ASP
> COMMET NET
> COMMET Tools
> COMMET Access
> Installations
> Βιβλιοθήκη
> CMS
> MS ASP, .NET
> JAVA
> Συμμετέχοντες
> Υπεύθυνοι εφαρμογής
> Προγραμματιστές
> Γενικές Ανακοινώσεις
> Γενικές Συναντήσεις
> Μηχανισμός Patches
> Add-on Tools

## Δυναμικό Μενού

Η λειτουργία «Δυναμικό Μενού» θα μπορούσε να είναι μία έκδοση της λειτουργίας «Μενού» με διαφορετικό τρόπο εμφάνισης του περιεχομένου που έχει συγκεντρώσει. Η διαφορά εμφάνισης έγκειται στο εξής:

- Το στατικό μενού παρουσιάζει δέντρο με όλους τους απογόνους της ρίζας (μέχρι το βάθος που έχει οριστεί)
- Το δυναμικό μενού παρουσιάζει το ίδιο δέντρο, εμφανίζοντας επιπλέον και τα παιδιά του φακέλου που έχει επιλεγεί (τρέχοντος φακέλου)

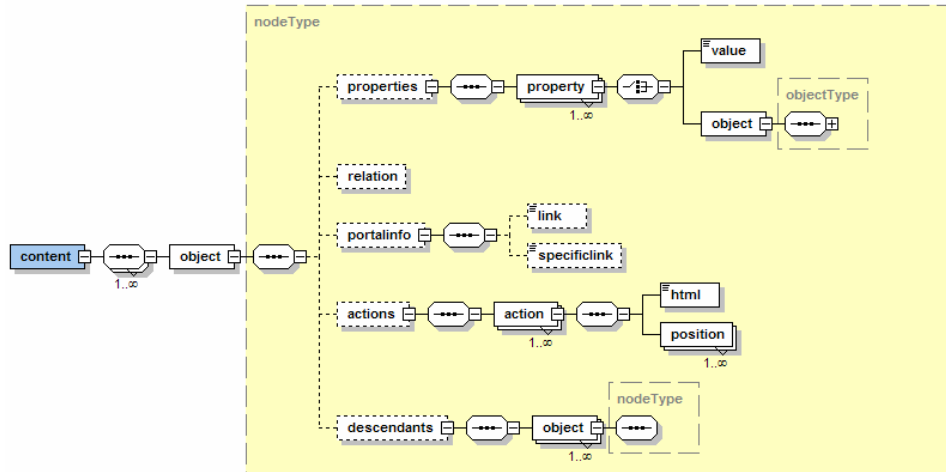
Για λόγους επίδοσης (ταχύτητα δημιουργίας των σελίδων του Portal), η λειτουργία «Δυναμικό Μενού» συγκεντρώνει πληροφορίες μόνο για τους Φακέλους που πρόκειται να εμφανίσει. Δηλαδή, συγκεντρώνει πληροφορίες μόνο για τους Φακέλους που ανήκουν στα σταθερά επίπεδα που εμφανίζει, και επιπλέον για τους Υπο-Φακέλους του τρέχοντος Φακέλου.

Η λειτουργία αυτή είναι μια παραλλαγή της λειτουργίας «Μενού». Η διαφορά έγκειται στο ότι οι κόμβοι που συμπεριλαμβάνονται στο περιεχόμενο της ενδέχεται να είναι περισσότεροι από αυτούς που θα περιείχε το αποτέλεσμα της λειτουργίας «Μενού» με τις ίδιες παραμέτρους. Συγκεκριμένα, θα συμπεριληφθούν και οι υπο-φάκελοι του επιλεγμένου φάκελου, ακόμα κι αν το μέγιστο βάθος αναζήτησης ξεπερνιέται με αυτό τον τρόπο κατά ένα.

## Παράμετροι κλήσης της λειτουργίας

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Περιγραφή Παραμέτρου
lngRootId	Κωδικός Ρίζας Μενού	Ο Φάκελος που θα αποτελεί τη ρίζα του Μενού. - Default τιμή: <a href="#">Ρίζα του τρέχοντος Portal</a> (lngPortalRoot) Χρησιμοποιείται μόνο εάν το μενού πρέπει να ξεκινά από φάκελο διαφορετικό από τη ρίζα του Portal
lngLanguageId	Γλώσσα	Γλώσσα των προς εμφάνιση πληροφοριών - Default τιμή: <a href="#">Τρέχουσα Γλώσσα</a> (curlang) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να δείξει το περιεχόμενο της σε γλώσσα διαφορετική από την τρέχουσα
lngUserId	Χρήστης	Χρήστης που προσπελαύνει τις πληροφορίες του Φακέλου - Default τιμή: <a href="#">Τρέχον Χρήστης</a> (lngUserId) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη της τα δικαιώματα πρόσβασης διαφορετικού από τον τρέχοντα χρήστη
blnShowButtons	Παραπομπές	Αν η λειτουργία θα περιλάβει και τις απαραίτητες πληροφορίες για τις παραπομπές
blnIncludePresentationInfo	Τίτλοι	Αν η λειτουργία θα περιλάβει και τους τίτλους (labels) των μετα-δεδομένων
arrAllowedTypes	Συμπεριλαμβανόμενοι τύποι Φακέλων	Κλάσεις φακέλων (σύνολο ζευγών <a href="#">κλάση/τύπος</a> ) που θα περιληφθούν στους απογόνους. Οι απόγονοι που δεν ανήκουν στις κλάσεις αυτές δεν θα περιληφθούν στα αποτελέσματα της λειτουργίας.
arrHiddenTypes	Εξαιρούμενοι τύποι Φάκελοι	Κλάσεις φακέλων (σύνολο ζευγών <a href="#">κλάση/τύπος</a> ) που θα εξαιρεθούν από τους απογόνους. Η παράμετρος αυτή λαμβάνεται υπόψη, μόνο εάν η παράμετρος <a href="#">arrAllowedTypes</a> είναι κενή.
lngDepth	Βάθος απογόνων	Το βάθος μέχρι το οποίο θα φτάσει το μενού. Αν έχει τιμή 1 τότε το μενού θα έχει βάθος 1, αν έχει τιμή 2 τότε το μενού θα έχει βάθος 2, κ.ο.κ

## XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία



## Παράδειγμα περιεχομένου που επιστρέφει η λειτουργία

content									
object									
= classification		node							
= class		root							
= visible		true							
properties									
property		name=nid							
property		name=nstype							
= value		root							
property		name=nsubtype							
property		name=nname							
= value		COMMENT							
property		name=nsavedscr							
property		name=nsavepath							
property		name=ncolor							
relation		level=0							
portalinfo									
descendants									
object (5)									
1	node	folder	true	properties	relation	portalinfo	descendants		
properties									
property (8)									
1	name	sort	value	object					
1	nid		2417						
2	nstype		folder						
3	nsubtype		persongroupContainer						
4	nname		Συμμετέχοντες						
5	ncreator			object					
6	nsavepath		main/						
7	ncolor		#CC0066						
8	ndatecreated		20040917 17/9/2004						
4	node	blackboard	true	properties	relation level=1	portalinfo	descendants		
properties									
relation level=1									
portalinfo									
descendants									
object									
= classification		node							
= class		folder							
= visible		true							
properties									
relation		level=2							
portalinfo									
object									
= classification		node							
= class		folder							
= visible		true							
properties									
relation		level=2							
portalinfo									
5	node	folder	true	properties	relation level=1	portalinfo			
properties									
relation level=1									
portalinfo									

## Παράδειγματα εμφάνισης του περιεχομένου που επιστρέφει η λειτουργία

### COMMENT

- > Αντικείμενα Εργασιών
- > Βιβλιοθήκη
- > Συμμετέχοντες
  - > Υπεύθυνοι εφαρμογής
  - > Προγραμματιστές
- > Γενικές Ανακοινώσεις
- > Γενικές Συναντήσεις

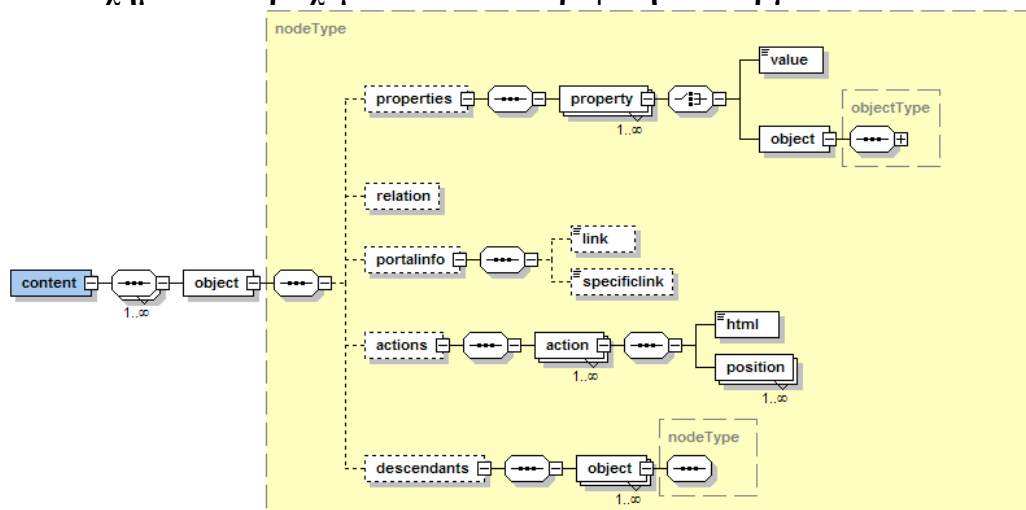
## Γραμμή πλοήγησης

Η λειτουργία αυτή αναζητά και παρουσιάζει το πλήρες μονοπάτι της ιεραρχίας των φακέλων ενός Portal, από τον τρέχοντα φάκελο μέχρι τον Φάκελο που βρίσκεται στην κορυφή της ιεραρχίας.

## Παράμετροι κλήσης της λειτουργίας

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Περιγραφή Παραμέτρου
lngRootId	Κωδικός Φακέλου	Ο Φάκελος που θα αποτελεί το τέλος του μονοπατιού, το οποίο ξεκινά από τη ρίζα του Portal. - Default τιμή: <a href="#">Τρέχον Φάκελος</a> (curmode) Χρησιμοποιείται μόνο εάν το μονοπάτι πρέπει να φτάσει σε διαφορετικό φάκελο από τον τρέχοντα φάκελο
lngLanguageId	Γλώσσα	Γλώσσα των προς εμφάνιση πληροφοριών - Default τιμή: <a href="#">Τρέχουσα Γλώσσα</a> (curlang) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να δείξει το περιεχόμενό της σε γλώσσα διαφορετική από την τρέχουσα
lngUserId	Χρήστης	Χρήστης που προσπελαύνει τις πληροφορίες του Φακέλου - Default τιμή: <a href="#">Τρέχον Χρήστης</a> (lngUserId) Χρησιμοποιείται μόνο εάν η λειτουργία πρέπει να λάβει υπόψη της τα δικαιώματα πρόσβασης διαφορετικού από τον τρέχοντα χρήστη
blnShowButtons	Παραπομπές	Αν η λειτουργία θα περιλάβει και τις απαραίτητες πληροφορίες για τις παραπομπές
blnIncludePresentationInfo	Τίτλοι	Αν η λειτουργία θα περιλάβει και τους τίτλους (labels) των μετα-δεδομένων
arrAllowedTypes	Συμπεριλαμβανόμενοι τύποι Φακέλων	Κλάσεις φακέλων (σύνολο ζευγών <a href="#">κλάση/τύπος</a> ) που θα περιληφθούν στους απογόνους. Οι απόγονοι που δεν ανήκουν στις κλάσεις αυτές δεν θα περιληφθούν στα αποτελέσματα της λειτουργίας.
arrHiddenTypes	Εξαιρούμενοι τύποι Φακέλοι	Κλάσεις φακέλων (σύνολο ζευγών <a href="#">κλάση/τύπος</a> ) που θα εξαιρεθούν από τους απογόνους. Η παράμετρος αυτή λαμβάνεται υπόψη, μόνο εάν η παράμετρος <a href="#">arrAllowedTypes</a> είναι κενή.

## XML σχήμα του περιεχομένου που επιστρέφει η λειτουργία



## Παράδειγμα περιεχομένου που επιστρέφει η λειτουργία

▲ content	
▲ object	
▣ classification	node
▣ class	root
▣ visible	false
▲ descendants	
▲ object	
▣ classification	node
▣ class	folder
▣ visible	true
▼ properties	
▼ relation level=1	
▼ portalinfo	
▲ descendants	
▲ object	
▣ classification	node
▣ class	team
▣ visible	true
▼ properties	
▼ relation level=2	
▼ portalinfo	
▲ descendants	
▲ object	
▣ classification	node
▣ class	folder
▣ visible	true
▼ properties	
▼ relation level=3	
▼ portalinfo	
▲ descendants	
▲ object	
▣ classification	node
▣ class	meeting
▣ visible	true
▼ properties	
▼ relation level=4	
▼ portalinfo	

## Παράδειγματα εμφάνισης του περιεχομένου που επιστρέφει η λειτουργία

[Αντικείμενα Εργασιών](#) ► [COMMET Tools](#) ► [Ειδικές Συναντήσεις](#) ► [Διαχείριση αρχείου Φακέλων](#)



# 5 Η γλώσσα σήμανσης XML

## 5.1.1 Γενικά

Η XML είναι markup γλώσσα για έγγραφα που περιέχουν δεδομένα με ιεραρχική δόμηση. Markup γλώσσα γενικά είναι ένας μηχανισμός που καθορίζει δομές σε ένα έγγραφο. Οι δομές αυτές περιλαμβάνουν δεδομένα και μεταδεδομένα (metadata) που περιγράφουν την σημασιολογία των δεδομένων αυτών.

Η XML είναι συντομογραφία για το EXtensible Markup Language. Είναι markup γλώσσα, η οποία μοιάζει με την HTML και σχεδιάστηκε για να περιγράφει δεδομένα. Αποτελεί απόγονο της γλώσσας σήμανσης SGML (Standardized General Markup Language), η οποία είναι βιομηχανικό πρότυπο και χρησιμοποιούνταν κυρίως με προγράμματα εμφάνισης ιεραρχικών δομών περιεχομένου (outliners). Επιχειρώντας μια σύγκριση μεταξύ XML και HTML καταλήγουμε στο ότι η XML σχεδιάστηκε για τη μεταφορά δεδομένων και δεν αποτελεί ένα αντικατάστατο της HTML. Η XML σχεδιάστηκε για να περιγράφει τα δεδομένα και εστιάζει στο τι είναι τα δεδομένα, ενώ η HTML σχεδιάστηκε για να παρουσιάζει τα δεδομένα και εστιάζει στο πώς φαίνονται τα δεδομένα. Η HTML έχει να κάνει με την μορφοποίηση/παρουσίαση των δεδομένων ενώ η XML έχει να κάνει με την περιγραφή των ίδιων των δεδομένων.

Η βασική δομή σε ένα έγγραφο XML είναι οι ετικέτες (tags). Τα XML tags δεν είναι προκαθορισμένα, αλλά ορίζονται από τον χρήστη. Τα tags που χρησιμοποιούνται στα HTML κείμενα είναι προκαθορισμένα. Ο συγγραφέας ενός HTML κειμένου μπορεί να χρησιμοποιήσει μόνο τα tags που ορίζονται στο HTML standard (όπως <p>, <h1> κ.λ.π.). Η XML επιτρέπει στο συγγραφέα ενός XML κειμένου να ορίσει τα δικά του tags. Είναι πολύ πιθανόν ότι οι μελλοντικές Web εφαρμογές θα χρησιμοποιούν την XML για να περιγράφουν τα δεδομένα, ενώ η HTML ή κάποια πιο εξελιγμένη έκδοσή της, όπως η eXtensible HTML (XHTML), θα χρησιμοποιείται για την απεικόνιση των δεδομένων αυτών.

Ακολουθεί ένα παράδειγμα με τα περιεχόμενα ενός αρχείου .xml στο οποίο διακρίνονται η ιεραρχική οργάνωση των ετικετών.

```
<?xml version='1.0' ?>
<product_list version='0.2.3' curdate='Fri, 17 Oct 2004 16:19:37 GMT'>
  <products>
    <product name='Helpdesk Kit' url='http://www.mysite.gr' descr='Technician Tool'
      IsEnabled='1' Priority='1' Order='1' >
      <Expansions>
        <Recovery consolesnapin='1' />
      </Expansions >
    </product>
  </products>
</product_list>
```

Σε ένα έγγραφο που χρησιμοποιεί την γλώσσα XML διακρίνουμε ως βασική δομική ετικέτα το element (στοιχείο), όπως στο παράδειγμά μας είναι το <xml> ή το <product\_list>. Όπως παρατηρούμε όλα τα στοιχεία περικλείονται μεταξύ των αγκυλών <...>. Μέσα σε κάθε στοιχείο τώρα μπορεί να υπάρχουν μια σειρά από attributes (χαρακτηριστικά) του στοιχείου αυτού. Επίσης το στοιχείο διαθέτει άλλα υποστοιχεία και αυτά με τη σειρά τους τα δικά τους υποστοιχεία, σε μια απόλυτα αυστηρή ιεραρχία.

Εκμεταλλευόμενοι την ιεραρχία που υποστηρίζεται εγγενώς από την XML, μπορούμε να περιγράψουμε μια σειρά από taxonomies σε μια οντολογία δεδομένων με χρήση ενός XML εγγράφου. Τα taxonomies είναι ιεραρχικά οργανωμένες δομές οντοτήτων για ένα δεδομένο σύνολο αντικειμένων. Αυτή η δυνατότητα είναι ιδιαίτερα χρήσιμη σε ένα σύστημα διαχείρισης περιεχομένου. Σε ένα CMS οι οντότητες μπορούν να οργανωθούν με βάση τα taxonomies.

## 5.1.2 DTD

Η XML χρησιμοποιεί ένα μοντέλο για την περιγραφή των εγγράφων XML. Ένα τέτοιο μοντέλο ονομάζεται Document Type Definition (DTD). Η XML το χρησιμοποιεί για να περιγράψει τα δεδομένα. Το DTD είναι ένας μηχανισμός που κληρονομήθηκε από την γλώσσα SGML και χρησιμοποιείται για την περιγραφή κάθε αντικειμένου σε ένα έγγραφο XML, όπως είναι τα στοιχεία και τα χαρακτηριστικά. Αποτελεί ένα σύνολο κανόνων που αφορούν στα tags της γλώσσας XML. Πιο συγκεκριμένα ορίζει κανόνες για το ποια είναι τα επιτρεπτά ονόματα στοιχείων και τα επιτρεπτά υποστοιχεία (sub-elements) και χαρακτηριστικά (attributes) για κάθε συγκεκριμένο στοιχείο. Ακολουθεί ένα έγγραφο DTD. Να αναφέρουμε ότι στα έγγραφα με μορφοποίηση XML τα σχόλια περιλαμβάνονται μεταξύ των <!-- και -->.

```
<!ELEMENT address-book (entry+)> <!-- Αυτό σημαίνει ότι το στοιχείο address-book περιλαμβάνει ένα ή περισσότερα το πλήθος στοιχεία entry -->
<!ELEMENT entry (name,address*,tel*,email*)> <!-- Σε όλες τις γραμμές του DTD εγγράφου τα περιεχόμενα των παρενθέσεων ονομάζονται content model (μοντέλο περιεχομένου). Εντός του μοντέλου περιεχομένου ενδέχεται να χρησιμοποιούνται περισσότερες παρενθέσεις για την ομαδοποίηση των στοιχείων. -->
```

```
<!ELEMENT name (#PCDATA | fname | lname)*>
<!ELEMENT fname (#PCDATA)> <!-- Η δεσμευμένη λέξη PCDATA σημαίνει parsed character data και σημαίνει ότι τα συγκεκριμένο στοιχείο μπορεί να περιέχει κείμενο. Χρησιμοποιείται συνήθως για τα στοιχεία «φύλλα». -->
<!ELEMENT lname (#PCDATA)>
```

```
<!ELEMENT address (street,region?,country)>
<!ATTLIST address preferred (true | false) "false">
<!ELEMENT street (#PCDATA)> <!-- Το στοιχείο street είναι ένα στοιχείο «φύλλο» διότι δεν έχει παιδιά -->
<!ELEMENT region (#PCDATA)>
<!ELEMENT country (#PCDATA)>
```

```
<!ELEMENT tel (#PCDATA)>
<!ATTLIST tel preferred (true | false) "false">
```

```
<!ELEMENT email (#PCDATA)>
<!ATTLIST email href CDATA #REQUIRED
preferred (true | false) "false">
```

Το παραπάνω DTD έγγραφο μπορεί να συνοδεύει το ακόλουθο έγγραφο XML:

```
<?xml version="1.0"?>
<!DOCTYPE address-book SYSTEM "address-book.dtd">
<address-book>
  <entry>
    <name>Γιάννης Παπαδόπουλος</name>
    <address>
      <street>Παπάγου 82</street>
      <region>Ζωγράφου</region>
      <country>Ελλάδα</country>
    </address>
    <tel preferred="true">2107732456</tel>
    <email href="mailto:gpadop@mail.gr">
  </entry>
</address-book>
```

Να κάνουμε τις εξής παρατηρήσεις σχετικά με την δομή και την σύνταξη των εγγράφων DTD:

- Όλα τα ονόματα των αντικειμένων ξεκινούν με γράμμα και δεν επιτρέπεται ο χαρακτήρας κενού διαστήματος. Στην θέση του χρησιμοποιείται το ενωτικό.
- Η XML υποστηρίζει πλήρως το πρότυπο κωδικοποίησης χαρακτήρων Unicode, συνεπώς μπορούμε ελεύθερα να χρησιμοποιούμε ελληνικούς χαρακτήρες σαν έγκυρους χαρακτήρες ονομάτων για τα στοιχεία του εγγράφου μας
- Οι χαρακτήρες \*, ? και + ονομάζονται χαρακτήρες ύπαρξης και υποδεικνύουν αν και πώς τα στοιχεία στην λίστα των παιδιών επαναλαμβάνονται. Αν δεν υπάρχει δείκτης ύπαρξης σε ένα στοιχείο αυτό σημαίνει ότι το συγκεκριμένο στοιχείο εμφανίζεται μια φορά. Στοιχείο με τον χαρακτήρα + πρέπει να εμφανίζεται μία τουλάχιστον φορά. Με τον χαρακτήρα \* καμία ή περισσότερες φορές. Με τον χαρακτήρα ? σημαίνει ότι το στοιχείο μπορεί να εμφανιστεί το πολύ μία φορά.
- Οι χαρακτήρες , και | ονομάζονται συζευκτήρες. Βοηθούν στον διαχωρισμό των παιδιών ενός στοιχείου μέσα στο μοντέλο περιεχομένου. Ο χαρακτήρας , σημαίνει ότι τα δύο στοιχεία αριστερά και δεξιά του κόμματος πρέπει να εμφανίζονται με την ίδια σειρά μέσα στο έγγραφο. Ενώ ο χαρακτήρας | σημαίνει ότι μόνο το ένα από τα στοιχεία πρέπει να εμφανίζεται στο έγγραφο.

Για την σχεδίαση των DTD υπάρχουν τρεις περιπτώσεις:

- Να σχεδιασθεί ένα DTD με βάση κάποιο υπάρχον μοντέλο αντικειμένων (object model). Η περιγραφή των μοντέλων δεδομένων γίνεται συνήθως μέσω της ευρέως χρησιμοποιούμενης γλώσσας μοντελοποίησης UML (Unified Modeling Language)
- Να σχεδιασθεί από την αρχή, χωρίς να βασισθούμε σε κάποιο μοντέλο αντικειμένων. Η περίπτωση αυτή είναι πιο ευέλικτη, αφού μας αφήνει το περιθώριο να ορίσουμε εμείς τα αντικείμενα και τις μεταξύ τους συσχετίσεις, προϋποθέτει όμως όπως είναι φυσικό και περισσότερη εργασία από τον προγραμματιστή
- Ενδεχομένως κάποιες εφαρμογές να απαιτούν την τροποποίηση ενός υπάρχοντος DTD, ώστε να ικανοποιεί τις ανάγκες μιας εφαρμογής. Αυτό υπαγορεύει είτε την αφαίρεση είτε την πρόσθεση στοιχείων (elements) και χαρακτηριστικών (attributes) από το υπάρχον DTD, ώστε να προκύψει το επιθυμητό

Ένα έγγραφο XML μπορεί να ανήκει σε μια από τις εξής δύο βασικές κατηγορίες:

- Έγκυρο (valid) , δηλαδή να διαθέτει ένα DTD που το συνοδεύει
- Σωστά μορφοποιημένο (well-formed) που δεν διαθέτει κάποιο συνοδευτικό DTD, αλλά υποχρεούται να εφαρμόζει όλους τους κανόνες σύνταξης της XML

Κάθε διαφορετικός τύπος έγκυρου εγγράφου XML χρησιμοποιεί το δικό του DTD. Για να είναι εφικτή η επαλήθευση της εγκυρότητας ενός XML εγγράφου, υπάρχουν τα προγράμματα XML validators , με τα οποία σαρώνονται τα tags ενός εγγράφου XML για συμμόρφωση με τους κανόνες που ορίζει το συνοδευτικό DTD. Με τα ίδια προγράμματα πρέπει να ελέγχονται και τα ίδια τα έγγραφα DTD για συνέπεια με τους κανόνες που ορίζονται στο πρότυπο DTD. Τα γνωστότερα προγράμματα validators είναι τα εξής:

- Γραμμής εντολών, όπως για παράδειγμα το XSV
- Αυτά που διαθέτουν API για τους προγραμματιστές, όπως είναι τα  
Apache Xerces  
IBM Schema Quality Checker  
Microsoft MSXML4.0
- Validators με γραφικό περιβάλλον  
XML Spy  
Turbo XML

Τέλος να αναφέρουμε ότι μια περίπτωση DTD που χρησιμοποιείται ευρύτατα σήμερα για την εμφάνιση των ιστοσελίδων είναι το DTD για την HTML 4.01. Το DTD αυτό μπορεί να βρεθεί στο URL <http://www.w3.org/TR/REC-html40/sgml/loosedtd.html>. Η HTML αποτελεί πρότυπο του W3C και η πλήρης περιγραφή της μπορεί να βρεθεί στο URL <http://www.w3.org/TR/html401/>.

### 5.1.3 Namespaces

Για την διαχείριση της επεκτασιμότητας της γλώσσας XML υπάρχουν οι χώροι ονοματοδοσίας XML (XML namespaces). Τα namespaces είναι μια σύσταση που εκδόθηκε από το W3C ([www.w3.org/TR/REC-xml-names](http://www.w3.org/TR/REC-xml-names)). Ο χώρος ονοματοδοσίας είναι ένας μηχανισμός που προσδιορίζει την προέλευση των στοιχείων της XML. Το όνομα του κάθε στοιχείου τοποθετείται σε μια ιεραρχία ονομάτων και αυτό είναι ο χώρος ονοματοδοσίας. Κάτι τέτοιο είναι ιδιαίτερα χρήσιμο σε ένα κατανεμημένο δικτυακό περιβάλλον, όπως είναι αυτό του Internet, για την αποφυγή σύγχυσης. Ο προγραμματιστής ή δημιουργός περιεχομένου είναι σε θέση να δημιουργεί δικές του ετικέτες και να τροποποιεί ένα υπάρχον XML έγγραφο, με στόχο την επέκτασή του. Για να διατηρηθεί αυτή η βασική δυνατότητα ορισμού δικών μας tags αλλά παράλληλα να ελέγχεται η επεκτασιμότητα ώστε να μην δημιουργούνται ασύμβατες επεκτάσεις του ίδιου XML εγγράφου, εισήχθη ο μηχανισμός των χώρων ονοματοδοσίας.

Κατά την εισαγωγή σε ένα έγγραφο ενός νέου tag και δημοσίευσης του τροποποιημένου εγγράφου στο Web θα πρέπει να τηρηθούν ορισμένες συμβάσεις. Ακολουθούν δύο έγγραφα XML με τις συμβάσεις χώρων ονοματοδοσίας και αμέσως μετά σχολιάζονται οι συμβάσεις αυτές.

```
<?xml version="1.0">
<references xmlns:first="http://mysiteNo1.com/default"
            xmlns:second="http://mysiteNo2.com/default"
            xmlns="http://mydefaultsite.com/default">

    <name>Stefanos</name>
    <link href=http://users.otenet.gr/~stef />

    <name>Maria</name>
    <link href=http://users.otenet.gr/~maria />

    <name>Giannis</name>
    <link href=http://users.otenet.gr/~john />

</references>
```

Το δεύτερο έγγραφο περιέχει μια λίστα με προσωπικά sites κάποιων πελατών ενός ISP

```
<?xml version="1.0">
<references xmlns:first="http://mysiteNo1.com/default"
            xmlns:second="http://mysiteNo2.com/default"
            xmlns="http://mydefaultsite.com/default">

    <name>Stefanos</name>
    <link href=http://users.otenet.gr/~stef />
    <first:eval>Great</first:eval>
    <second:eval>Accepted</second:eval>

    <name>Maria</name>
    <link href=http://users.otenet.gr/~maria />
    <first:eval>Average</first:eval>
    <second:eval>Rejected</second:eval>

    <name>Giannis</name>
    <link href=http://users.otenet.gr/~john />
    <first:eval>Average</first:eval>
    <second:eval>Accepted with merit</second:eval>
```

</references>

Στο δεύτερο έγγραφο έχουν προστεθεί κάποια tags που αφορούν στην αξιολόγηση του κάθε site. Η προσθήκη των tags αυτών έχει γίνει από διαφορετικούς φορείς και καθεμιά αξιολόγηση αφορά πιθανώς σε διαφορετικά θέματα. Συγκεκριμένα το tag <first:eval> περιέχει πληροφορία για την κατηγοριοποίηση των site με βάση ένα σχήμα αξιολόγησης. Ενώ το tag <second:eval> περιέχει πληροφορία για την κατηγοριοποίηση των site με βάση ένα διαφορετικό σχήμα αξιολόγησης. Για την διαφοροποίησή τους έχουν προστεθεί όπως φαίνεται στο παράδειγμα τα κατάλληλα προθέματα first και second. Οι συμβάσεις που πρέπει να ακολουθηθούν για αποφυγή σύγχυσης είναι:

- Το όνομα του κάθε νέου οριζόμενου tag θα ξεκινά με το πρόθεμα ακολουθούμενο από το σύμβολο : και αμέσως μετά το κυρίως όνομα του νέου στοιχείου (<second:eval>)
- Η δήλωση των χώρων ονοματοδοσίας γίνεται μέσω χαρακτηριστικών (attributes) με το πρόθεμα xmlns ακολουθούμενο από το ίδιο το πρόθεμα. Το όνομα του namespace είναι το δηλωμένο URI (Uniform Resource Identifier) και όχι το πρόθεμα. Τα URI είναι case sensitive και χρησιμοποιούνται για την απόδοση μοναδικότητας (με βάση το σύστημα DNS) σε κάθε πρόθεμα.

Η δήλωση xmlns="http://mydefaultsite.com/default" υπάρχει για να καθορίσει το προκαθορισμένο namespace για όλα τα στοιχεία του εγγράφου που δεν διαθέτουν πρόθεμα. Για κάθε έγκυρο έγγραφο XML πρέπει το πρόθεμα και τα χαρακτηριστικά της δήλωσης του namespace να δηλωθούν και στο DTD του.

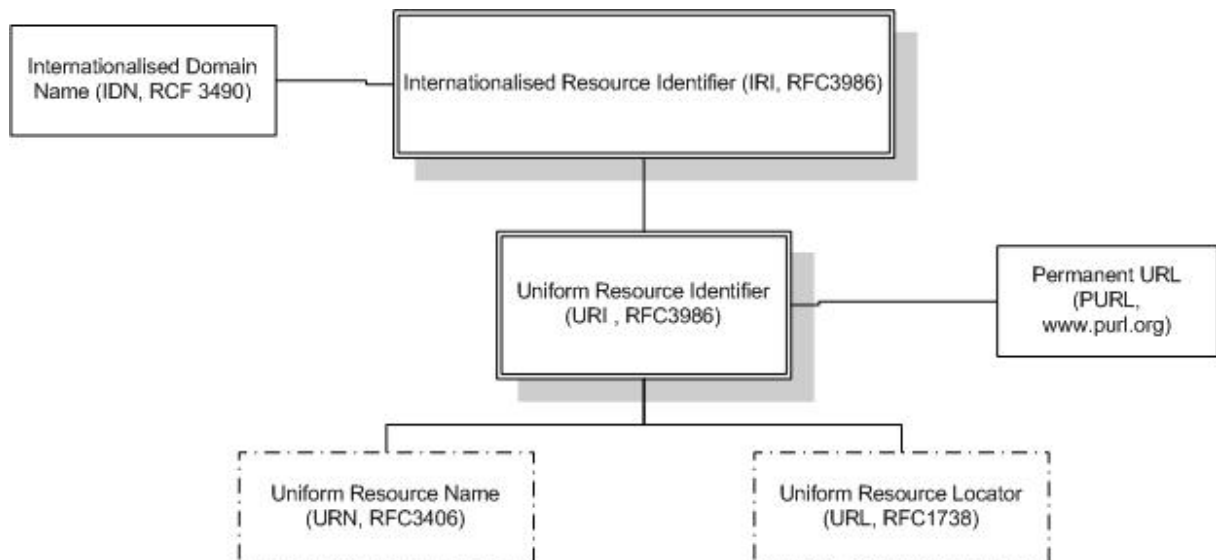
Για να είμαστε σε θέση να δημιουργήσουμε έγγραφα XML και να φτιάξουμε εφαρμογές που τα αξιοποιούν, πρέπει οπωσδήποτε να χρησιμοποιήσουμε έναν χώρο ονομάτων. Αυτό σημαίνει ότι πρέπει να κατοχυρωθεί ένα domain name. Αυτό συνήθως συνδυάζεται και με την δυνατότητα δημιουργίας web σελίδων σε κάποιο server. Ωστόσο για απλή χρήση του domain για την XML ονοματοδοσία, μπορεί κανείς να κάνει χρήση της δυνατότητας domain parking, με την οποία μπορεί να δεσμευθεί ένα συγκεκριμένο domain μόνο για τον παραπάνω λόγο, στην περίπτωση που δεν ενδιαφερόμαστε για την δημιουργία web σελίδων.

### *Άλλες τεχνολογίες XML που συνεργάζονται με τα namespaces*

Στο παρελθόν τα έγγραφα XML αναπτύσσονταν κατά κύριο λόγο απομονωμένα. Ένα DTD αναπτυσσόταν κατ' απαίτηση για μια συγκεκριμένη εφαρμογή. Τα namespaces παρακάμπτουν το πρόβλημα αυτό, επιτρέποντας την επαναχρησιμοποίηση στοιχείων, όπως τα DTD και τα στοιχεία στυλ της XML (XSLFO και CSS). Σε συνδυασμό με τα namespaces, η προδιαγραφή XML Linking Language (<http://www.w3.org/TR/xlink/>) του W3C παρέχει ένα τυποποιημένο τρόπο για την διασύνδεση των XML εγγράφων, παρόμοιο με αυτόν που χρησιμοποιεί η HTML για την διασύνδεση ιστοσελίδων. Παράλληλα με το Xlink, τεχνολογίες που είναι χρήσιμες στον χειρισμό του περιεχομένου των εγγράφων XML είναι οι τεχνολογίες XML Path Language ([www.w3.org/TR/xpath/](http://www.w3.org/TR/xpath/)) και XPointer (<http://www.w3.org/TR/xptr/>).

## URI

Τα URI (Uniform Resource Identifier) χρησιμοποιούνται στις δηλώσεις των namespaces. Τα URI (IETF RFC 2396) αποτελούν σύσταση του W3C. Είναι μια οντότητα γενικότερη από το URL. Τα URL (Uniform Resource Locator) είναι μια ειδική μορφή των URI και είναι διευθύνσεις σε συγκεκριμένους server στους οποίους βρίσκονται αυτοί οι πόροι. Συνεπώς μια μετακίνηση η διαγραφή ενός πόρου από την απόλυτη διεύθυνση του προκαλεί ένα σφάλμα στον browser και καθιστά άκυρο το URL που αντιστοιχούσε στην διεύθυνση. Υπάρχουν και τα URN (Uniform Resource Name) που είναι αντιστοιχίσεις με πραγματικούς πόρους. Επέκταση της λειτουργικότητας των URI προτείνεται με τα Internationalised Resource Identifier (IRI). Μια από τις καινοτομίες των IRI είναι ότι επιτρέπουν την δημιουργία domains με ονόματα σε οποιαδήποτε γλώσσα του κόσμου, σύμφωνα με το πρότυπο Unicode και την σύσταση του W3C Internationalised Domain Name (IDN). Τέλος να αναφέρουμε μια οντότητα ανάλογη με τα URN. Η οντότητα αυτή είναι τα PURL (permanent URL, [www.purl.org](http://www.purl.org)), τα οποία δεν καθίστανται άκυρα από την φυσική μετακίνηση ενός πόρου, αλλά ενημερώνονται κατάλληλα. Τα παραπάνω εκφράζονται σχηματικά ως εξής:



**Σχήμα 7 : Classification των URI**

### 5.1.4 Schemas

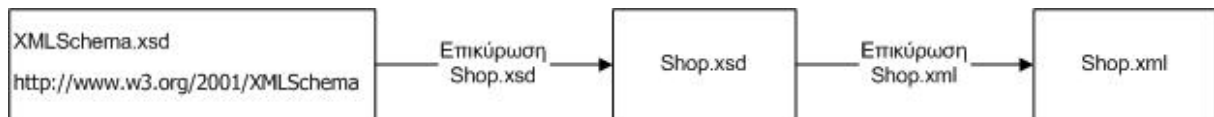
Τα DTD είναι ένα χαρακτηριστικό της XML που κληρονομήθηκε από τον πρόγονο της γλώσσας, την SGML. Για τον λόγο αυτό τα DTD δεν υποστηρίζουν ενδογενώς τους χώρους ονοματοδοσίας. Εξαιτίας του παραπάνω γεγονότος αλλά και πολλών άλλων περιορισμών που θέτει το DTD λόγω της συσχέτισής του με την SGML, το W3C εξετάζει την αντικατάσταση των DTD με τη νέα σύσταση των XML Schemas. Ένα XML Schema είναι ένα XML έγγραφο με επέκταση αρχείου .xsd το οποίο εκφράζει τους κανόνες που πρέπει να ακολουθεί ένα XML έγγραφο. Ανάλογα με την λειτουργία των DTD, κάθε καλά έγκυρο έγγραφο XML μπορεί να συνοδεύεται εναλλακτικά από ένα XML Schema.

Λόγω των πολλών πλεονεκτημάτων του XML Schema έναντι του DTD, πολλές φορές είναι χρήσιμο να μετατρέπουμε τα διαθέσιμα .dtd αρχεία σε αρχεία .xsd. Τα βασικότερα πλεονεκτήματα που προσφέρουν τα XML Schemas έναντι των DTD είναι τα εξής:

- Η γλώσσα που χρησιμοποιείται για την σύνταξη των .xsd αρχείων ακολουθεί σύνταξη ανάλογη με την σύνταξη των .xml εγγράφων. Εν αντιθέσει τα έγγραφα .dtd χρησιμοποιούν σύνταξη ασύμβατη με αυτή των .xml εγγράφων.
- Τα XML Schemas υποστηρίζουν τους χώρους ονοματοδοσίας (namespaces). Ένα namespace ορίζει το σύνολο στο οποίο ανήκουν τα ονόματα των στοιχείων σε ένα έγγραφο XML
- Υποστηρίζουν ενδογενώς περισσότερους τύπους δεδομένων από ότι τα DTD. Ενώ τα DTD υποστηρίζουν 10 τύπους δεδομένων, τα Schemas υποστηρίζουν τουλάχιστον 44 τύπους. Επίσης τα XML Schemas επιτρέπουν τον ορισμό νέων σύνθετων τύπων και έτσι αυξάνονται κατά πολύ οι δυνατότητες περιγραφής δεδομένων και κανόνων που τα διέπουν.
- Υποστηρίζουν κανονικές εκφράσεις (regular expressions). Ειδικά κατά τον ορισμό νέων σύνθετων τύπων η δυνατότητα αυτή είναι πάρα πολύ χρήσιμη.
- Τα Schemas προσφέρουν δυνατότητες συμπίκνωσης των tags (inlined element declarations), με αποτέλεσμα την μείωση της πολυπλοκότητας και την ευκολότερη σάρωση και ανάγνωση του εγγράφου κανόνων .xsd
- Στα Schemas μπορούν να οριστούν στοιχεία με μηδενικό (null) περιεχόμενο ή ακόμα και δύο ή περισσότερα στοιχεία που έχουν το ίδιο όνομα, αλλά διαφορετικό περιεχόμενο.
- Μπορούν να χρησιμοποιηθούν σε ένα έγγραφο περισσότερα του ενός Schemas, αρκεί όλα να ανήκουν στο ίδιο namespace.
- Ένα Schema με το στοιχείο <include> μπορεί να χρησιμοποιεί στοιχεία που είναι δηλωμένα σε ένα άλλο Schema και με το στοιχείο <redefine> μπορεί επίσης να επανακαθορίζει τα στοιχεία αυτά.
- Η επεκτασιμότητα ενός Schema αυξάνεται αν ενσωματωθούν στο αρχείο .xsd οι τεχνολογίες XSLT/XPath και Schematron [XFR02].
- Επιτρέπεται στα XML έγγραφα να χρησιμοποιούν πρόσθετα στοιχεία, πέραν αυτών που ορίζονται στο XML Schema, με βάση τις ανάγκες του δημιουργού περιεχομένου. Αυτή η δυνατότητα ονομάζεται Schema ανοικτού περιεχομένου (open content schema). Στην περίπτωση αυτή δηλώνονται τα στοιχεία <any> και <anyAttribute> στο .xsd αρχείο για να δηλώσουν την επεκτασιμότητα που περιγράφηκε.

#### Schema validation

Όπως και με τα DTD, η επικύρωση των Schemas αποτελείται από δύο στάδια. Πρώτα γίνεται η επικύρωση του ίδιου του εγγράφου Schema με βάση το πρότυπο XML Schema του W3C. Το πρότυπο αυτό αποτελεί ένα Schema για το συγκεκριμένο Schema που εξετάζουμε. Κάθε νέο Schema που δημοσιεύεται οφείλει να ακολουθεί το πρότυπο W3C XML Schema. Ακολουθεί η επικύρωση του .xml εγγράφου το οποίο χρησιμοποιεί το συγκεκριμένο Schema. Αυτό φαίνεται καλύτερα με το σχήμα που ακολουθεί.



**Σχήμα 8: Παράδειγμα XML Schema**

Στο παραπάνω παράδειγμα χρησιμοποιούνται τα αρχεία:

- XML Schema.xsd. Εκφράζει τους κανόνες του προτύπου XML Schema και με αυτό πρέπει να συμμορφώνονται όλα τα XML schemas. Το αρχείο αυτό παρέχεται από το W3C και μπορεί να βρεθεί στο URL <http://www.w3.org/2001/XMLSchema>. Ονομάζεται δε “XML schema for XML schemas”.
- Shop.xsd . Είναι το Schema για την έκφραση των κανόνων που πρέπει να διέπουν το έγγραφο Shop.xml
- Shop.xml . Είναι το XML έγγραφο που περιλαμβάνει το καθ’αυτό περιεχόμενο και το οποίο έχει tag που δηλώνει ποιο Schema χρησιμοποιεί (στην προκειμένη περίπτωση το «Shop.xsd»).

Ακολουθεί ο κώδικας των αρχείων Shop.xsd και Shop.xml. Στον κώδικα διακρίνονται καθαρά οι δηλώσεις των namespaces σαν attributes στην αρχή κάθε εγγράφου. Εν συνέχεια, για λόγους σύγκρισης, αντιπαραθέτουμε το περιεχόμενο ενός DTD (Shop.dtd) που εκφράζει τους ίδιους κανόνες με το XML Schema.

```
<!-- Shop.xml -->
<?xml version="1.0"?>
<Shop xmlns="http://www.mybooks.gr"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.mybooks.org/Shop.xsd">
  <Book>
    <Title>A new title</Title>
    <Author>John Adams</Author>
    <Date>July, 1998</Date>
    <ISBN>12345-67890-54321</ISBN>
    <Publisher>Stef Publishers</Publisher>
  </Book>
  ...
</Shop>
```

```
<!-- Shop.xsd -->
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://www.mybooks.org"
            xmlns="http://www.mybooks.org"
            elementFormDefault="qualified">

  <xsd:element name="Shop">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Book" minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="Book">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Title" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Author" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
```



```

    <xsd:element ref="Date" minOccurs="1" maxOccurs="1"/>
    <xsd:element ref="ISBN" minOccurs="1" maxOccurs="1"/>
    <xsd:element ref="Publisher" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>

```

```

<xsd:element name="Title" type="xsd:string"/>
<xsd:element name="Author" type="xsd:string"/>
<xsd:element name="Date" type="xsd:string"/>
<xsd:element name="ISBN" type="xsd:string"/>
<xsd:element name="Publisher" type="xsd:string"/>

```

```
</xsd:schema>
```

```

<!-- Shop.dtd -->
<!ELEMENT Shop (Book+)>
<!ELEMENT Book (Title, Author, Date, ISBN, Publisher)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Author (#PCDATA)>
<!ELEMENT Date (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
<!ELEMENT Publisher (#PCDATA)>

```

Αναφέρουμε συμπληρωματικά ότι εκτός από το XML Schema έχουν γίνει και άλλες προτάσεις από το W3C και άλλους φορείς για περιγραφή των κανόνων σύνταξης ενός XML εγγράφου. Μερικές από τις γνωστότερες προτάσεις είναι:

- RDF Schema. Η σύσταση RDF (Resource Description Framework) διαθέτει ένα δικό της πρότυπο για την σύνταξη schemas, το οποίο ονομάζεται RDF Schema. Όπως τα XML schemas που ακολουθούν το W3C XML schema είναι αρχεία με κατάληξη .xsd , έτσι τα RDF schemas που ακολουθούν το W3C RDF schema είναι αρχεία με κατάληξη .rdf
- XML-Data, που προσφέρει τύπους εμπνευσμένους από τους SQL τύπους
- Document Content Description (DCD), που αποτελεί απλοποιημένη έκδοση του XML-Data
- Schema of Object oriented XML (SOX)
- Document Definition Markup Language (DDML)

Περισσότερες πληροφορίες για την προδιαγραφή XMLSchema μπορούν να αναζητηθούν στα URL <http://www.w3.org/TR/XMLSchema> και <http://www.w3.org/XML>.

### 5.1.5 XSLT/XSLFO

Για το στυλ και την μορφοποίηση των εγγράφων XML, δηλαδή τον τρόπο παρουσίασης του εγγράφου μέσα από μια εφαρμογή, όπως ένας browser ή ένας επεξεργαστής κειμένου, υπάρχουν τεχνολογίες που καθιστούν τα έγγραφα XML πιο εύκολα παραμετροποιήσιμα σε ότι έχει να κάνει με την εμφάνισή τους.

Ένας τρόπος ελέγχου της μορφοποίησης των XML εγγράφων είναι να ενσωματώσουμε στο έγγραφο XML κάποια HTML tags, που συνεισφέρουν στην μορφοποίηση, με στόχο την επιθυμητή παρουσίαση του εγγράφου μέσα από έναν web browser. Ειδικότερα αν καταφέρουμε να έχουμε μια αυστηρή έκδοση της HTML που να υπακούει σε όλους τους κανόνες δομής της XML, τότε μπορούμε να δημιουργήσουμε την μορφοποίηση που θέλουμε, διατηρώντας όμως την απαραίτητη αυστηρή δομή της XML. Μια τέτοια προσπάθεια εκφράζεται από την εξελιγμένη γλώσσα σήμανσης, την XHTML. Η XHTML βρίσκεται στην έκδοση 1.0 (<http://www.w3.org/TR/xhtml1/>).

Για την επεξεργασία των εγγράφων XML χρησιμοποιούνται και οι τεχνολογίες XSLT (XSL Transformation, <http://www.w3.org/TR/xslt>) και XSLFO (XSL Formatting Objects), οι οποίες μαζί με την σύσταση xpath (XML Path Language) υπάγονται στην σύσταση XSL (XML Stylesheet Language) του W3C (<http://www.w3.org/Style/XSL/>). Η σύσταση XSL αφορά στο στυλ των εγγράφων XML και μας δίνει την δυνατότητα να μετατρέπουμε έγγραφα XML σε άλλα έγγραφα XML με διαφορετικό DTD ή XML Schema, όπως επίσης και σε έγγραφα διαφορετικών μορφών (HTML, text και άλλα). Ακόμα η XSL βοηθά όταν θέλουμε να δημοσιεύσουμε έναν μεγάλο όγκο από έγγραφα είτε να αναδιοργανώσουμε κάποια έγγραφα XML ώστε να δημιουργήσουμε πίνακες περιεχομένων ή άλλες χρηστικές δομές αναπαράστασης των πληροφοριών. Η σύσταση XSLT αφορά στον μετασχηματισμό ενός εγγράφου XML, ενώ η σύσταση XSLFO στην μορφοποίησή του για παρουσίαση. Την ίδια λειτουργία με την XSLFO επιτελεί και η τεχνολογία CSS (Cascading Style Sheets), ωστόσο η CSS είναι πολύ δημοφιλέστερη και μονοπωλεί το ενδιαφέρον των κατασκευαστών λογισμικού και παρόχων περιεχομένου. Ακολουθεί αμέσως μια ανάλυση των τριών βασικών τεχνολογιών με τις οποίες επιτυγχάνεται ο μετασχηματισμός και η μορφοποίηση των εγγράφων XML.

#### CSS

Η τεχνολογία CSS (Cascading Style Sheets) είναι ένας μηχανισμός κανόνων που χρησιμοποιείται για να προσδώσει συγκεκριμένο στυλ στα στοιχεία ενός εγγράφου. Με την χρήση του ενός αρχείου κανόνων CSS παρέχονται εντολές στον browser πώς να εμφανίσει ένα αρχείο XML ή HTML. Έτσι αποσυνδέονται γενικά η εμφάνιση μιας σελίδας από το περιεχόμενό της. Το CSS συστάθηκε από το W3C και έχει δύο εκδόσεις: την έκδοση CSS 1.0 και την έκδοση CSS 2.0 που υποστηρίζει την XML σε μεγαλύτερο βαθμό. Με βάση το CSS δημιουργήθηκε και η σύσταση XSLFO από το W3C, όπως θα αναφέραμε παρακάτω.

## XSLT

Τα αρχεία XSLT ονομάζονται και XSLT stylesheets και έχουν σύνταξη παρόμοια με αυτή της XML. Κάθε XSLT stylesheet είναι ένα καλά δομημένο (well formed) XML έγγραφο. Τα αρχεία XSL stylesheets βοηθούν πολύ σε κάθε προσπάθεια δημοσίευσης περιεχομένου που υποστηρίζει XML. Ένα χαρακτηριστικό παράδειγμα είναι η δημιουργία της μορφοποίησης ενός website από περιεχόμενο μιας βάσης δεδομένων βασισμένης σε XML. Στην περίπτωση αυτή ένα αρχείο με οδηγίες μορφοποίησης XSLT (αρχείο επέκτασης .xsl) βοηθά στην μετατροπή του εγγράφου .xml σε ένα έγγραφο .html. Η συνεισφορά του XSLT επεκτείνεται όταν θέλουμε να έχουμε πολλές διαφορετικές εκδόσεις του περιεχομένου ενός website, όπως για παράδειγμα αρχεία .html , αρχεία απλού κειμένου .txt (plain text), αρχεία για εμφάνιση από φορητές συσκευές μέσω WAP (WML), κ.ά. Για την αναγνώριση των XSLT stylesheets και την εκτέλεση των XML μετασχηματισμών που αυτά ορίζουν απαιτείται κάποιο πρόγραμμα XSLT processor. Τέτοια προγράμματα είναι ενσωματωμένα σε μεγαλύτερα πακέτα επεξεργασίας εγγράφων XML, όπως είναι το XML Spy. XSLT processors επίσης ενσωματώνονται και σε browsers, όπως για παράδειγμα ο Internet Explorer 6, ο οποίος διαθέτει το πρόγραμμα MSXML parser. Ωστόσο υπάρχουν και ανεξάρτητα προγράμματα XSLT processors, τα οποία μπορούν να ανακτηθούν από το Internet, όπως για παράδειγμα το Java 1.4.1 XML parser από την εταιρεία Sun ή το LotusXSL από την IBM στο URL <http://www.alphaworks.ibm.com>.

Το δομικό στοιχείο σε ένα XSLT stylesheet είναι το element `<xsl:stylesheet>` [XSL04]. Διαθέτει σαν υποστοιχεία μια σειρά από elements `<xsl:template>` ... `<xsl:template>`. Μέσα σε αυτά τα `<xsl:template>` elements εκφράζονται οι μετασχηματισμοί που πρέπει να γίνουν στο έγγραφο XML. Κάθε template περιέχει ένα attribute με όνομα match. Το match είναι μια διαδρομή για το στοιχείο του XML εγγράφου στο οποίο αναφέρεται το template. Παράλληλα με το match, σε κάθε template υπάρχει μια δήλωση `<xsl:apply-templates/>` η οποία είναι μια κλήση που πρέπει να γίνει στο stylesheet κατά την ιεραρχική σάρωση των στοιχείων του πηγαίου XML εγγράφου από τον αναλυτή.

Τα βασικά σύμβολα που χρησιμοποιούνται για την σύνταξη των διαδρομών είναι τα σύμβολα / (στοιχείο ρίζα και άμεσος απόγονος), // (όλοι οι απόγονοι), \* (όλοι οι άμεσοι απόγονοι) και | (επιλογή ενός εκ των δύο στοιχείων που βρίσκονται εκατέρωθεν του συμβόλου). Οι διαδρομές αντιστοιχίζονται τόσο σε elements όσο και σε attributes του πηγαίου XML εγγράφου. Η διαδρομή που αναφέρεται σε ένα στοιχείο url με attribute το protocol που έχει τιμή mailto γράφεται ως `match="url[@protocol='mailto']"`, δηλαδή το attribute στο οποίο γίνεται η αναφορά μπαίνει σε αγκύλες και πριν από το όνομά του τοποθετείται το σύμβολο @.

Μπορούμε να χρησιμοποιήσουμε μια από τις έτοιμες συναρτήσεις της XSL, οι οποίες παίρνουν μηδέν ή περισσότερα ορίσματα και επιστρέφουν κάτι σχετικό με το πηγαίο έγγραφο. Για παράδειγμα η συνάρτηση `text()` επιστρέφει τα περιεχόμενα ενός στοιχείου. Παράλληλα με τις έτοιμες συναρτήσεις μπορούμε να ορίσουμε και δικές μας προσθέτοντας στο XSLT stylesheet ένα στοιχείο `<xsl:functions>` ως εξής:

```
<xsl:functions ns="myns" type="text/javascript">
function today()
{
return Date().toString()
}
</xsl:functions>
```

αν θέλουμε για παράδειγμα να χρησιμοποιήσουμε Javascript. Η κλήση της παραπάνω συνάρτησης μέσα από το stylesheet γίνεται με το element `<xsl:value-of>`

```
<xsl:template match="/">
<xsl:value-of select="myns:today()"/>
</xsl:template>
```

Το πρόγραμμα parser ξεκινά την σάρωση του πηγαίου XML εγγράφου από την ρίζα προς τους τελευταίους κόμβους ιεραρχικά, με βάση έναν αλγόριθμο DFS (Depth First Search). Σε κάθε βήμα της σάρωσης, ο αναλυτής προσπαθεί να αντιστοιχίσει στον τρέχοντα κόμβο ένα από τα πρότυπα που

υπάρχουν στο XSLT stylesheet. Αν υπάρχει μια τέτοια αντιστοιχία, ο αναλυτής δημιουργεί τους κόμβους που του λείει το stylesheet στο παραγόμενο δέντρο (παραγόμενο XML έγγραφο). Όταν ο αναλυτής συναντήσει το στοιχείο `<xsl:apply-templates/>`, τότε προχωρά στα παιδιά του τρέχοντος κόμβου και επαναλαμβάνει την διαδικασία, δηλαδή προσπαθεί να τα αντιστοιχίσει με κάποιο πρότυπο. Αν ο αναλυτής δεν βρει καμία αντιστοιχία για ένα στοιχείο τότε εφαρμόζει το ενσωματωμένο πρότυπο του αναλυτή. Το ενσωματωμένο πρότυπο περιέχει συνήθως μόνο μια κλήση `<xsl:apply-templates/>` με χαρακτηριστικό `match="* | /"` που σημαίνει ότι ο αναλυτής οδηγείται στα παιδιά του στοιχείου για το οποίο δεν είχε αρχικά βρει αντιστοιχία στο XSLT stylesheet. Έτσι η διαδικασία δεν τερματίζεται, παρά μόνο όταν ολοκληρωθεί η σάρωση όλου του πηγαίου XML εγγράφου. Επίσης κάποια στοιχεία έχουν σαν μοναδικό παιδί τους ένα κείμενο και για το κείμενο αυτό δεν έχει οριστεί κάποιο template στο XSLT. Τότε πάλι εκτελείται ένα ενσωματωμένο template του αναλυτή που υπαγορεύει την παρουσίαση του κειμένου στο τρέχον σημείο του παραγόμενου δέντρου.

Τα attributes που ορίζονται στο παραγόμενο XML δέντρο είναι είτε ρητά δηλωμένα στο XSLT είτε υπολογίζονται κατά την δημιουργία του παραγόμενου δέντρου. Η δεύτερη περίπτωση γίνεται με την δήλωση `<xsl:attribute name="..."> .... </xsl:attribute>` για τον υπολογισμό της τιμής του attribute name κατά την δημιουργία του παραγόμενου δέντρου.

Για την δημιουργία νέων αντικειμένων υπάρχουν μια σειρά από εντολές XSLT. Για παράδειγμα υπάρχουν οι εντολές `<xsl:element>` και `<xsl:attribute>` για την δημιουργία ενός στοιχείου ή ενός χαρακτηριστικού στο παραγόμενο δέντρο και υπολογισμού της τιμής του κατά την δημιουργία του δέντρου.

Όπως έχουμε πει υπάρχουν δύο τρόποι για να παράξουμε το τελικό επιθυμητό HTML έγγραφο. Είτε να εφαρμόσουμε εμείς εσωτερικά στον αναλυτή μας το XSLT και να παράξουμε το HTML είτε να δώσουμε στον τελικό χρήστη το πηγαίο XML έγγραφο και το XSLT και ο μετασχηματισμός να εκτελεστεί στην πλατφόρμα του χρήστη. Στην δεύτερη αυτή περίπτωση πρέπει να είμαστε προσεκτικοί με θέματα συμβατότητας ώστε το XSLT μας να καλύπτει τις προδιαγραφές του αναλυτή του χρήστη (π.χ. ένας browser).

Σχετικά με την προτεραιότητα εφαρμογής των προτύπων του XSLT, ισχύουν τα εξής:

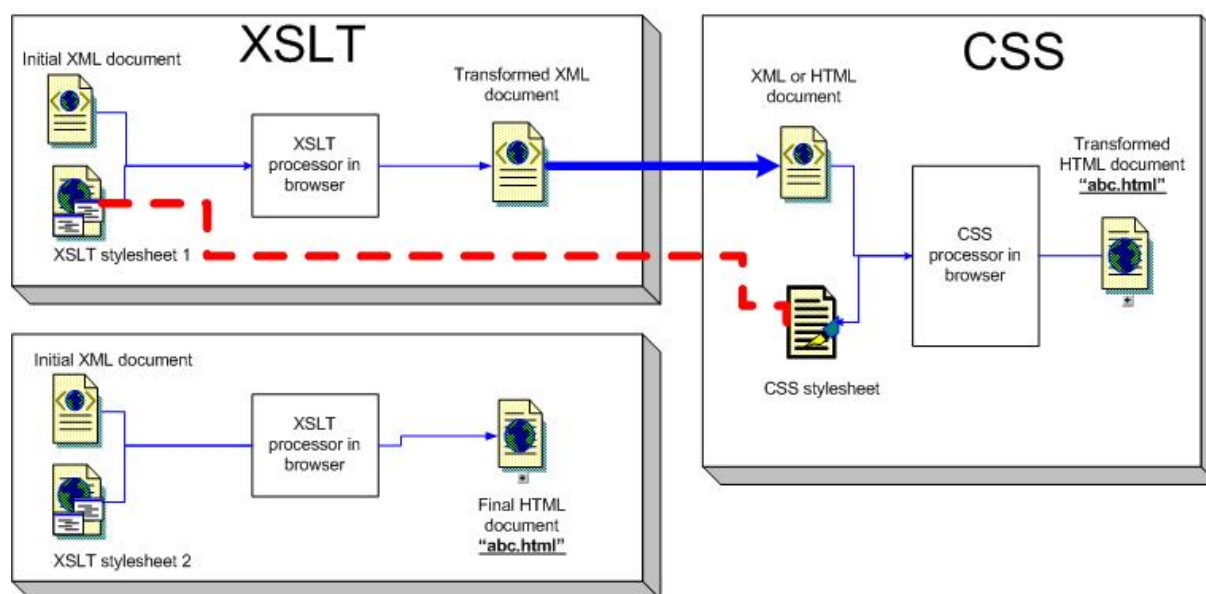
Τα πρότυπα με τις πιο συγκεκριμένες διαδρομές προηγούνται αυτών που είναι λιγότερο συγκεκριμένα. Αν αυτό το κριτήριο δεν αρκεί για την επιλογή του επόμενου template, τότε μεγαλύτερη προτεραιότητα έχει το template που βρίσκεται τελευταίο στο XSLT stylesheet.

Όταν θέλουμε να επιβάλλουμε σε ένα XML έγγραφο με ένα αρχικό XML schema να ακολουθεί ένα άλλο XML schema, τότε πρέπει να δημιουργήσουμε ένα XSLT stylesheet για μετατροπή μεταξύ των δύο schemas και εν συνεχεία να εκτελέσουμε αυτό το XSLT stylesheet μέσα από έναν browser ή έναν XSLT αναλυτή. Εκεί θα πρέπει στην κορυφή του εγγράφου να γίνει μια δήλωση για αναφορά στο XSLT που χρησιμοποιούμε. Η δήλωση αυτή είναι της μορφής `<?xml-stylesheet href="..." type="text/xsl"?>`.

Ένα XSLT stylesheet μπορεί επίσης να περιέχει μια αναφορά σε ένα CSS stylesheet για την τελική μορφοποίηση του παραγόμενου εγγράφου XML. Έτσι με την χρήση των templates στα XSLT stylesheets έχουμε δύο επιλογές:

- Να χρησιμοποιήσουμε XSLT, το οποίο εφαρμόζεται πάνω στο επιθυμητό XML έγγραφο και να παράγουμε το τελικό HTML (XHTML) έγγραφο.
- Να χρησιμοποιήσουμε XSLT που έχει αναφορά σε κάποιο CSS και να παράγουμε ένα XML έγγραφο. Το τελικό αυτό XML έγγραφο μορφοποιείται με βάση τις οδηγίες μορφοποίησης του CSS στο οποίο υπάρχει αναφορά.

Τα παραπάνω παρουσιάζονται και σχηματικά ως εξής:



Σχήμα 9: Σύγκριση των XSLT και CSS

Ειδικότερα μπορούμε από το αρχικό περιεχόμενο που διαθέτουμε να δημοσιεύσουμε μια ειδική έκδοση εγγράφων XML που να μπορούν αυτόματα να διαβαστούν από άλλα website ή συστήματα διαχείρισης περιεχομένου. Τα XML έγγραφα αυτά θα πρέπει να υποστηρίζουν κάποιο γνωστό schema για syndication, όπως είναι το RSS ή το Atom. Περισσότερα για το syndication μπορεί να βρει ο αναγνώστης στο Κεφάλαιο 5.

### XSLFO

Τα XSLFO (XSL Formatting Objects) είναι αρχεία XSL που όμως έχουν διαφορετικά tags μέσα σε κάθε <xsl:template> element από αυτά που υπάρχουν μέσα σε ένα XSLT stylesheet. Τα νέα tags που χρησιμοποιούνται στα XSLFO βασίζονται στην λογική των CSS stylesheets. Κανένας από τους γνωστούς browser δεν υποστηρίζει το XSLFO αλλά οι περισσότεροι υποστηρίζουν πλέον εναλλακτικά το XSLT και το CSS2. Η συντριπτική πλειοψηφία των εγγράφων βασίζονται στην HTML και την XHTML για την εμφάνισή τους στο Web. Με αυτή τη λογική το XSLFO μπορεί να αποτελεί μια τελείως νέα λογική, αλλά αυτό που χρησιμοποιείται στην πράξη και θα συνεχίσει να υποστηρίζεται στο μέλλον είναι ο συνδυασμός XSLT και XHTML, ως εξέλιξη της απλής HTML.

### 5.1.6 DOM/SAX

Για την ανάγνωση και επεξεργασία των εγγράφων XML μέσα από μια εφαρμογή έχουν αναπτυχθεί δύο αρκετά διαδεδομένα API (Application Program Interfaces). Το ένα βασίζεται σε μια αντικειμενοστρεφή λογική και ονομάζεται DOM (Document Object Model). Το DOM έχει συσταθεί και υποστηρίζεται σαν πρότυπο από το W3C (World Wide Web Consortium). Διαθέτει επίσης ευρεία υποστήριξη από πολλά προγράμματα φυλλομέτρησης του Ιστού (web browsers). Το DOM εκτός από ανάγνωση προσφέρει δυνατότητες και για τροποποίηση υπαρχόντων ή για δημιουργία νέων XML εγγράφων. Το δεύτερο API ονομάζεται SAX (Simple API for XML). Το SAX βασίζεται σε μια λογική συμβάντων (events) και χειριστών συμβάντων (event handlers) και παρέχει στον προγραμματιστή λειτουργίες χαμηλότερου επιπέδου. Το SAX, σε αντίθεση με το DOM, δεν υποστηρίζεται από καμία επίσημη τυποποίηση, αλλά χρησιμοποιείται ευρύτατα και θεωρείται ως μια de facto τυποποίηση. Παρακάτω αναλύουμε περαιτέρω τα χαρακτηριστικά του DOM API, διότι αυτό είναι το δημοφιλέστερο από το δύο διαθέσιμα XML API.

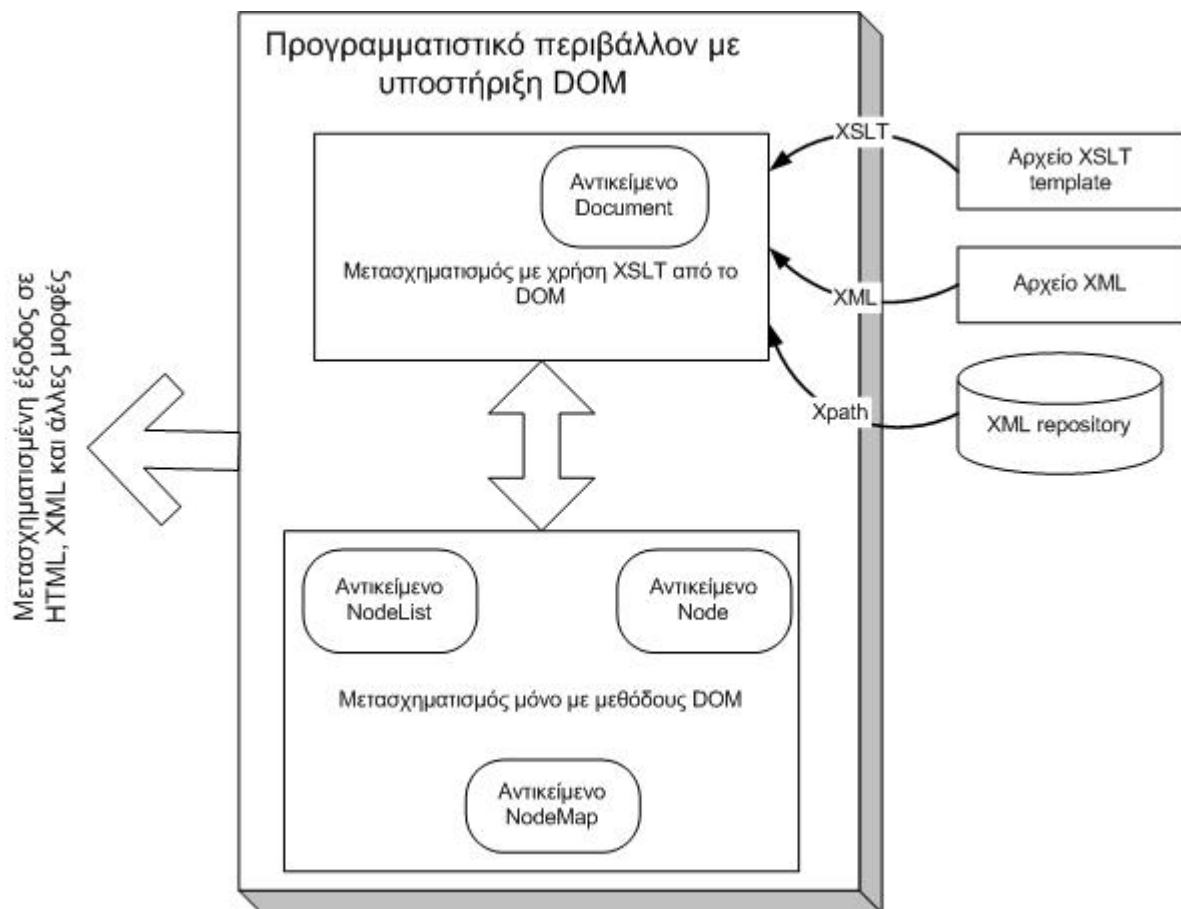
Το DOM είναι αντικειμενοστρεφές και μπορεί να ενσωματωθεί σε ένα μεγάλο αριθμό από προγραμματιστικά περιβάλλοντα τόσο με γλώσσες προγραμματισμού όσο και με γλώσσες script. Το DOM αποτελείται από ένα σύνολο αντικειμένων που διαθέτουν ιδιότητες και μεθόδους για επεξεργασία των εγγράφων XML. Τα βασικά αντικείμενα που διαθέτει το DOM είναι τα ακόλουθα:

- Document: Μας επιτρέπει να διαβάζουμε αρχεία XML, να εκτελούμε αναζητήσεις μέσα σε έγγραφα XML και να μετασχηματίζουμε έγγραφα χρησιμοποιώντας XSLT templates.
- Node: Μας επιτρέπει να χειριζόμαστε μεμονωμένα το υποδέντρο ενός κόμβου μέσα στην ιεραρχία ενός εγγράφου XML και να επεξεργαζόμαστε, ανανεώνουμε, προσθέτουμε, διαγράφουμε και να αναζητούμε υποκόμβους και πόρους αυτών.
- Node List: Αυτό το αντικείμενο είναι μια συλλογή (collection) από αντικείμενα τύπου Node.
- Node Map: Το αντικείμενο αυτό χειρίζεται τα attributes για μια συλλογή από κόμβους.

Η επεξεργασία ενός εγγράφου XML από το DOM περιλαμβάνει τυπικά τρία στάδια, τα οποία είναι τα ακόλουθα:

- Εισαγωγή: Αρχικά εισάγουμε το περιεχόμενο XML που θέλουμε να χειριστούμε σε ένα αντικείμενο Document. Το περιεχόμενο αυτό μπορεί να βρίσκεται είτε σε αρχεία XML είτε σε αρχεία που περιέχουν ένα text string, που όμως είναι καλά δομημένο XML περιεχόμενο (well formed). Παράλληλα με τα έγγραφα XML εισάγονται και τυχόν υπάρχοντα XSLT έγγραφα, τα οποία έχουν την ίδια σύνταξη με τα XML έγγραφα. Επίσης το περιεχόμενο μπορεί να ανακτηθεί από ένα XML repository ή μια βάση δεδομένων, χρησιμοποιώντας το XPath.
- Επεξεργασία: Εδώ έχουμε δύο επιλογές. Η πρώτη επιλογή είναι να δημιουργήσουμε ένα αντικείμενο Document για το έγγραφο XML και ένα άλλο αντικείμενο Document για το XSLT έγγραφο. Μετά με εντολές του DOM εφαρμόζουμε τον XSLT μετασχηματισμό. Η δεύτερη επιλογή είναι να φορτώσουμε μόνο ένα αντικείμενο Document για το έγγραφο XML και μετά να το χειριστούμε αποκλειστικά με μεθόδους επεξεργασίας που προσφέρει το DOM, χωρίς την χρήση του XSLT. Συμπεραίνουμε ότι αν θέλουμε να κάνουμε μετατροπή στο Schema που χρησιμοποιεί μια κατηγορία XML εγγράφων, μπορούμε είτε να χρησιμοποιήσουμε XSLT είτε να τον δικό μας κώδικα DOM μέσα από το οικείο μας προγραμματιστικό περιβάλλον.
- Εξαγωγή: Η έξοδος μπορεί να είναι είτε HTML, XML (για μετατροπή από ένα XML schema σε ένα άλλο) είτε μια σχεσιακή βάση. Στην περίπτωση της βάσης δεδομένων γίνεται αντιστοίχιση από το XML περιεχόμενο σε εγγραφές και πεδία της βάσης και κατόπιν εισαγωγή του περιεχομένου στην βάση. Πέραν των παραπάνω μορφών εξόδου μπορούν να παραχθούν βεβαίως και άλλες μορφές, ανάλογα με τις εκάστοτε ανάγκες.

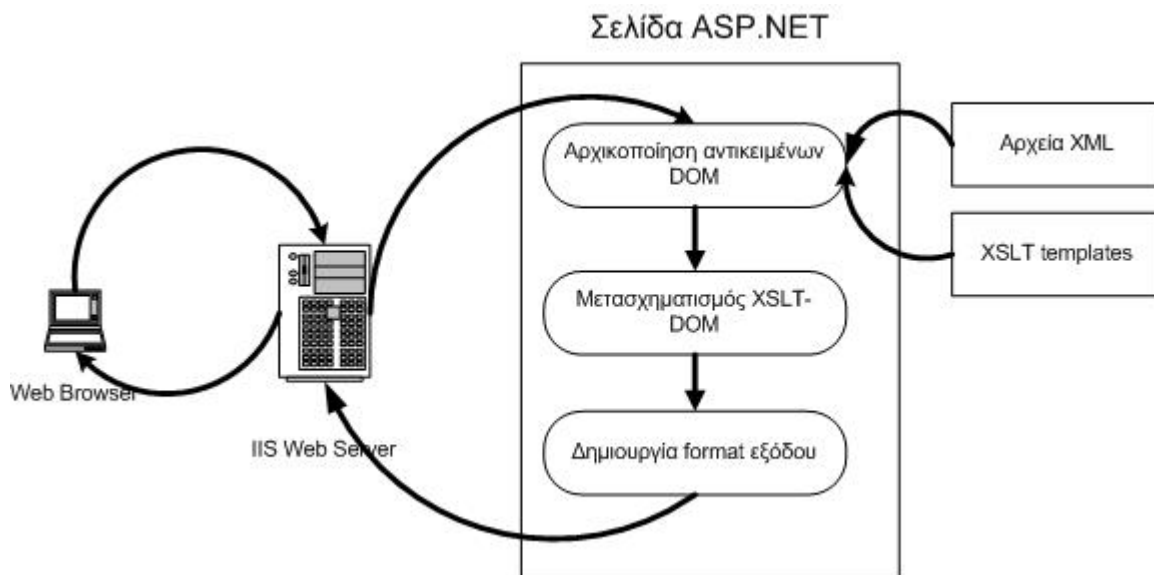
Τα παραπάνω τρία στάδια της επεξεργασίας εγγράφων XML από το DOM API φαίνονται στο παρακάτω σχήμα:



Σχήμα 10: Επεξεργασία DOM με και χωρίς XSLT

Το DOM ακολουθεί επίσης τις αρχές του δικτυακού προγραμματισμού client-server και συνεπώς διαθέτει συστατικά στοιχεία και στον client (browser) και στον server (για παράδειγμα ενσωμάτωση σε ASP ή JSP). Πιο συγκεκριμένα:

- Server: Κάθε ενέργεια που γίνεται σε έναν server είναι ένας XML μετασχηματισμός. Για τον μετασχηματισμό αυτόν χρησιμοποιείται το DOM API με εισόδους τα απαιτούμενα XML και XSLT έγγραφα καθώς και είσοδο του τελικού χρήστη μέσω φορμών. Η διαδικασία αυτή για την περίπτωση ενός περιβάλλοντος που χρησιμοποιεί εργαλεία Microsoft περιγράφεται ικανοποιητικά από το παρακάτω σχήμα.



**Σχήμα 11: Επεξεργασία DOM στην πλευρά του Microsoft IIS server**

- Client: Αν και στα μεγάλα συστήματα CMS ο περισσότερος κώδικας XML βρίσκεται στον server, μπορούμε σε πολλές περιπτώσεις να δουλέψουμε εναλλακτικά με επεξεργασία XML και στον client. Έχουμε τις εξής δύο επιλογές:  
Μπορούμε να βασιστούμε στην αυτοματοποιημένη διαδικασία που προσφέρει ο browser για επεξεργασία των εγγράφων XML. Για παράδειγμα ο Mozilla Firefox όταν του ζητηθεί ένα έγγραφο XML ελέγχει αν αυτό διαθέτει αναφορά σε κάποιο XSLT. Αν ναι τότε προβάλλεται αυτόματα η HTML σελίδα μετά την μετατροπή. Αν δεν υπάρχει αναφορά σε XSLT, τότε προβάλλεται το XML έγγραφο με δενδρική δομή.  
Εκτός από τις ενσωματωμένες δυνατότητες του προγράμματος περιήγησης μπορούμε να γράψουμε τα δικά μας προγράμματα script (για παράδειγμα Javascript) για να φορτώσουμε τα XML και XSLT και μετά να εκτελέσουμε εντολές DOM χειρωνακτικά. Αυτό προσφέρει μεγαλύτερη ευελιξία και δυνατότητες παραμετροποίησης της τελικής εμφάνισης των εγγράφων μας.

Τα προγράμματα που χρησιμοποιούνται για την ανάγνωση των XML εγγράφων και που συνεργάζονται με τα δύο βασικά XML API είναι οι αναλυτές (parsers). Ορισμένοι από αυτούς υποστηρίζουν ένα από τα δύο διαθέσιμα API, ενώ άλλοι υποστηρίζουν αμφότερα. Από την άλλη πλευρά τα προγράμματα που χρησιμοποιούνται για την τροποποίηση ή δημιουργία XML εγγράφων ονομάζονται γεννήτριες (XML generators). Στην πράξη ωστόσο πολλοί αναλυτές έχουν και ενσωματωμένες δυνατότητες τροποποίησης και δημιουργίας εγγράφων. Ένας γνωστός XML parser που διανέμεται στο Διαδίκτυο είναι από την εταιρεία IBM και είναι διαθέσιμος στο URL <http://www.alphaworks.ibm.com>.



# 6 *Web Services*

## 6.1 *Εισαγωγή στις Web Services*

### 6.1.1 *Τι είναι μια διαδικτυακή υπηρεσία (Web Service)*

Η έννοια «υπηρεσία λογισμικού» (software service) περιγράφει μια οντότητα λογισμικού που προσφέρει λειτουργικότητα δεχόμενη αιτήσεις και αποστέλλοντας αντίστοιχες αποκρίσεις. Με βάση αυτή τη θεώρηση, ως υπηρεσίες λογισμικού μπορούν να θεωρηθούν οι διαδικασίες (procedures) σε μια διαδικαστική γλώσσα προγραμματισμού, ένα αντικείμενο σε Java ή μια αποθηκευμένη διαδικασία SQL. Με την έννοια υπηρεσία λογισμικού συνήθως όμως αναφερόμαστε σε προγράμματα που προσφέρουν λειτουργίες χαμηλού επιπέδου, τις οποίες χρειάζονται άλλες εφαρμογές. Παράδειγμα είναι οι δαίμονες (daemons) του λειτουργικού συστήματος Linux.

Μια Διαδικτυακή υπηρεσία είναι μια υπηρεσία λογισμικού, η οποία μπορεί να κληθεί από μια εφαρμογή, ασχέτως μηχανήματος, λειτουργικού συστήματος ή γλώσσας προγραμματισμού που χρησιμοποιεί, διαμέσου των ευρέως χρησιμοποιούμενων πρωτοκόλλων του Web, όπως είναι το HTTP και η XML. Η διαφορά των Διαδικτυακών Υπηρεσιών σε σχέση με όλες τις υπόλοιπες υπηρεσίες λογισμικού είναι ότι οι πρώτες είναι ανεξάρτητες από το μηχάνημα στο οποίο βρίσκονται (λογισμικό υποδομής και φυσική τοποθεσία) και την τεχνολογία με την οποία έχουν υλοποιηθεί.

#### *Σύγκριση με τις εφαρμογές ASP*

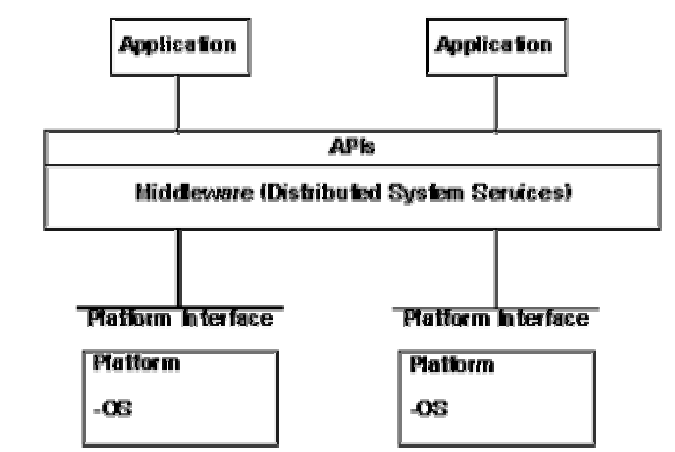
Για αποφυγή σύγχυσης, συγκρίνουμε τις Διαδικτυακές Υπηρεσίες με την έννοια των παρόχων υπηρεσιών λογισμικού (Application Service Providers ASP). Παρότι και οι Web Services και οι ASP εφαρμόζουν την έννοια του «λογισμικό ως υπηρεσία», ωστόσο διαφέρουν ριζικά. Οι ASP παραδίδουν ολόκληρες εφαρμογές από μια κεντρική θέση φιλοξενίας, ενώ οι web services περιέχουν καταναεμημένα συστατικά. Επίσης οι ASP εφαρμογές εξαρτώνται συνήθως από την υποδομή λογισμικού του μηχανήματος υποδοχής της υπηρεσίας, κάτι που δεν συμβαίνει με τις Web Services.

### 6.1.2 *Σύντομη ιστορική αναδρομή των υπηρεσιών λογισμικού, εξέλιξη των οποίων αποτελούν οι Διαδικτυακές Υπηρεσίες.*

Οι πρώτες υπηρεσίες λογισμικού εμφανίστηκαν με τις συμβολικές γλώσσες προγραμματισμού (assembly languages). Στην συνέχεια δημιουργήθηκαν γλώσσες υψηλότερου επιπέδου, όπως η Fortran και η COBOL. Μετά έκανε την εμφάνισή του το λειτουργικό σύστημα UNIX μαζί με την γλώσσα C και τα πρωτόκολλα TCP/IP. Η γλώσσα C, σε συνδυασμό με τις τεχνολογίες Remote Procedure Call (RPC) και eXternal Data Representation (XDR), έδωσε τη δυνατότητα δημιουργίας δικτυακών εφαρμογών για την επικοινωνία διεργασιών που έτρεχαν σε διαφορετικά μηχανήματα. Αμέσως μετά ακολουθεί η εποχή των γλωσσών λογικού προγραμματισμού, όπως για παράδειγμα η Lisp και εν συνεχεία γίνεται μετάβαση στον αντικειμενοστρεφή προγραμματισμό (Object Oriented Programming) με γλώσσες όπως η C++ και η Java.

Παράλληλα εμφανίστηκαν πρωτόκολλα για την επικοινωνία καταναεμημένων δικτυακών συστημάτων. Τα πρωτόκολλα αυτά ανήκουν στην κατηγορία του λογισμικού middleware που αναπτύχθηκε κυρίως κατά την διάρκεια της δεκαετίας του '90. Το λογισμικό middleware αναφέρεται σε μια σειρά από υπηρεσίες που επιτρέπουν την επικοινωνία διεργασιών που τρέχουν σε διαφορετικά μηχανήματα. Τα συστήματα middleware ήταν και είναι πολύ χρήσιμα σε περιβάλλοντα όπου συνυπάρχουν ετερογενείς πλατφόρμες, όπως για παράδειγμα συστήματα mainframe παράλληλα με client-server.

Πρωτοβουλίες που αναπτύχθηκαν για το middleware είναι οι Distributed Computing Environment (DCE) του Open Software Foundation , το Common Object Request Broker Architecture (CORBA) του Object Management Group, και το Microsoft Distributed Common Object Model (DCOM). Τα εργαλεία και οι εφαρμογές middleware δουλεύουν με βάση την αρχιτεκτονική που απεικονίζεται στην παρακάτω εικόνα:



**Σχήμα 12: Αρχιτεκτονική middleware**

Πέραν των πλεονεκτημάτων που έχουν να κάνουν με την διαλειτουργικότητα μεταξύ των μηχανημάτων, το middleware έχει και κάποια μειονεκτήματα. Μερικά από τα πιο σημαντικά είναι τα εξής:

- Δημοφιλή συστήματα και υπηρεσίες middleware χρησιμοποιούν τμήματα ιδιόκτητου λογισμικού και έτσι καθιστούν πολλές εφαρμογές middleware εξαρτημένες από έναν συγκεκριμένο κατασκευαστή.
- Τα πρωτόκολλα CORBA και DCOM είναι ασύμβατα μεταξύ τους, με αποτέλεσμα να απαιτείται κάποιου είδους γέφυρα στην διασύνδεση δικτύων με διαφορετική middleware αρχιτεκτονική.
- Παρόλο που τα middleware εισάγουν ένα στρώμα αφαίρεσης για τον προγραμματισμό κατακεντρωμένων δικτυακών εφαρμογών, ωστόσο δεν απαλλάσσουν τον προγραμματιστή από την επιφόρτιση δύσκολων επιλογών σχεδίασης , όπως το ποια συστατικά κώδικα να τοποθετήσει στον client και ποια στον server.
- Τα μηνύματα που στέλνουν τα προγράμματα middleware έχουν δυαδική κωδικοποίηση και δεν μπορούν να αποδώσουν στην διακινούμενη πληροφορία κάποιο νόημα για τις μηχανές που την διαχειρίζονται. Αυτό συμβάλλει στην εκτεταμένη συμβατότητα μεταξύ κάποιων συστημάτων.

Για την υπέρβαση των περιορισμών που θέτουν οι τεχνολογίες middleware, έπρεπε να γίνει επένδυση σε περισσότερο διαλειτουργικά πρότυπα. Έπρεπε να βρεθεί ένας τρόπος να μπορεί να γίνεται από κοινού χρήση δεδομένων και εφαρμογών σε διαδίκτυα. Για την από κοινού χρήση των δεδομένων προτάθηκε η γλώσσα σήμανσης Extensible Markup Language (XML) και η χρήση των μεταδεδομένων (metadata). Για την συνεργασία των εφαρμογών έχει προταθεί μια σειρά τεχνολογιών με το όνομα Διαδικτυακές Υπηρεσίες (Web Services).

Οι Διαδικτυακές Υπηρεσίες βασίζονται στις τεχνολογίες SOAP (επίσημα Simple Object Access Protocol, WSDL (Web Services Description Language) και UDDI (Universal Description, Discovery and Integration). Στο πρωτόκολλο SOAP συχνά αποδίδεται ανεπίσημα και ο τίτλος Service-Oriented Architecture Protocol.

## **6.2 Τεχνολογίες των *Web Services***

Μια Διαδικτυακή Υπηρεσία (*Web Service*) είναι ένα μια υπηρεσία λογισμικού που αναγνωρίζεται από ένα URI και το περιβάλλον διεπαφής (*interface*) της καθώς και οι δράσεις της περιγράφονται σε XML μορφή. Η *Web Service* μπορεί να βρεθεί και να χρησιμοποιηθεί εύκολα από άλλα συστήματα. Αυτά τα συστήματα μπορούν να αλληλεπιδρούν με την *Web Service* χρησιμοποιώντας υποχρεωτικά τεχνολογίες του Internet και αλλάζοντας πληροφορίες σε μορφή XML. Οι *Web Services* βασίζονται σε μια σειρά από XML τεχνολογίες, οι οποίες αναλύονται αμέσως παρακάτω.

### 6.2.1 Πρωτόκολλο SOAP

Το πρωτόκολλο SOAP είναι το νέο standard για την επικοινωνία μεταξύ Διαδικτυακών Υπηρεσιών και άλλων εφαρμογών με κατάλληλα μηνύματα. Το SOAP αναπτύχθηκε αρχικά το 1999 πριν ακόμα από την εμφάνιση της τεχνολογίας των Διαδικτυακών Υπηρεσιών ως μια απάντηση στην ανάγκη να υπάρξει ένα πρωτόκολλο επικοινωνίας των εφαρμογών διαμέσου του Διαδικτύου με τρόπο ανεξάρτητο της γλώσσας και της πλατφόρμας κάθε εφαρμογής. Αυτή τη στιγμή το SOAP βρίσκεται στην έκδοση 1.2 και η ανάπτυξή του είναι υπό εξέλιξη από ομάδα εργασίας του W3C.

Το SOAP δεν χρησιμοποιείται αποκλειστικά για τις Διαδικτυακές Υπηρεσίες, αυτή όμως είναι η κύρια και πιο διαδεδομένη χρήση του. Χρησιμοποιείται για την αποστολή μηνυμάτων που περιγράφουν προγραμματιστικές ενέργειες για την επικοινωνία μεταξύ Διαδικτυακών Υπηρεσιών και εφαρμογών, με τρόπο ανεξάρτητο από την γλώσσα, το λειτουργικό σύστημα και το μηχάνημα στα οποία τρέχουν οι κατανεμημένες εφαρμογές.

Όλα τα παραπάνω προϋποθέτουν την ύπαρξη ενός μοντέλου σύμφωνα με το οποίο γίνεται η ανταλλαγή μηνυμάτων. Σύμφωνα με το μοντέλο αυτό, ένα κατανεμημένο δικτυακό περιβάλλον αποτελείται από ανεξάρτητους κόμβους. Οι κόμβοι αυτοί μπορεί να είναι εφαρμογές ή διεργασίες ή Διαδικτυακές Υπηρεσίες που ανταλλάσσουν μεταξύ τους μηνύματα SOAP.

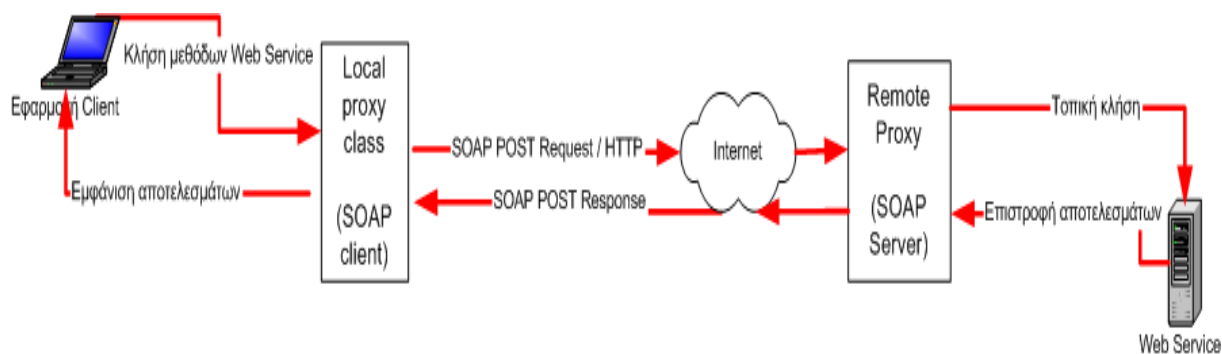
Κάθε μήνυμα SOAP αποτελείται από:

- Μια επικεφαλίδα HTTP (HTTP Header). Στην επικεφαλίδα βρίσκονται πληροφορίες δρομολόγησης για το μήνυμα και ή πληροφορίες για συναλλαγές (transactions) που λαμβάνουν χώρα στον εκάστοτε κόμβο. Συγκεκριμένα υπάρχει το πεδίο SOAPAction, το οποίο ελέγχεται από τον κόμβο που παραλαμβάνει το μήνυμα SOAP. Το πεδίο αυτό λαμβάνει συνήθως ως τιμή το όνομα της μεθόδου της Web Service που θέλουμε να καλέσουμε.
- Ένα έγγραφο XML με κορυφαίο στην ιεραρχία το στοιχείο <envelope>. Στο στοιχείο <envelope> περιέχεται ένα προαιρετικό υποστοιχείο επικεφαλίδα (header) και ένα υποχρεωτικό υποστοιχείο σώμα (body). Στο στοιχείο body περιέχονται τα δεδομένα ή η περιγραφή για τις εντολές που πρέπει να εκτελεστούν από την εφαρμογή που λαμβάνει το SOAP μήνυμα. Το XML έγγραφο επίσης περιλαμβάνει δήλωση για τον χώρο ονομάτων και το XML Schema που χρησιμοποιείται για την μορφοποίηση του περιεχομένου.

Το πρωτόκολλο SOAP περιλαμβάνει επίσης ένα σύνολο από κανόνες για το πώς θα γίνεται η κωδικοποίηση κάθε διαφορετικού τύπου δεδομένων που αποστέλλεται σε μια εφαρμογή. Η ενσωμάτωση αυτών των κανόνων στο μήνυμα SOAP γίνεται με το χαρακτηριστικό encodingStyle, το οποίο παραπέμπει σε ένα URI με τους κανόνες κωδικοποίησης. Υποστηρίζονται τόσο απλοί όσο και σύνθετοι τύποι δεδομένων.

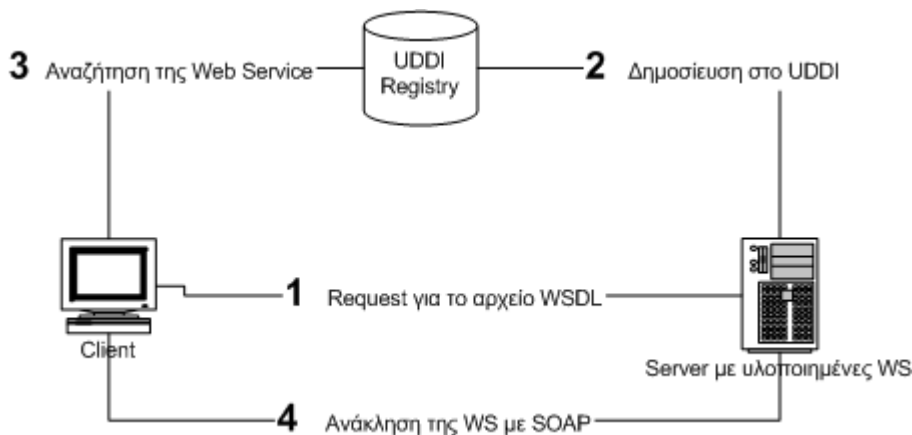
Τα μηνύματα SOAP υλοποιούν κλήσεις RPC (Remote Procedure Call). Πιο συγκεκριμένα, για την κλήση μιας μεθόδου ενός απομακρυσμένου αντικειμένου μιας Διαδικτυακής Υπηρεσίας, στέλνεται ένα μήνυμα SOAP που περιλαμβάνει στο σώμα του το όνομα της μεθόδου, τις παραμέτρους που αυτή πρέπει να δεχθεί και το URI της. Μετά την εκτέλεσή της μεθόδου από την απομακρυσμένη Υπηρεσία, τα αποτελέσματα επιστρέφονται στο καλούν πρόγραμμα μέσω ενός άλλου μηνύματος SOAP.

Η όλη επικοινωνία πραγματοποιείται μεταξύ του πελάτη (SOAP client) και του διακομιστή (SOAP server), οι οποίοι είναι τμήματα προγράμματος που αναλαμβάνουν την μετατροπή από την μορφή που είναι κατανοητή στην τοπική εφαρμογή ή υπηρεσία αντίστοιχα σε μορφή SOAP XML envelope και το αντίστροφο. Λειτουργούν έτσι ως μεσάζοντες (proxy) μεταξύ των δύο επικοινωνούντων μηχανών. Για να αντιληφθούμε καλύτερα την παραπάνω αμφίδρομη διαδικασία, μπορούμε να παρατηρήσουμε το ακόλουθο σχήμα.



**Σχήμα 13: Κλήση των Web Services**

Μια πρόσθετη χρήσιμη λειτουργία του SOAP server είναι ότι παράγει το αρχείο περιγραφής της Διαδικτυακής Υπηρεσίας (WSDL) και το δημοσιεύει σε κάποιο ειδικό μητρώο στο Διαδίκτυο (UDDI Registry). Όταν η εφαρμογή client θέλει να καλέσει μια Web Service, απλά αναζητά στο UDDI registry την υπηρεσία που επιθυμεί, ανακτά το αρχείο WSDL και με βάση αυτό διαμορφώνει κατάλληλα το SOAP μήνυμα που πρέπει να αποσταλεί στην υπηρεσία. Αυτό γίνεται περισσότερο κατανοητό από το παρακάτω σχήμα. Οι αριθμοί συμβολίζουν την χρονική σειρά των βημάτων που ακολουθούνται.



**Σχήμα 14: Αναζήτηση σε UDDI**

Για την μετάδοση των SOAP μηνυμάτων μπορεί να γίνει χρήση πολλών πρωτοκόλλων, π.χ. SMTP ή HTTPS ή JMS (Java Messaging Service), αρκεί τα πρωτόκολλα αυτά να υποστηρίζονται και από την καλούσα και από την καλούμενη εφαρμογή. Το πιο διαδεδομένο πρωτόκολλο που χρησιμοποιείται στην πράξη είναι το HTTP. Το πρωτόκολλο SOAP χρησιμοποιεί την μέθοδο POST του HTTP για την αποστολή μηνυμάτων. Μια επικοινωνία τύπου RPC αποτελείται από ζεύγη μηνυμάτων SOAP POST Request / POST Response. Σε ένα SOAP POST Response μήνυμα περιέχεται ένα tag <Result> το οποίο έχει σαν τιμή το αποτέλεσμα της κλήσης της μεθόδου και είναι ένα αλφαριθμητικό (string) Unicode.

Ακολουθεί παράδειγμα ενός μηνύματος SOAP Request.

```
POST /WebServicesFolder/MyService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
User-Agent: SOAP SDK
Connection: Keep-Alive
SOAPAction: http://www.mysite.com/Service
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
<soap:Body
  <MyService xmlns=http://www.mysite.com/>
    <first>string1</first>
    <second>string2</second>
  </MyService>
</soap:Body>
</soap:Envelope>
```

Ακολουθεί τώρα ένα παράδειγμα μηνύματος SOAP Response.

```
HTTP/1.1 200 OK
Date: Mon, 23 March 2002 15:34:23 GMT
Content-Type: text/xml; charset=utf-8
Content-Length: length
Server: SOAP Server
```

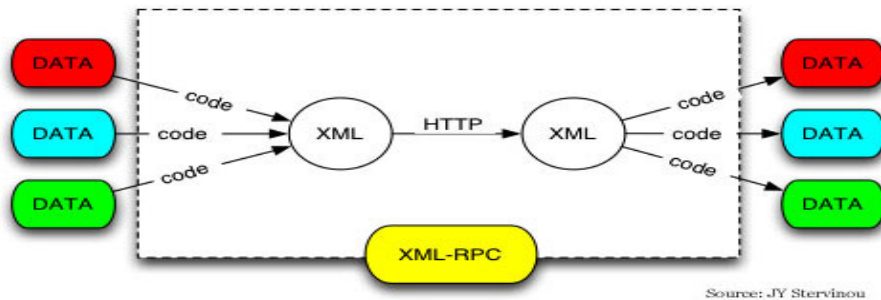
```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
<soap:Body
  <MyServiceResponse xmlns=http://www.mysite.com/>
    <Result>string3</Result>
  </MyServiceResponse>
</soap:Body>
</soap:Envelope>
```

Για τον έλεγχο λαθών κατά τις μεταδόσεις το SOAP έχει το <SOAP:fault>. Σε περίπτωση σφάλματος αποστέλλεται πίσω στην καλούσα διεργασία ένα SOAP Response μήνυμα, στο οποίο περιέχεται ένα tag <SOAP:fault>. Το tag αυτό έχει υποστοιχεία που περιγράφουν το SOAP exception που συνέβη.

Εκτός από την υλοποίηση του RPC για κλήση απομακρυσμένων μεθόδων, ένα μήνυμα SOAP μπορεί υπό άλλες συνθήκες να σταλεί ως ειδοποίηση ή ενημέρωση για κάποια τρέχουσα κατάσταση από έναν

κόμβο σε κάποιον άλλο, χωρίς να περιμένει κάποια απάντηση. Αυτή είναι η δεύτερη βασική λειτουργία των SOAP μηνυμάτων.

Το SOAP είναι μια ειδική υλοποίηση του πρωτοκόλλου XML-RPC. Το XML-RPC, ανάλογα με το SOAP, δουλεύει με βάση την αρχιτεκτονική που φαίνεται στην ακόλουθη εικόνα:

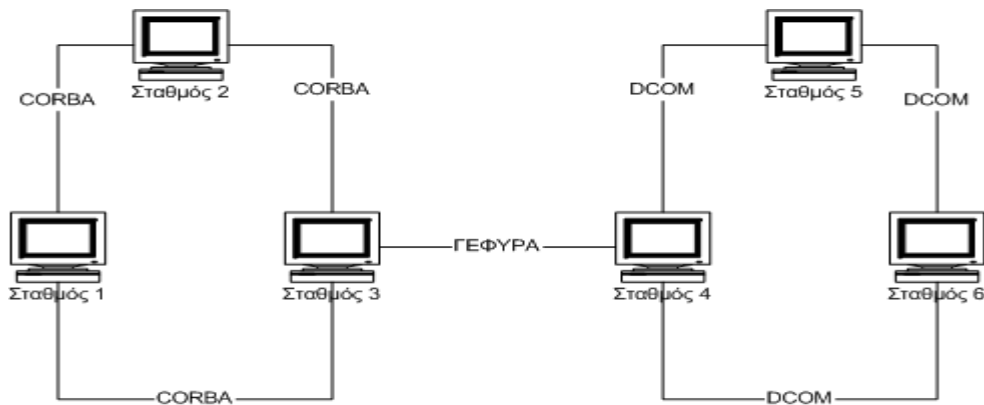


Σχήμα 15 : XML-RPC

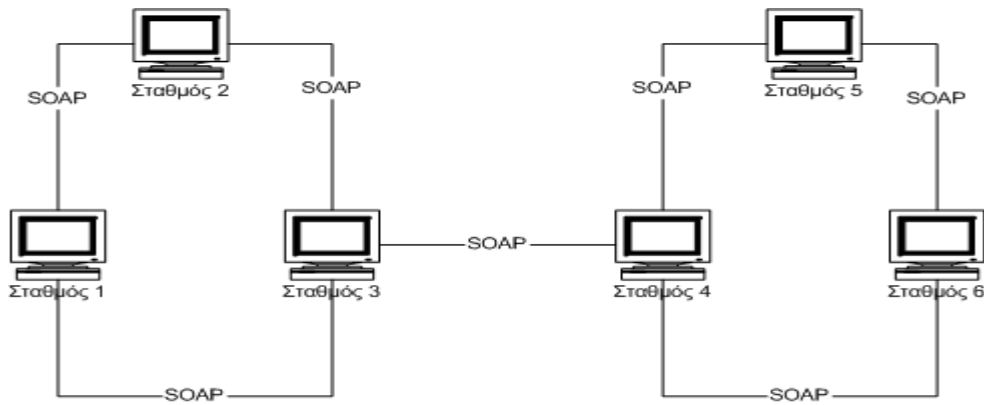
### Απόδοση του SOAP

Τα πρωτόκολλα CORBA και DCOM χρησιμοποιούν δυαδική κωδικοποίηση (binary encoding) για τα μηνύματα που ανταλλάσσουν και θεωρούν δεδομένη την συμφωνία των δύο επικοινωνούντων κόμβων ως προς συμβάσεις σχετικά με το νόημα των δεδομένων που ανταλλάσσονται. Αντίθετα το SOAP χρησιμοποιεί κωδικοποίηση XML στα μηνυματά του για να προσδώσει νόημα στα ανταλλασσόμενα δεδομένα. Το γεγονός αυτό το καθιστά εύκολο στην διαχείριση, διαλειτουργικό και επεκτάσιμο, ωστόσο υστερεί κάπως σε επιδόσεις, συγκριτικά με τα CORBA/DCOM. Οι τυπικές επιδόσεις που έχουν μετρηθεί για το πρωτόκολλο SOAP είναι 500 μηνύματα/sec στην περίπτωση που ο πελάτης και ο διακομιστής SOAP βρίσκονται στο ίδιο μηχάνημα και 300 μηνύματα/sec όταν είναι σε διαφορετικά μηχανήματα [GRA02].

Το μεγάλο πλεονέκτημα της συμβατότητας του SOAP φαίνεται στα παρακάτω σχήματα.



Σχήμα 16 Ασύμβατα CORBA/DCOM δίκτυα που απαιτούν γέφυρα για να επικοινωνήσουν



**Σχήμα 17 Η χρήση του SOAP εξασφαλίζει μια κοινή γλώσσα επικοινωνίας**

Όπως φαίνεται από τα παραπάνω σχήματα, το μεγάλο πλεονέκτημα του SOAP και των Web Services είναι ότι μπορούν να διασυνδέουν συστήματα που χρησιμοποιούν διαφορετικά middleware σε μια ενιαία πλατφόρμα επικοινωνίας. Η πλατφόρμα αυτή είναι ανεξάρτητη από το λογισμικό που τρέχει σε κάθε ένα σύστημα. Η συμβατότητα είναι ένα πολύ σημαντικό χαρακτηριστικό του SOAP. Οι υπάρχουσες υποδομές σε middleware δεν είναι απαραίτητο να καταργηθούν. Στο σχήμα 18 μπορούμε να θεωρήσουμε τα εξής: Οι σταθμοί 1, 2 και 3 επικοινωνούν μεταξύ τους με CORBA. Επίσης οι σταθμοί 4, 5 και 6 επικοινωνούν μεταξύ τους με DCOM. Ωστόσο οι σταθμοί 3 και 4 επικοινωνούν μεταξύ τους με SOAP. Με βάση αυτό το σκεπτικό πετυχαίνουμε διαλειτουργικότητα των συστημάτων χωρίς να καταργούμε τις υπάρχουσες επενδύσεις σε λογισμικό middleware.

Το πρωτόκολλο SOAP είναι αρκετά παραμετροποιήσιμο και βρίσκεται υπό διαρκή εξέλιξη. Μια πρόσφατη σημαντική αλλαγή στην τρέχουσα έκδοση 1.2 είναι η δυνατότητα που υπάρχει να επισυνάπτονται αρχεία στα μηνύματα SOAP, με την χρήση των προδιαγραφών MIME (Multipurpose Internet Mail Extensions) και DIME (Direct Internet Message Encapsulation).



## 6.2.2 Γλώσσα περιγραφής Διαδικτυακών Υπηρεσιών WSDL

Η γλώσσα περιγραφής των Διαδικτυακών Υπηρεσιών εξελίχθηκε από κοινού από τις IBM και Microsoft με στόχο να επιτευχθεί ένα κατάλληλο σχήμα περιγραφής των υπηρεσιών, ώστε αυτές να μπορούν να προσφέρουν τις υλοποιημένες μεθόδους τους στους προγραμματιστές. Κάθε Διαδικτυακή Υπηρεσία συνδέεται με ένα αρχείο με επέκταση .wsdl , στο οποίο περιλαμβάνονται πληροφορίες σχετικά με τις διαθέσιμες μεθόδους της Web Service και τον τρόπο κλήσης τους. Το αρχείο αυτό για κάθε Υπηρεσία είναι προσβάσιμο από τον χρήστη της Υπηρεσίας και βρίσκεται συνήθως αποθηκευμένο σε κάποιο ειδικό μητρώο. Με τις πληροφορίες που περιέχει το αρχείο .wsdl ο πελάτης μπορεί να καλέσει την Υπηρεσία και να αξιοποιήσει τις διαθέσιμες μεθόδους της. Το αρχείο δημιουργείται είτε στατικά από τον προγραμματιστή είτε συνήθως δυναμικά από την εκάστοτε πλατφόρμα στην οποία εκτελείται η Υπηρεσία (διακομιστής SOAP) και είναι ένα XML αρχείο με σημαντικές σε πληροφορία ετικέτες. Οι ετικέτες αυτές είναι οι εξής:

- <definition> Είναι η ετικέτα που βρίσκεται στην κορυφή της ιεραρχίας . Στην ετικέτα αυτή ορίζονται μια ή περισσότερες Υπηρεσίες και επίσης περιέχει κάποια χαρακτηριστικά (attributes), όπως για παράδειγμα τον χώρο ονομάτων στον οποίο εντάσσονται οι ορισμοί στο αρχείο .wsdl
- <type> Είναι προαιρετική ετικέτα και περιέχει δηλώσεις σύνθετων τύπων δεδομένων που πιθανώς χρησιμοποιούνται από την Διαδικτυακή Υπηρεσία.
- <message> Αναφέρεται στην πληροφορία (με τη μορφή μηνυμάτων) που μεταφέρεται από τον πελάτη στον διακομιστή όπου φιλοξενείται η Υπηρεσία και αντίστροφα . Αποτελείται από ένα ή περισσότερα μέρη.
- <portType> Στην ετικέτα αυτή περιγράφονται μια ή περισσότερες ενέργειες, δηλαδή μια σειρά από μηνύματα εισόδου/εξόδου , που έχουν δηλωθεί προηγουμένως στο αρχείο .wsdl , στην/από την Διαδικτυακή Υπηρεσία.
- <operation> Πρόκειται για μια ενέργεια που εκτελείται όπως περιγράφηκε προηγουμένως.
- <binding> Αναφέρεται σε μια ετικέτα <portType> που έχει υλοποιηθεί με χρήση κάποιου συγκεκριμένου πρωτοκόλλου μετάδοσης, όπως το SOAP ή το CORBA.
- <service> Καθορίζει την τοποθεσία (URL) , όπου βρίσκεται η Υπηρεσία στον διακομιστή.

Για την διευκόλυνση των προγραμματιστών που θέλουν να καλέσουν εύκολα μεθόδους από μια Διαδικτυακή Υπηρεσία, υπάρχουν εργαλεία τα οποία σαρώνοντας το αρχείο περιγραφής της Υπηρεσίας (WSDL) παράγουν αυτόματα κώδικα , ο οποίος λειτουργεί ως μεσολαβητής (client proxy) από μια εφαρμογή πελάτη προς την συγκεκριμένη Υπηρεσία.

### 6.2.3 Πρωτόκολλο UDDI

Το πρωτόκολλο UDDI είναι μια υποδομή δημοσίευσης και ανάκλησης των Διαδικτυακών Υπηρεσιών με την χρήση κατάλληλου API . Αυτό διευκολύνει σημαντικά τόσο τους δημιουργούς όσο και τους χρήστες των Διαδικτυακών Υπηρεσιών.

Για την δημοσίευση και εκμετάλλευση των διαθέσιμων Διαδικτυακών Υπηρεσιών λειτουργούν διάφοροι διακομιστές UDDI, οι οποίοι χωρίζονται σε κατηγορίες.

- Πρώτη κατηγορία είναι οι διακομιστές δημόσιας χρήσης, που επιτρέπουν στον καθένα να ψάξει ή να δημοσιεύσει τις δικές του Υπηρεσίες . Οι διακομιστές αυτοί συντηρούνται από τις εταιρείες HP, IBM, Microsoft και SAP. Το δίκτυο αυτό φιλοξενεί το λεγόμενο UDDI Business Registry (UBR). Δημόσιας χρήσης διακομιστές UDDI μπορούν να συντηρούν τόσο οι δημιουργοί εταιρείες των Υπηρεσιών όσο και εναλλακτικοί παροχείς που απλά φιλοξενούν τις υπηρεσίες τρίτων.
- Δεύτερη κατηγορία είναι οι προστατευμένοι διακομιστές, οι οποίοι διαθέτουν το περιεχόμενο τους αποκλειστικά σε ερευνητικές ομάδες και επιτροπές του Διαδικτύου, όπως για παράδειγμα το WWW Consortium.
- Τρίτη κατηγορία είναι οι ιδιωτικής χρήσης διακομιστές, οι οποίοι ευρετηριοποιούν τις εσωτερικές υπηρεσίες μιας εταιρείας ή οργανισμού και προορίζονται για ενδοεπιχειρησιακή και ελεγχόμενη διεπιχειρησιακή χρήση.

Ένας διακομιστής UDDI αποθηκεύει 4 κλάσεις αντικειμένων:

- Το αντικείμενο businessEntity, το οποίο σχετίζεται με κάθε ιδιοκτήτη – κατασκευαστή μιας Διαδικτυακής Υπηρεσίας.
- Το αντικείμενο businessService, το οποίο περιέχει τις υπηρεσίες που προσφέρει μια επιχείρηση.
- Σε κάθε μια από αυτές τις Υπηρεσίες αντιστοιχίζεται ένα αντικείμενο bindingTemplate, που περιέχει πληροφορίες για την συγκεκριμένη Υπηρεσία, δηλαδή μια σύντομη περιγραφή της, με ποιες παραμέτρους μπορεί να κληθεί από εξωτερική εφαρμογή και σε ποιον διακομιστή φιλοξενείται.
- Τέλος υπάρχει το αντικείμενο tModel, το οποίο περιέχει περισσότερες πληροφορίες για την λογική που υλοποιεί η Διαδικτυακή Υπηρεσία, δηλαδή έχει έναν δείκτη προς το αρχείο WSDL της Υπηρεσίας.

Με την αξιοποίηση των διακομιστών UDDI μπορεί να γίνει δημοσίευση (publishing) και ανάκληση (invoking) των Διαδικτυακών Υπηρεσιών. Όλα τα αντικείμενα του UDDI registry διαθέτουν μεθόδους για την αναζήτησή τους με βάση κάποια χαρακτηριστικά τους, όπως για παράδειγμα όνομα, λειτουργία, ιδιοκτήτης και αναγνωριστικό κλειδί. Για την δημοσίευση μιας Υπηρεσίας απαιτείται η δημιουργία λογαριασμού σε έναν διακομιστή UDDI και κατόπιν η δημιουργία των κατάλληλων αντικειμένων στον διακομιστή αυτόν, που θα αντιστοιχούν στον συγκεκριμένο λογαριασμό.

Η ανάκληση μιας Web Service μπορεί να γίνει είτε στατικά κατά την κωδικοποίηση είτε δυναμικά, κατά την εκτέλεση μιας εφαρμογής. Η δεύτερη περίπτωση συμβαίνει με την κατάλληλη χρήση του UDDI API μέσα από το προγραμματιστικό μας περιβάλλον. Πριν από την χρήση μιας Υπηρεσίας συνήθως πραγματοποιείται αναζήτηση και αξιολόγηση διαμέσου ενός διακομιστή UDDI. Για μια αποτελεσματική αναζήτηση μεταξύ όλων των παροχέων διακομιστών UDDI υπάρχει μια χρήσιμη εφαρμογή για επίδειξη στο Διαδίκτυο, στην διεύθυνση [www.soapclient.com/uddisearch.html](http://www.soapclient.com/uddisearch.html).

Αυτή τη στιγμή η προδιαγραφή UDDI βρίσκεται στην έκδοση 2 και η έκδοση 3 είναι υπό εξέλιξη. Μετά την οριστικοποίηση της έκδοσης 3, το UDDI θα προταθεί ως πρότυπο για έλεγχο από κάποια επιτροπή προτύπων. Μέχρι τότε το UDDI θα είναι υπό συνεχή έλεγχο για βελτίωση πολλών παραμέτρων. Για παράδειγμα μια παράμετρος που εν γένει δεν προσφέρεται από ένα μητρώο UDDI είναι η ποιότητα υπηρεσίας (Quality of Service QoS), η οποία μπορεί να βοηθήσει σημαντικά τον χρήστη μιας Διαδικτυακής Υπηρεσίας ώστε να εκτιμήσει την αξιοπιστία της.

Τέλος να αναφερθεί απλά ότι παράλληλα με την τεχνολογία UDDI εξελίσσονται και άλλες δύο εναλλακτικές τεχνολογίες εντοπισμού και ανάκλησης Διαδικτυακών Υπηρεσιών, η Electronic Business XML (ebXML) και Web Services Inspection (WS-Inspection).

#### 6.2.4 Ασφάλεια των Διαδικτυακών Υπηρεσιών

Για την επίτευξη ασφαλούς ανταλλαγής πληροφοριών μεταξύ πελάτη-διακομιστή σε μια σύνοδο Διαδικτυακών Υπηρεσιών (Web Services session) μπορούμε να διακρίνουμε τις εξής περιπτώσεις:

- Σύνδεση με HTTP
- Σύνδεση με HTTPS
- Σύνδεση με SOAP Security Extensions. Τα extensions αυτά είναι:
  - Security Assertion Markup Language (SAML)
  - Extensible Access Control Markup Language (XACML)
  - XML Key Management Specification (XKMS)
  - XML Encryption
  - XML Signature και
  - WS-Security

##### *Βασική πιστοποίηση χρήστη για συνδέσεις πάνω από HTTP 1.1*

Χρησιμοποιείται όνομα χρήστη και κωδικός για την είσοδο σε προστατευμένους πόρους του διακομιστή. Τα στοιχεία αυτά στέλνονται συνήθως με κάποια κωδικοποίηση, όπως για παράδειγμα με χρήση του αλγορίθμου MD5, για προστασία από υποκλοπή που θα συνέβαινε στην περίπτωση μετάδοσης απλού κειμένου. Το πρόβλημα εδώ είναι ότι το σώμα του αποστελλόμενου μηνύματος δεν κρυπτογραφείται. Για περισσότερη ασφάλεια στην πράξη χρησιμοποιούνται κατά την μετάδοση συνδυασμοί από τις τεχνολογίες δημοσίου κλειδιού και Kerberos.

##### *Πιστοποίηση χρήστη για συνδέσεις πάνω από HTTPS*

Στην περίπτωση αυτή γίνεται χρήση ψηφιακών πιστοποιητικών και του SSL για ασφαλή μετάδοση. Συνήθως η σύνδεση γίνεται στην θύρα 443, αντί για την θύρα 80 στην οποία γίνονται οι συνδέσεις HTTP. Το πρόβλημα με τις Διαδικτυακές Υπηρεσίες είναι ότι κατά την εκτέλεσή τους μερικές φορές χρειάζεται η πληροφορία να περάσει και από κάποιον τρίτο ενδιάμεσο κόμβο και έτσι το HTTPS δεν μπορεί να εξασφαλίσει ότι η πληροφορία δεν αλλοιώθηκε κατά την μετάβαση στον ενδιάμεσο αυτό κόμβο.

##### *Άλλες μέθοδοι πιστοποίησης χρήστη*

Γενικά οι εφαρμογές που κάνουν χρήση εγγράφων XML μέσω SOAP έχουν πολύ υψηλές απαιτήσεις σε ασφάλεια, μιας και τα έγγραφα XML είναι σε απλό κείμενο και όχι σε δυαδική μορφή. Έτσι είναι πάρα πολύ εύκολη η αλλοίωσή τους. Για την επίτευξη ασφάλειας εφαρμόζονται οι παρακάτω τεχνολογίες:

- Η XML Signature. Το World Wide Web Consortium μαζί με την επιτροπή Internet Engineering Task Force έχει ορίσει το πρότυπο της υπογραφής XML (XML Signature), δηλαδή ενός αρχείου XML, που αποτελεί ψηφιακή υπογραφή για κάποιο περιεχόμενο. Το πρότυπο κάνει χρήση των αλγορίθμων DSS και SHA-1. Με την χρήση της υπογραφής XML μπορούν να υπογραφούν αρχεία οποιουδήποτε τύπου. Το περιεχόμενο που υπογράφεται μπορεί να βρίσκεται εντός ή και εκτός του αρχείου XML της υπογραφής. Στην δεύτερη περίπτωση απλά υπάρχει μια αναφορά με URI στο περιεχόμενο που υπογράφεται. Επειδή η τιμή σωρού, που προκύπτει με την εφαρμογή της συνάρτησης σωρού, θα είναι διαφορετική για αρχεία XML με ίδιο περιεχόμενο αλλά διαφορετική δομή, θα πρέπει πρώτα όλα τα έγγραφα XML να λαμβάνουν μια “κανονική μορφή”, πριν δημιουργηθεί η XML υπογραφή τους. Παράλληλα με τα παραπάνω εφαρμόζεται και το πρότυπο της κρυπτογράφησης XML (XML Encryption) με το οποίο μπορούν να κρυπτογραφηθούν ανεξάρτητα τα διάφορα στοιχεία ενός εγγράφου XML.
- Το πρότυπο XKMS, με το οποίο γίνεται η διαχείριση των δημόσιων και ιδιωτικών κλειδιών στις Διαδικτυακές Υπηρεσίες. Αποτελείται από τα βασικά συστατικά XML Key Information Service Specification (X-KISS) και XML Key Registration Service Specification (X-KRSS).
- Η γλώσσα SAML, με την οποία υλοποιείται το πρότυπο PMI (Permissions Management Infrastructure). Το πρότυπο PMI διαμέσου ενός συνόλου κανόνων και πολιτικών ελέγχει την

ταυτοποίηση χρηστών σε ένα σύστημα. Η υλοποίηση του PMI με την SAML αφορά σε B2C (εμπορικά sites) και B2B εφαρμογές. Με την SAML γίνεται εφαρμογή συγκεκριμένων πολιτικών ασφαλείας (security policies). Η περιγραφή των πολιτικών αυτών και η πιστοποίηση χρηστών και διεργασιών βάσει κριτηρίων γίνεται με πολλούς τρόπους. Ένας από αυτούς είναι με την μορφή Extensible Access Control Markup Language (XACML) που υλοποιείται από την εταιρεία Oasis Group ([www.oasis-open.org](http://www.oasis-open.org)). Η XACML είναι μια γλώσσα που περιγράφει αναλυτικά τα κριτήρια πρόσβασης σε υπολογιστικούς πόρους και σε πληροφορίες διαθέσιμες στο Διαδίκτυο.

- Το WS Security, ένα σύνολο προδιαγραφών ασφαλείας που υλοποιείται από τις εταιρείες IBM, Microsoft και Verisign. Χρησιμοποιεί πολλές από τις γνωστές μεθόδους πιστοποίησης και κρυπτογράφησης, όπως η τεχνολογία Public Key Infrastructure (PKI) και Secure Socket Layer (SSL). Παράλληλα υλοποιείται και μια σειρά από επιπρόσθετες προδιαγραφές, που είναι υπό εξέλιξη και οι οποίες είναι οι WS-Policy, WS-Trust, WS-Privacy, WS-Secure Conversation, WS-Federation και WS-Authorization.

## 6.3 Εργαλεία για υλοποίηση των Web Services

Υπάρχουν αρκετές πλατφόρμες στις οποίες μπορεί να βασιστεί κανείς για τη δημιουργία ενός web service. Από τη μεριά της Microsoft, οι έτοιμες λύσεις που δίνει το περιβάλλον Visual Studio .Net έχουν προσελκύσει πολλούς προγραμματιστές για να δημιουργούν τέτοιες υπηρεσίες. Επίσης άλλες μεγάλες εταιρίες όπως η IBM και η ORACLE χρησιμοποιούν τα δικά τους προγραμματιστικά εργαλεία. Εκτός από τα εμπορικά πακέτα, οι περισσότερες γλώσσες προγραμματισμού, όπως για παράδειγμα η Java, PHP, έχουν ενσωματώσει στις δυνατότητές τους την δυνατότητα για αυτόματη δημιουργία SOAP servers και δημοσίευση και ανάκληση των web services.

Υπάρχουν μερικά εργαλεία που διατίθενται δωρεάν στο διαδίκτυο και απλουστεύουν σημαντικά τον προγραμματισμό Διαδικτυακών Υπηρεσιών. Ακολουθεί λίστα με τα σημαντικότερα από αυτά.

### 6.3.1 Apache Axis

Ο οργανισμός Apache κατασκεύασε ένα εργαλείο συμβατό με το SOAP 1.1, το οποίο ονομάστηκε Apache SOAP και το οποίο αργότερα μετεξελίχθηκε σε περιβάλλον ανάπτυξης Java, με το όνομα Apache Axis. Για να λειτουργήσει σε ένα σύστημα, το Apache Axis απαιτεί την εγκατάσταση ενός διακομιστή εφαρμογών, όπως είναι ο Apache Tomcat, και έναν σαρωτή XML (XML parser). Αρχικά δημιουργείται η εφαρμογή της Διαδικτυακής Υπηρεσίας σε γλώσσα Java και κατόπιν δίνεται η δυνατότητα στον διαχειριστή του διακομιστή να δημοσιεύσει επιλεκτικά κάποιες από τις μεθόδους της εφαρμογής ή και όλες καθώς και να ρυθμίσει διάφορες άλλες σχετικές παραμέτρους. Το αρχείο WSDL επίσης παράγεται αυτόματα.

### 6.3.2 IBM Web Services Toolkit

Η άλλη μεγάλη εταιρία που έχει συμβάλει τα μέγιστα στον σχεδιασμό, ανάπτυξη και καθιέρωση των τεχνολογιών των Διαδικτυακών Υπηρεσιών, διαθέτει και αυτή ένα δικό της σύνολο εργαλείων με όνομα Web Services Toolkit 3.0.

Στο πακέτο υπάρχει ενσωματωμένη μια περιορισμένη έκδοση του διακομιστή εφαρμογών WebSphere της IBM. Το toolkit παρέχει την λειτουργικότητα που παρέχουν και τα προηγούμενα πακέτα, όπως για παράδειγμα αυτόματη ή χειρωνακτική δημιουργία του αρχείου περιγραφής των δημιουργηθεισών Διαδικτυακών Υπηρεσιών (WSDL). Αυτό που πρέπει να τονισθεί, διότι κάνει το πακέτο να ξεχωρίζει από τα προηγούμενα, είναι η πολύ καλή τεκμηρίωση και το πλούσιο συνοδευτικό υλικό που περιλαμβάνει έτοιμα παραδείγματα σύνθετων εφαρμογών που βασίζονται σε Διαδικτυακές Υπηρεσίες, όσο και μελέτες περιπτώσεων (case studies) για την επιτυχή αξιοποίηση και εισαγωγή των σχετικών τεχνολογιών στον κύκλο συναλλαγών μιας σύγχρονης επιχείρησης.

Εκδόσεις του toolkit τόσο για Windows 2000 όσο και για Linux μπορούν να βρεθούν στην διεύθυνση [www.alphaworks.ibm.com/tech/webservicestoolkit](http://www.alphaworks.ibm.com/tech/webservicestoolkit).

### 6.3.3 Microsoft SOAP Toolkit 2

Με το MS SOAP Toolkit 2 είναι δυνατή στον προγραμματιστή η αξιοποίηση των Διαδικτυακών Υπηρεσιών σε πλατφόρμες Windows στις οποίες δεν έχει εγκατασταθεί το .NET Framework. Μπορεί να ενσωματωθεί στο περιβάλλον του Visual Studio 6. Υποστηρίζει το SOAP έκδοση 1.1, την WSDL έκδοση 1.1 και το UDDI.

Προσφέρει συστατικά τόσο πελάτη όσο και διακομιστή. Υπάρχει η δυνατότητα για αλληλεπίδραση των Διαδικτυακών Υπηρεσιών με στοιχεία COM και για τον λόγο αυτό παράγονται αυτόματα δύο αρχεία, ένα με επέκταση .wsdl και ένα με επέκταση .wsml (Web Services Meta Language).

Υπάρχει επίσης και το συστατικό Remote Object Proxy Engine (ROPE) για την αυτοματοποίηση της διαδικασίας παραγωγής μηνυμάτων για την επικοινωνία με απομακρυσμένα αντικείμενα COM και αντικείμενα των Διαδικτυακών Υπηρεσιών. Τέλος για την απλούστευση της διαδικασίας δημιουργίας και επεξεργασίας των XML εγγράφων που στέλνονται ως SOAP μηνύματα, χρησιμοποιείται το SOAP Messaging Object (SMO) framework. Το toolkit μπορεί να βρεθεί στην διεύθυνση <http://msdn.microsoft.com/xml>

#### 6.3.4 Οι Web Services στο περιβάλλον .NET

Η πλατφόρμα .NET βασίζεται στο .NET Framework, το οποίο διαθέτει μια σειρά από βιβλιοθήκες κλάσεων (Framework Class Library - FCL) και ένα περιβάλλον εκτέλεσης των εφαρμογών καλούμενο Common Language Runtime (CLR). Ο κώδικας που αναπτύσσεται με μια από τις γλώσσες που υποστηρίζουν τα FCL και CLR (όπως είναι για παράδειγμα οι Visual Basic .NET, Visual C++ .NET, Visual C#, Perl και Python) μεταφράζεται πρώτα σε μια ενδιάμεση μεταφέρσιμη μορφή, που καλείται Microsoft Intermediate Language (MSIL). Ο binary κώδικας MSIL μπορεί να μεταφερθεί και να «τρέξει» σε οποιοδήποτε μηχάνημα έχει εγκατεστημένο το CLR, σε αντιστοιχία με την λογική εκτέλεσης των προγραμμάτων με Java Virtual Machine. Το .NET Framework περιγράφεται αναλυτικά στο Common Language Specification (CLS), το οποίο έχει υποβληθεί για τυποποίηση στην ECMA (European Computer Manufacturers Association).

Κατά την δημιουργία και δημοσίευση μιας Web Service στο Visual Studio, δημιουργούνται τα εξής αρχεία στο σύστημα:

- Το αρχείο που περιέχει τον πηγαίο κώδικα της Web Service και μπορεί να είναι για παράδειγμα .vb ή .cs, ανάλογα με την γλώσσα που χρησιμοποιείται για την κωδικοποίηση της υπηρεσίας
- Το αρχείο .asmx που περιέχει τεκμηρίωση για την υπηρεσία. Πλήρης περιγραφή της υπηρεσίας παρέχεται και από το αρχείο WSDL, αλλά η περιγραφή του asmx δίνει χρήσιμα στοιχεία για την υπηρεσία, όπως το που βρίσκεται το source αρχείο που αναφέρθηκε παραπάνω
- Το αρχείο .asax, που αποτελεί μια περιγραφή της συνολικής εφαρμογής στην οποία ανήκει η συγκεκριμένη Web Service
- Τα αρχεία .disco, .vdisco και .map που χρησιμοποιούνται για την αναζήτηση και εντοπισμό της υπηρεσίας από πιθανούς clients που θέλουν να την χρησιμοποιήσουν. Πρόκειται για μια τεχνολογία που χρησιμοποιείται από την Microsoft για να ευρετηριοποιούνται οι τοπικά διαθέσιμες Web Services. Η τεχνολογία των αρχείων .disco λειτουργεί παράλληλα με το UDDI, το οποίο χρησιμεύει στην ευρετηριοποίηση όλων των εξωτερικών για ένα σύστημα διαθέσιμων Web Services. Ωστόσο η χρήση των αρχείων DISCO αντικαθίσταται βαθμιαία από την τεχνολογία WS-Inspection, με την οποία δημιουργούνται αρχεία WS-Inspection για αναφορές σε τοπικές Web Services σε έναν server.
- Το αρχείο .wsdl. Είναι ένα έγγραφο xml που αποτελεί μια λεπτομερή τεχνική περιγραφή της υπηρεσίας. Το αρχείο .wsdl μπορεί κανείς να το προβάλει στον browser πληκτρολογώντας [Διαδρομή αρχείου asmx].asmx?WSDL, π.χ. file:///C:/SystemRoot%/Visual Studio/My Project/MyService.asmx?WSDL

Για την ανάκληση (invoking) μιας Web Service πρέπει να αναπτυχθεί με την βοήθεια του Visual Studio μια client εφαρμογή. Πρέπει να εκτελεστεί στην συνέχεια μέσα από το τρέχον έργο (project) η εντολή “Add Web Reference”. Με την εντολή αυτή δημιουργείται ένας φάκελος ο οποίος περιέχει ένα αντίγραφο του .wsdl αρχείου της καλούμενης υπηρεσίας, αρχεία .disco και μια κλάση διαμεσολάβησης (proxy class). Η κλάση proxy για την υπηρεσία κατασκευάζεται με βάση το αρχείο περιγραφής WSDL.

Η δοκιμαστική ανάκληση των Web Services στο .NET στην σελίδα του αρχείου .asmx γίνεται όχι με το SOAP αλλά με ένα πρωτόκολλο που βασίζεται σε μηνύματα HTTP POST (request/respose) και χρησιμοποιεί την σύνταξη URI/method?arg1=value1&arg2=value2.

Το πρόγραμμα client δημιουργεί ένα instance της κλάσης proxy και μετά στέλνει σε αυτή τα μηνύματα (με την κλήση των κατάλληλων μεθόδων). Η κλάση proxy έχει το ίδιο όνομα με την αντίστοιχη κλάση της Web Service και βρίσκεται καθ' όλη τη διάρκεια μεταξύ της Web Service και της εφαρμογής client. Για κάθε κλήση μιας μεθόδου της Web Service ουσιαστικά η εφαρμογή client καλεί μια αντίστοιχη μέθοδο της κλάσης proxy. Η κλάση proxy ανακατευθύνει την κλήση αυτή μέσω SOAP στην μέθοδο της Web Service. Μετά την εκτέλεση της μεθόδου από την Web Service και επιστροφή του αποτελέσματος στην κλάση proxy, η κλάση μορφοποιεί τα αποτελέσματα ώστε να είναι κατανοητά από την εφαρμογή client. Το σημαντικό είναι ότι σε όλη την διαδικασία η κλάση proxy είναι διαφανής από την εφαρμογή client. Έτσι δημιουργείται η εντύπωση ότι καλούμε μεθόδους των Web Services σαν να βρίσκονταν διαθέσιμες τοπικά.

Η Microsoft διαθέτει και μια σειρά από άλλες συνοδευτικές τεχνολογίες για την βελτίωση των Web Services με επιπρόσθετα λειτουργικά χαρακτηριστικά. Μερικές από αυτές τις τεχνολογίες είναι:

- Global XML Web Services Architecture, για ανάπτυξη σύνθετων Web Services
- WS-Inspection, για δημιουργία αναφορών (references) προς μια υπηρεσία σε κάποιον server
- WS-Routing και WS-Referral για παραμετροποίηση της δρομολόγησης των SOAP μηνυμάτων
- WS-Security, για εξασφάλιση της ιδιωτικότητας και της ακεραιότητας των μηνυμάτων
- WSE (Web Services Enhancements), μια σειρά από υπηρεσίες που προσφέρουν αυξημένα λειτουργικά χαρακτηριστικά κατά την δημιουργία, δημοσίευση και κατανάλωση μιας Web Service

### 6.3.5 Παράδειγμα μιας Web Service

Έστω μια Web Service που προσφέρει δύο μεθόδους. Η μία μέθοδος υπολογίζει το αόριστο ολοκλήρωμα της συνάρτησης  $f(x)$  για κάποια όρια  $a, b$ , που δίνονται. Η δεύτερη μέθοδος υπολογίζει την τιμή της παραγώγου συνάρτησης  $f'(c)$  σε ένα σημείο  $c$  που δίνεται. Το πρόγραμμα που δηλώνει αυτή την υπηρεσία και που βρίσκεται στον server θα έχει περίπου ως εξής :

```
Public Class MyWebService
{

/* Δηλώσεις για την εισαγωγή των απαιτούμενων βιβλιοθηκών για υποστήριξη Web Services */

/* Ορισμός του namespace και μιας περιγραφής της Web Service */

/* Δηλώσεις σταθερών και μεταβλητών του προγράμματος */

/Συνάρτηση που υπολογίζει το αόριστο ολοκλήρωμα της συνάρτησης f(x) μεταξύ δύο ορισμάτων που δέχεται
ως είσοδο */
    public Integral(float Param1, float Param2)
    {
/* Υπολογισμός τιμής ολοκληρώματος */
    }

/* Συνάρτηση που υπολογίζει την τιμή της παραγώγου συνάρτησης f'(x) σε ένα σημείο c που δίνεται ως
παράμετρος */
    public Differential(float Param3)
    {
/* Υπολογισμός τιμής παραγώγου */
    }

} / * Class MyWebService
```



Ο κώδικας για την εφαρμογή client που κάνει χρήση των μεθόδων της προηγούμενης Web Service θα έχει περίπου ως εξής.

```
Public Class MyClient
{
```

```
/* Δηλώσεις για την εισαγωγή των απαιτούμενων βιβλιοθηκών για υποστήριξη Web Services */
```

```
/* Δημιουργία ενός instance της Web Service. Η Web Service είναι διαφανής για τον τελικό χρήστη. Με την δημιουργία μιας τοπικής κλάσης proxy, ο χρήστης αντιλαμβάνεται την υπηρεσία σαν να ήταν διαθέσιμη στον τοπικό διακομιστή (localhost). Το αντικείμενο TestService που είναι instance της τοπικής κλάσης MyWebService υλοποιεί ουσιαστικά την απομακρυσμένη Web Service. Η κλάση proxy προωθεί όλα τα μηνύματα στην απομακρυσμένη Web Service */
```

```
Private TestService = new localhost.MyWebService()
```

```
float CallIntegral(float a,float b)
{
    float x = TestService.Integral(a,b);
    CallIntegral = x;
}
```

```
float CallDifferential(float c)
{
    float y = TestService.Differential(c);
    CallDifferential = y;
}
```

```
public static void main(String[] args)
{
    float inp1=args(0);
    float inp2=args(1);
    float inp3=args(2);
    /* inp1, inp2 είναι τα όρια για την συνάρτηση Integral και inp3 είναι το σημείο για τον υπολογισμό της συνάρτησης differential */
    double out1= CallIntegral(inp1, inp2);
    double out2= CallDifferential(inp3);
    System.out.println("The Integral of f(x)= " + out1);
    System.out.println("The Differential of f(x)= " + out2);
}
```

```
} ' Class MyClient
```

### 6.3.6 Παραδείγματα υλοποιημένων Web Services στο Διαδίκτυο

Ψάχνοντας στη διεύθυνση <http://www.webservicelist.com> μπορούμε να βρούμε μία πληθώρα από web services συνοδευόμενα από το wsdl file καθώς και ένα on-line demo της λειτουργίας τους. Παραθέτουμε παρακάτω μερικά παραδείγματα που διαθέσιμα στο διαδίκτυο.

- Country Population Lookup Service - Δέχεται το όνομα μίας χώρας και επιστρέφει τον πληθυσμό της  
<http://www.cs.uga.edu/~sent>
- BN Quote Service - Δέχεται το ISBN ενός βιβλίου και επιστρέφει την τιμή του  
<http://www.xmethods.com/help/addspecs.html>
- Rich Credit Card Validator - Ελέγχει τα στοιχεία μίας πιστωτικής κάρτας αν αντιστοιχούν σε υπάρχουσα πιστωτική  
<http://www.richsolutions.com>
- SOAP SMS - Στέλνει κείμενα των χρηστών ως sms σε κινητά.  
<http://www.redcoal.com>
- Microsoft Web Services - Οι πιο δημοφιλείς υπηρεσίες που προσφέρει η εταιρεία είναι οι .NET Passport, .NET Alerts και MapPoint .NET  
<http://www.microsoft.com/myservices>
- Google Web Service - Απαντά σε queries εύρεσης σελίδων και επιστρέφει τα αποτελέσματα σε μορφή επεξεργάσιμη. Επίσης προσφέρει δυνατότητα ορθογραφικού ελέγχου (στα Αγγλικά) για οποιοδήποτε κείμενο  
<http://www.google.com/apis>

## 6.4 Το μέλλον των Web Services

Οι Διαδικτυακές Υπηρεσίες έχουν βοηθήσει και θα βοηθήσουν στην επικοινωνία καταναμημένων συστημάτων. Αυτό που συμβαίνει στην πράξη σήμερα είναι η εκτεταμένη χρήση λογισμικού middleware για την επικοινωνία ετερογενών συστημάτων. Όπως είδαμε όμως και πάλι προκύπτουν ασυμβατότητες μεταξύ των συστημάτων που υποστηρίζουν διαφορετικά middleware. Για την διασύνδεση και επικοινωνία των διαφορετικών middleware, οι Web Services μπορούν να παίξουν καθοριστικό ρόλο. Με τον ίδιο τρόπο που η τεχνολογία HTML μπόρεσε να κάνει εφικτή την πρόσβαση στον χρήστη ενός μεγάλου όγκου δεδομένων και πληροφοριών, έτσι και οι Διαδικτυακές Υπηρεσίες επιδιώκουν να κάνουν εφικτή την πρόσβαση των μηχανών σε έναν Ιστό από υπηρεσίες.

Ειδικά στον τομέα της δημοσίευσης περιεχομένου με content syndication οι Web Services προσφέρουν μια λύση για την γεφύρωση του χάσματος μεταξύ ανομοιογενών συστημάτων, τα οποία επιθυμούν να ανταλλάσσουν περιεχόμενο. Περισσότερα για το θέμα αυτό μπορούν να αναζητηθούν στο αμέσως επόμενο κεφάλαιο 7: Content Syndication και aggregation.

Παρόλο πάντως που με την αρχιτεκτονική των web services αντιμετωπίστηκαν τα περισσότερα προβλήματα συμβατότητας και διαλειτουργικότητας, υπάρχουν ακόμα αρκετά ανοικτά θέματα. Τα πιο σημαντικά από αυτά είναι τα εξής:

- Ασφάλεια
- Ποιότητα επικοινωνίας
- Αυτόματος τρόπος αναζήτησης και ενσωμάτωσης των Web Services σε ένα σύστημα.

### *Semantic Web και Web Services*

Ειδικά για το τελευταίο από τα θέματα που προαναφέρθηκαν διεξάγεται σημαντική έρευνα το τελευταίο διάστημα. Η έρευνα έχει επικεντρωθεί στην ανάπτυξη του Semantic Web (<http://www.w3.org/2001/sw/>), δηλαδή μιας εξελιγμένης μορφής του Παγκόσμιου Ιστού με την οποία επιτυγχάνεται:

- Σημασιολογική ερμηνεία όλων των κόμβων πληροφορίας του Web
- Ορισμός νέων δομών που θα επιτρέπουν την αυτόματη αναζήτηση, σημασιολογική ερμηνεία και κλήση των Web Services.

# 7 *Content aggregation & syndication*

## 7.1 *Εισαγωγή στα πρωτόκολλα syndication*

Όπως αναφέρθηκε στο κεφάλαιο 1, οι διαδικασίες του content aggregation και syndication είναι δύο συμπληρωματικές διαδικασίες που έχουν να κάνουν με την συλλογή και δημοσίευση περιεχομένου αντίστοιχα. Ένα σύστημα CMS διαθέτει μηχανισμούς με τους οποίους συγχωνεύει περιεχόμενο από ετερογενείς εξωτερικές πηγές και τις ενσωματώνει στο δικό του μοντέλο δεδομένων. Παράλληλα το CMS δίνει την δυνατότητα μαζικής δημοσίευσης του περιεχομένου που περιέχει με την μορφή μικρών δομημένων τμημάτων περιεχομένου, τα λεγόμενα "feeds", ώστε να τροφοδοτήσει άλλα συστήματα διαχείρισης περιεχομένου, εφαρμογές και χρήστες που ενδιαφέρονται να συγκεντρώσουν και να αξιοποιήσουν το περιεχόμενο αυτό.

Η διαδικασία της μαζικής διάθεσης περιεχομένου στο εξωτερικό περιβάλλον του CMS, το syndication, είναι ευρέως χρησιμοποιούμενη μέθοδος από μεγάλα ειδησεογραφικά πρακτορεία, όπως το γερμανικό Reuters, το αγγλικό BBC ή τα αμερικανικά Associated Press (AP) και το United Press International (UPI). Τα πρακτορεία αυτά διαθέτουν το περιεχόμενό τους στο κοινό, είτε δωρεάν είτε έναντι κάποιας συνδρομής. Το περιεχόμενο, αφού εισαχθεί στο σύστημα του χρήστη, είναι έτοιμο για επαναδημοσίευση. Το syndicated περιεχόμενο έχει τα παρακάτω βασικά χαρακτηριστικά:

- Έχει ξεκάθαρο στόχο (για παράδειγμα ενημέρωση ή διαφήμιση) και κοινό στο οποίο απευθύνεται
- Είναι αυστηρά δομημένο και ανεξάρτητο κάθε συμφραζόμενης πληροφορίας μορφοποίησης, ώστε η παρουσίασή του να μπορεί εύκολα να αναπροσαρμόζεται σε κάθε σύστημα
- Είναι διαχωρισμένο κατάλληλα σε τμήματα, ανάλογα με το είδος και την θεματική ενότητα του περιεχομένου που περιλαμβάνει
- Διαθέτει πλούσιο σύνολο από metadata, που το καθιστούν ευέλικτο και του προσδίδουν χρήσιμες ιδιότητες

Αν και αρκετές παλιές πηγές περιεχομένου διαθέτουν το περιεχόμενό τους με syndication κάνοντας χρήση proprietary πρωτοκόλλων, η μεγάλη πλειοψηφία των content providers χρησιμοποιούν πλέον την ανοικτή γλώσσα XML για το syndication. Η XML εστιάζει στην οργάνωση της δομής του syndicated περιεχομένου και της πρόσδοσης σε αυτό χαρακτηριστικών μέσω των metadata. Για την υποστήριξη του syndicated περιεχομένου έχουν αναπτυχθεί και χρησιμοποιούνται ευρέως μια σειρά από XML έγγραφα και πρότυπα, στα οποία αναφερόμαστε παρακάτω.


Το RSS αποτελεί το βασικό πρωτόκολλο για syndication, με βάση το οποίο μπορούμε να συλλέγουμε και να αναζητούμε πληροφορίες από πληθώρα πηγών. Παραδείγματα τέτοιων πηγών είναι ειδησεογραφικά πρακτορεία, δελτία τύπου, ενημερώσεις προϊόντων και υπηρεσιών μιας εταιρείας και web sites. Τα content feeds στέλνονται αυτόματα σε ένα πρόγραμμα πελάτη, το λεγόμενο RSS reader ή aggregator. Το feed είναι ένας wrapper για μικρά τμήματα περιεχομένου, όπως είναι μια επικεφαλίδα νέων, ένα κείμενο, μια εικόνα κ.ά.

Μέσα από ένα πρόγραμμα aggregator έχουμε την δυνατότητα της ταυτόχρονης προβολής σε μια σελίδα επικεφαλίδων από πολλά feeds. Η ομαδοποίηση των feeds για καλύτερη παρουσίαση επιτυγχάνεται με την οργάνωσή τους σε κανάλια ή κάποια ανάλογη οντότητα. Κάθε κανάλι αντιστοιχεί στο περιεχόμενο μιας συγκεκριμένης θεματικής κατηγορίας από ένα site. Το πρόγραμμα ανάγνωσης μπορεί να είναι τοπικές είτε web-based εφαρμογές. Ελέγχουν αυτόματα ποια από τα κανάλια έχουν ανανεωθεί στον server και παρουσιάζουν στον χρήστη το νέο ανανεωμένο περιεχόμενο. Η παρουσίαση γίνεται σε δύο φάσεις, πρώτα επισκόπηση των κεφαλίδων των feeds και μετά ανάγνωσης ολόκληρου του περιεχομένου από το URL στον server.

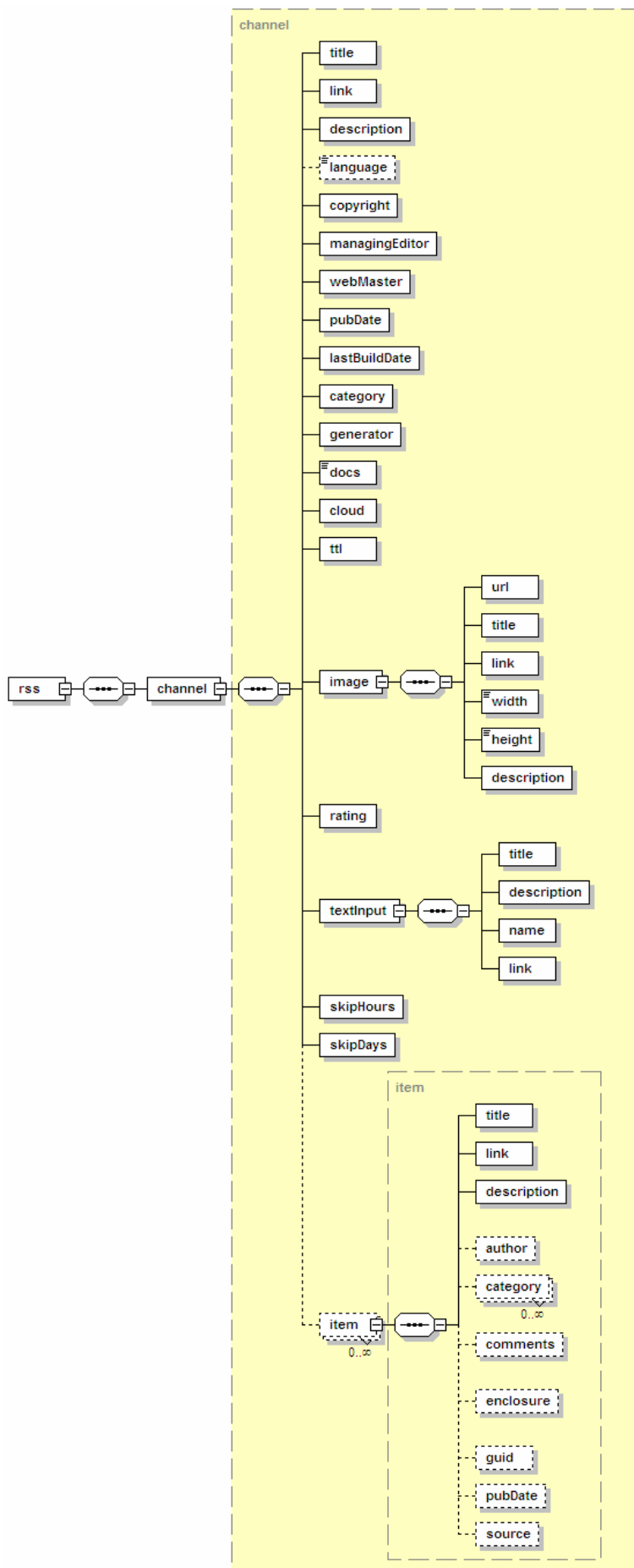
Παράλληλα με το RSS υπάρχουν και άλλα δημοφιλή format, όπως το Atom και μια σειρά από τεχνολογίες σχετικές με syndication. Οι πιο γνωστές είναι οι Outline Processor Markup Language (OPML), Publishing Requirements for Industry Standard Metadata (PRISM), Information and Content Exchange (ICE) και Dublin Core Metadata Set. Μερικές από τις τεχνολογίες αυτές αποτελούν de facto standard, ενώ άλλες βρίσκονται υπό μορφή Internet drafts. Περισσότερα γενικά περί τυποποίησης ο αναγνώστης μπορεί να βρει στο παράρτημα Δ.

Ακολουθεί μια αναλυτικότερη παρουσίαση των specifications για τις τεχνολογίες syndication-aggregation.

### **7.1.1 RSS**

Το RSS είναι ένα format XML εγγράφου για syndication περιεχομένου. Κάθε έγγραφο .rss , ως XML έγγραφο, ακολουθεί το XML 1.0 specification του W3C. Ο κώδικας των RSS αρχείων μπορεί να αναζητηθεί σε εικονίδια με την μορφή , τα οποία παραπέμπουν σε XML αρχεία XML RSS, με κύριο tag το <rss> ή το <rdf>, ανάλογα με την έκδοση του RSS που χρησιμοποιείται.

Ακολουθεί το XML Schema για την έκδοση RSS 2.0, σε μορφή ιεραρχικού διαγράμματος.



Σχήμα 18: Schema για το RSS 2.0

Στην κορυφή της ιεραρχίας έχουμε το στοιχείο <rss>, το οποίο έχει υποχρεωτικά το attribute version και στην περίπτωση μας αυτό έχει την τιμή 2. Αυτό συμβαίνει διότι έχουμε εδώ την υλοποίηση της Really Simple Syndication έκδοσης του RSS. Σε περίπτωση που είχαμε την υλοποίηση RDF Site Summary (έκδοση 1.x και 0.9), τότε το κορυφαίο στοιχείο στην ιεραρχία θα ήταν το <rdf:RDF>.

Περιγράφουμε τα elements του RSS 2.0. Το κύριο υποστοιχείο κάθε rss είναι το στοιχείο <channel>, που είναι το κανάλι περιεχομένου σχετικού με κάποια θεματική ενότητα. Αυτή είναι η ανώτερη δομή περιεχομένου σε ένα αρχείο rss.

#### ΥΠΟΧΡΕΩΤΙΚΑ ΥΠΟΣΤΟΙΧΕΙΑ ΤΟΥ <channel> element [HAR04]:

<title>: ο τίτλος του συγκεκριμένου καναλιού.

<link>: το URL που δείχνει στο site που περιέχει το περιεχόμενο

<description>: μια σύντομη περιγραφή του καναλιού

Κατ' επιλογή μπορούν να προστεθούν και άλλα υποστοιχεία στο στοιχείο channel και αυτά είναι τα ακόλουθα:

#### ΠΡΟΑΙΡΕΤΙΚΑ ΥΠΟΣΤΟΙΧΕΙΑ ΤΟΥ <channel> element

<language>: γλώσσα στην οποία είναι γραμμένο το περιεχόμενο που περιλαμβάνει το συγκεκριμένο κανάλι. Αυτό το χαρακτηριστικό είναι πολύ χρήσιμο, αφού επιτρέπει στο εκάστοτε πρόγραμμα aggregator να συλλέγει περιεχόμενο μόνο από κανάλια που είναι γραμμένα σε μια συγκεκριμένη γλώσσα.

<pubDate>: Η ημερομηνία και ώρα κατά την οποία έγινε η έκδοση του περιεχομένου που βρίσκεται στο κανάλι. Η μορφή της ημερομηνίας-ώρας συμμορφώνεται με το έγγραφο [RFC 822](#).

<lastBuildDate>: Ημερομηνία και ώρα κατά την οποία έγινε η τελευταία πρόσβαση στο περιεχόμενο.

<docs>: Είναι ένα URL που δείχνει στην τεκμηρίωση του ειδικού format στο οποίο είναι γραμμένο το αρχείο, όπως στην περίπτωση αυτή είναι το RSS 2.0.

<generator>: Το όνομα του προγράμματος (Content Management System) που χρησιμοποιήθηκε για να δημιουργηθεί το κανάλι.

<managingEditor>: Ηλεκτρονική διεύθυνση του υπεύθυνου για την διαχείριση του περιεχομένου του καναλιού.

<webMaster>: Ηλεκτρονική διεύθυνση του τεχνικού υπεύθυνου.

<copyright>: Δήλωση προστασίας πνευματικών δικαιωμάτων για το περιεχόμενο που περιέχεται στο κανάλι.

<category>: Ορίζονται μια ή περισσότερες θεματικές ενότητες στις οποίες ανήκει το κανάλι του περιεχομένου, π.χ. «Διεθνείς Ειδήσεις» και «Τεχνολογικές Εξελίξεις».

<cloud>: Είναι ένα στοιχείο που χρησιμοποιείται σαν proxy ώστε μια απομακρυσμένη ή τοπική διεργασία να μπορεί να δηλώσει ότι επιθυμεί να λαμβάνει ενημερώσεις για νέες δημοσιεύσεις στο κανάλι. Για την μεταφορά των πληροφοριών μπορεί να χρησιμοποιηθεί οποιοδήποτε πρωτόκολλο μεταφοράς. Ένα παράδειγμα δήλωσης ενός τέτοιου στοιχείου cloud είναι το εξής: <cloud domain="rpc.sys.com" port="80" path="/RPC2" registerProcedure="pingMe" protocol="soap"/>

<ttl>: Σημαίνει time to live και είναι ένας αριθμός που εκφράζει σε λεπτά πόσο μπορεί ένα κανάλι να παραμείνει στην cache ενός συστήματος πριν να ξαναγίνει ενημέρωση από τον διακομιστή.

<image>: Μια συνοδευτική εικόνα σε μορφή GIF, JPEG ή PNG για το κανάλι.

<rating>: Στοιχείο για παροχή πληροφοριών κατηγοριοποίησης του καναλιού. Αυτό σχετίζεται με την αξιολόγηση του περιεχομένου του καναλιού σύμφωνα με το πρότυπο Platform for Internet Content Selection (PICS) από το W3C.

<textInput>: Ένα ελεγκτήριο εισαγωγής κειμένου για είσοδο κειμένου από τους χρήστες. Η είσοδος που παρέχεται γίνεται συνήθως parse από ένα CGI script. Περιλαμβάνει τα υποστοιχεία title, description, name, link.

<skipHours>: Μια βοήθεια για τα προγράμματα aggregators που συμβουλεύει ποιες ώρες μπορούν να αποφύγουν να ενημερωθούν από το κανάλι, για αποφυγή της κίνησης.



**<skipDays>**: Αντίστοιχα με το παραπάνω, αλλά η αναφορά τώρα είναι για μέρες που μπορούν να αποφευχθούν.

**<item>**: Ένα στοιχείο channel μπορεί να περιλαμβάνει ένα ή περισσότερα υποστοιχεία item. Τα στοιχεία αυτά περιέχουν το περιεχόμενο. Παρακάτω ακολουθεί λεπτομερής ανάλυση των attributes και των elements του στοιχείου <item>.

#### *Αναλυτικότερη περιγραφή για το υποστοιχείο cloud*

Το στοιχείο <cloud> είναι ένα προαιρετικό υποστοιχείο του στοιχείου <channel>. Είναι πολύ σημαντικό διότι καθορίζει μια web service που υποστηρίζει το interface rssCloud, το οποίο μπορεί να υλοποιηθεί με HTTP-POST, XML-RPC ή SOAP 1.1. Ο στόχος αυτού του Interface είναι να επιτρέπει σε μια διεργασία να κάνει register με ένα στοιχείο cloud, ώστε να μπορεί να ειδοποιείται για ενημερώσεις στο συγκεκριμένο κανάλι. Τα περισσότερα προγράμματα news aggregators κάνουν έλεγχο για ενημερώσεις στο περιεχόμενο περίπου κάθε μία ώρα, αλλά σε μερικές περιπτώσεις μας ενδιαφέρει να ενημερωνόμαστε αμέσως μόλις ανανεώνεται το περιεχόμενο ενός καναλιού. Έτσι υλοποιείται μια λειτουργία συνδρομής σε syndicated περιεχόμενο με βάση το RSS format.

Ακολουθούν 2 παραδείγματα για να γίνουν κατανοητά τα παραπάνω:

- <cloud domain="rpc.sys.com" port="80" path="/RPC2" registerProcedure="myCloud.rssPleaseNotify" protocol="xml-rpc" />
- <cloud domain="radio.xmlstoragesystem.com" port="80" path="/SOAP11" registerProcedure="xmlStorageSystem.rssPleaseNotify" protocol="soap"/>

Στο πρώτο παράδειγμα για να ζητηθεί ενημέρωση όταν ανανεωθεί το τρέχον κανάλι, στέλνουμε ένα μήνυμα XML-RPC στην διεύθυνση rpc.sys.com στην θύρα 80, με τοπική διαδρομή φακέλου την /RPC2. Η κληθείσα procedure είναι η myCloud.rssPleaseNotify.

Στο δεύτερο παράδειγμα δίνεται η εντολή στο πρόγραμμα aggregator να στείλει ένα μήνυμα SOAP στην διεύθυνση xmlstoragesystem.com:80, με τοπική διαδρομή /SOAP11 στην μέθοδο με όνομα xmlStorageSystem.rssPleaseNotify για να ζητήσει ειδοποίηση για ενημερώσεις.

Το στοιχείο <cloud>, όπως φαίνεται και από τα παραπάνω παραδείγματα έχει 5 υποχρεωτικά attributes:

- domain είναι το όνομα περιοχής (domain name) ή η διεύθυνση IP του cloud server,
- port είναι η θύρα TCP στην οποία τρέχει το cloud server,
- path είναι η ακριβής τοποθεσία του cloud,
- registerProcedure είναι το όνομα της διαδικασίας ή μεθόδου που καλείται για να ζητηθεί η ενημέρωση
- protocol είναι ένα από τα δυνατά πρωτόκολλα μεταφοράς, δηλαδή ένα από τα xml-rpc, soap και http-post. Το στοιχείο αυτό είναι case sensitive.

Ένας client καλεί μέσω μιας διεργασίας του το στοιχείο cloud server για να κάνει register. Η κλήση αυτή απαιτεί 5 παραμέτρους:

- Το όνομα της μεθόδου που πρέπει το cloud να καλέσει ώστε να ειδοποιηθεί ο client για τυχόν αλλαγές,
- Την TCP θύρα που ακούει η καλούσα διεργασία στον client
- Την τοπική διαδρομή στην οποία βρίσκεται η διεργασία που θα «ακούσει» την ενημέρωση
- Ένα αναγνωριστικό για την δήλωση του πρωτοκόλλου που θα χρησιμοποιηθεί για την μετάδοση των μηνυμάτων
- Μια λίστα με τα URL των αρχείων RSS που θα παρακολουθούνται

Η διεύθυνση IP του καλούντα client μπορεί να προκύψει για την πλευρά του server (για το cloud) από το μήνυμα που στέλνει ο client. Το στοιχείο <cloud> επιστρέφει true αν έγινε επιτυχώς το registration, αλλιώς επιστρέφει false.

Όταν συμβεί ανανέωση του περιεχομένου ενός καναλιού, τότε το στοιχείο cloud καλεί την μέθοδο ή διαδικασία στην μεριά του client που περιλαμβανόταν στην αρχική κλήση registration. Η κλήση αυτή από το στοιχείο cloud γίνεται με ένα όρισμα μόνο και αυτό είναι το url του καναλιού που ανανεώθηκε.

Αν όλα πάνε καλά ο client απαντά με true. Σε αυτό το σημείο ο client μπορεί να διαβάσει το ανανεωμένο περιεχόμενο, να ενημερώσει άλλους clients ότι έγινε ανανέωση, να διαγράψει μια μνήμη

cache, να στείλει ένα email ή και να μη κάνει τίποτα. Κατά σύμβαση, όλα τα registrations σε κανάλια λήγουν μετά από 25 ώρες. Συνεπώς ένας client θα πρέπει να κάνει register κάθε 24 ώρες για κάθε κανάλι ώστε να είναι σίγουρος ότι θα ειδοποιείται όταν ανανεώνεται το περιεχόμενο. Παράδειγμα εφαρμογής όλων των παραπάνω είναι αυτό της εταιρείας Userland στην διεύθυνση [radio.xmlstoragesystem.com](http://radio.xmlstoragesystem.com), με τον client να βρίσκεται στην διεύθυνση [radio.userland.com](http://radio.userland.com).

### **ΥΠΟΧΡΕΩΤΙΚΑ ΥΠΟΣΤΟΙΧΕΙΑ ΤΟΥ <item> element**

Κάθε στοιχείο item πρέπει να διαθέτει υποχρεωτικά είτε ένα στοιχείο description είτε ένα στοιχείο title.

<description> Περιγραφή του περιεχομένου που περικλείει το στοιχείο item

<title> Τίτλος του συγκεκριμένου item

Ένα κανάλι μπορεί να περιλαμβάνει μια σειρά από διαφορετικά στοιχεία <item>. Ένα item μπορεί να είναι ένα άρθρο ή μια ιστορία. Στις περιπτώσεις αυτές είτε έχουμε ολόκληρο το άρθρο στο υποστοιχείο description είτε μόνο τον τίτλο του και το πλήρες άρθρο βρίσκεται από την τιμή του υποστοιχείου link.

### **ΠΡΟΑΙΡΕΤΙΚΑ ΥΠΟΣΤΟΙΧΕΙΑ ΤΟΥ <item> element**

<link>: Είναι το URL όπου βρίσκεται το πλήρες περιεχόμενο, στην περίπτωση που αυτό δεν έχει ενσωματωθεί εξ'ολοκλήρου στο στοιχείο description του <item>.

<source> : είναι το όνομα του RSS καναλιού από το οποίο προήλθε το συγκεκριμένο στοιχείο, με μοναδικό attribute το url. Το στοιχείο source χρησιμοποιείται όταν το item προέρχεται από εξωτερικό feed.

<source url="http://static.userland.com/tomalak/links2.xml">Tomalak's Realm</source>

<enclosure> : Παρέχει πληροφορίες σχετικά με ένα αρχείο που συνάπτεται με το συγκεκριμένο item. Έχει τρία απαιτούμενα attributes. Το url είναι ένα HTTP URL που δηλώνει πού βρίσκεται το αρχείο, το length που είναι το μήκος σε bytes και το type που δηλώνει τι MIME τύπος είναι.

<enclosure url="http://www.scripting.com/mp3s/weatherReportSuite.mp3" length="12216320" type="audio/mpeg" />

<category> : Περιγράφει την θεματική κατηγορία στην οποία ανήκει το συγκεκριμένο item. Αν και μπορούμε να ορίσουμε δικές μας κατηγορίες, συνήθως οι δημιουργοί των feeds χρησιμοποιούν σε αυτό το element κάποιο γνωστό taxonomy, όπως για παράδειγμα το DMOZ ([www.dmoz.org](http://www.dmoz.org)). Το στοιχείο <category> έχει ένα προαιρετικό attribute, το domain, ένα αλφαριθμητικό που ορίζει την διαδρομή στην οποία βρίσκεται το taxonomy κατηγοριοποίησης.

<category>Grateful Dead</category>

<category domain="http://www.dmoz.org">RSS Readers</category>

<pubDate> Είναι η ημερομηνία δημοσίευσης του item.

<pubDate>Sun, 19 May 2002 15:21:36 GMT</pubDate>

<guid> : Είναι τα αρχικά των λέξεων Globally Unique Identifier. Είναι ένα αλφαριθμητικό που προσδιορίζει μοναδικά το εκάστοτε στοιχείο item. Με βάση αυτό το στοιχείο, ένα πρόγραμμα news aggregator μπορεί να συμπεράνει αν ένα item είναι καινούργιο ή όχι.

<guid>http://some.server.com/weblogItem3207</guid>

Το στοιχείο έχει ένα προαιρετικό attribute, το isPermaLink που λαμβάνει τιμές true-false. Όταν είναι true, ο reader μπορεί να καταλάβει ότι πρόκειται για permalink προς το συγκεκριμένο στοιχείο item, δηλαδή ένα url που δείχνει στο πλήρες στοιχείο item και μπορεί να ανοιχθεί για να παρουσιαστούν όλα τα περιεχόμενα του item.

<guid isPermaLink="true">http://inessential.com/2002/09/01.php#a2</guid>

<comments> : Το URL της σελίδας σχολίων για το συγκεκριμένο αντικείμενο

<comments>http://rtyourmusic.com/yacs/commentsn/blogId=705245&amp;itemId=271</comments>

<author> : Η διεύθυνση email του δημιουργού του feed

<author>lawyer@boyer.net (Lawyer Boyer)</author>

Οι τιμές των στοιχείων <link> και <url> πρέπει να ξεκινούν με μια επιτρεπτή URI μορφή και

συγκεκριμένα μια από τις http://, https://, news://, mailto: ή ftp://. Ακολουθούν μερικές ακόμα παρατηρήσεις σχετικά με τα υποστοιχεία του <item>.

- Ένα κανάλι μπορεί να συνδεθεί με ένα αναγνωριστικό όνομα σε ένα σύστημα καταλόγου για κανάλια περιεχομένου RSS. Αυτό μπορεί να γίνει προσθέτοντας στο στοιχείο channel ένα υποστοιχείο τύπου <category>, χρησιμοποιώντας το attribute domain για να δηλώσουμε το όνομα του αναγνωριστικού στον κατάλογο. Για παράδειγμα μπορούμε να δηλώσουμε <category domain="Syndic8">1765</category>.
- Η σχέση των στοιχείων guid με τα στοιχεία link είναι ότι σε κάποια CMS είναι το ίδιο πράγμα, σε κάποια άλλα όμως όχι. Σε κάποια συστήματα το <link> είναι ένα permalink σε ένα weblog item. Σε άλλα συστήματα ωστόσο κάθε στοιχείο item είναι μια περίληψη ενός μεγαλύτερου άρθρου, το <link> δείχνει στο άρθρο και το <guid> είναι ένα permalink με το weblog entry. Για να μην επαναλαμβάνονται τα items από τους aggregators ο καλύτερος τρόπος είναι να παρέχεται στις δηλώσεις πάντα το guid και αν είναι δυνατόν να δηλώνεται ως permalink.

#### *ΥΠΟΧΡΕΩΤΙΚΑ ΥΠΟΣΤΟΙΧΕΙΑ ΤΟΥ <image> element*

<title> Ο τίτλος του γραφικού συνήθως είναι ίδιος με τον τίτλο του feed. Είναι η ίδια τιμή με την τιμή του attribute 'alt' ενός γραφικού σε μια ιστοσελίδα.

<link> Το URL της σελίδας που σχετίζεται με το feed. Συνήθως έχει την ίδια τιμή με το στοιχείο link του καναλιού.

<url> Το URL που βρίσκεται το αρχείο του γραφικού.

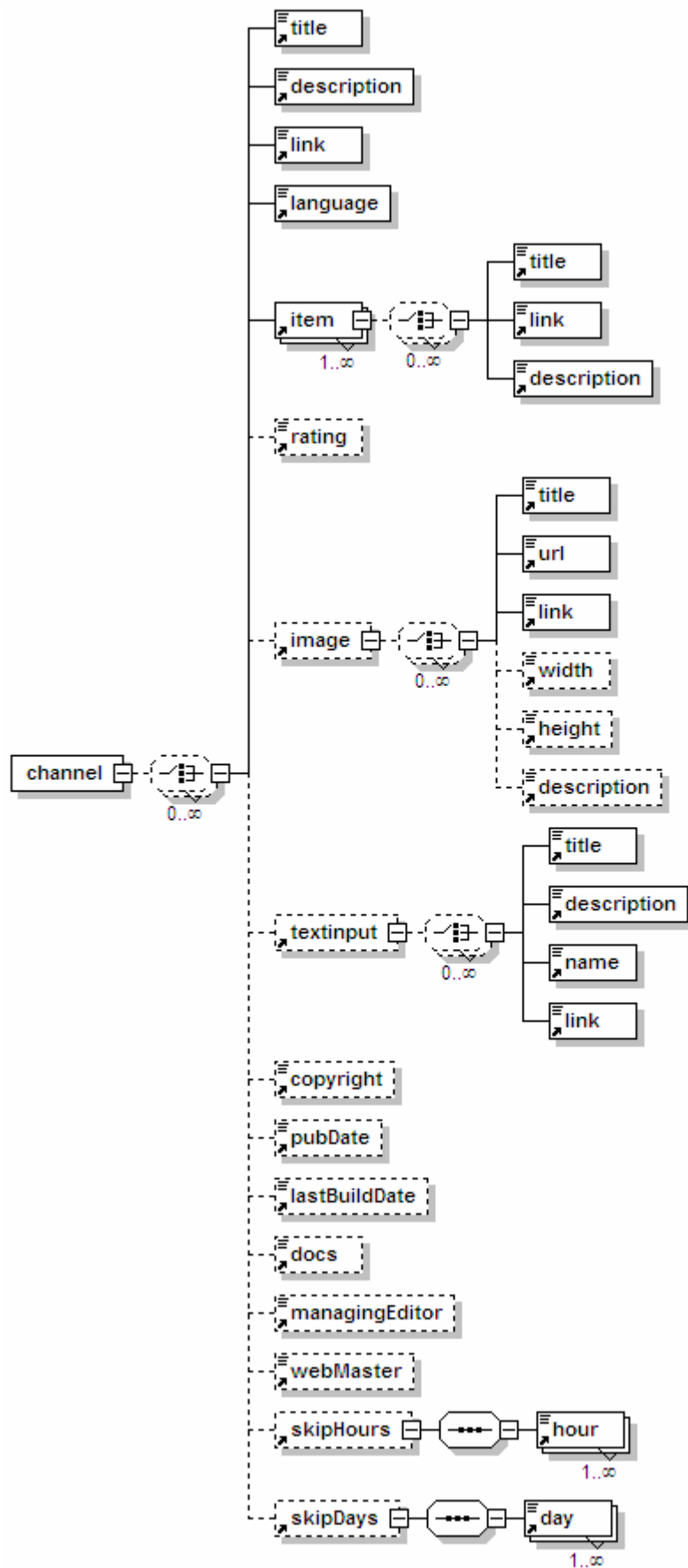
#### *ΠΡΟΑΙΡΕΤΙΚΑ ΥΠΟΣΤΟΙΧΕΙΑ ΤΟΥ <image> element*

<description> Η περιγραφή αυτή είναι αντίστοιχη με το HTML attribute 'title' του γραφικού.

<width> Το πλάτος του γραφικού σε εικονοστοιχεία

<height> Το ύψος του γραφικού σε εικονοστοιχεία

Για λόγους σύγκρισης ακολουθεί το DTD για την έκδοση RSS 0.91 σε μορφή διαγράμματος. Η έκδοση 0.91 του RSS εμφανίστηκε όταν δεν είχαν εμφανιστεί τα XML Schemas.



Σχήμα 19: DTD για το RSS 0.91

### *Μια ιστορική αναδρομή των εκδόσεων του RSS [RSS04]*

Η πρώτη έκδοση του RSS ήταν η έκδοση 0.9 και κυκλοφόρησε από την εταιρεία Netscape το Μάρτιο του 1999. Την εποχή εκείνη χρησιμοποιήθηκε για την δημοσίευση νέων οργανωμένων σε κανάλια στο portal της Netscape, το <http://my.netscape.com>. Αμέσως μετά τον Ιούλιο του 1999 κυκλοφόρησε η επόμενη έκδοση του RSS, η έκδοση 0.91. Στην έκδοση αυτή ενσωματώθηκαν κάποια χαρακτηριστικά με την γενική περιγραφή <scriptingNews>, τα οποία είχαν αναπτυχθεί από την εταιρεία UserLand. Από το σημείο αυτό και μετά η εταιρεία Netscape σταμάτησε επίσημα την ανάπτυξη του RSS, αλλά η Userland συνέχισε να το υποστηρίζει και να το αναπτύσσει. Τον Δεκέμβριο του 2000, η ομάδα εργασίας RSS-DEV ανακοίνωσε την κυκλοφορία του RSS 1.0 ενώ παράλληλα η Userland κυκλοφόρησε το RSS 0.92 και μετά από ένα χρόνο το RSS 0.93. Το RSS αυτή τη στιγμή βρίσκεται στην έκδοση 2.0.2.

Τον Ιούλιο του 2003 η εταιρεία UserLand μετέφερε την ιδιοκτησία της προδιαγραφής RSS 2.0 στο Πανεπιστήμιο “Harvard Law School” και πιο συγκεκριμένα σε έναν φορέα του Πανεπιστημίου, το “Berkman Center for Internet & Society”. Δεν υπάρχουν πλέον πνευματικά δικαιώματα στην εταιρεία Userland, αλλά η προδιαγραφή του XML RSS είναι ελεύθερη για διαμόρφωση και επεκτάσιμη με βάση ειδική άδεια. Παράλληλα έχει συσταθεί ειδική ομάδα εργασίας, αποτελούμενη από εκπροσώπους εταιρειών και εκδοτικών οίκων, με στόχο την ανάπτυξη και διάδοση του RSS στην κοινότητα του Διαδικτύου.

Μερικές φορές προκύπτει η ανάγκη για επέκταση των βασικών δυνατοτήτων του RSS 2.0 με την χρήση συστατικών (modules) που ορίζονται σε συγκεκριμένα namespaces. Ένα feed μπορεί να περιλαμβάνει elements που δεν υπάρχουν στις δηλώσεις του μόνο στην περίπτωση που αυτά τα elements είναι δηλωμένα σε ένα namespace.

Οι εκδόσεις RSS 1.0 και 0.9 ονομάζονται RDF Site Summary και οι εκδόσεις RSS 0.9x και 2.0 Really Simple Syndication ή αλλιώς Rich Site Summary. Η έκδοση RDF Site Summary του RSS αποτελεί μια υλοποίηση της τεχνολογίας Resource Description Framework (RDF), η οποία περιγράφεται αργότερα στο παρόν κεφάλαιο. Οι RDF εκδόσεις του RSS (εκδόσεις 0.9 και 1.0) υποστηρίζονται από το W3C ως εφαρμογές του RDF, το οποίο αποτελεί επίσημη τεχνική σύσταση (Technical Recommendation) του W3C. Οι υπόλοιπες εκδόσεις του RSS είναι εν γένει ασύμβατες με τις RDF εκδόσεις.

Περισσότερες πληροφορίες για την χρήση του πρωτοκόλλου RSS και των υπόλοιπων σχετικών πρωτοκόλλων μπορούν να αναζητηθούν στο Παράρτημα Α.

### 7.1.2 OPML (Outline Processor Markup Language)

Είναι ένα ανοικτό format αρχείου που βασίζεται στην XML και που επιτρέπει την ανταλλαγή outline-structured πληροφορίας, όπως είναι αυτή που δομείται με βάση το RSS, μεταξύ εφαρμογών που τρέχουν σε διαφορετικές πλατφόρμες και λειτουργικά συστήματα. Γενικά όταν ένα αρχείο με επέκταση .opml μεταφέρεται πάνω από HTTP, το mimetype του είναι text/xml. Αυτό επιτρέπει στους Web browsers να χρησιμοποιούν τους κανόνες μορφοποίησης της XML για την προβολή ενός τέτοιου εγγράφου. Σε ένα HTTP POST request υπάρχει το header "Accept". Από αυτό προκύπτει αν ο Web browser υποστηρίζει το mimetype text/xml. Αν όχι, τότε το έγγραφο OPML μπορεί να στέλνεται εναλλακτικά από το server ως mimetype text/html. Mimetype είναι ένα standard για την κατηγοριοποίηση αρχείων (ascii και binary) και email μηνυμάτων που μεταδίδονται μέσω του HTTP και του Internet. Το OPML βρίσκεται στην έκδοση 1.1.

Με τη χρήση του OPML μπορούμε να δημιουργούμε ευρετήρια με πηγές XML Syndicated Content. Κάθε OPML directory μπορεί να οργανώνει feeds τα οποία βρίσκονται σε διάφορα formats. Τα RSS feeds είναι οργανωμένα σε κανάλια περιεχομένου. Τα OPML βρίσκονται υπό μορφή αρχείου .opml που μπορεί να διαβαστεί από έναν reader. Ο reader με τη σειρά του εμφανίζει τα κανάλια διανομής των feeds που περιγράφει το OPML. Ένα έγγραφο OPML καλείται λίστα συνδρομής (subscription list).

#### Δομή ενός εγγράφου OPML

Το κορυφαίο στοιχείο <opml> έχει ένα μοναδικό attribute, το version, αντίστοιχο με αυτό που συναντάμε σε κάθε xml αρχείο. Το opml έχει ως **υποχρεωτικά sub-elements** τα <head> και <body> [OPM04].

Το στοιχείο <head> μπορεί να είναι κενό ή να περιλαμβάνει μια σειρά από υποστοιχεία που σχετίζονται με την μορφοποίηση της παρουσίασης ενός feed στον τελικό χρήστη. Τα υποστοιχεία αυτά είναι τα εξής:

<title> που είναι ο τίτλος του OPML εγγράφου, όπως αναφέρθηκε προηγουμένα

<dateCreated> για την ημερομηνία και ώρα δημιουργίας του εγγράφου

<dateModified> αντίστοιχη πληροφορία για την τελευταία αλλαγή

<ownerName> το όνομα του δημιουργού του εγγράφου

<ownerEmail> το email του

<expansionState> είναι μια λίστα από αριθμούς, χωριζόμενους με κόμματα, οποίοι αντιπροσωπεύουν τις γραμμές-τίτλους που επεκτείνονται κατά την προβολή του εγγράφου στον χρήστη, με άλλα λόγια είναι οι τίτλοι εκείνοι περιεχομένου που γίνονται expand όταν παρουσιάζεται το περιεχόμενο στον χρήστη. Το συγκεκριμένο στοιχείο μπορεί και να αγνοηθεί από τον επεξεργαστή OPML που χρησιμοποιούμε

<vertScrollState> ένας αριθμός που υπαγορεύει ποια από τις γραμμές-τίτλους του εγγράφου θα εμφανίζεται στην κορυφή

<windowTop> είναι ένας αριθμός που εκφράζει τον αριθμό pixel της θέσης του πάνω τμήματος του παραθύρου στο οποίο εμφανίζεται το έγγραφο

<windowLeft>, <windowBottom>, <windowRight> αντίστοιχοι αριθμοί για τα όρια του παραθύρου


Το στοιχείο <body> περιλαμβάνει ένα ή περισσότερα στοιχεία <outline>. Το στοιχείο <outline> με την σειρά του περιλαμβάνει μια σειρά από απεριόριστα σε πλήθος attributes και υπο-στοιχεία επίσης τύπου outline. Τα **βασικότερα attributes για ένα outline** είναι τα παρακάτω:

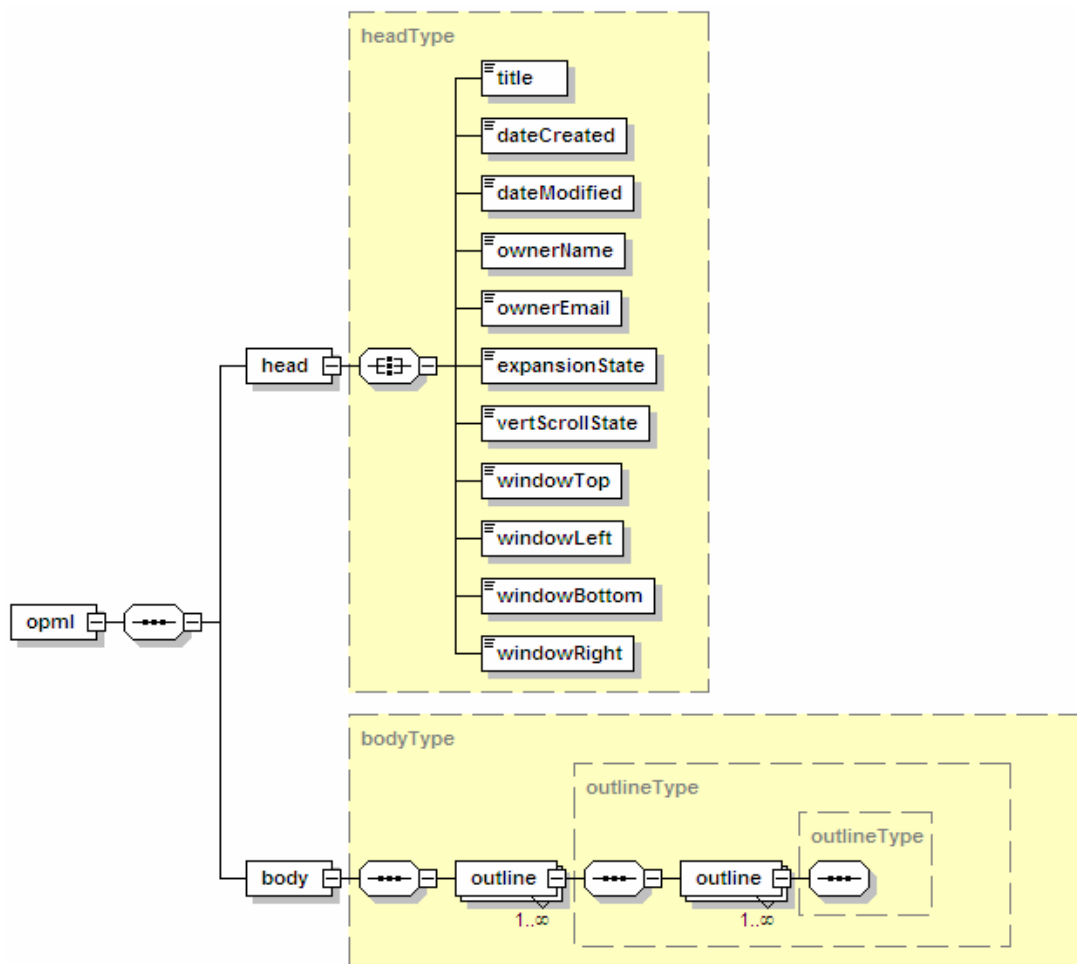
- **text** είναι ένα string χαρακτήρων απεριόριστου μήκους που εμφανίζεται όταν το OPML έγγραφο τροποποιείται ή εμφανίζεται στον χρήστη. Text είναι το όνομα του συγκεκριμένου feed, το οποίο είναι το ίδιο με το στοιχείο title του καναλιού RSS.
- **type** είναι ένα string, που μας υπαγορεύει πώς μεταφράζονται τα άλλα attributes του <outline>, δηλώνοντας το εκάστοτε syndication format
- **isComment** είναι ένα string, που λαμβάνει τις τιμές "true" – "false" και μας υπαγορεύει αν το outline είναι ανενεργό (commented) ή όχι.

- **isBreakpoint** είναι ένα string, που λαμβάνει τις τιμές “true” – “false” και μας υπαγορεύει αν έχει οριστεί κάποιο breakpoint σε αυτό το outline. Αυτό το attribute χρησιμοποιείται για outline που χρησιμοποιούνται για επεξεργασία script.

Σύμφωνα με το schema του OPML, στο element <outline> μπορούν να προστεθούν οσαδήποτε νέα attributes επιθυμούμε. Σαν παράδειγμα αναφέρουμε τα παρακάτω:

- **htmlUrl** μπορεί να είναι ένα string που δείχνει στην web σελίδα όπου βρίσκεται μια οντότητα περιεχομένου (προαιρετικό)
- **Description**, που είναι μια περιγραφή για το συγκεκριμένο outline
- **url** είναι η Web διεύθυνση του feed.

Ακολουθεί ένα **schema για τα αρχεία OPML** σε μορφή διαγράμματος. Ένα αρχείο OPML μπορούμε να βρεθεί σε sites που έχουν OPML directories, συνήθως ψάχνοντας για δεσμούς στα πορτοκαλί εικονίδια . Τα εικονίδια αυτά παραπέμπουν σε κάποιο XML έγγραφο, το οποίο έχει επέκταση είτε .xml είτε .opml.



**Σχήμα 20: Schema OPML**

Το στοιχείο <body> αποτελείται από ένα ή περισσότερα υποστοιχεία <outline>. Το κάθε στοιχείο <outline> έχει αναδρομική δομή, δηλαδή τα υποστοιχεία του είναι επίσης στοιχεία <outline>. Αυτή η αναδρομικότητα μας βοηθά να αναπαραστήσουμε ιεραρχικές δομές καναλιών περιεχομένου. Αυτό το χαρακτηριστικό είναι αρκετά χρήσιμο, δεδομένου ότι ένα σύστημα διαχείρισης περιεχομένου συνήθως χειρίζεται την οντολογία του περιεχομένου του με μια ιεραρχική δομή, όπως είναι για παράδειγμα η δενδρική δομή του συστήματος COMMENT.

Συγκεκριμένα παραδείγματα αρχείων OPML μπορούν να αναζητηθούν στο παράρτημα Α. Επίσης να αναφέρουμε ότι παρόμοια λειτουργία με αυτή του OPML εξυπηρετεί η τεχνολογία Open Content Syndication (OCS). Η τεχνολογία αυτή περιγράφεται αργότερα στο παρόν κεφάλαιο.



### 7.1.3 Atom

Το Atom ορίζεται ως ένα format για αναπαράσταση syndicated περιεχομένου αλλά και ως πρωτόκολλο για επεξεργασία περιεχομένου και ειδικότερα μέσα από εφαρμογές όπως τα Weblogs και τα Wikis. Βρίσκεται αυτή τη στιγμή σε μορφή Internet Draft. Το Atom, τόσο ως format αναπαράστασης syndicated περιεχομένου όσο και ως πρωτόκολλο επεξεργασίας περιεχομένου, αποτελεί αντικείμενο επεξεργασίας από το Internet Engineering Task Force (IETF). Το IETF θα προτείνει το Atom για Proposed Draft στην επιτροπή Internet Engineering Steering Group (IESG). Το μέρος της προδιαγραφής του Atom που αφορά στον πρωτόκολλο επεξεργασίας περιεχομένου μελετάται και από το W3C για την διαμόρφωση κατάλληλης σύστασης (recommendation). Το Atom βρίσκεται στην έκδοση 0.3 .

Σύμφωνα με τις προδιαγραφές [ATO04] , το μοντέλο δεδομένων που χρησιμοποιεί το Atom αποτελείται από:

- Ένα μοντέλο αναπαράστασης πόρων
- Κανόνες σύνταξης
- Ένα format κατάλληλο για syndication περιεχομένου που να ακολουθεί τους κανόνες σύνταξης
- Ένα πρωτόκολλο κατάλληλο για επεξεργασία περιεχομένου που ακολουθεί τους κανόνες σύνταξης

Το Atom specification ορίζει τον τρόπο με τον οποίο το Atom είναι σε θέση να αναπαριστά:

- Έναν πόρο περιεχομένου με βάση την XML
- Ένα feed ή ένα κανάλι περιεχομένου
- Ένα αρχείο με όλες τις εγγραφές σε ένα feed περιεχομένου

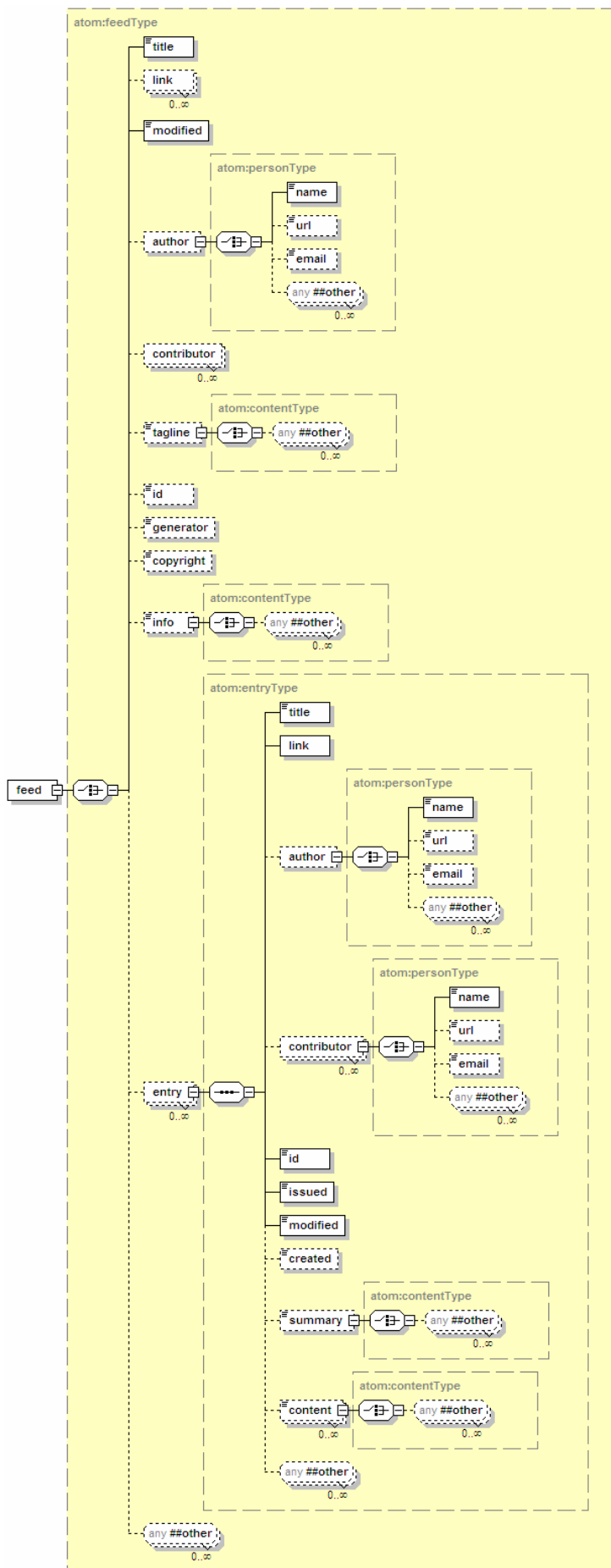
Επίσης το Atom specification ορίζει τον τρόπο με τον οποίο το πρωτόκολλο μπορεί να:

- Δημιουργεί, επεξεργάζεται και διαγράφει εγγραφές περιεχομένου
- Να συλλέγει feeds από πολλές πηγές
- Να χειρίζεται τις ετερογενείς εγγραφές ενός feed
- Ταυτοποιεί την είσοδο εξουσιοδοτημένου χρήστη για επεξεργασία περιεχομένου σε ένα σύστημα
- Προσθέτει ή διαγράφει χρήστες που έχουν την δυνατότητα επεξεργασίας περιεχομένου
- Συλλέγει προτιμήσεις των χρηστών ενός συστήματος περιεχομένου

Σε κάθε αρχείο Atom αντί για τα items του RSS υπάρχουν τα στοιχεία entries. Κάθε <entry> περιέχει την ακόλουθη πληροφορία: τίτλο, δημιουργό περιεχομένου με ηλεκτρονική διεύθυνση και ονοματεπώνυμο και ημερομηνία δημιουργίας και τελευταίας επεξεργασίας των εγγραφών περιεχομένου.

Σημαντικό ρόλο παίζει το attribute permalink που συναντήσαμε και στο πρωτόκολλο RSS. Κάθε στοιχείο στο Atom πρέπει υποχρεωτικά να διαθέτει ένα permalink που να δείχνει στην μόνιμη διεύθυνση του στοιχείου αυτού. Αν μιλάμε σε επίπεδο feed, δηλαδή στην κορυφή της ιεραρχίας, τότε ένα permalink δείχνει στο website που κάνει syndication του συγκεκριμένου feed. Αν μιλάμε σε επίπεδο entry, τότε δείχνει στο URI που βρίσκεται το entry με κάποια τυχαία μορφοποίηση, αν και συνήθως είναι σε μορφή HTML.

Ακολουθεί το XML schema Atom 0.3 υπό μορφή διαγράμματος, όπου διακρίνονται τα βασικά Atom tags με κορυφαίο στην ιεραρχία το <atom : feed> element με attributes τα “version” και “lang”.



Σχήμα 21: Schema Atom

Τα υποστοιχεία του στοιχείου <feed> είναι τα εξής:

- <title>, **υποχρεωτικό**
- <link>, είναι το permalink που προαναφέραμε για την περιγραφή του feed, **υποχρεωτικό**
- <modified>, **υποχρεωτικό**
- <author>, ο συγγραφέας του feed, **υποχρεωτικό**
- <contributor>, κάποιο άτομο που συνέφερε στην δημιουργία του feed, **προαιρετικό**
- <tagline>, είναι μια περίληψη/περιγραφή του συγκεκριμένου feed, **προαιρετικό**
- <id>, link προς μια περιγραφή του feed, **προαιρετικό**
- <generator>, το πρόγραμμα που δημιούργησε το feed, **προαιρετικό**
- <copyright>, **προαιρετικό**
- <info>, μια σειρά πληροφοριών για τους εκδότες περιεχομένου, **προαιρετικό**
- **Entry**, το οποίο διαθέτει με τη σειρά του τα εξής υποστοιχεία:
  - Title**, υποχρεωτικό
  - Link**, υποχρεωτικό
  - Author**, υποχρεωτικό
  - Contributor**, προαιρετικό
  - ID**, προαιρετικό
  - Modified**, υποχρεωτικό
  - Issued**, υποχρεωτικό
  - Created**, προαιρετικό
  - Summary**, προαιρετικό
  - Content**, προαιρετικό

Πέραν του RSS και του Atom έχουν αναπτυχθεί και αρκετά άλλα πρωτόκολλα για aggregation και syndication περιεχομένου. Κάθε ένα από αυτά χρησιμοποιούνταν ή χρησιμοποιούνται από ενώσεις βιομηχανιών και επιχειρήσεων για την κάλυψη των αναγκών τους. Τα μέλη των ενώσεων αυτών είναι και οι μοναδικοί υποστηρικτές των αντίστοιχων format. Κάποια από τα format εκφράζονται με Schemas και κάποια άλλα παλιότερα με DTD. Γεγονός είναι ότι κάποια από αυτά δεν έχουν λάβει την έγκριση ως πρότυπα από τους οργανισμούς IETF και W3C. Παρόλα κάποιοι εκδότες συνεχίζουν να τα χρησιμοποιούν με στόχο την καθιέρωση τους ως πρότυπα. Ακολουθεί μια συνοπτική παρουσίαση των κυριότερων format και μαζί άλλων συνοδευτικών τεχνολογιών που επηρεάζουν τις εξελίξεις στον τομέα του syndication και aggregation περιεχομένου.

#### 7.1.4 Άλλα πρωτόκολλα

##### *Resource Description Framework (RDF)*

Η RDF (πλαίσιο περιγραφής πόρων) είναι ένα πρότυπο για περιγραφή και ανταλλαγή μεταδεδομένων. Είναι επεκτάσιμο και επιτρέπει την προσθήκη νέων στοιχείων. Με την χρήση των μεταδεδομένων, το RDF αποδίδει σημασιολογία (νόημα) σε κάθε πόρο του διαδικτύου. Τελικός στόχος είναι η αυτοματοποίηση της επεξεργασίας των δεδομένων.

Το RDF αναπτύσσεται από το World Wide Web Consortium. Η ανάπτυξη ξεκίνησε από τον μηχανικό πληροφορικής Ramanathan V. Guha και συνεχίστηκε από την εταιρεία Netscape Communications. Το RDF βασίζεται στην ιδέα της δημιουργίας προτάσεων σχετικά με πόρους στο Web με την συντακτική μορφή υποκείμενο-κατηγορημα-αντικείμενο, μια ειδική μορφή που ονομάζεται τριάδα (triple). Το υποκείμενο παίζει το ρόλο του εκάστοτε πόρου που περιγράφεται. Το κατηγορημα είναι το όνομα κάποιου ειδικού χαρακτηριστικού σχετικά με τον πόρο που περιγράφεται. Τέλος το αντικείμενο είναι η τιμή του χαρακτηριστικού αυτού. Με το σχήμα αυτό επιτυγχάνεται η προσθήκη σημασιολογίας σε πόρους του ιστού, όπως είναι οι σελίδες ενός web site. Όλα τα παραπάνω βασίζονται στην γλώσσα Web Ontology Language (OWL).

Η έκδοση 1.x του RSS (RDF Site Summary) αποτελεί μια υλοποίηση της RDF για την διαδικασία του syndication.

Αλλά η χρήση της RDF μπορεί να παίξει σημαντικό ρόλο και στην διαδικασία του aggregation. Με το aggregation περιεχομένου γίνεται συλλογή περιεχομένου από διαφορετικές ετερογενείς πηγές. Στην πράξη η διαδικασία του aggregation λαμβάνει χώρα σε διαφορετικά επίπεδα σε πολλές διαδικτυακές και intranet εφαρμογές. Οι εφαρμογές αυτές είτε παρέχουν περιεχόμενο το οποίο συλλέγεται offline, είτε χρησιμοποιούν δικό τους server-side και client-side κώδικα για να εκτελέσουν την διαδικασία του aggregation δυναμικά. Η δύναμη της τεχνολογίας RDF βρίσκεται στην χρήση της XML και των metadata για να ελέγχει τόσο το στατικό όσο και το δυναμικό aggregation. Ας προχωρήσουμε βλέποντας τι μπορεί να προσφέρει η τεχνολογία RDF στην αποτελεσματικότερη υλοποίηση του aggregation.

Η λύση στο πρόβλημα της υλοποίησης του content aggregation ήρθε αρχικά από τα συστήματα διαχείρισης περιεχομένου, τα οποία δίνουν στους παρόχους περιεχομένου τα εργαλεία που χρειάζονται για την συλλογή ετερογενών δεδομένων, την μετατροπή τους σε μια κατάλληλη εσωτερική για το CMS μορφή (φυσική αναπαράσταση) και τελικά την εισαγωγή τους στο repository. Με την εισαγωγή του περιεχομένου στην βάση του CMS είναι εφικτό από τους προγραμματιστές να έχουν πρόσβαση σε αυτά κάνοντας χρήση server-side τεχνολογιών, όπως CGI, Perl scripts και Java servlets, για την παραγωγή δυναμικών ιστοσελίδων.

Εάν μια εφαρμογή που προσπελαύνει το περιεχόμενο του CMS αλλάζει, τότε αν η ίδια η εφαρμογή δεν έχει υποδομή για την επαναχρησιμοποίηση του μοντέλου δεδομένων που χρησιμοποιεί το CMS, θα πρέπει μέσα στο σύστημα διαχείρισης να γίνουν μια σειρά από μετατροπές δεδομένων, ώστε το σύστημα να είναι συμβατό με την εφαρμογή που καλείται να εξυπηρετήσει. Για παράδειγμα, έστω μια επιχείρηση η οποία συντηρεί μια σειρά από επιχειρηματικά έγγραφα σε μορφή MS Word. Αν η επιχείρηση αλλάξει την εφαρμογή που χρησιμοποιεί και η νέα αυτή εφαρμογή δουλεύει με μια σχεσιακή βάση δεδομένων, τότε θα πρέπει όλα τα δεδομένα που ήταν καταχωρημένα σε έγγραφα Word να μετατραπούν και να περαστούν στην βάση, χρησιμοποιώντας τα κατάλληλα εργαλεία. Αυτό προϋποθέτει επένδυση σε χρόνο και κεφάλαια.

Η συμβολή της RDF στην περίπτωση αυτή είναι ότι παρέχει την δυνατότητα για πρόσβαση τόσο στη νέα όσο και στην παλιά μορφή του περιεχομένου, χωρίς να απαιτούνται πολλές αλλαγές στο CMS. Με την RDF γίνεται μια συσχέτιση αυτού που ονομάζεται εικονικοί πόροι (virtual resources) με πραγματικά τμήματα περιεχομένου, όπως είναι ένα έγγραφο Word. Οι εικονικοί πόροι δημοσιεύονται και σε αυτούς ενσωματώνεται κώδικας που μπορεί να διαχειρίζεται HTTP requests. Οι εικονικοί πόροι επίσης περιέχουν μεταδεδομένα που δημιουργούν δεσμούς με μεθόδους Java. Οι μέθοδοι αυτές υλοποιούν δυναμική ανάκτηση και τροποποίηση περιεχομένου, με βάση ορίσματα που λαμβάνουν από τα μεταδεδομένα των virtual resources. Αυτό που πετυχαίνει εν γένει η RDF είναι η δυνατότητα επιλεκτικής εισαγωγής περιεχομένου σε ένα σύστημα διαχείρισης περιεχομένου. Μέρος του περιεχομένου του Διαδικτύου μπορεί να παραμείνει στην αρχική του μορφή και ένα μέρος μπορεί να εισαχθεί στο Repository, αλλάζοντας μορφή.

Ωστόσο οι δυνατότητες της RDF δεν περιορίζονται στην υλοποίηση του aggregation. Πολύ σημαντική είναι η συμβολή της τεχνολογίας στην ανάπτυξη του σημασιολογικού Ιστού (Semantic Web).

Πρόκειται για μια κίνηση του W3C για βαθμιαία απόδοση σημασιολογίας στο περιεχόμενο του Παγκόσμιου Ιστού με τρόπο που να είναι «κατανοητό» από τους υπολογιστές, δηλαδή να μπορεί το λογισμικό να προσπελαίνει πιο εύκολα δεδομένα που μέχρι σήμερα η σημασιολογία τους ήταν κατανοητή μόνο από τον άνθρωπο. Βασικά εργαλεία για την επίτευξη του σκοπού αυτού είναι η RDF και η XML, μαζί με μια σειρά από συνοδευτικές τεχνολογίες.

Όσο η RDF εξελίσσεται θα καταστήσει απλή τη δημιουργία μιας νέας γενιάς εφαρμογών με την χρήση μοντέλων RDF. Η δημιουργία αυτών των μοντέλων θα βασίζεται σε μια πολύ υψηλού επιπέδου ανάλυση δεδομένων και μια σειρά από επίσης υψηλού επιπέδου προδιαγραφές. Υπάρχει μάλιστα από πολλούς η προσδοκία ότι με την εξέλιξη αυτών των προδιαγραφών θα μπορούν και άτομα που δεν γνωρίζουν από προγραμματισμό να παράγουν σύνθετες δικτυακές εφαρμογές.

### *Publishing Requirements for Industry Standard Metadata (PRISM)*

Η ομάδα εργασίας PRISM ξεκίνησε το 2001. Δημιούργησε δύο προδιαγραφές:

Την PRISM Specification, που παρέχει ορισμούς για όλα τα θέματα σχετικά με το πλαίσιο λειτουργίας του PRISM και την PRISM Aggregator DTD που είναι ένα νέο standard format για χρήση από διαχειριστές περιεχομένου, δίνοντάς τη δυνατότητα για aggregation και syndication. Πρόκειται για ένα ευέλικτο XML DTD που παρέχει ένα απλό μοντέλο για μετάδοση περιεχομένου και μεταδεδομένων του PRISM.

Η προδιαγραφή τώρα PRISM Specification ορίζει μια συλλογή στοιχείων metadata που καλύπτουν τις βασικές ανάγκες για δημοσίευση περιεχομένου. Για πιο ειδικές περιπτώσεις δημοσίευσης περιεχομένου σε web sites είναι απαραίτητο να οριστούν formats με την χρήση κάποιων DTD, τα οποία θα συνδυάζουν metadata PRISM μαζί με κατάλληλη σήμανση του περιεχομένου. Για τον σκοπό αυτό διατίθεται το Aggregator Message DTD. Ένα σύνολο από XML tags που ανήκουν στην υλοποίηση του PRISM είναι σε θέση να αντιμετωπίζει διάφορες εξειδικευμένες περιπτώσεις μεταφοράς περιεχομένου και ανταλλαγής μηνυμάτων, ειδικά κατά την διάρκεια επιχειρηματικών συναλλαγών. Ευρεία είναι η μέχρι τώρα χρήση του PRISM για syndication περιεχομένου που προέρχεται από ηλεκτρονικά περιοδικά.

Επίσης η προδιαγραφή ορίζει και την γλώσσα PRISM Rights Language (PRL). Η γλώσσα αυτή χρησιμοποιείται για τον ορισμό συγκεκριμένων κανόνων ελέγχου πρόσβασης στο περιεχόμενο. Οι κανόνες αυτοί ισχύουν για τους συνδρομητές περιεχομένου (είτε ιδιώτες είτε εταιρικοί χρήστες) και εφαρμόζονται κυρίως σε συνδυασμό με συστήματα διαχείρισης ψηφιακού περιεχομένου (Digital Assets Management Systems) για την διασφάλιση των σχετικών πνευματικών δικαιωμάτων που συνοδεύουν το περιεχόμενο.

Ακολουθεί ένα παράδειγμα εγγράφου PRISM.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Sample RDF from the PRISM specification - section 2.7.2
  Version 1.0; April 9, 2001
-->
<rdf:RDF xmlns:prism="http://prismstandard.org/1.0#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://wanderlust.com/2000/08/Corfu.jpg">
    <dc:identifier rdf:resource="http://wanderlust.com/content/2357845"/>
    <prism:copyright>Copyright 2001, Wanderlust Publications. All rights
reserved.</prism:copyright>
    <prism:rightsAgent>Phantasy Photos, Philadelphia</prism:rightsAgent>
```

</rdf:Description>  
</rdf:RDF>

### Information and Content Exchange (ICE)

Ξεκίνησε το 1998 με την υποστήριξη 80 μεγάλων εταιριών πληροφορικής σαν ένα πρωτόκολλο για την διαχείριση XML εγγράφων και εξελίχθηκε σε format για υλοποίηση του syndication από τους παρόχους περιεχομένου και τους συνδρομητές τους. Αυτή τη στιγμή βρίσκεται στην έκδοση 2.0 υπό μορφήν πρότασης draft για υιοθέτηση ως προτύπου από το W3C.

Το ICE ορίζει μια σειρά από προδιαγραφές που έχουν να κάνουν με τους ρόλους και τις υποχρεώσεις των παρόχων περιεχομένου (syndicators) και των συνδρομητών (subscribers). Ορίζει επίσης την μορφή και τον τρόπο διάθεσης του syndicated περιεχομένου και παρέχει υποστήριξη για τον έλεγχο των σχέσεων μεταξύ syndicators και subscribers. Βασικός στόχος είναι η αποδοτική ανταλλαγή και επαναχρησιμοποίηση του περιεχομένου, δίνοντας έμφαση στις επιχειρηματικές eBusiness και Business2Business εφαρμογές.

Παρότι το πρωτόκολλο βρίσκεται ακόμα υπό εξέλιξη, ένα πολύ ενδιαφέρον χαρακτηριστικό του σύμφωνα με τους κατασκευαστές είναι η ενσωμάτωση των τεχνολογιών των Web Services στις επίσημες προδιαγραφές του ICE. Σε αντίθεση με το RSS το Atom, το ICE υπόσχεται να δώσει έμφαση στην υποστήριξη του επιχειρηματικής κλίμακας content syndication, ώστε να καλύψει κυρίως την επιχειρηματική και την βιομηχανική κοινότητα. Στοιχεί στην εξάλειψη τυχόν προβλημάτων απορρέουν από την χρήση του RSS και σχετίζονται με την αποδοτικότερη διαχείριση των συνδρομών (content subscriptions), την αξιοπιστία των μεταδόσεων με επιβεβαίωση παράδοσης του περιεχομένου και την συνολική ασφάλεια κατά την μετάδοση, τόσο στην λειτουργία pull όσο και push.

### NewsML

Το format αυτό είναι παρόμοιο με τα PRISM και RSS. Έχει υλοποιηθεί από το International Press Telecommunications Council (IPTC), έναν φορέα που συγκροτείται από πολλούς παρόχους περιεχομένου κυρίως προερχόμενοι από τις βιομηχανίες των εκδόσεων και του Τύπου. Λειτουργεί συμπληρωματικά με τις υπηρεσίες που προσφέρουν τα πρωτόκολλα NITF και ICE. Παράδειγμα παρόχων που χρησιμοποιούν το NewsML είναι οι Reuters και Agence France Presse.

Ακολουθεί ένα έγγραφο XML που αποτελεί παράδειγμα εγγράφου NewsML σε μορφή grid.

XML	version	1.0
	encoding	UTF-8
DOCTYPE NewsML	ID: PUBLIC	"urn:newsml:iptc.org:20001006:NewsMLv1.0.dtd:1" "http://www.altova.com/schemas/NewsML/NewsMLv1.0.dtd"
xmispysps	http://www.altova.com/sps/template/News/NewsML.sps	
xm1-stylesheet	type="text/xsl" href="http://www.altova.com/stylesheet/NewsML/IPTCNewsML.xsl"	
NewsML	NewsEnvelope	DateAndTime
	NewsItem	Identification
		NewsIdentifier
		ProviderId
		DateId
		NewsItemId
		RevisionId
		PreviousRevision
		Update
		PublicIdentifier
		NewsManagement
		NewsItemType
		FormalName
		FirstCreated
		ThisRevisionCr...
		Status
		FormalName

Σχήμα 22: NewsML format example

### *Dublin Core Metadata Element Set (DCMES)*

Αποτελεί πρότυπο που ορίζει έναν ενιαίο τρόπο περιγραφής και αναφοράς σε πόρους περιεχομένου, όπως για παράδειγμα έγγραφα, με χρήση μεταδεδομένων. Στόχος είναι η επίτευξη διαλειτουργικότητας σε δικτυακά περιβάλλοντα. Παράγει μια σειρά από μεταδεδομένα και XML Schemas. Η ανάπτυξή του βρίσκεται στην έκδοση 1.1 και υποστηρίζεται από το Dublin Core Metadata Initiative Consortium.

Παράδειγμα υιοθέτησης των προδιαγραφών Dublin Core Metadata Element Set είναι το format RSS 1.0 (RDF Site Summary), το οποίο διαθέτει προαιρετικά elements που έχουν προταθεί από το DCMES. Τα elements αυτά τοποθετούνται τόσο σε επίπεδο στοιχείου <channel> όσο και επίπεδο στοιχείου <item>. Παραδείγματα αυτών των elements είναι τα εξής:

- **dc:date**, για συσχετισμό ημερομηνιών με στοιχεία <item>
- **dc:subject**, για την κατηγοριοποίηση στοιχείων <item> ή <rss>
- **dc:rights**, για την προστασία των πνευματικών δικαιωμάτων που σχετίζονται με ένα <item> ή feed.

Παρακάτω δίνεται μια περιγραφή σε μορφή DTD όλων των υποστοιχείων Dublin Core που χρησιμοποιούνται στην έκδοση RSS 1.0 στα στοιχεία <channel>, <item>, <image> και <textinput> :

- <dc:title> ( #PCDATA )
- <dc:creator> ( #PCDATA )
- <dc:subject> ( #PCDATA )
- <dc:description> ( #PCDATA )
- <dc:publisher> ( #PCDATA )
- <dc:contributor> ( #PCDATA )
- <dc:date> ( #PCDATA ) [W3CDTF]
- <dc:type> ( #PCDATA )
- <dc:format> ( #PCDATA )
- <dc:identifier> ( #PCDATA )
- <dc:source> ( #PCDATA )
- <dc:language> ( #PCDATA )
- <dc:relation> ( #PCDATA )
- <dc:coverage> ( #PCDATA )
- <dc:rights> ( #PCDATA )

Για περισσότερες παραδείγματα σχετικά με τις διαφορετικές εκδόσεις του RSS ο αναγνώστης μπορεί να καταφεύγει στο Παράρτημα Α.

### *Necho ή echo*

Βρίσκεται σε φάση ανάπτυξης ως εναλλακτικό format. Παρέχει τις παρακάτω κατηγορίες feed elements.

- title: απλό κείμενο
- subtitle: απλό κείμενο
- link: URI
- modified: ημερομηνία σε μορφή UTC
- entry: ένα feed μπορεί να διαθέτει μηδέν ή περισσότερα entries. Κάθε entry έχει υποστοιχεία, τα οποία είναι τα ακόλουθα:
  - title: κείμενο
  - subtitle: κείμενο
  - summary: κείμενο
  - author: κείμενο
  - contributor: κείμενο
  - link: URI
  - id: URI

created: ημερομηνία σε μορφή UTC  
issued: ημερομηνία σε τοπική μορφή  
modified: ημερομηνία σε μορφή UTC

### Open Content Syndication (OCS)

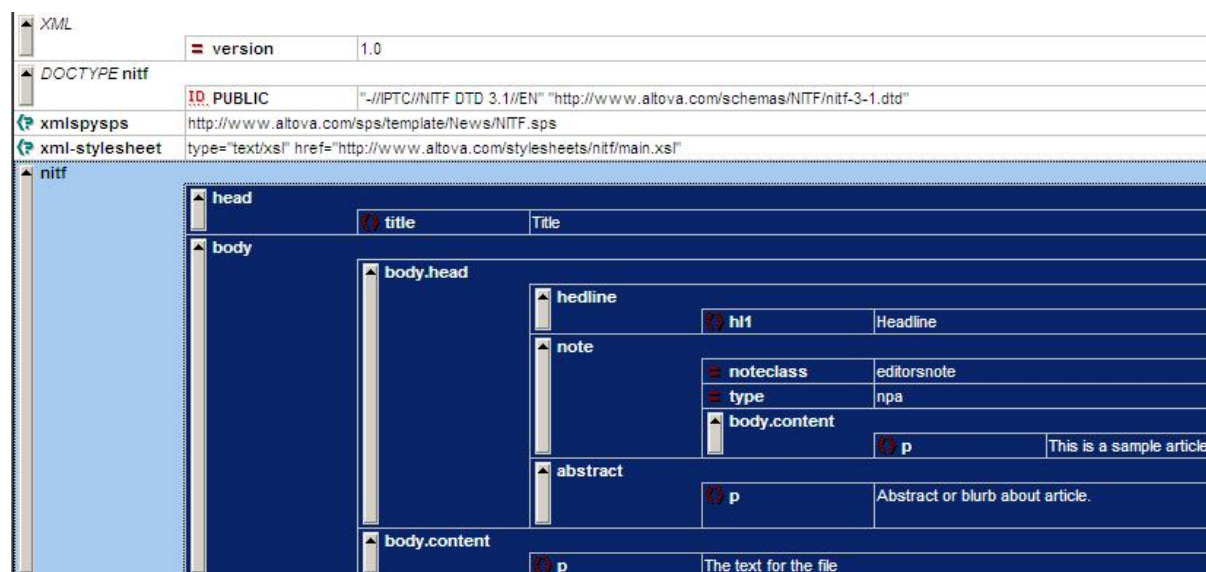
Είναι μια τεχνολογία βασισμένη στην XML και ειδικότερα στην σύσταση RDF, η οποία σχεδιάστηκε για να καθιστά δυνατή την δημιουργία ευρετηρίων καναλιών περιεχομένου για χρήση από portals, client λογισμικό ανάγνωσης outlines και άλλες τοπικές και δικτυακές εφαρμογές. Το ευρετήριο αυτό μπορεί να περιλαμβάνει πληροφορίες για περιεχόμενο από διαφορετικά sites που διαθέτουν feeds. Κάθε κανάλι μπορεί να διατίθεται σε μια σειρά από formats, όπως στις διάφορες εκδόσεις του RSS (Rich Site Summary) versions 0.90 or 0.91, σε μορφή απλού κειμένου καθώς και σε formats για πρόσβαση από κινητές συσκευές, όπως είναι το WML. Οι λειτουργίες ευρετηριοποίησης που προσφέρει το OCS για την αναζήτηση syndicated content feeds μπορούν να παραλληλιστούν με τις λειτουργίες που επιτελεί ένα UDDI Registry για τις Web Services. Το namespace για το OCS είναι το <http://purl.org/ocs/directory/0.5/#>.

Ένα γνωστό παράδειγμα χρήσης της τεχνολογίας OCS είναι το portal Reptile (Reputation-Enhanced personal and distributed Portal). Αποτελεί ένα ελεύθερου λογισμικού portal, γραμμένο σε Java/XML, που προσφέρει την δυνατότητα ανταλλαγής πληροφοριών μέσω Web με αποδοτικό τρόπο με βάση το πρότυπο καταναμημένων δικτύων Peer to peer (P2P). Στο portal αυτό «τρέχει» μια μηχανή για content syndication. Το περιεχόμενο που ανταλλάσσεται μπορεί να είναι είτε feeds κωδικοποιημένα σε οποιοδήποτε γνωστό XML format (π.χ. RSS) είτε και δυαδικά αρχεία (π.χ. MP3, Ogg Vorbis). Η εφαρμογή Reptile είναι διαθέσιμη για download (client application) και εγκατάσταση στον υπολογιστή του τελικού χρήστη. Ο χρήστης του portal μπορεί να ζητήσει συνδρομή σε όποιο από τα διαθέσιμα κανάλια περιεχομένου τον ενδιαφέρει.

Τα OCS directories μαζί με τα OPML directories είναι οι δύο τρόποι δημοσίευσης και διάθεσης για syndication των καναλιών περιεχομένου από έναν provider.

### News Industry Text Format (NITF)

Ακόμα ένα παλιότερο XML-based format, το οποίο ορίζει το περιεχόμενο και την δομή άρθρων με νέα (news articles). Ευρεία έχει υπάρξει η χρήση του για την μετάδοση syndicated περιεχομένου ηλεκτρονικών εφημερίδων. Ο provider με την χρήση του NITF μπορεί να εκδώσει το περιεχόμενό του σε μια σειρά από μορφοποιήσεις, όπως HTML, WML και RTF. Ακολουθεί ένα έγγραφο XML που αποτελεί παράδειγμα εγγράφου NITF σε μορφή grid. Το πλήρες DTD για το NITF μπορεί να βρεθεί στην διεύθυνση <http://www.nitf.org>.



XML	version	1.0				
DOCTYPE nitf	ID: PUBLIC	"-//PTC//NITF DTD 3.1//EN" "http://www.altova.com/schemas/NITF/nitf-3-1.dtd"				
xmispysps	http://www.altova.com/sps/template/News/NITF.sps					
xml-stylesheet	type="text/xsl" href="http://www.altova.com/style sheets/nitf/main.xsl"					
nitf	head	title	Title			
	body	body.head	headline	h1	Headline	
			note	noteclass	editorsnote	
				type	npa	
				body.content	p	This is a sample article
			abstract	p	Abstract or blurb about article.	
			body.content	p	The text for the file	

Σχήμα 23: NITF format example



### *Channel Definition Format (CDF)*

Αποτελεί μια εφαρμογή της τεχνολογίας PUSH βασισμένης στην XML. Η τεχνολογία PUSH δίνει την δυνατότητα μετάδοσης στον χρήστη δυναμικού περιεχομένου από κανάλια που επιλέγει ο ίδιος. Η τεχνολογία εξελίχθηκε από την εταιρεία Microsoft. Ένα κλασικό παράδειγμα χρήσης της τεχνολογίας CDF είναι η ενεργή επιφάνεια εργασίας των Windows (Windows Active Desktop) σε συνδυασμό με εγγραφή σε κανάλια περιεχομένου, που διανέμουν δυναμικό περιεχόμενο, για παράδειγμα news tickers και weather forecasts. Η ενημέρωση για τα διαθέσιμα κανάλια και η συνδρομή σε αυτά ελέγχεται από τα ειδικά αρχεία .cdf, τα οποία βρίσκονται στα sites που υποστηρίζουν την τεχνολογία CDF. Η τεχνολογία CDF από πολλούς θεωρείται ο «πρόγονος» της ιδέας του syndication αλλά σήμερα δεν χαίρει ευρείας χρήσης.

## 7.2 Εφαρμογές του content syndication

### 7.2.1 Newsletters - spam

Τα feeds υπερέχουν σε σχέση με τις παραδοσιακές λίστες email ως μέσο διανομής περιεχομένου. Από την πλευρά του εκδότη, η χρήση του syndication στην θέση των mailing lists, newsletters και των εμπορικών emails περιορίζει το κόστος και δίνει την δυνατότητα ταχύτερης προσέγγισης ενός ευρύτερου κοινού. Το κόστος για έναν εκδότη περιορίζεται ακόμα περισσότερο αν ληφθεί υπ' όψη πως για την αποστολή εμπορικών email απαιτείται χρησιμοποίηση πολλών ενδιάμεσων SMTP servers, ώστε το περιεχόμενο να καταλήξει στο mailbox του τελικού χρήστη. Η χρήση του syndication παρακάμπτει αυτή την ανάγκη. Επιπλέον η παροχή περιεχομένου στους τελικούς χρήστες με syndication τους κάνει πιο δεκτικούς στην υιοθέτησή του, διότι δεν απαιτείται η παροχή προσωπικών δεδομένων ή ηλεκτρονικών διευθύνσεων στον εκδότη, γεγονός που ενέχει τον κίνδυνο του spam και της απώλειας προσωπικών δεδομένων.

### 7.2.2 Αναζήτηση περιεχομένου

Η χρήση του syndication επιταχύνει κατά μέσο όρο σημαντικά τις αναζητήσεις στο Internet. Η αναζήτηση δεν γίνεται σειριακά site προς site. Από το πρόγραμμα ανάγνωσης γίνεται εγγραφή του χρήστη στα επιθυμητά κανάλια περιεχομένου και αυτόματα αποστέλλεται το νέο περιεχόμενο δυναμικά, όταν λαμβάνουν χώρα ανανεώσεις ή προθήκες νέου υλικού στον server. Επίσης υπάρχουν και μηχανές αναζήτησης που χρησιμοποιούν την τεχνολογία των feeds σε συνδυασμό με κλήσεις στο API γνωστών μηχανών αναζήτησης. Τα αποτελέσματα των αναζητήσεων είναι στοιχεία <item> περιεχομένου, εν αντιθέσει με τις κλασικές μηχανές αναζήτησης, οι οποίες επιστρέφουν web σελίδες. Τέλος πολλά newsgroups, όπως για παράδειγμα τα Yahoo Groups προσφέρουν τα news σε μορφή κάποιου syndication format, όπως το RSS.

### 7.2.3 Blogs

Ένα από τα βασικά πλεονεκτήματα πρωτοκόλλων όπως το RSS είναι ότι καθιστούν εύκολη την διανομή syndicated περιεχομένου σε websites, portals και εφαρμογές desktop. Ένα παράδειγμα μελέτης είδαμε στην περίπτωση του COMMET. Παράλληλα με το RSS εμφανίστηκε ένα πολύ χρήσιμο εργαλείο που τον τελευταίο καιρό γίνεται ολοένα και δημοφιλέστερο στους απλούς χρήστες του Web και αυτό είναι τα Web Logs ή Blogs για συντομία.

Πρόκειται μια μορφή web sites με αυξημένες δυνατότητες aggregation και syndication. Οι δυνατότητες του syndication επιτρέπουν την εύκολη επαναχρησιμοποίηση του περιεχομένου του blog από άλλα μέσα, όπως τα προγράμματα newsreaders, άλλα websites και τις φορητές συσκευές. Οι περισσότερες κοινότητες Blog στο Web βασίζονται σε κάποιο γνωστό syndication format (RSS, Atom). Παράλληλα παρέχουν API με βιβλιοθήκες συστατικών για μια σειρά γλωσσών προγραμματισμού. Με τον τρόπο αυτό οι προγραμματιστές μπορούν να συμβάλλουν ώστε μια εφαρμογή, ένα site ή μια φορητή συσκευή να μπορεί να επωφεληθεί από το διαθέσιμο περιεχόμενο της κοινότητας.

Για περισσότερες πληροφορίες σχετικά με τα Blog, ο αναγνώστης μπορεί να ανατρέξει στο Παράρτημα Β.

### 7.2.4 Dynamic Bookmarks

Ένα άλλο πλεονέκτημα από την χρήση των feeds και του syndicated content είναι ότι με την χρήση των reader και των browser τα feeds είναι μεταφέρσιμα ανεξάρτητα από γεωγραφικά όρια. Τα αγαπημένα κανάλια ενημέρωσης μπορούν να λειτουργήσουν σαν δυναμικοί σελιδοδείκτες (dynamic bookmarks). Εφαρμογή της ιδέας αυτής είναι τα Live Bookmarks. Είναι μια τεχνολογία που χρησιμοποιείται στον browser Mozilla Firefox και επιτρέπει την αποθήκευση των news feeds ως bookmarks. Έτσι μπορεί ο χρήστης να βλέπει μια προεπισκόπηση των τίτλων των feeds για το site ή το blog που τον ενδιαφέρει και αν θέλει να επισκέπτεται το site, ώστε να δει το πλήρες περιεχόμενο. Η ενημέρωση των Live Bookmarks γίνεται αυτόματα όταν ανανεωθεί το περιεχόμενο. Παράλληλα με την υπηρεσία διαχείρισης bookmarks <http://del.icio.us/> γίνεται έκδοση των Live Bookmarks ως RSS feeds ώστε αυτά να είναι μεταφέρσιμα σε οποιοδήποτε μηχάνημα.

### **7.2.5 Mobile Devices**

Ένα πλεονέκτημα των news feeds είναι ότι μπορούν να αποσταλούν μέσα από ασύρματα κανάλια διανομής, δηλαδή στα κινητά τηλέφωνα τρίτης γενιάς και σε συσκευές τύπου Personal Digital Assistant (PDA) και Personal Information Manager (PIM). Μέσα από τις συσκευές αυτές μπορούν οι χρήστες να εγγράφονται συνδρομητές σε διάφορες πηγές περιεχομένου. Το λαμβανόμενο περιεχόμενο εισάγεται στο προσωπικό πρόγραμμα του χρήστη διαμέσου εφαρμογών ημερολογίου και reminders, επιτρέποντας καλύτερο χρονοπρογραμματισμό των δραστηριοτήτων.

### **7.2.6 Επιχειρηματικότητα**

Όσο οι τεχνολογίες syndication γίνονται πιο διαδεδομένες και εμποδώνονται από την κοινότητα του Διαδικτύου, τόσο οι μεγάλοι φορείς περιεχομένου θα μετακινούνται από τα παραδοσιακά κανάλια διανομής σε πιο χρηστικούς τρόπους πώλησης ή δωρεάν παροχής περιεχομένου, όπως τα news feeds. Επίσης οι νέες τεχνολογίες syndication εισάγουν νέες επιχειρηματικές προκλήσεις. Σε μια εταιρεία παρέχονται πλέον νέοι τρόποι προσέγγισης πελατών και προσφοράς πιο εξελιγμένων προσωποποιημένων υπηρεσιών στους υπάρχοντες πελάτες.


## 7.3 Syndication περιεχομένου από το COMMET

### 7.3.1 Γενικά

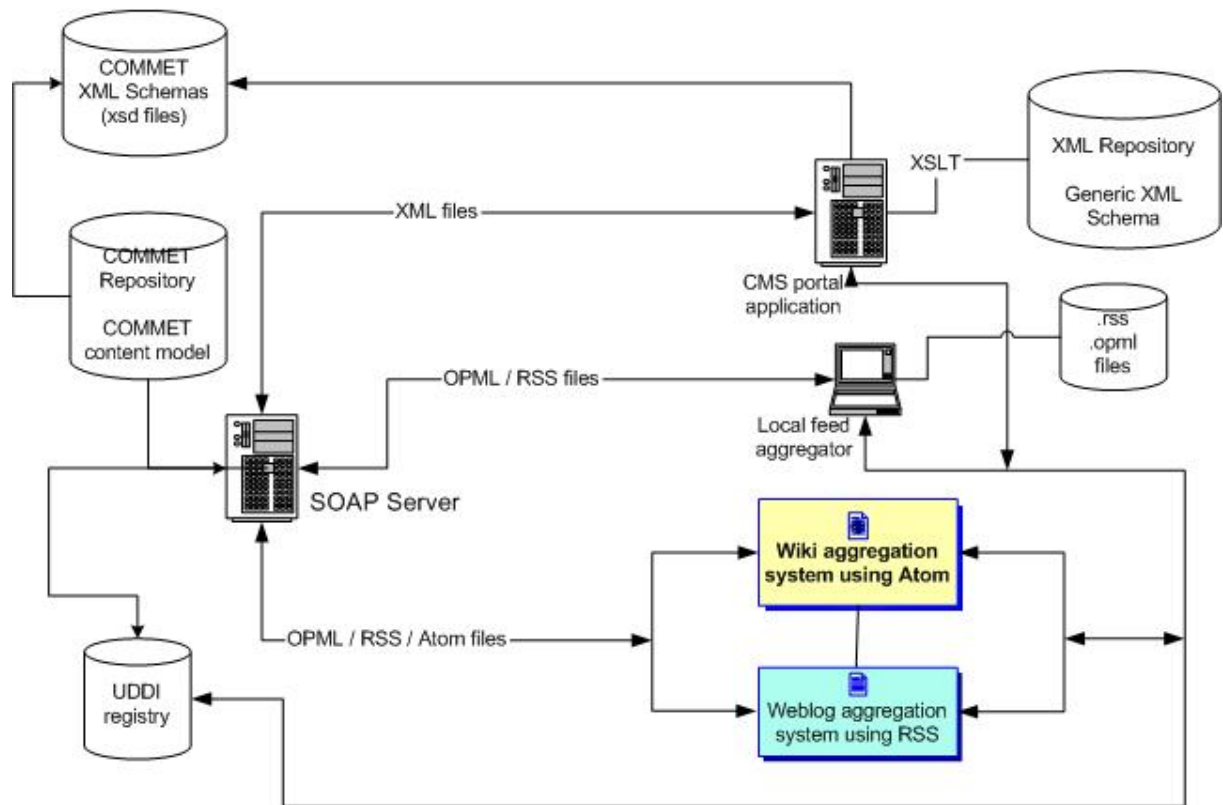
Στην παρούσα εργασία μας ενδιαφέρει η υλοποίηση Web Services για syndication περιεχομένου από το COMMET. Έτσι κάθε εξωτερικό σύστημα ή εφαρμογή, ανεξαρτήτως XML Schema που χρησιμοποιεί, να μπορεί να αξιοποιεί το διαθέσιμο περιεχόμενο. Υπάρχουν δύο σχετικά πιθανά σενάρια:

- Το COMMET παρέχει τα δεδομένα του σε μορφή RSS και Atom, υλοποιώντας εσωτερικά μετασχηματισμούς XSL για την μετατροπή των δεδομένων από το δικό του schema στα γνωστά syndication format. Έτσι τα δεδομένα μπορούν να εισαχθούν αμέσως σε οποιαδήποτε μηχανή υποστηρίζει κάποιο από τα format αυτά.
- Το COMMET παρέχει τα δεδομένα του σε μορφή XML και επίσης παρέχει το XML Schema (μορφή αρχείου .xsd) που χρησιμοποιεί. Έτσι ο κάθε χρήστης μπορεί να αξιοποιήσει τα δεδομένα του COMMET, αρκεί να κατασκευάσει τον κατάλληλο μετασχηματισμό από το schema του COMMET στο schema που αυτός χρησιμοποιεί.

Για να ικανοποιηθούν και τα δύο παραπάνω σενάρια για την διαδικασία του syndication, πρέπει να εκτελεστούν οι ακόλουθες εργασίες:

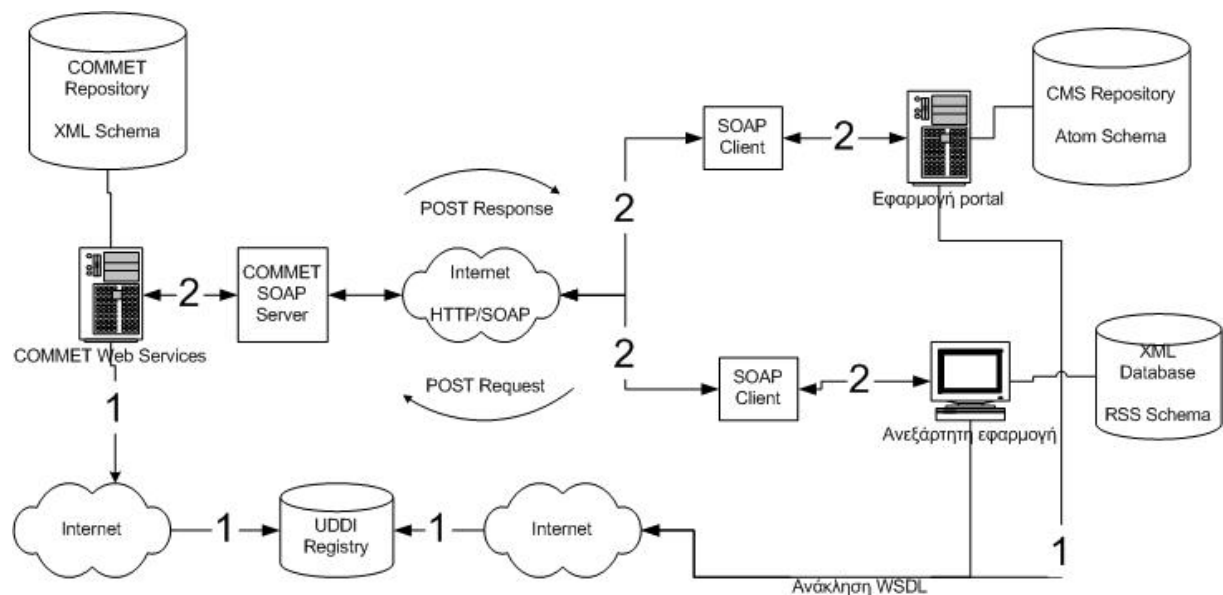
- Δημιουργία Web Service για την διανομή του περιεχομένου του COMMET υπό μορφή εγγράφων XML. Για κάθε μια δημοσιεύσιμη λειτουργία (μέθοδο) του COMMET δημιουργείται και μια μέθοδος στην Web Service. Η μέθοδος αυτή θα επιστρέφει τα αποτελέσματα κλήσης της λειτουργίας σε μορφή XML, με βάση το schema που της έχει αποδοθεί από το COMMET.
- Δημοσίευση των Schemas για τα αποτελέσματα όλων των λειτουργιών που το COMMET πρόκειται να δημοσιεύσει με την προαναφερθείσα Web Service. Τα XML έγγραφα που επιστρέφονται από τις λειτουργίες ακολουθούν το Schema του COMMET. Η δημοσίευση των schemas των λειτουργιών του COMMET επιτρέπει στον χρήστη των λειτουργιών αυτών να μπορεί να μετατρέψει τα επιστρεφόμενα XML έγγραφα στο schema που αυτός χρησιμοποιεί.
- Δημιουργία Web Service με μεθόδους που υλοποιούν εσωτερικά XSL μετασχηματισμούς από το schema του COMMET σε άλλα γνωστά format για syndication, όπως είναι το RSS 1.0, το RSS 2.0 και το Atom. Κάθε μια από τις μεθόδους αυτές θα πρέπει να δέχεται σαν είσοδο ένα έγγραφο XML που επιστρέφει η κλήση μιας λειτουργίας του COMMET, να το μετατρέπει στο κατάλληλο schema και να επιστρέφει την μετασχηματισμένη μορφή στην καλούσα εφαρμογή. Αυτό θα πρέπει να γίνει για κάθε δημοσιεύσιμη λειτουργία (μέθοδο) του COMMET. Δηλαδή αν οι δημοσιεύσιμες λειτουργίες του COMMET είναι τρεις και ενδιαφερόμαστε να εκδώσουμε τα αποτελέσματα τους σε μορφή RSS 1.0, RSS 2.0 και Atom, τότε πρέπει να υλοποιήσουμε έξι συνολικά μεθόδους στην Web Service. Η υλοποίηση των μετασχηματισμών σε γνωστά syndication format καθιστά γενικά την διανομή του περιεχομένου του COMMET πιο εύκολη. Τα αποτελέσματα όλων των παραπάνω μεθόδων μπορούν να ενσωματώνονται κάθε φορά στις σελίδες μιας εφαρμογής βασισμένης σε COMMET με το εικονίδιο , ώστε να είναι όλο το διαθέσιμο περιεχόμενο προσβάσιμο από το Web μέσω OPML και RSS ή Atom. Δηλαδή κάθε σελίδα θα έχει ένα feed με τους πόρους του τρέχοντος φακέλου και κάθε φάκελος θα μπορεί να παρουσιάσει με OPML ολόκληρο το υποδέντρο του.

Ένα σχήμα που περιγράφει καλύτερα τις παραπάνω εκτελεστέες εργασίες είναι το ακόλουθο:



Σχήμα 24: Αρχιτεκτονική για syndication περιεχομένου από το COMMET

Ένα σχήμα που μας βοηθά να κατανοήσουμε γενικά την επικοινωνία των εξωτερικών εφαρμογών με το COMMET μέσω των Web Services είναι το ακόλουθο:



Σχήμα 25 Δημοσίευση και διάθεση Web Services από το COMMET

Στο παραπάνω σχήμα παρατηρούμε τις βασικές οντότητες που εμπλέκονται στην επικοινωνία μεταξύ του COMMET και ενός εξωτερικού CMS ή μιας εξωτερικής εφαρμογής. Η επικοινωνία αυτή γίνεται σε δύο φάσεις. Πρώτα γίνεται δημοσίευση από το COMMET των διαθέσιμων Web Services που θα χρησιμεύσουν για syndication του περιεχομένου του. Η δημοσίευση γίνεται σε κάποιο γνωστό UDDI registry και οι πελάτες του COMMET αποκτούν έτσι στην διάθεσή τους το αρχείο .wsdl της υπηρεσίας που θέλουν να καλέσουν. Εν συνεχεία δημιουργούν μια εφαρμογή πελάτη SOAP, η οποία επικοινωνεί με την αντίστοιχη εφαρμογή διακομιστή SOAP στο COMMET. Με την επικοινωνία αυτή ανταλλάσσονται τα επιθυμητά XML έγγραφα μέσω SOAP.

Πριν προχωρήσουμε στην σχεδίαση πρέπει να μελετήσουμε προσεκτικά την οντολογία του COMMET. Το COMMET, όπως είδαμε στο κεφάλαιο 3, διαθέτει δύο βασικές κλάσεις: την κλάση «Φάκελος» και την abstract κλάση «Πόρος». Αυτές οι κλάσεις και οι υλοποιήσιμες υποκλάσεις τους διαθέτουν κάποιες ιδιότητες (properties). Κατά την κλήση των διαθέσιμων μεθόδων, το περιεχόμενο που επιστρέφεται είναι σε XML μορφή. Αν λοιπόν θέλουμε να πετύχουμε συμβατότητα με κάποιο syndication XML format, πρέπει να πετύχουμε μια αντιστοίχιση από το schema του περιεχομένου που επιστρέφει μια λειτουργία (μέθοδος) του COMMET, στο schema του format που θέλουμε να χρησιμοποιήσουμε για syndication. Στην παρούσα εργασία θα ασχοληθούμε με την ανάπτυξη XSL templates για μετατροπή από XML σε RSS/OPML μορφή. Θα μελετήσουμε την υλοποίηση με χρήση του format RSS 2.0 και του OPML 1.0.

### 7.3.2 Αντιστοίχιση οντοτήτων COMMET με RSS 2.0 και OPML 1.0.

Χρησιμοποιούμε το δημοφιλές RSS 2.0 ως syndication format σε συνδυασμό με το OPML 1.0. Το RSS δεν παρέχει δυνατότητα για αναπαράσταση ιεραρχικών δομών, όπως είναι η δεντρική δομή του COMMET. Ωστόσο το OPML παρέχει αυτή τη δυνατότητα για ιεραρχική παρουσίαση των καναλιών περιεχομένου. Αντιστοιχίζουμε το στοιχείο «κόμβος» του COMMET με το στοιχείο <channel> (κανάλι) του RSS και το «πόρος» του COMMET με το στοιχείο <item> του RSS.

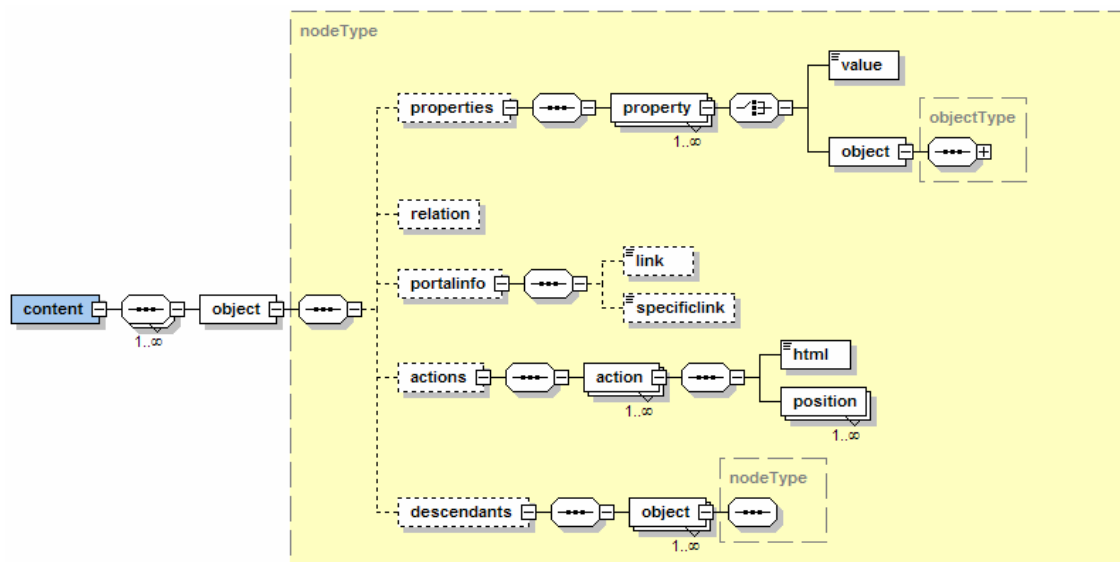
Έτσι υπάρχει η δυνατότητα να αναπαραστήσουμε αρχικά μια ιεραρχική δομή κόμβων με ένα αρχείο OPML. Στο αρχείο αυτό κάθε ιεραρχικά διατεταγμένο στοιχείο <outline> θα αναπαριστά ένα κόμβο περιεχομένου. Για κάθε κόμβο σχεδιάζεται το κατάλληλο RSS αρχείο που θα περιέχει όλους τους πόρους που περιλαμβάνει ο κόμβος αυτός. Στο αρχείο RSS ο κόμβος θα παριστάνεται με το στοιχείο <channel> και κάθε ξεχωριστός πόρος του με ένα στοιχείο <item>.

Με βάση την παραπάνω ανάλυση προκύπτει άμεσα ότι από τις διαθέσιμες λειτουργίες του COMMET που αναπτύξαμε στο Κεφάλαιο 3, οι λειτουργίες που μπορούν να υποστηριχθούν ώστε να εξασφαλίζουν συμβατότητα με το RSS format είναι οι εξής:

1. Εμφάνιση απογόνων ενός φακέλου - **PrintMySubtreeNodes**. Με την λειτουργία αυτή θα επιστρέφεται σε ένα αρχείο OPML όλο το υποδέντρο της ιεραρχικής δομής του Portal που έχει ως ρίζα τον συγκεκριμένο φάκελο.
2. Εμφάνιση των υποφακέλων ενός φακέλου - **PrintMyChildren**. Με την λειτουργία αυτή εμφανίζονται μόνο οι υποφάκελοι του συγκεκριμένου φακέλου.
3. Εμφάνιση πόρων φακέλου - **PrintMyResources**. Με την λειτουργία αυτή εμφανίζονται όλοι οι πόροι που περιέχονται στον συγκεκριμένο φάκελο.

#### 1. Απόγονοι φακέλου (**PrintMySubtreeNodes**).

Για την λειτουργία αυτή παραθέτουμε πάλι για ευκολία το XML schemas που επιστρέφει.



Στο παραπάνω αναδρομικό schema το στοιχείο properties και τα υποστοιχεία του είναι χρήσιμα στο syndication, γιατί αναπαριστούν τις ιδιότητες των κόμβων. Στο αρχείο OPML μερικά από τα properties ενός κόμβου αντιστοιχίζονται άμεσα με τα attributes του στοιχείου <outline> που περιγράφει τον συγκεκριμένο κόμβο. Για παράδειγμα, έστω ότι ο τρέχων φάκελος (ρίζα) είναι ο φάκελος «Μαθήματα» και έχει δύο υποφακέλους, «Βάσεις Δεδομένων» και «Πληροφοριακά Συστήματα». Ο φάκελος «Βάσεις Δεδομένων» έχει με την σειρά του 2 υποφακέλους, τους «Εργαστήριο» και «Θεωρία». Τότε η αναπαράσταση των κόμβων σε ένα αρχείο OPML μετά την εκτέλεση της λειτουργίας 1 «Απόγονοι φακέλου», θα έχει την παρακάτω μορφή. (χρησιμοποιούμε κατάλληλη στοίχιση για την ιεραρχία)

```
<?xml version="1.0" encoding="utf-8"?>
<opml version="1.0">
  <head>
    <title>PrintMySubtreeNodes</title>
    <ownerName>Stefanos</ownerName>
    <ownerEmail>evangelous@gmail.com</ownerEmail>
  </head>

  <body>

    <outline text="Μαθήματα" type="rss" version="RSS2" title=" Μαθήματα" description="Ο φάκελος ρίζα" xmlUrl=" " htmlUrl="">

      <outline text=" Βάσεις Δεδομένων" type="rss" version="RSS2" title=" Βάσεις Δεδομένων" description="" xmlUrl=" " htmlUrl="">
        <outline text="Εργαστήριο" type="rss" version="RSS2" title="Εργαστήριο" description="" xmlUrl=" " htmlUrl="" />
        <outline text="Θεωρία" type="rss" version="RSS2" title="Θεωρία" description="" xmlUrl=" " htmlUrl="" />
      </outline>

      <outline text=" Πληροφοριακά Συστήματα" type="rss" version="RSS2" title=" Πληροφοριακά Συστήματα" description=" " xmlUrl=" " htmlUrl="" />
    </outline>

  </body>
</opml>
```



Με κόκκινα γράμματα φαίνεται η αναπαράσταση του τρέχοντα υποκόμβου, δηλαδή του κόμβου για τον οποίο θέλουμε να εμφανίσουμε ολόκληρο το υποδέντρο. Επίσης με τονισμένα γράμματα φαίνονται τα δύο attributes που θα εμφανίζουμε για κάθε φάκελο. Όλα τα υπόλοιπα attributes που εμφανίζονται είναι ορισμένα από εμάς και μπορούν να παραληφθούν. Με βάση το παραπάνω έγγραφο και έχοντας υπ’ όψη τις διαθέσιμες ιδιότητες για κάθε κόμβο του COMMET (βλ. Οντολογία COMMET στο Κεφάλαιο 3) κάνουμε τις παρακάτω αντιστοιχίσεις:

**Πίνακας 3: Αντιστοίχιση COMMET – OPML για κάθε κόμβο του COMMET**

OPML <outline> attributes	Ιδιότητες COMMET
Text	nid
Title	nname
Description	ndescr
htmlUrl	savepath

Παρατηρούμε ότι μπορούμε να εκφράσουμε όλες τις ιδιότητες όλων των υποκόμβων με το OPML αρχείο, μόνο αν ορίσουμε δικά μας πρόσθετα attributes μέσα στο element <outline>. Όμως κάτι τέτοιο δεν θα εξασφάλιζε συμβατότητα με άλλα συστήματα που χρησιμοποιούν τα standard tags του OPML. Για τον λόγο αυτό κάνουμε χρήση μόνο των διαθέσιμων attributes που φαίνονται στον πίνακα 3, τα οποία αντιστοιχίζονται με τις ιδιότητες κόμβου του COMMET που θέλουμε να εμφανίσουμε. Η έκφραση όλων των ιδιοτήτων του κάθε κόμβου μαζί με όλους τους πόρους που περιέχει μπορεί όμως να γίνει από την λειτουργία 3 του COMMET: “Εμφάνιση πόρων φακέλου”, αν κανείς γνωρίζει την τιμή της ιδιότητας nid του κόμβου που τον ενδιαφέρει.

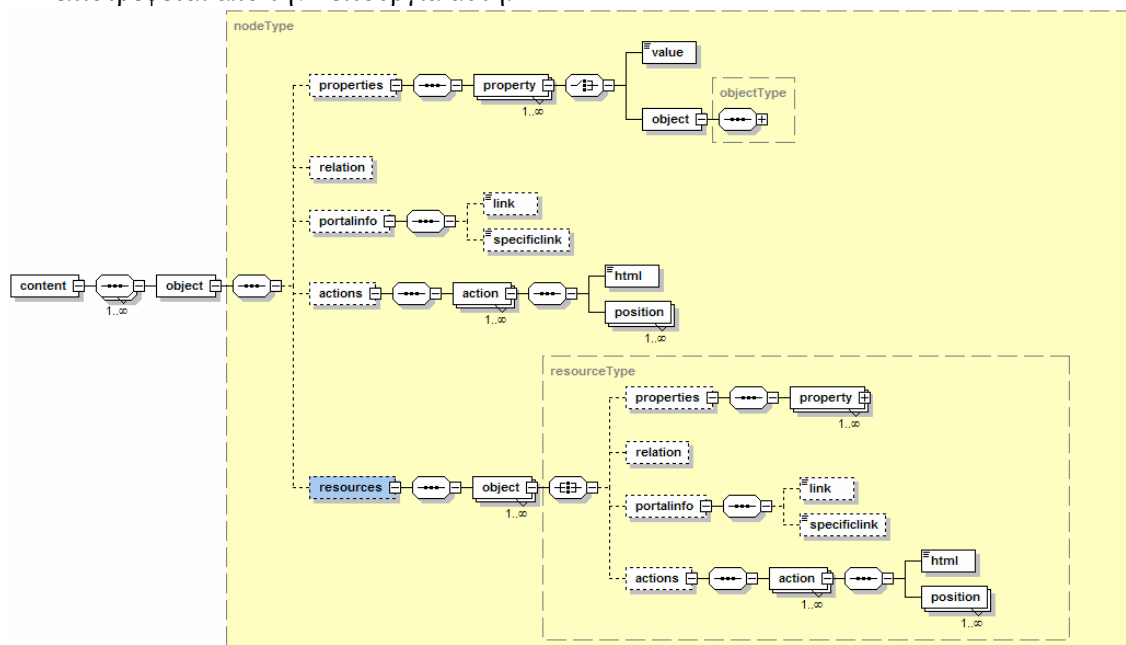
Επεκτείνουμε τον παραπάνω πίνακα, ώστε να κάνουμε μια πλήρη αντιστοίχιση μεταξύ όλων των οντοτήτων του COMMET στην λειτουργία 1 (**PrintMySubtreeNodes**) και όλων των οντοτήτων που προσφέρει το schema του OPML 1.0. Βασιζόμαστε πάντα στα διαγράμματα που εκφράζουν τα schemas για το COMMET και για το OPML αντίστοιχα. Έτσι προκύπτει ο παρακάτω πίνακας, τον οποίο και θα χρησιμοποιήσουμε για τον σχεδιασμό του πρώτου XSLT stylesheet. Με κόκκινο τονίζεται κάθε στοιχείο που μπορούμε να ορίσουμε προαιρετικά.

**Πίνακας 4: Τελική αντιστοίχιση COMMET-OPML για την λειτουργία PrintMySubtreeNodes**

OPML 1.0	PrintMySubtreeNodes.xsd
opml	content
outline	<b>object (τρέχων κόμβος)</b>
--	properties (ιδιότητες του τρέχοντα κόμβου)
--	property
--	value
Text	nid
Title	nname
Description	ndescr
htmlUrl	savepath
--	relation
--	portalinfo
--	link

--	specificlink
--	actions
--	action
--	html
--	position
--	<b>descendants</b>
	object (υποκόμβος)

- Εμφάνιση των υποφακέλων ενός φακέλου (**PrintMyChildren**). Η λειτουργία 2 από τον ορισμό της είναι απλά υποπερίπτωση της λειτουργίας 1. Άρα η λειτουργία αυτή δεν χρειάζεται να μας απασχολήσει περαιτέρω.
- Εμφάνιση πόρων φακέλου (**PrintMyResources**). Επαναλαμβάνουμε για ευκολία το schema που επιστρέφεται από την λειτουργία αυτή.



Στο παραπάνω αναδρομικό schema μόνο το στοιχείο `<object>` → `<properties>` και το στοιχείο `<resources>` → `object` → `<properties>` είναι χρήσιμα στο syndication, γιατί αναπαριστούν τις ιδιότητες του τρέχοντα κόμβου και τις ιδιότητες των πόρων του κόμβου αντίστοιχα. Τα `properties` ενός κόμβου αντιστοιχίζονται άμεσα με τα `elements` του αρχείου RSS που περιγράφει τον συγκεκριμένο κόμβο. Οι δε ιδιότητες των πόρων του κόμβου αντιστοιχίζονται άμεσα με τα υποστοιχεία του στοιχείου `<item>` που περιγράφει τον πόρο στο αρχείο RSS.

Με βάση τις παραπάνω διαπιστώσεις και έχοντας υπ' όψη τις διαθέσιμες ιδιότητες για κάθε κόμβο και πόρο του COMMET (βλ. Οντολογία COMMET στο Κεφάλαιο 3) κάνουμε τις παρακάτω αντιστοιχίσεις (Με όσα στοιχεία του RSS δεν υπάρχει αντιστοίχιση δεν υφίσταται πρόβλημα διότι δεν είναι υποχρεωτικά):

**Πίνακας 5: Αντιστοίχιση RSS <channel> - COMMET**

<b>RSS &lt;channel&gt; elements</b>	<b>Ιδιότητες κόμβων COMMET</b>
Title (υποχρεωτικό)	nname
Link (υποχρεωτικό)	nid
Description (υποχρεωτικό)	ndescr
Language	--
copyright	--
ManagingEditor	--
Webmaster	ncreator
PubDate	--
LastBuildDate	ncreated
Category	nstype
Generator	--
Docs	--
Cloud	--
ttl	--
image	--
rating	--
TextInput	--
SkipHours	--
SkipDays	--

**Πίνακας 6: Αντιστοίχιση RSS <item> - COMMET**

<b>RSS &lt;item&gt; elements</b>	<b>Ιδιότητες πόρων COMMET</b>
title (υποχρεωτικό)	rname
Link	rid
Description (υποχρεωτικό)	rdescr
Author	rowner
Category	rstype
comments	rublished
enclosure	rsmlltext
Guid	--
PubDate	rposted
source	--

Να σημειωθεί ότι με τις ιδιότητες `nid` και `rid` προσδιορίζουμε κατ'ουσία την φυσική θέση του κόμβου και του πόρου αντίστοιχα. Άρα η ιδιότητα αυτή αντιστοιχεί με το στοιχείο `link` που δείχνει στην φυσική θέση του `channel` ή `item` αντίστοιχα. Επίσης αντιστοιχίζουμε κατά σύμβαση το στοιχείο `comments` με την ιδιότητα `rublished`, σχετικά με το αν ένας πόρος είναι δημοσιεύσιμος ή όχι.

Κάθε μια από τις παραπάνω μεθόδους επιστρέφει ένα έγγραφο XML με τα αποτελέσματα. Τα αποτελέσματα αυτά υπακούουν σε ένα XML Schema, διαφορετικό προφανώς για κάθε μέθοδο. Τα αποτελέσματα της πρώτης μεθόδου θέλουμε να μετατραπούν σε μορφή OPML για την ιεραρχική παρουσίαση του υποδέντρου των κόμβων. Τα αποτελέσματα της δεύτερης μεθόδου θέλουμε να είναι σε μορφή RSS 2.0. Έχοντας τα ανεξάρτητα XML Schemas του COMMET διαθέσιμα και γνωρίζοντας πολύ καλά τα Schemas των OPML και RSS 2.0 μπορούμε να σχεδιάσουμε XSL templates για να μετατρέπουμε αυτόματα τα έγγραφα που επιστρέφει το COMMET στην μορφή που επιθυμούμε.

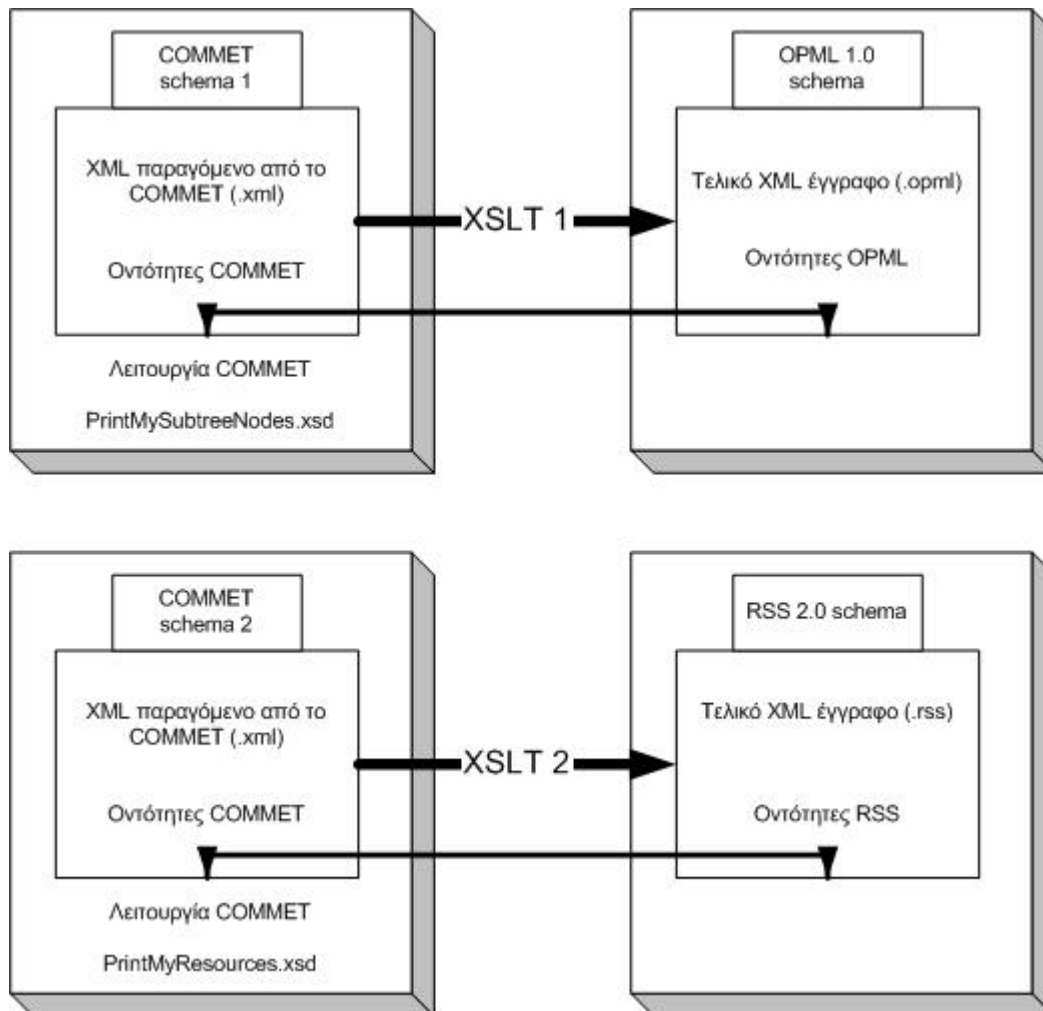
Επεκτείνουμε τους παραπάνω δύο πίνακες, ώστε να κάνουμε μια πλήρη αντιστοίχιση μεταξύ όλων των οντοτήτων του COMMET στην λειτουργία 3 (**PrintMyResources**) και όλων των οντοτήτων που προσφέρει το schema του RSS 2.0. Βασιζόμαστε πάντα στα διαγράμματα που εκφράζουν τα schemas για το COMMET και για το RSS 2.0 αντίστοιχα. Έτσι προκύπτει ο παρακάτω πίνακας, τον οποίο και θα χρησιμοποιήσουμε για τον σχεδιασμό του δεύτερου XSLT stylesheet.

**Πίνακας 7: Τελική αντιστοίχιση μεταξύ COMMET και RSS2 για την λειτουργία PrintMyResources**

RSS 2.0	PrintMyResources.xsd
<b>channel</b>	<code>content</code>
Title (υποχρεωτικό)	<code>nname</code>
Link (υποχρεωτικό)	<code>nid</code>
Description (υποχρεωτικό)	<code>ndescr</code>
Language	--
copyright	--
ManagingEditor	--
Webmaster	<code>ncreator</code>
PubDate	--
LastBuildDate	<code>ncreated</code>
Category	<code>ntype</code>
<b>item</b>	<b>object</b>
--	<b>resources</b>
title (υποχρεωτικό)	<code>rname</code>
Link	<code>rid</code>
Description (υποχρεωτικό)	<code>rdescr</code>
Author	<code>rowner</code>
Category	<code>rtype</code>
comments	<code>rublished</code>
enclosure	<code>rsmalltext</code>
Guid	--
PubDate	<code>rposted</code>

### 7.3.3 Σχεδίαση XSLT stylesheet

Σχηματικά η μετατροπή από τα έγγραφα που παράγει το COMMET σε έγγραφα τύπου OPML και RSS2 αντίστοιχα φαίνεται παρακάτω:



**Σχήμα 26: XSLT stylesheets για το COMMET**

Με βάση το παραπάνω σχήμα και τους πίνακες αντιστοίχισης 4 και 7, οι οποίοι δίνουν τις τελικές αντιστοιχίσεις οντοτήτων μεταξύ των schemas, παράγουμε τα δύο ζητούμενα XSLT stylesheets. Τα δύο αυτά stylesheets μπορούν να ενσωματωθούν στην Web Service που παράγει τα τελικά αποτελέσματα της πρώτης λειτουργίας COMMET σε .opml και της δεύτερης λειτουργίας σε .rss αντίστοιχα. Ο κώδικας των δύο XSLT stylesheets εκτίθεται στο παράρτημα Ε.

Για καλύτερη εποπτική παρουσίαση των αποτελεσμάτων από την εφαρμογή των XSLT εκθέτουμε αμέσως παρακάτω για κάθε μια από τις λειτουργίες του COMMET:

- Τα περιεχόμενα ενός αρχείου .xml που επιστρέφεται από την κλήση της λειτουργίας
- Το αντίστοιχο αρχείο .rss ή .opml (ανάλογα με την λειτουργία) που προκύπτει μετά την εφαρμογή του κατάλληλου XSLT stylesheet στο .xml που επιστρέφει η λειτουργία

## Λειτουργία *PrintMySubtreeNodes*

The screenshot displays a complex hierarchical tree structure within a software application. The root node is 'content', which contains an 'object' node. This 'object' node has several properties: 'classification' (node), 'class' (team), and 'visible' (false). It also contains a 'properties' section with a 'property (8)' entry, a 'relation level=0', and 'actions'. The 'descendants' section of the root 'object' contains two 'object (2)' nodes. Each of these nodes is represented as a table with columns for 'classification', 'class', 'visible', 'properties', 'relation', 'actions', and 'descendants'. The first 'object (2)' node has two rows: row 1 has 'node', 'folder', 'true', 'properties', 'relation level=1', and 'actions'; row 2 has 'node', 'folder', 'true', 'properties', 'relation level=1', and 'actions'. The 'descendants' column of the first row contains a nested 'object' node with properties: 'classification' (node), 'class' (theme), 'visible' (true), 'properties', 'relation level=2', and 'actions'. The second 'object (2)' node also has two rows, both with 'node', 'meeting', 'true', 'properties', 'relation level=2', and 'actions' in their respective columns. The 'descendants' column of the first row of this second node contains another 'object (2)' node, which is a table with two rows: row 1 has 'node', 'meeting', 'true', 'properties', 'relation level=2', and 'actions'; row 2 has 'node', 'meeting', 'true', 'properties', 'relation level=2', and 'actions'.

XML			
version	1.0		
encoding	UTF-8		
opml			
head			
t.Trexon kombos me olakliro to ypodentro toy			
body			
outline			
text	1968		
title	Onoma kombou		
description	Perigrafi tou kombou		
outline (2)			
	text	title	description
1	2460	title1	
outline (1)			
	text	title	description
1	2591	XSL Configuration	description 1
2	2461	title2	Description 2
outline (2)			
	text	title	description
1	2621	title 1	description 1
2	2625	title 2	description 2

## Λειτουργία PrintMyResources

content

object

- classification node
- class blackboard
- visible true
- properties
  - property (7)
- relation level=0
- portalinfo
- actions
- resources
  - object (4)
 

	classification	class	properties	relation	portalinfo	actions
1	resource	anndocument	properties	relation state=main order=9 visibility=-1	portalinfo	actions
2	resource	annusertext	properties	relation state=main order=11 visibility=-1	portalinfo	actions
3	resource	annusertext	properties	relation state=main order=16 visibility=-1	portalinfo	actions
4	resource	annusertext	properties	relation state=main presentation=important order=18 visibility=-1		actions



XML							
= version		1.0					
= encoding		UTF-8					
rss							
= version		2.0					
channel							
link		1970					
Category		blackboard					
title		blackboard title					
webMaster		our webmaster					
LastBuildDate		14/6/2004					
item (4)							
	link	category	comments	title	author	PubDate	description
1	845	annusertext	-1	title1	author1	19/12/2003	description1
2	846	annusertext	-1	title2	author2	18/12/2003	description2
3	748	annusertext	-1	title3	author3	8/6/2003	description3
4	1074	annusertext	-1	title4	author4	22/12/2004	

## 7.4 Το μέλλον για τις τεχνολογίες aggregation-syndication

### 7.4.1 Μειονεκτήματα – παράμετροι υπό βελτίωση

#### *Διαχείριση εύρους ζώνης*

Ένα από τα βασικότερα προβλήματα που παρουσιάζονται είναι η κλιμάκωση των εφαρμογών syndication όταν έχουμε να κάνουμε με έναν μεγάλο αριθμό clients που ζητούν ταυτόχρονα περιεχόμενο από τον server. Πλέον εκτός από το περιεχόμενο που ζητούν οι Web clients, το οποίο αφορά στις ιστοσελίδες, υπάρχει μια ραγδαία αυξανόμενη ζήτηση για syndicated περιεχόμενο, ειδικά στα μεγάλα ειδησεογραφικά sites. Για τέτοιες περιπτώσεις ο σχεδιασμός του RSS δεν είναι ο βέλτιστος δυνατός και ενδέχεται να προκαλούνται προβλήματα στην λειτουργία ενός server αν εκατομμύρια πελατών ζητά περιεχόμενο για ένα συγκεκριμένο χρονικό διάστημα. Αναφορές αναφέρουν ότι τέτοιες μαζικές κλήσεις διαθέτουν όλα τα χαρακτηριστικά μιας επίθεσης ασφάλειας τύπου Denial of Service (DoS).

Το πρόβλημα είναι ότι οι RSS clients πραγματοποιούν κλήσεις για ανάκτηση περιεχομένου, ακόμα και αν το ζητούμενο περιεχόμενο δεν έχει ανανεωθεί από το προηγούμενο download. Η παραπάνω αδυναμία είναι μια ένδειξη ότι το πρωτόκολλο επιδέχεται βελτιώσεις ούτως ώστε να βελτιωθεί η «ευφυΐα» που σχετίζεται με την διαχείριση των συνδέσεων για αναζήτηση και ανάκτηση περιεχομένου.

#### *Συμβατότητα*

Παρά την ευρεία χρήση των πρωτοκόλλων RSS και Atom, εξακολουθούν να υφίστανται πάρα πολλά διαφορετικά formats για syndication και aggregation, από τα οποία μόνο μερικά είναι επίσημα αναγνωρισμένα ως πρότυπα. Στην περίπτωση του RSS οι διάφορες εκδόσεις του δεν παρέχουν συμβατότητα προς παλιότερες υλοποιήσεις. Σχετικά με το κατά πόσον αυτό αποτελεί δεσμευτικό παράγοντα για την ευρύτερη υιοθέτηση του RSS, υπάρχουν δύο επικρατούσες απόψεις. Σύμφωνα με την πρώτη άποψη, οι πολλές διαφορετικές εκδόσεις του RSS και η έλλειψη υποστήριξης για τις προηγούμενες εκδόσεις είναι στοιχεία που προκαλούν σύγχυση και ασυμβατότητα. Σύμφωνα με την δεύτερη άποψη, μετά την έκδοση 0.9 του RSS, όλες οι εκδόσεις είναι πλήρως συμβατές με το XML 1.0 specification και συνεπώς μπορεί να επιτευχθεί συμβατότητα με την χρήση των κατάλληλων μετασχηματισμών XSLT, τους οποίους κάθε πρόγραμμα RSS aggregator πρέπει να εφαρμόσει στο περιεχόμενο ώστε να το καταστήσει αναγνώσιμο από τους χρήστες, ασχέτως του σε ποια έκδοση RSS έχει γίνει η κωδικοποίηση.

Ειδικά για το θέμα της συμβατότητας μεταξύ διαφορετικών εκδόσεων του RSS, ο αναγνώστης μπορεί να ανατρέξει στο παράρτημα Β.

## 7.4.2 Συμπερασματικά

### Γενικά

Παρ' όλα τα πιθανά προβλήματα, είναι γεγονός ότι οι τεχνολογίες aggregation και syndication είναι πολλά υποσχόμενες σχετικά με την δημοσίευση περιεχομένου μέσω του παγκόσμιου ιστού. Τα πολλά πλεονεκτήματα από την εφαρμογή των τεχνολογιών αυτών υπερκαλύπτουν τα όποια μειονεκτήματά τους. Η ανάπτυξη web sites, portals, intranets, extranets, blogs και wikis εξελίσσονται με ραγδαίους ρυθμούς. Τα CMS είναι ο θεμελιώδης λίθος στον οποίο βασίζεται η υλοποίηση και η διαχείριση όλων των παραπάνω συστημάτων. Η διαχείριση περιεχομένου μπορεί να παρομοιαστεί με ένα «όχημα» που μας οδηγεί στην ενοποίηση της διαχείρισης και αμοιβαίας ανταλλαγής των πληροφοριών. Για το όχημα αυτό, η γλώσσα XML μαζί με τις Web Services είναι τα «καύσιμα υλικά».

Η διανομή του περιεχομένου καθίσταται πρακτική όταν με την εφαρμογή ενός καλά ορισμένου schema (για παράδειγμα το RSS 2.0) χρησιμοποιηθούν οι Web Services για την δημιουργία διαύλων ανταλλαγής περιεχομένου μεταξύ οποιονδήποτε συστημάτων. Ένα CMS που χρησιμοποιεί Web Services για syndication και aggregation περιεχομένου μπορεί να αποτελέσει το κεντρικό κομβικό σημείο σε μια αρχιτεκτονική εφαρμογής πολλαπλών επιπέδων. Στην αρχιτεκτονική αυτή η εισαγωγή και εξαγωγή περιεχομένου προς και από το CMS αντίστοιχα θα μπορούν να γίνονται ανεξαρτήτως του λογισμικού υποδομής κάθε εμπλεκόμενου συστήματος.

Η διαχείριση και διανομή του περιεχομένου με syndication θα αρχίζει να γίνεται ακόμα πιο ενδιαφέρουσα και θα αποκτά περισσότερες εφαρμογές όταν αναπτυχθούν σε ικανοποιητικό βαθμό μια σειρά από υποστηρικτικές τεχνολογίες. Μερικές από τις τεχνολογίες αυτές είναι το Tablet PC, οι ασύρματες συσκευές και τα ευρυζωνικά ασύρματα δίκτυα καθώς και η αναγνώριση γραφής και ομιλίας από τους Η/Υ. Παράλληλα με την εφαρμογή των προτάσεων του W3C Semantic Web (<http://www.w3.org/2001/sw/>) ίσως να γίνει εφικτή η ενσωμάτωση «ευφυΐας» στα CMS με στόχο την αυτόματη αναζήτηση, σημασιολογική ερμηνεία και aggregation και syndication περιεχομένου και υπηρεσιών.

### Καθιέρωση πρότυπων syndication schemas

Το δημοφιλέστερο format για content syndication αυτή τη στιγμή είναι το RSS, ωστόσο δεν είναι απόλυτα σίγουρο ποιες τάσεις θα επικρατήσουν τελικά στην κοινότητα του Διαδικτύου. Γεγονός είναι ότι το RSS χαίρει ευρείας υποστήριξης και οι προδιαγραφές του έχουν φτάσει σε μια σταθερή έκδοση. Αυτή τη στιγμή βρίσκεται στην έκδοση 2.0.2. Το πρωτόκολλο Atom από την άλλη πλευρά έρχεται να γεφυρώσει τις ασυμβατότητες μεταξύ των εκδόσεων 1.0 (RDF) και 2.0 του RSS. Το Atom αποτελεί μια συνολική πρόταση για το μέλλον και σαν syndication schema και σαν Weblog API. Για το λόγο αυτό μελετάται σοβαρά τόσο από το IETF όσο και από το W3C η υιοθέτησή του ως πρότυπο.

Προς το παρόν τα συστήματα ανάγνωσης και επεξεργασίας news feeds υποστηρίζουν τόσο το Atom όσο και το RSS, πολλές φορές και όλες τις ασύμβατες μεταξύ τους εκδόσεις του RSS. Αυτή τη στιγμή η κατάσταση είναι κάπως ρευστή. Κρίνεται ότι το επόμενο έτος θα είναι καθοριστικό σχετικά με τη καθιέρωση προτύπων στην αναπτυσσόμενη αγορά του content syndication.

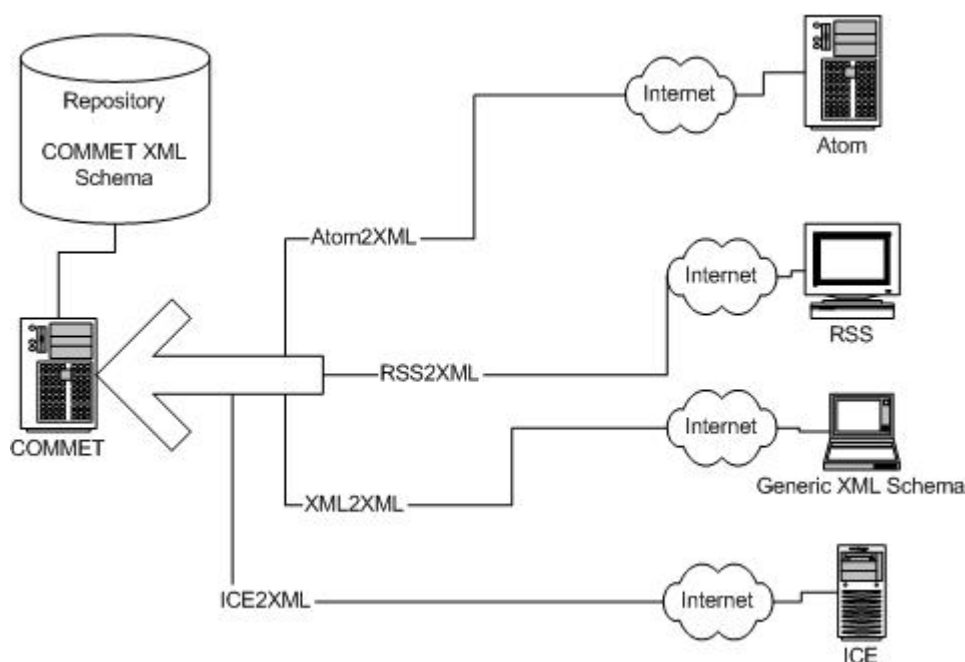
## 8 Επεκτάσεις της παρούσας εργασίας

### 8.1 Επέκταση του syndication στο COMMET με τα υπόλοιπα γνωστά schemas

Για τον σχεδιασμό XSL stylesheet χρησιμοποιήσαμε το RSS 2.0 ως format εξόδου. Θα μπορούσαμε να χρησιμοποιήσουμε είτε το RSS 1.0 που προσφέρει την ευέλικτη λειτουργικότητα Dublin Core, είτε το Atom, που προσφέρει δυνατότητα για ιεραρχία στα feed elements, άρα και ιεραρχία στην αναπαράσταση των πόρων του COMMET. Ο σχεδιασμός των υπόλοιπων XSLT stylesheets καθώς και μια έρευνα για τα πλεονεκτήματα και τα μειονεκτήματα κάθε μιας υλοποίησης θα ήταν πολύ χρήσιμη.

### 8.2 Aggregation περιεχομένου από το COMMET

Μπορούν να υλοποιηθούν μια σειρά από XSL templates που θα παρέχουν στο COMMET δυνατότητα για ενσωμάτωση περιεχομένου από εξωτερικές πηγές. Οι πηγές αυτές ενδέχεται να χρησιμοποιούν κάποιο από τα γνωστά syndication format ή και κάποιο δικό τους ανεξάρτητο Schema, το οποίο όμως θα είναι καθορισμένο και δημοσιευμένο στο Web. Παράλληλα μπορεί να υλοποιηθεί ένα απλό πρόγραμμα news aggregator, το οποίο όταν ενσωματωθεί σε σελίδα μιας εφαρμογής του COMMET να μπορεί να συλλέγει περιεχόμενο από κάθε πηγή που διαθέτει σε μορφή RSS. Εφαρμογές ενός τέτοιου χαρακτηριστικού είναι πάρα πολλές και αφορούν ειδικότερα στην ενημέρωση των τελικών χρηστών για ανανεώσεις σελίδων περιεχομένου από διάφορες πηγές αυτόματα, χωρίς την ανάγκη ξεχωριστής παρακολούθησής τους.



Σχήμα 27 Aggregation περιεχομένου από εξωτερικές πηγές

Το παραπάνω σχήμα μας δείχνει την φιλοσοφία υλοποίησης μετασχηματισμών για την εισαγωγή περιεχομένου στο schema του COMMET. Εναλλακτικά μπορούμε να αποφύγουμε την δημιουργία των XSLT μετασχηματισμών αν δημιουργήσουμε μεθόδους στο COMMET που θα δέχονται σαν είσοδο ένα έγγραφο XML του οποίου το Schema είναι γνωστό και θα δίνουν σαν έξοδο ένα έγγραφο XML συμβατό με το XML Schema του COMMET. Οι μέθοδοι αυτές που θα υλοποιηθούν στο COMMET μπορούν να βασίζονται στο DOM, ένα αντικειμενοστρεφές API για επεξεργασία εγγράφων XML.

### ***8.3 Application syndication***

Μια σημαντική εφαρμογή των Web Services είναι η έννοια του application syndication. Με την έννοια αυτή γίνεται μια επέκταση του syndication για την διάθεση και αξιοποίηση όχι μόνο περιεχομένου αλλά και διαδικτυακών εφαρμογών (web applications). Έτσι εταιρείες και κατασκευαστές μπορούν να διαθέσουν τις εφαρμογές τους για χρήση από εφαρμογές πελάτες (συνήθως web-based). Η επαναχρησιμοποίηση ολόκληρων εφαρμογών ή και τμημάτων αυτών, γενικά επεκτείνει τις δυνατότητες ενός φορέα παροχής περιεχομένου. Επίσης βοηθά τους δημιουργούς portal στο να ενσωματώνουν από εξωτερικές πηγές στα portal τους προσωποποιημένο περιεχόμενο μαζί με την λειτουργικότητα (εφαρμογές) που το συνοδεύει. Έτσι η λειτουργικότητα από ένα site, όπως φόρμες και προσωποποιημένα μενού, μπορούν να ενσωματωθούν σε ένα άλλο site, τροποποιώντας την εμφάνισή τους ώστε να ταιριάζουν με το στυλ του site στο οποίο ενσωματώνονται. Η έννοια του application syndication βασίζεται στις διαδραστικές (interactive) Web Services.

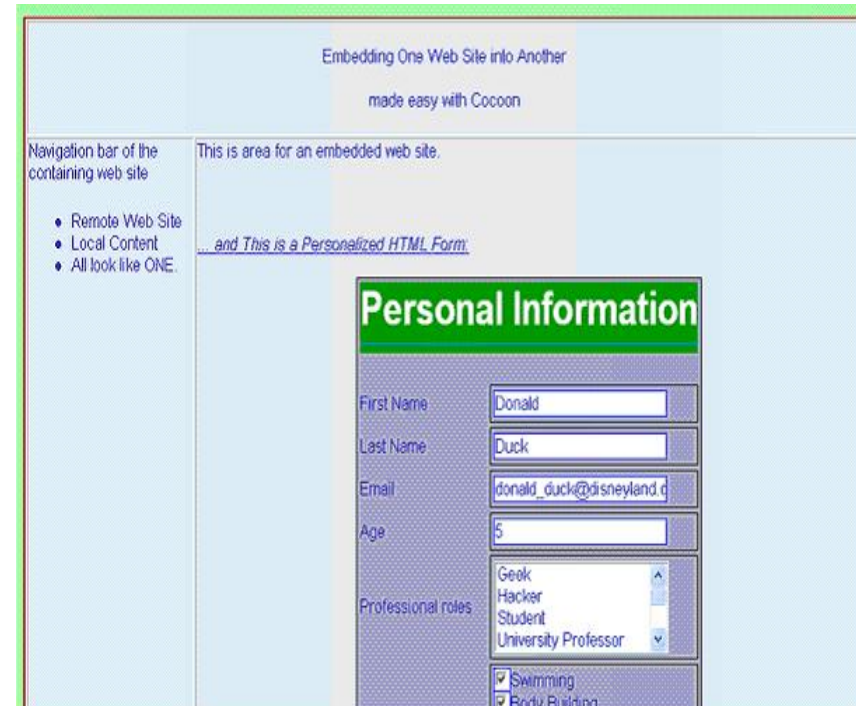
Στα παρακάτω σχήματα παρουσιάζεται μια ανεξάρτητη εφαρμογή, η οποία ενσωματώνεται με την μορφή Web Service σε ένα άλλο site.



The image shows a standalone web form titled "Personal Information" with a blue header. The form contains the following fields and options:

- First Name:
- Last Name:
- Email:
- Age:
- Professional roles: A dropdown menu with options: Geek, Hacker, Student, University Professor. "Geek" is selected.
- Hobbies: A list of checkboxes with the following checked options: Swimming, Body Building, Skiing. Other options include Running, Football, Reading, Writing, Soccer, and Blogging.
- A "Next" button is located at the bottom of the form.

Σχήμα 28 Ανεξάρτητη εφαρμογή συλλογής προσωπικών στοιχείων χρήστη [COC04]



The image shows the "Personal Information" form embedded within a larger website. The surrounding website has a light blue background and a navigation bar. The embedded form is titled "Personal Information" with a green header and contains the same data as in Figure 28:

- First Name:
- Last Name:
- Email:
- Age:
- Professional roles: A dropdown menu with options: Geek, Hacker, Student, University Professor. "Geek" is selected.
- Hobbies: A list of checkboxes with the following checked options: Swimming, Body Building. Other options include Running, Football, Reading, Writing, Soccer, and Blogging.

Text on the page includes: "Embedding One Web Site into Another made easy with Cocoon", "Navigation bar of the containing web site", "This is area for an embedded web site.", and a list of bullet points: "Remote Web Site", "Local Content", "All look like ONE." Below the list is the text "...and This is a Personalized HTML Form."

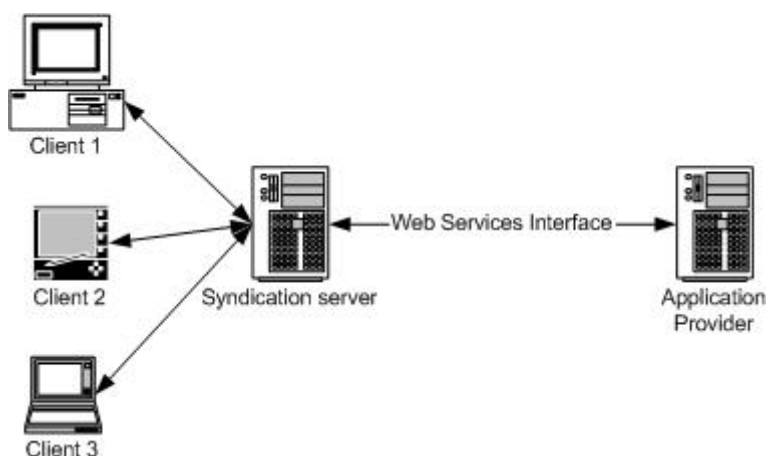
Σχήμα 29 Η παραπάνω εφαρμογή ενσωματωμένη σε ένα website [COC04]

Ο πάροχος εφαρμογών (application provider) διαθέτει τις εφαρμογές του υπό μορφήν Web Services. Για να είναι μια εφαρμογή διαθέσιμη για syndication, ο κατασκευαστής υιοθετεί ειδικά tag attributes για τις web σελίδες που κάνουν χρήση της εφαρμογής του, ώστε να αναγνωρίζονται τα τμήματα της σελίδας στα οποία ενσωματώνεται η εφαρμογή. Η δημοσίευση των Web Services ανακοινώνεται σε κάποιο UDDI ή private registry ή μπορεί και να γνωστοποιείται στέλνοντας απευθείας το αρχείο WSDL της υπηρεσίας στον πελάτη. Ο πελάτης ζητά από τον provider container web σελίδες. Στις σελίδες αυτές τοποθετούν HTML tags στο σημείο όπου θέλουν να καλέσουν και να ενσωματώσουν σε πραγματικό χρόνο μια Web Service. Σε κάποιο ανεξάρτητο site εκτελείται παράλληλα ένας ειδικός διακομιστής (syndication server). Να σημειωθεί επίσης ότι οι interactive Web Services μπορούν να υποστηρίξουν μεθόδους για Transactions με βάσεις δεδομένων (Transactional Web Services).

Για την ανάκληση των interactive Web Services έχει προταθεί το εξής βασικό σχήμα:

- Αρχικά ο πελάτης στέλνει ένα HTTP Request μήνυμα με το οποίο ζητάει και κάνει download την container HTML σελίδα.
- Μετά στέλνει άλλο request για την Web Service που επιθυμεί να ενσωματώσει στην σελίδα του. Το μήνυμα αυτό δρομολογείται στον syndication server.
- Με την σειρά του ο syndication server στέλνει μήνυμα στην δικτυακή εφαρμογή του provider για την αποστολή της τελικής HTML σελίδας.
- Χρησιμοποιώντας τα βοηθητικά tags για την αναγνώριση της ζητούμενης Web Service, ο server επεξεργάζεται την σελίδα και την επιστρέφει στον τελικό χρήστη.

Να σημειωθεί ότι ενώ ο application server βρίσκεται στο site του provider της εφαρμογής ή του portal, ο syndication server μπορεί να βρίσκεται σε κάποιο partner site, δηλαδή να φιλοξενείται από κάποιο συνεργάτη (partner) του provider.



**Σχήμα 30 Application Syndication**

Η έννοια του Application Syndication βρίσκεται ακόμα σε επίπεδο έρευνας. Οι βασικότερες προτάσεις στις οποίες βασίζεται είναι οι εξής:

- Web Services Experience Language (WSXL) [WSE04] και Web Services for Interactive Applications (WSIA) [WSA04]. Ορίζουν ένα πλαίσιο (framework) για δημιουργία interactive Web Services.
- Web Services Inspection Language (WSIL). Ορίζει την δημιουργία εγγράφων .wsil τα οποία θα περιλαμβάνουν μια λίστα με τις διαθέσιμες Web Services και τα αντίστοιχα αρχεία περιγραφής .wsdl. Το αρχείο αυτό τοποθετείται σε ένα προσπελάσιμο σημείο στο site που προσφέρει τις interactive Web Services. Παράλληλα προτείνεται και η δημιουργία μηχανών αναζήτησης Web Services, οι οποίες θα σαρώνουν το περιεχόμενο των αρχείων .wsil και θα τα εμφανίζουν στα αποτελέσματα μιας αναζήτησης.
- Web Services for Remote Portals (WSRP) [WSP04]. Επιτρέπει την ενσωμάτωση λειτουργικότητας από ένα portal σε άλλο.

- WS-AtomicTransaction, WS-Coordination, WS-BusinessActivity. Τα specifications αυτά έχουν προταθεί για πρότυπα από ιδιωτικούς φορείς και σχετίζονται με την υλοποίηση transactional Web Services [ION04].

Για την πρακτική υλοποίηση του Application Syndication έχει προταθεί μια αρχιτεκτονική που βασίζεται στην πλατφόρμα Apache Cocoon και η οποία μπορεί να εφαρμοστεί πρακτικά και σε άλλες πλατφόρμες [COC04]. Με βάση τα όσα ειπώθηκαν σε αυτή την παράγραφο, μια μελλοντική ενδιαφέρουσα εργασία μπορεί να είναι η δημιουργία Web Services για syndication των εφαρμογών του COMMET σε εξωτερικές εφαρμογές.



# **ΠΑΡΑΡΤΗΜΑ Α**

## **Πρωτόκολλα για syndication**

### **Σύγκριση μεταξύ των διαφορετικών εκδόσεων του RSS**

Υπάρχουν πολλές διαθέσιμες εκδόσεις του RSS, που εκδόθηκαν κυρίως από τις εταιρείες Netscape και Userland. Οι περισσότερες από τις εκδόσεις αυτές διαφοροποιούνται ελαφρώς ως προς τα elements και attributes που διαθέτουν και συνεπώς είναι σε ένα βαθμό ασύμβατες μεταξύ τους. Κάθε ένα από τα παραδείγματα που ακολουθούν προβάλλεται τόσο υπό μορφή XML όσο και υπό μορφή πινάκων (grid). Η δημιουργία της οπτικοποίησης σε μορφή grid έγινε με το λογισμικό XML Spy.

### Παράδειγμα αρχείου Netscape RSS 0.90

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns="http://my.netscape.com/rdf/simple/0.9/">
<channel>
<title>Mozilla Dot Org</title>
<link>http://www.mozilla.org</link>
<description>the Mozilla Organization web site</description>
</channel>
<image>
<title>Mozilla</title>
<url>http://www.mozilla.org/images/logo.gif</url>
<link>http://www.mozilla.org</link>
</image>
<item>
<title>New Status Updates</title>
<link>http://www.mozilla.org/status/</link>
</item>
</rdf:RDF>
```



rdf:RDF	
xmlns:rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
xmlns	http://my.netscape.com/rdf/simple/0.9/
channel	
title	Mozilla Dot Org
link	http://www.mozilla.org
description	the Mozilla Organization web site
image	
title	Mozilla
url	http://www.mozilla.org/images/logo.gif
link	http://www.mozilla.org
item	
title	New Status Updates
link	http://www.mozilla.org/status/

Σχήμα 31: RSS 0.90

### Παράδειγμα αρχείου Netscape RSS 0.91

```
<!DOCTYPE rss SYSTEM "http://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
<channel>
<title>Example Channel</title>
<link>http://example.com/</link>
<description>an example feed</description>
<language>en</language>
<rating>(PICS-1.1 "http://www.classify.org/safesurf/" 1 r (SS~~000 1))</rating>
<textinput>
<title>Search this site:</title>
<description>Find:</description>
<name>q</name>
<link>http://example.com/search</link>
</textinput>
<skipHours>
<hour>0</hour>
</skipHours>
<item>
<title>1 &lt; 2</title>
<link>http://example.com/1_less_than_2.html</link>
<description>1 &lt; 2, 3 &lt; 4.
In HTML, &lt;b> starts a bold phrase
and you start a link with &lt;a href=
</description>
</item>
</channel>
</rss>
```

XML	version	1.0
	encoding	UTF-8
DOCTYPE rss	SYSTEM	"http://my.netscape.com/publish/formats/rss-0.91.dtd"
rss	version	0.91
channel	title	Example Channel
	link	http://example.com/
	description	an example feed
	language	en
	rating	(PICS-1.1 "http://www.classify.org/safesurf/" 1 r (SS~~000 1))
textinput	title	Search this site:
	description	Find:
	name	q
	link	http://example.com/search
skipHours	hour	0
item	title	1 &lt; 2
	link	http://example.com/1_less_than_2.html
	description	1 &lt; 2, 3 &lt; 4. In HTML, &lt;b> starts a bold phrase and you start a link with &lt;a href=

Σχήμα 32: RSS 0.91 Netscape

### Παράδειγμα αρχείου Userland RSS 0.91

```

<rss version="0.91">
<channel>
<title>Example Channel</title>
<link>http://example.com/</link>
<description>an example feed</description>
<language>en</language>
<rating>(PICS-1.1 "http://www.classify.org/safesurf/" 1 r (SS~000 1))</rating>
<textInput>
<title>Search this site:</title>
<description>Find:</description>
<name>q</name>
<link>http://example.com/search</link>
</textInput>
<skipHours>
<hour>24</hour>
</skipHours>
<item>
<title>1 &lt; 2</title>
<link>http://example.com/1_less_than_2.html</link>
<description>1 &lt; 2, 3 &lt; 4.
In HTML, &lt;b> starts a bold phrase
and you start a link with &lt;a href=
</description>
</item>
</channel>
</rss>

```

version	1.0
encoding	UTF-8

version	0.91																																
channel	<table border="1"> <tr> <td>title</td> <td>Example Channel</td> </tr> <tr> <td>link</td> <td>http://example.com/</td> </tr> <tr> <td>description</td> <td>an example feed</td> </tr> <tr> <td>language</td> <td>en</td> </tr> <tr> <td>rating</td> <td>(PICS-1.1 "http://www.classify.org/safesurf/" 1 r (SS~000 1))</td> </tr> <tr> <td>textInput</td> <td> <table border="1"> <tr> <td>title</td> <td>Search this site:</td> </tr> <tr> <td>description</td> <td>Find:</td> </tr> <tr> <td>name</td> <td>q</td> </tr> <tr> <td>link</td> <td>http://example.com/search</td> </tr> </table> </td> </tr> <tr> <td>skipHours</td> <td> <table border="1"> <tr> <td>hour</td> <td>24</td> </tr> </table> </td> </tr> <tr> <td>item</td> <td> <table border="1"> <tr> <td>title</td> <td>1 &amp;lt; 2</td> </tr> <tr> <td>link</td> <td>http://example.com/1_less_than_2.html</td> </tr> <tr> <td>description</td> <td>1 &amp;lt; 2, 3 &amp;lt; 4. In HTML, &amp;lt;b&gt; starts a bold phrase and you start a link with &amp;lt;a href=</td> </tr> </table> </td> </tr> </table>	title	Example Channel	link	http://example.com/	description	an example feed	language	en	rating	(PICS-1.1 "http://www.classify.org/safesurf/" 1 r (SS~000 1))	textInput	<table border="1"> <tr> <td>title</td> <td>Search this site:</td> </tr> <tr> <td>description</td> <td>Find:</td> </tr> <tr> <td>name</td> <td>q</td> </tr> <tr> <td>link</td> <td>http://example.com/search</td> </tr> </table>	title	Search this site:	description	Find:	name	q	link	http://example.com/search	skipHours	<table border="1"> <tr> <td>hour</td> <td>24</td> </tr> </table>	hour	24	item	<table border="1"> <tr> <td>title</td> <td>1 &amp;lt; 2</td> </tr> <tr> <td>link</td> <td>http://example.com/1_less_than_2.html</td> </tr> <tr> <td>description</td> <td>1 &amp;lt; 2, 3 &amp;lt; 4. In HTML, &amp;lt;b&gt; starts a bold phrase and you start a link with &amp;lt;a href=</td> </tr> </table>	title	1 &lt; 2	link	http://example.com/1_less_than_2.html	description	1 &lt; 2, 3 &lt; 4. In HTML, &lt;b> starts a bold phrase and you start a link with &lt;a href=
title	Example Channel																																
link	http://example.com/																																
description	an example feed																																
language	en																																
rating	(PICS-1.1 "http://www.classify.org/safesurf/" 1 r (SS~000 1))																																
textInput	<table border="1"> <tr> <td>title</td> <td>Search this site:</td> </tr> <tr> <td>description</td> <td>Find:</td> </tr> <tr> <td>name</td> <td>q</td> </tr> <tr> <td>link</td> <td>http://example.com/search</td> </tr> </table>	title	Search this site:	description	Find:	name	q	link	http://example.com/search																								
title	Search this site:																																
description	Find:																																
name	q																																
link	http://example.com/search																																
skipHours	<table border="1"> <tr> <td>hour</td> <td>24</td> </tr> </table>	hour	24																														
hour	24																																
item	<table border="1"> <tr> <td>title</td> <td>1 &amp;lt; 2</td> </tr> <tr> <td>link</td> <td>http://example.com/1_less_than_2.html</td> </tr> <tr> <td>description</td> <td>1 &amp;lt; 2, 3 &amp;lt; 4. In HTML, &amp;lt;b&gt; starts a bold phrase and you start a link with &amp;lt;a href=</td> </tr> </table>	title	1 &lt; 2	link	http://example.com/1_less_than_2.html	description	1 &lt; 2, 3 &lt; 4. In HTML, &lt;b> starts a bold phrase and you start a link with &lt;a href=																										
title	1 &lt; 2																																
link	http://example.com/1_less_than_2.html																																
description	1 &lt; 2, 3 &lt; 4. In HTML, &lt;b> starts a bold phrase and you start a link with &lt;a href=																																

Σχήμα 33: RSS 0.91 Userland

### Παράδειγμα αρχείου *RSS-DEV* RSS 1.0

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns="http://purl.org/rss/1.0/">
<channel>
<title>Example Dot Org</title>
<link>http://www.example.org</link>
<description>the Example Organization web site</description>
<items>
<rdf:Seq>
<rdf:li resource="http://www.example.org/status"/>
</rdf:Seq>
</items>
</channel>
<image rdf:about="http://www.example.org/images/logo.gif"/>
<image>
<title>Example</title>
<url>http://www.example.org/images/logo.gif</url>
<link>http://www.example.org</link>
</image>
<item rdf:about="http://www.example.org/status">
<title> New Status Updates </title>
<link>http://www.example.org/status </link>
<description> News about the Example project </description>
</item>
</rdf:RDF>
```

The screenshot displays an XML parser interface. The root node is 'XML', which contains 'version' (1.0) and 'encoding' (UTF-8). The next level is 'rdf:RDF', which includes 'xmlns:rdf' and 'xmlns'. The 'channel' element is expanded to show 'title' (Example Dot Org), 'link' (http://www.example.org), and 'description' (the Example Organization web site). Below the channel, there is an 'items' element containing an 'rdf:Seq' element, which in turn contains an 'rdf:li' element with a 'resource' attribute pointing to 'http://www.example.org/status'. The 'image (2)' element is also shown, with a table of properties: 'rdf:about' (http://www.example.org/images/logo.gif), 'title' (Example), 'url' (http://www.example.org/images/logo.gif), and 'link' (http://www.example.org). Finally, the 'item' element is expanded to show 'title' (New Status Updates), 'link' (http://www.example.org/status), and 'description' (News about the Example project).

Σχήμα 34: RSS 1.0

Δίνουμε παράλληλα και την επίσημη δήλωση για το **RDF Schema του RSS 1.0** (RDF Site Summary) από το RSS-DEV Working Group.

```
<?xml version="1.0"?>
<!-- RDF Schema declaration for Rich Site Summary (RSS) 1.0 <http://purl.org/rss/1.0/>
```

note: This schema currently is defining RSS-specific constructs (resource and relationship types) only. No constraints have been introduced.

Note: this schema is represented in the RDF M&S abbreviated syntax <http://www.w3.org/TR/REC-rdf-syntax/#abbreviatedSyntax> for syntactic inclusion in an HTML/XHTML document

```
-->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:dc="http://purl.org/dc/elements/1.1/">
  <!-- Class declarations -->
  <rdfs:Class rdf:about="http://purl.org/rss/1.0/channel" rdfs:label="Channel" rdfs:comment="An RSS information channel.">
    <rdfs:isDefinedBy rdf:resource="http://purl.org/rss/1.0/">
  </rdfs:Class>
  <rdfs:Class rdf:about="http://purl.org/rss/1.0/image" rdfs:label="Image" rdfs:comment="An RSS image.">
    <rdfs:isDefinedBy rdf:resource="http://purl.org/rss/1.0/">
  </rdfs:Class>
  <rdfs:Class rdf:about="http://purl.org/rss/1.0/item" rdfs:label="Item" rdfs:comment="An RSS item.">
    <rdfs:isDefinedBy rdf:resource="http://purl.org/rss/1.0/">
  </rdfs:Class>
  <rdfs:Class rdf:about="http://purl.org/rss/1.0/textinput" rdfs:label="Text Input" rdfs:comment="An RSS text input.">
    <rdfs:isDefinedBy rdf:resource="http://purl.org/rss/1.0/">
  </rdfs:Class>

  <!-- Property declarations -->
  <rdf:Property rdf:about="http://purl.org/rss/1.0/items" rdfs:label="Items" rdfs:comment="Points to a list of rss:item elements - members of the subject channel.">
    <rdfs:isDefinedBy rdf:resource="http://purl.org/rss/1.0/">
  </rdf:Property>
  <rdf:Property rdf:about="http://purl.org/rss/1.0/title" rdfs:label="Title" rdfs:comment="A descriptive title for the channel.">
    <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/elements/1.1/title"/>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/rss/1.0/">
  </rdf:Property>
```

```
<rdf:Property rdf:about="http://purl.org/rss/1.0/link" rdfs:label="Link" rdfs:comment="The URL to which an HTML rendering of the subject will link.">
  <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/elements/1.1/identifier"/>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/rss/1.0/">
</rdf:Property>
<rdf:Property rdf:about="http://purl.org/rss/1.0/url" rdfs:label="URL" rdfs:comment="The URL of image to used in 'src' attribute of channel's image tag when
rendered as HTML.">
  <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/elements/1.1/identifier"/>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/rss/1.0/">
</rdf:Property>
<rdf:Property rdf:about="http://purl.org/rss/1.0/description" rdfs:label="Description" rdfs:comment="A short text description of the subject.">
  <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/elements/1.1/description"/>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/rss/1.0/">
</rdf:Property>
<rdf:Property rdf:about="http://purl.org/rss/1.0/name" rdfs:label="Name" rdfs:comment="The text input field's (variable) name.">
  <rdfs:isDefinedBy rdf:resource="http://purl.org/rss/1.0/">
</rdf:Property>
</rdf:RDF>
```

### Παράδειγμα αρχείου Userland RSS 0.92

```
<rss version="0.92">
<channel>
<title>Example Channel</title>
<link>http://example.com/</link>
<description>an example feed</description>
<language>en</language>
<rating>(PICS-1.1 "http://www.classify.org/safesurf/" 1 r (SS~000 1))</rating>
<textInput>
<title>Search this site:</title>
<description>Find:</description>
<name>q</name>
<link>http://example.com/search</link>
</textInput>
<skipHours>
<hour>24</hour>
</skipHours>
<item>
<title>1 &lt; 2</title>
<link>http://example.com/1_less_than_2.html</link>
<description>1 &lt; 2, 3 &lt; 4.
In HTML, &lt;b> starts a bold phrase
and you start a link with &lt;a href=
</description>
</item>
</channel>
</rss>
```

version	1.0
encoding	UTF-8

version	0.92
title	Example Channel
link	http://example.com/
description	an example feed
language	en
rating	(PICS-1.1 "http://www.classify.org/safesurf/" 1 r (SS~000 1))
textInput	
title	Search this site:
description	Find:
name	q
link	http://example.com/search
skipHours	
hour	24
item	
title	1 &lt; 2
link	http://example.com/1_less_than_2.html
description	1 &lt; 2, 3 &lt; 4. In HTML, &lt;b> starts a bold phrase and you start a link with &lt;a href=

Σχήμα 35: RSS 0.92



### Παράδειγμα αρχείου Userland RSS 0.93

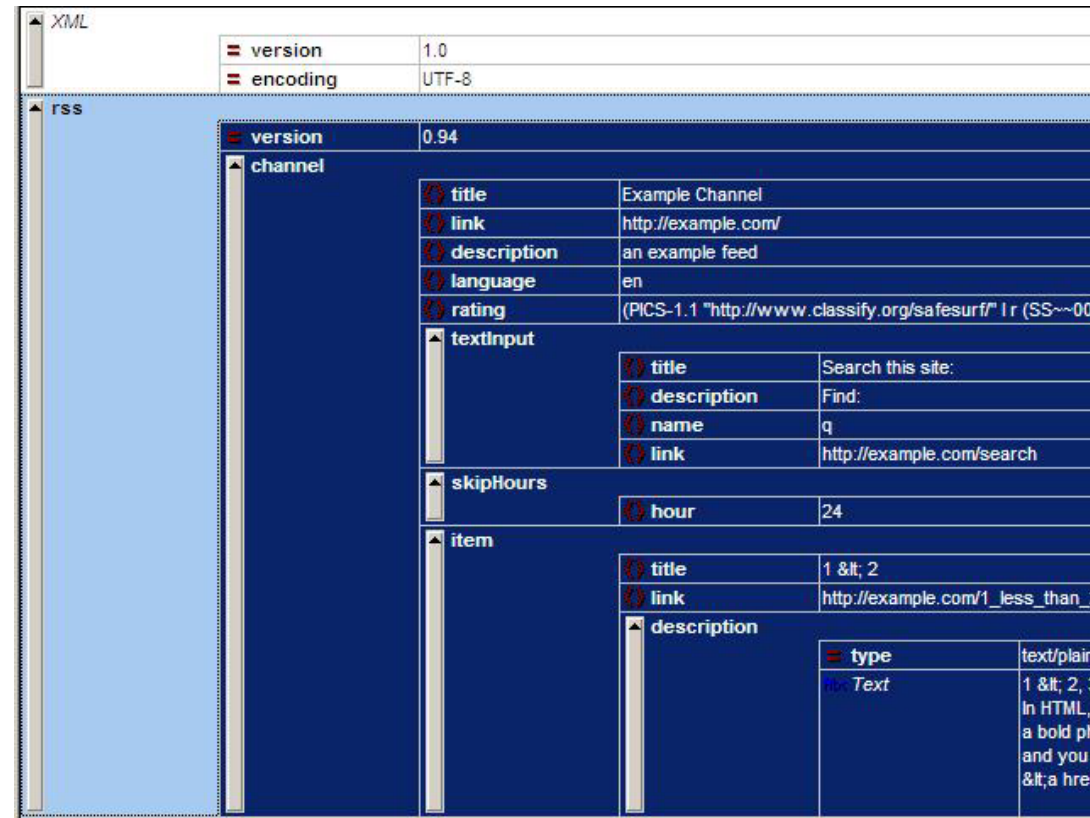
```
<rss version="0.93">
<channel>
<title>Example Channel</title>
<link>http://example.com/</link>
<description>an example feed</description>
<language>en</language>
<rating>(PICS-1.1 "http://www.classify.org/safesurf/" 1 r (SS~000 1))</rating>
<textInput>
<title>Search this site:</title>
<description>Find:</description>
<name>q</name>
<link>http://example.com/search</link>
</textInput>
<skipHours>
<hour>24</hour>
</skipHours>
<item>
<title>1 &lt; 2</title>
<link>http://example.com/1_less_than_2.html</link>
<description>1 &lt; 2, 3 &lt; 4.
In HTML, &lt;b&gt; starts a bold phrase
and you start a link with &lt;a href=
</description>
<expirationDate>Sat, 29 Nov 2003 10:17:13 GMT</expirationDate>
</item>
</channel>
</rss>
```

XML	version	1.0
	encoding	UTF-8
rss	version	0.93
channel	title	Example Channel
	link	http://example.com/
	description	an example feed
	language	en
	rating	(PICS-1.1 "http://www.classify.org/safesurf/" 1 r (SS~000 1))
textInput	title	Search this site:
	description	Find:
	name	q
	link	http://example.com/search
skipHours	hour	24
item	title	1 &lt; 2
	link	http://example.com/1_less_than_2.html
	description	1 &lt; 2, 3 &lt; 4. In HTML, &lt;b&gt; starts a bold phrase and you start a link with &lt;a href=
	expirationDate	Sat, 29 Nov 2003 10:17:13 GMT

Σχήμα 36: RSS 0.93

### Παράδειγμα αρχείου Userland RSS 0.94

```
<rss version="0.94">
<channel>
<title>Example Channel</title>
<link>http://example.com/</link>
<description>an example feed</description>
<language>en</language>
<rating>(PICS-1.1 "http://www.classify.org/safesurf/" 1 r (SS~~000 1))</rating>
<textInput>
<title>Search this site:</title>
<description>Find:</description>
<name>q</name>
<link>http://example.com/search</link>
</textInput>
<skipHours>
<hour>24</hour>
</skipHours>
<item>
<title>1 &lt; 2</title>
<link>http://example.com/1_less_than_2.html</link>
<description type="text/plain">1 &lt; 2, 3 &lt; 4.
In HTML, &lt;b&gt; starts a bold phrase
and you start a link with &lt;a href=
</description>
</item>
</channel>
</rss>
```



Σχήμα 37: RSS 0.94

### Παράδειγμα αρχείου RSS 2.0

```
<rss version="2.0">
<channel>
<title>Example Channel</title>
<link>http://example.com/</link>
<description>an example feed</description>
<language>en</language>
<textInput>
<title>Search this site:</title>
<description>Find:</description>
<name>q</name>
<link>http://example.com/search</link>
</textInput>
<skipHours>
<hour>24</hour>
</skipHours>
<item>
<title>1 &lt; 2</title>
<link>http://example.com/1_less_than_2.html</link>
<description>1 &lt; 2, 3 &lt; 4.
In HTML, &lt;b&gt; starts a bold phrase
and you start a link with &lt;a href=
</description>
</item>
</channel>
</rss>
```

XML	version	1.0
	encoding	UTF-8
rss	version	2.0
channel	title	Example Channel
	link	http://example.com/
	description	an example feed
	language	en
	textInput	
	title	Search this site:
	description	Find:
	name	q
	link	http://example.com/search
	skipHours	
	hour	24
	item	
	title	1 &lt; 2
	link	http://example.com/1_less_than_2.html
	description	1 &lt; 2, 3 &lt; 4. In HTML, &lt;b&gt; starts a bold phrase and you start a link with &lt;a href=

Σχήμα 38: RSS 2.0

### Παράδειγμα αρχείου Userland RSS 2.0.1

```
<rss version="2.0">
<channel>
<title>Example Channel</title>
<link>http://example.com/</link>
<description>an example feed</description>
<language>en</language>
<textInput>
<title>Search this site:</title>
<description>Find:</description>
<name>q</name>
<link>http://example.com/search</link>
</textInput>
<skipHours>
<hour>0</hour>
</skipHours>
<item>
<title>1 &lt; 2</title>
<link>http://example.com/1_less_than_2.html</link>
<description>1 &lt; 2, 3 &lt; 4.
In HTML, &lt;b&gt; starts a bold phrase
and you start a link with &lt;a href=
</description>
</item>
</channel>
</rss>
```

XML	version	1.0
	encoding	UTF-8
RSS	version	2.0
	channel	
	title	Example Channel
	link	http://example.com/
	description	an example feed
	language	en
	textInput	
	title	Search this site:
	description	Find:
	name	q
	link	http://example.com/search
	skipHours	
	hour	0
	item	
	title	1 &lt; 2
	link	http://example.com/1_less_than_2.html
	description	1 &lt; 2, 3 &lt; 4. In HTML, &lt;b&gt; starts a bold phrase and you start a link with &lt;a href=

Σχήμα 39: RSS 2.0.1

### Παράδειγμα αρχείου Userland RSS 2.0.2

```
<rss version="2.0">
<channel>
<title>Example Channel</title>
<link>http://example.com/</link>
<description>an example feed</description>
<language>en</language>
<rating>(PICS-1.1 "http://www.classify.org/safesurf" 1 r (SS~000 1))</rating>
<textInput>
<title>Search this site:</title>
<description>Find:</description>
<name>q</name>
<link>http://example.com/search</link>
</textInput>
<skipHours>
<hour>0</hour>
</skipHours>
<item>
<title>1 &lt; 2</title>
<link>http://example.com/1_less_than_2.html</link>
<description>1 &lt; 2, 3 &lt; 4.
In HTML, &lt;b&gt; starts a bold phrase
and you start a link with &lt;a href=
</description>
</item>
</channel>
</rss>
```

version	1.0
encoding	UTF-8
rss	
channel	
title	Example Channel
link	http://example.com/
description	an example feed
language	en
rating	(PICS-1.1 "http://www.classify.org/safesurf" 1 r (SS~000 1))
textInput	
title	Search this site:
description	Find:
name	q
link	http://example.com/search
skipHours	
hour	0
item	
title	1 &lt; 2
link	http://example.com/1_less_than_2.html
description	1 &lt; 2, 3 &lt; 4. In HTML, &lt;b&gt; starts a bold phrase and you start a link with &lt;a href=

Σχήμα 40: RSS 2.0.2

Συγκεντρωτικά, οι διαφορές μεταξύ των elements και attributes των δημοφιλέστερων εκδόσεων του RSS φαίνονται στον ακόλουθο πίνακα. Σημειώνεται ότι στον ακόλουθο πίνακα τα attributes συνοδεύονται από το σύμβολο @ για να ξεχωρίζουν.

Πίνακας 8: Σύγκριση των δημοφιλέστερων εκδόσεων του RSS

RSS <a href="#">0.9</a>	RSS <a href="#">0.91</a>	RSS <a href="#">1.0</a>	RSS <a href="#">0.92</a>	RSS <a href="#">2.0</a>
" <a href="#">RDF Site Summary</a> "	"Rich SiteSummary"	" <a href="#">RDF Site Summary</a> "		"Really Simple Syndication"
<ul style="list-style-type: none"> <li>• Channel               <ul style="list-style-type: none"> <li>○ description</li> <li>○ link</li> <li>○ title</li> </ul> </li> <li>• image               <ul style="list-style-type: none"> <li>○ title</li> <li>○ url</li> <li>○ link</li> </ul> </li> <li>• item               <ul style="list-style-type: none"> <li>○ link</li> <li>○ title</li> </ul> </li> <li>• textinput               <ul style="list-style-type: none"> <li>○ description</li> <li>○ link</li> <li>○ name</li> <li>○ title</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• @version</li> <li>• channel               <ul style="list-style-type: none"> <li>○ copyright</li> <li>○ description</li> <li>○ docs</li> <li>○ image                   <ul style="list-style-type: none"> <li>▪ height</li> <li>▪ link</li> <li>▪ title</li> <li>▪ url</li> <li>▪ width</li> </ul> </li> </ul> </li> <li>○ item               <ul style="list-style-type: none"> <li>▪ description</li> <li>▪ link</li> <li>▪ title</li> </ul> </li> <li>○ language</li> <li>○ lastBuildDate</li> <li>○ link</li> <li>○ managingEditor</li> <li>○ pubDate</li> <li>○ rating</li> </ul>	<ul style="list-style-type: none"> <li>• channel               <ul style="list-style-type: none"> <li>○ @about</li> <li>○ description</li> <li>○ link</li> <li>○ title</li> <li>○ items</li> </ul> </li> <li>• image               <ul style="list-style-type: none"> <li>○ @about</li> <li>○ link</li> <li>○ title</li> <li>○ url</li> </ul> </li> <li>• item               <ul style="list-style-type: none"> <li>○ @about</li> <li>○ description</li> <li>○ link</li> <li>○ title</li> </ul> </li> <li>• textinput               <ul style="list-style-type: none"> <li>○ @about</li> <li>○ description</li> <li>○ link</li> <li>○ name</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• @version</li> <li>• channel               <ul style="list-style-type: none"> <li>○ cloud</li> <li>○ copyright</li> <li>○ description</li> <li>○ docs</li> <li>○ image                   <ul style="list-style-type: none"> <li>▪ height</li> <li>▪ link</li> <li>▪ title</li> <li>▪ url</li> <li>▪ width</li> </ul> </li> </ul> </li> <li>○ item               <ul style="list-style-type: none"> <li>▪ category</li> <li>▪ @domain</li> <li>▪ description</li> <li>▪ enclosure</li> <li>▪ @length</li> <li>▪ @type</li> <li>▪ @url</li> <li>▪ link</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• @version</li> <li>• channel               <ul style="list-style-type: none"> <li>○ category</li> <li>○ cloud</li> <li>○ copyright</li> <li>○ description</li> <li>○ docs</li> <li>○ generator</li> <li>○ image                   <ul style="list-style-type: none"> <li>▪ height</li> <li>▪ link</li> <li>▪ title</li> <li>▪ url</li> <li>▪ width</li> </ul> </li> <li>○ item               <ul style="list-style-type: none"> <li>▪ author</li> <li>▪ category</li> <li>▪ comments                   <ul style="list-style-type: none"> <li>▪ @domain</li> </ul> </li> <li>▪ description                   <ul style="list-style-type: none"> <li>▪ @type</li> </ul> </li> </ul> </li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ textinput <ul style="list-style-type: none"> <li>▪ description</li> <li>▪ link</li> <li>▪ name</li> <li>▪ title</li> </ul> </li> <li>○ skipHours</li> <li>○ skipDays</li> <li>○ title</li> <li>○ webMaster</li> </ul>	<ul style="list-style-type: none"> <li>○ title</li> </ul>	<ul style="list-style-type: none"> <li>▪ source <ul style="list-style-type: none"> <li>▪ @url</li> <li>▪ title</li> </ul> </li> <li>○ language</li> <li>○ lastBuildDate</li> <li>○ link</li> <li>○ managingEditor</li> <li>○ pubDate</li> <li>○ rating</li> <li>○ textInput <ul style="list-style-type: none"> <li>▪ description</li> <li>▪ link</li> <li>▪ name</li> <li>▪ title</li> </ul> </li> <li>○ title</li> <li>○ skipHours</li> <li>○ skipDays</li> <li>○ webMaster</li> </ul>	<ul style="list-style-type: none"> <li>▪ enclosure <ul style="list-style-type: none"> <li>▪ @length</li> <li>▪ @type</li> <li>▪ @url</li> </ul> </li> <li>▪ guid <ul style="list-style-type: none"> <li>▪ @isPermaLink</li> </ul> </li> <li>▪ link</li> <li>▪ pubDate</li> <li>▪ source <ul style="list-style-type: none"> <li>▪ @url</li> </ul> </li> <li>▪ title</li> <li>○ language</li> <li>○ lastBuildDate</li> <li>○ link</li> <li>○ managingEditor</li> <li>○ pubDate</li> <li>○ rating</li> <li>○ skipHours</li> <li>○ skipDays</li> <li>○ textInput <ul style="list-style-type: none"> <li>▪ description</li> <li>▪ link</li> <li>▪ name</li> <li>▪ title</li> </ul> </li> <li>○ title</li> <li>○ ttl</li> <li>○ webMaster</li> </ul>
--	--	---	---	--

Τέλος παρέχεται μια αντιστοίχιση από το RSS 0.9x στο RSS 1.0 (το οποίο υποστηρίζει και τα Dublin Core - DC elements)

### **Πίνακας 9: Αντιστοίχιση RSS 0.9x – RSS 1.0**

<b>Έκδοση 0.9x/2.0</b>	<b>Έκδοση 1.0</b>
category	dc:subject
copyright	dc:rights
docs	xmlns
generator	admin:generatorAgent
cloud	cp:server
guid	rdf:about
language	dc:language
lastBuildDate	dcterms:modified
managingEditor	dc:creator
pubDate	dcterms:available
ttl	--
webMaster	dc:publisher
comments	annotate:reference



## Σύγκριση των δημοφιλέστερων format RSS 1.0 , RSS 2.0 και Atom με εικόνες

Παρακάτω παραθέτουμε για σύγκριση την μορφή ενός τυχαίου feed περιεχομένου, όταν αυτό κωδικοποιείται με βάση τα δημοφιλέστερα αυτή τη στιγμή schemas, τα οποία είναι τα RSS 1, RSS 2 και Atom.

### **RSS 1.0**

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:content="http://purl.org/rss/1.0/modules/content/"
xmlns:admin="http://webns.net/mvcb/" xmlns="http://purl.org/rss/1.0/">
  <channel rdf:about="http://solution-channel-guid/795740118939166043">
    <title>Channel 1</title>
    <link>http://radiouserland.blogspot.com</link>
    <description>This is an English channel - We can only insert entries in a
channel, not directly in a solution</description>
    <dc:date>2005-01-28T13:00:38+02:00</dc:date>
    <dc:creator>Stefanos (evangelous@gmail.com)</dc:creator>
    <dc:language>en-us</dc:language>
    <dc:rights>My company</dc:rights>
    <admin:generatorAgent
rdf:resource="http://www.howdev.com/products/syndicationstudio2004/" />
  </channel>
  <rdf:Seq>
    <rdf:li rdf:resource="http://solution-entry-guid/779438310461024229" />
    <rdf:li rdf:resource="http://solution-entry-guid/344325489121092372" />
  </rdf:Seq>
</rdf:RDF>
```

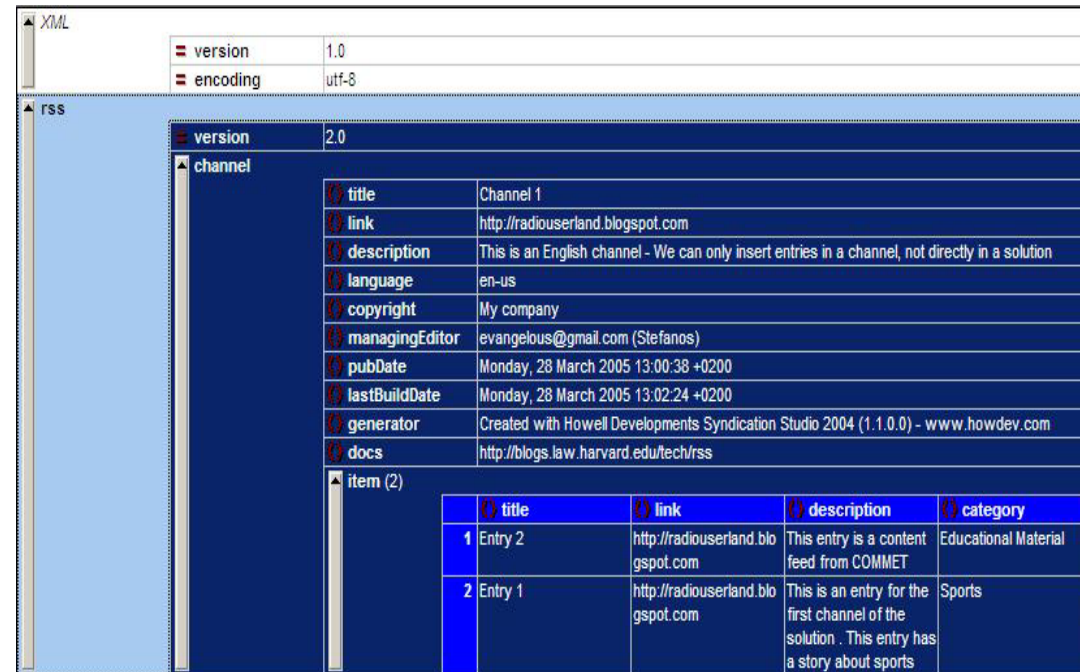
```
</rdf:Seq>
</items>
</channel>
<item rdf:about="http://solution-entry-guid/779438310461024229">
  <title>Entry 2</title>
  <link>http://radiouserland.blogspot.com</link>
  <content:encoded>This entry is a content feed from
COMMET</content:encoded>
  <dc:date>2005-01-28T12:54:23+02:00</dc:date>
  <dc:creator>COMMET</dc:creator>
  <dc:subject>Educational Material</dc:subject>
</item>
<item rdf:about="http://solution-entry-guid/344325489121092372">
  <title>Entry 1</title>
  <link>http://radiouserland.blogspot.com</link>
  <content:encoded>This is an entry for the first channel of the solution. This
entry has a story about sports</content:encoded>
  <dc:date>2005-01-28T12:53:02+02:00</dc:date>
  <dc:creator>Stefanos</dc:creator>
  <dc:subject>Sports</dc:subject>
</item>
</rdf:RDF>
```

rdf:RDF																																																								
xmlns:rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#																																																							
xmlns:dc	http://purl.org/dc/elements/1.1/																																																							
xmlns:content	http://purl.org/rss/1.0/modules/content/																																																							
xmlns:admin	http://webns.net/mvcb/																																																							
xmlns	http://purl.org/rss/1.0/																																																							
channel	<table border="1"> <tr> <td>rdf:about</td> <td>http://solution-channel-guid/795740118939166043</td> </tr> <tr> <td>title</td> <td>Channel 1</td> </tr> <tr> <td>link</td> <td>http://radiuserland.blogspot.com</td> </tr> <tr> <td>description</td> <td>This is an English channel - We can only insert entries in a channel, not directly in a solution</td> </tr> <tr> <td>dc:date</td> <td>2005-01-28T13:00:38+02:00</td> </tr> <tr> <td>dc:creator</td> <td>Stefanos (evangelous@gmail.com)</td> </tr> <tr> <td>dc:language</td> <td>en-us</td> </tr> <tr> <td>dc:rights</td> <td>My company</td> </tr> <tr> <td>admin:generatorAgent</td> <td> <table border="1"> <tr> <td>rdf:resource</td> <td>http://www.howdev.com/products/syndicationstudio2004/</td> </tr> </table> </td> </tr> <tr> <td>items</td> <td> <table border="1"> <tr> <td>rdf:Seq</td> <td> <table border="1"> <tr> <td>rdf:li (2)</td> <td> <table border="1"> <tr> <td>rdf:resource</td> <td> <table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table> </td> </tr> </table> </td> </tr> </table> </td> </tr> </table> </td> </tr> <tr> <td>item (2)</td> <td> <table border="1"> <thead> <tr> <th></th> <th>rdf:about</th> <th>title</th> <th>link</th> <th>content:encoded</th> <th>dc:date</th> <th>dc:creator</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> <td>Entry 2</td> <td>http://radiuserland.blogspot.com</td> <td>This entry is a content feed from COMMENT</td> <td>2005-01-28T12:54:23+02:00</td> <td>COMMENT</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> <td>Entry 1</td> <td>http://radiuserland.blogspot.com</td> <td>This is an entry for the first channel of the solution. This entry has a story about sports</td> <td>2005-01-28T12:53:02+02:00</td> <td>Stefanos</td> </tr> </tbody> </table> </td> </tr> </table>	rdf:about	http://solution-channel-guid/795740118939166043	title	Channel 1	link	http://radiuserland.blogspot.com	description	This is an English channel - We can only insert entries in a channel, not directly in a solution	dc:date	2005-01-28T13:00:38+02:00	dc:creator	Stefanos (evangelous@gmail.com)	dc:language	en-us	dc:rights	My company	admin:generatorAgent	<table border="1"> <tr> <td>rdf:resource</td> <td>http://www.howdev.com/products/syndicationstudio2004/</td> </tr> </table>	rdf:resource	http://www.howdev.com/products/syndicationstudio2004/	items	<table border="1"> <tr> <td>rdf:Seq</td> <td> <table border="1"> <tr> <td>rdf:li (2)</td> <td> <table border="1"> <tr> <td>rdf:resource</td> <td> <table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table> </td> </tr> </table> </td> </tr> </table> </td> </tr> </table>	rdf:Seq	<table border="1"> <tr> <td>rdf:li (2)</td> <td> <table border="1"> <tr> <td>rdf:resource</td> <td> <table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table> </td> </tr> </table> </td> </tr> </table>	rdf:li (2)	<table border="1"> <tr> <td>rdf:resource</td> <td> <table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table> </td> </tr> </table>	rdf:resource	<table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table>	1	http://solution-entry-guid/779438310461024229	2	http://solution-entry-guid/344325489121092372	item (2)	<table border="1"> <thead> <tr> <th></th> <th>rdf:about</th> <th>title</th> <th>link</th> <th>content:encoded</th> <th>dc:date</th> <th>dc:creator</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> <td>Entry 2</td> <td>http://radiuserland.blogspot.com</td> <td>This entry is a content feed from COMMENT</td> <td>2005-01-28T12:54:23+02:00</td> <td>COMMENT</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> <td>Entry 1</td> <td>http://radiuserland.blogspot.com</td> <td>This is an entry for the first channel of the solution. This entry has a story about sports</td> <td>2005-01-28T12:53:02+02:00</td> <td>Stefanos</td> </tr> </tbody> </table>		rdf:about	title	link	content:encoded	dc:date	dc:creator	1	http://solution-entry-guid/779438310461024229	Entry 2	http://radiuserland.blogspot.com	This entry is a content feed from COMMENT	2005-01-28T12:54:23+02:00	COMMENT	2	http://solution-entry-guid/344325489121092372	Entry 1	http://radiuserland.blogspot.com	This is an entry for the first channel of the solution. This entry has a story about sports	2005-01-28T12:53:02+02:00	Stefanos
rdf:about	http://solution-channel-guid/795740118939166043																																																							
title	Channel 1																																																							
link	http://radiuserland.blogspot.com																																																							
description	This is an English channel - We can only insert entries in a channel, not directly in a solution																																																							
dc:date	2005-01-28T13:00:38+02:00																																																							
dc:creator	Stefanos (evangelous@gmail.com)																																																							
dc:language	en-us																																																							
dc:rights	My company																																																							
admin:generatorAgent	<table border="1"> <tr> <td>rdf:resource</td> <td>http://www.howdev.com/products/syndicationstudio2004/</td> </tr> </table>	rdf:resource	http://www.howdev.com/products/syndicationstudio2004/																																																					
rdf:resource	http://www.howdev.com/products/syndicationstudio2004/																																																							
items	<table border="1"> <tr> <td>rdf:Seq</td> <td> <table border="1"> <tr> <td>rdf:li (2)</td> <td> <table border="1"> <tr> <td>rdf:resource</td> <td> <table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table> </td> </tr> </table> </td> </tr> </table> </td> </tr> </table>	rdf:Seq	<table border="1"> <tr> <td>rdf:li (2)</td> <td> <table border="1"> <tr> <td>rdf:resource</td> <td> <table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table> </td> </tr> </table> </td> </tr> </table>	rdf:li (2)	<table border="1"> <tr> <td>rdf:resource</td> <td> <table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table> </td> </tr> </table>	rdf:resource	<table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table>	1	http://solution-entry-guid/779438310461024229	2	http://solution-entry-guid/344325489121092372																																													
rdf:Seq	<table border="1"> <tr> <td>rdf:li (2)</td> <td> <table border="1"> <tr> <td>rdf:resource</td> <td> <table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table> </td> </tr> </table> </td> </tr> </table>	rdf:li (2)	<table border="1"> <tr> <td>rdf:resource</td> <td> <table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table> </td> </tr> </table>	rdf:resource	<table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table>	1	http://solution-entry-guid/779438310461024229	2	http://solution-entry-guid/344325489121092372																																															
rdf:li (2)	<table border="1"> <tr> <td>rdf:resource</td> <td> <table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table> </td> </tr> </table>	rdf:resource	<table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table>	1	http://solution-entry-guid/779438310461024229	2	http://solution-entry-guid/344325489121092372																																																	
rdf:resource	<table border="1"> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> </tr> </table>	1	http://solution-entry-guid/779438310461024229	2	http://solution-entry-guid/344325489121092372																																																			
1	http://solution-entry-guid/779438310461024229																																																							
2	http://solution-entry-guid/344325489121092372																																																							
item (2)	<table border="1"> <thead> <tr> <th></th> <th>rdf:about</th> <th>title</th> <th>link</th> <th>content:encoded</th> <th>dc:date</th> <th>dc:creator</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>http://solution-entry-guid/779438310461024229</td> <td>Entry 2</td> <td>http://radiuserland.blogspot.com</td> <td>This entry is a content feed from COMMENT</td> <td>2005-01-28T12:54:23+02:00</td> <td>COMMENT</td> </tr> <tr> <td>2</td> <td>http://solution-entry-guid/344325489121092372</td> <td>Entry 1</td> <td>http://radiuserland.blogspot.com</td> <td>This is an entry for the first channel of the solution. This entry has a story about sports</td> <td>2005-01-28T12:53:02+02:00</td> <td>Stefanos</td> </tr> </tbody> </table>		rdf:about	title	link	content:encoded	dc:date	dc:creator	1	http://solution-entry-guid/779438310461024229	Entry 2	http://radiuserland.blogspot.com	This entry is a content feed from COMMENT	2005-01-28T12:54:23+02:00	COMMENT	2	http://solution-entry-guid/344325489121092372	Entry 1	http://radiuserland.blogspot.com	This is an entry for the first channel of the solution. This entry has a story about sports	2005-01-28T12:53:02+02:00	Stefanos																																		
	rdf:about	title	link	content:encoded	dc:date	dc:creator																																																		
1	http://solution-entry-guid/779438310461024229	Entry 2	http://radiuserland.blogspot.com	This entry is a content feed from COMMENT	2005-01-28T12:54:23+02:00	COMMENT																																																		
2	http://solution-entry-guid/344325489121092372	Entry 1	http://radiuserland.blogspot.com	This is an entry for the first channel of the solution. This entry has a story about sports	2005-01-28T12:53:02+02:00	Stefanos																																																		

Σχήμα 41: RSS 1.0

## RSS 2

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
  <channel>
    <title>Channel 1</title>
    <link>http://radiouserland.blogspot.com</link>
    <description>This is an English channel - We can only insert entries in a
channel, not directly in a solution</description>
    <language>en-us</language>
    <copyright>My company</copyright>
    <managingEditor>evangelous@gmail.com (Stefanos)</managingEditor>
    <pubDate>Monday, 28 January 2005 13:00:38 +0200</pubDate>
    <lastBuildDate>Monday, 28 January 2005 13:02:24 +0200</lastBuildDate>
    <generator>Created with Howell Developments Syndication Studio 2004
(1.1.0.0) - www.howdev.com</generator>
    <docs>http://blogs.law.harvard.edu/tech/rss</docs>
    <item>
      <title>Entry 2</title>
      <link>http://radiouserland.blogspot.com</link>
      <description>This entry is a content feed from COMMET</description>
      <category>Educational Material</category>
      <pubDate>Monday, 28 January 2005 12:54:23 +0200</pubDate>
    </item>
    <item>
      <title>Entry 1</title>
      <link>http://radiouserland.blogspot.com</link>
      <description>This is an entry for the first channel of the solution . This entry
has a story about sports</description>
      <category>Sports</category>
      <pubDate> Monday, 28 May 2005 12:53:02 +0200</pubDate>
    </item>
  </channel>
</rss>
```



XML	
version	1.0
encoding	utf-8

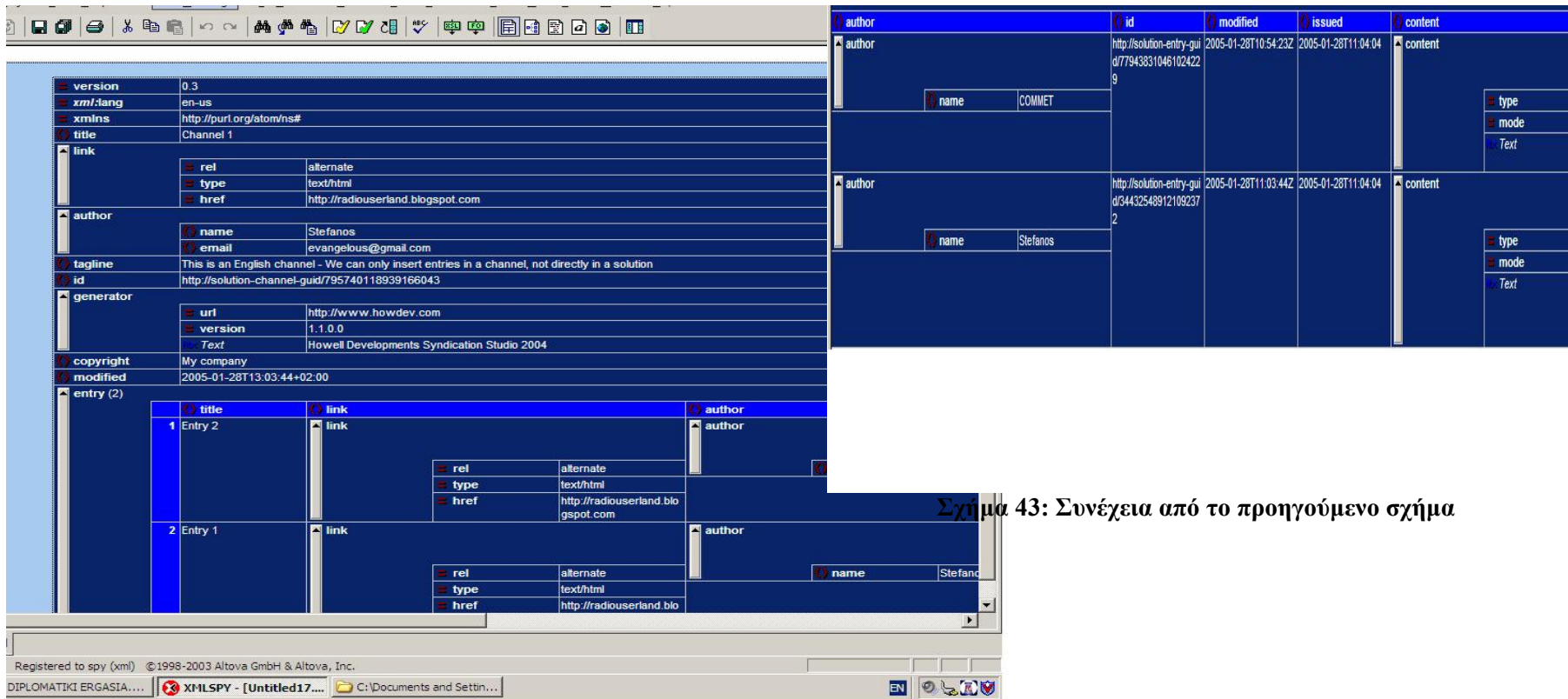
  

rss				
version	2.0			
channel				
title	Channel 1			
link	http://radiouserland.blogspot.com			
description	This is an English channel - We can only insert entries in a channel, not directly in a solution			
language	en-us			
copyright	My company			
managingEditor	evangelous@gmail.com (Stefanos)			
pubDate	Monday, 28 March 2005 13:00:38 +0200			
lastBuildDate	Monday, 28 March 2005 13:02:24 +0200			
generator	Created with Howell Developments Syndication Studio 2004 (1.1.0.0) - www.howdev.com			
docs	http://blogs.law.harvard.edu/tech/rss			
item (2)				
title	link	description	category	
1	Entry 2	http://radiouserland.blogspot.com	This entry is a content feed from COMMET	Educational Material
2	Entry 1	http://radiouserland.blogspot.com	This is an entry for the first channel of the solution . This entry has a story about sports	Sports

Σχήμα 42: RSS 2.0

## ***Atom***

```
<?xml version="1.0" encoding="utf-8"?>
<feed version="0.3" xml:lang="en-us" xmlns="http://purl.org/atom/ns#">
  <title>Channel 1</title>
  <link rel="alternate" type="text/html" href="http://radiouserland.blogspot.com" />
  <author>
    <name>Stefanos</name>
    <email>evangelous@gmail.com</email>
  </author>
  <tagline>This is an English channel - We can only insert entries in a channel, not directly in a solution</tagline>
  <id>http://solution-channel-guid/795740118939166043</id>
  <generator url="http://www.howdev.com" version="1.1.0.0">Howell Developments Syndication Studio 2004</generator>
  <copyright>My company</copyright>
  <modified>2005-01-28T13:03:44+02:00</modified>
  <entry>
    <title>Entry 2</title>
    <link rel="alternate" type="text/html" href="http://radiouserland.blogspot.com" />
    <author>
      <name>COMMET</name>
    </author>
    <id>http://solution-entry-guid/779438310461024229</id>
    <modified>2005-01-28T10:54:23Z</modified>
    <issued>2005-01-28T11:04:04</issued>
    <content type="text/html" mode="escaped">This entry is a content feed from COMMET</content>
  </entry>
  <entry>
    <title>Entry 1</title>
    <link rel="alternate" type="text/html" href="http://radiouserland.blogspot.com" />
    <author>
      <name>Stefanos</name>
    </author>
    <id>http://solution-entry-guid/344325489121092372</id>
    <modified>2005-01-28T11:03:44Z</modified>
    <issued>2005-01-28T11:04:04</issued>
    <content type="text/html" mode="xml">This is an entry for the first channel of the solution . This entry has a story about sports</content>
  </entry>
</feed>
```



Σχήμα 43: Συνέχεια από το προηγούμενο σχήμα

Σχήμα 44: Atom

## **Σύγκριση των OPML Directories για κάθε ένα από τα παραπάνω RSS1, RSS2, Atom**

Υποθέτουμε ότι ένα CMS διαθέτει το περιεχόμενό του και στα τρία προαναφερθέντα format. Το διαθέσιμο περιεχόμενο υποθέτουμε επίσης ότι είναι οργανωμένο σε πέντε κανάλια. Για κάθε ένα από τα διαθέσιμα format κατασκευάζεται και το αντίστοιχο .opml αρχείο για την ευρετηριοποίηση των καναλιών.

### **OPML έκδοση RSS 1.0**

```
<?xml version="1.0" encoding="utf-8"?>
<opml version="1.0">
  <head>
    <title>Solution 1</title>
    <ownerName>Stefanos</ownerName>
    <ownerEmail>evangelous@gmail.com</ownerEmail>
  </head>
  <body>
    <outline text="Channel 1" type="rss" version="RSS" title="Channel 1" description="This is an English channel - We can only insert entries in a channel, not directly in a solution" xmlUrl="http://radiuserland.blogspot.com/channel1.rdf" htmlUrl="" />

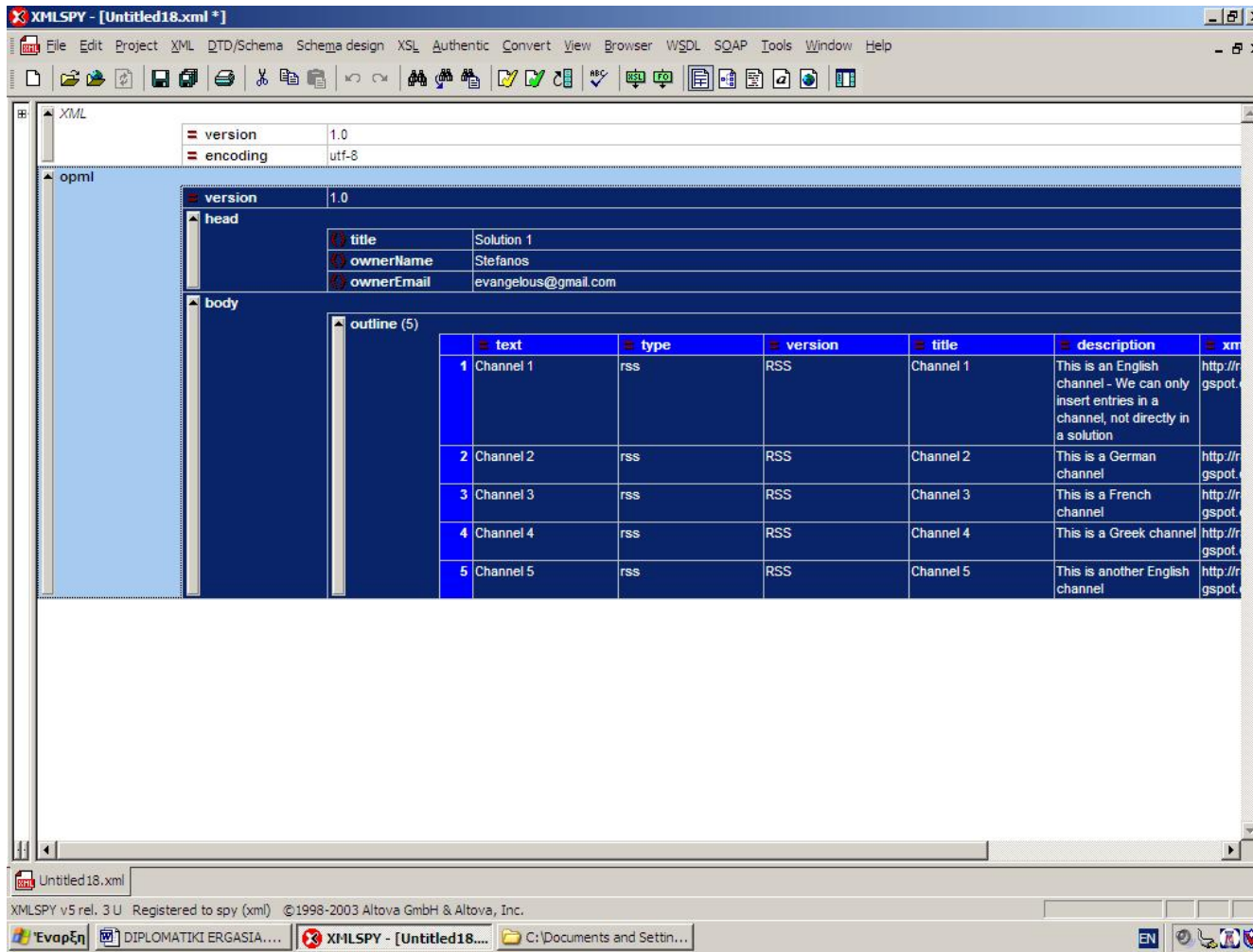
    <outline text="Channel 2" type="rss" version="RSS" title="Channel 2" description="This is a German channel"
xmlUrl="http://radiuserland.blogspot.com/channel2.rdf" htmlUrl="" />

    <outline text="Channel 3" type="rss" version="RSS" title="Channel 3" description="This is a French channel"
xmlUrl="http://radiuserland.blogspot.com/channel3.rdf" htmlUrl="" />

    <outline text="Channel 4" type="rss" version="RSS" title="Channel 4" description="This is a Greek channel"
xmlUrl="http://radiuserland.blogspot.com/channel4.rdf" htmlUrl="" />

    <outline text="Channel 5" type="rss" version="RSS" title="Channel 5" description="This is another English channel"
xmlUrl="http://radiuserland.blogspot.com/channel5.rdf" htmlUrl="" />

  </body>
</opml>
```



Σχήμα 45: OPML για RSS 1

## *OPML έκδοση RSS 2.0*

```
<?xml version="1.0" encoding="utf-8"?>
<opml version="1.0">
  <head>
    <title>Solution 1</title>
    <ownerName>Stefanos</ownerName>
    <ownerEmail>evangelous@gmail.com</ownerEmail>
  </head>
  <body>
    <outline text="Channel 1" type="rss" version="RSS2" title="Channel 1" description="This is an English channel - We can only insert entries in a channel, not
directly in a solution" xmlUrl="http://radiouserland.blogspot.com/channel1.rss" htmlUrl="" />

    <outline text="Channel 2" type="rss" version="RSS2" title="Channel 2" description="This is a German channel"
xmlUrl="http://radiouserland.blogspot.com/channel2.rss" htmlUrl="" />

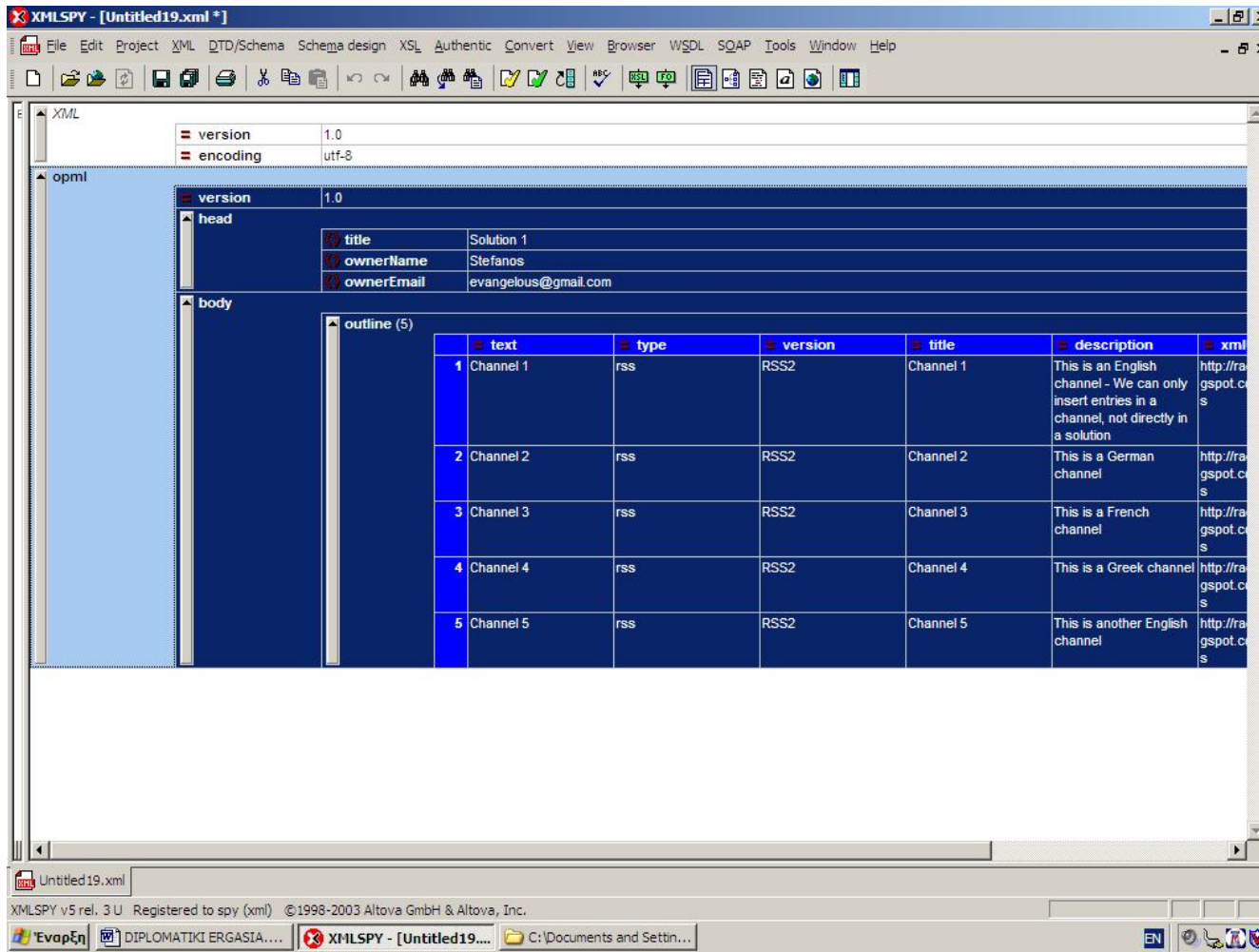
    <outline text="Channel 3" type="rss" version="RSS2" title="Channel 3" description="This is a French channel"
xmlUrl="http://radiouserland.blogspot.com/channel3.rss" htmlUrl="" />

    <outline text="Channel 4" type="rss" version="RSS2" title="Channel 4" description="This is a Greek channel"
xmlUrl="http://radiouserland.blogspot.com/channel4.rss" htmlUrl="" />

    <outline text="Channel 5" type="rss" version="RSS2" title="Channel 5" description="This is another English channel"
xmlUrl="http://radiouserland.blogspot.com/channel5.rss" htmlUrl="" />

  </body>
</opml>
```

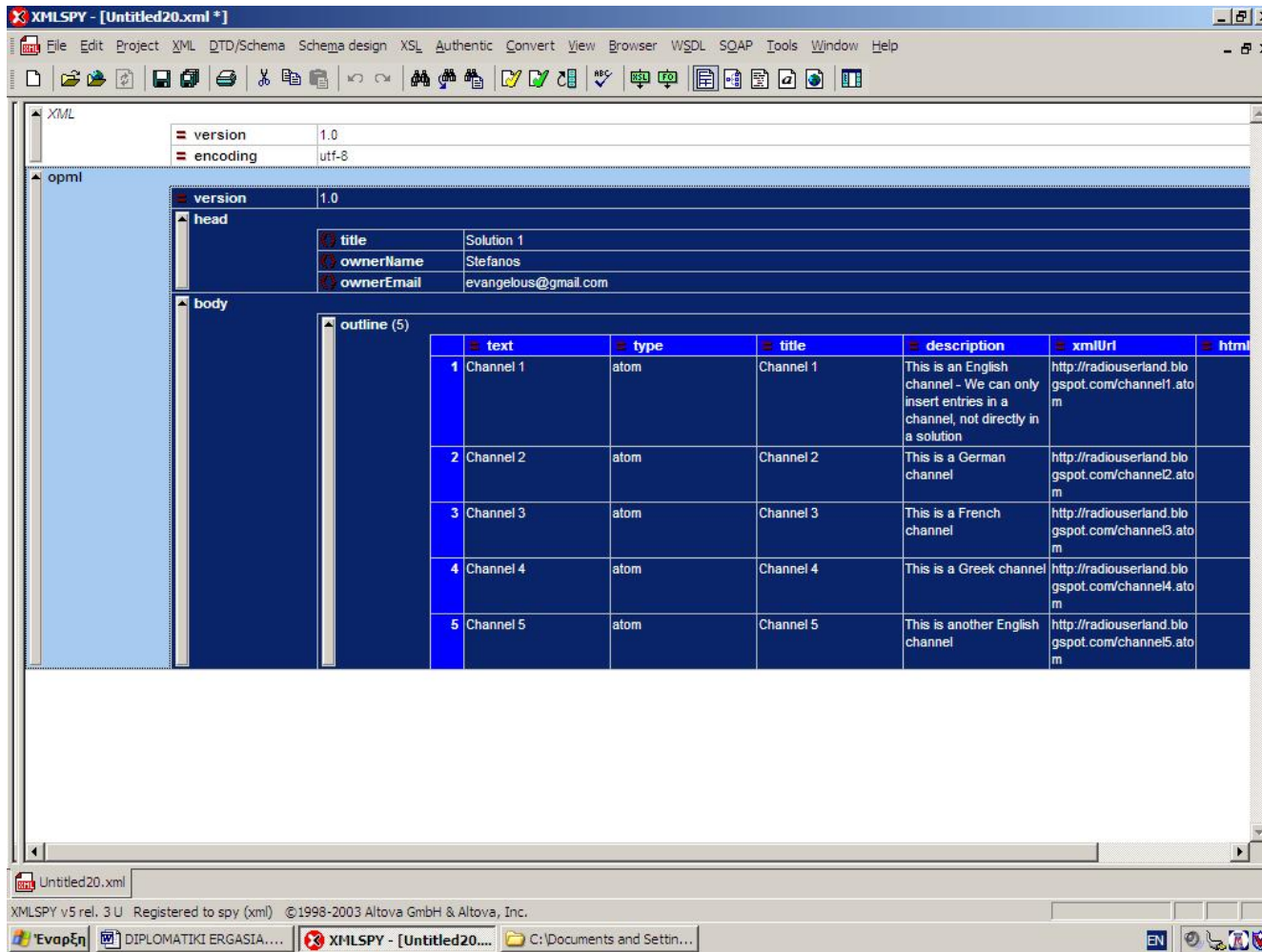




Σχήμα 46: OPML για RSS 2

## *OPML έκδοση Atom*

```
<?xml version="1.0" encoding="utf-8"?>
<opml version="1.0">
  <head>
    <title>Solution 1</title>
    <ownerName>Stefanos</ownerName>
    <ownerEmail>evangelous@gmail.com</ownerEmail>
  </head>
  <body>
    <outline text="Channel 1" type="atom" title="Channel 1" description="This is an English channel - We can only insert entries in a channel, not directly in a solution" xmlUrl="http://radiouserland.blogspot.com/channel1.atom" htmlUrl="" />
    <outline text="Channel 2" type="atom" title="Channel 2" description="This is a German channel" xmlUrl="http://radiouserland.blogspot.com/channel2.atom" htmlUrl="" />
    <outline text="Channel 3" type="atom" title="Channel 3" description="This is a French channel" xmlUrl="http://radiouserland.blogspot.com/channel3.atom" htmlUrl="" />
    <outline text="Channel 4" type="atom" title="Channel 4" description="This is a Greek channel" xmlUrl="http://radiouserland.blogspot.com/channel4.atom" htmlUrl="" />
    <outline text="Channel 5" type="atom" title="Channel 5" description="This is another English channel" xmlUrl="http://radiouserland.blogspot.com/channel5.atom" htmlUrl="" />
  </body>
</opml>
```



Σχήμα 47: OPML για Atom

## Παραδείγματα από Web site feeds

Η πλειοψηφία των Web site παρέχουν πλέον το περιεχόμενό τους με την μορφή feeds. Το πρωτόκολλο RSS 2.0 είναι το πλέον διαδεδομένο, ωστόσο μερικοί φορείς υποστηρίζουν είτε το RSS 1.0 και 0.9x και/ή το Atom, όπως για παράδειγμα η εταιρεία Google. Παρακάτω παραθέτουμε ενδεικτικά εικόνες από web site γνωστών παρόχων περιεχομένου.

### Syndicated feeds

You can subscribe to headline feeds from various sections of the Loosely Coupled site using additional feeds, including combination feeds, will be introduced in the near future. Currently, available:

Name (links to section homepage)	RSS version	Link to feed
<a href="#">Loosely Coupled weblog</a>	0.92	<a href="#">XML</a>
<a href="#">Loosely Coupled news aggregator</a>	0.92	<a href="#">XML</a>
<a href="#">Loosely Coupled news releases live feed</a>	2.0	<a href="#">XML</a>
<a href="#">Loosely Coupled feature articles</a>	2.0	<a href="#">XML</a>
<a href="#">Loosely Coupled glossary</a>	2.0	<a href="#">XML</a>
<a href="#">AppSwitching Diary</a> (currently dormant)	0.92	<a href="#">XML</a>

\* JavaScript options are in beta and subject to change. The number shown is the number of items in the feed, as they are provided as headlines only (h) or as headlines plus truncated descriptions (d).

JavaScript options are JavaScript files that can be embedded in any web page using an HTML

```
<script src="http://www.looselycoupled.com/news/releases.js"></script>
```

Σχήμα 48 Περιεχόμενο website που προσφέρεται με μορφή RSS

**RSS at Harvard Law**  
Syndication technology hosted by the Berkman Center

### RSS 2.0 Specification

Contents

- [What is RSS?](#)
- [Sample files](#)
- [About this document](#)
- [Required channel elements](#)
- [Optional channel elements](#)
- [Elements of <item>](#)
- [Comments](#)
- [Extending RSS](#)
- [Roadmap](#)
- [License and authorship](#)

**What is RSS?**

RSS is a Web content syndication format.

Its name is an acronym for *Really Simple Syndication*.

RSS is a dialect of XML. All RSS files must conform to the XML 1.0 **specification**, as published on the World Wide Web Consortium (W3C) website.

A summary of **RSS version history**.

At the top level, a RSS document is a <rss> element, with a mandatory attribute called version, that specifies the version of RSS that the document conforms to. If it conforms to this specification, the version attribute must be 2.0.

Subordinate to the <rss> element is a single <channel> element, which contains information about the channel (metadata) and its contents.

**Sample files**

Here are sample files for: RSS **0.91**, **0.92** and **2.0**.

**RSS Directory**

- About this website
- Specifications
- Feeds
- Aggregators
- Validators
- Howtos/Articles
- Tools
- Utilities

**Subscribe via RSS**

**RSS News** 5:35AM EST

- The Dallas Morning News has **RSS 2.0 feeds**.
- US Department of State **supports RSS 2.0**.
- Business Week **RSS feeds**.
- **RSS news feeds** from South Africa.
- MSNBC's all-new **RSS support. Excellent!**
- **RSS feeds** from Brazil from Folha Online.
- The state of Missouri

Σχήμα 49: Blog για την παρουσίαση των επίσημων προδιαγραφών του RSS 2

That link will take you to a page like this, with a link to the corresponding feed at the top.



• Back to top

Below is a selection of the BBC News pages available for syndication

- XML** [News Front Page](#)
- XML** [World](#)
- XML** [UK](#)
- XML** [England](#)
- XML** [Northern Ireland](#)
- XML** [Scotland](#)
- XML** [Wales](#)
- XML** [Business](#)
- XML** [Politics](#)
- XML** [Health](#)
- XML** [Education](#)
- XML** [Science/Nature](#)
- XML** [Technology](#)
- XML** [Entertainment](#)
- XML** [Have Your Say](#)
- XML** [Magazine](#)
- XML** [Week At a Glance](#)
- XML** [Programmes](#)

Σχήμα 50 BBC News Feeds

**Η ΝΑΥΤΕΜΠΟΡΙΚΗ**  
WWW.NAFTEMPORIKI.GR

4422  
ΧΡΗΜΑΤΙΣΤΡΙΑΚΗ ΕΝΗΜΕΡΩΣΗ

Δευτέρα, 7 Φεβρουαρίου 2005    Αναζήτηση: Στις Ειδήσεις

**ΕΙΔΗΣΕΙΣ / RSS FEEDS**

## RSS Feeds

Οι ειδήσεις της «N» είναι τώρα διαθέσιμες μέσω της υπηρεσίας «RSS feeds». Μέσω αυτής της υπηρεσίας μπορείτε να λαμβάνετε τους τίτλους των ειδήσεων της «N» δωρεάν και να τους ενσωματώνετε στο πρόγραμμα ανάγνωσης ειδήσεων που χρησιμοποιείτε ή στο προσωπικό σας weblog.

**Τι είναι το RSS ;**  
Το RSS (Really Simple Syndication) είναι ένα format ανταλλαγής περιεχομένου βασισμένο στη γλώσσα XML. Ένα κανάλι RSS αποτελείται από μία λίστα στοιχείων που περιέχουν ένα τίτλο καθώς και το σύνδεσμο προς την αντίστοιχη ιστοσελίδα. Στην περίπτωση της υπηρεσίας της «N» το πλήρες περιεχόμενο ενός άρθρου είναι διαθέσιμο στο site της (www.naftemporiki.gr) και είναι προσβάσιμο μέσω της υπηρεσίας «RSS feeds».

**Πως χρησιμοποιώ την υπηρεσία ;**  
Οι πιο συνηθισμένοι τρόποι είναι:

- Χρησιμοποιώντας ένα πρόγραμμα ανάγνωσης ειδήσεων (News Aggregator)
- Ενσωματώνοντας το RSS feed στο προσωπικό σας weblog

Τα προγράμματα ανάγνωσης ειδήσεων είναι αυτόνομες εφαρμογές οι οποίες ανακτούν και εμφανίζουν τα περιεχόμενα των RSS feeds που έχετε επιλέξει. Για να προσθέσετε κάποια κατηγορία στο πρόγραμμα ανάγνωσης ειδήσεων που χρησιμοποιείτε, πηγαίνετε στην κατηγορία που θέλετε να χρησιμοποιήσετε.

**ΔΙΑΘΕΣΙΜΑ ΚΑΝΑΛΙΑ**

<b>XML</b>	<a href="#">Newsroom</a>	RSS
<b>XML</b>	<a href="#">Οικονομία</a>	RSS
<b>XML</b>	<a href="#">Πολιτικά</a>	RSS
<b>XML</b>	<a href="#">Ελλάδα</a>	RSS
<b>XML</b>	<a href="#">Κόσμος</a>	RSS
<b>XML</b>	<a href="#">Επιχειρήσεις</a>	RSS
<b>XML</b>	<a href="#">Αγορές</a>	RSS
<b>XML</b>	<a href="#">Ναυτιλία</a>	RSS
<b>XML</b>	<a href="#">The Wall Street Journal</a>	RSS
<b>XML</b>	<a href="#">Απόψεις</a>	RSS
<b>XML</b>	<a href="#">Κόσμος</a>	RSS
<b>XML</b>	<a href="#">Αθλητικά</a>	RSS
<b>XML</b>	<a href="#">Πολιτιστικά</a>	RSS
<b>XML</b>	<a href="#">Media</a>	RSS

**ΕΠΙΛΟΓΕΣ**

- Θέματα
- Σημερινό Θέμα
- Αρχείο άρθρων
- Φυλακμένα άρθρα
- Υποφορές
- Προτιμήσεις ανωνυμάτων
- Ειδήσεις μέσω email

**ΑΝΑΖΗΤΗΣΗ**

Λέξεις:

Χωρισμένες με κενό

Αναζήτηση

Σχήμα 51: Τα νέα από την εφημερίδα Ναυτεμπορική

HOME
INVESTING >
<b>NEWS</b> v
Business
U.S.
International
Politics
Entertainment
Technology & Science
Sports
Health
Oddly Enough
Life & Leisure
The News Room
Weather
Television
Pictures

REUTERS RECOMMENDS →
REUTERS ALERTNET →
REUTERS RSS <b>XML</b>
INTERACTIVE TV →
MOBILE →



advertisement

Investment Profile Reports  
for 25,000 companies

## Reuters RSS

Reuters news and television is now available through the Reuters RSS service. With Reuters RSS you can take Reuters world class news and television headlines, free of charge, for individual, non-commercial use and incorporate them into your preferred newsreaders and web logs. If you would like Reuters news for your commercial website, please visit: <http://about.reuters.com/media/>. We hope you enjoy this new service from Reuters.com.

### What is RSS?

RSS (Really Simple Syndication) is an XML format designed for sharing web content like news headlines. An RSS service (also known as an RSS channel) consists of a list of items, each of which contains a headline, description, and a link to a web page. For Reuters.com RSS service, the full content, such as a full-text article or video, is made available on [www.reuters.com](http://www.reuters.com) and can be accessed by the link in the RSS service.

### How do I use RSS?

To use RSS, you will need a special RSS news reader or aggregator that will allow you to collect and display RSS service. RSS news readers allow you to view the service you select all together in the same place and, by automatically retrieving updates, stay current with new content soon after it is published.

There are many different news readers available, many for free. Most news readers are applications that you download and install. Some others are web-based service you sign-up for that work inside your browser.

### List of News Readers

After setting up your news reader, you will be ready to use Reuters RSS. To subscribe to Reuters RSS, right-click on one of the channels and select "Copy Shortcut?" or "Copy Link Location?" to copy the location (URL) of the service. Follow the instructions for your particular news reader, and then paste this location wherever it asks for the location of the service you wish to subscribe to.

## Channels

### News

- XML** Top News
- XML** Business News
- XML** US News
- XML** International
- XML** Politics
- XML** Entertainment
- XML** Technology
- XML** Science
- XML** Sports
- XML** Health news
- XML** Oddly Enough
- XML** Life & Leisure

### Television

- XML** World News
- XML** Business
- XML** Entertainment
- XML** Life!
- XML** Oddly Enough

### What are the terms of use?

Reuters offers RSS as a free service to any individual user or non-profit organization subject to the following terms and conditions:


- Use will be for non-commercial purposes.

Σχήμα 52 Reuters News Feeds

Για την συνδρομή στα παραπάνω κανάλια περιεχομένου είναι απαραίτητο κάποιο πρόγραμμα News Aggregator ή αλλιώς Newsreader. Ένα από τα πιο γνωστά προγράμματα αυτής της κατηγορίας είναι το πρόγραμμα Radio της εταιρείας Userland. Το γραφικό περιβάλλον ενός news aggregator λειτουργεί παρόμοια με αυτό ενός γραφικού προγράμματος για ανάγνωση των newsgroups του δικτύου Usenet. Το περιεχόμενο είναι οργανωμένο σε κανάλια και ο χρήστης μπορεί να ενημερώνεται αυτόματα κάθε φορά που γίνεται ανανέωση του περιεχομένου. Στην παρακάτω εικόνα φαίνεται ένα τμήμα του γραφικού περιβάλλοντος για το πρόγραμμα Radio.

## Subscriptions

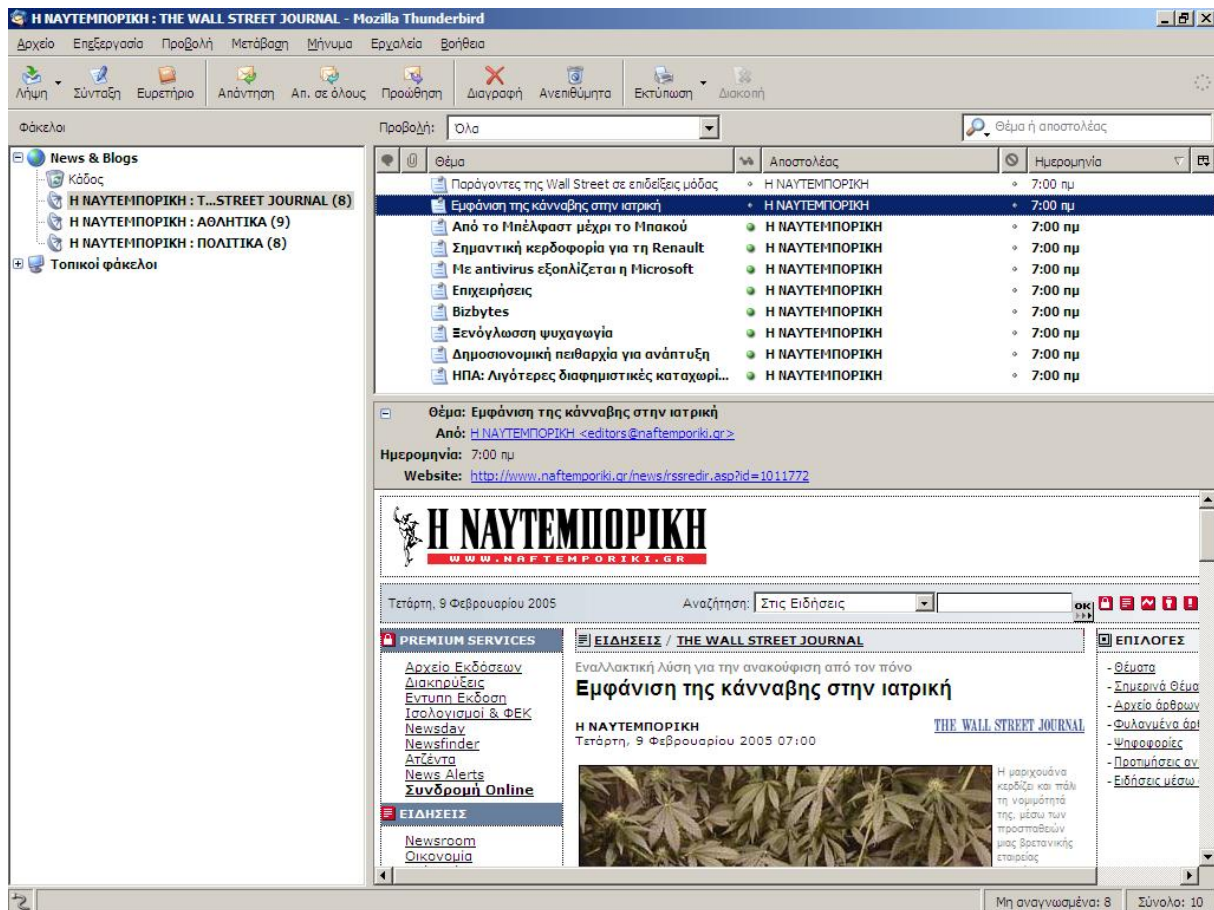
Enter the URL of an XML news feed you want to subscribe to in the box below, then click on the Add button.

The following table lists the XML news feeds you've subscribed to. Included is the name of the source, linked to its Web page, the time or day it last changed, the number of times it has changed since you subscribed, and a link to the XML file for the channel. To delete a subscription, check it and then click on the Unsubscribe button at the bottom of the page. 

<input type="checkbox"/>	<a href="#">Dina Mehta: Skype and VOIP</a>	24/1/2005	1	<input type="button" value="XML"/>
<input type="checkbox"/>	<a href="#">Google Blog - Live</a>	22/1/2005	1	<input type="button" value="XML"/>
<input type="checkbox"/>	<a href="#">Matt Powell</a>	24/1/2005	1	<input type="button" value="XML"/>
<input type="checkbox"/>	<a href="#">The Audioblogging Revolution</a>	24/1/2005	1	<input type="button" value="XML"/>

## Σχήμα 53 Γραφικό περιβάλλον Radio News Aggregator

Δυνατότητα για ανάγνωση syndicated περιεχομένου προσφέρει και το πρόγραμμα Thunderbird που εντάσσεται στο δημοφιλές έργο Mozilla. Στο παρακάτω σχήμα παρουσιάζεται το περιβάλλον ανάγνωσης feeds μέσα από το Thunderbird.



Σχήμα 54: Mozilla Thunderbird RSS Reader






Τα παραπάνω προγράμματα επιτρέπουν την ανάγνωση των news feeds από έναν προσωπικό υπολογιστή. Ωστόσο το syndicated περιεχόμενο μπορεί να επαναχρησιμοποιηθεί και από websites, blogs και wikis. Τα sites αυτά εκτελούν aggregation διαθέσιμου περιεχομένου από τις επιθυμητές πηγές και το επαναδημοσιεύουν. Περισσότερα για τα blogs και wikis στο παράρτημα Β. Για τα feed ισχύουν οι βασικές αρχές προστασίας των πνευματικών δικαιωμάτων. Για την επαναδημοσίευση τους από άλλα site συνήθως απαιτείται από τους αρχικούς δημιουργούς να μην αλλάζεται το περιεχόμενο των feeds και να συμπεριλαμβάνεται αναφορά στην πηγή του. Γενικά ένα website ή ένα intranet μπορεί να εκδώσει με την μέθοδο του syndication περιεχόμενο που αλλάζει σε τακτά χρονικά διαστήματα όπως για παράδειγμα:

- Νέα και ανακοινώσεις
- Λίστες από έγγραφα και αντικείμενα περιεχομένου που βρίσκονται υπό επεξεργασία. Αυτό είναι πολύ χρήσιμο για την διευκόλυνση του workflow σε ένα CMS
- Internet Bookmarks και links
- Ημερολόγια με καταχωρήσεις συναντήσεων και σημαντικών ημερομηνιών
- Διευθύνσεις λιστών email (mailing lists)
- Αποτελέσματα αναζήτησης στο ίδιο το site ή στο Web

Για την αναζήτηση των feeds ενός site υπάρχουν στο Διαδίκτυο ευρετήρια όπου καταχωρούνται τα διαθέσιμα κανάλια περιεχομένου. Παραδείγματα είναι τα ευρετήρια [BlogStreet RSS Discovery](#), [XMLTree](#) και [My.Userland.Com](#).

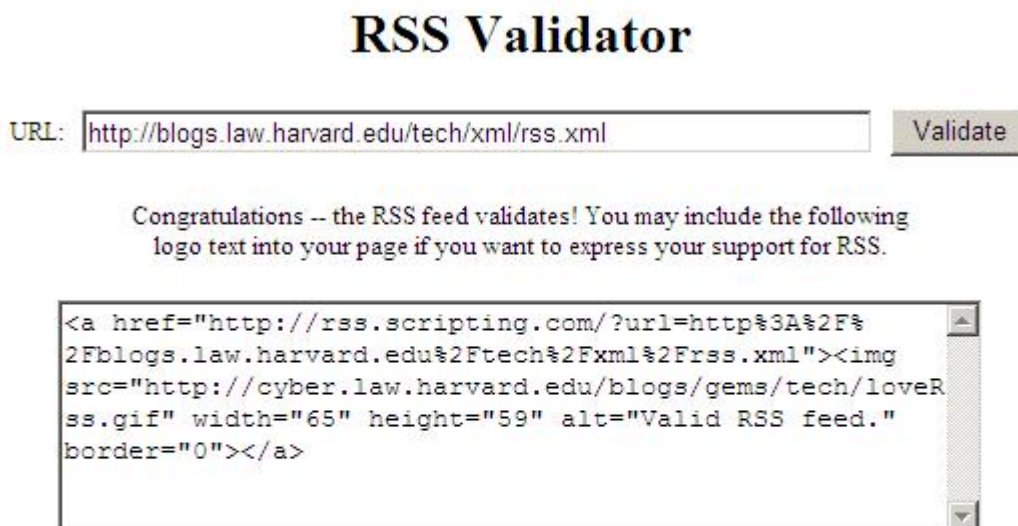
Στην περίπτωση που η αναζήτηση σε ένα τέτοιο ευρετήριο ή η χρήση μιας μηχανής αναζήτησης δεν δώσει αποτελέσματα, τότε το RSS feed μπορεί να προστεθεί από τον χρήστη χειρωνακτικά στο πρόγραμμα ανάγνωσης. Όλα τα site που υποστηρίζουν syndication έχουν στην κεντρική τους σελίδα



ένα link προς το αρχείο περιγραφής των feed. Το link αυτό σηματοδοτείται συνήθως από ένα χρωματιστό εικονίδιο όπως τα ,  και . Επίσης σε ορισμένα site εμφανίζονται τα εικονίδια  ή . Αυτά συνεργάζονται με δύο συγκεκριμένα προγράμματα news aggregators. Τα προγράμματα αυτά είναι το [AmphetaDesk](#) και το [Radio UserLand](#), για απευθείας εγγραφή του χρήστη στα διαθέσιμα κανάλια περιεχομένου.

Μερικά sites προσφέρουν την δυνατότητα της επαναδημοσίευσης των feed που διαθέτουν απευθείας μέσα σε κάποιο άλλο site. Αυτό γίνεται με ενσωμάτωση script κώδικα στην web σελίδα του συνδρομητή των feeds. Με την εκτέλεση του script τα feed προβάλλονται αυτόματα μέσα στην σελίδα την στιγμή που ανανεώνονται από την πηγή. Παρέχεται επίσης και η δυνατότητα παραμετροποίησης της εμφάνισης των feed με χρήση κυρίως CSS. Παράδειγμα είναι η υπηρεσία Meerkat από το δίκτυο εκδόσεων O'Reilly.net ([http://www.oreillynet.com/pub/a/rss/2000/05/09/meerkat\\_api.html](http://www.oreillynet.com/pub/a/rss/2000/05/09/meerkat_api.html)).

Επίσης για την επαλήθευση της εγκυρότητας και επικαιρότητας των RSS αρχείων υπάρχει κατάλληλη υπηρεσία, το RSS validation. Με την υπηρεσία αυτή γίνεται σάρωση (parse) του αρχείου RSS και ελέγχονται οι τιμές των στοιχείων του, όπως τα <pubDate>, <link> και <description> elements. Η υπηρεσία αυτή σε web-based μορφή φαίνεται στο επόμενο σχήμα.



Σχήμα 55 Web-based RSS Validator

## **ΠΑΡΑΡΤΗΜΑ Β**

### **Web logs (Blogs)**

#### ***Τί είναι τα Blog***

Είναι μια μορφή website ή intranet/extranet site με αυξημένες δυνατότητες aggregation και syndication περιεχομένου. Η βασική μονάδα περιεχομένου σε ένα Blog είναι το άρθρο (post). Παράλληλα με τα άρθρα υπάρχουν links που δείχνουν είτε σε άρθρα δημοσιευμένα στο ίδιο το Blog είτε σε πόρους που βρίσκονται σε άλλες τοποθεσίες του Web. Σε κάθε post μπορούν να προστίθενται σχόλια από τους επισκέπτες του Blog.

Διακρίνουμε τις εξής βασικές κατηγορίες blogs:

- Προσωπικά Blog. Είναι ένας χώρος όπου κάθε χρήστης μπορεί να εκφράσει τις απόψεις του, να δημοσιεύσει άρθρα και ιστορίες, κάτι σαν ένα προσωπικό ημερολόγιο. Επίσης ο χρήστης μπορεί να κάνει aggregation και syndication πολλών μορφών περιεχομένου, όπως εικόνες, video, ήχος, κείμενο αλλά και να μοιραστεί εφαρμογές άλλων στο δικό του blog, με χρήση των Web Services. Παράδειγμα είναι το Blogger.com
- Filter Blog. Δημιουργούνται στα πλαίσια μιας online κοινότητας. Κάθε μέλος της κοινότητας αυτής μπορεί να ανταλλάσσει απόψεις και περιεχόμενο με τα υπόλοιπα μέλη. Επίσης στα Blog αυτά δημοσιεύονται post σχετικά με πόρους που ενδιαφέρουν τα μέλη της κοινότητας. Οι λίστες με τους πόρους αυτούς φιλτράρονται και δημοσιεύονται στο blog μετά από αναζητήσεις των μελών της κοινότητας στο Διαδίκτυο. Παράδειγμα είναι το τεχνικού περιεχομένου <http://slashdot.org/>.
- Knowledge Blog. Χρησιμοποιούνται ως πηγή διάχυσης γνώσης. Σε αυτά τα blog κάποιοι ειδικοί δημοσιεύουν τις γνώσεις τους σχετικά με ένα γνωστικό αντικείμενο. Εν συνεχεία όποιος ενδιαφέρεται μπορεί να ενημερώνεται εύκολα σχετικά με τα θέματα αυτά. Πολλά παραδείγματα μπορούν να βρεθούν στο διαδίκτυο αλλά και σε intranets εταιρειών.
- Project Blog. Τα Blog αυτά χρησιμοποιούνται ως μέσο επικοινωνίας και ενημέρωσης των συμμετεχόντων σε ένα έργο (project). Παράδειγμα είναι το προϊόν Microsoft SharePoint Team Services, το οποίο ενσωματώνει δυνατότητες Project Blog.

#### ***Ιστορική αναδρομή***

Το πρώτο Blog που δημιουργήθηκε είναι ταυτόχρονα και το παλιότερο site του παγκόσμιου ιστού. Αυτό δημιουργήθηκε από τον Tim Berners Lee <http://info.cern.ch> για την ευρετηριοποίηση όλων των νέων sites που δημιουργούνταν εκείνη την εποχή. Τα περιεχόμενα του site αυτού έχουν αρχειοθετηθεί από το W3C. Το παλαιότερο Weblog, που είναι ακόμα σε λειτουργία, δημιουργήθηκε από τον Dave Winer. Βρίσκεται στην διεύθυνση [www.scripiting.com](http://www.scripiting.com). Τα τελευταία δύο χρόνια η χρήση των Blog και ειδικά των προσωπικών Blog έχει αυξηθεί δραστικά, φθάνοντας σε πλήθος τις δεκάδες εκατομμύρια.

#### ***Τί είναι τα Wiki***

Τα Wiki είναι Web-based εφαρμογές οι οποίες αναπτύσσονται στην βάση της ιδέας του ελεύθερου περιεχομένου. Πρόκειται για διαδικτυακές κοινότητες εθελοντών, οι οποίοι προσφέρουν τις γνώσεις και τον χρόνο τους στην υλοποίηση έργων λογισμικού. Μια από τις πιο διαδεδομένες χρήσεις των Wiki είναι η δημιουργία Website, τα οποία είναι δυνατόν να παραμετροποιηθούν ως προς το

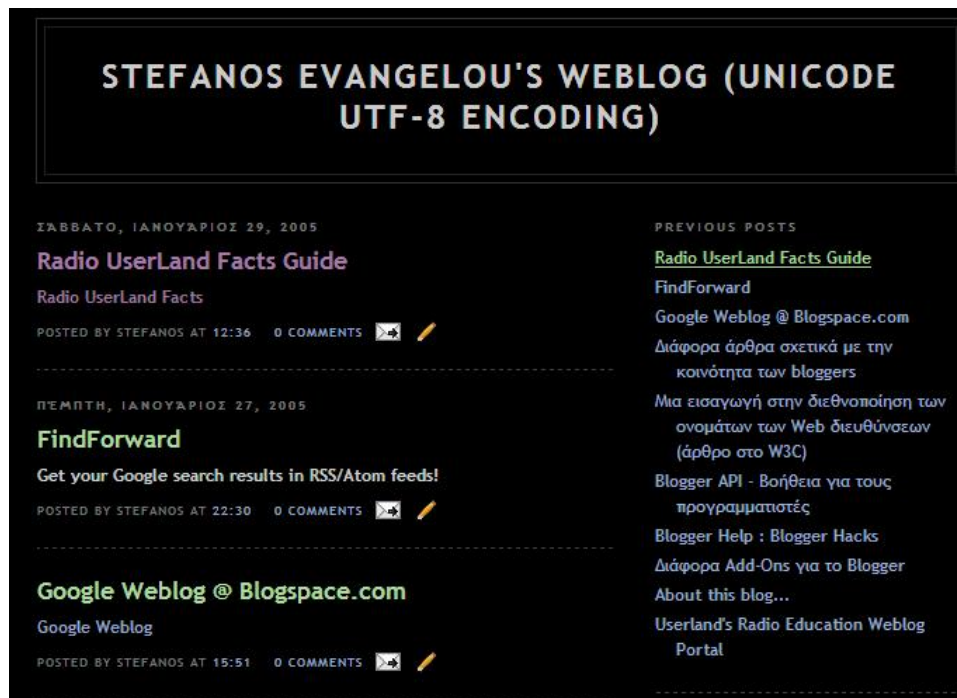
περιεχόμενο και την εμφάνιση από τον κάθε επισκέπτη, ώστε να ταιριάζουν στις δικές του ανάγκες. Ένα τέτοιο παράδειγμα είναι η Wikipedia ([www.wikipedia.org](http://www.wikipedia.org)), η πολύ γνωστή εγκυκλοπαίδεια ανοικτού περιεχομένου [WIK04]. Τα Wiki μαζί με τα Blog είναι οι δύο πιο διαδεδομένες εφαρμογές της φιλοσοφίας του ανοικτού περιεχομένου (open content), δηλαδή περιεχομένου που είναι εύκολα προσβάσιμο από όλους.

## Παραδείγματα Blog

Παραθέτουμε μερικές εικόνες για γίνει καλύτερα αντιληπτή η μορφή των Blog.



Σχήμα 56 Προσωπικό Blog



Σχήμα 57 Ακόμα ένα προσωπικό Blog

Σχήμα 58: Παράδειγμα Filter Blog για software developers

Σχήμα 59 MSDN Knowledge Blog

Μερικά εργαλεία που επιτρέπουν την χρήση του RSS και των blogs από τον απλό χρήστη είναι τα Bloglines (<http://www.bloglines.com/>) , η μηχανή αναζήτησης feeds Feedster (<http://www.feedster.com/>) και το ecto blog (<http://www.kung-foo.tv/ecto/>), μια πολύ χρήσιμη εφαρμογή για συγγραφή και δημοσίευση σε εσωτερικά και εξωτερικά blogs. Εξίσου χρήσιμες είναι οι εφαρμογές της κοινότητας Blogger ([www.blogger.com](http://www.blogger.com)) και η εφαρμογή Radio Userland (<http://radio.userland.com>) , η οποία προσφέρει την δυνατότητα για συγγραφή και δημοσίευση blogs όσο και ανάγνωση feeds με τον ενσωματωμένο reader που διαθέτει. Μια συλλογή από διαθέσιμα εργαλεία για blogs μπορεί να βρει ο αναγνώστης και στην διεύθυνση <http://lists.gpick.com/pages/Blogs.htm>

Οι περισσότερες κοινότητες Blog, όπως για παράδειγμα το blogger.com, προσφέρουν στους χρήστες τους κάποιο API, παρέχοντας έτσι την δυνατότητα για συνεργασία του blog με εξωτερικές εφαρμογές. Με βάση το API ανεξάρτητοι developers δημιουργούν βοηθητικά προγράμματα plug-in. Με τα προγράμματα αυτά προσφέρεται μεταξύ άλλων η δυνατότητα δημοσίευσης μορφών περιεχομένου πέραν του απλού κειμένου σε ένα Blog, όπως για παράδειγμα εικόνες και αρχεία ήχου και video.

# ΠΑΡΑΡΤΗΜΑ Γ

## Περί αδειών λογισμικού

Ξεκινούμε αναλύοντας τις έννοιες και την τεχνική ορολογία που σχετίζονται με την χρήση του λογισμικού, καθώς και αναφέροντας εν συντομία το νομικό καθεστώς που διέπει τις σχετικές ανά περίπτωση άδειες λογισμικού.

### **Ελεύθερο Λογισμικό / Λογισμικό Ανοικτού Κώδικα**

Είναι το λογισμικό που ο καθένας μπορεί ελεύθερα να χρησιμοποιεί, να διανέμει, να αντιγράφει και να τροποποιεί ανάλογα με τις ανάγκες του, χωρίς να απαιτείται η απόκτηση άδειας. Είναι ένα εναλλακτικό μοντέλο ανάπτυξης και χρήσης λογισμικού, στο οποίο η δυνατότητα αλλαγών ή βελτιώσεων παρέχεται στο χρήστη μέσω της ελεύθερης διάθεσης του πηγαίου κώδικα του λογισμικού. Γύρω από αυτή τη λογική δημιουργήθηκε μια τεράστια κοινότητα χρηστών και προγραμματιστών, με βάση το Διαδίκτυο, οι οποίοι συμβάλλουν από κοινού στη συνεχή βελτίωση του λογισμικού, παρέχοντας δωρεάν τις γνώσεις και τα έργα τους.

Η εξάπλωση του ΕΛ/ΛΑΚ έχει στηριχθεί στην ευρεία χρήση του Διαδικτύου και η διαδικασία ανάπτυξης του Διαδικτύου βασίζεται, κατά κύριο λόγο, στο ΕΛ/ΛΑΚ. Το Λογισμικό Ανοικτού Κώδικα, με συνεχείς βελτιώσεις και αυξημένη πλέον φιλικότητα προς το χρήστη, κερδίζει διαρκώς νέους υποστηρικτές παγκοσμίως. Στην εκπαίδευση, στη δημόσια διοίκηση και στις επιχειρήσεις, ενδιαφέρονται, ενημερώνονται και αποκτούν ένα ιδιαίτερα ελκυστικό εργαλείο, αξιόπιστο, σταθερό στη λειτουργία και απαλλαγμένο από σημαντικά κόστη, απόκτησης και αναβαθμίσεων.

Ο όρος open source (ανοικτού κώδικα) χρησιμοποιείται παράλληλα με τον όρο free (ελεύθερο) λογισμικό για να δηλώσουν περίπου το ίδιο. Οι δύο ανεξάρτητες κοινότητες Free Software από την μια πλευρά και Open Source Software από την άλλη, παρόλο που στην πράξη συνεργάζονται για την εκπόνηση μεγάλων projects, έχουν μια διαφοροποίηση στην φιλοσοφία τους και στους όρους που αναφέρονται στις άδειες διανομής. Για την κοινότητα Open Source, και σε αντίθεση με την κοινότητα Free Source, το θέμα αν το λογισμικό θα πρέπει να είναι ελεύθερο κώδικα είναι μια ερώτηση πρακτικού και όχι ηθικού περιεχομένου. Για να περιγράψουμε την διαφορά μεταξύ των ομάδων με ένα ρητό ενός guru της Πληροφορικής: « Το open-source είναι μια μεθοδολογία ανάπτυξης λογισμικού, ενώ το Free software είναι μια πρωτοβουλία με κοινωνικές προεκτάσεις». Όπως προαναφέρθηκε πάντως, στην πράξη οι διαφορές αυτές είναι αμελητέες για τον τελικό χρήστη ή προγραμματιστή.

Πρέπει να ξεκαθαριστεί η διαφορά των εννοιών δωρεάν και ελεύθερο. Μιλώντας για ελεύθερο λογισμικό, σύμφωνα με την άδεια GNU/GPL, αναφερόμαστε κατά κύριο λόγο στην δυνατότητα αναδιανομής και επέκτασής του ελεύθερα από τον οποιονδήποτε και όχι στο κόστος απόκτησής του. Η έννοια free μπορεί να χρησιμοποιείται επίσης καταχρηστικά από εταιρείες λογισμικού για να δηλώσουν προγράμματα για τα οποία διανέμουν την εκτελέσιμη έκδοση χωρίς χρέωση. Δυστυχώς λόγω του ότι η αγγλική γλώσσα μας περιορίζει στην έννοια αυτή (το free μεταφράζεται και ελεύθερο και δωρεάν), έχουν δημιουργηθεί παρανοήσεις σχετικά με λογισμικό που είναι γνωστό ότι διανέμεται ελεύθερα, όπως για παράδειγμα το Linux. Το γεγονός ότι είναι δωρεάν, είναι μόνο ένα από τα πλεονεκτήματα του Linux, αφού επίσης δωρεάν μπορεί να διανέμει μια εταιρία το λειτουργικό της για να αυξήσει την βάση χρηστών της, ώστε να τους εκμεταλλευτεί αργότερα. Η διαφορά είναι ότι το Linux διανέμεται ελεύθερα και αυτό δίνει την δυνατότητα όχι μόνο της χρησιμοποίησής του λειτουργικού, αλλά και της αναδιανομής του, της πώλησής του, τροποποίησής του, της επέκτασής του, πρόσβαση στον πηγαίο κώδικα και συνήθως σε εκτενή τεκμηρίωση.

Οι διανομές του λειτουργικού συστήματος Linux, καθώς και τα περισσότερα συνοδευτικά προγράμματα, είναι ένα κλασικό παράδειγμα λογισμικού που διανέμεται με την άδεια GNU General Public License. Η άδεια αυτή δημιουργήθηκε για να παράγει προγράμματα που θα διανεμηθούν ελεύθερα, αλλά και για να διατηρήσει αυτή την ελευθερία των προγραμμάτων. Έτσι κάποιο πρόγραμμα κάτω από την GNU GPL πρέπει υποχρεωτικά να συνοδεύεται από τον πηγαίο κώδικα του.

Στην περίπτωση που κάποιος τροποποιήσει ένα GNU/GPL πρόγραμμα και θέλει να το διανέμει είναι υποχρεωμένος να διανέμει τον αρχικό κώδικα καθώς και τις δικές του αλλαγές σε αυτόν. Δυστυχώς ή ευτυχώς δεν είναι όλα τα προγράμματα κάτω από την GNU GPL, υπάρχουν πολλά κάτω από μία άδεια πιο χαλαρή από την GNU GPL παρόμοια με την άδεια του UNIX BSD. Άλλες διανομές συμπεριλαμβάνουν και προγράμματα κάτω από διάφορες δικές τους άδειες που δεν έχουν σχέση με τις παραπάνω.

### ***Ελεύθερο Λογισμικό Copylefted***

Πρόκειται για ελεύθερο λογισμικό το οποίο με την άδεια διανομής του δεν επιτρέπει στους χρήστες να προσθέσουν νέους ή τροποποιημένους όρους χρήσης στην δική τους αναδιανομή. Σε αυτή την κατηγορία ανήκει το λογισμικό που υπόκειται στην άδεια GPL.

### ***Ελεύθερο Λογισμικό Non-copylefted***

Σε αντίθεση με την παραπάνω κατηγορία, πρόκειται για ελεύθερο λογισμικό που επιτρέπει στους αναδιανομείς του να τροποποιήσουν την άδεια χρήσης ή να προσθέσουν νέους όρους σε αυτή. Κάτι τέτοιο αφήνει κάθε περιθώριο στον αναδιανομέα για μετατροπή τμημάτων και ή και ολόκληρης της δικής του διανομής σε ιδιόκτητο λογισμικό. Παράδειγμα προγράμματος που ανήκει σε αυτή την κατηγορία είναι το X-Window system.

### ***Λογισμικό Public domain***

Είναι το λογισμικό που δεν έχει προστασία πνευματικών δικαιωμάτων. Αν ο πηγαίος κώδικας διανέμεται και είναι public domain, αυτό αποτελεί μια ειδική περίπτωση ελεύθερου λογισμικού non-copylefted και αυτό σημαίνει ότι μερικά τμήματα ή αντίγραφα του λογισμικού μπορεί να μην είναι καν ελεύθερα.

Σε μερικές περιπτώσεις προγραμμάτων ενδέχεται τα εκτελέσιμα αρχεία να είναι public domain, ενώ ο πηγαίος κώδικας να μην είναι διαθέσιμος. Αυτό δεν αποτελεί περίπτωση ελεύθερου λογισμικού, διότι το ελεύθερο λογισμικό απαιτεί να είναι διαθέσιμος ο πηγαίος κώδικας. Επίσης να αναφερθεί ότι η πλειοψηφία του ελεύθερου λογισμικού δεν είναι public domain. Έχουν copyright και απλά οι κάτοχοι των πνευματικών δικαιωμάτων δίνουν νομικά το δικαίωμα σε καθέναν να κάνει αλλαγές στον πηγαίο κώδικα, πάντα σύμφωνα με την άδεια ελεύθερου λογισμικού.

Παρόλο που ο όρος public domain χρησιμοποιείται μερικές φορές καταχρηστικά για να δηλώσει το δωρεάν ή ελεύθερο λογισμικό, αυτό που κανονικά σημαίνει είναι λογισμικό χωρίς κατοχυρωμένα πνευματικά δικαιώματα. Κάθε πρόγραμμα που δημιουργείται ελεύθερα θεωρείται αυτομάτως ότι αποτελεί πνευματική ιδιοκτησία του δημιουργού του, εκτός και αν αυτός νομικά αποποιηθεί των πνευματικών δικαιωμάτων και καταχωρήσει το πρόγραμμα στην λίστα με τα public domain προγράμματα.

### ***Λογισμικό άδειας GPL***

Όπως προαναφέρθηκε, το GNU (Gnu is Not Unix) GPL (General Public License) είναι μια ειδική άδεια διανομής ελεύθερου λογισμικού. Το project GNU χρησιμοποιεί αυτή την άδεια για την διανομή του λογισμικού που δημιουργείται στους κόλπους του.

### ***Μη ελεύθερο Λογισμικό***

Κάθε είδος λογισμικού που δεν είναι ελεύθερο. Αυτό περιλαμβάνει τόσο το ημιελεύθερο όσο και το ιδιόκτητο λογισμικό (proprietary).



## ***Ημιελεύθερο Λογισμικό***

Δεν είναι ελεύθερο, αλλά συνοδεύεται από ειδική άδεια για μεμονωμένους χρήστες να χρησιμοποιούν, να αντιγράφουν, να κάνουν μετατροπές και να αναδιανέμουν το λογισμικό αυτό για μη εμπορικό όμως σκοπό. Παράδειγμα ενός πολύ γνωστού ημιελεύθερου λογισμικού είναι το γνωστό PGP (Pretty Good Privacy). Το ημιελεύθερο λογισμικό δεν μπορεί να συμπεριληφθεί σε μια διανομή ενός ελεύθερου λειτουργικού συστήματος, διότι τότε αυτόματα το σύστημα καθίσταται μη ελεύθερο και έτσι δεν μπορεί να απευθύνεται σε όλους τους εταιρικούς και απλούς χρήστες, είτε ως βασική είτε ως εμπορική διανομή.

## ***Λογισμικό Proprietary***

Είναι το ιδιόκτητο λογισμικό. Κάθε χρήση, αναδιανομή ή τροποποίηση συστατικών μερών των proprietary προγραμμάτων είτε απαγορεύεται είτε είναι αυστηρά περιορισμένη σε ειδική άδεια, για την επέκταση της οποίας πρέπει να υπάρχει η ρητή συναίνεση του ιδιοκτήτη του προγράμματος. Παράδειγμα ιδιόκτητου λογισμικού είναι το λειτουργικό σύστημα Windows.

## ***Λογισμικό Freeware***

Ο πηγαίος κώδικας δεν είναι διαθέσιμος και οι δημιουργοί των προγραμμάτων αυτών επιτρέπουν την ελεύθερη αναδιανομή τους αλλά απαγορεύουν την οποιαδήποτε τροποποίησή τους. Το λογισμικό αυτής της κατηγορίας λοιπόν δεν έχει καμία σχέση με το ελεύθερο λογισμικό.

## ***Λογισμικό Shareware***

Οι δημιουργοί αυτών των προγραμμάτων δίνουν το δικαίωμα ελεύθερης αναδιανομής του προγράμματος, αλλά για να μπορέσει κάποιος να συνεχίσει να χρησιμοποιεί ένα αντίγραφο του προγράμματος, ακόμα και μη κερδοσκοπικά, είναι υποχρεωμένος να καταβάλλει κάποιο αντίτιμο στον φορέα – δημιουργό. Ο πηγαίος κώδικας στα περισσότερα shareware προγράμματα δεν είναι διαθέσιμος. Συνεπώς και το shareware λογισμικό επίσης δεν είναι ελεύθερο.

## ***Ιδιωτικό Λογισμικό***

Γνωστό επίσης και με τον όρο custom software, το ιδιωτικό λογισμικό είναι λογισμικό που αναπτύσσεται για τις ανάγκες ενός συγκεκριμένου πελάτη, πιθανώς για έναν οργανισμό ή εταιρεία. Ο πελάτης αυτός έχει δικαίωμα πλήρους χρήσης και εκμετάλλευσης του προγράμματος για τις δικές του ανάγκες, αλλά δεν δικαιούται να το διανείμει στο εξωτερικό του περιβάλλον κάτω από οποιεσδήποτε συνθήκες.

## ***Εμπορικό Λογισμικό***

Είναι το λογισμικό που αναπτύσσεται από μια εταιρεία η οποία στοχεύει να έχει κέρδος από την εμπορική εκμετάλλευσή του. Οι όροι εμπορικό και ιδιόκτητο λογισμικό διαφέρουν. Στην πράξη βέβαια τα περισσότερα εμπορικά προγράμματα είναι ιδιόκτητα, αλλά να μην ξεχνάμε ότι υπάρχει και εμπορικό λογισμικό που ανήκει στην κοινότητα του free software, όπως επίσης υπάρχει και μη εμπορικό λογισμικό που δεν είναι ελεύθερο λογισμικό. Το ελεύθερο λογισμικό δεν αποκλείει κάποιο πρόγραμμα να είναι εμπορικό. Υπάρχουν παραδείγματα πολλών εταιριών που αναπτύσσουν και πωλούν ελεύθερο λογισμικό.

# ΠΑΡΑΡΤΗΜΑ Δ

## Πρότυπα στην Πληροφορική

### Γενικά περί προτύπων

Τα πρότυπα γενικά διακρίνονται σε δύο κατηγορίες [TAN96]:

- τα de facto , τα οποία επικρατούν λόγω της ευρείας υιοθέτησής τους από την αγορά. Παράδειγμα de facto προτύπου είναι η αρχιτεκτονική των IBM PC συμβατών υπολογιστών.
- τα de jure, τα οποία θεσπίζονται από τους οργανισμούς τυποποίησης. Παράδειγμα de jure προτύπου είναι η γλώσσα XML, που έχει προτυποποιηθεί από το World Wide Web Consortium (W3C).

Οι οργανισμοί τυποποίησης διακρίνονται σε:

- Τοπικής εμβέλειας, οι οποίοι δημιουργούν πρότυπα που έχουν ισχύ σε τοπικό, εθνικό συνήθως, επίπεδο. Παράδειγμα αυτής της κατηγορίας είναι ο οργανισμός NIST (National Institute of Standards and Technology) που παράγει πρότυπα για όλα τα υπουργεία της κυβέρνησης των ΗΠΑ, εκτός του υπουργείου Άμυνας, το οποίο διαθέτει τα δικά του πρότυπα. Άλλο παράδειγμα είναι ο αμερικανικός οργανισμός τυποποίησης ANSI (American National Standards Institute).
- Διεθνούς εμβέλειας, οι οποίοι επιβάλλουν διεθνή πρότυπα. Παράδειγμα αυτής της κατηγορίας είναι ο μεγαλύτερος επαγγελματικός οργανισμός στον κόσμο, το IEEE (Institute of Electrical and Electronics Engineers). Άλλο παράδειγμα είναι ο οργανισμός ISO (International Standards Organization). Ο ISO αποτελείται από 89 κράτη μέλη, κάθε ένα από τα οποία εκπροσωπείται στον ISO από τον δικό του εθνικό οργανισμό τυποποίησης. Η Ελλάδα συμμετέχει στον ISO μέσω του ΕΛΟΤ (Ελληνικός Οργανισμός Τυποποίησης). Τέλος ένα άλλο παράδειγμα οργανισμού τυποποίησης είναι η επιτροπή ITU (International Telecommunications Union), η οποία θεσπίζει όλα τα διεθνή πρότυπα στις τηλεπικοινωνίες.

Παράλληλα με την παραπάνω κατηγοριοποίηση, οι οργανισμοί τυποποίησης μπορεί να είναι είτε κυβερνητικοί είτε εθελοντικοί.

### Πρότυπα για το Διαδίκτυο

Το Διαδίκτυο αποτελεί μια ξεχωριστή περίπτωση στο θέμα της προτυποποίησης. Οι αρχές που εποπτεύουν και συμβάλλουν στην υιοθέτηση προτύπων γι' αυτό είναι στην συντριπτική τους πλειοψηφία εθελοντικοί οργανισμοί, στους οποίους συμμετέχουν και άτομα από επιχειρήσεις και κυβερνητικούς οργανισμούς.

Ο σημαντικότερος φορέας τυποποίησης του Internet είναι το IAB (Internet Architecture Board). Ιδρύθηκε με την επωνυμία Internet Activities Board από το Υπουργείο Άμυνας των ΗΠΑ, με στόχο να επιβλέπει το ερευνητικό δίκτυο ARPANET. Εν συνεχεία μετονομάστηκε σε Internet Architecture Board. Ενώ αρχικά το IAB είχε καθαρά ερευνητικό προσανατολισμό, το 1989 αναδιοργανώθηκε και δημιουργήθηκαν τρεις φορείς, που τοποθετήθηκαν υπό την επίβλεψη του IAB. Οι φορείς αυτοί είναι:

- Internet Research Task Force (IRTF). Ασχολείται με μακροπρόθεσμα ερευνητικά θέματα.
- Internet Engineering Task Force (IETF). Ασχολείται με βραχυπρόθεσμα πρακτικά θέματα λειτουργίας του Internet. Αποτελείται από ομάδες εργασίες, οι οποίες οργανώνονται ανά θεματικό αντικείμενο. Παραδείγματα θεματικών αντικειμένων για τις επιτροπές είναι η ασφάλεια των υπολογιστικών συστημάτων, τα πρωτόκολλα TCP/IP και τα πρωτόκολλα διαχείρισης δικτύων. Προκειμένου μια ιδέα να ελεγχθεί περαιτέρω για προτυποποίηση, πρέπει πρώτα εξηγηθεί λεπτομερώς σε ειδικό τεχνικό έγγραφο που ονομάζεται RFC (Request for Comments). Αφού το RFC δημοσιευθεί και προσελκύσει το ενδιαφέρον της κοινότητας, πρέπει να τεθεί σε λειτουργία μέσω κάποιας συγκεκριμένης υλοποίησης. Αφού ελεγχθεί για χρονικό διάστημα τουλάχιστον τεσσάρων μηνών, κατόπιν το IAB προάγει το RFC αυτό σε

Draft Standard και εν συνεχεία σε Proposed Standard. Τελικά αν όλες οι πλευρές πεισθούν για την χρησιμότητα της νέας προτεινόμενης τεχνολογίας, τότε το αντίστοιχο RFC γίνεται πλέον Internet Standard. Ένας σημαντικός φορέας εντός του IETF είναι το Internet Engineering Steering Group (IESG). Είναι υπεύθυνο για την διαχείριση των προτύπων που προτείνονται από το IETF. Το IESG είναι ο φορέας που δίνει την τελική επικύρωση για την κατοχύρωση μιας τεχνολογίας ως Internet standard.

- Internet Society (IS). Απαρτίζεται από άτομα που ενδιαφέρονται εθελοντικά για το Internet και συνεισφέρουν με την εργασία τους στην επίλυση διαφόρων ζητημάτων.

Τα θέματα που αφορούν ειδικά στον Παγκόσμιο Ιστό (World Wide Web) ελέγχονται από τον οργανισμό World Wide Web Consortium (W3C), ο οποίος ιδρύθηκε από τον δημιουργό του παγκόσμιου ιστού, Tim Berners Lee. Το W3C αντίστοιχα με τους υπόλοιπους οργανισμούς τυποποίησης διεξάγει έρευνα και τελικά δημιουργεί συστάσεις (Technical Recommendations TR) που σχετίζονται με τον παγκόσμιο ιστό.

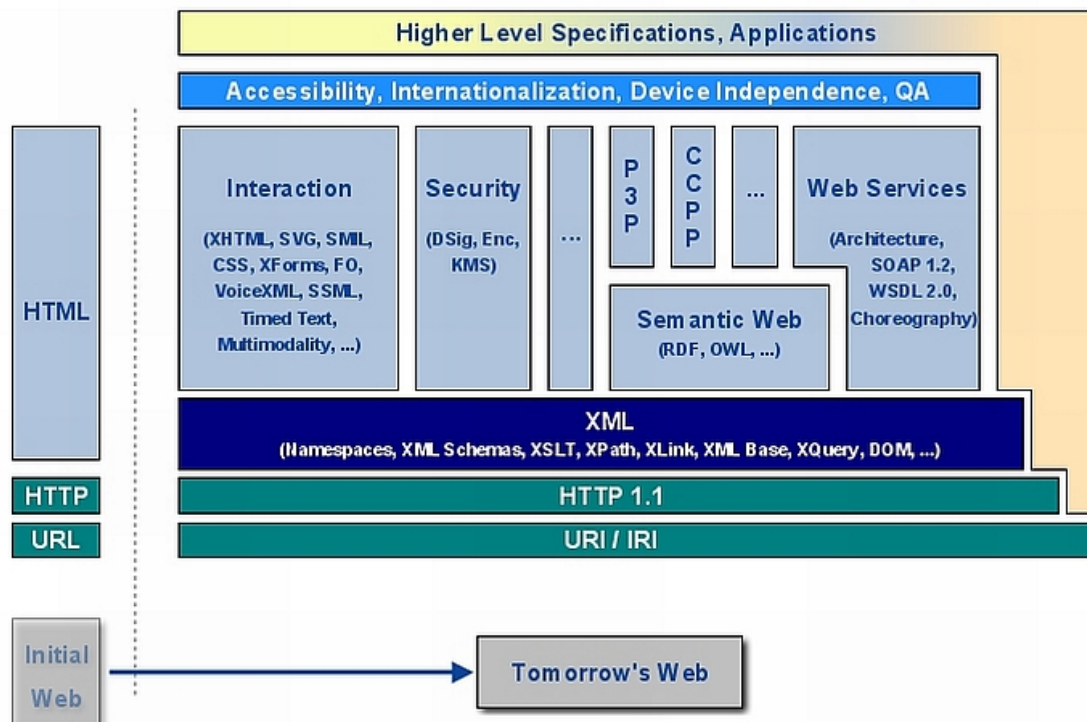
Αποτελείται από ομάδες εργασίας (Working Group - WG). Κάθε ομάδα εργασίας ασχολείται με την μελέτη ενός ειδικού προτύπου, για παράδειγμα η XML WG. Η ομάδα αυτή απαρτίζεται από υπο-ομάδες που ασχολούνται με ένα ειδικό θέμα του προτύπου της γλώσσας XML, όπως είναι για παράδειγμα οι τεχνολογίες XSL, XML Schema, XML Voice, κ.ά. Παραδείγματα συστάσεων που έχουν θεσπιστεί από το W3C είναι η WSDL και το SOAP, που αποτελούν την τεχνολογική βάση για τις Διαδικτυακές Υπηρεσίες. Το W3C εκδίδει τεχνικές αναφορές ([www.w3c.org/TR](http://www.w3c.org/TR)), οι οποίες μπορεί να είναι:

- Working Drafts, δηλαδή προσχέδια της μελέτης μια ομάδας εργασίας για κάποιο υπό διαμόρφωση πρότυπο
- Candidate Recommendations, δηλαδή τελικές προτάσεις με αναλυτική περιγραφή όλων των λεπτομερειών μιας υπό εξέταση σύστασης. Μόλις ένα candidate recommendation εγκριθεί από τις κατάλληλες επιτροπές του W3C δημοσιεύεται ως Technical Recommendation.
- Technical Recommendations, δηλαδή η τελική μορφή της σύστασης (πρότυπου) που έχει θεσπίσει το W3C και συνιστά την ευρεία υιοθέτησή του από τους κατασκευαστές και τους χρήστες. Οι συστάσεις του W3C ενδέχεται να υπόκεινται σε αλλαγές και να βελτιώνονται. Οι αναβαθμίσεις αυτές γίνονται γνωστές από το ίδιο URL.

Παράλληλα με το W3C δραστηριοποιείται το OASIS (Organization for the Advancement of Structured Information Standards), το οποίο είναι φορέας που υποστηρίζει και συμπληρώνει την δραστηριότητα του W3C ειδικά για τις εφαρμογές της γλώσσας XML.

Το W3C δραστηριοποιείται παράλληλα με το IETF. Το W3C ασχολείται κυρίως με θέματα της αρχιτεκτονικής και των πόρων του παγκόσμιου Ιστού (World Wide Web), ενώ το IETF ασχολείται κυρίως με θέματα πρωτοκόλλων στο Διαδίκτυο, όπως το TCP/IP και DNS. Ενδεχομένως σε κάποια θέματα να υπάρχει παράλληλη ενασχόληση των δύο φορέων. Για τον λόγο αυτό έχουν οριστεί κανόνες συνεργασίας μεταξύ IETF και W3C, ώστε να προκύπτει το καλύτερο δυνατό αποτέλεσμα. Το IETF είναι ένας απόλυτα εθελοντικός και ανοικτός οργανισμός, στον οποίο τα μέλη μπορούν να συμμετέχουν ελεύθερα και να προσφέρουν την εργασία τους. Αντιθέτως το W3C απαρτίζεται από μεγάλους φορείς, όπως επιχειρήσεις και άλλους οργανισμούς, οι οποίοι ως μέλη οφείλουν να καταβάλλουν συνδρομή.

Το W3C μέχρι σήμερα έχει δημιουργήσει παραπάνω από ογδόντα συστάσεις (Technical Recommendations). Κεντρικός του στόχος είναι η μετεξέλιξη του σημερινού συμβατικού Web στην μορφή του σημασιολογικού ιστού (Semantic Web). Η οργάνωση της μελλοντικής αυτής μορφής και οι επικρατούσες τεχνολογίες φαίνονται καλύτερα στο παρακάτω σχήμα:



Σχήμα 60: Το μελλοντικό Web

## ΠΑΡΑΡΤΗΜΑ Ε: XSLT stylesheets

Με τα παρακάτω stylesheets μετατρέπουμε το περιεχόμενο που επιστρέφεται από την κλήση μιας λειτουργίας (μεθόδου) του COMMET από xml σε RSS2 και OPML μορφή. Συγκεκριμένα υποστηρίζονται οι λειτουργίες “PrintMySubtreeNodes” και “PrintMyResources”.

### Κώδικας για το αρχείο “PrintMySubtreeNodes-to-OPML.xsl”

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method = "xml" indent = "yes" />

<!-- Λειτουργία COMMET PrintMySubtreeNodes - Στέφανος Ευαγγέλου DBLAB -->

<xsl:template match="/">
<opml>
<head>
<title>Τrexon kombos me olokliro to ypodentro toy</title>
</head>
<body>

<xsl:apply-templates/>

</body>
</opml>
</xsl:template>

<!-- Εισαγωγή του πρώτου στοιχείου outline που αντιστοιχεί στον τρέχοντα κόμβο. Αν ο κόμβος αυτός δεν έχει
ιδιότητες στο .xml αρχείο τότε πρέπει να δημιουργείται το ιεραρχικό opml και το outline-ρίζα να είναι τελείως
κενό -->

<xsl:template match="content/object">
<outline>

<xsl:if test="/content/object/properties/property[@name='nid']/value/text() != &quot;&quot; ">
<xsl:attribute name="text">
<xsl:value-of select="/content/object/properties/property[@name='nid']/value/text()"/>
</xsl:attribute>
</xsl:if>

<xsl:if test="/content/object/properties/property[@name='nname']/value/text() != &quot;&quot; ">
<xsl:attribute name="title">
<xsl:value-of select="/content/object/properties/property[@name='nname']/value/text()"/>
</xsl:attribute>
</xsl:if>

<xsl:if test="/content/object/properties/property[@name='ndescr']/value/text() != &quot;&quot; ">
<xsl:attribute name="description">
<xsl:value-of select="/content/object/properties/property[@name='ndescr']/value/text()"/>
</xsl:attribute>
</xsl:if>

<xsl:apply-templates/>

</outline>
</xsl:template>
```

<!-- Ενώ το προηγούμενο template αναφέρεται μόνο στο πρώτο ιεραρχικά object, δηλαδή στον τρέχοντα κόμβο, το επόμενο template αναφέρεται στους υποκόμβους -->

```
<xsl:template match="object/descendants">
<xsl:for-each select="object">
<outline>

<xsl:if test="/properties/property[@name='nid']/value/text() != &quot;&quot; ">
<xsl:attribute name="text">
<xsl:value-of select="/properties/property[@name='nid']/value/text()"/>
</xsl:attribute>
</xsl:if>

<xsl:if test="/properties/property[@name='nname']/value/text() != &quot;&quot; ">
<xsl:attribute name="title">
<xsl:value-of select="/properties/property[@name='nname']/value/text()"/>
</xsl:attribute>
</xsl:if>

<xsl:if test="/properties/property[@name='ndescr']/value/text() != &quot;&quot; ">
<xsl:attribute name="description">
<xsl:value-of select="/properties/property[@name='ndescr']/value/text()"/>
</xsl:attribute>
</xsl:if>
```

```
<xsl:apply-templates/>
```

```
</outline>
```

```
</xsl:for-each>
```

```
</xsl:template>
```

<!-- Οι μόνες ιδιότητες κάθε κόμβου που μας ενδιαφέρουν είναι αυτές που αντιστοιχίζονται σε attributes του στοιχείου outline. Αυτό το έχουμε ήδη ορίσει από προηγούμενο template. Τις υπόλοιπες ιδιότητες έτσι κι αλλιώς τις αγνοούμε. -->

```
<xsl:template match="property[@name='nid']">
</xsl:template>

<xsl:template match="property[@name='nname']">
</xsl:template>

<xsl:template match="property[@name='ndescr']">
</xsl:template>

<xsl:template match="property[@name='ntype']">
</xsl:template>

<xsl:template match="property[@name='nsubtype']">
</xsl:template>

<xsl:template match="property[@name='ncreator']">
</xsl:template>

<xsl:template match="property[@name='nsavepath']">
</xsl:template>

<xsl:template match="property[@name='ncolor']">
</xsl:template>
```

```
<xsl:template match="property[@name='ndatecreated']">
</xsl:template>

<xsl:template match="property[@name='ndate']">
</xsl:template>

<xsl:template match="property[@name='nstatus']">
</xsl:template>

<xsl:template match="property[@name='nhost']">
</xsl:template>

<xsl:template match="property[@name='nrespons']">
</xsl:template>

<xsl:template match="portalinfo|actions|relation">
</xsl:template>

</xsl:stylesheet>
```

## Κώδικας για το αρχείο "PrintMyResources-to-RSS2.xsl"

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" indent="yes" />

<!-- Λειτουργία COMMET PrintMyResources - Στέφανος Ευαγγέλου DBLAB -->

  <!-- Έλεγχος υποστοιχείων του RSS channel-->
  <xsl:template match="/content">
    <rss version="2.0">
      <channel>
        <xsl:apply-templates/>
      </channel>
    </rss>
  </xsl:template>
  <xsl:template match="property[@name='nid']">
    <link>
      <xsl:value-of select="value"/>
    </link>
  </xsl:template>
  <xsl:template match="property[@name='ndescr']">
    <description>
      <xsl:value-of select="value"/>
    </description>
  </xsl:template>
  <xsl:template match="property[@name='nname']">
    <title>
      <xsl:value-of select="value"/>
    </title>
  </xsl:template>

  <xsl:template match="property[@name='ncreator']">
    <webMaster>
      <xsl:value-of select="object/properties/property[@name='ufirstname']"/>_<xsl:value-of
select="object/properties/property[@name='ulastname']"/>
    </webMaster>
  </xsl:template>

  <!-- Για τα στοιχεία που δεν ενδιαφέρομαι, αλλά δημιουργώ ένα κενό template ώστε να παρακάμψω
την εκτέλεση των built-in templates, που έχουν
σαν αποτέλεσμα να τυπώνονται στοιχεία που δεν θέλω -->

  <xsl:template match="property[@name='nsavepath']">
  </xsl:template>

  <xsl:template match="property[@name='ncolor']">
  </xsl:template>

  <xsl:template match="portalinfo">
  </xsl:template>

  <xsl:template match="actions">
  </xsl:template>

  <xsl:template match="property[@name='ndatecreated']">
    <LastBuildDate>
      <xsl:value-of select="value"/>
    </LastBuildDate>
  </xsl:template>
```



```

<xsl:template match="property[@name='ntype']">
  <Category>
    <xsl:value-of select="value"/>
  </Category>
</xsl:template>

```

<!-- Έλεγχος υποστοιχείων του RSS item -->

```

<xsl:template match="resources">
<xsl:for-each select="object">
<item>
<xsl:apply-templates/>
</item>
</xsl:for-each>
</xsl:template>

```

```

<xsl:template match="property[@name='rid']">
<link>
<xsl:value-of select="value"/>
</link>
</xsl:template>

```

```

<xsl:template match="property[@name='rname']">
<title>
<xsl:value-of select="value"/>
</title>
</xsl:template>

```

```

<xsl:template match="property[@name='rdescr']">
<description>
<xsl:value-of select="value"/>
</description>
</xsl:template>

```

<!-- Εδώ όπως και σε κάποια στοιχεία προηγουμένως, έχω να αντιμετωπίσω το θέμα με το στοιχείο object σαν υποστοιχείο στο property

Κάποια στοιχεία property δεν έχουν μόνο name και value, αλλά έχουν ολόκληρο υποδέντρο που ξεκινά από το object, βλ. και schema της λειτουργίας-->

```

<xsl:template match="property[@name='rowner']">
<author>
<xsl:value-of select="object/properties/property[@name='ulastname']"/>
</author>
</xsl:template>

```

```

<xsl:template match="property[@name='rtype']">
<category>
<xsl:value-of select="value"/>
</category>
</xsl:template>

```

```

<xsl:template match="property[@name='rpublished']">
<comments>
<xsl:value-of select="value"/>
</comments>
</xsl:template>

```

```

<xsl:template match="property[@name='rsmalltext']">
<enclosure>
<xsl:value-of select="value"/>
</enclosure>
</xsl:template>

<xsl:template match="property[@name='rposted']">
<PubDate>
<xsl:value-of select="value"/>
</PubDate>
</xsl:template>

<xsl:template match="property[@name='rauthor']">
</xsl:template>

<xsl:template match="property[@name='rtext']">
</xsl:template>

<xsl:template match="property[@name='rlink']">
</xsl:template>

<xsl:template match="property[@name='rimage']">
</xsl:template>

<xsl:template match="property[@name='rsmalltext']">
</xsl:template>

<xsl:template match="property[@name='rtext_html']">
</xsl:template>

<xsl:template match="property[@name='rsmalltext_html']">
</xsl:template>

<xsl:template match="property[@name='rdate']">
</xsl:template>

<xsl:template match="property[@name='reventdate']">
</xsl:template>

<xsl:template match="property[@name='rfile']">
</xsl:template>

<xsl:template match="property[@name='rfile_type']">
</xsl:template>

<xsl:template match="property[@name='rfile_size']">
</xsl:template>

<xsl:template match="portalinfo|actions|relation">
</xsl:template>

```

```
</xsl:stylesheet> <!-- Τέλος αρχείου XSLT stylesheet -->
```

# ΑΝΑΦΟΡΕΣ

## Βιβλία και άρθρα στο Web

1. [ATO04] Άρθρα στο Web, <http://www.atomenabled.org> (Προδιαγραφές και εργαλεία σχετικά με το πρωτόκολλο Atom)
2. [BOI02] Bob Boiko: “The Content Management Domain – CM Bible”, Wiley Publishing, June 2002
3. [BOI02] Bob Boiko “The Content Management Domain – CM Bible”, Wiley Publishing, June 2002  
<http://bibliophile.ischool.washington.edu/cmdomain/>
4. [BYR03] Tony Byrne : The CMS Report, CMSWatch, September 2003
5. [CMS04] Άρθρο στο Web, Λειτουργικά χαρακτηριστικά κατηγοριοποιημένα με βάση τον κύκλο ζωής και τον βασικό άξονα collection->management->publication . Επίσης προσφέρεται δυνατότητα σύγκρισης δύο πραγματικών συστημάτων ως προς τα χαρακτηριστικά που προσφέρουν.  
<http://www.cmsreview.com/Comparator.html>
6. [CMW04] Άρθρο στο Web , CMSWatch : Όταν ένα προϊόν καλύπτει ένα μέρος της λειτουργικότητας ενός μεγαλύτερου συστήματος, εξακολουθεί να είναι ένα CMS; Ποιά είναι τα ελάχιστα διακριτικά χαρακτηριστικά;  
[http://www.cmswatch.com/Features/ProductWatch/FeaturedProduct/?feature\\_id=94](http://www.cmswatch.com/Features/ProductWatch/FeaturedProduct/?feature_id=94)
7. [CMW04] Άρθρο στο Web , CMSWatch : Μια θεώρηση για το ιδανικό CMS , με βάση σύνθεση λειτουργικών χαρακτηριστικών από διαφορετικά εμπορικά και ελεύθερα πακέτα (CMSWatch)  
[http://www.cmswatch.com/Features/ProductWatch/FeaturedProduct/?feature\\_id=96](http://www.cmswatch.com/Features/ProductWatch/FeaturedProduct/?feature_id=96)
8. [CMW04] Άρθρο στο Web , CMSWatch : Κατηγοριοποίηση όλων των προϊόντων CMS με κριτήριο το μέσο κόστος και το κοινό στο οποίο απευθύνονται (CMSWatch)  
<http://www.cmswatch.com/ContentManagement/Products/>
9. [COC04] Άρθρο στο Web, Application Syndication using Apache Cocoon  
(<http://cocoon.apache.org/index.html>)
10. [COM01] Douglas E. Comer : Computer Networks and Internets, Prentice Hall, June 2001
11. [CYB04] Άρθρο στο Web, Παρουσίαση μιας ολοκληρωμένης λύσης CMS με άξονα τον πυρήνα του CMS (repository) και τα περιφερειακά modules (εταιρεία CyberTeams)  
<http://www.cyberteams.com/company/whitepapers/WebContentManagement.html>
12. [DBL04] Τεκμηρίωση του συστήματος COMMET  
<http://www.dblab.ntua.gr/COMMET> και <http://commet.dbnet.ntua.gr>
13. [DD02] Harvey & Paul Deitel: VB.NET, How to program , Prentice Hall , June 2002
14. [DEI03] Deitel Developer Series: Web Services – A Technical Introduction, Prentice Hall, March 2003
15. [DUB04], Άρθρα στο Web, <http://dublincore.org> (Dublin Core Metadata Set)
16. [EN01] Elmasri – Navathe : Θεμελιώδεις Αρχές Συστημάτων Βάσεων Δεδομένων Τόμοι I και II , Εκδόσεις Δίαυλος, Ιούνιος 2002
17. [GRA02] Graham Glass: Web Services – Building Blocks for Distributed Systems, Prentice Hall Publishing October 2002
18. [HAR04] Blog στο Web, <http://blogs.law.harvard.edu/tech/rss> (RSS 2.0 Official Specification)
19. [ION04] Άρθρο στο Web, <http://www.iona.com/pressroom/2005/20050208.htm> (Transactional Web Services specifications)
20. [IP98] Valerie Illingworth, Ian Pyle : Το επιστημονικό λεξικό της Πληροφορικής, Εκδόσεις Κλειδάριθμος, June 1998
21. [MAR00] Benoit Marchal: XML by Example, Que Publishing, June 2000
22. [MIC04] Dimitrios Michelinakis : Open Source Content Management Systems: An Argumentative Approach – University of Warwick, August 2004
23. [MSD04] MSDN whitepaper about Content Management System Features  
<http://www.microsoft.com/technet/archive/itsolutions/ecommerce/maintain/operate/contmgmt.msp#EDAA>

24. [NAK02] Russell Nakamo, Interwoven Inc. : Web Content Management Systems, A collaborative approach, Addison Wesley, June 2002
25. [NEW04] , Άρθρα στο Web, <http://www.xmlnews.org/NewsML> (NewsML XML format)
26. [NIT04] Άρθρα στο Web, <http://www.nitf.org/> (NITF XML format)
27. [OMN04] PDF Whitepaper by Omniupdate : Functional Characteristics  
<http://www.delijst.net/delijst/pdf/omniupdate.pdf>
28. [OPM04], Άρθρα στο Web, <http://www.opml.org/spec> (Πρωτόκολλο OPML)
29. [OVU04] PDF Whitepaper by Ovum Research : Strategies, Technologies and Markets  
<http://www.delijst.net/delijst/pdf/ovumcmswhitepaper.pdf>
30. [RKM03] Ann Rockley, Pamela Kostur, Steve Manning : Managing Enterprise Content, A unified Strategy : June 2003
31. [RSS04], Άρθρα στο Web, <http://www.rss-specifications.com/rss-specifications.htm>  
(Λεπτομέρειες σχετικά με το RSS)
32. [STA03] Καθηγητή ΣΗΜΜΥ κ.Στασινόπουλου : Σημειώσεις «XML και σχετικές τεχνολογίες» του μαθήματος ΣΗΜΜΥ «Διαδίκτυο και Εφαρμογές» από το URL: [www.telecom.ntua.gr/edu/di](http://www.telecom.ntua.gr/edu/di)
33. [STE04] Άρθρο στο Web, StepTwo : Συστήματα παραπλήσια των CMS (π.χ. Document Management , Digital Asset Management) και σύντομοι ορισμοί τους:  
[http://www.steptwo.com.au/papers/cmb\\_definition/index.html](http://www.steptwo.com.au/papers/cmb_definition/index.html)
34. [STE04] Άρθρο στο Web, StepTwo : Μερικές δοκιμαστικές σελίδες από μια πλήρη αναφορά σχετικά με τα βασικά λειτουργικά requirements για ένα CMS (PDF έγγραφο)  
[http://www.steptwo.com.au/products/toolkit/pdf/CMRTk\\_Sample.pdf](http://www.steptwo.com.au/products/toolkit/pdf/CMRTk_Sample.pdf)
35. [STE04] Άρθρο στο Web, StepTwo : Σειρά από πολύ χρήσιμα archives για όλα τα θέματα σχετικά με τα CMS (π.χ. μελέτη αγοράς, σύγκριση προϊόντων και λειτουργικών χαρακτηριστικών) . Αποτελείται από σχόλια επαγγελματιών στον χώρο της διαχείρισης περιεχομένου.  
[http://www.steptwo.com.au/columntwo/archives/cat\\_content\\_management.html](http://www.steptwo.com.au/columntwo/archives/cat_content_management.html)
36. [SYS04] Άρθρο στο Web, Μια εισαγωγή στο θέμα της εισαγωγής των Web Services στα συστήματα διαχείρισης περιεχομένου για συλλογή και δημοσίευση περιεχομένου.  
<http://www.sys-con.com/xml/article.cfm?id=688>
37. [TAN96] Andrew S.Tanenbaum, Computer Networks, Prentice Hall, 3rd Edition, June 1996
38. [UWA04] Άρθρο στο Web, Υλοποίηση ενός CMS βασισμένου σε Zope για τις ανάγκες των εγγράφων του Πανεπιστημίου του Waterloo (Functional Requirements)  
[http://www.lib.uwaterloo.ca/staff/dlcc/cmfc/CMF\\_Final\\_Report.htm#requirements](http://www.lib.uwaterloo.ca/staff/dlcc/cmfc/CMF_Final_Report.htm#requirements)
39. [VIG04] Άρθρο στο Web, Παράδειγμα λειτουργικών χαρακτηριστικών της Vignette (PDF έγγραφο)  
[http://a112.g.akamai.net/7/112/812/29b2dbe3becb19/www.vignette.com/Downloads/WP\\_CMConsiderations.pdf](http://a112.g.akamai.net/7/112/812/29b2dbe3becb19/www.vignette.com/Downloads/WP_CMConsiderations.pdf)
40. [WEB04] Άρθρο στο Web, Webreference.com : Σχεδίαση ενός CMS από την σκοπιά του προγραμματιστή  
<http://www.webreference.com/new/020523.html>
41. [WDV04] Άρθρο στο Web, Μια ολοκληρωμένη σειρά άρθρων για την προσωποποίηση σε μια εφαρμογή CMS  
<http://www.wdvl.com/Authoring/ASP/Personalization/index.html>
42. [WSA04], Άρθρο στο Web, Web Services for Interactive Applications  
<http://www.oasis-open.org/committees/wsia/>
43. [WSD04], Άρθρο στο Web, Web Services Dynamic Discovery Technologies  
<http://www.mywebservices.org/index.php/article/articleview/66/>
44. [WSE04] Άρθρο στο Web, Web Services Experience Language  
(<http://www.webservices.org/index.php/article/articleview/345/>)
45. [WSI04], Άρθρο στο Web, Web Services Inspection Language από την Microsoft  
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsrvspec/html/ws-inspection.asp>
46. [WSP04], Άρθρο στο Web, Web Services for Remote Portals  
<http://www.oasis-open.org/committees/wsrp/>
47. [WIK04] Άρθρο στο Web, Ένα γενικό WIKI site, που χρησιμεύει ως εργαλείο αναφοράς. Συγκεντρώνει όλα τα διαφορετικά είδη πληροφορίας που σχετίζονται με την ενημέρωση πάνω σε θέματα CMS. Το πιο χρήσιμο είναι το παρακάτω link που αντιστοιχεί σε λίστα με ενδιαφέροντα

άρθρα πάνω στο content management.

<http://www.cmswiki.com/tiki-index.php?page=CmsArticles>

48. [XFR02] Περιγραφή της προδιαγραφής XML Schema του W3C και της λειτουργίας των XML Schema Validators. Σύγκριση του XSD με το DTD.

<http://www.xfront.com>

49. [XMP04], Άρθρο στο Web, XML Terms Short Definitions

[http://www.arbortext.com/html/xml\\_pub\\_network.html](http://www.arbortext.com/html/xml_pub_network.html)

50. [XSL04], Άρθρο στο Web, XSLT Tutorial and Reference

<http://www.zvon.org/xxl/XSLTreference/Output/index.html> και

<http://www.zvon.org/xxl/XSLTutorial/Books/Output/contents.html>

## Διάφορα URL

### XML Aggregation – Syndication

- [www.rfc-editor.org](http://www.rfc-editor.org) (RFCs για τα πρωτόκολλα ATOM – RSS)
- <http://www.w3.org/TR/rdf-schema/> (RDF Schema )
- <http://www.w3.org/TR/xmlschema-0> (XML Schema second edition, Parts 1,2,3)
- <http://www.content-wire.com/Online/Syndication.cfm?ecs=111&cs=1691> (Άρθρο σχετικό με XML syndication)
- <http://www.developer.com/tech/article.php/616671> (Άρθρο “ Java, RDF, and the Virtual Web Part three: Content aggregation” )
- <http://www.methodize.org/nntprrs/index.php> (Χρήση NNTP readers για πρόσβαση σε news feeds)
- <http://www.purplepages.ie/rss/> (Πόροι και άρθρα σχετικά με το RSS)
- <http://channels.lockergnome.com/rss/resources/> (Πόροι για RSS & Atom)
- <http://blog.ziffdavis.com/gallagher/archive/2004/06/05/1208.aspx> (Άρθρο “ RSS, Atom, And the Syndication Standards Dance”)
- <http://www.devx.com/DevX/Article/16190> (Πρακτικό παράδειγμα χρήσης των feeds από το προγραμματιστικό site devx.com)
- <http://www.wdvl.com/Authoring/Languages/CDF.html> (Περιγραφή των αρχείων CDF)
- <http://www.technorati.com/> (Technorati Feeds Search και Live Web feeds)
- <http://xml.coverpages.org/> (Αναλυτικά specifications για όλα τα XML standards για aggregation και syndication, Internet drafts για PRISM, ICE, κ.ά.)
- <http://www.intertwingly.net/blog/1506.html> (Necho format)
- <http://google.com/googleblog> (Ένα από τα δημοφιλέστερα site που χρησιμοποιεί το Atom για την δημιουργία blogs)
- <http://blogspace.com/rss/tools> (Εργαλεία για RSS και Blogs)
- <http://www.redbrick.dcu.ie/~proche/index/projects/MMIS-essay.html> (Εργασία σχετικά με τα Blogs)
- [http://dmoz.org/Reference/Libraries/Library\\_and\\_Information\\_Science/Technical\\_Services/Cataloguing/Metadata/RDF/Applications/RSS/News\\_Readers](http://dmoz.org/Reference/Libraries/Library_and_Information_Science/Technical_Services/Cataloguing/Metadata/RDF/Applications/RSS/News_Readers) (λίστα από RSS Readers)

### Πηγές περιεχομένου

- [www.in.gr](http://www.in.gr)
- [www.yahoo.gr](http://www.yahoo.gr)
- [www.flash.gr](http://www.flash.gr)
- [www.wikipedia.org](http://www.wikipedia.org)
- [www.oreillynet.com](http://www.oreillynet.com)
- [www.timewarner.com](http://www.timewarner.com)
- [www.pathfinder.gr](http://www.pathfinder.gr)

- [www.naftemporiki.gr](http://www.naftemporiki.gr)
- [www.ert.gr](http://www.ert.gr)
- [www.atosorigin.com](http://www.atosorigin.com)
- [www.bbc.co.uk](http://www.bbc.co.uk)
- [www.reuters.de](http://www.reuters.de)
- <http://eclass.gunet.gr> (Εφαρμογή διαχείρισης περιεχομένου ελληνικών Πανεπιστημίων)

### ***Web Services***

- [www.webservices.org](http://www.webservices.org)
- [www.webservicesarchitect.com](http://www.webservicesarchitect.com)
- [www.xml.com](http://www.xml.com)
- <http://msdn.microsoft.com/webservices>
- <http://msdn.microsoft.com/soap>
- <http://www.xmlwebservices.cc>
- [www.w3.org](http://www.w3.org)
- [www.oasis-open.org](http://www.oasis-open.org)
- [www.sun.com](http://www.sun.com)
- [www.go-online.gr](http://www.go-online.gr)
- [www.xignite.com](http://www.xignite.com)
- [uddi.microsoft.com](http://uddi.microsoft.com) και [test.uddi.microsoft.com](http://test.uddi.microsoft.com)
- [www-3.ibm.com/services/uddi/protect/registry.html](http://www-3.ibm.com/services/uddi/protect/registry.html) και
- [www-3.ibm.com/services/uddi/testregistry/protect/registry.html](http://www-3.ibm.com/services/uddi/testregistry/protect/registry.html)
- [www.uddi.org](http://www.uddi.org)
- [www.xmethods.net](http://www.xmethods.net)
- <http://developer.apple.com/internet/webservices/soapphp.html>
- [www.thewebserviceoftheday.com](http://www.thewebserviceoftheday.com)
- [http://www.xmlspy.com/download\\_components.html](http://www.xmlspy.com/download_components.html)

### ***Πατέντες και άδειες λογισμικού***

- <http://epatents.hellug.gr/>
- [http://bladeenc.mp3.no/articles/software\\_patents.html](http://bladeenc.mp3.no/articles/software_patents.html)
- <http://gnu.org>
- <http://www.open-source.gr>