



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

**Σχεδιασμός και Υλοποίηση ενός Συστήματος Βασισμένου σε
Πρότυπα για τη Δυναμική Παρουσίαση Μαθησιακού
Περιεχομένου**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΜΙΧΑΛΗΣ ΠΑΠΑΔΟΠΟΥΛΟΣ

Επιβλέπων : ΝΙΚΟΛΑΟΣ ΠΑΠΑΣΠΥΡΟΥ
Λέκτορας Ε.Μ.Π.

Αθήνα, Ιούλιος 2005



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**Σχεδιασμός και Υλοποίηση ενός Συστήματος Βασισμένου σε
Πρότυπα για τη Δυναμική Παρουσίαση Μαθησιακού
Περιεχομένου**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΜΙΧΑΛΗΣ ΠΑΠΑΔΟΠΟΥΛΟΣ

Επιβλέπων : ΝΙΚΟΛΑΟΣ ΠΑΠΑΣΠΥΡΟΥ

Λέκτορας Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11^η Ιουλίου 2005.

.....
Νικόλαος Παπασπύρου
Λέκτορας Ε.Μ.Π.

.....
Ανδρέας Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

.....
Τιμολέον Σελλής
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2005

.....
ΜΙΧΑΛΗΣ ΠΑΠΑΔΟΠΟΥΛΟΣ

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

© 2005 – Michael Papadopoulos, 2005. All rights reserved

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Πρόλογος

Η παρούσα διπλωματική εργασία εκπονήθηκε στο Εργαστήριο Τεχνολογίας Λογισμικού (Softlab) του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Υπεύθυνος καθηγητής είναι ο κ. Νίκος Παπασπύρου τον οποίο ευχαριστώ για την εμπιστοσύνη που μου έδειξε κατά τη διάρκεια της εκπόνησης της παρούσας εργασίας. Αισθάνομαι τυχερός που γνώρισα και συνεργάστηκα με τον επιστημονικό συνεργάτη του Εργαστηρίου κ. Ανδρέα Παπασαλούρο, ο οποίος ήταν ο επιβλέπων της παρούσας διπλωματικής, και τον οποίο ευχαριστώ για την ουσιαστική βοήθεια και καθοδήγηση που μου παρείχε τόσο κατά τη διάρκεια ανάπτυξης του λογισμικού συστήματος που υλοποίησα όσο και κατά τη συγγραφή της παρούσας αναφοράς.

Ιούλιος 2005

Παπαδόπουλος Μιχάλης

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός web-based περιβάλλοντος παρουσίασης μαθημάτων (courses) με τη μορφή κατάλληλα δομημένου μαθησιακού υλικού (instructional content). Το σύστημα συμμορφώνεται με τις προδιαγραφές IMS Content Packaging και IMS Simple Sequencing που έχει δημιουργήσει ο οργανισμός IMS Global Learning Consortium για τα Συστήματα Μαθησιακής Τεχνολογίας. Η προδιαγραφή IMS Content Packaging αφορά στον ορισμό ενός πρότυπου τρόπου συσκευασίας (packaging) του μαθησιακού υλικού έτσι ώστε να είναι δυνατή η ανταλλαγή και επαναχρησιμοποίησή του μεταξύ διαφορετικών συστημάτων. Επίσης ορίζει έναν τρόπο οργάνωσης του μαθησιακού υλικού σε δενδρικές ιεραρχίες οι οποίες εκφράζονται με χρήση της γλώσσας XML. Η προδιαγραφή IMS Simple Sequencing αφορά στον ορισμό ενός μοντέλου δεδομένων και αλγορίθμων που δρουν σε αυτά, με τα οποία μπορεί κάποιος να ορίζει τη δυναμική σειρά παρουσίασης (sequencing) του μαθησιακού υλικού. Ο χαρακτηρισμός «δυναμική» σημαίνει ότι οι αποφάσεις με τις οποίες σχηματίζεται αυτή η σειρά λαμβάνονται σε χρόνο εκτέλεσης (run-time) σύμφωνα με κάποιους κανόνες, οι οποίοι αναπαριστώνται επίσης με χρήση της γλώσσας XML. Το σύστημα αναπτύχθηκε με χρήση τεχνολογιών δικτυακού προγραμματισμού της πλατφόρμας της JAVA και συγκεκριμένα με τις τεχνολογίες JSP και JSTL.

Λέξεις Κλειδιά: Ηλεκτρονική Μάθηση, Web-based learning, Συστήματα Μαθησιακής Τεχνολογίας, Συστήματα Διαχείρισης Μάθησης, Διαχείριση Μαθησιακού Υλικού, Προδιαγραφές Μαθησιακής Τεχνολογίας, IMS Content Packaging, IMS Simple Sequencing.

Abstract

The objective of this Diploma Thesis is the design and implementation of a web-based environment for the delivery of courses in the form of appropriately structured instructional content. The system conforms to the IMS Content Packaging and IMS Simple Sequencing specifications that have been created by the IMS Global Learning Consortium for Learning Technology Systems. The IMS Content Packaging specification deals with the definition of a standardized method of packaging instructional content in order to enable its exchange and reuse across different systems. Furthermore, it defines a way to organize instructional content in tree hierarchies that are represented using the language XML. The IMS Simple Sequencing specification deals with the definition of a data model and of algorithms that act on it, with which one can define the dynamic presentation order (sequencing) of instructional content. The characterization "dynamic" means that the decisions with which this order is formed are taken in run-time according to rules that are also represented in XML. The system has been developed using web-programming technologies of the JAVA platform and in particular with the JSP and JSTL technologies.

Keywords: eLearning, Web-based learning, Learning Technology Systems, Learning Management Systems, Learning Content Management, Learning Technology Specifications, IMS Content Packaging, IMS Simple Sequencing.

Πίνακας Περιεχομένων

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....	9
1 ΕΙΣΑΓΩΓΗ	11
1.1 E-LEARNING ΚΑΙ LEARNING MANAGEMENT SYSTEMS.....	11
1.2 ΑΝΑΓΚΑΙΟΤΗΤΑ ΤΩΝ ΠΡΟΔΙΑΓΡΑΦΩΝ	13
1.3 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΕΡΓΑΣΙΑΣ	14
2 ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ ΥΠΟΣΤΗΡΙΖΟΜΕΝΩΝ ΠΡΟΔΙΑΓΡΑΦΩΝ.....	15
2.1 IMS CONTENT PACKAGING SPECIFICATION.....	15
2.1.1 <i>Content Packages</i>	17
2.1.2 <i>Δομή του IMS Manifest</i>	19
2.2 IMS SIMPLE SEQUENCING SPECIFICATION	23
2.2.1 <i>Learning Activity</i>	24
2.2.2 <i>Activity Clusters</i>	26
2.2.3 <i>Sequencing Control Modes</i>	26
2.2.4 <i>Objectives</i>	27
2.2.5 <i>Tracking Model</i>	27
2.2.6 <i>Activity State Model</i>	28
2.2.7 <i>Sequencing Rules</i>	29
2.2.8 <i>Rollups</i>	31
2.2.9 <i>Overall Sequencing Process</i>	32
2.3 ΣΥΝΑΦΕΙΣ ΠΡΟΔΙΑΓΡΑΦΕΣ.....	36
2.3.1 <i>IMS Learning Resource Meta-Data Specification</i>	36
2.3.2 <i>IMS Learner Information Package Specification</i>	37
2.3.3 <i>IMS Question & Test Interoperability Specification</i>	38
3 ΤΕΚΜΗΡΙΩΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	43
3.1 ΑΝΑΛΥΣΗ	43
3.1.1 <i>Ανάλυση Απαιτήσεων (Requirements Analysis)</i>	43
3.1.2 <i>Domain (Conceptual) Model</i>	44
3.2 ΣΧΕΔΙΑΣΜΟΣ	45
3.2.1 <i>Γενικά</i>	45
3.2.2 <i>Χρηστική όψη</i>	45
3.2.3 <i>Αρχιτεκτονική Σχεδίαση</i>	51
3.3 ΥΛΟΠΟΙΗΣΗ	61
3.3.1 <i>Λεπτομέρειες Υλοποίησης</i>	61
3.3.2 <i>System Requirements</i>	62
3.3.3 <i>Λεπτομερής Σχεδίαση</i>	62
4 CASE STUDY.....	77
4.1 ΣΧΕΔΙΑΣΜΟΣ ΤΗΣ ΜΑΘΗΣΙΑΚΗΣ ΕΜΠΕΙΡΙΑΣ.....	77
4.2 ΥΛΟΠΟΙΗΣΗ	81
5 ΕΠΙΛΟΓΟΣ.....	88
5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ	88
5.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	89
ΠΑΡΑΡΤΗΜΑ Α	90
ΠΑΡΑΡΤΗΜΑ Β.....	93
B.1 JAVA 2 PLATFORM, ENTERPRISE EDITION (J2EE).....	93
B.2 XML	101

B.3 XMLBEANS	103
ΠΑΡΑΡΤΗΜΑ Γ.....	106
Γ.1 IMS CONTENT PACKAGING SPECIFICATION	106
Γ.2 IMS SIMPLE SEQUENCING SPECIFICATION	107
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	109

1 Εισαγωγή

1.1 e-Learning και Learning Management Systems

Το *e-Learning* είναι ένας πολύ γενικός όρος που σημαίνει μάθηση με τη βοήθεια του ηλεκτρονικού υπολογιστή. Αυτό μπορεί να περιλαμβάνει τη χρήση εκπαιδευτικών CD-ROM ή τη χρήση προηγμένων τεχνολογιών των Δικτύων Υπολογιστών, όπως είναι το Internet και το World Wide Web. Στην τελευταία περίπτωση αναφερόμαστε με τον όρο *web based training*¹. Στην περίπτωση αυτή χρησιμοποιούνται συνήθως εξειδικευμένα συστήματα λογισμικού που είναι γνωστά ως *Learning Management Systems (LMSs)* ή *Managed Learning Environments (MLEs)*.

Ένα Learning Management System είναι ένα σύστημα λογισμικού το οποίο επιτρέπει τη διαχείριση μαθημάτων (courses) και αντίστοιχου μαθησιακού υλικού (instructional ή learning content)². Τα συστήματα αυτά είναι κατά κανόνα web-based ώστε να επιτρέπουν την εύκολη πρόσβαση στο μαθησιακό υλικό και στις διαχειριστικές λειτουργίες. Κατ' ελάχιστο, ένα LMS οφείλει να επιτρέπει την παράδοση (delivery) και την παρακολούθηση (tracking) των μαθημάτων και του μαθησιακού υλικού που αντιστοιχεί σε αυτά, και μπορεί επίσης να επιτρέπει τη διαχείριση ομάδων μαθητών (student group management). Σε ένα τυπικό LMS ένας εκπαιδευτικός (instructor) μπορεί να δημιουργεί online μαθήματα (courses), όπου κάθε course είναι ουσιαστικά ένα website το οποίο περιέχει πληροφορίες και εκπαιδευτικό υλικό για κάποιο μάθημα, π.χ. περιγραφή του μαθήματος, agenda, ανακοινώσεις, σημειώσεις του μαθήματος, εργασίες κ.α. Επίσης μπορεί να παρέχονται τρόποι σύγχρονης ή ασύγχρονης επικοινωνίας των μαθητών μεταξύ τους, π.χ. forums, chat, mailing lists. Ένας μαθητής για να αποκτήσει πρόσβαση σε κάποιο course συνήθως πρέπει να ακολουθήσει μια διαδικασία εγγραφής (registration), ενώ σε κάποια μαθήματα μπορεί να υπάρχει ελεύθερη πρόσβαση. Σε πιο προχωρημένα LMSs συναντάμε ακόμη εργαλεία και χαρακτηριστικά όπως competency management, skills-gap analysis,

¹ Υπάρχει αρκετή αυθαιρεσία και ασάφεια στον τρόπο που χρησιμοποιούνται οι διάφοροι όροι στο eLearning. Κάποιοι ορίζουν ως eLearning αυτό που εμείς εδώ ονομάζουμε web based training. Για παράδειγμα, στο [GOODYEAR 00] προτείνεται ο ακόλουθος ορισμός : *eLearning is the systematic use of networked multimedia computer technologies to empower learners, improve learning, connect learners to people and resources supportive of their needs, and to integrate learning with performance and individual with organisational goals.*

² Το IMS Global Learning Consortium ορίζει το Learning Management System ως «*a computer application that enables learning management*», όπου το Learning Management ορίζεται ως εξής : *Learning Management refers to the processes of administering, tracking, scheduling, and reporting activities associated with a learner interacting with (online) instructional content.* [AF GLOSSARY 03]

succession planning, certifications, virtual live classes, content management, content authoring κ.α.

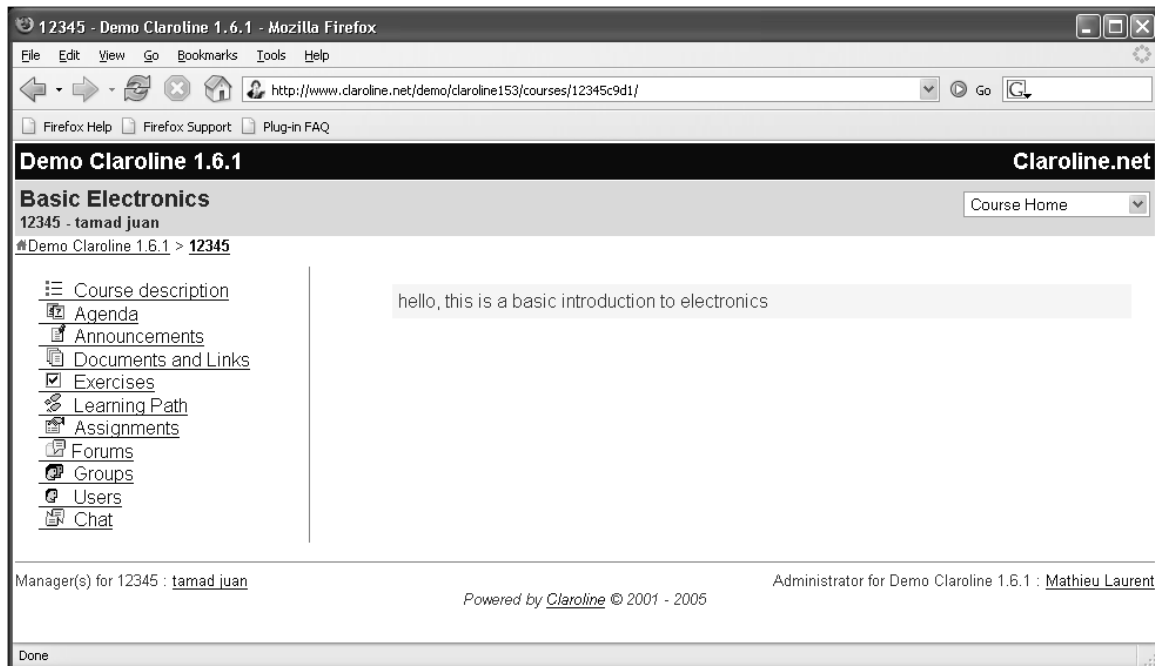
Ο καλύτερος τρόπος για να καταλάβει κανείς τί ακριβώς κάνει ένα LMS είναι να χρησιμοποιήσει κάποιο. Στο σχήμα 1.1 φαίνεται το Moodle, το οποίο είναι σήμερα ένα από τα πιο γνωστά open source LMSs. Το συγκεκριμένο screenshot είναι από το Moodle που έχει στηθεί στο Εργαστήριο Τεχνολογίας Λογισμικού (Softlab) της σχολής των ΗΜΜΥ του ΕΜΠ για την υποστήριξη των μαθημάτων που διδάσκονται από τα μέλη του εργαστηρίου. Στο σχήμα 1.2 φαίνεται το Claroline, ένα επίσης πολύ γνωστό open source LMS που αναπτύχθηκε στο Catholic University of Louvain (Βέλγιο) και χρησιμοποιείται από εκατοντάδες οργανισμούς παγκοσμίως, κυρίως πανεπιστήμια.

The screenshot displays the Moodle LMS interface for a course titled "Τεχνολογία Λογισμικού". The browser address bar shows the URL: <http://moodle.softlab.ntua.gr/course/view.php?id=3>. The user is logged in as "Μικάλης Παπαδόπουλος (Εξοδος)".

The interface is organized into several sections:

- Διαχείριση (Management):** Includes options for "Βαθμοί...", "Αλλαγή του κωδικού πρόσβασης...", and "softeng - ακύρωση της εγγραφής μου...".
- Μαθήματα (Courses):** Lists the current course "Τεχνολογία Λογισμικού" with a link to "Όλα τα μαθήματα...".
- Αναζήτηση (Search):** A search box with the button "Αναζήτηση στις ομάδες συζητήσεων".
- Συνδεδεμένοι Χρήστες (Logged-in Users):** Shows "Μικάλης Παπαδόπουλος" as the last user active 5 minutes ago.
- Περιγραφή θέματος (Course Description):** The main content area displays the course title "Τεχνολογία Λογισμικού", the course type "Υποχρεωτικό Μάθημα Ροής Α (κύρια ροή)", and the semester "9ου Εξαμήνου, Κωδικός 3.4.56.9".
- 1 Πληροφορίες για το μάθημα (Course Information):** Includes "Ανακοινώσεις" and "Ενημερωτικό φυλλάδιο".
- 2 Υλικό του μαθήματος (Course Materials):** Lists resources such as "Ηλεκτρονικές Σημειώσεις", "Διαφάνειες και διαλέξεις", "Συζήτηση για την ύλη", "Ερωτηματολόγιο προ-αξιολόγησης", "Χρήσιμες Μαθησιακές Πηγές", and "Ερωτηματολόγιο μετα-ελέγχου".
- 3 Εργαστηριακά έργα (Laboratory Works):** A section for practical assignments.
- Ημερολόγιο (Calendar):** Shows the month of June 2005 with a calendar grid. The 29th is highlighted.
- Επικείμενα γεγονότα (Upcoming Events):** A section for upcoming events, currently empty.
- Δραστηριότητες (Activities):** Lists activities like "Ομάδες συζήτησης" and "Πηγές πληροφοριών".
- Γίτομα (Tools):** Includes "Συμμετέχοντες", "Groups", and "Επεξεργασία του προφίλ".

Σχήμα 1.1 Η κεντρική οθόνη ενός μαθήματος στο σύστημα Moodle



Σχήμα 1.2 Το σύστημα Claroline

Τα LMSs που είναι διαθέσιμα σήμερα είναι είτε εμπορικά συστήματα (π.χ. Blackboard, WebCT, Brightspark, eCollege, Desire2Learn, ANGEL Learning), είτε freeware και open source συστήματα (π.χ. ILIAS, Manhattan Virtual Classroom, OLAT, Atutor, Moodle, Claroline, Dokeos), είτε εξειδικευμένα συστήματα που εξυπηρετούν τις ανάγκες ενός συγκεκριμένου οργανισμού.

Μια συστηματική κατηγοριοποίηση των LMSs και των χαρακτηριστικών τους έχει προταθεί στο [APR 01] και συνοψίζεται στο Παράρτημα Α. Επίσης στο [AVG 03] έχει προταθεί ένα business model και μια αρχιτεκτονική αναφοράς καθώς και ένα σύνολο από design patterns για LMSs.

1.2 Αναγκαιότητα των προδιαγραφών

Ένα σοβαρό πρόβλημα των LMSs που είναι διαθέσιμα σήμερα είναι ότι συχνά δεν υπάρχει εύκολος τρόπος μεταφοράς ενός μαθήματος από ένα σύστημα σε ένα άλλο, γιατί το κάθε σύστημα έχει ένα δικό του τρόπο αναπαράστασης της πληροφορίας που αφορά το μάθημα, με αποτέλεσμα ο κόπος που καταβλήθηκε για τη δημιουργία ενός course για ένα συγκεκριμένο LMS να πηγαίνει χαμένος αν ο οργανισμός αλλάξει το LMS που χρησιμοποιεί, ή αν οργανισμοί που χρησιμοποιούν διαφορετικά LMSs επιθυμούν να συνεργαστούν. Υπάρχει λοιπόν ανάγκη να οριστούν standards για το μαθησιακό υλικό και αυτά να υιοθετηθούν από τους κατασκευαστές των LMSs.

Το ρόλο αυτό της δημιουργίας προδιαγραφών για μαθησιακή τεχνολογία (eLearning technology specifications) έχουν αναλάβει κάποιοι οργανισμοί με κυριότερους την Advanced Distributed Learning (ADL, <http://www.adlnet.org>) και το IMS Global Learning Consortium (<http://www.imsglobal.org>).

1.3 Αντικείμενο της εργασίας

Αντικείμενο της παρούσας εργασίας είναι η ανάπτυξη ενός περιβάλλοντος δυναμικής παρουσίασης μαθημάτων (μπορεί να θεωρηθεί ως ένα υποτυπώδες LMS) το οποίο θα ακολουθεί δύο σημαντικές προδιαγραφές που έχει εκδώσει το IMS Global Learning Consortium, και συγκεκριμένα το IMS Content Packaging Specification και το IMS Simple Sequencing Specification. Οι προδιαγραφές αυτές παρουσιάζονται στο κεφάλαιο 2.

Λέμε ότι το σύστημα είναι ένα *υποτυπώδες* LMS, γιατί σε σχέση με τα συνήθη χαρακτηριστικά των LMSs που αναφέρθηκαν στην παράγραφο 1.1, το σύστημα μας επικεντρώνεται μόνο στη διαχείριση και παρουσίαση του μαθησιακού υλικού. Όμως μια βασική διαφοροποίηση σε σχέση με τα υπάρχοντα συστήματα είναι η υποστήριξη της *δυναμικής* παρουσίασης του μαθησιακού υλικού ανάλογα με το ιστορικό της αλληλεπίδρασης του μαθητή με το σύστημα, όπως περιγράφεται αναλυτικά στο κεφάλαιο 2. Το χαρακτηριστικό αυτό αποτελεί σημαντική πρωτοτυπία, γιατί στα υπάρχοντα LMSs το παρεχόμενο μαθησιακό υλικό είναι *στατικό*, δηλαδή δεν υπάρχει αλληλεπίδραση (interaction) του μαθητή με αυτό, και συνήθως πρόκειται απλά για έγγραφα που ο μαθητής μπορεί να κατεβάζει (download) στον υπολογιστή του.

Η τεκμηρίωση του συστήματος που αναπτύχθηκε περιλαμβάνεται στο κεφάλαιο 3. Στο κεφάλαιο 4 παρουσιάζουμε ένα case study που δείχνει πώς το σύστημα μπορεί να χρησιμοποιηθεί στην πράξη και τέλος το κεφάλαιο 5 περιέχει τα συμπεράσματά μας και κάποιες προτάσεις για μελλοντική εργασία στη συγκεκριμένη περιοχή.

2 Παρουσίαση των υποστηριζόμενων προδιαγραφών

Στο κεφάλαιο αυτό παρουσιάζονται οι προδιαγραφές που υλοποιήθηκαν. Η παρουσίαση είναι σκόπιμα συνοπτική και περιληπτική μιας και ο στόχος είναι να αποκτήσει ο αναγνώστης μια γενική εικόνα του περιεχομένου των προδιαγραφών. Για περισσότερες λεπτομέρειες μπορεί κανείς να ανατρέξει στα έγγραφα των προδιαγραφών (βλ. βιβλιογραφία).

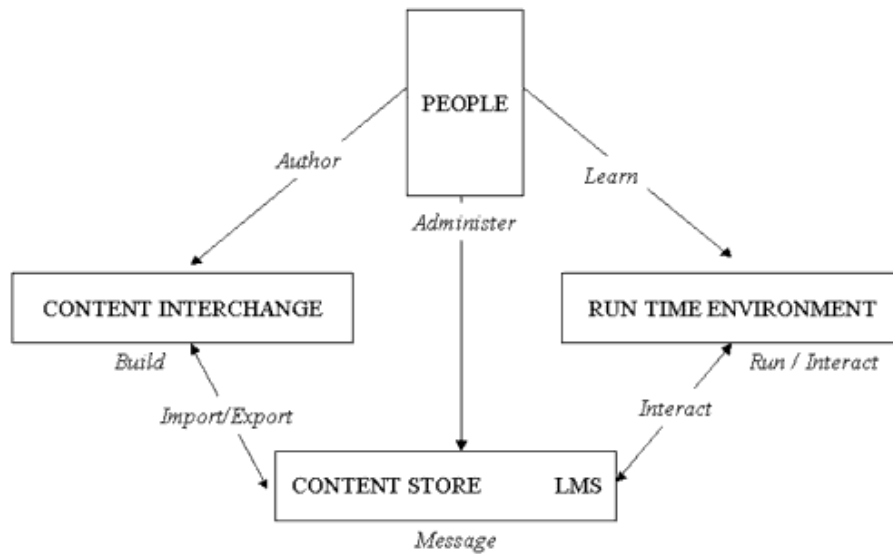
2.1 IMS Content Packaging Specification

Το *μαθησιακό υλικό* (learning/instructional content) συχνά χρειάζεται να συγκεντρωθεί και να συσκευαστεί σε κάποια ψηφιακή μορφή ώστε να γίνει δυνατή η αποδοτική διανομή, διαχείρισή και παράταξη (deployment) του. Οι δημιουργοί του μαθησιακού υλικού θέλουν να έχουν διαθέσιμα εργαλεία και τεχνολογίες που διευκολύνουν την δημιουργία του υλικού. Οι κατασκευαστές (vendors) του e-Learning λογισμικού θέλουν να κατασκευάζουν εργαλεία που να επιτρέπουν την αποδοτική διανομή και διαχείριση του μαθησιακού υλικού. Τέλος, οι μαθητές ενδιαφέρονται για υψηλής ποιότητας μαθησιακές εμπειρίες που παρέχονται μέσα από υψηλής ποιότητας λογισμικό.

Όταν το μαθησιακό υλικό συσκευάζεται με έναν προτυποποιημένο, κοινώς αποδεκτό τρόπο, οι ανάγκες της online learning κοινότητας μπορούν να ικανοποιηθούν καλύτερα. Η κοινότητα αυτή χρειάζεται κατευθυντήριες γραμμές (guidelines) και προδιαγραφές (specifications) για το μαθησιακό υλικό που θα επιτρέπουν :

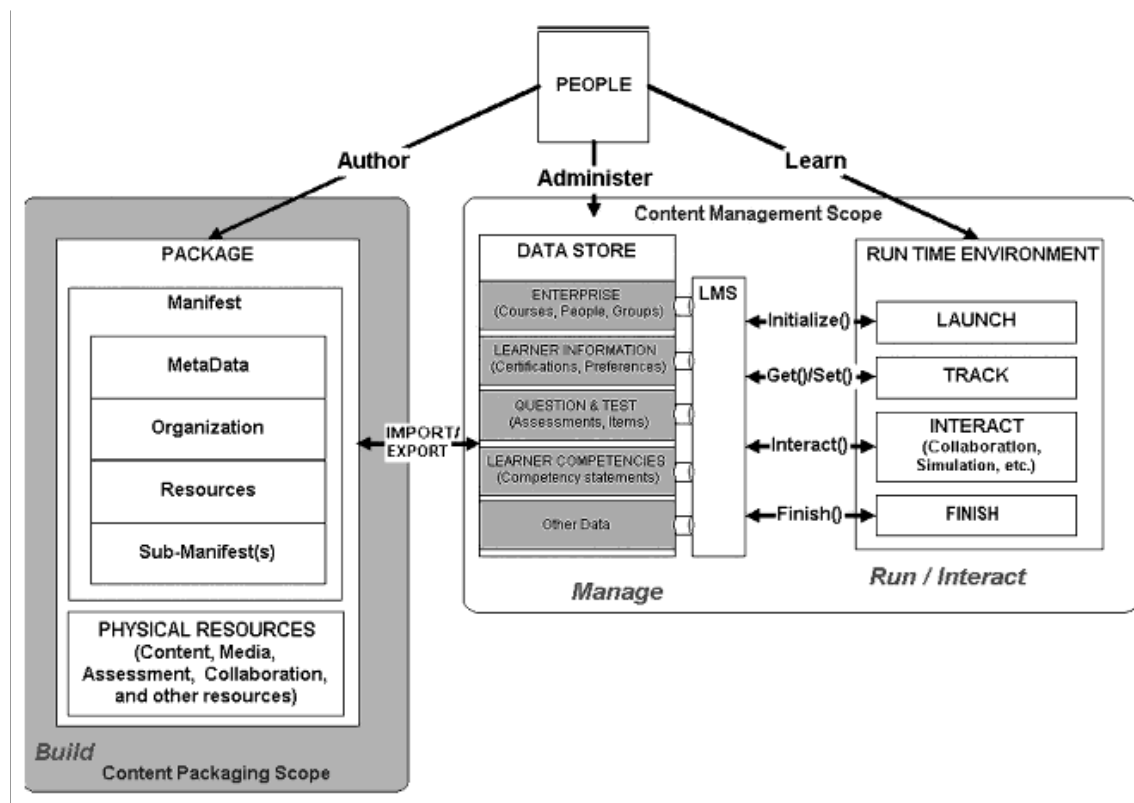
- στους **Δημιουργούς (Authors)** να παράγουν το υλικό,
- στους **Διαχειριστές (Administrators)** να διαχειρίζονται και να διανέμουν το υλικό,
- στους **Μαθητές (Learners)** να αλληλεπιδρούν και να μαθαίνουν από το υλικό.

Το IMS Global Learning Consortium έχει δημιουργήσει ένα πλαίσιο (framework) έχοντας στο μυαλό τους παραπάνω στόχους (βλ. σχήμα 2.1). Το πλαίσιο αυτό ονομάζεται IMS Content framework ([CP, 03c]) .



Σχήμα 2.1 – Στόχοι του IMS Content framework

Το πεδίο εφαρμογής του IMS Content framework φαίνεται στο σχήμα 2.2.



Σχήμα 2.2 – IMS Content framework

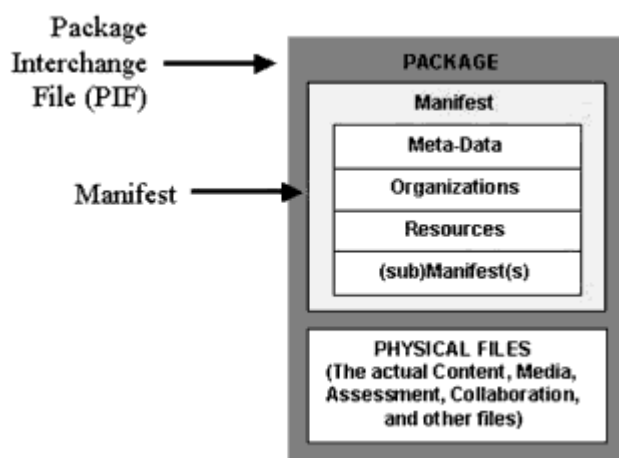
Όπως βλέπουμε, το IMS Content Framework έχει διαιρεθεί σε τρία τμήματα :

- Content Packaging – Το τμήμα που αφορά την συνάθροιση των πόρων του μαθησιακού υλικού (content resource aggregation), δηλαδή το «πακετάρισμα» του υλικού καθώς και την οργάνωση της μαθησιακής εμπειρίας (δηλ. του μαθήματος).
- Data Model – Το τμήμα που αφορά την αποθήκευση και διαχείριση του υλικού.
- Run Time Environment – Το τμήμα που αφορά την αλληλεπίδραση του μαθητή με το υλικό που του παρουσιάζεται.

Το IMS Content Packaging (CP) Specification περιγράφει δομές δεδομένων που χρησιμοποιούνται για την επίτευξη της *διαλειτουργισμότητας* (interoperability) ψηφιακού Internet-based μαθησιακού υλικού (το οποίο στο εξής θα ονομάζουμε *content*) με εργαλεία συγγραφής μαθησιακού υλικού (authoring tools), learning management systems (LMS), και run time περιβάλλοντα. Σκοπός του IMS CP είναι ο ορισμός ενός προτυποποιημένου (standardized) συνόλου από δομές που μπορούν να χρησιμοποιηθούν για την ανταλλαγή του content μεταξύ διαφορετικών συστημάτων.

2.1.1 Content Packages

Το μαθησιακό υλικό οργάνωνται σε μια δομή που ονομάζεται (Content) **Package**. Η δομή αυτή απεικονίζεται στο σχήμα 2.3.



Σχήμα 2.3 Η δομή ενός Package

Το Package όπως φαίνεται στο σχήμα αποτελείται από δύο μέρη : ένα ειδικό XML αρχείο που περιγράφει την οργάνωση του content και τους πόρους (resources - δηλαδή τα φυσικά αρχεία από τα οποία αποτελείται το content), και τα ίδια τα φυσικά αρχεία. Το ειδικό XML αρχείο ονομάζεται **IMS Manifest file**.

Package Interchange File - Όταν το Package έχει συμπιεστεί σε ένα αρχείο (π.χ. .zip, .jar, .cab), τότε ονομάζεται **Package Interchange File (PIF)**. Το IMS Manifest file πρέπει να βρίσκεται στο root του PIF και να ονομάζεται **imsmanifest.xml**. Το PIF είναι ένα μέσο για τη μεταφορά δομημένης πληροφορίας μέσα σε ένα δίκτυο. Ως default format του PIF προτείνεται το PKZip v2.04g (.zip).

Package – ένα νοητό directory, που περιέχει το IMS Manifest file, άλλα XML έγγραφα στα οποία αναφέρεται το IMS Manifest file (όπως π.χ. ένα DTD ή XSD αρχείο), και τα sub-directories όπου βρίσκονται τα resources.

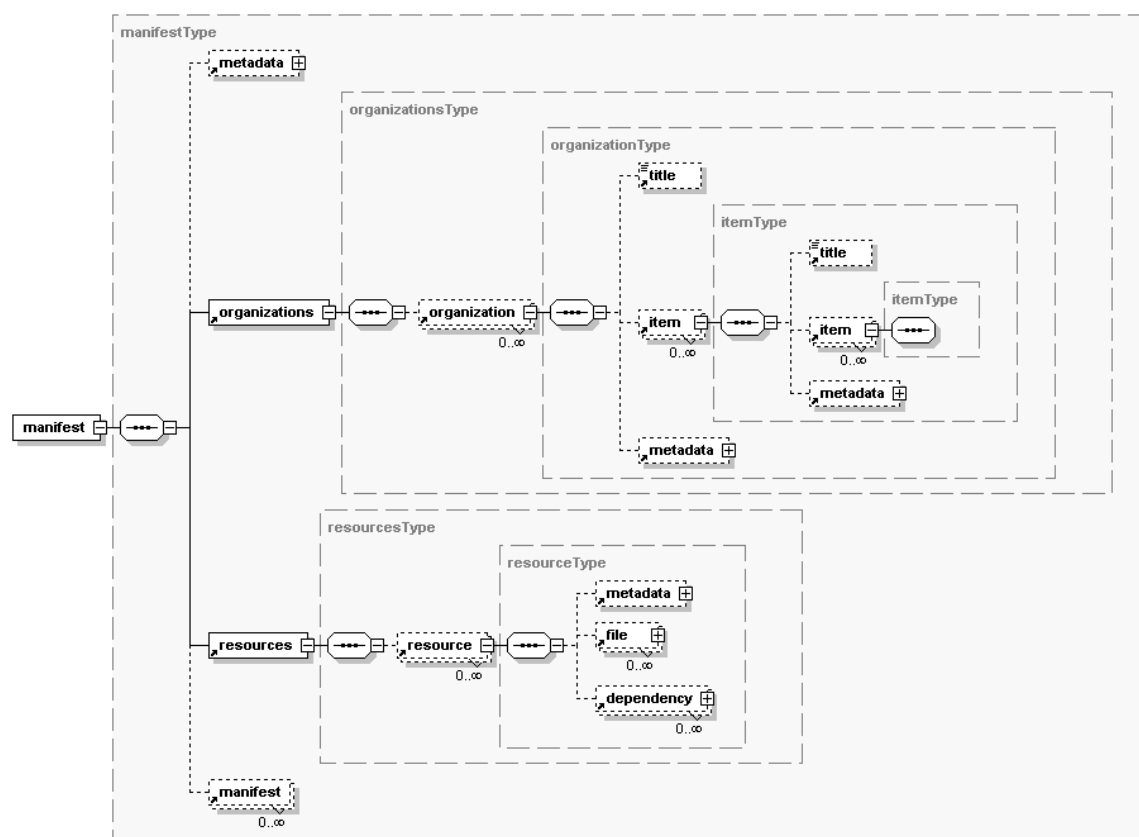
- Top-level Manifest – είναι το root element του IMS Manifest file και περιγράφει τη δομή του ίδιου του Package. Προαιρετικά μπορεί επίσης να περιέχει άλλα (sub)Manifests. Κάθε στιγμιότυπο (instance) ενός manifest περιλαμβάνει τα ακόλουθα τμήματα :
 - Μετα-δεδομένα (Meta-data) – ένα XML element που περιγράφει το manifest.
 - Organizations - ένα XML element που ορίζει μηδέν, μια, ή περισσότερες οργανώσεις (organizations) του content.
 - Resources - ένα XML element που περιέχει αναφορές σε όλους τους πόρους (resources) που χρειάζονται στο manifest, καθώς επίσης και μετα-δεδομένα που περιγράφουν τους πόρους αυτούς.
 - (sub)Manifest – ένα η περισσότερα λογικά εμφωλιασμένα (nested) manifests.
- Physical Files – τα ίδια τα φυσικά αρχεία που αποτελούν το content, όπως π.χ. αρχεία κειμένου, εικόνες, και άλλοι πόροι που βρίσκονται στα διάφορα sub-directories μέσα στο Package.

Το Package αντιπροσωπεύει μια μονάδα επανα-χρησιμοποιήσιμου content. Αυτό μπορεί να είναι ένα αυτοτελές μέρος ενός μαθήματος, ένα μάθημα, ή μια συλλογή από μαθήματα. Ένα Package πρέπει να είναι αυτόνομο (stand-alone), δηλαδή πρέπει να περιέχει όλη την πληροφορία που χρειάζεται για να χρησιμοποιηθούν τα περιεχόμενά του για μάθηση.

Τα Packages δεν είναι απαραίτητο να πακετάρονται σε ένα Package Interchange File. Ένα Package μπορεί να διανέμεται π.χ. σε CD-ROM ή άλλο αποθηκευτικό μέσο χωρίς να έχει συμπεριστεί σε ένα αρχείο. Όμως το IMS Manifest και όλα τα απαιτούμενα XML αρχεία (DTD, XSD) πρέπει να βρίσκονται στο root του μέσου διανομής.

2.1.2 Δομή του IMS Manifest

Η δομή του IMS Manifest απεικονίζεται στο σχήμα 2.4.



Σχήμα 2.4 – Δομή του IMS Manifest

Το root element είναι το <manifest>, το οποίο περιέχει τέσσερα sub-elements : <metadata>, <organizations>, <resources> και (έναν αριθμό από) <manifest>, εκ των οποίων μόνο το <organizations> και το <resources> είναι υποχρεωτικά. Τα elements αυτά περιγράφονται στη συνέχεια :

1. <manifest> : (required) Root element του IMS Manifest (imsmanifest.xml).
2. <metadata> : (optional) Περιέχει μετα-πληροφορία για το IMS Manifest αυτό καθαυτό (π.χ. τίτλος, περιγραφή, λέξεις-κλειδιά, δημιουργοί, εκπαιδευτικοί στόχοι, πνευματικά δικαιώματα κτλ). Τα sub-elements μπορεί να προέρχονται από οποιοδήποτε XML namespace υπό την προϋπόθεση ότι το αντίστοιχο XML Schema δηλώνεται στο IMS Manifest και περιλαμβάνεται στο Package. Συνιστάται να χρησιμοποιούνται τα elements που ορίζει το IMS Meta-Data Specification.
3. <organizations> : Το content που περιέχεται σε ένα Package μπορεί να χρησιμοποιηθεί για τη δημιουργία διαφορετικών μαθησιακών εμπειριών, επιλέγοντας κάθε φορά διαφορετικά υποσύνολα του content και/ή οργανώνοντάς τα σε διαφορετικές δομές. Για παράδειγμα, ας υποθέσουμε ότι διαθέτουμε μαθησιακό υλικό που χωρίζεται σε τρεις ενότητες, με την ενότητα i ($i = 1, 2, 3$) να αποτελείται από τις υποενότητες εισαγωγή i , κύριο μέρος i και σύνοψη i . Ένας τρόπος οργάνωσης του υλικού είναι

- Ενότητα 1
 - i. Εισαγωγή 1
 - ii. Κύριο Μέρος 1
 - iii. Σύνοψη 1
- Ενότητα 2
 - i. Εισαγωγή 2
 - ii. Κύριο Μέρος 2
 - iii. Σύνοψη 2
- Ενότητα 3
 - i. Εισαγωγή 3
 - ii. Κύριο Μέρος 3
 - iii. Σύνοψη 3

Σε κάποια άλλη περίπτωση μπορεί να μας ενδιαφέρει μόνο η σύνοψη κάθε ενότητας, οπότε μπορούμε να οργανώσουμε το υλικό ως εξής :

- Σύνοψη 1
- Σύνοψη 2
- Σύνοψη 3

Έτσι, από το ίδιο content μπορούν να προκύψουν διαφορετικές μαθησιακές εμπειρίες, ορίζοντας διαφορετικές όψεις ή - στην ορολογία του Content Packaging - *οργανώσεις* του υλικού. Οι οργανώσεις αυτές ορίζονται στο **<organizations>** element του IMS Manifest. Το **<organizations>** element μπορεί να περιέχει οποιοδήποτε αριθμό από **<organization>** sub-elements, όπου κάθε **<organization>** ορίζει μια συγκεκριμένη οργάνωση του υλικού. Η οργάνωση έχει δενδρική δομή, αν και σε μελλοντικές εκδόσεις του specification ενδέχεται να οριστούν και άλλες δομές (π.χ. κατευθυνόμενος γράφος). Κάθε **<organization>** μπορεί να περιέχει οποιοδήποτε αριθμό από **<item>** elements, όπου κάθε **<item>** είναι ένας κόμβος του δέντρου και μπορεί με τη σειρά του να περιέχει άλλα **<item>**s. Στο προηγούμενο παράδειγμα ένα item θα ήταν π.χ. η Ενότητα 2, και υπο-items αυτού θα ήταν τα items Εισαγωγή 2, Κύριο Μέρος 2 και Σύνοψη 2. Κάθε **<item>** χαρακτηρίζεται από ένα μοναδικό **identifier**, ενώ μπορεί να αντιστοιχίζεται σε κάποιο πόρο (resource) μέσω ενός **identifierref** attribute. Προαιρετικά μπορεί επίσης να περιέχει έναν τίτλο (**<title>**) και **<metadata>**.

Αν ορίζονται περισσότερα του ενός **<organization>**s, τότε το default δίνεται από την τιμή του **default** attribute του **<organizations>** element. Αν το default attribute έχει παραληφθεί τότε ως default θεωρείται το πρώτο **<organization>**.

Ένα παράδειγμα <organization> φαίνεται παρακάτω :

```
<organization identifier="TOC1">
  <title>Default Organization</title>
  <item identifier="ITEM1" identifierref="RESOURCE1">
    <title>Lesson 1</title>
  </item>
  <item identifier="ITEM2" identifierref="RESOURCE2">
    <title>Lesson 2</title>
  </item>
  <item identifier="ITEM3" identifierref="RESOURCE3">
    <title>Lesson 3</title>
  </item>
</organization>
```

Το LMS μπορεί να παρουσιάσει το <organization> αυτό στο μαθητή ως εξής :

- Lesson 1
 - Lesson 2
 - Lesson 3
4. <resources> : Το <resources> element ορίζει μια συλλογή από αρχεία τα οποία αποτελούν τους πόρους του content. Κάθε πόρος δηλώνεται σε ένα <resource> element μέσα στο σώμα του <resources> element. Ένα <resource> δεν είναι απαραίτητα ένα μεμονωμένο αρχείο. Μπορεί να είναι μια συλλογή από σχετιζόμενα αρχεία που όλα μαζί αποτελούν την παρουσίαση ενός <item> element. Ένα αρχείο δηλώνεται σε ένα <file> element και μπορεί να είναι εσωτερικό στο Package ή εξωτερικό (με αναφορά σε αυτό μέσω ενός URL). Τα εσωτερικά αρχεία πρέπει να συμπεριλαμβάνονται στο Package Interchange File.
5. sub-<manifest> : Ένα <manifest> μπορεί να περιέχει οποιοδήποτε αριθμό από sub-<manifest>s. Με τον τρόπο αυτό διαφορετικά Packages μπορούν να συναθροίζονται (aggregation) σε ένα σύνθετο Package, ή αντίστροφα ένα σύνθετο Package να αποσυντίθεται (disaggregation) σε απλούστερα Packages.

Ένα πλήρες παράδειγμα ενός IMS Manifest αρχείου φαίνεται παρακάτω :

```
<?xml version="1.0"?>
<manifest identifier="MANIFEST1" xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
xmlns:imsmd="http://www.imsglobal.org/xsd/imsmd_v1p2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2.xsd">
  <metadata>
    <schema>IMS Content</schema>
    <schemaversion>1.1.3</schemaversion>
    <imsmd:lom>
      <imsmd:general>
        <imsmd:title>
          <imsmd:langstring xml:lang="en-US">IMS Content Packaging Sample - Multiple
Organizations</imsmd:langstring>
        </imsmd:title>
      </imsmd:general>
    </imsmd:lom>
  </metadata>
```

```

<organizations default="TOC1">
  <organization identifier="TOC1" structure="hierarchical">
    <title>All Lessons</title>
    <item identifier="ITEM1" identifierref="RESOURCE1">
      <title>Lesson 1</title>
      <item identifier="ITEM2" identifierref="RESOURCE2">
        <title>Introduction 1</title>
      </item>
      <item identifier="ITEM3" identifierref="RESOURCE3">
        <title>Content 1</title>
      </item>
      <item identifier="ITEM4" identifierref="RESOURCE4">
        <title>Summary 1</title>
      </item>
    </item>
    <item identifier="ITEM5" identifierref="RESOURCE5">
      <title>Lesson 2</title>
      <item identifier="ITEM6" identifierref="RESOURCE6">
        <title>Introduction 2</title>
      </item>
      <item identifier="ITEM7" identifierref="RESOURCE7">
        <title>Content 2</title>
      </item>
      <item identifier="ITEM8" identifierref="RESOURCE8">
        <title>Summary 2</title>
      </item>
    </item>
    <item identifier="ITEM9" identifierref="RESOURCE9">
      <title>Lesson 3</title>
      <item identifier="ITEM10" identifierref="RESOURCE10">
        <title>Introduction 3</title>
      </item>
      <item identifier="ITEM11" identifierref="RESOURCE11">
        <title>Content 3</title>
      </item>
      <item identifier="ITEM12" identifierref="RESOURCE12">
        <title>Summary 3</title>
      </item>
    </item>
  </organization>
  <organization identifier="TOC2" structure="hierarchical">
    <title>Content Topics</title>
    <item identifier="TOC2_ITEM3" identifierref="RESOURCE3">
      <title>Content 1</title>
    </item>
    <item identifier="TOC2_ITEM7" identifierref="RESOURCE7">
      <title>Content 2</title>
    </item>
    <item identifier="TOC2_ITEM11" identifierref="RESOURCE11">
      <title>Content 3</title>
    </item>
  </organization>
  <organization identifier="TOC3" structure="hierarchical">
    <title>Summary Topics</title>
    <item identifier="TOC3_ITEM4" identifierref="RESOURCE4">
      <title>Summary 1</title>
    </item>
    <item identifier="TOC3_ITEM8" identifierref="RESOURCE8">
      <title>Summary 2</title>
    </item>
    <item identifier="TOC3_ITEM12" identifierref="RESOURCE12">
      <title>Summary 3</title>
    </item>
  </organization>
</organizations>
<resources>
  <resource identifier="RESOURCE1" type="webcontent" href="lesson1.htm">
    <file href="lesson1.htm"/>
  </resource>
  <resource identifier="RESOURCE2" type="webcontent" href="intro1.htm">
    <file href="intro1.htm"/>
  </resource>

```

```

</resource>
<resource identifier="RESOURCE3" type="webcontent" href="content1.htm">
  <file href="content1.htm"/>
</resource>
<resource identifier="RESOURCE4" type="webcontent" href="summary1.htm">
  <file href="summary1.htm"/>
</resource>
<resource identifier="RESOURCE5" type="webcontent" href="lesson2.htm">
  <file href="lesson2.htm"/>
</resource>
<resource identifier="RESOURCE6" type="webcontent" href="intro2.htm">
  <file href="intro2.htm"/>
</resource>
<resource identifier="RESOURCE7" type="webcontent" href="content2.htm">
  <file href="content2.htm"/>
</resource>
<resource identifier="RESOURCE8" type="webcontent" href="summary2.htm">
  <file href="summary2.htm"/>
</resource>
<resource identifier="RESOURCE9" type="webcontent" href="lesson3.htm">
  <file href="lesson3.htm"/>
</resource>
<resource identifier="RESOURCE10" type="webcontent" href="intro3.htm">
  <file href="intro3.htm"/>
</resource>
<resource identifier="RESOURCE11" type="webcontent" href="content3.htm">
  <file href="content3.htm"/>
</resource>
<resource identifier="RESOURCE12" type="webcontent" href="summary3.htm">
  <file href="summary3.htm"/>
</resource>
</resources>
</manifest>

```

2.2 IMS Simple Sequencing Specification

Όπως είδαμε, το Content Packaging προδιαγράφει τον τρόπο συσκευασίας του μαθησιακού υλικού και επιπλέον δίνει τη δυνατότητα οργάνωσης του υλικού σε μια δενδρική δομή (ή πολλές εναλλακτικές). Δεν μας λέει όμως τίποτα για το πώς το υλικό αυτό πρέπει να παρουσιαστεί στο μαθητή. Προφανώς μια δυνατότητα είναι να αφήσουμε το μαθητή να επιλέγει αυθαίρετα οποιαδήποτε μονάδα επιθυμεί. Γενικά όμως μας ενδιαφέρει να υπάρχει κάποια επίβλεψη/καθοδήγηση επιβάλλοντας περιορισμούς στον τρόπο που ο μαθητής αλληλεπιδρά με το υλικό. Για παράδειγμα, αν η β' ενότητα ενός μαθήματος έχει ως προαπαιτούμενο την επιτυχή παρακολούθηση της α' ενότητας, θέλουμε να απαγορεύεται στο μαθητή η πρόσβαση στη β' ενότητα αν δεν καλύπτει το προαπαιτούμενο αυτό.

Η προδιαγραφή IMS Simple Sequencing ορίζει μια μέθοδο για την αναπαράσταση της επιδιωκόμενης συμπεριφοράς μια *μαθησιακής εμπειρίας (learning experience)*, έτσι ώστε κάθε LTS (*learning technology system*) να μπορεί να διατάσσει (sequence) τις επιμέρους δραστηριότητες που αποτελούν την μαθησιακή εμπειρία σε ακολουθίες, με συνεπή τρόπο.

Με πιο απλά λόγια, ο δημιουργός του μαθησιακού υλικού μπορεί να καθορίζει τη σειρά με την οποία πρέπει να παρουσιαστούν στο μαθητή τα περιεχόμενα του μαθησιακού υλικού. Η σειρά αυτή υπολογίζεται δυναμικά σύμφωνα με κανόνες που ορίζουν πότε (υπό ποιές συνθήκες) ένα τμήμα του περιεχομένου παρουσιάζεται ή

παραλείπεται. Οι συνθήκες που εμφανίζονται στους κανόνες αυτούς έχουν να κάνουν με τα αποτελέσματα της αλληλεπίδρασης του μαθητή με το υλικό.

Το επίθετο “simple” δεν σημαίνει ότι το Simple Sequencing είναι απλό. Αναφέρεται στο γεγονός ότι το Simple Sequencing καλύπτει μόνο ένα υποσύνολο των ευρέως χρησιμοποιούμενων μεθόδων sequencing, π.χ. δεν καλύπτει το artificial intelligence-based sequencing, collaborative learning κ.α.

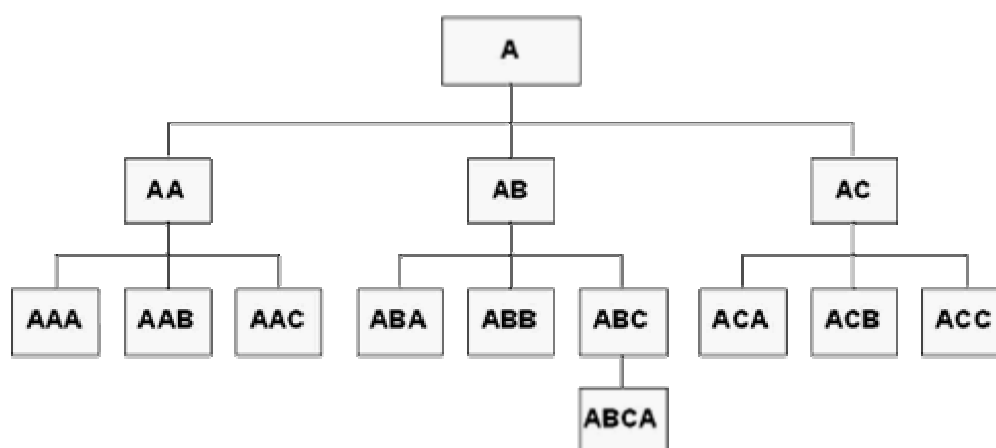
Το Simple Sequencing είναι ανεξάρτητο της φύσης του μαθησιακού υλικού και του τρόπου επικοινωνίας του μαθητή με το LTS. Μια συνήθης περίπτωση είναι το υλικό να παρουσιάζεται μέσω κάποιου browser, όμως το εύρος εφαρμογής του Simple Sequencing δεν περιορίζεται σε web-based περιβάλλοντα.

2.2.1 Learning Activity

Κεντρική έννοια στο Simple Sequencing είναι αυτή του *learning activity* (μαθησιακή δραστηριότητα). Ένα learning activity είναι μια μονάδα εκπαίδευσης (instruction) ή γενικά γνώσης, που είτε αντιστοιχεί σε ένα συγκεκριμένο πόρο (content resource) του μαθησιακού υλικού είτε αναλύεται σε επιμέρους υπό- activities (που αναδρομικά ανάγονται σε ένα σύνολο από content resources).

Οι learning activities οργανώνονται σε μια δενδρική δομή, το *activity tree* (δένδρο δραστηριοτήτων). Σε κάθε activity αντιστοιχεί ένα σύνολο πληροφοριών sequencing (sequencing information), δηλαδή γνωρισμάτων (attributes) και κανόνων που ορίζουν με ποιό τρόπο το activity ή τα παιδιά του χρησιμοποιούνται από το Simple Sequencing. Οι αλγόριθμοι του Simple Sequencing διασχίζουν το activity tree εφαρμόζοντας τους κανόνες και ελέγχοντας τις τιμές των γνωρισμάτων για κάθε activity για να καθορίσουν ποιό activity πρέπει να «παραδοθεί» (deliver) στο μαθητή. Σημειώνουμε ότι μόνο οι activities – φύλλα (leaf activities) θεωρούνται «παραδοτέες», δηλαδή ότι αντιστοιχούν σε κάποιο συγκεκριμένο content resource. Οι activities που δεν είναι φύλλα χρησιμοποιούνται για την ομαδοποίηση των παιδιών τους.

Στο σχήμα 2.5 φαίνεται ένα παράδειγμα ενός activity tree. Η δενδρική ιεραρχία είναι μια ιεραρχία containment, δηλαδή π.χ. το activity AB είναι μέρος του activity A, ενώ το activity ABCA είναι μέρος του activity ABC το οποίο είναι μέρος του activity AB.



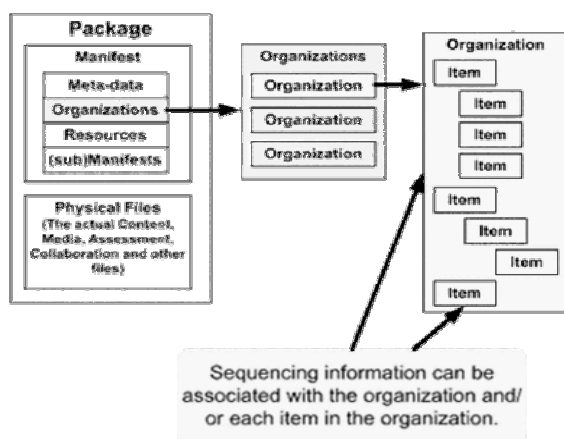
Σχήμα 2.5 Παράδειγμα ενός activity tree

Η σημασιολογία των activities είναι αόριστη. Για παράδειγμα, κάποιος μπορεί να θεωρήσει ότι το A είναι ένα μάθημα σε ένα πανεπιστήμιο, το AA είναι μια ενότητα του μαθήματος, και το AAA μια παράδοση σε αυτή την ενότητα. Με μια άλλη θεώρηση μπορεί κάποιος να ορίσει το A ως ένα κύκλο σπουδών, το AA ως μια κατηγορία μαθημάτων στον κύκλο σπουδών (π.χ. τα μαθήματα φυσικής στον κύκλο σπουδών του Ηλεκτρολόγου Μηχανικού), και το AAA ως ένα μάθημα σε αυτή την κατηγορία (π.χ. Ηλεκτρομαγνητισμός).

Το Simple Sequencing ορίζει πως η default διάσχιση του activity tree είναι η διάσχιση pre-order. Στο προηγούμενο παράδειγμα, ξεκινώντας από το A, με την pre-order διάσχιση η σειρά επίσκεψης των activities είναι A, AA, AAA, AAB, AAC, AB, ABA κτλ. Η default διάσχιση μπορεί να τροποποιηθεί ανάλογα με τις πληροφορίες sequencing που αντιστοιχούν σε κάθε activity. Για παράδειγμα, η τιμή του γνωρίσματος *Sequencing Control Flow* καθορίζει αν επιτρέπεται να γίνει διάσχιση των παιδιών ενός activity. Έτσι αν κατά τη διάσχιση του δέντρου συναντηθεί ένα activity για τον πατέρα του οποίου είναι *Sequencing Control Flow* = false, η διάσχιση σταματά και δεν επιστρέφεται κάποιο activity για παράδοση (προγραμματιστικά *nextActivity* == null). Παρεμπιπτόντως σημειώνουμε ότι η default τιμή του *Sequencing Control Flow* είναι false, δηλαδή by default δεν επιτρέπεται διάσχιση του δέντρου, αλλά μόνο απευθείας επιλογή (Choice) του επόμενου activity από το μαθητή.

2.2.1.1 Αντιστοιχία με το IMS Content Packaging Specification

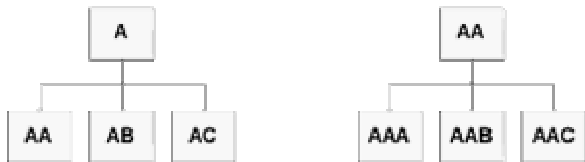
Αν και το Simple Sequencing ορίζεται ανεξάρτητα από το Content Packaging, όταν το μαθησιακό υλικό συσκευάζεται σύμφωνα με το Content Packaging τότε το activity tree αντιστοιχεί στο <organization> element του manifest. Κάθε <item> αντιστοιχεί σε ένα activity, ενώ το root activity (ρίζα του δέντρου) είναι το ίδιο το <organization>. Τα activities αντιστοιχίζονται με πληροφορίες sequencing συμπεριλαμβάνοντας ένα <sequencing> element μέσα σε κάθε <item> ή στο <organization>, όπως φαίνεται στο σχήμα 2.6. Επίσης εκτός από τις τοπικές πληροφορίες sequencing μπορούν να οριστούν και πληροφορίες sequencing με global ορατότητα (στις οποίες μπορεί να αναφέρεται οποιοδήποτε activity), συμπεριλαμβάνοντας ένα <sequencingCollection> element ως παιδί του <manifest> element. Για περισσότερες πληροφορίες όσον αφορά τη δομή και τα περιεχόμενα των elements αυτών, ο αναγνώστης παραπέμπεται στο [SS 03b].



Σχήμα 2.6 Συμπερίληψη πληροφοριών sequencing μέσα σε ένα manifest

2.2.2 Activity Clusters

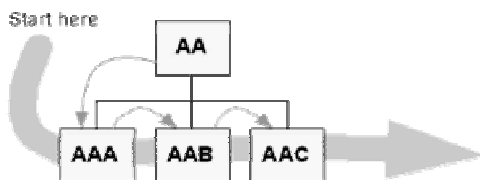
Στο Simple Sequencing υπάρχουν κανόνες και γνωρίσματα που επιδρούν σε ένα μεμονωμένο activity, και κανόνες και γνωρίσματα που επιδρούν σε clusters. Ο όρος cluster αναφέρεται σε ένα activity και τα (άμεσα) παιδιά του. Το score ενός συγκεκριμένου κανόνα ποτέ δεν επεκτείνεται πέραν του cluster.



Σχήμα 2.7 – Παραδείγματα clusters

2.2.3 Sequencing Control Modes

Εκτός αν ορίζεται διαφορετικά στις πληροφορίες sequencing, ο μαθητής μπορεί να επιλέξει απευθείας (choice) οποιαδήποτε υπο-activity σε ένα cluster, εκτός όμως από αυτό υπάρχει και η δυνατότητα διάσχισης (flow) του cluster (β.λ. σχήμα 2.8). Στο σχήμα 2.8 οι AAA, AAB και AAC είναι leaf activities, οπότε είναι παραδοτέες (θα παραδοθούν με τη σειρά που αναφέρθηκαν). Αν όμως π.χ. η AAB δεν ήταν φύλλο, η διάσχιση θα συνέχιζε με τα παιδιά της AAB πριν πάει στην AAC (διάσχιση pre-order). Η διάσχιση γίνεται μέχρι να βρεθεί κάποια activity για παράδοση ή μέχρι να διακοπεί επειδή σε κάποιο cluster δεν επιτρέπεται το flow. Υπάρχει και η δυνατότητα διάσχισης με την αντίστροφη φορά (reverse pre-order traversal). Στο σχήμα 2.8 με αντίστροφη διάσχιση θα παραδοθούν οι AAC, AAB και AAA.



Σχήμα 2.8 – Enabling forward flow for activity AA. The learner sees AAA, AAB, then AAC

Η συμπεριφορά ενός cluster όσον αφορά το choice και το flow καθορίζεται από την ομάδα γνωρισμάτων *Sequencing control modes* και συγκεκριμένα από τα γνωρίσματα *Sequencing Control Choice*, *Sequencing Control Choice Exit*, *Sequencing Control Flow*, και *Sequencing Control Forward Only*.

Οι τιμές των *Sequencing Control Choice* και *Sequencing Control Flow* για ένα activity καθορίζουν αν επιτρέπεται το choice και το flow, αντίστοιχα, για το cluster ρίζα του οποίου είναι το activity. Το *Sequencing Control Forward Only* καθορίζει αν μέσα στο cluster επιτρέπεται μόνο η προς τα μπρος διάσχιση ή αν επιτρέπονται και οι δύο φορές. Το *Sequencing Control Choice Exit* καθορίζει αν ένα ενεργό (active) activity επιτρέπεται να τερματιστεί επειδή ο μαθητής επέλεξε κάποιο άλλο activity.

2.2.4 Objectives

Στις πληροφορίες sequencing ενός activity μπορεί να ορίζεται ένα σύνολο από *objectives*. Ένα objective είναι μια οντότητα που αντιπροσωπεύει έναν μαθησιακό στόχο του συγκεκριμένου activity ή πιο γενικά κάποιο αποτέλεσμα της αλληλεπίδρασης του μαθητή με το συγκεκριμένο activity.

Για παράδειγμα, μετά την παρουσίαση ενός μέρους της διδακτέας ύλης μπορεί να υπάρχει μια εξέταση με στόχο την εκτίμηση του βαθμού κατανόησης της ύλης από το μαθητή. Στο activity «εξέταση» μπορεί να οριστεί το objective «κατανόηση» όπου θα αποθηκευτεί το αποτέλεσμα της αλληλεπίδρασης του μαθητή με το activity, δηλαδή ο βαθμός της εξέτασης.

Για κάθε objective το σύστημα κρατά στη μνήμη ένα σύνολο πληροφοριών σχετικά με το κατά πόσο το objective έχει «επιτευχθεί». Αυτές οι πληροφορίες ονομάζονται *Objective Progress Information* και περιγράφονται στην επόμενη παράγραφο.

Υπάρχουν δύο είδη από objectives : τα *primary objectives* και τα «απλά» objectives (ας τα ονομάσουμε εδώ *secondary objectives*). Η διαφορά μεταξύ τους είναι ότι μόνο τα *primary objectives* συμμετέχουν στα rollups. Κάθε activity έχει ακριβώς ένα primary objective. Αν δεν ορίζεται κάποιο primary objective στις πληροφορίες sequencing του activity τότε χρησιμοποιείται ένα default primary objective. Εκτός του primary, το activity μπορεί να έχει και οποιοδήποτε αριθμό από secondary objectives.

Κάθε objective έχει ένα μοναδικό objective ID. Επειδή κάθε activity έχει ακριβώς ένα primary objective, ειδικά στα primary objectives δεν είναι υποχρεωτικό να δηλώνεται κάποιο objective ID. Αν δεν δηλώνεται, τότε χρησιμοποιείται το ID του activity στο οποίο ανήκει το primary objective. Αντίθετα στα secondary objectives πάντα δηλώνεται ένα objective ID.

Στις πληροφορίες sequencing ενός activity μπορεί να υπάρχουν κανόνες sequencing (βλ. παράγραφο 2.2.7) οι οποίοι στις συνθήκες τους να αναφέρονται σε κάποιο από τα objectives του activity (τοπικά objectives). Απαγορεύεται ένας κανόνας να αναφέρεται άμεσα σε objectives άλλων activities (εξωτερικά objectives). Η αναφορά αυτή μπορεί να γίνει μόνο έμμεσα, μέσω *Objective Maps*. Ένα *Objective Map* είναι ένα mapping των δεδομένων *Objective Progress Information* ενός τοπικού activity από ('read' map) και/ή προς ('write' map) τα αντίστοιχα δεδομένα ενός εξωτερικού (global) objective. Έτσι ένας κανόνας μπορεί να αναφέρεται σε ένα τοπικό objective, το οποίο όμως να «αντλεί» τα δεδομένα του από ένα εξωτερικό objective, μέσω ενός *Objective Map* που έχει οριστεί στο τοπικό objective.

2.2.5 Tracking Model

Τα αποτελέσματα της αλληλεπίδρασης του μαθητή με τα activities καταγράφονται σε ένα σύνολο από δεδομένα που ονομάζεται tracking model. Τα δεδομένα του tracking model χωρίζονται σε 2 κατηγορίες : αυτά που αφορούν τα activities (*Activity Progress Information*) και αυτά που αφορούν τα objectives (*Objective Progress Information*).

2.2.5.1 Objective Progress Information

Το Objective Progress Information αποτελείται από τα παρακάτω δεδομένα :

1. *Objective Progress Status* : Boolean μεταβλητή που εκφράζει το αν είναι γνωστό αν το objective έχει ικανοποιηθεί. Default τιμή : false.
2. *Objective Satisfied Status* : Boolean μεταβλητή που εκφράζει το αν το objective έχει ικανοποιηθεί. Η τιμή αυτή είναι αδιάφορη αν το *Objective Progress Status* είναι false. Default τιμή : false.
3. *Objective Measure Status* : Boolean μεταβλητή που εκφράζει το αν είναι γνωστός ο βαθμός (measure) του objective. Default τιμή : false.
4. *Objective Normalized Measure* : Δεκαδικός αριθμός στο [-1..1]. Είναι ο βαθμός του objective. Η τιμή αυτή είναι αδιάφορη αν το *Objective Measure Status* είναι false. Default τιμή : 0.0.

2.2.5.2 Activity Progress Information

Το Activity Progress Information αποτελείται από τα παρακάτω δεδομένα :

1. *Activity Attempt Count* : Μη αρνητικός ακέραιος. Είναι ο αριθμός των «προσπαθειών» που έχουν γίνει στο συγκεκριμένο activity, δηλαδή το πόσες φορές το activity έχει παραδοθεί. Default τιμή : 0.
2. *Attempt Progress Status* : Boolean μεταβλητή που εκφράζει το αν είναι γνωστό αν το activity έχει ολοκληρωθεί. Default τιμή : false.
3. *Attempt Completion Status* : Boolean μεταβλητή που εκφράζει το αν το activity έχει ολοκληρωθεί. Η τιμή αυτή είναι αδιάφορη αν το *Attempt Progress Status* είναι false. Default τιμή : false.

Εκτός από τα παραπάνω το specification ορίζει και μερικές ακόμη μεταβλητές που δεν έχουμε υλοποιήσει και επομένως δεν κρίνεται σκόπιμο να παρουσιαστούν εδώ. Για περισσότερες πληροφορίες ο αναγνώστης παραπέμπεται στο [SS, 03a]. Επίσης στην υλοποίηση το *Activity Attempt Count* μετονομάστηκε σε *Activity Attempted* και είναι boolean μεταβλητή που εκφράζει το αν το activity έχει παραδοθεί ή όχι.

2.2.6 Activity State Model

Εκτός από το Tracking Model, κάποιες επιπλέον πληροφορίες για τα activities αποθηκεύονται στο Activity State Model. Συγκεκριμένα για κάθε activity αποθηκεύεται το αν είναι ενεργό (active). Ένα activity είναι ενεργό όταν μια προσπάθεια σε αυτό είναι σε εξέλιξη, δηλαδή όταν το activity έχει παραδοθεί αλλά ακόμα δεν έχει τερματιστεί. Όταν ένα activity είναι ενεργό, τότε όλοι οι πρόγονοί του είναι και αυτοί ενεργοί. Επίσης στο Activity State Model αποθηκεύεται το ποιο είναι το τρέχον (current) activity. Το *Current Activity* είναι συνήθως το τελευταίο activity που παραδόθηκε, υπάρχουν όμως και κάποιες ειδικές περιπτώσεις που αυτό δεν ισχύει (βλ. [SS, 03a]).

2.2.7 Sequencing Rules

Η συμπεριφορά ενός activity ως προς το sequencing μπορεί να διαμορφωθεί αναθέτοντας σε αυτό κανόνες sequencing (*sequencing rules*). Οι κανόνες αυτοί χωρίζονται σε τρεις κατηγορίες :

1. *Precondition Rules* : Εφαρμόζονται κατά την αναζήτηση του επόμενου activity που θα παραδοθεί. Καθορίζουν αν το συγκεκριμένο activity θα παραδοθεί ή όχι.
2. *Exit Condition Rules* : Εφαρμόζονται όταν ένα activity τερματίζεται, σε όλους τους προγόνους αυτού του activity και καθορίζουν αν κάποιος πρόγονος πρέπει και αυτός να τερματιστεί.
3. *Post Condition Rules* : Εφαρμόζονται στο activity που τερματίζεται και μπορούν να έχουν ως αποτέλεσμα την έκδοση ενός νέου sequencing ή termination request.

Κάθε κανόνας αποτελείται από 2 μέρη : την ενέργεια (*rule action*) και τις συνθήκες (*rule conditions*) που πρέπει να ισχύουν για να εκτελεστεί η ενέργεια. Οι δυνατές συνθήκες είναι κοινές για τους τρεις τύπους κανόνων και αφορούν τις τιμές στο Tracking Model για το activity ή τα objectives του. Για παράδειγμα η συνθήκη *Objective Measure Greater Than* είναι true όταν για το objective που προσδιορίζει η συνθήκη (*Rule Condition Referenced Objective*) είναι *Objective Measure Status* = true και το *Objective Normalized Measure* είναι μεγαλύτερο από κάποια τιμή που προσδιορίζει η συνθήκη (*Rule Condition Measure Threshold*). Ένας κανόνας μπορεί να περιέχει πολλές τέτοιες συνθήκες, οι οποίες αφού αποτιμηθούν μια-μια συνδυάζονται με AND ή OR για να προκύψει η συνολική συνθήκη. Η επιλογή του AND ή του OR γίνεται σύμφωνα με την τιμή του *Condition Combination*.

Οι δυνατές τιμές για την ενέργεια του κανόνα εξαρτώνται από τον τύπο του. Για παράδειγμα ένας precondition κανόνας μπορεί να έχει μια από τις παρακάτω 4 ενέργειες :

Skip, Disabled, Hidden from Choice και Stop Forward Traversal.

Τα παραπάνω διασαφηνίζονται με το ακόλουθο παράδειγμα του σχήματος 2.9, όπου φαίνεται ένα τμήμα της XML περιγραφής (βλ. §2.2.1.1) των πληροφοριών sequencing ενός activity. Το element `<imsss:sequencingRules>` υποτίθεται ότι βρίσκεται μέσα στο `<imsss:sequencing>` element του activity και περιέχει τους κανόνες sequencing για το activity αυτό.

```
<imsss:sequencingRules>
  <imsss:preConditionRule>
    <imsss:ruleConditions conditionCombination="all">
      <imsss:ruleCondition condition="attempted"/>
      <imsss:ruleCondition condition="completed" operator="not"/>
    </imsss:ruleConditions>
    <imsss:ruleAction action="stopForwardTraversal"/>
  </imsss:preConditionRule>
</imsss:sequencingRules>
```

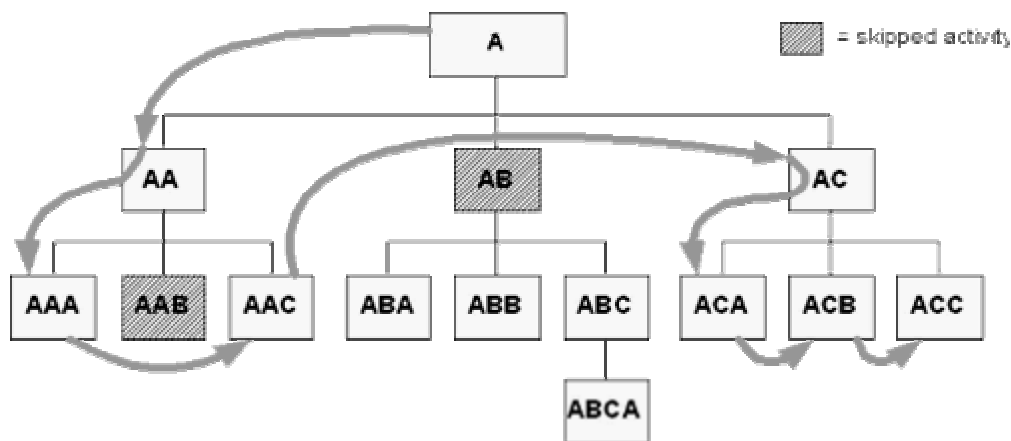
Σχήμα 2.9 Παράδειγμα κανόνα Simple Sequencing

Όπως βλέπουμε, ορίζεται ένας precondition κανόνας με ενέργεια *Stop Forward Traversal*, η οποία θα εκτελεστεί αν και μόνο αν η (συνολική) συνθήκη του κανόνα αποτιμηθεί ως true. Η συνολική συνθήκη αποτελείται από δύο επιμέρους συνθήκες

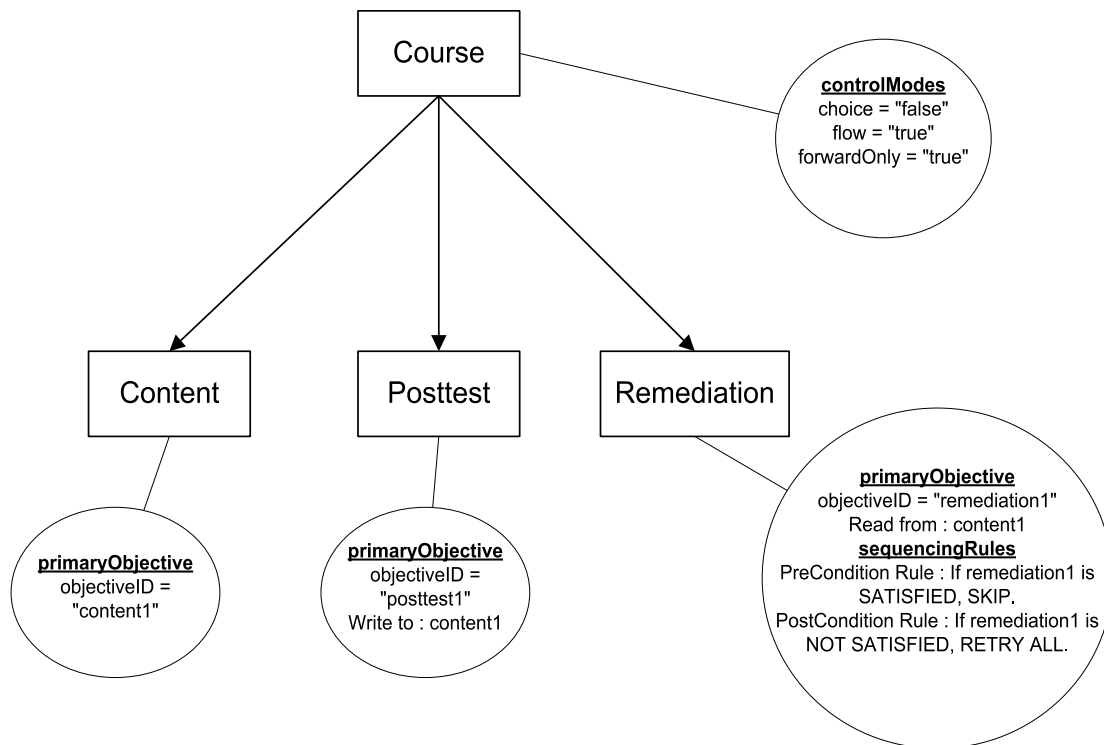
και συγκεκριμένα τις *Attempted* και *Completed*. Στην τελευταία ορίζεται ότι πρέπει να αντιστραφεί (τελεστής NOT) πριν συνδυαστεί με τις υπόλοιπες. Επειδή η τιμή του *Condition Combination* είναι *All*, η συνολική συνθήκη θα προκύψει παίρνοντας το λογικό γινόμενο (τελεστής AND) των επιμέρους συνθηκών. Έτσι έχουμε τη συνολική συνθήκη *Attempted AND NOT Completed*.

Στο σχήμα 2.10 φαίνεται το πώς η διάσχιση του *activity tree* (θεωρούμε ότι το flow είναι enabled σε όλα τα clusters) μπορεί να επηρεαστεί από κανόνες sequencing. Όπως φαίνεται στο σχήμα, τα activities AAB και AB έχουν κάποιον precondition κανόνα με *rule action = Skip*, με αποτέλεσμα τα activities αυτά μαζί με τα παιδιά τους να παραλείπονται κατά τη διάσχιση.

Στο σχήμα 2.11 φαίνεται ένα παράδειγμα του πώς μπορούν να χρησιμοποιηθούν οι κανόνες sequencing για την υλοποίηση ενός συγκεκριμένου σεναρίου μάθησης. Αρχικά στο μαθητή θα παραδοθεί το activity Content, που περιέχει την ύλη του μαθήματος. Στη συνέχεια το Posttest θα αξιολογήσει το βαθμό κατανόησης της ύλης από το μαθητή. Αν ο βαθμός κατανόησης είναι ικανοποιητικός, το objective content1 (primary objective του activity Content) θα τεθεί (μέσω του *Objective Map* από το τοπικό objective posttest1 προς το global objective content1) ότι είναι satisfied (δηλαδή *Objective Progress Status = true* και *Objective Satisfied Status = true*) και θα παραληφθεί το activity Remediation (λόγω του precondition rule με ενέργεια *Skip* στο activity Remediation). Διαφορετικά, μετά το Posttest θα παρασχεθεί ενισχυτική διδασκαλία (Remediation) για την κάλυψη των κενών του μαθητή και στη συνέχεια το μάθημα θα επαναληφθεί (λόγω του post condition rule με ενέργεια *Retry All* στο activity Remediation).



Σχήμα 2.10 – Παράδειγμα της επίδρασης κανόνων που ορίζουν πως υπό κάποιες συνθήκες ορισμένα activities πρέπει να προσπεραστούν (conditional skip) κατά τη διάσχιση.



Σχήμα 2.11 - Χρήση κανόνων sequencing για την υλοποίηση ενός συγκεκριμένου σεναρίου μάθησης

2.2.8 Rollups

Όταν ένα activity έχει παιδιά, με άλλα λόγια όταν είναι ο πατέρας σε ένα activity cluster, υπάρχει η δυνατότητα τα δεδομένα του Tracking Model που αφορούν το activity ή τα objectives του να υπολογίζονται συναρτήσει των αντίστοιχων δεδομένων των παιδιών του. Αυτός ο υπολογισμός ονομάζεται *rollup* και γίνεται από τη ρουτίνα *Overall Rollup Process*. Η ρουτίνα αυτή καλείται κάθε φορά που τερματίζεται κάποιος activity και ξεκινώντας από το activity που τερματίστηκε υπολογίζει και ενημερώνει τα δεδομένα του Tracking Model διαδοχικά για όλους τους προγόνους του, μέχρι τη ρίζα του δέντρου.

Ο τρόπος με τον οποίο γίνεται το *rollup* σε ένα cluster καθορίζεται από δύο παράγοντες : τα γνωρίσματα *Rollup Controls* των παιδιών του cluster και τους κανόνες rollup (*rollup rules*) που έχουν οριστεί στον πατέρα του cluster.

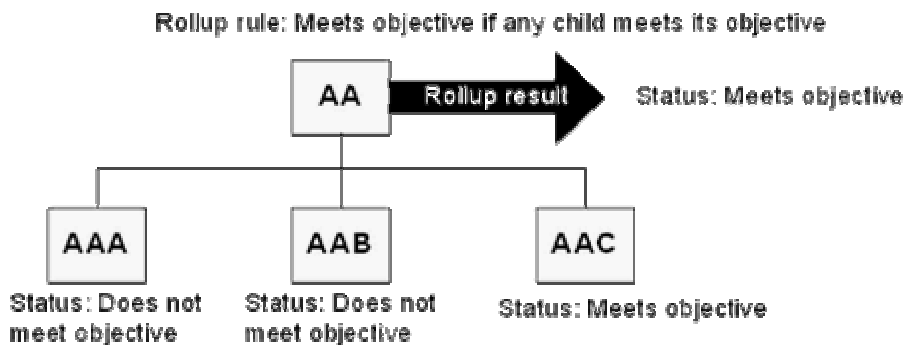
Οι κανόνες rollup έχουν μια αρκετά διαφορετική λογική από τους κανόνες sequencing γιατί ενώ ένας κανόνας sequencing αναφέρεται μόνο στο activity στο οποίο ορίζεται, ένας κανόνας rollup αναφέρεται σε όλο το cluster. Η μορφή ενός κανόνα rollup μπορεί να εκφραστεί ως : *If child-activity set, condition set Then action*. Το *Child Activity Set* είναι το υποσύνολο εκείνο των παιδιών του activity που θα ληφθεί υπόψη στην αποτίμηση του κανόνα. Η συνολική συνθήκη του κανόνα μπορεί να αποτελείται από επιμέρους συνθήκες που συνδυάζονται με AND ή OR για να προκύψει η συνολική συνθήκη, όπως ακριβώς συμβαίνει και με τους κανόνες sequencing (βλ. §2.2.7).

Υπάρχουν οι εξής επιλογές για το *Child Activity Set* :

1. *All* - Ο κανόνας αποτιμάται ως true αν και μόνο αν η συνολική συνθήκη είναι true σε όλα τα παιδιά του activity.
2. *Any* - Ο κανόνας αποτιμάται ως true αν και μόνο αν η συνολική συνθήκη είναι true σε κάποιο από τα παιδιά του activity.
3. *None* - Ο κανόνας αποτιμάται ως true αν και μόνο αν η συνολική συνθήκη είναι false σε όλα τα παιδιά του activity.
4. *At Least Count* - Ο κανόνας αποτιμάται ως true αν και μόνο αν η συνολική συνθήκη είναι true σε τουλάχιστον τόσα παιδιά του activity, όσα προσδιορίζει το πεδίο *Rollup Minimum Count* του κανόνα.
5. *At Least Percent* - Ο κανόνας αποτιμάται ως true αν και μόνο αν η συνολική συνθήκη είναι true στο ποσοστό εκείνο των παιδιών του activity που προσδιορίζει το πεδίο *Rollup Minimum Percent* του κανόνα.

Η ενέργεια (*Rollup Action*) του κανόνα καθορίζει το αν το activity θα γίνει *Completed* ή αν το primary objective του θα γίνει *Satisfied*. Έχουμε δηλαδή τέσσερις δυνατές τιμές για το *Rollup Action* : *Satisfied*, *Not Satisfied*, *Completed*, και *Incomplete*.

Στο σχήμα 2.12 φαίνεται ένα παράδειγμα rollup. Στον πατέρα του cluster έχει οριστεί ο εξής rollup κανόνας : Το primary objective του πατέρα γίνεται *satisfied* αν το primary objective οποιουδήποτε εκ των παιδιών του είναι *satisfied*.



Σχήμα 2.12 Παράδειγμα rollup

2.2.9 Overall Sequencing Process

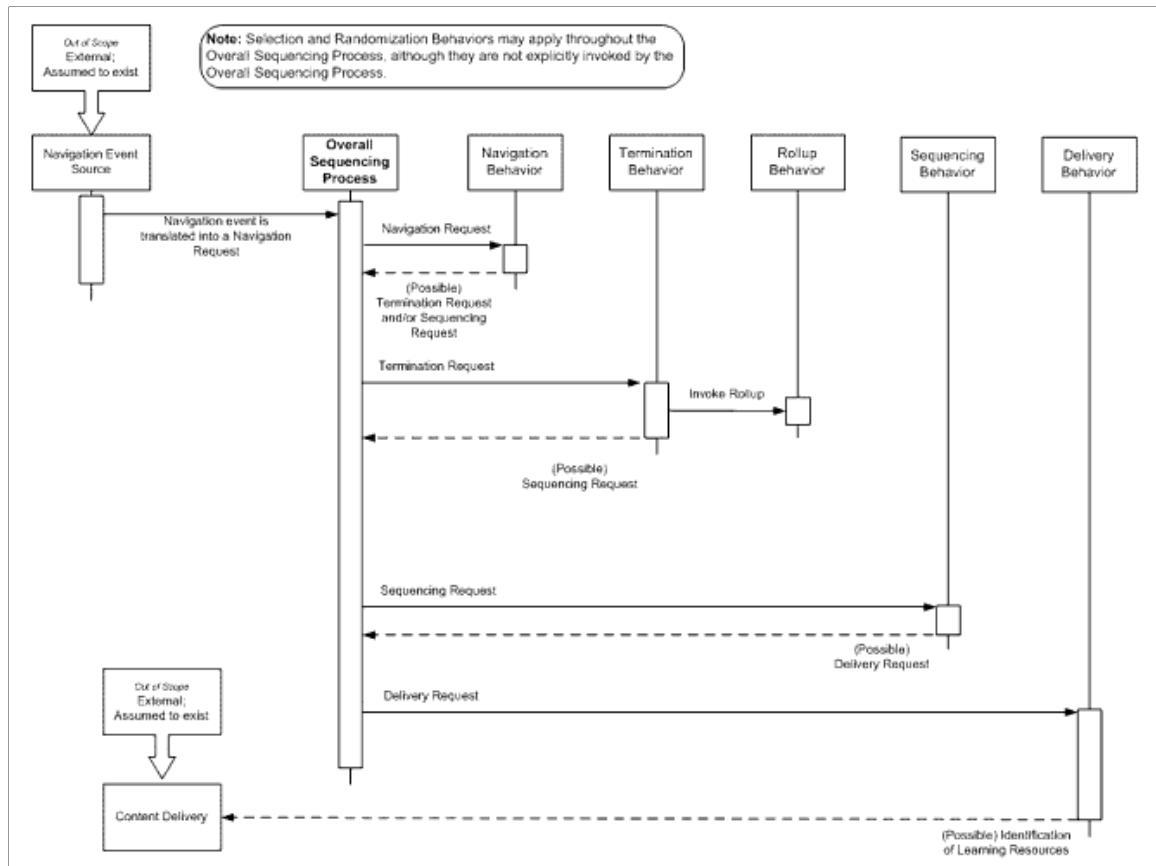
Στις προηγούμενες παραγράφους είδαμε τα βασικά χαρακτηριστικά (features) του Simple Sequencing Specification. Τώρα θα παρουσιαστεί μια υψηλού επιπέδου όψη (high level view) της sequencing διεργασίας (process) αυτής καθαυτής. Η κεντρική διεργασία του Simple Sequencing ονομάζεται *Overall Sequencing Process* (OP).

Η OP εκτελείται μέσα στο scope κάποιου Learning Technology System (LTS), όπως για παράδειγμα σε κάποιο LMS, στο οποίο έχει υλοποιηθεί κάποιος τρόπος αναπαράστασης και διαχείρισης των activity trees, των πληροφοριών sequencing, των content resources, των information models για την παρακολούθηση (tracking) του μαθητή, και υπάρχει κάποιο runtime περιβάλλον και user interface για την επικοινωνία του συστήματος με το μαθητή. Στο σχήμα 2.13 φαίνεται το context στο οποίο τρέχει η OP, δηλαδή τα υπόλοιπα μέρη του LTS με τα οποία αλληλεπιδρά.

Activity δεν ενημερώνει μόνο του το Tracking Model - ενημερώνεται το Tracking Model με default τιμές.

- Καλείται η *OverallRollupProcess*, με την οποία γίνονται τα rollups (βλ. §2.2.8).
 - Καλείται η *Sequencing Exit Action Rules Subprocess*, με την οποία εφαρμόζονται οι *Exit Condition Rules* (βλ. §2.2.7) σε όλους τους προγόνους του *Current Activity*, ξεκινώντας από το root activity.
 - Καλείται η *Sequencing Post Condition Rules Subprocess*, με την οποία εφαρμόζονται οι *Post Condition Rules* (βλ. §2.2.7) στο *Current Activity*. Αποτέλεσμα της εφαρμογής αυτών των κανόνων μπορεί να είναι η έκδοση ενός νέου termination request (το οποίο χειρίζεται η ίδια η *Termination Request Process*) και/ή ενός sequencing request (το οποίο επιστρέφεται στην OP).
3. Αν στο προηγούμενο βήμα η *Termination Request Process* επέστρεψε κάποιο sequencing request, αυτό αντικαθιστά το sequencing request που επέστρεψε η *Navigation Request Process* στο βήμα 1.
 4. Το sequencing request περνιέται στην *Sequencing Request Process*, η οποία ανάλογα με το είδος του sequencing request καλεί την κατάλληλη διεργασία για να το χειριστεί (π.χ. στην περίπτωση του *Continue* sequencing request θα κληθεί η *Continue Sequencing Request Process*). Αποτέλεσμα της επεξεργασίας του sequencing request είναι κάποιο *delivery request*, το οποίο επιστρέφεται στην OP. Το *delivery request* είναι ουσιαστικά ένα activity που είναι υποψήφιο για παράδοση. Φυσικά το *delivery request* ενδέχεται να είναι null.
 5. Καλείται η *Delivery Request Process* η οποία ελέγχει την εγκυρότητα του *delivery request*, το αν δηλαδή το activity μπορεί να παραδωθεί. Για παράδειγμα αν στο υποψήφιο activity ή σε κάποιο πρόγονό του υπάρχει κάποιος precondition κανόνας (βλ. §2.2.7) με ενέργεια *Disabled* ο οποίος αποτιμηθεί ως true, το υποψήφιο activity δεν μπορεί να παραδωθεί.
 6. Καλείται η *Content Delivery Environment Process* η οποία κάνει το παραδοτέο activity και όλους του προγόνους του active και θέτει ως *Current Activity* το παραδοτέο activity.
 7. Το παραδοτέο activity επιστρέφεται στο LTS, το οποίο αναλαμβάνει την παρουσίαση στο μαθητή των content resources που αντιστοιχούν στο activity. Τα content resources ενδέχεται να ενημερώσουν το Tracking Model, ο τρόπος όμως με τον οποίο μπορεί να γίνει αυτό δεν προδιαγράφεται στο Simple Sequencing Specification. Τα παραπάνω βήματα επαναλαμβάνονται με κάθε νέο navigation request.

Τα παραπάνω βήματα συνοψίζονται στο σχήμα 2.14.



Σχήμα 2.14 Overall Sequencing Process

Navigation Request	Action
<i>Start</i>	If the <i>Current Activity (AM.1.2)</i> is undefined, Issue a <i>Start</i> sequencing request.
<i>Resume All</i>	If the <i>Current Activity (AM.1.2)</i> is undefined and the <i>Suspended Activity (AM.1.2)</i> is defined, Issue a <i>Resume All</i> sequencing request.
<i>Continue</i>	If <i>Activity is Active (AM.1.1)</i> for the <i>Current Activity (AM.1.2)</i> is <i>True</i> , Issue an <i>Exit</i> termination request. Issue a <i>Continue</i> sequencing request.
<i>Previous</i>	If <i>Activity is Active (AM.1.1)</i> for the <i>Current Activity (AM.1.2)</i> is <i>True</i> , Issue an <i>Exit</i> termination request. Issue a <i>Previous</i> sequencing request.
<i>Forward</i>	Issue a sequencing request to traverse the "history-based activity record" forward in time. The corresponding sequencing request and associated behavior is not specified in this version of the Simple Sequencing Specification.
<i>Backward</i>	Issue a sequencing request to traverse the "history-based activity record" backward in time. The corresponding sequencing request and associated behavior is not specified in this version of the Simple Sequencing Specification.
<i>Choice</i>	If <i>Activity is Active (AM.1.1)</i> for the <i>Current Activity (AM.1.2)</i> is <i>True</i> , Issue an <i>Exit</i> termination request. Issue a <i>Choice</i> sequencing request. The request is accompanied by the identification of the target activity.

<i>Exit</i>	Issue an <i>Exit</i> termination request. Issue an <i>Exit</i> sequencing request. The current attempt on the <i>Current Activity</i> (AM.1.2) is terminated normally; the attempt is over. The termination of the activity was not the result of any other external navigation event (e.g., <i>Continue</i> , <i>Previous</i> , <i>Choice</i>).
<i>Exit All</i>	Issue an <i>Exit All</i> termination request. Issue an <i>Exit</i> sequencing request.
<i>Suspend All</i>	Issue a <i>Suspend All</i> termination request. Issue an <i>Exit</i> sequencing request. The current attempt on the <i>Current Activity</i> (AM.1.2) and all of its ancestors are terminated normally; the attempts are not over and the activities are not complete. The activities may be resumed at some time in the future (resumption is not a new attempt). A Simple Sequencing processor must record sufficient state and tracking information so that the activities may be resumed in the future.
<i>Abandon</i>	Issue an <i>Abandon</i> termination request. Issue an <i>Exit</i> sequencing request. The current attempt on the <i>Current Activity</i> (AM.1.2) is terminated abnormally and the activity is not complete. The attempt may not be resumed. There is no rollback of any tracking data.
<i>Abandon All</i>	Issue an <i>Abandon All</i> termination request. Issue an <i>Exit</i> sequencing request. The current attempt on the <i>Current Activity</i> (AM.1.2) and all of its ancestors are terminated abnormally and the activities are not complete. The attempts may not be resumed. There is no rollback of any tracking data.

Πίνακας 2.1 Navigation Requests και μετάφρασή τους σε Sequencing και Termination Requests.

2.3 Συναφείς προδιαγραφές

Θα παρουσιάσουμε τώρα τρεις ακόμα προδιαγραφές που έχει δημιουργήσει το IMS Global Learning Consortium και οι οποίες σχετίζονται άμεσα ή έμμεσα με αυτές που έχουμε παρουσιάσει ως τώρα. Αν και το σύστημα στην παρούσα μορφή του δεν περιλαμβάνει κάποια υλοποίηση των προδιαγραφών αυτών, είναι πολύ πιθανό αυτό να συμβεί σε κάποια μελλοντική επέκτασή του, για αυτό και κρίνεται σκόπιμο να γίνει μια σύντομη παρουσίασή τους εδώ.

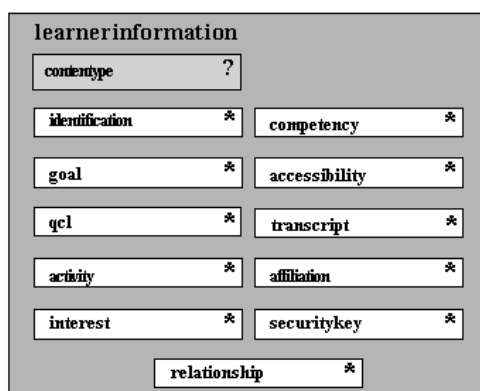
2.3.1 IMS Learning Resource Meta-Data Specification

Ένα σημαντικό πρόβλημα στη σημερινή κατάσταση του eLearning είναι ότι ενώ υπάρχει πληθώρα software εργαλείων για τη δημιουργία μαθησιακού υλικού και τεράστιος όγκος μαθησιακού υλικού που είναι διαθέσιμο ελεύθερα, παρόλα αυτά δεν υπάρχει κάποιος κοινός μηχανισμός αναζήτησης και χρήσης του υλικού. Η λύση του προβλήματος προϋποθέτει τον ορισμό ενός τρόπου «σήμανσης» του υλικού με πληροφορίες που το περιγράφουν, και οι οποίες θα χρησιμοποιούνται για την αναζήτηση μαθησιακού υλικού με αυτόματο τρόπο. Αναφερόμαστε σε αυτές τις πληροφορίες με τον όρο *μέτα-δεδομένα* (*meta-data*), γιατί ουσιαστικά πρόκειται για δεδομένα που περιγράφουν άλλα δεδομένα. Η προδιαγραφή *IMS Learning Resource Meta-Data* ορίζει ένα μοντέλο τέτοιων μετα-δεδομένων [MD, 01a], τα οποία μπορούν να χρησιμοποιηθούν για την περιγραφή του μαθησιακού υλικού. Για παράδειγμα, σε ένα Content Package μπορούμε να συμπεριλάβουμε μετα-δεδομένα

όπως περιγραφή του μαθήματος, λέξεις κλειδιά (keywords), δημιουργοί, πνευματικά δικαιώματα κτλ. Η αξία αυτών των πληροφοριών είναι κυρίως ότι μας δίνουν τη δυνατότητα να δημιουργήσουμε μηχανές αναζήτησης με τις οποίες θα μπορεί κάποιος να αναζητά και να εντοπίζει μαθησιακό υλικό σύμφωνα με κάποια κριτήρια αναζήτησης. Έτσι θα μπορεί κάποιος να αναζητά μαθήματα που έχουν ως αντικείμενο π.χ. την Οργανική Χημεία, ή που έχουν γραφτεί από κάποιον συγκεκριμένο άνθρωπο κάποια συγκεκριμένη χρονιά. Το μοντέλο μετα-δεδομένων που ορίζεται στο IMS Learning Resource Meta-Data βασίζεται στην προδιαγραφή *IEEE Learning Object Meta-Data [IEEE LOM 6.1]*, με κάποιες μικρές τροποποιήσεις που κατά πάσα πιθανότητα θα συμπεριληφθούν σε επόμενη έκδοση της τελευταίας. Όμως εκτός από το μοντέλο μετα-δεδομένων το IMS Global Learning Consortium έχει ορίσει και ένα XML binding αυτού **[MD, 01b]**.

2.3.2 IMS Learner Information Package Specification

Ένα *Learner Information System (LIS)* είναι ένα σύστημα αποθήκευσης και διαχείρισης πληροφοριών για μαθητές (learners) ή παραγωγούς μαθησιακού υλικού (producers). Το IMS Global Learning Consortium έχει δημιουργήσει την προδιαγραφή *IMS Learner Information Package Specification [LIP, 01a]* η οποία ορίζει ένα data model που μπορεί να χρησιμοποιηθεί για την ανταλλαγή πληροφορίας μεταξύ διαφορετικών LISs. Το data model αποτελείται από 11 κατηγορίες πληροφορίας (learner information), οι οποίες φαίνονται στο σχήμα 2.15.



Σχήμα 2.15 – Learner Information Categories

Κάθε κατηγορία περιλαμβάνει ένα σύνολο από πεδία. Για παράδειγμα, η κατηγορία identification περιέχει πεδία που αφορούν πληροφορίες όπως όνομα, διεύθυνση, contact information (τηλέφωνο, email, web address κ.α.), η κατηγορία qcl περιέχει πληροφορίες για τα προσόντα (qualifications), τις πιστοποιήσεις (certifications) και τις άδειες (licenses) που κατέχει ο learner, ενώ η κατηγορία activity αφορά τις σπουδές και την εργασιακή εμπειρία του learner.

Επίσης έχει οριστεί ένα XML binding **[LIP, 01b]** για τα πεδία αυτά. Έτσι είναι δυνατή η ανταλλαγή πληροφορίας σε XML μορφή μεταξύ διαφορετικών LISs.

2.3.3 IMS Question & Test Interoperability Specification

Τέλος, μια ακόμη πολύ ενδιαφέρουσα προδιαγραφή είναι το *IMS Question & Test Interoperability (QTI)*, η οποία ορίζει ένα (abstract) data model [QTI 2.0, 05a] για την αναπαράσταση online εξετάσεων (tests), ή, στην ορολογία του QTI, αξιολογήσεων (assessments). Ένα assessment αποτελείται από ένα ή περισσότερα *assessment items*, όπου κάθε *assessment item* είναι ένα σύνολο από *interactions* (π.χ. μια multiple choice ερώτηση είναι ένα είδος interaction) μαζί με ένα σύνολο κανόνων για τον υπολογισμό των αποτελεσμάτων (*outcomes*) συναρτήσει των απαντήσεων (*responses*) του εξεταζόμενου. Εκτός από το abstract data model έχει οριστεί και ένα XML binding [QTI 2.0, 05b] αυτού έτσι ώστε να είναι δυνατή η ανταλλαγή των assessments και των αποτελεσμάτων τους μεταξύ διαφορετικών συστημάτων (π.χ. authoring tools, item banks, learning systems και assessment delivery systems) σε μια κοινή XML μορφή.

Το QTI specification είναι αρκετά μεγάλο και μια λεπτομερής παρουσίασή του είναι εκτός των σκοπών αυτής της εργασίας. Θα επιχειρήσουμε όμως να δώσουμε μια γενική ιδέα του τι περίπου μπορεί να κάνει το QTI μέσα από δύο απλά παραδείγματα. Στο σχήμα 2.16 φαίνεται μια multiple choice ερώτηση, η οποία αντιστοιχεί σε ένα interaction που περιέχεται σε κάποιο assessment item. Συγκεκριμένα πρόκειται για τον τύπο *choiceInteraction*, όπου στον εξεταζόμενο (*candidate* στην ορολογία του QTI) παρουσιάζεται μια λίστα επιλογών και αυτός καλείται να *επιλέξει* μια ή περισσότερες από αυτές.



UNATTENDED LUGGAGE

Look at the text in the picture.

**NEVER LEAVE
LUGGAGE
UNATTENDED**

What does it say?

You must stay with your luggage at all times.	<input type="radio"/>
Do not let someone else look after your luggage.	<input type="radio"/>
Remember your luggage when you leave.	<input type="radio"/>

Σχήμα 2.16 – QTI example 1 : choiceInteraction

Η XML αναπαράσταση του παραπάνω interaction δίνεται στο σχήμα 2.17. Όπως φαίνεται στο σχήμα, η απάντηση (response) του εξεταζόμενου δηλώνεται ως μεταβλητή στην αρχή του item με το <responseDeclaration> element. Οι τιμές που μπορεί να πάρει το response είναι οι τιμές των αντίστοιχων identifier attributes των μεμονωμένων <simpleChoice>s. Η σωστή απάντηση περιλαμβάνεται στη δήλωση του response. Το response αντιστοιχίζεται στο choiceInteraction μέσω του responseIdentifier attribute του <choiceInteraction> element. Το αποτέλεσμα (outcome) του assessment item είναι κάποιος βαθμός (μεταβλητή SCORE). Η βαθμολόγηση γίνεται σύμφωνα με ένα

response processing template, δηλαδή ένα σύνολο κανόνων που περιγράφουν πώς θα υπολογιστεί ο βαθμός. Στη συγκεκριμένη περίπτωση χρησιμοποιείται ένα standard template που ονομάζεται *Match Correct*. Ο αλγόριθμος του Match Correct είναι πολύ απλός : αν το response είναι ίσο (matches) με την σωστή απάντηση που έχει δηλωθεί, τότε θέσει τη μεταβλητή SCORE ίση με 1, διαφορετικά θέσει τη ίση με 0. Υπάρχουν και άλλα templates εκτός από το Match Correct, π.χ. το *Map Response*, όπου κάθε δυνατή επιλογή αντιστοιχίζεται με μια τιμή. Στην περίπτωση που το response αποτελείται από μια επιλογή (cardinality="single") τότε το Map Response απλά θέτει το SCORE ίσο με την τιμή της επιλογής αυτής. Αν το response αποτελείται από περισσότερες επιλογές (cardinality="multiple"), όπως π.χ. στην περίπτωση της ερώτησης «από ποιά από τα παρακάτω χημικά στοιχεία αποτελείται το μόριο του νερού», τότε το Map Response θέτει το SCORE ίσο με το άθροισμα των τιμών των επιμέρους επιλογών. Σημειώνουμε ότι με τον τρόπο αυτό μπορούμε και να αφαιρούμε μονάδες για κάθε λάθος επιλογή, ορίζοντας ως τιμή της λάθος επιλογής έναν αρνητικό αριθμό.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This example adapted from the PET Handbook, copyright University of Cambridge ESOL Examinations -->
<assessmentItem xmlns="http://www.imslobal.org/xsd/imsqti_v2p0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imslobal.org/xsd/imsqti_v2p0_imsqti_v2p0.xsd" identifier="choice"
title="Unattended Luggage" adaptive="false" timeDependent="false">
  <responseDeclaration identifier="RESPONSE" cardinality="single" baseType="identifier">
    <correctResponse>
      <value>ChoiceA</value>
    </correctResponse>
  </responseDeclaration>
  <outcomeDeclaration identifier="SCORE" cardinality="single" baseType="integer">
    <defaultValue>
      <value>0</value>
    </defaultValue>
  </outcomeDeclaration>
  <itemBody>
    <p>Look at the text in the picture.</p>
    <p>
      
    </p>
    <choiceInteraction responseIdentifier="RESPONSE" shuffle="false" maxChoices="1">
      <prompt>What does it say?</prompt>
      <simpleChoice identifier="ChoiceA">You must stay with your luggage at all times.</simpleChoice>
      <simpleChoice identifier="ChoiceB">Do not let someone else look after your luggage.</simpleChoice>
      <simpleChoice identifier="ChoiceC">Remember your luggage when you leave.</simpleChoice>
    </choiceInteraction>
  </itemBody>
  <responseProcessing template="http://www.imslobal.org/question/qti_v2p0/rptemplates/match_correct"/>
</assessmentItem>
```

Σχήμα 2.17 – QTI example 1 : XML αναπαράσταση του choiceInteraction

Ένα response processing template αναπαριστάται και αυτό ως XML document. Στο σχήμα 2.18 δίνουμε την XML αναπαράσταση του Match Correct.

```

<?xml version="1.0" encoding="UTF-8"?>
<responseProcessing xmlns="http://www.imsglobal.org/xsd/imsqti_v2p0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqti_v2p0 imsqti_v2p0.xsd">
  <responseCondition>
    <responseSelf>
      <match>
        <variable identifier="RESPONSE"/>
        <correct identifier="RESPONSE"/>
      </match>
      <setOutcomeValue identifier="SCORE">
        <baseValue baseType="integer">1</baseValue>
      </setOutcomeValue>
    </responseSelf>
    <responseElse>
      <setOutcomeValue identifier="SCORE">
        <baseValue baseType="integer">0</baseValue>
      </setOutcomeValue>
    </responseElse>
  </responseCondition>
</responseProcessing>

```

Σχήμα 2.18 – QTI example 1 : XML αναπαράσταση του Match Correct response processing template

Εκτός από το *choiceInteraction* υπάρχουν διάφορα άλλα είδη από interactions, π.χ. σε ένα *orderInteraction* ο εξεταζόμενος πρέπει να βάλει στη σωστή σειρά ένα σύνολο από αντικείμενα. Για παράδειγμα, στο σχήμα 2.19 φαίνεται ένα *orderInteraction* όπου ο εξεταζόμενος πρέπει να διατάξει τα ονόματα τριών οδηγών της φόρμουλας 1 σύμφωνα με τη σειρά που τερμάτισαν σε έναν αγώνα.

GRAND PRIX OF BAHRAIN

The following F1 drivers finished on the podium in the first ever Grand Prix of Bahrain. Can you rearrange them into the correct finishing order?

Rubens Barrichello
Jenson Button
Michael Schumacher

Σχήμα 2.19 – QTI example 2 : orderInteraction

Η XML αναπαράσταση του παραπάνω *orderInteraction* δίνεται στο σχήμα 2.20. Το response processing template που χρησιμοποιείται είναι και εδώ το Match Correct.


```

<?xml version="1.0" encoding="UTF-8"?>
<assessmentItem xmlns="http://www.imsglobal.org/xsd/imsqti_v2p0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqti_v2p0 imsqti_v2p0.xsd" identifier="order"
title="Grand Prix of Bahrain" adaptive="false" timeDependent="false">
  <responseDeclaration identifier="RESPONSE" cardinality="ordered" baseType="identifier">
    <correctResponse>
      <value>DriverC</value>
      <value>DriverA</value>
      <value>DriverB</value>
    </correctResponse>
  </responseDeclaration>
  <outcomeDeclaration identifier="SCORE" cardinality="single" baseType="integer"/>
  <itemBody>
    <orderInteraction responseIdentifier="RESPONSE" shuffle="true">
      <prompt>The following F1 drivers finished on the podium in the first ever Grand Prix of
      Bahrain. Can you rearrange them into the correct finishing order?</prompt>
      <simpleChoice identifier="DriverA">Rubens Barrichello</simpleChoice>
      <simpleChoice identifier="DriverB">Jenson Button</simpleChoice>
      <simpleChoice identifier="DriverC" fixed="true">Michael Schumacher</simpleChoice>
    </orderInteraction>
  </itemBody>
  <responseProcessing template="http://www.imsglobal.org/question/qti_v2p0/rptemplates/match_correct"/>
</assessmentItem>

```

Σχήμα 2.20 – QTI example 2 : XML αναπαράσταση του orderInteraction

Έτσι λοιπόν μια επέκταση του παρόντος συστήματος θα μπορούσε να είναι η υλοποίηση ενός QTI υποσυστήματος, το οποίο θα επεξεργάζεται τις XML αναπαραστάσεις των assessment items και θα τις μετασχηματίζει σε HTML σελίδες που θα προβάλλονται στον browser. Για την ολοκλήρωση του QTI με το Simple Sequencing, πρέπει να οριστεί ένα mapping των αποτελεσμάτων (*outcomes*) των assessment items προς τα δεδομένα του Tracking Model που ορίζει το Simple Sequencing. Σημειώνουμε ότι προς το παρόν ένα τέτοιο mapping δεν έχει ακόμα προταθεί από το IMS [QTI 2.0, 05c].

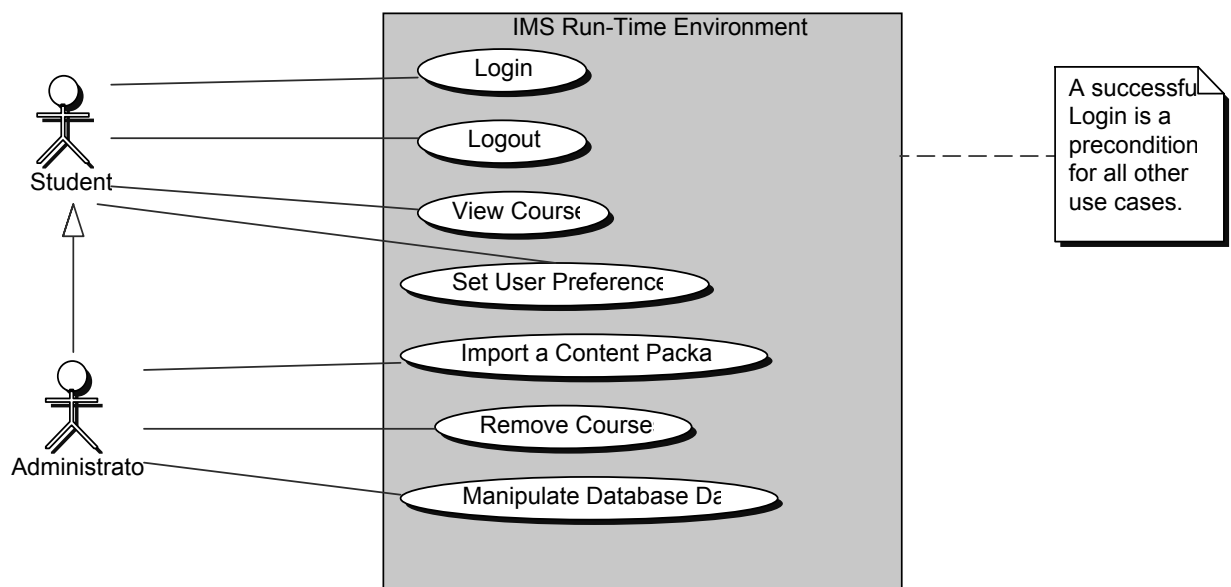
3 Τεκμηρίωση του συστήματος

Στο κεφάλαιο αυτό περιέχεται η τεκμηρίωση του συστήματος που υλοποιήθηκε. Η τεκμηρίωση αποτελείται από ένα σύνολο UML διαγραμμάτων και κειμένου. Η μορφή της τεκμηρίωσης και η ορολογία που χρησιμοποιείται είναι ένας συνδυασμός στοιχείων των μεθόδων ανάπτυξης λογισμικού UP (Unified Process) και YASM (Yet Another Software Development Methodology). Η UP περιγράφεται λεπτομερώς στο [JBR99], ενώ ένα καλό εισαγωγικό βιβλίο είναι το [KRUCHTEN99]. Η YASM παρουσιάζεται στο [ΣΚΟ03].

3.1 Ανάλυση

3.1.1 Ανάλυση Απαιτήσεων (Requirements Analysis)

Οι λειτουργικές απαιτήσεις που έχουμε από το σύστημα αποτυπώνονται στο παρακάτω *Use-Case diagram* (σχήμα 3.1) :



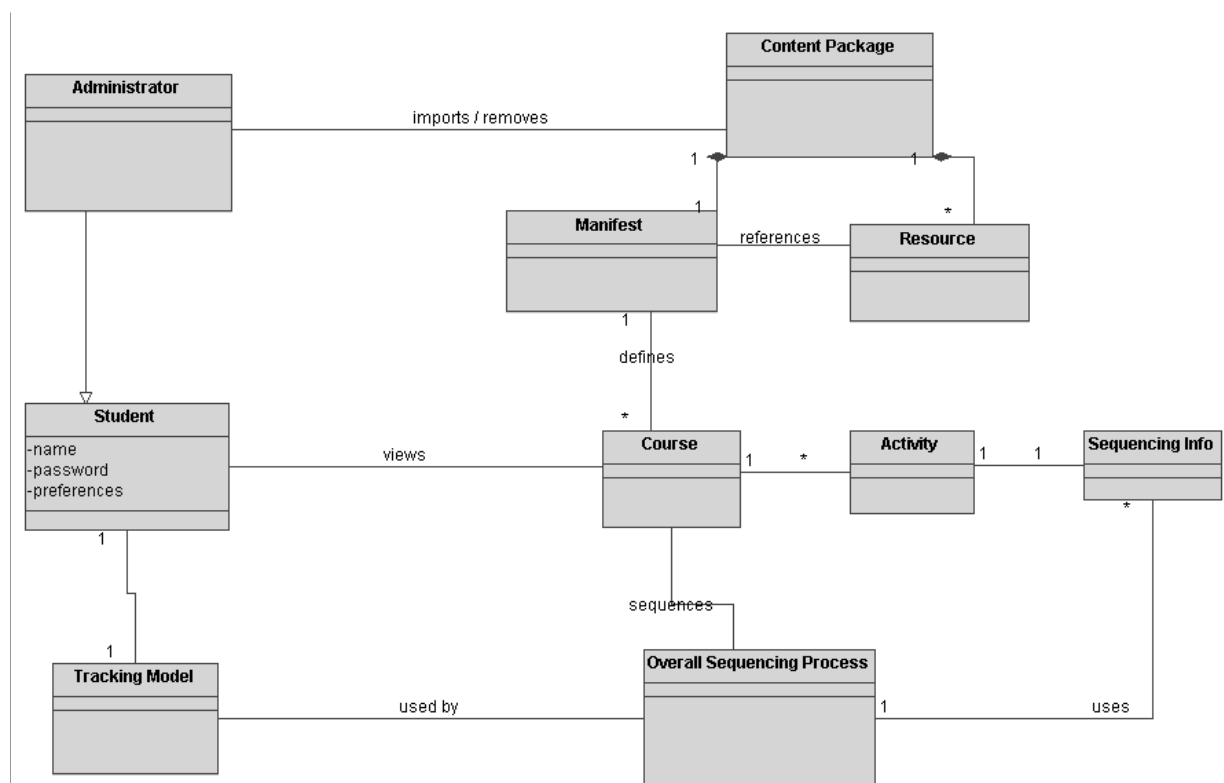
Σχήμα 3.1 Διάγραμμα Use-Case

Όπως φαίνεται στο σχήμα 3.1, υπάρχουν δύο τύποι (ρόλοι) χρηστών του συστήματος, ο *Μαθητής* (Student) και ο *Διαχειριστής* (Administrator). Ο Διαχειριστής είναι μια ειδική περίπτωση Μαθητή, γιατί έχει πρόσβαση σε όλες τις λειτουργίες που έχει πρόσβαση ο Μαθητής, αλλά επιπλέον και σε διαχειριστικές λειτουργίες όπως είναι η εισαγωγή ενός Content Package στο σύστημα.

Το use case “Login” είναι χαμηλότερου επιπέδου από τα υπόλοιπα (*fish level* κατά το [COCKBURN01]) και περιλήφθηκε στο διάγραμμα μόνο επειδή περιέχεται σε όλα τα υπόλοιπα use cases (included use case). Όμως χάριν απλότητας δεν έχουν σχεδιαστεί τα «include» relationships. Το use case “Logout” είναι επίσης χαμηλού επιπέδου (*fish level*) και περιλήφθηκε στο διάγραμμα μόνο χάριν συμμετρίας με το “Login”.

3.1.2 Domain (Conceptual) Model

Οι βασικές έννοιες με τις οποίες έχει να κάνει το σύστημα απεικονίζονται στο *Domain Model* (γνωστό και ως *Conceptual Model*) που φαίνεται στο σχήμα 3.2. Τονίζεται ότι ένα Domain Model είναι απλά μια γραφική αναπαράσταση των βασικών εννοιών (*conceptual classes* στην ορολογία της UP) ενός domain και των μεταξύ τους σχέσεων. ΔΕΝ αναπαριστά software components, αλλά βεβαίως συμβάλλει στο σχεδιασμό του software.



Σχήμα 3.2 Domain Model

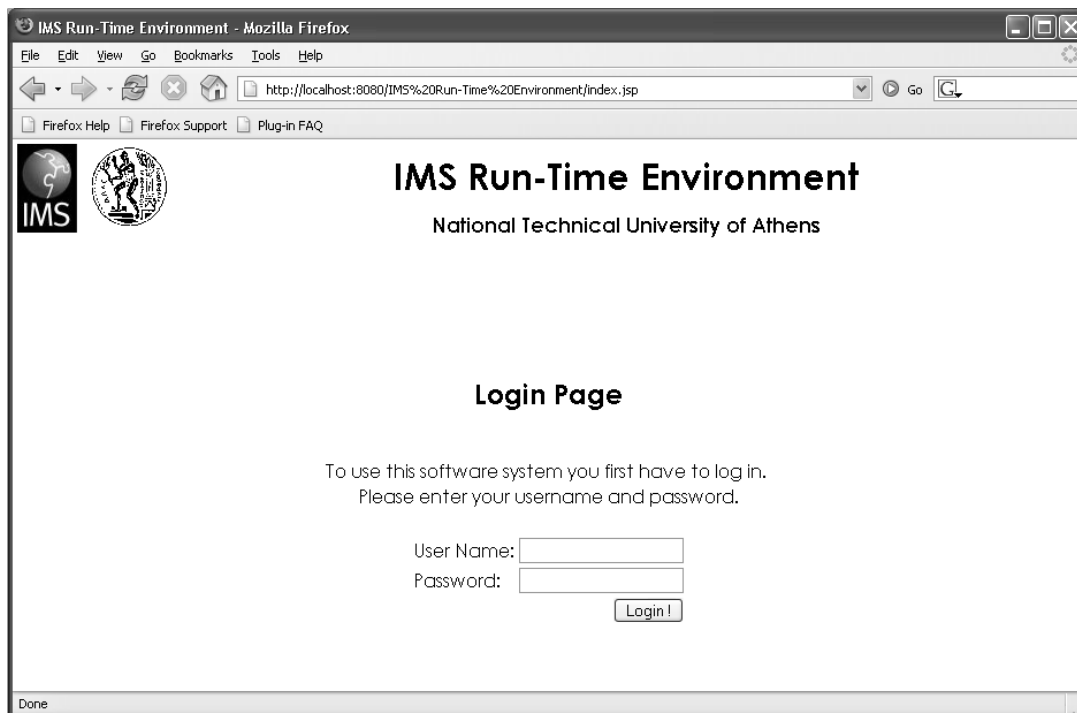
3.2 Σχεδιασμός

3.2.1 Γενικά

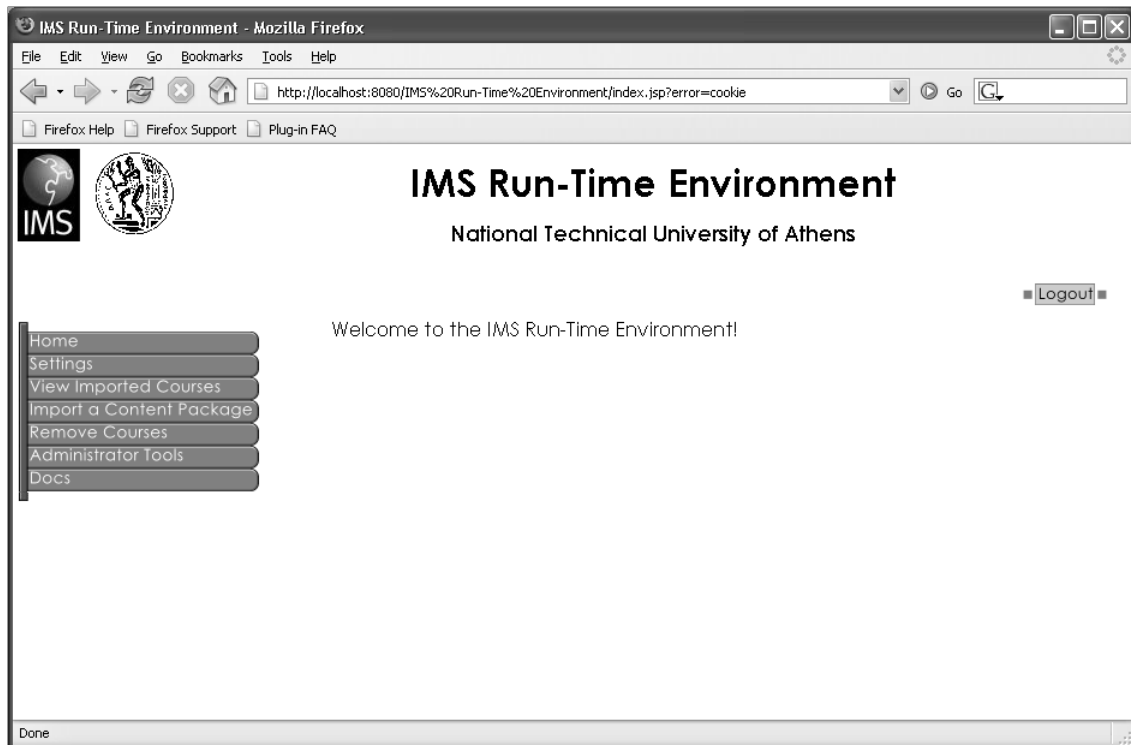
Το σύστημα έχει υλοποιηθεί ως μια δικτυακή εφαρμογή (web application) σε Java (πλατφόρμα J2EE), η οποία τρέχει μέσα σε έναν web container όπως είναι π.χ. ο Apache Tomcat. Ο χρήστης του συστήματος έχει πρόσβαση σε αυτό μέσω κάποιου browser. Η πλατφόρμα J2EE και οι υπόλοιπες τεχνολογίες που χρησιμοποιήθηκαν παρουσιάζονται στο Παράρτημα Β.

3.2.2 Χρηστική όψη

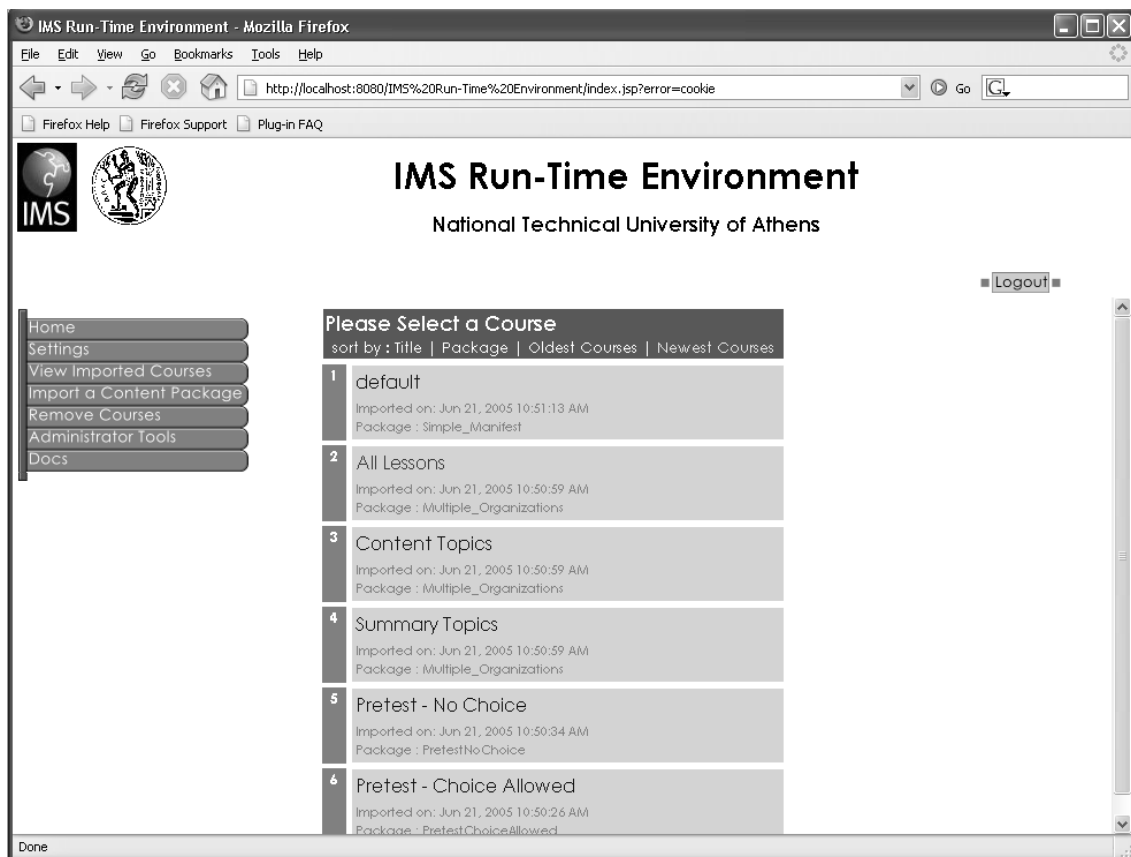
Σε αυτήν την όψη φαίνονται οι σελίδες που αποτελούν τη γραφηματική διαπροσωπεία (graphical user interface) του συστήματος. Ουσιαστικά κάθε use case αντιστοιχεί και σε μια σελίδα ή ομάδα από σελίδες. Σημειώνουμε ότι οι χρήστες με ρόλο *Μαθητής* έχουν πρόσβαση μόνο στις σελίδες Login, Home, View Courses, View Course και User Preferences, ενώ οι χρήστες με ρόλο *Διαχειριστής* έχουν πρόσβαση σε όλες τις σελίδες.



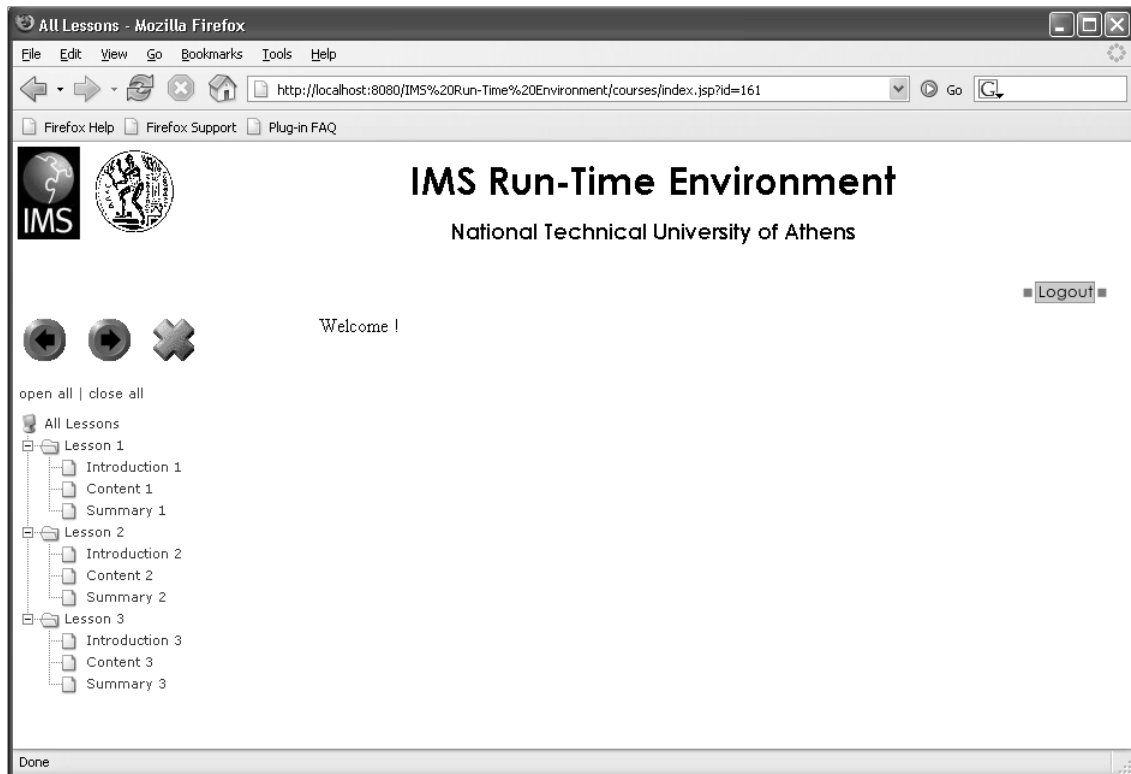
Σχήμα 3.3 – Login



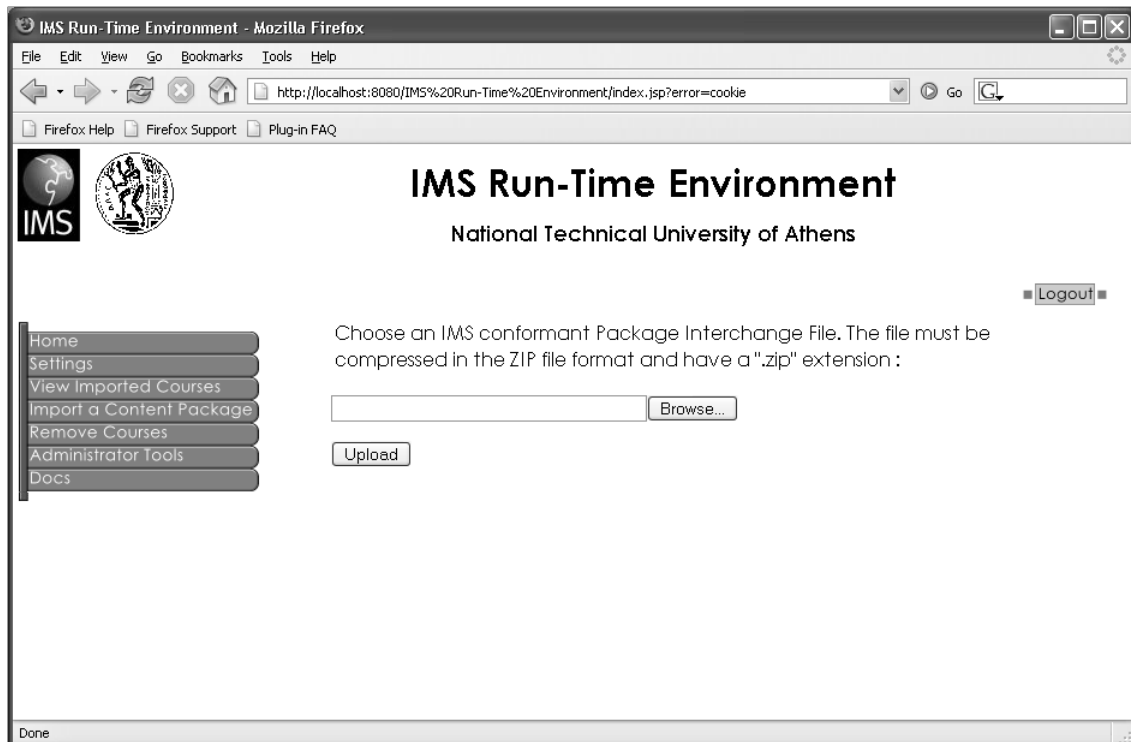
Σχήμα 3.4 – Home (Αρχική Σελίδα)



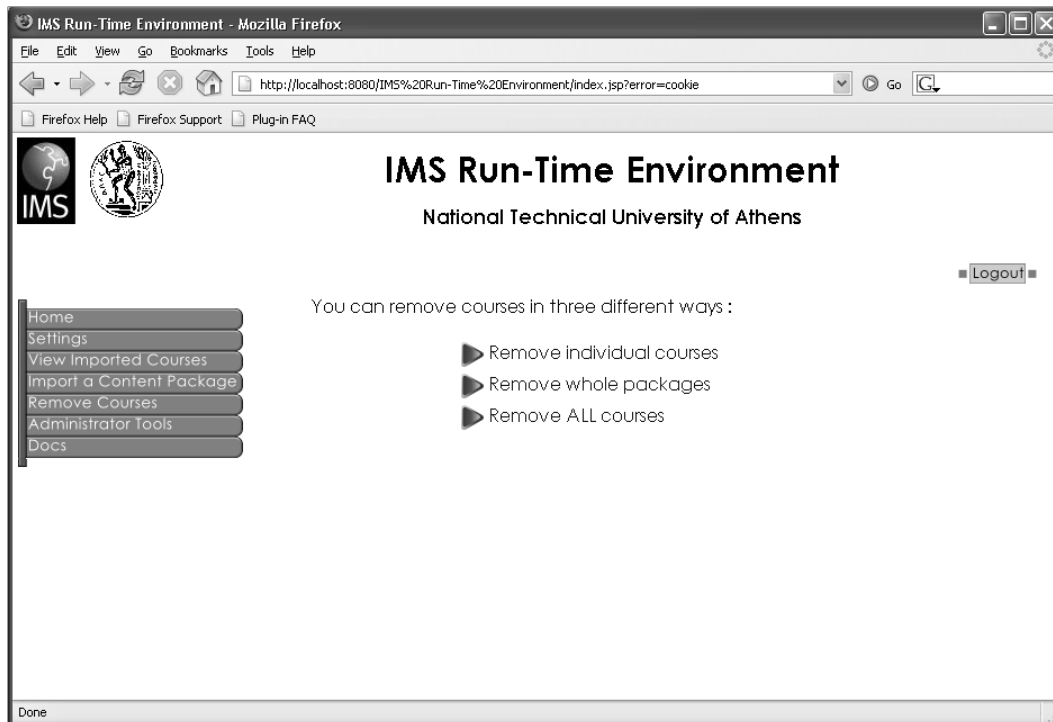
Σχήμα 3.5 – View Courses (λίστα όλων των διαθέσιμων courses)



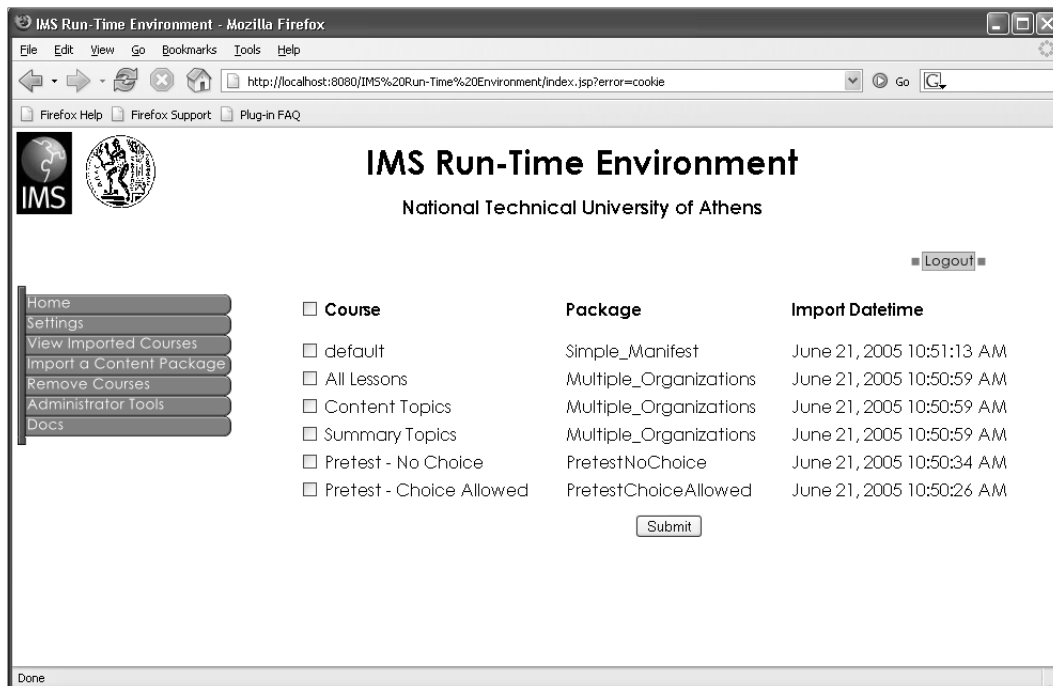
Σχήμα 3.6 – View Course (παρουσίαση συγκεκριμένου course)



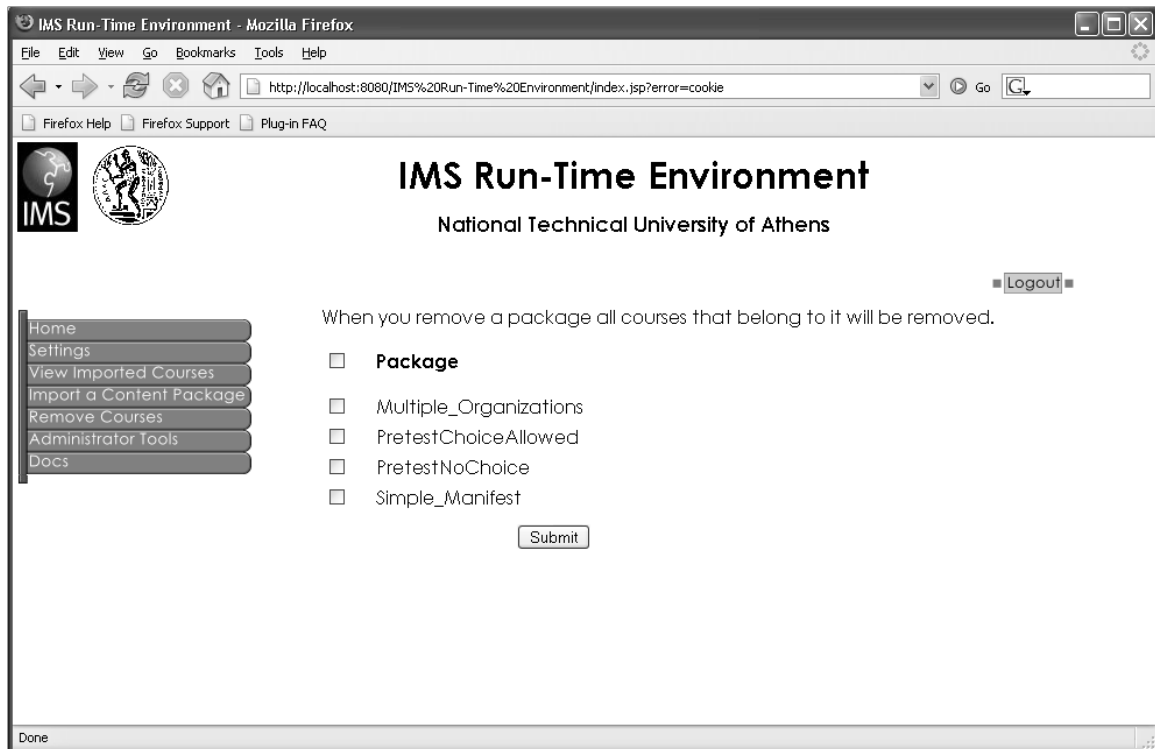
Σχήμα 3.7 – Import Package (εισαγωγή νέου Package στο σύστημα)



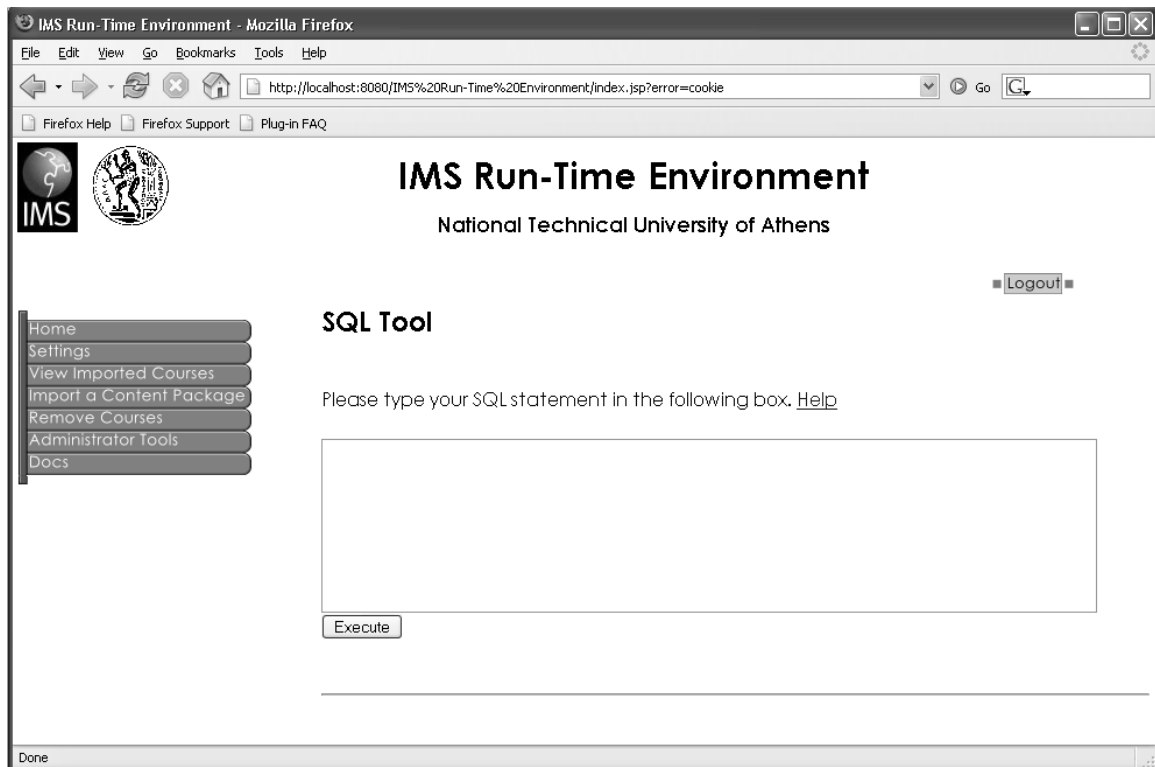
Σχήμα 3.8 – Remove Courses (Κεντρική σελίδα – 3 τρόποι διαγραφής courses)



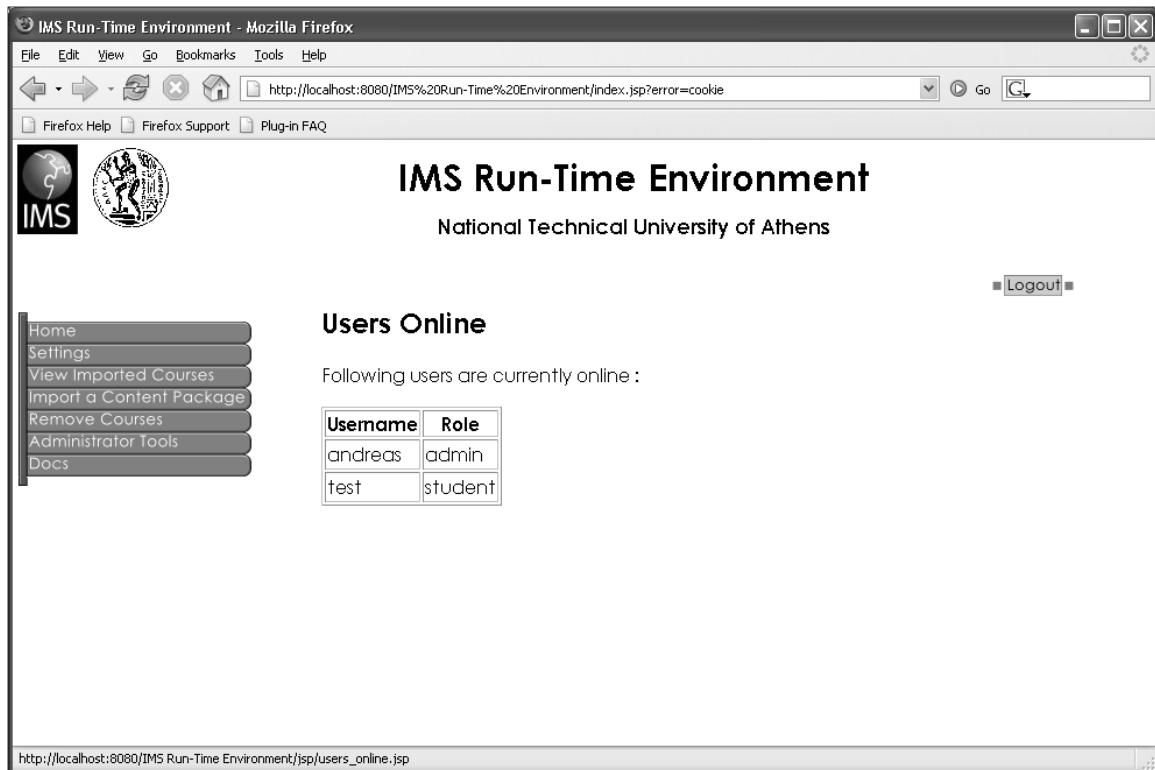
Σχήμα 3.9 – Remove Courses (Επιλογή 1 - Διαγραφή μεμονωμένων courses)



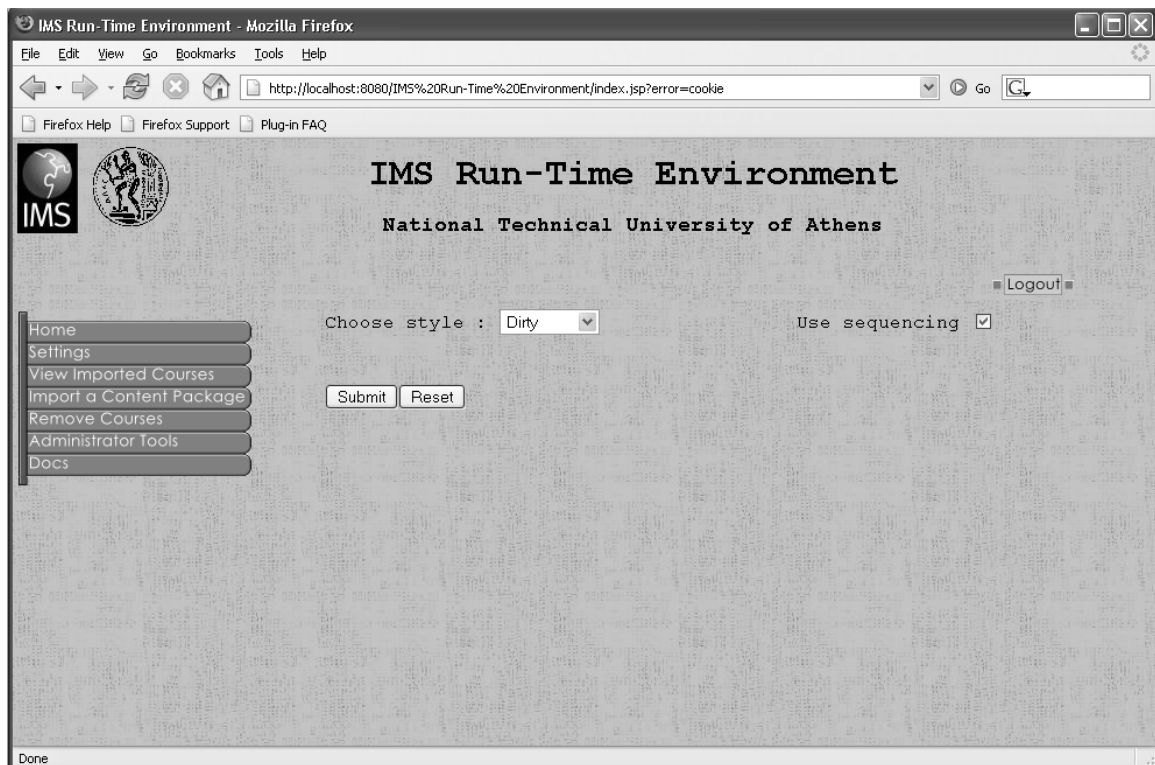
Σχήμα 3.10 – Remove Courses (Επιλογή 2 - Διαγραφή ολόκληρων packages)



Σχήμα 3.11 – SQL Tool (Άμεση πρόσβαση στη Βάση Δεδομένων στέλνοντας SQL εντολές μέσω του browser)



Σχήμα 3.12 – Users Online (λίστα όλων των χρηστών που είναι logged in)



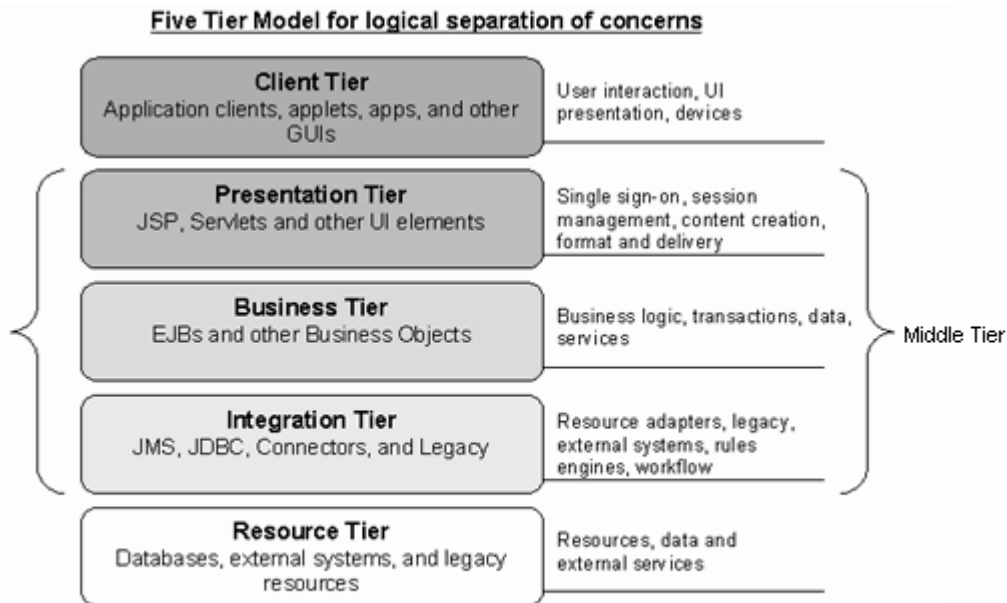
Σχήμα 3.13 – Προτιμήσεις Χρήστη (Στυλ Εμφάνισης, Χρήση Sequencing)

3.2.3 Αρχιτεκτονική Σχεδίαση

Η γενική αρχιτεκτονική ενός web (enterprise) application μπορεί να θεωρηθεί ότι αποτελείται από διάφορα *tiers* (στρώματα). Συγκεκριμένα το J2EE ορίζει τα παρακάτω tiers [J2EE DESIGN] :

- Client Tier : Το μέρος του συστήματος που εκτελείται στην πλευρά του χρήστη (client). Αυτό μπορεί να περιλαμβάνει απλές HTML σελίδες, HTML σελίδες παραγόμενες δυναμικά μέσω π.χ. της τεχνολογίας JSP, Java applets ή ακόμα και stand alone εφαρμογές.
- Middle Tier : Το μέρος του συστήματος που εκτελείται στην πλευρά του server (σε έναν ή περισσότερους servers). Είναι συνήθως το κύριο μέρος του συστήματος και χωρίζεται στα εξής επιμέρους tiers :
 - Presentation Tier : Το τμήμα του middle tier που χειρίζεται την αλληλεπίδραση του χρήστη με το σύστημα. Συγκεκριμένα το τμήμα αυτό είναι υπεύθυνο για την λήψη και το χειρισμό των requests του client (π.χ. HTTP requests στην περίπτωση που ο client είναι ένας browser), την δημιουργία (creation), μορφοποίηση (formatting) και παράδοση (delivery) της κατάλληλης πληροφορίας ως απόκριση στα requests αυτά, και το session management.
 - Business Tier : Περιέχει το *business logic* (γνωστό και ως *domain logic*), δηλαδή τις λειτουργίες που αφορούν το συγκεκριμένο domain στο οποίο ανήκει το σύστημα.
 - Integration Tier : Το τμήμα του middle tier που παρέχει πρόσβαση στο resource tier.
- Resource (ή Back-End) Tier : Εξωτερικά συστήματα (π.χ. Βάσεις Δεδομένων, Enterprise Information Systems (EIS), legacy συστήματα κ.α.), άλλου είδους data sources.

Τα παραπάνω tiers και τα συστατικά τους (στην περίπτωση του J2EE) φαίνονται στο σχήμα 3.14, το οποίο έχουμε δανειστεί από το [ACM03].



Σχήμα 3.14 J2EE Application Tiers

Τονίζουμε ότι το συγκεκριμένο μοντέλο διαστρωμάτωσης είναι ένας γενικός τρόπος θεώρησης που μπορεί να εφαρμοστεί στις περισσότερες σύγχρονες δικτυακές εφαρμογές τύπου client – server (και γενικά σε enterprise εφαρμογές) και δεν αφορά αποκλειστικά το J2EE ή το σύστημα που υλοποιήσαμε.

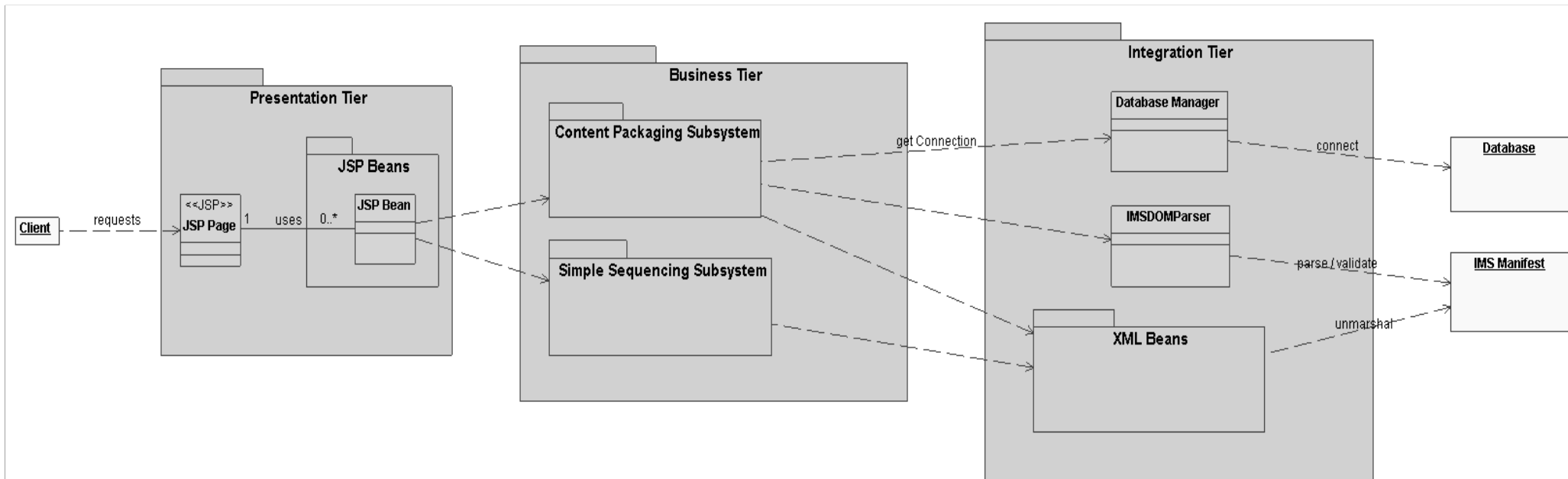
Στη συνέχεια θα περιγραφεί η αρχιτεκτονική του συστήματος, ξεχωριστά για κάθε tier.

Client Tier

Client του συστήματος είναι ο web browser που χρησιμοποιεί ο χρήστης. Ο browser προβάλλει στο χρήστη τις HTML σελίδες που παρουσιάστηκαν στην παράγραφο 3.2.2. Οι σελίδες αυτές παράγονται δυναμικά στο middle tier με χρήση της τεχνολογίας JSP. Δεν χρησιμοποιούνται applets. Ο μόνος client-side κώδικας είναι JavaScript scripts που είναι ενσωματωμένα στις HTML σελίδες. Αυτά περιλαμβάνουν τον κώδικα του δέντρου (tree menu) με το οποίο αναπαριστώνται τα περιεχόμενα ενός course (βλ. σχήμα 3.6 - σελίδα View Course) και ο οποίος είναι ένα έτοιμο component (υπάρχουν πολλά freeware JavaScript trees διαθέσιμα στο Internet) και κάποια στοιχειώδη scripts π.χ. για form validation ή redirection.

Middle Tier

Μια γενική εικόνα της αρχιτεκτονικής του middle tier δίνεται στο σχήμα 3.15.



Σχήμα 3.15 Middle Tier Architecture

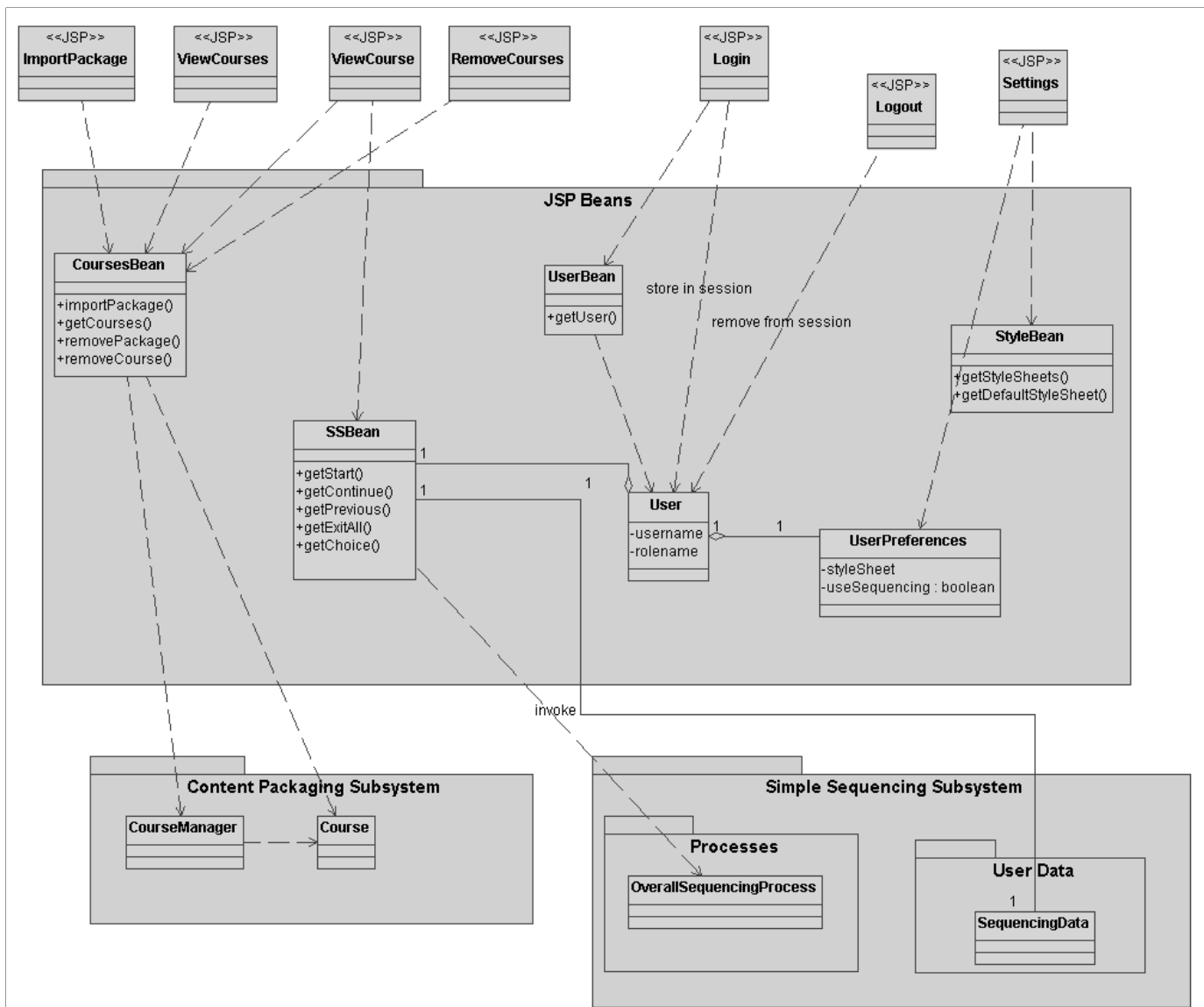
Στη συνέχεια επεξηγούνται τα επιμέρους tiers που αποτελούν το middle tier.

A. **Presentation Tier**

Το presentation tier περιλαμβάνει JSP σελίδες και JavaBean κλάσεις που χρησιμοποιούνται από τις JSP σελίδες (γνωστές και ως JSP Beans). Οι JSP σελίδες περιέχουν το *presentation logic*, δηλαδή λογική που έχει άμεση σχέση με την παρουσίαση της πληροφορίας, ενώ τα JSP Beans περιέχουν το *processing logic*, δηλαδή την ανάκτηση (data retrieval) και επεξεργασία (processing) της πληροφορίας. Με τον τρόπο αυτό επιτυγχάνεται ο διαχωρισμός του κώδικα από την παρουσίαση, το οποίο αποτελεί σημαντικό σχεδιαστικό στόχο γιατί συμβάλλει στην επαναχρησιμοποίηση (reuse) του κώδικα, την εύκολη συντήρηση (maintenance) του συστήματος και στο διαχωρισμό και την ελάττωση της συσχέτισης (coupling) των ρόλων μέσα στην ομάδα που αναπτύσσει το σύστημα (π.χ. ο web designer γράφει τον HTML κώδικα ενώ ο προγραμματιστής τον Java κώδικα ανεξάρτητα ο ένας από τον άλλο). Κάποια JSP Beans εκπληρώνουν τον ρόλο τους καλώντας (delegation) μεθόδους των business objects που βρίσκονται στο Business Tier (ουσιαστικά λειτουργούν ως adapters του model στο view), ενώ άλλα είναι αυτόνομα (π.χ. αυτά που αφορούν το user management).

Η παραπάνω τεχνική είναι πολύ συνηθισμένη στις σύγχρονες δικτυακές εφαρμογές και είναι μια ειδική περίπτωση του *View Helper* design pattern (βλ. [ACM03]). Στην περίπτωσή μας το ρόλο του View παίζουν οι JSP σελίδες ενώ το ρόλο των Helpers παίζουν τα JSP Beans (*JavaBean Helper Strategy*).

Στο σχήμα 3.16 φαίνονται οι JSP σελίδες και τα JSP Beans που χρησιμοποιούν.



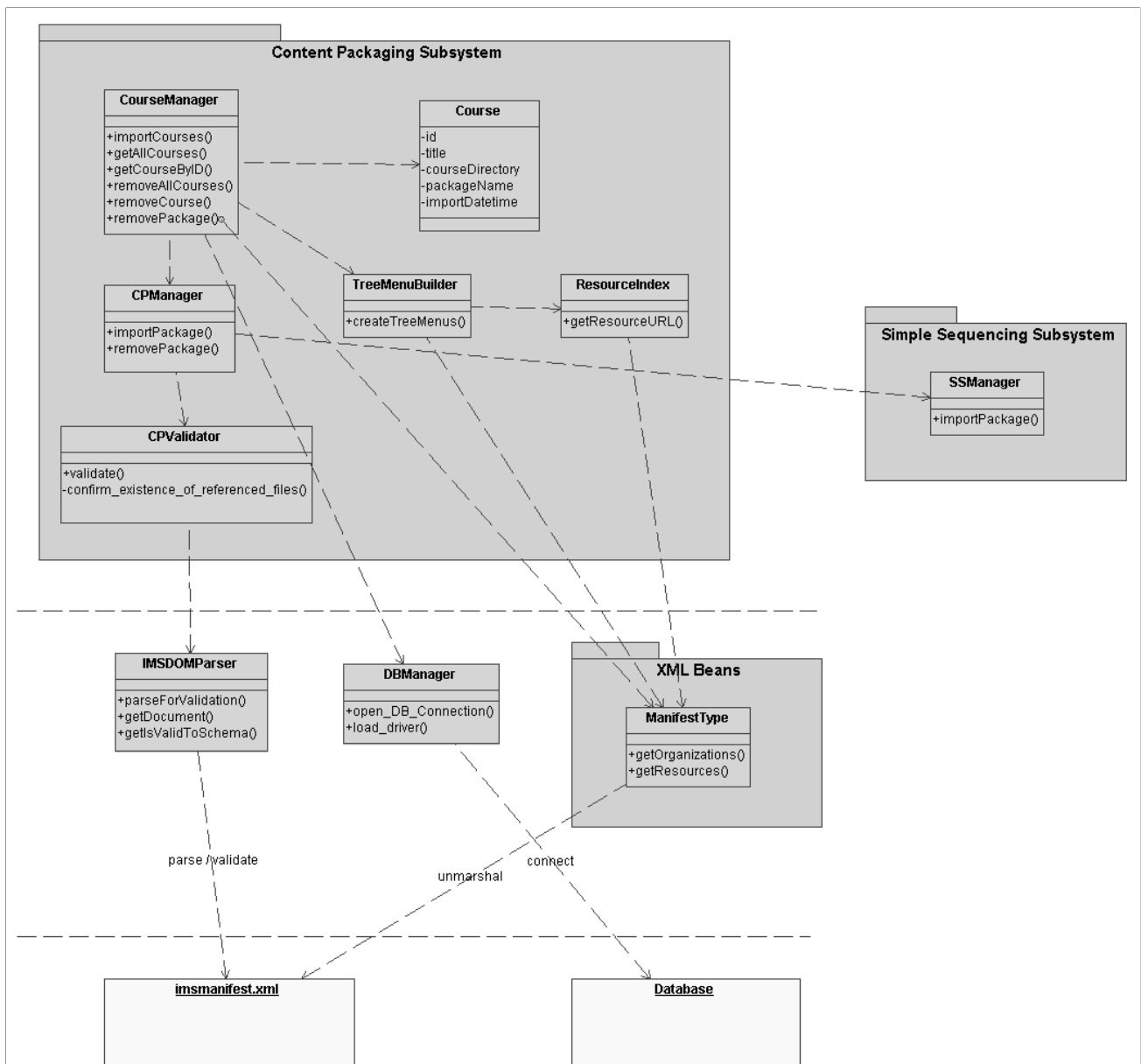
Σχήμα 3.16 Presentation Tier Architecture

B. Business Tier

Το Business Tier αποτελείται από το υποσύστημα Content Packaging και το υποσύστημα Simple Sequencing.

Content Packaging Subsystem

Το υποσύστημα Content Packaging φαίνεται στο σχήμα 3.17.



Σχήμα 3.17 Business Tier : Content Packaging Subsystem

Όπως φαίνεται στο σχήμα, οι κλάσεις CourseManager, CPManger, CPValidator και IMSDOMParser ακολουθούν το αρχιτεκτονικό pattern των layers (στρώματα), όπου κάθε layer είναι υπεύθυνο για μια σειρά από λειτουργίες τις οποίες υλοποιεί χρησιμοποιώντας τις λειτουργίες που του παρέχει το αμέσως χαμηλότερο layer. Η ανάθεση των ευθυνών σε κάθε layer είναι η εξής :

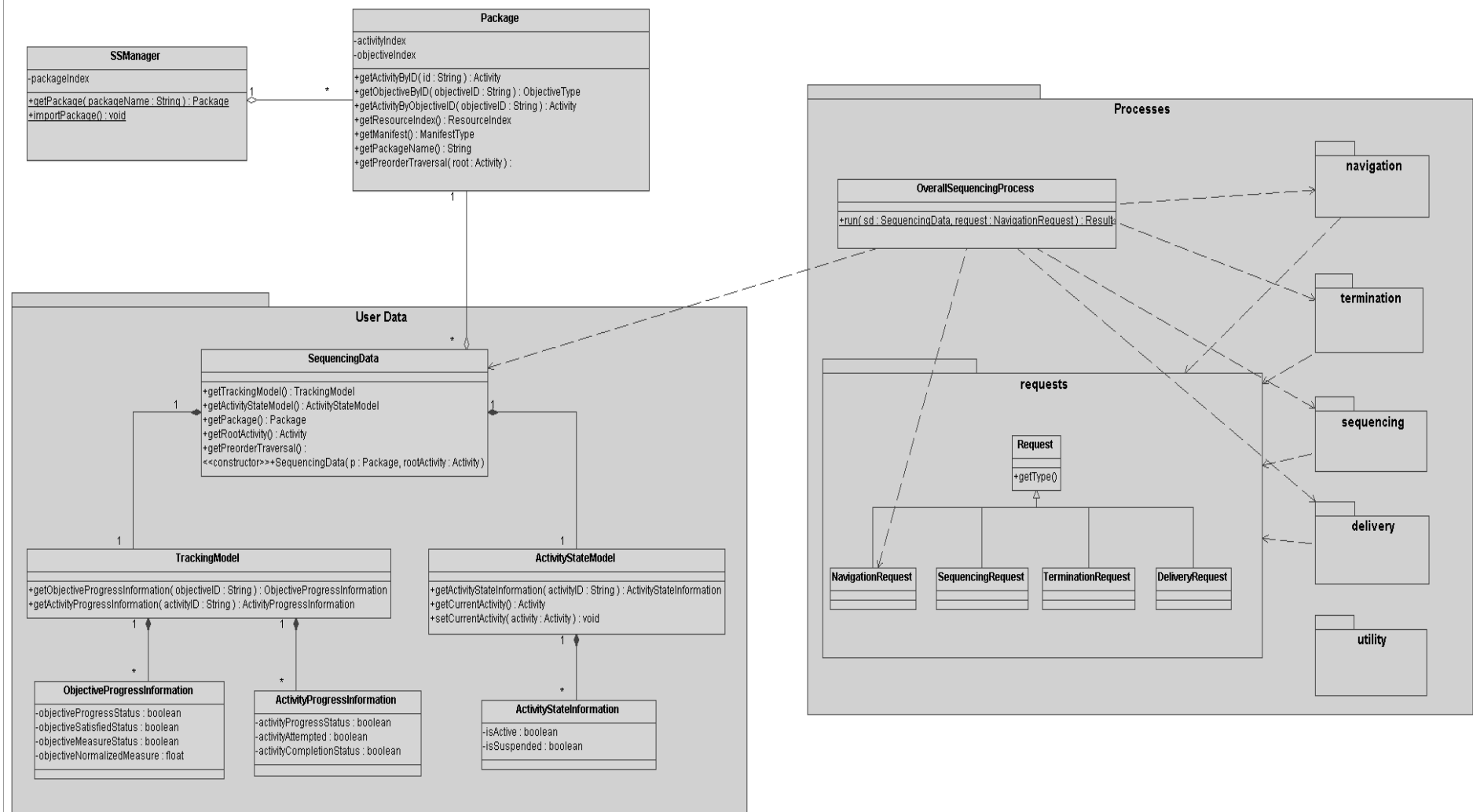
- CourseManager : Είναι το ανώτερο layer και είναι υπεύθυνο για λειτουργίες σε επίπεδο course, όπου με τον όρο course εννοούμε το μάθημα που αντιστοιχεί σε ένα <organization> που περιέχεται στο IMS Manifest (αρχείο imsmanifest.xml). Το IMS Manifest μπορεί να ορίζει πολλά εναλλακτικά <organization>s, επομένως σε κάθε Package θα αντιστοιχούν ένα ή περισσότερα courses. Κατά την εισαγωγή ενός νέου Package στο σύστημα, για κάθε course που περιέχει το Package ο CourseManager εισάγει και ένα αντίστοιχο record στον πίνακα course της Βάσης Δεδομένων (χρησιμοποιεί τον DBManager στο integration tier για

να συνδεθεί στη βάση). Επίσης χρησιμοποιεί την κλάση TreeMenuBuilder για τη δημιουργία των JavaScript tree menus με τα παρουσιάζεται η δομή του course στον browser.

- CPManger : Είναι υπεύθυνος για λειτουργίες σε επίπεδο Package. Χρησιμοποιείται από τον CourseManager κατά την εισαγωγή ή διαγραφή κάποιου Package. Στην πρώτη περίπτωση αποσυμπιέζει το Package Interchange File σε κάποιο subdirectory του directory content_packages και καλεί τον CPValidator για τον έλεγχο της εγκυρότητας του Package, ενώ στη δεύτερη περίπτωση διαγράφει το subdirectory που βρίσκεται το Package.
- CPValidator : Ελέγχει την εγκυρότητα του Package. Για να είναι έγκυρο ένα Package πρέπει να ισχύουν τα εξής (βλ. [CP, 03c] §6.1: Package Conformance) :
 1. Να περιέχει ένα αρχείο με όνομα imsmanifest.xml στο root του Package.
 2. Το imsmanifest.xml να περιέχει well-formed XML που να είναι valid προς το XML Schema του Content Packaging (imscp_v1p1.xsd). Μπορεί επίσης να περιέχει και elements από άλλα namespaces (π.χ. IMS Simple Sequencing, IMS Meta Data), υπό την προϋπόθεση ότι τα αντίστοιχα schemas δηλώνονται και είναι valid ως προς αυτά.
 3. Όλα τα αρχεία στα οποία αναφέρεται ένα resource πρέπει να δηλώνονται μέσα σε <file> elements μέσα στο <resources> element του imsmanifest.xml και όσα είναι τοπικά (δηλαδή το URL τους είναι relative) να υπάρχουν μέσα στο directory του Package.Για τον έλεγχο της εγκυρότητας του imsmanifest.xml ως προς το XML Schema (συνθήκη 2) ο CPValidator χρησιμοποιεί τον IMSDOMParser.
- IMSDOMParser : Υπεύθυνος για το parsing και το schema validation του imsmanifest.xml.

Simple Sequencing Subsystem

Το υποσύστημα Simple Sequencing φαίνεται στο σχήμα 3.18.



Σχήμα 3.18 Business Tier : Simple Sequencing Subsystem

Η κλάση Package είναι ένας wrapper του manifest (κλάση ManifestType στον generated κώδικα των XML Beans), η οποία επιπλέον παρέχει χρήσιμες λειτουργίες όπως :

- getActivityByID() - αναζήτηση ενός activity με βάση το activity ID
- getObjectiveByID() - αναζήτηση ενός objective με βάση το objective ID
- getActivityByObjectiveID – αναζήτηση του activity στο οποίο ανήκει το objective με το δοθέν objective ID
- getPreorderTraversal – επιστρέφει μια array με όλα τα activities του activity tree ταξινομημένα σύμφωνα με την preorder διάσχιση του δέντρου

Οι τρεις πρώτες λειτουργίες υλοποιούνται με χρήση της java.util.HashMap για τη δημιουργία ενός activityIndex και ενός objectiveIndex.

Η κλάση SSManager είναι ο διαχειριστής των instances της κλάσης Package. Συγκεκριμένα, ένα τέτοιο instance δημιουργείται όταν ένα content package εισάγεται στο σύστημα αλλά για εξοικονόμηση μνήμης το αντικείμενο αποθηκεύεται στο δίσκο (με serialization) και όχι στη μνήμη. Όταν ένας μαθητής θελήσει να παρακολουθήσει κάποιο course, τότε το Package instance που περιέχει το course φορτώνεται στη μνήμη (μέθοδος getPackage()), δηλαδή διαβάζεται από το δίσκο και τοποθετείται σε ένα packageIndex. Ένα instance μπορεί να χρησιμοποιείται από πολλούς μαθητές (χρήστες) ταυτόχρονα. Για κάθε instance, ο SSManager κρατά τον αριθμό των χρηστών που το χρησιμοποιούν τη συγκεκριμένη χρονική στιγμή (μέθοδοι incrementUsers() και decrementUsers()). Όταν ο αριθμός αυτός γίνει μηδέν, δηλαδή όταν και ο τελευταίος χρήστης του instance πάψει να το χρησιμοποιεί, τότε για εξοικονόμηση μνήμης το instance διαγράφεται από το packageIndex.

Η κλάση SequencingData ενθυλακώνει όλη τη sequencing πληροφορία που αφορά ένα συγκεκριμένο χρήστη. Συγκεκριμένα περιέχει :

- το Package instance που χρησιμοποιεί ο χρήστης
- το root activity του course που παρακολουθεί ο χρήστης
- το TrackingModel
- το ActivityStateModel

Τέλος, έχουμε τα processes του Simple Sequencing, με κεντρικό process το OverallSequencingProcess. Εδώ η τακτική που ακολουθήθηκε είναι κάθε process που ορίζεται στο specification (βλ. [SS, 03a]) να αποτελεί και μια ξεχωριστή κλάση. Τα processes ομαδοποιούνται στα packages navigation, termination, sequencing, delivery και utility, όπως ακριβώς είναι ομαδοποιημένα και στα διάφορα τμήματα του specification. Στο σχήμα 3.18 για λόγους απλότητας φαίνεται μόνο το OverallSequencingProcess και τα packages που περιέχουν τα υπόλοιπα processes.

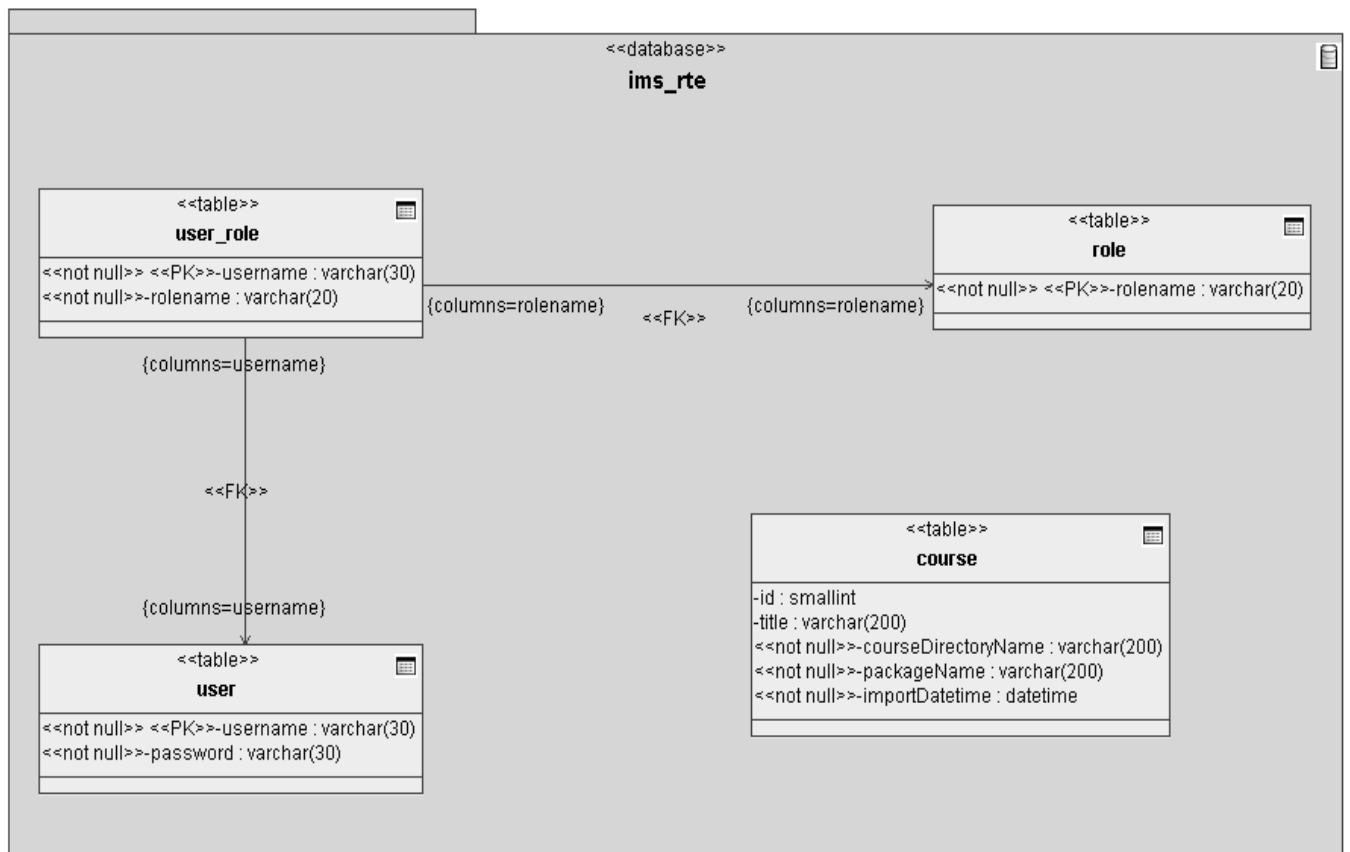
C. Integration Tier

Στο integration tier υπάρχει η κλάση DBManager που είναι υπεύθυνη για τη δημιουργία συνδέσεων με τη βάση δεδομένων, η κλάση IMSDOMParser που είναι υπεύθυνη για το parsing και το schema validation του imsmanifest.xml, και τέλος ο κώδικας που έχει παραχθεί αυτόματα από τον schema compiler των XML

Beans (βλ. Παράρτημα Β) και με τον οποίο γίνεται το data binding του imsmanifest.xml σε Java κλάσεις. Επειδή ο κώδικας αυτός παράγεται αυτόματα, τον αντιμετωπίζουμε σαν “μαύρο κουτί”, δηλαδή χρησιμοποιούμε τα interfaces που μας παρέχει χωρίς να μας ενδιαφέρει το πώς αυτά υλοποιούνται.

Resource Tier

Στο resource tier υπάρχει η βάση δεδομένων. Το σχήμα της βάσης δίνεται στο σχήμα 3.19.



Σχήμα 3.19 Βάση Δεδομένων `ims_rte`

Ο πίνακας `user` περιέχει το `username` και το `password` κάθε χρήστη. Κάθε χρήστης έχει ακριβώς ένα ρόλο. Ο πίνακας `role` περιέχει τους ρόλους, ενώ ο πίνακας `user_role` αντιστοιχίζει κάθε χρήστη σε ένα ρόλο. Στο σύστημα έχουν οριστεί δύο ρόλοι, ο ρόλος `student` και ο ρόλος `administrator`. Σε κάποια μελλοντική επέκταση του συστήματος μπορούν να προστεθούν και άλλοι ρόλοι στον πίνακα `roles`. Τέλος, ο πίνακας `course` αποθηκεύει για κάθε `course` το όνομα του `directory` που βρίσκεται το `package` στο οποίο ανήκει το `course`, το όνομα του `directory` που βρίσκεται το `tree menu` του `course` και η ημερομηνία εισαγωγής του `package` στο σύστημα.

3.3 Υλοποίηση

3.3.1 Λεπτομέρειες Υλοποίησης

3.3.1.1 Πλατφόρμα και Τεχνολογίες

Όπως έχει ήδη αναφερθεί, η πλατφόρμα στην οποία αναπτύχθηκε το σύστημα IMS Run-Time Environment είναι το J2EE της Sun Microsystems, και συγκεκριμένα οι τεχνολογίες *JavaServer Pages (JSP)* και *JavaServer Pages Standard Tag Library (JSTL)*. Επίσης ως XML data binding framework χρησιμοποιήθηκε το *XML Beans* της Apache Software Foundation. Οι παραπάνω τεχνολογίες παρουσιάζονται στο Παράρτημα Β.

Ως DBMS επιλέχτηκε η MySQL. Ο λόγος είναι ότι η MySQL είναι λογισμικό ανοιχτού κώδικα (open source) και κατά συνέπεια είναι διαθέσιμη δωρεάν, ενώ ταυτόχρονα έχει υψηλή απόδοση και αξιοπιστία (είναι ένας από τους δημοφιλέστερους database servers).

Για το JavaScript tree menu με το οποίο παρουσιάζονται τα περιεχόμενα ενός course στον browser χρησιμοποιήθηκε το dTree 2.05, το οποίο είναι διαθέσιμο δωρεάν στη διεύθυνση www.destroydrop.com/javascript/tree/.

Για το upload του Package Interchange File κατά την εισαγωγή ενός νέου content package στο σύστημα χρησιμοποιήθηκε το πακέτο *org.apache.commons.fileupload* της Apache Software Foundation.

3.3.1.2 Υλοποίηση των specifications

Το IMS Content Packaging Specification έχει υλοποιηθεί πλήρως.

Από το IMS Simple Sequencing Specification δεν έχουν υλοποιηθεί τα Selection and Randomization processes (βλ. [SS, 03a] § SR : *Selection and Randomization Behavior Model*), τα selection και randomization controls ((βλ. [SS, 03a] § SM.9 : *Selection Controls* και § SM.10 : *Randomization Controls*) και τα limit conditions (βλ. [SS, 03a] § SM.3 : *Limit Conditions Description*).

3.3.1.3 Περιβάλλον Ανάπτυξης

Ως περιβάλλον ανάπτυξης (Integrated Development Environment – IDE) χρησιμοποιήθηκε ο JBuilder 2005 της Borland. Το σύστημα ελέγχτηκε χρησιμοποιώντας τον servlet container Apache Tomcat 5.5.9 και τους browsers Internet Explorer και Mozilla Firefox. Τα UML διαγράμματα που περιλαμβάνονται στην παρούσα αναφορά σχεδιάστηκαν με τα προγράμματα MagicDraw UML 9.5 και Borland Together Architect.

3.3.2 System Requirements

Για το τρέξιμο του IMS Run-Time Environment είναι αναγκαία τα ακόλουθα :

Στον server :

a. Java(TM) 2 Runtime Environment Version 5.0 or later

Το IMS Run-Time Environment ΔΕΝ λειτουργεί με παλαιότερες εκδόσεις της Java.

b. MySQL database server version 4.1.0 or later

Το IMS Run-Time Environment έχει ελεγχτεί με τη MySQL 4.1.7 αλλά μπορεί να δουλέψει και με οποιαδήποτε άλλη 4.1.x έκδοση.

c. Ένας servlet container που υποστηρίζει τα Servlet 2.4 και JSP 2.0 APIs

Το IMS Run-Time Environment έχει ελεγχτεί με τον Apache Tomcat 5.0.24 και τον Apache Tomcat 5.5.4 αλλά κατά πάσα πιθανότητα μπορεί να δουλέψει και με άλλους servlet containers όπως ο Resin κτλ. Όμως σίγουρα ΔΕΝ δουλεύει με containers οι οποίοι δεν υποστηρίζουν JSP 2.0.

Στον client :

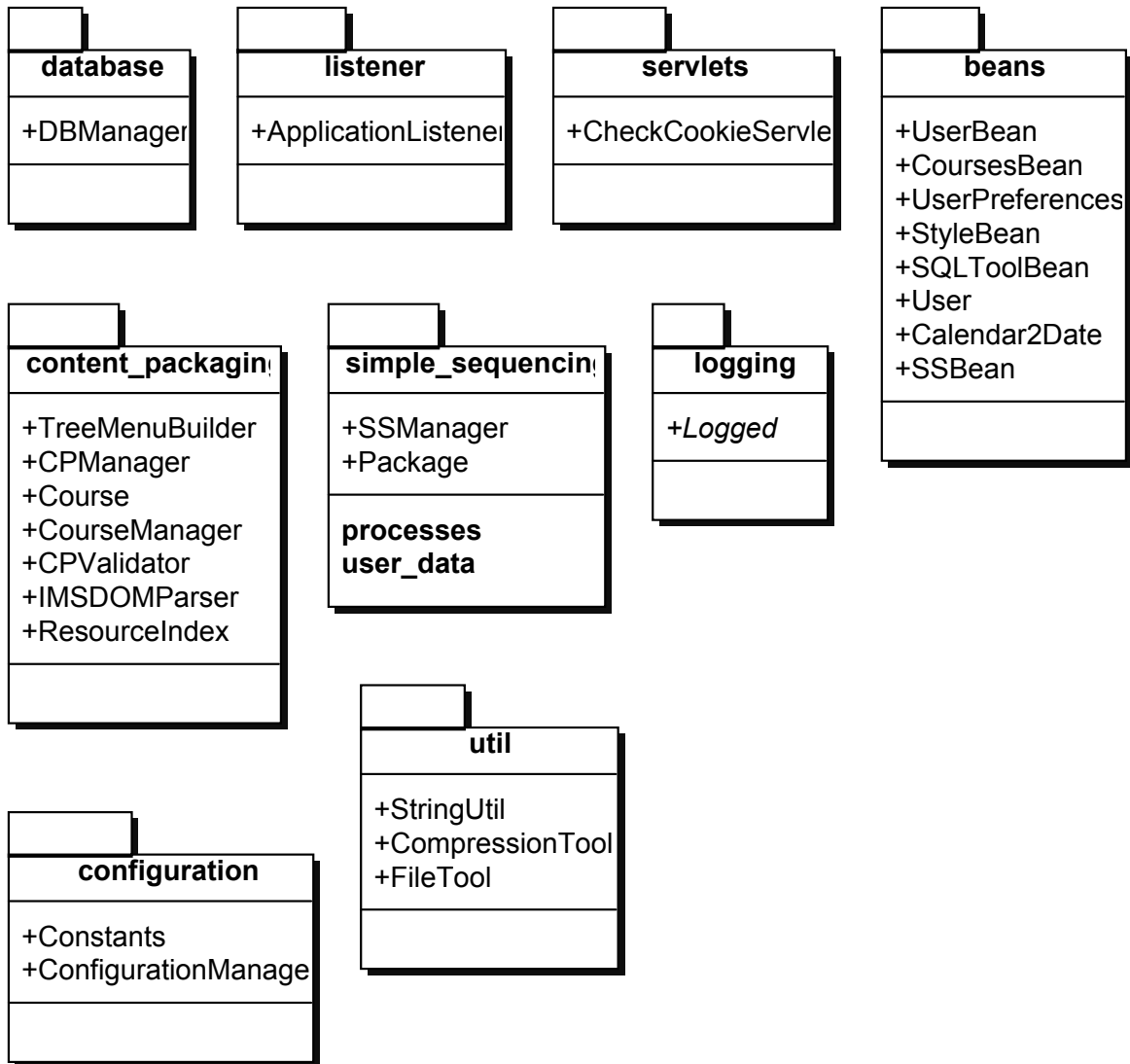
Οποιοσδήποτε σύγχρονος browser με JavaScript και Cookies enabled.

3.3.3 Λεπτομερής Σχεδίαση

Στη συνέχεια παρουσιάζονται λεπτομερή UML package και class diagrams των packages και των κλάσεων, αντίστοιχα, του συστήματος.

3.3.3.1 Packages

Τα packages που αποτελούν τον κώδικα του συστήματος φαίνονται στο σχήμα 3.20 :

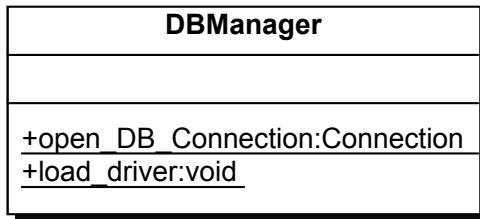


Σχήμα 3.20 Packages του IMS Run-Time Environment

Σημειώνουμε ότι για λόγους απλότητας στο σχήμα δεν φαίνεται το πλήρες όνομα κάθε package. Το πλήρες όνομα ενός package είναι `gr.ntua.softlab.ims.` συν το όνομα που φαίνεται στο σχήμα, π.χ. `gr.ntua.softlab.ims.content_packaging`

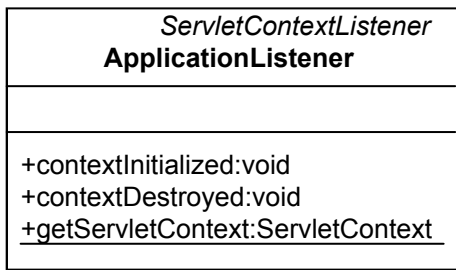
Στη συνέχεια για κάθε package θα παρουσιαστεί ένα class diagram των κλάσεων που το αποτελούν.

3.3.3.2 Package gr.ntua.softlab.ims.database



Σχήμα 3.21 Class diagram για το package gr.ntua.softlab.ims.database

3.3.3.3 Package gr.ntua.softlab.ims.listener



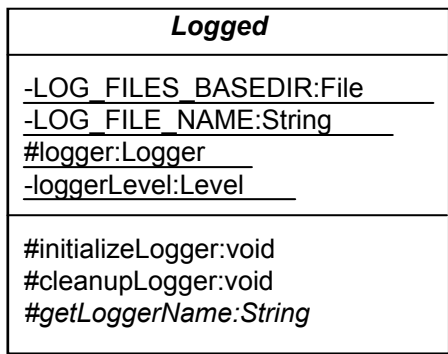
Σχήμα 3.22 Class diagram για το package gr.ntua.softlab.ims.listener

3.3.3.4 Package gr.ntua.softlab.ims.servlets



Σχήμα 3.23 Class diagram για το package gr.ntua.softlab.ims.servlets

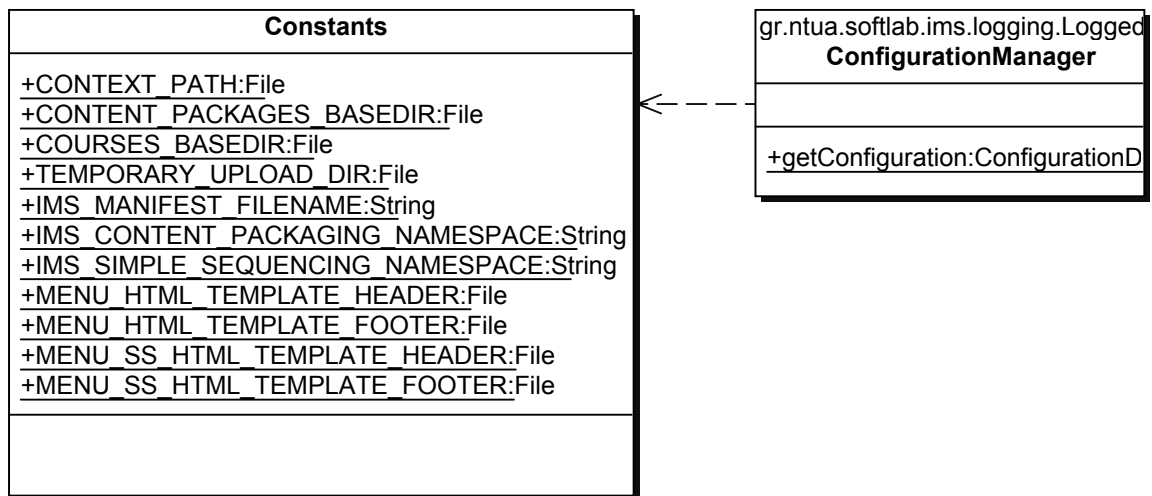
3.3.3.5 Package gr.ntua.softlab.ims.logging



Σχήμα 3.24 Class diagram για το package gr.ntua.softlab.ims.logging

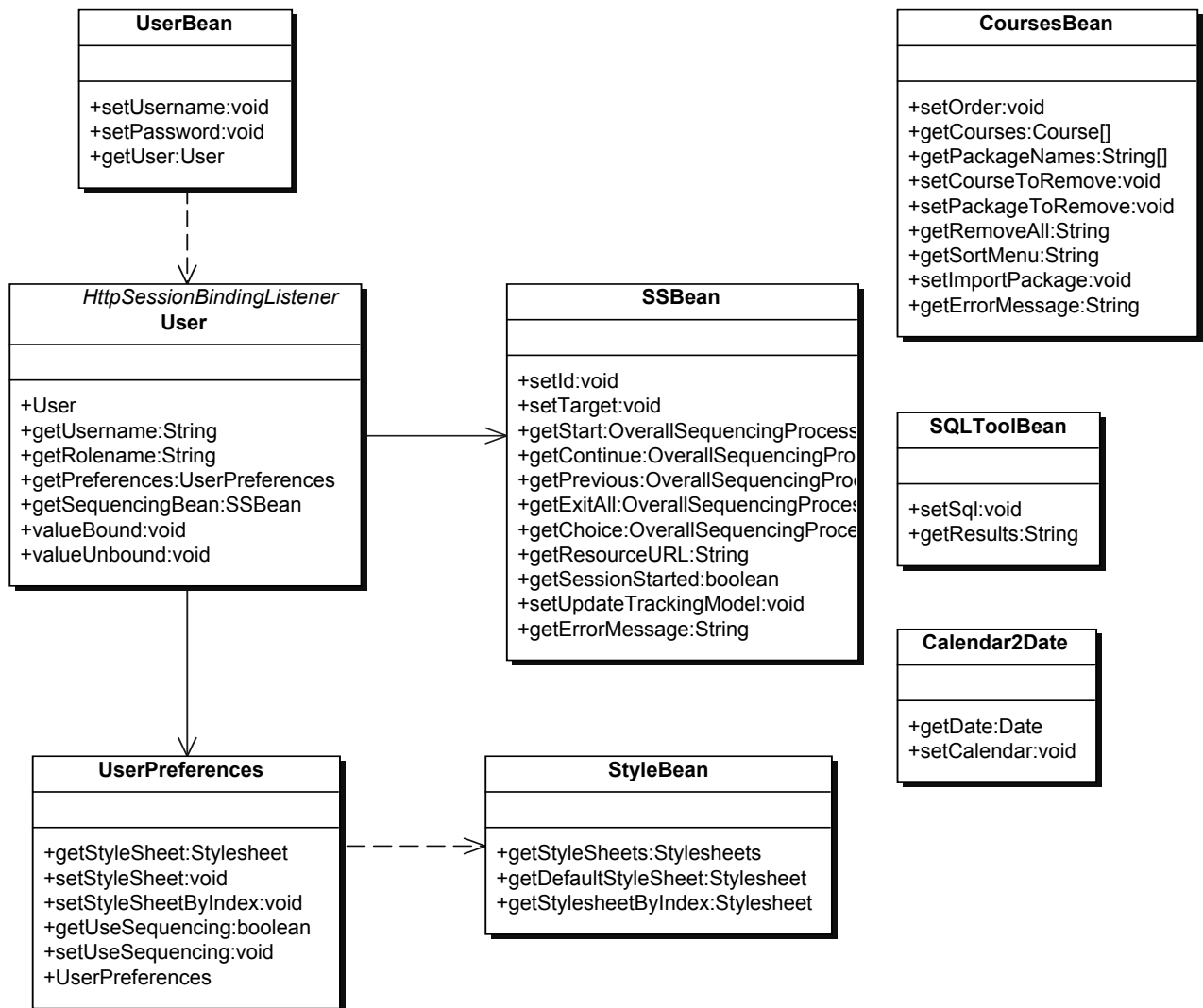
3.3.3.6 Package gr.ntua.softlab.ims.configuration

Note : This package also contains interfaces that were generated from the ims_rte_configuration.xsd schema using Apache XMLBeans. These interfaces are located in the ims_rte_configuration.jar JAR file.



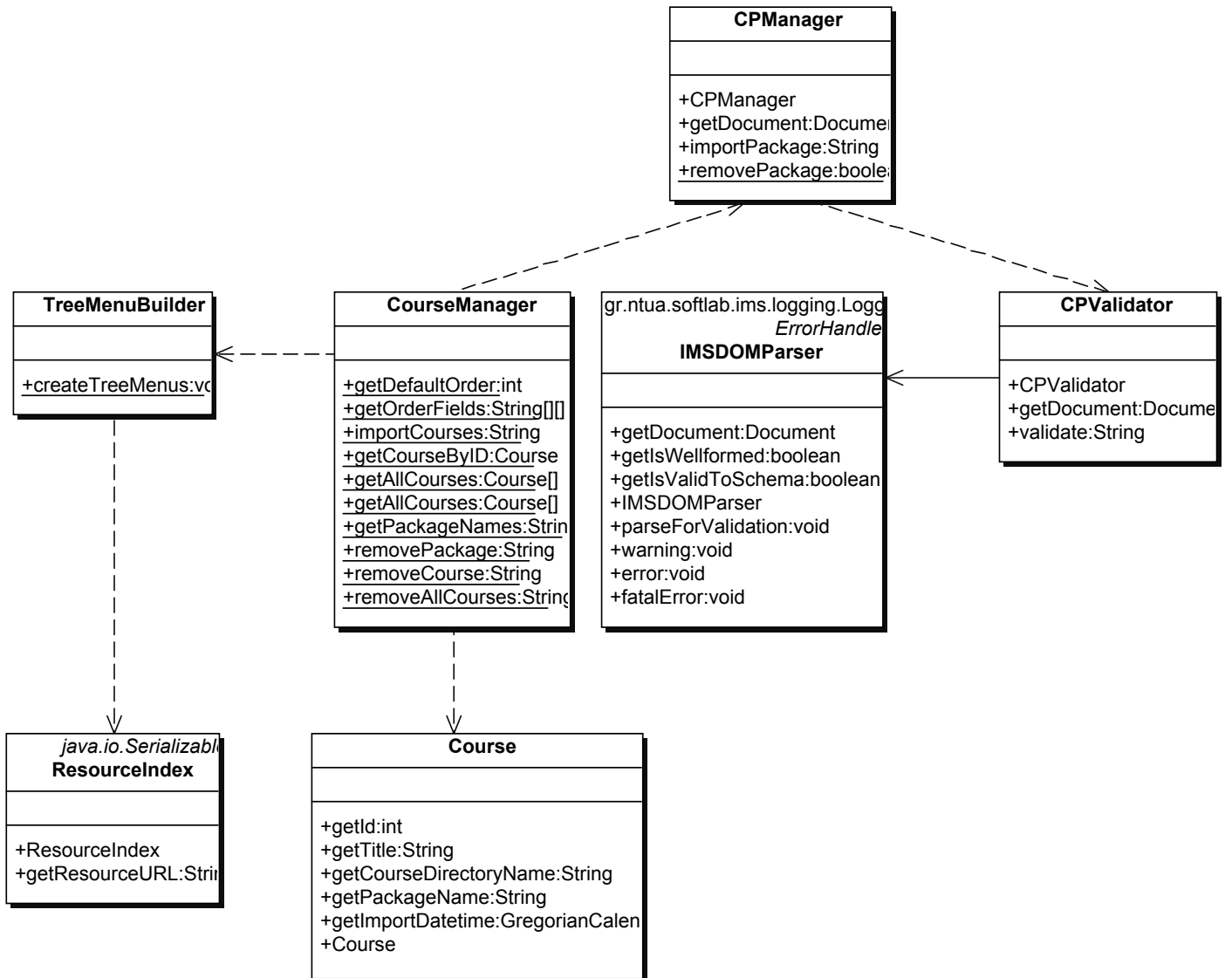
Σχήμα 3.25 Class diagram για το package gr.ntua.softlab.ims.configuration

3.3.3.7 Package gr.ntua.softlab.ims.beans



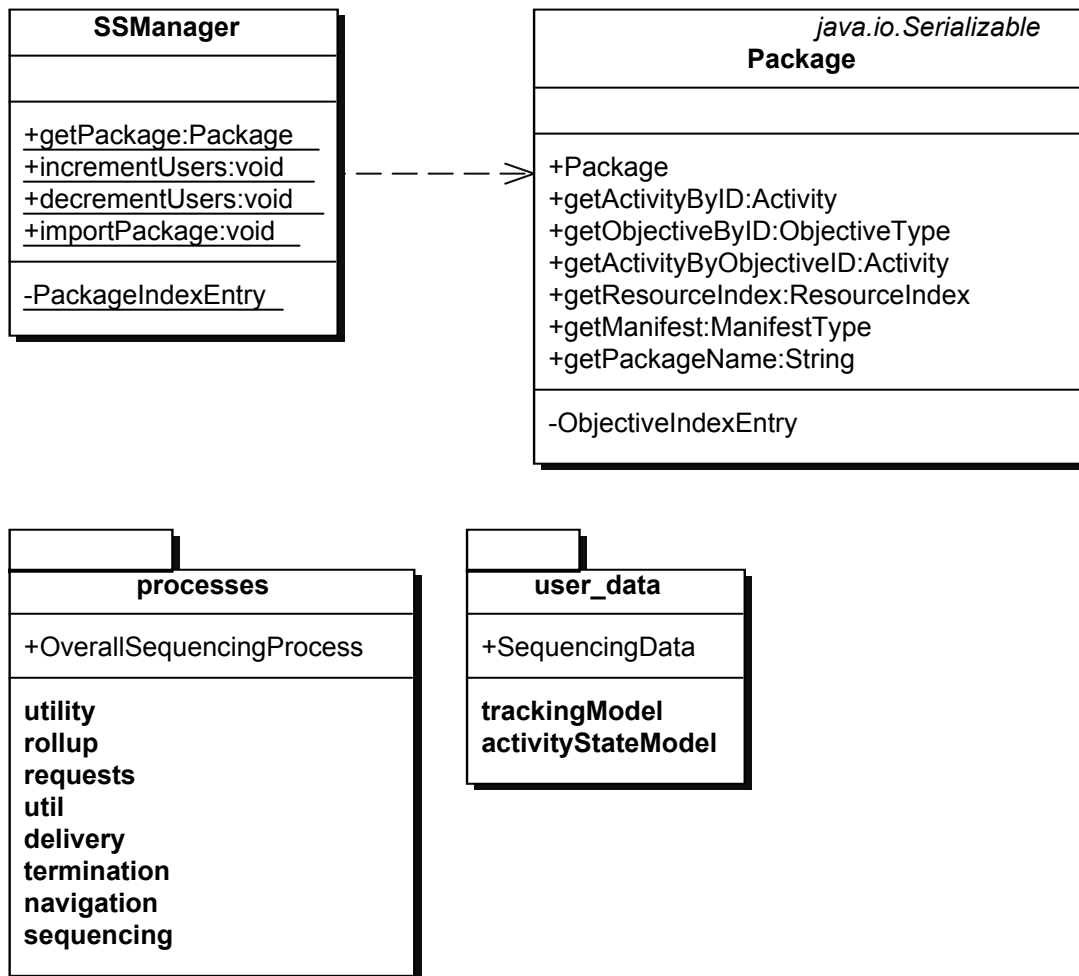
Σχήμα 3.26 Class diagram για το package `gr.ntua.softlab.ims.beans`

3.3.3.8 Package gr.ntua.softlab.ims.content_packaging



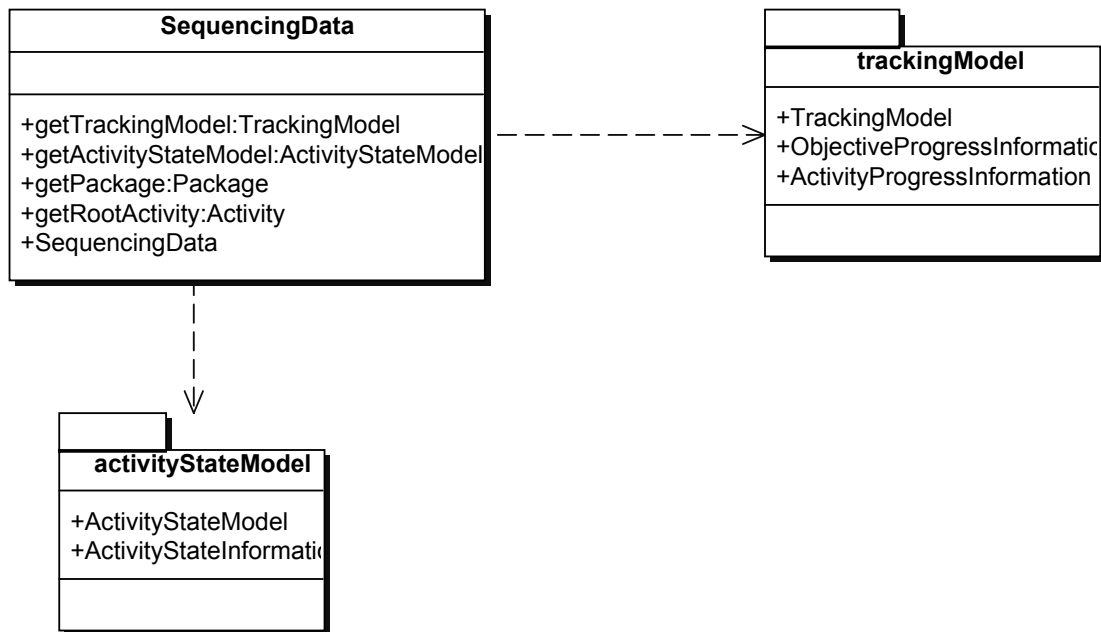
Σχήμα 3.27 Class diagram για το package `gr.ntua.softlab.ims.content_packaging`

3.3.3.9 Package gr.ntua.softlab.ims.simple_sequencing



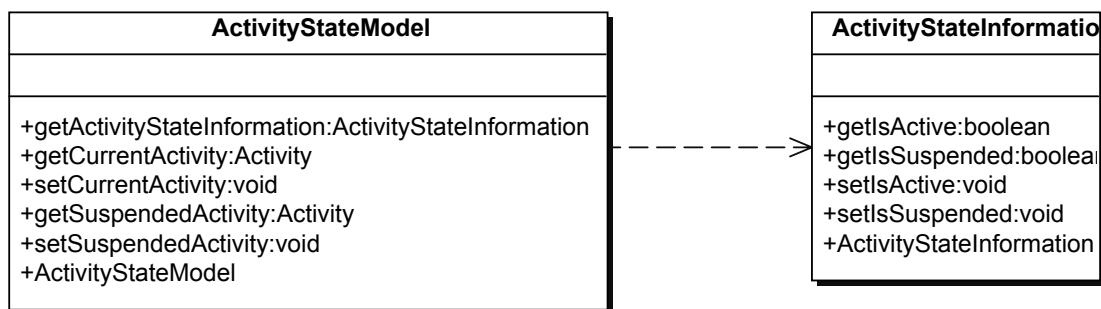
Σχήμα 3.28 Class diagram για το package `gr.ntua.softlab.ims.simple_sequencing`

3.3.3.10 Package gr.ntua.softlab.ims.simple_sequencing.user_data



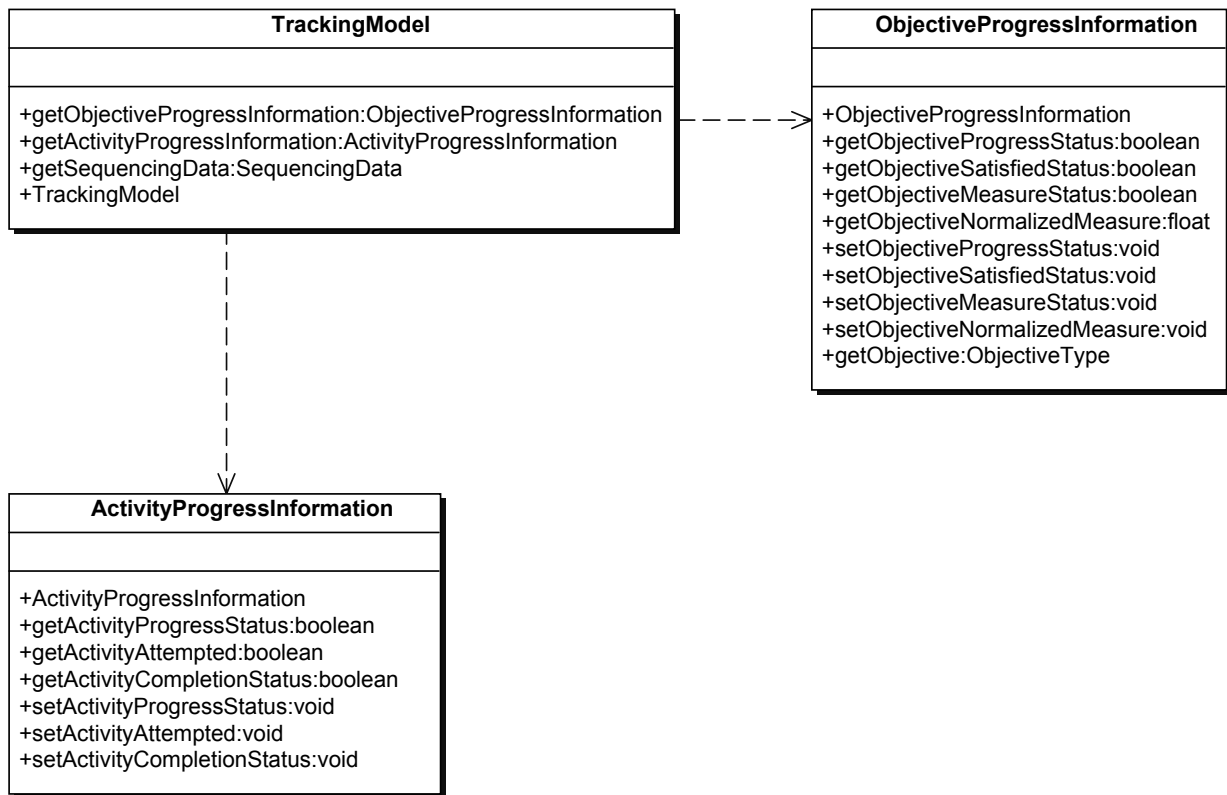
Σχήμα 3.29 Class diagram για το package gr.ntua.softlab.ims.simple_sequencing.user_data

3.3.3.11 Package gr.ntua.softlab.ims.simple_sequencing.user_data.activityStateModel



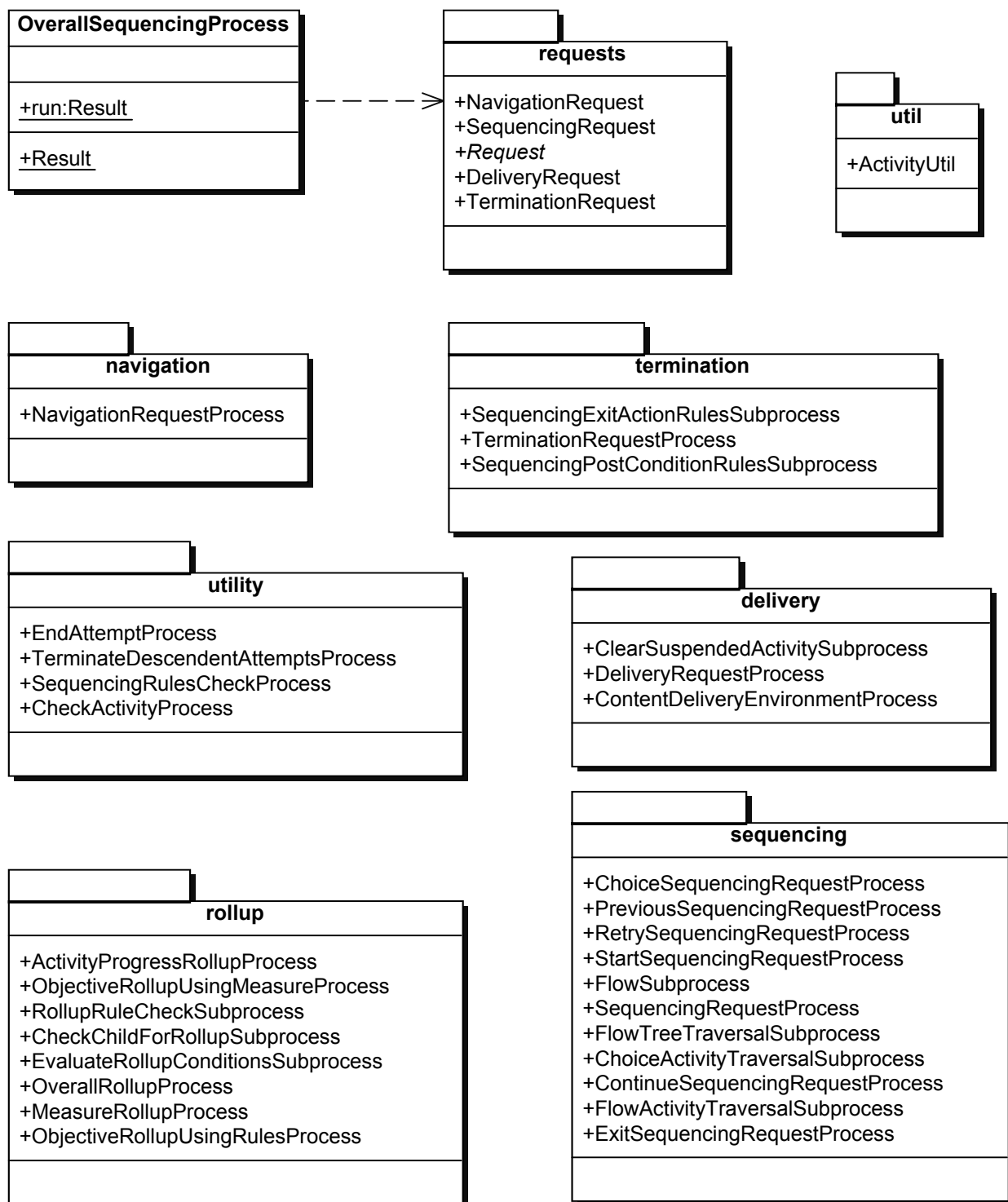
Σχήμα 3.30 Class diagram για το package gr.ntua.softlab.ims.simple_sequencing.user_data.activityStateModel

3.3.3.12 Package `gr.ntua.softlab.ims.simple_sequencing.user_data.trackingModel`



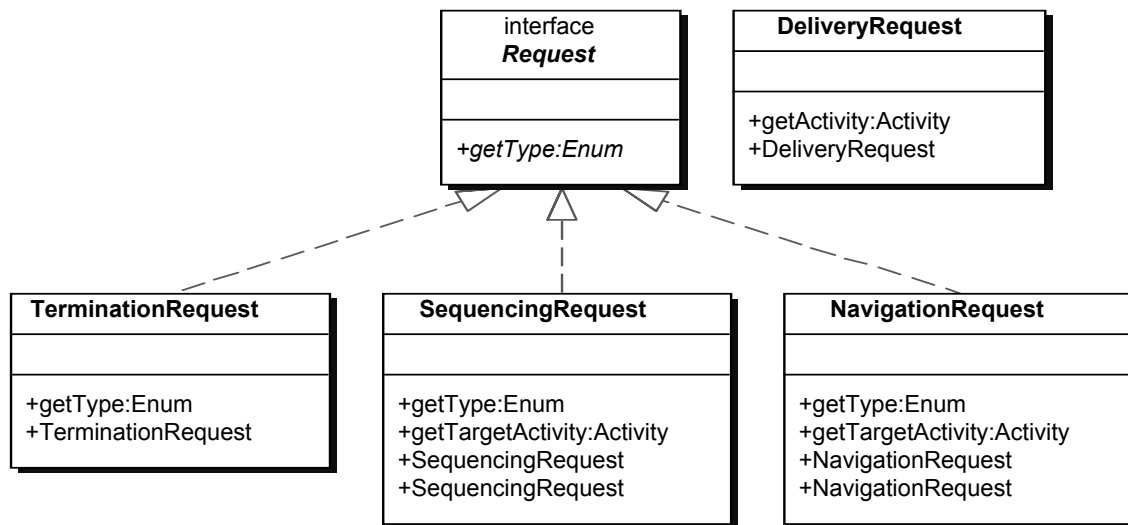
Σχήμα 3.31 Class diagram για το package `gr.ntua.softlab.ims.simple_sequencing.user_data.trackingModel`

3.3.3.13 Package gr.ntua.softlab.ims.simple_sequencing.processes



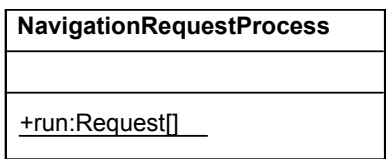
Σχήμα 3.32 Class diagram για το package gr.ntua.softlab.ims.simple_sequencing.processes

3.3.3.14 Package `gr.ntua.softlab.ims.simple_sequencing.processes.requests`



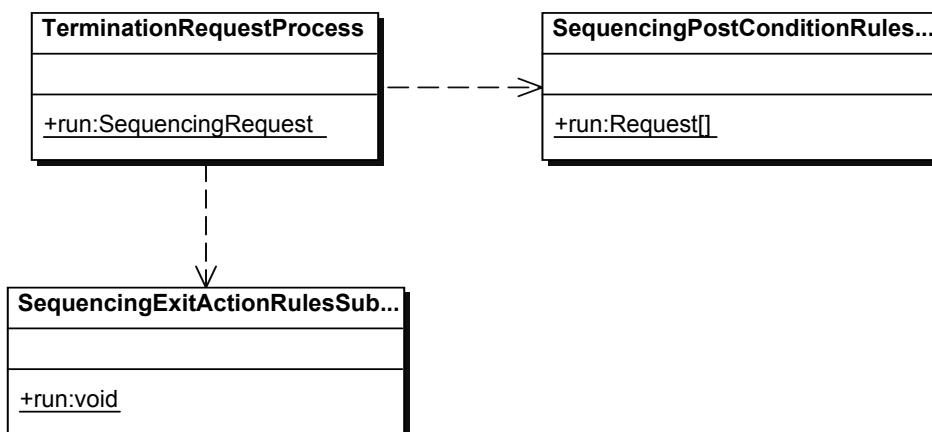
Σχήμα 3.33 Class diagram για το package `gr.ntua.softlab.ims.simple_sequencing.processes.requests`

3.3.3.15 Package `gr.ntua.softlab.ims.simple_sequencing.processes.navigation`



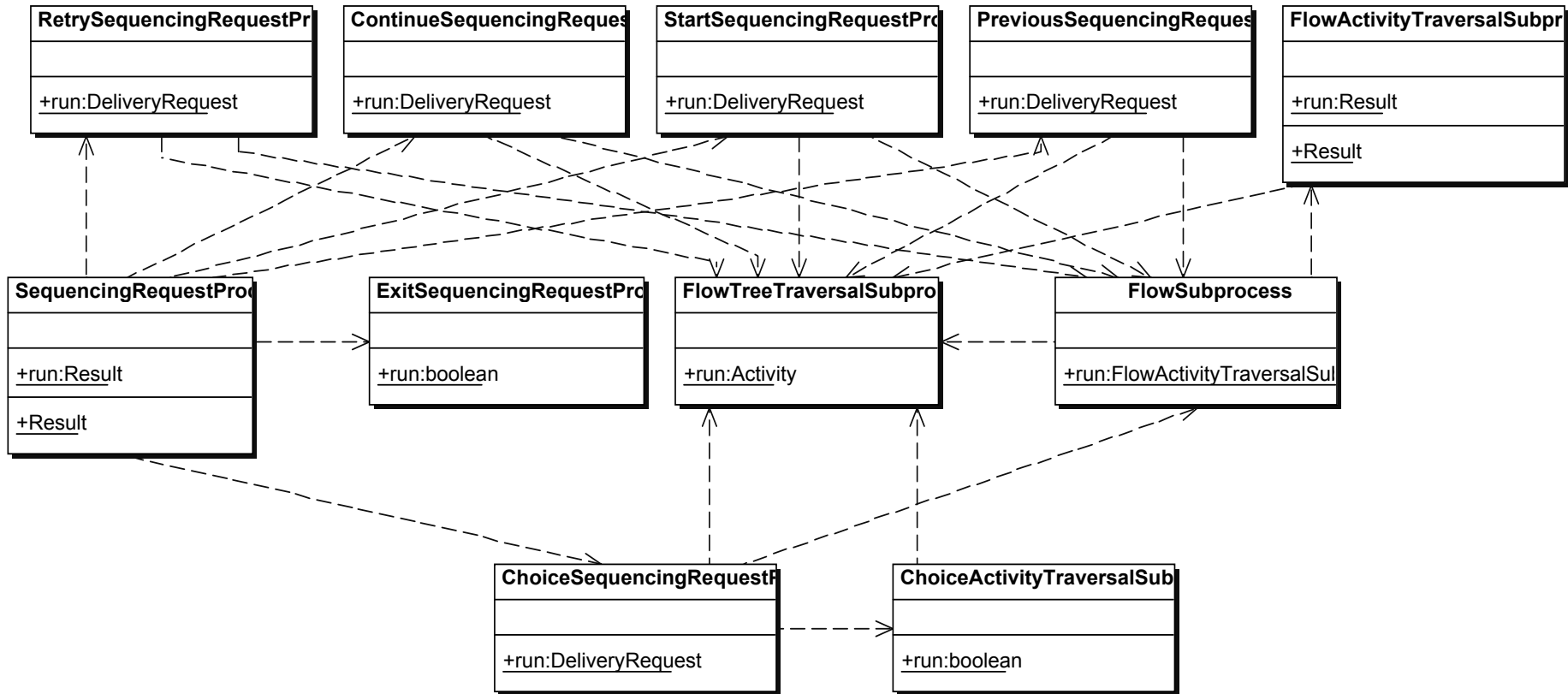
Σχήμα 3.34 Class diagram για το package `gr.ntua.softlab.ims.simple_sequencing.processes.navigation`

3.3.3.16 Package `gr.ntua.softlab.ims.simple_sequencing.processes.termination`



Σχήμα 3.35 Class diagram για το package `gr.ntua.softlab.ims.simple_sequencing.processes.termination`

3.3.3.17 Package gr.ntua.softlab.ims.simple_sequencing.processes.sequencing



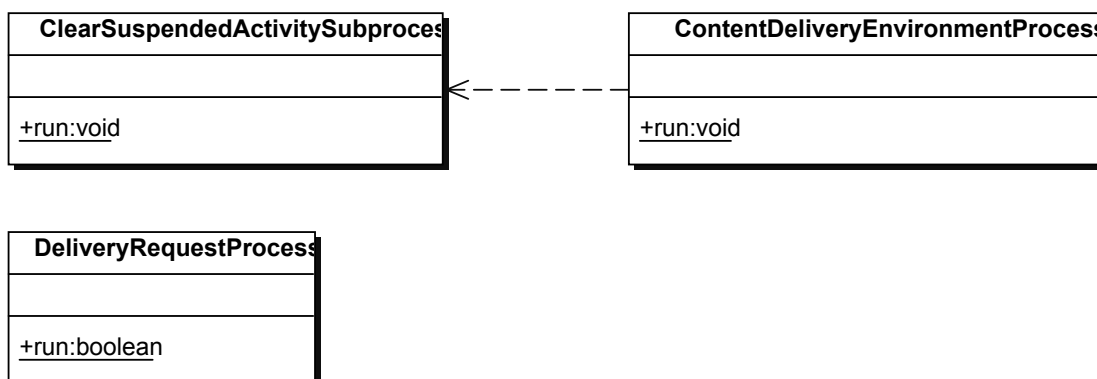
Σχήμα 3.36 Class diagram για το package gr.ntua.softlab.ims.simple_sequencing.processes.sequencing

3.3.3.18 Package `gr.ntua.softlab.ims.simple_sequencing.processes.rollup`



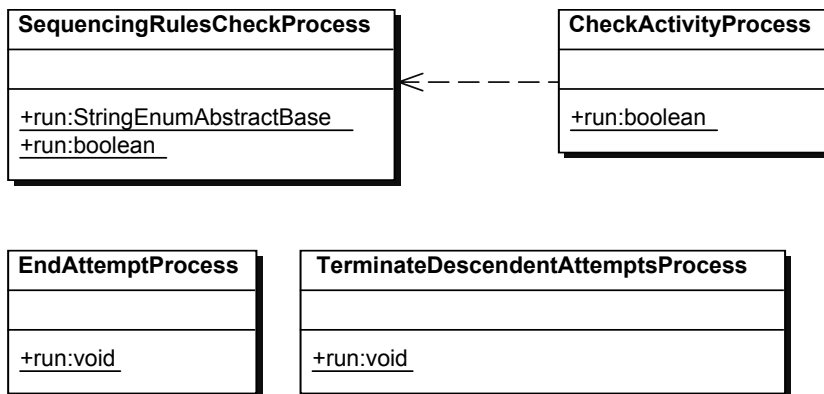
Σχήμα 3.37 Class diagram για το package `gr.ntua.softlab.ims.simple_sequencing.processes.rollup`

3.3.3.19 Package `gr.ntua.softlab.ims.simple_sequencing.processes.delivery`



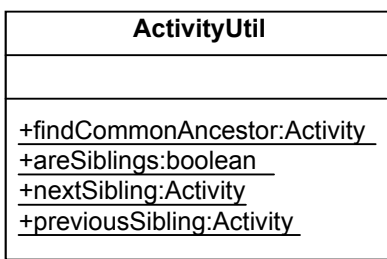
Σχήμα 3.38 Class diagram για το package `gr.ntua.softlab.ims.simple_sequencing.processes.delivery`

3.3.3.20 Package `gr.ntua.softlab.ims.simple_sequencing.processes.utility`



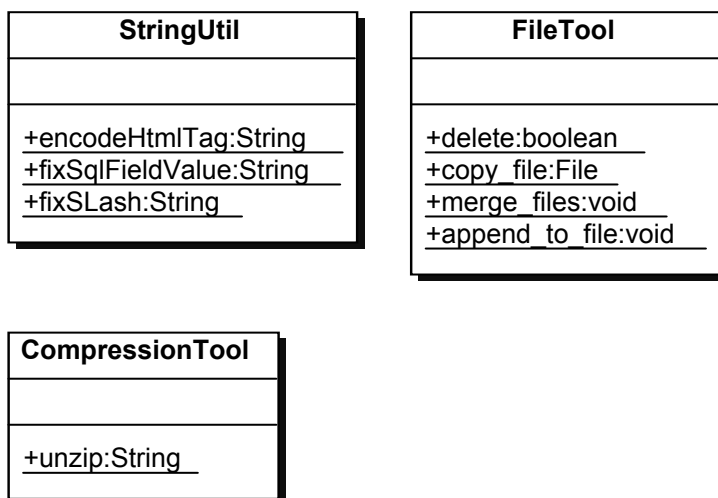
Σχήμα 3.39 Class diagram για το package `gr.ntua.softlab.ims.simple_sequencing.processes.utility`

3.3.3.21 Package `gr.ntua.softlab.ims.simple_sequencing.processes.util`



Σχήμα 3.40 Class diagram για το package `gr.ntua.softlab.ims.simple_sequencing.processes.util`

3.3.3.22 Package `gr.ntua.softlab.ims.util`



Σχήμα 3.41 Class diagram για το package `gr.ntua.softlab.ims.util`

4 Case Study

Στο κεφάλαιο αυτό παρουσιάζεται μια μελέτη περίπτωσης (case study), δηλαδή ένα ρεαλιστικό παράδειγμα χρήσης του συστήματος και των υποστηριζόμενων προδιαγραφών. Συγκεκριμένα, πρώτα θα περιγράψουμε ένα συγκεκριμένο σενάριο μάθησης, μια *μαθησιακή εμπειρία*, η οποία θέλουμε να παρέχεται από το σύστημα ως κάποιο course. Στη συνέχεια θα αναπαραστήσουμε τη μαθησιακή εμπειρία σύμφωνα με τη προδιαγραφή IMS Simple Sequencing ως ένα activity tree με αντίστοιχες πληροφορίες sequencing. Η αναπαράσταση θα γίνει πρώτα σχηματικά και ύστερα θα κωδικοποιηθεί σε XML, σύμφωνα με το XML binding που ορίζεται στο [SS 03b].

4.1 Σχεδιασμός της μαθησιακής εμπειρίας

Υποθέτουμε ότι θέλουμε να δημιουργήσουμε ένα ηλεκτρονικό μάθημα προγραμματισμού στη γλώσσα μηχανής (assembly) του επεξεργαστή 80x86. Είναι γνωστό ότι η εκμάθηση προγραμματισμού σε assembly προϋποθέτει τη γνώση ορισμένων βασικών στοιχείων από την Αρχιτεκτονική Υπολογιστών, π.χ. αριθμητικά συστήματα και αναπαράσταση αριθμών στον Η/Υ, οργάνωση της μνήμης κ.α. Το μάθημα θέλουμε να σχεδιαστεί με τέτοιο τρόπο ώστε να μπορεί να είναι χρήσιμο σε μαθητές διαφόρων επιπέδων : από τους μαθητές που ήδη έχουν κάποια μικρή εμπειρία στον προγραμματισμό σε assembly μέχρι εκείνους που είναι εντελώς αρχάριοι και δεν διαθέτουν καν τις προαναφερθείσες βασικές γνώσεις. Προφανώς δεν έχει νόημα να παρουσιαστεί ακριβώς το ίδιο μαθησιακό υλικό σε όλους τους μαθητές. Αντιθέτως, αυτό που θέλουμε είναι να κατατάξουμε τους μαθητές σε 4 κατηγορίες, με την κατηγορία 1 να περιλαμβάνει τους εντελώς αρχάριους, την κατηγορία 4 τους πολύ προχωρημένους, και τις δύο ενδιάμεσες να περιλαμβάνουν μαθητές μεσαίου επιπέδου, και να παρουσιάσουμε το κατάλληλο μαθησιακό υλικό ανάλογα με την κατηγορία στην οποία ανήκει ο μαθητής. Η κατάταξη των μαθητών θα γίνει με αυτόματο τρόπο, βάζοντάς τους να απαντήσουν σε ένα τεστ γνώσεων αποτελούμενο από 10 multiple choice ερωτήσεις, οι οποίες φαίνονται στο σχήμα 4.1.

1. Which is the equivalent decimal number of the binary number 1000111 ?

- 61
- 135
- 360

2. What is the range of decimal integers that can be stored in 2's complement form in a byte ?

- [0,255]
- [-255, 255]
- [-32768, 32767]
- [-128, 127]

3. What is the segment size in the 8086/8088 segmented memory model ?

- 32 Kbytes
- 64 Kbytes
- 128 Kbytes
- 256 Kbytes

4. What is the smallest possible exponent for IEEE 754 single precision *normalized* floating point numbers?

- 126
- 127
- 128
- None of the above

5. Suppose we want to add $(+80)_{10}$ and $(+50)_{10}$ using an 8 bit 2's complement format. What is the result ?

- $(+130)_{10}$
- An overflow will occur producing the result $(-126)_{10}$
- An overflow will occur producing the result $(-2)_{10}$
- None of the above

6. Suppose that you have an Intel 8086. Which five-hex-digit address corresponds to the segment:offset pair 18A3:5B27 ?

- 2B8CD
- 059A7
- 12345
- 1E557

7. Suppose that register EAX initially contains the word FF FF FF F3. What will the content of EAX be after the execution of the following instruction ?

add eax, 158

- 00 00 00 91
- 00 00 00 45
- 7F FF FF FF

8. Which is the decimal number corresponding to the BCD binary number 1001 0010 0110 0111 0000 1000 ?

- 523665
- 926708
- 372349

9. What is the size of register CS in the 80x86 CPU ?

- 8 bits
- 16 bits
- 32 bits

10. What is the maximum memory size of a modern IBM PC ?

- 4 gigabytes
- 2 gigabytes
- 1 gigabyte
- None of the above

Σχήμα 4.1 Ερωτηματολόγιο για την αξιολόγηση του επιπέδου γνώσεων των μαθητών

Ο μαθητής θα κατατάσσεται σε μια από τις 4 κατηγορίες ανάλογα με το πόσες από τις 10 ερωτήσεις απάντησε σωστά. Συγκεκριμένα έχουμε την ακόλουθη αντιστοίχιση :

Αριθμός σωστών απαντήσεων	Κατηγορία
0-2	1
3-7	2
8-9	3
10	4

Πίνακας 4.1 Κατάταξη των μαθητών σε κατηγορίες

Ανάλογα με την κατηγορία στην οποία ανήκει ένας μαθητής θα του παρουσιάζεται και το κατάλληλο μαθησιακό υλικό. Για να γίνει αυτό χωρίζουμε το μαθησιακό υλικό σε 3 μέρη : Remediation, General και Advanced. Το Remediation περιλαμβάνει τις βασικές γνώσεις από την Αρχιτεκτονική Υπολογιστών που απαιτούνται για την παρακολούθηση του μαθήματος. Το General περιλαμβάνει το κύριο μέρος της ύλης και τέλος το Advanced περιλαμβάνει κάποια προχωρημένα θέματα. Στον παρακάτω πίνακα φαίνεται ποιά μέρη του μαθησιακού υλικού θα παρουσιάζονται σε κάθε κατηγορία μαθητών :

Κατηγορία	Μαθησιακό υλικό
1	Remediation, General
2	General
3	General, Advanced
4	Advanced

Πίνακας 4.2 Αντιστοίχιση μαθησιακού υλικού σε κατηγορίες μαθητών

Καθένα από τα τρία μέρη αποτελείται από κάποια κεφάλαια, όπως φαίνεται παρακάτω :

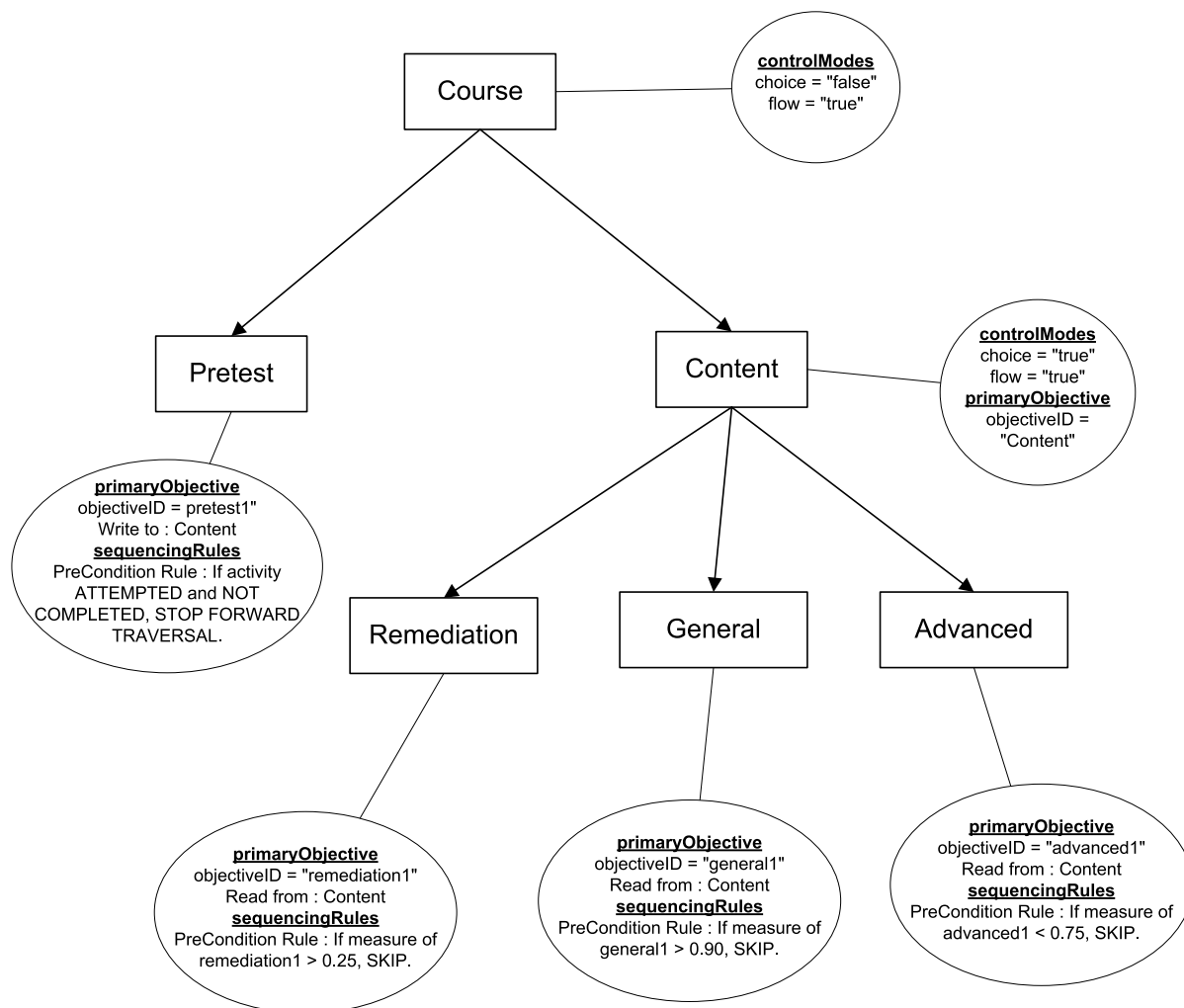
1	Remediation
1.1	Representing Data in a Computer
1.2	Parts of a Computer System
2	General
2.1	Elements of Assembly Language
2.2	Basic Instructions
2.3	Branching and Looping
3	Advanced
3.1	Procedures
3.2	String Operations

Σχήμα 4.2 Κεφάλαια του μαθήματος

Κάθε κεφάλαιο αποτελείται από ενότητες, οι οποίες όμως δεν φαίνονται εδώ για λόγους απλότητας.

4.2 Υλοποίηση

Έχοντας σχεδιάσει τη μαθησιακή εμπειρία σε υψηλό επίπεδο, το επόμενο βήμα είναι η αναπαράστασή της ως ένα activity tree σύμφωνα με τη προδιαγραφή IMS Simple Sequencing. Το activity tree φαίνεται στο σχήμα 4.3 :



Σχήμα 4.3 Activity tree και πληροφορίες sequencing

Στο παραπάνω σχήμα για λόγους απλότητας δεν φαίνονται τα κεφάλαια και οι ενότητες από τα οποία αποτελούνται τα Remediation, General και Advanced. Αυτό όμως δεν μας πειράζει, γιατί οι σημαντικές αποφάσεις sequencing λαμβάνονται στο επίπεδο των Remediation, General και Advanced και όχι στα χαμηλότερα επίπεδα του δέντρου. Το activity Pretest αντιπροσωπεύει το τεστ αξιολόγησης που πρέπει να συμπληρωθεί αρχικά, ώστε στη συνέχεια να επιλεγεί το κατάλληλο μαθησιακό υλικό. Ο βαθμός του τεστ αξιολόγησης θα αποθηκευτεί στο measure του objective pretest1 και από κει μέσω του objective mapping που έχει οριστεί θα αποθηκευτεί και στο measure του global objective Content. Στο activity Pretest έχει επίσης οριστεί ένας precondition κανόνας με ενέργεια (rule action) *Stop Forward Traversal*, ο οποίος πυροδοτείται όταν το activity Pretest είναι *Attempted* αλλά όχι *Completed*, δηλαδή όταν έχει παραδοθεί στο μαθητή αλλά ο μαθητής δεν έχει ακόμα υποβάλλει (submit)

τις απαντήσεις του, με αποτέλεσμα να μην είναι δυνατή η επιλογή του κατάλληλου υλικού για αυτόν.

Επίσης σε καθένα από τα τρία μέρη Remediation, General και Advanced έχει οριστεί και ένας precondition κανόνας με ενέργεια *Skip*, έτσι ώστε αυτό να παραδίδεται ή να μην παραδίδεται ανάλογα με το βαθμό που πήρε ο μαθητής στο τεστ, και ο οποίος όπως είπαμε έχει αποθηκευτεί στο measure του global objective Content.

Τέλος σημειώνουμε ότι η καθοδήγηση αυτή που παρέχουμε είναι προαιρετική, μιας και όπως φαίνεται από την τιμή του control mode *Choice* στο activity Content, ο μαθητής είναι ελεύθερος να επιλέξει οποιοδήποτε μέρος του μαθησιακού υλικού αυτός επιθυμεί.

Στη συνέχεια κωδικοποιούμε το activity tree του σχήματος 4.3 σε XML, σύμφωνα με το XML binding που ορίζεται στο [SS 03b]. Το προκύπτον IMS Manifest δίνεται παρακάτω :

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1" xmlns:imsss="http://www.imsglobal.org/xsd/imsss"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1
xsd/imscp_v1p1.xsd http://www.imsglobal.org/xsd/imsss xsd/imsss_v1p0.xsd" identifier="BBT_TLCExamplePostTest">
  <!-- This manifest contains a solution to the following ID Problem:
  * Require pretest before content
  * Deliver material based on score in pretest
    * N < 25% => Remediation then General
    * 25% < N < 75% => General only
    * 75% < N < 90% => General then Advanced
    * N > 90% => Advanced only
  * The student is not allowed to choose the material.
-->
  <organizations>
    <organization identifier="Course">
      <title>80x86 Assembly Language Programming</title>
      <item identifier="Pretest" identifierref="RESOURCE1">
        <imsss:sequencing>
          <imsss:sequencingRules>
            <imsss:preConditionRule>
              <!-- This rule means that if this activity is attempted but not completed then it won't be
possible to go
              to the 'next' activity (because traversal of the tree in the forward direction won't be allowed). -->
              <imsss:ruleConditions conditionCombination="all">
                <imsss:ruleCondition condition="attempted"/>
                <!-- Instead of condition="completed" we could have set
condition="objectiveMeasureKnown"
                because in this case the two conditions are equivalent : If the user has taken the test (activity
and
                completed) then there will be a measure for the objective pretest1 (objective measure known),
                vice versa. If we used condition="objectiveMeasureKnown" then we also wouldn't need to
set
                completionSetByContent to true in the delivery controls. But we chose this solution for
demonstration purposes.-->
                <imsss:ruleCondition condition="completed" operator="not"/>
              </imsss:ruleConditions>
              <imsss:ruleAction action="stopForwardTraversal"/>
            </imsss:preConditionRule>
          </imsss:sequencingRules>
        </imsss:sequencing>
      </item>
    </organization>
  </organizations>
  <imsss:primaryObjective objectiveID="pretest1">
```

```

        <!-- Because writeNormalizedMeasure is set to true, when we set Objective Measure Status or
        Objective Normalized Measure for objective 'pretest1', it will also be set for objective 'Content'.
        Objective 'Content' is the primary objective of activity 'Content'. It's not defined here explicitly,
        but an activity has always exactly one primary objective. If it's not defined explicitly, then a
        default primary objective is used. The objectiveID of the default primary objective is equal to
        the identifier of the activity. -->
        <imsss:mapInfo targetObjectiveID="Content" writeNormalizedMeasure="true"/>
    </imsss:primaryObjective>
</imsss:objectives>
<!-- We set completionSetByContent to true because we want the content itself (i.e. the pretest) to set
the
only if
logic is
value of ActivityCompletionStatus for this activity. ActivityCompletionStatus will be set to true if and
the user clicks on the Submit button of the pretest (see pretest.html and ssUpdate.js for details). The
that this activity should be considered completed only if the user takes (=submits) the test. -->
    <imsss:deliveryControls completionSetByContent="true"/>
</imsss:sequencing>
</item>
<item identifier="Content" isVisible="false">
    <item identifier="Remediation">
        <item identifier="R.1">
            <title>1. Representing Data in a Computer</title>
            <item identifier="R.1.1" identifierref="RESOURCE_R.1.1">
                <title>1.1 Binary and Hexadecimal Numbers</title>
            </item>
            <item identifier="R.1.2" identifierref="RESOURCE_R.1.2">
                <title>1.2 Character Codes</title>
            </item>
            <item identifier="R.1.3" identifierref="RESOURCE_R.1.3">
                <title>1.3 2's Complement Representation for Signed Integers</title>
            </item>
            <item identifier="R.1.4" identifierref="RESOURCE_R.1.4">
                <title>1.4 Addition and Subtraction of 2's Complement Numbers</title>
            </item>
            <item identifier="R.1.5" identifierref="RESOURCE_R.1.5">
                <title>1.5 Other Systems for Representing Numbers</title>
            </item>
            <imsss:sequencing IDRef="GLOBAL_SEQ_1"/>
        </item>
        <item identifier="R.2">
            <title>2. Parts of a Computer System</title>
            <item identifier="R.2.1" identifierref="RESOURCE_R.2.1">
                <title>2.1 PC Hardware: Memory</title>
            </item>
            <item identifier="R.2.2" identifierref="RESOURCE_R.2.2">
                <title>2.2 PC Hardware: The CPU</title>
            </item>
            <item identifier="R.2.3" identifierref="RESOURCE_R.2.3">
                <title>2.3 PC Hardware: Input/Output Devices</title>
            </item>
            <item identifier="R.2.4" identifierref="RESOURCE_R.2.4">
                <title>2.4 PC Software</title>
            </item>
            <imsss:sequencing IDRef="GLOBAL_SEQ_1"/>
        </item>
    <imsss:sequencing IDRef="GLOBAL_SEQ_1">
        <imsss:sequencingRules>
            <imsss:preConditionRule>
                <imsss:ruleConditions>
                    <imsss:ruleCondition condition="objectiveMeasureGreaterThan"
referencedObjective="remediation1" measureThreshold="0.25"/>
                </imsss:ruleConditions>
                <imsss:ruleAction action="skip"/>
            </imsss:preConditionRule>
        </imsss:sequencingRules>
    </imsss:sequencing IDRef="GLOBAL_SEQ_1">

```



```

</item>
<item identifier="G.3.2" identifierref="RESOURCE_G.3.2">
  <title>5.2 Conditional Jumps, Compare Instructions, and if Structures</title>
</item>
<item identifier="G.3.3" identifierref="RESOURCE_G.3.3">
  <title>5.3 Implementing Loop Structures</title>
</item>
<item identifier="G.3.4" identifierref="RESOURCE_G.3.4">
  <title>5.4 for Loops in Assembly Language</title>
</item>
<item identifier="G.3.5" identifierref="RESOURCE_G.3.5">
  <title>5.5 Arrays</title>
</item>
<item identifier="G.3.6" identifierref="RESOURCE_G.3.6">
  <title>5.6 Something Extra: Pipelining</title>
</item>
<imsss:sequencing IDRef="GLOBAL_SEQ_1"/>
</item>
<imsss:sequencing IDRef="GLOBAL_SEQ_1">
  <imsss:sequencingRules>
    <imsss:preConditionRule>
      <imsss:ruleConditions>
        <imsss:ruleCondition condition="objectiveMeasureGreaterThan"
referencedObjective="generall" measureThreshold="0.90"/>
      </imsss:ruleConditions>
      <imsss:ruleAction action="skip"/>
    </imsss:preConditionRule>
  </imsss:sequencingRules>
  <imsss:objectives>
    <imsss:primaryObjective objectiveID="generall">
      <!-- Because readNormalizedMeasure is set to true, Objective Measure Status and
Objective Normalized Measure for objective 'generall' will be read from objective 'Content' --
>
      <imsss:mapInfo targetObjectiveID="Content" readNormalizedMeasure="true"/>
    </imsss:primaryObjective>
  </imsss:objectives>
</imsss:sequencing>
</item>
<item identifier="Advanced" identifierref="RESOURCE4">
  <item identifier="A.1">
    <title>6. Procedures</title>
    <item identifier="A.1.1" identifierref="RESOURCE_A.1.1">
      <title>6.1 The 80×86 Stack</title>
    </item>
    <item identifier="A.1.2" identifierref="RESOURCE_A.1.2">
      <title>6.2 Procedure Body, Call and Return</title>
    </item>
    <item identifier="A.1.3" identifierref="RESOURCE_A.1.3">
      <title>6.3 Parameters and Local Variables</title>
    </item>
    <item identifier="A.1.4" identifierref="RESOURCE_A.1.4">
      <title>6.4 Recursion</title>
    </item>
    <item identifier="A.1.5" identifierref="RESOURCE_A.1.5">
      <title>6.5 Other Architectures: Procedures Without a Stack</title>
    </item>
    <imsss:sequencing IDRef="GLOBAL_SEQ_1"/>
  </item>
  <item identifier="A.2">
    <title>7. String Operations</title>
    <item identifier="A.2.1" identifierref="RESOURCE_A.2.1">
      <title>7.1 Using String Instructions</title>
    </item>
    <item identifier="A.2.2" identifierref="RESOURCE_A.2.2">

```

```

        <title>7.2 Repeat Prefixes and More String Instructions</title>
    </item>
    <item identifier="A.2.3" identifierref="RESOURCE_A.2.3">
        <title>7.3 Character Translation</title>
    </item>
    <item identifier="A.2.4" identifierref="RESOURCE_A.2.4">
        <title>7.4 Converting a 2's Complement Integer to an ASCII String</title>
    </item>
    <imsss:sequencing IDRef="GLOBAL_SEQ_1"/>
</item>
<imsss:sequencing IDRef="GLOBAL_SEQ_1">
    <imsss:sequencingRules>
        <imsss:preConditionRule>
            <imsss:ruleConditions>
                <imsss:ruleCondition condition="objectiveMeasureLessThan"
referencedObjective="advanced1" measureThreshold="0.75"/>
            </imsss:ruleConditions>
            <imsss:ruleAction action="skip"/>
        </imsss:preConditionRule>
    </imsss:sequencingRules>
    <imsss:objectives>
        <imsss:primaryObjective objectiveID="advanced1">
            <!-- Because readNormalizedMeasure is set to true, Objective Measure Status and
Objective Normalized Measure for objective 'advanced1' will be read from objective
'Content' -->
            <imsss:mapInfo targetObjectiveID="Content" readNormalizedMeasure="true"/>
        </imsss:primaryObjective>
    </imsss:objectives>
</imsss:sequencing>
</item>
<imsss:sequencing IDRef="GLOBAL_SEQ_1"/>
</item>
<imsss:sequencing>
    <imsss:controlMode choice="false" flow="true"/>
</imsss:sequencing>
</organization>
</organizations>
<resources>

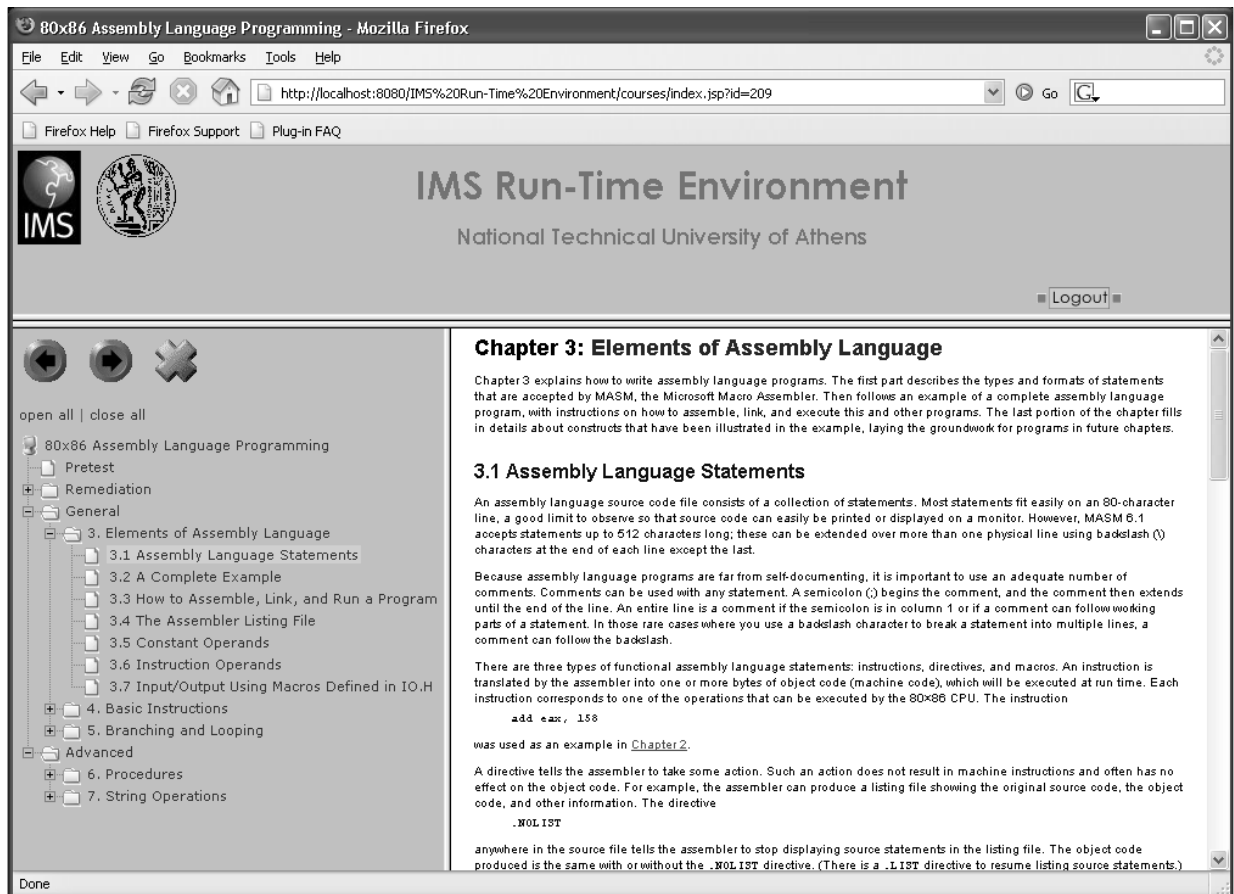
    .....

    .....

</resources>
<imsss:sequencingCollection>
    <imsss:sequencing ID="GLOBAL_SEQ_1">
        <imsss:controlMode choice="true" flow="true"/>
    </imsss:sequencing>
</imsss:sequencingCollection>
</manifest>

```

Τέλος στο σχήμα 4.4 φαίνεται ένα screenshot που δείχνει πώς το μάθημα που σχεδιάσαμε προβάλλεται από το σύστημα :



Σχήμα 4.4 Παρουσίαση του μαθήματος από το σύστημα

5 Επίλογος

5.1 Συμπεράσματα

Η αξία των προδιαγραφών για τα συστήματα μαθησιακής τεχνολογίας είναι πολύπλευρη. Πρώτα από όλα η ύπαρξη ενός κοινού προτύπου για την οργάνωση και διανομή του μαθησιακού υλικού επιτρέπει την χρήση του ίδιου μαθησιακού υλικού από διαφορετικά LMS με συνεπή και προβλέψιμο τρόπο και συμβάλλει στη διαλειτουργισμότητα (interoperability) των LMS.

Δεύτερον, μια προδιαγραφή περιγράφει μια ελεγμένη και αποδοτική λύση σε ένα πρόβλημα που είναι κοινό σε όλα τα LMS (π.χ. το sequencing των μαθησιακών δραστηριοτήτων), οπότε απαλλάσσει τους σχεδιαστές των LMS από την ανάγκη να «εφευρίσκουν τον τροχό» από την αρχή κάθε φορά που σχεδιάζουν ένα τέτοιο σύστημα.

Τρίτον, οι προδιαγραφές μπορούν να συμβάλλουν στην επαναχρησιμοποίηση του κώδικα και στη μείωση του κόστους υλοποίησης ενός LMS γιατί μια προδιαγραφή μπορεί να υλοποιηθεί ως μια open source βιβλιοθήκη η οποία θα μπορεί να χρησιμοποιείται ελεύθερα από οποιονδήποτε θέλει να κατασκευάσει ένα LMS.

Η διαδικασία παραγωγής προδιαγραφών μαθησιακής τεχνολογίας ξεκίνησε μόλις πριν λίγα χρόνια και σίγουρα υπάρχει αρκετός δρόμος μέχρι οι υπάρχουσες προδιαγραφές να ωριμάσουν, να δοκιμαστούν και τελικά να υιοθετηθούν ευρέως από την e-Learning κοινότητα. Στη διαδικασία ωρίμανσης των προδιαγραφών σημαντικότατο ρόλο παίζει η δημιουργία δοκιμαστικών υλοποιήσεων έτσι ώστε οι προδιαγραφές να δοκιμάζονται στην πράξη και να αξιολογείται το πόσο καλά λύνουν το πρόβλημα για τη λύση του οποίου δημιουργήθηκαν και να εντοπίζονται πιθανά λάθη.

Για αυτό και η σημασία του συστήματος που υλοποιήσαμε είναι μεγάλη, ειδικά αν λάβει κάποιος υπόψιν του ότι τη παρούσα χρονική στιγμή το σύστημά μας είναι το μοναδικό (από όσο είμαστε σε θέση να γνωρίζουμε) που περιλαμβάνει μια (σχεδόν πλήρη) υλοποίηση του IMS Simple Sequencing Specification. Συγκεκριμένα, παρόμοια συστήματα με το δικό μας είναι αυτή τη στιγμή το *SCORM Sample Run-Time Environment* (<http://www.adlnet.org/downloads/197.cfm>) και το *RELOAD SCORM Player* (www.reload.ac.uk). Όμως το πρώτο έχει μια κάπως πρόχειρη υλοποίηση του Simple Sequencing, η οποία, όπως διαπιστώσαμε, δεν δουλεύει, ενώ το δεύτερο δεν έχει ακόμη καμία υλοποίηση.

Υλοποιώντας τις προδιαγραφές IMS Content Packaging και IMS Simple Sequencing εντοπίσαμε κάποια λάθη σε αυτές, τα οποία αναφέρονται στο Παράρτημα Β. Επίσης πολλά λάθη υπήρχαν στα παραδείγματα που συνοδεύουν το Simple Sequencing

specification. Αυτό είναι ένα χαρακτηριστικό παράδειγμα του πώς η υλοποίηση μιας προδιαγραφής μπορεί να συμβάλλει στη βελτίωσή της.

Τέλος, μέσα από τη διεξοδική μελέτη και κατανόηση του IMS Simple Sequencing Specification έγινε σαφές ότι εξαιτίας της πολυπλοκότητάς του είναι πολύ σημαντικό να αναπτυχθούν εργαλεία που θα διευκολύνουν τη δημιουργία (authoring) μαθησιακού υλικού που περιλαμβάνει πληροφορίες sequencing, χωρίς να απαιτείται ο χρήστης τους να γνωρίζει τις λεπτομέρειες του specification.

5.2 Μελλοντικές Επεκτάσεις

Όπως αναφέρθηκε στην προηγούμενη παράγραφο, ένας σημαντικό αντικείμενο της μελλοντικής έρευνας είναι η ανάπτυξη authoring εργαλείων για μαθησιακό υλικό με υποστήριξη για Simple Sequencing. Ο στόχος είναι να μπορεί κάποιος σχετικά εύκολα να δημιουργεί content packages που περιλαμβάνουν sequencing πληροφορία, χωρίς να απαιτείται να έχει γνώση του –ιδιαίτερος πολύπλοκου- Simple Sequencing Specification. Για να γίνει αυτό πρέπει να επινοηθεί κάποια γραφική γλώσσα, π.χ. κάποια επέκταση της UML, η οποία θα επιτρέπει την περιγραφή της sequencing λογικής με γραφικό τρόπο. Η γλώσσα πρέπει να είναι απλή και εύκολη στην εκμάθηση, μιας και τα authoring εργαλεία θα απευθύνονται και σε ανθρώπους που δεν έχουν ιδιαίτερες τεχνικές γνώσεις, π.χ. εκπαιδευτικούς. Εκτός από τη γλώσσα πρέπει παράλληλα να δημιουργηθούν αλγόριθμοι που θα κάνουν τη μετάφραση από τη γραφική περιγραφή στην XML περιγραφή που ορίζει το specification [SS 03b]. Προς αυτή την κατεύθυνση κινούνται προσπάθειες όπως ο RELOAD content packaging editor [<http://www.reload.ac.uk>] καθώς και η μέθοδος CADMOS που περιγράφεται στο [RETAL 05].

Ένας άλλος στόχος είναι η επέκταση του υπάρχοντος συστήματος με επιπλέον χαρακτηριστικά καθώς και η υλοποίηση άλλων προδιαγραφών μαθησιακών τεχνολογιών. Μια τέτοια προδιαγραφή είναι το *SCORM Run-Time Environment Model* [SCORM RTE], το οποίο ορίζει ένα API (βασισμένο στο [IEEE 1484.11.2-2003]) και ένα Data Model (βασισμένο στο [IEEE P1484.11.1]) για την επικοινωνία των content objects (που στα πλαίσια του SCORM ονομάζονται *Sharable Content Objects (SCOs)* και *Assets*) με το LMS. Το API είναι ένα σύνολο από JavaScript συναρτήσεις (π.χ. Initialize(), GetValue(), SetValue(),...) τις οποίες καλεί ένα SCO για την μεταφορά δεδομένων από και προς το LMS (data-transfer methods), την έναρξη και τερματισμό της επικοινωνίας του με το LMS (session methods) και το χειρισμό λαθών (support methods).

Μια άλλη δυνατότητα επέκτασης του συστήματος είναι η ανάπτυξη ενός *Learner Information System* και η ολοκλήρωσή του με το παρόν σύστημα. Τα Learner Information Systems και η αντίστοιχη προδιαγραφή *IMS Learner Information Package Specification* έχουν παρουσιαστεί στην παράγραφο 2.3.2.

Τέλος, μια ακόμα δυνατότητα επέκτασης του παρόντος συστήματος θα μπορούσε να είναι η υλοποίηση ενός QTI υποσυστήματος για την επεξεργασία και παρουσίαση online αξιολογήσεων (assessments) σύμφωνα με τη προδιαγραφή *IMS Question & Test Interoperability (QTI)*, η οποία έχει παρουσιαστεί στην παράγραφο 2.3.3.

Παράρτημα Α

Κατηγοριοποίηση των LMS και των χαρακτηριστικών τους³

Κατηγορίες των Learning Management Systems (LMSs) :

- **Γενικά Συστήματα (General systems)**, τα οποία έχουν έναν αριθμό εργαλείων για τη δημιουργία και διαχείριση μαθημάτων και δεν δίνουν έμφαση σε κάποιο συγκεκριμένο σύνολο χαρακτηριστικών.
- **Συστήματα Υποστήριξης Συνεργατικής Μάθησης (Collaborative learning support systems)**, τα οποία εστιάζουν στη δημιουργία και διαχείριση ομάδων μαθητών (team building, student group management) και στη παροχή εργαλείων που επιτρέπουν τη σύγχρονη και ασύγχρονη συνεργασία (synchronous and asynchronous collaboration) υποστηρίζοντας έτσι τις προαναφερθείσες δραστηριότητες.
- **Συστήματα Συγγραφής και Διαχείρισης Εξετάσεων (Question and test authoring and management systems)**, τα οποία επιτρέπουν το σχεδιασμό και τη δημιουργία εξετάσεων (tests) και κουίζ, τα οποία είναι προσβάσιμα μέσω του Διαδικτύου και συμπληρώνονται on-line. Τα συστήματα αυτά παρέχουν εργαλεία για τη δημιουργία εξετάσεων και την on-line παράδοσή τους, καθώς και για την αυτόματη βαθμολόγηση, τη διαχείριση των αποτελεσμάτων και την παραγωγή αναφορών.
- **Συστήματα Διαχείρισης Ανθρώπινων Πόρων και Οργανισμών (People and Institute resources management systems)**, τα οποία αφορούν τη διαχείριση του προσωπικού και των οικονομικών ενός οργανισμού.
- **Εικονικές Τάξεις (Virtual classrooms)**, τα οποία παρέχουν έναν εικονικό χώρο για ζωντανή (live) αλληλεπίδραση όλων των συμμετεχόντων μιας μαθησιακής διαδικασίας, δηλ. των εκπαιδευτικών και των μαθητών.

Κατηγορίες χαρακτηριστικών (features) των LMSs :

- Course management
 - Syllabus Template
 - Calendar
 - Announcements
 - Glossary
 - Structure course access pages
 - Customizable-consistent course interface
 - Links to web resources

³ Η κατηγοριοποίηση αυτή έχει προταθεί στο [APR 01].

- Class Management
 - Create assignments
 - Organize students in groups for team working on assignment projects
 - Assignments grading
 - Assignments grades statistical analysis
 - Students' progress tracking
- Assessment tools
 - Test content authoring tools/wizards
 - Test content upload
 - Automatic, scheduled test delivery
 - Automatic test scoring
 - Automatic generation of results' statistical analysis reports
 - Advanced security for increased examination procedure's reliability
 - Automatic generation of quizzes from a pool
 - Adaptive and personalized quiz presentation
- Communication tools
 - E-mail
 - Chat
 - Discussion forums
 - Video conference
 - Voice conference
 - Shared white board
 - Application Sharing
 - Sessions scheduling
 - File transfer utilities
- Student Tools
 - Add Bookmarks
 - Add Personal notes
 - Get information about study progress
 - Self-assessment quizzes
 - Adaptive/personalized quiz presentation
 - Search-in-content tools
 - Self registration in courses
 - Personal web page creation
 - View his own grades
 - Groups of students can share resources to facilitate team working
- Content management
 - Content upload
 - Content authoring
 - Content structuring
 - Multimedia content insertion
 - Other (than html) content types supported
 - Tools for effective navigation inside content
 - No HTML knowledge required for content authoring
 - Tools/wizard for content authoring
 - Use of templates – reusability of content appearance
 - Tools/wizards for content publishing
 - Meta-data specification
- School-Organization Management
 - Student admissions management

- Human resources management
- School financial management support
- Student mentoring – Advising for courses selection and prerequisites

Κατάταξη των LMSs σε κατηγορίες ανάλογα με το ποιές κατηγορίες χαρακτηριστικών υποστηρίζουν :

LTS Categories	General Systems	Collaborative Learning Support Systems	Virtual Classrooms	Question and Test Authoring & Management Systems	People and Institute Resources Management Systems
Course Management	X				X
Class Management	X	X	X		X
Communication Tools	X	X	X		
Student Tools	X	X	X	X	
Content Management	X			X	X
Assessment Tools	X			X	
School-Management	X				X

Πίνακας Α.1 Κατηγοριοποίηση των LMSs με βάση τα χαρακτηριστικά που υποστηρίζουν

Παράρτημα Β

Τεχνολογίες που χρησιμοποιήθηκαν

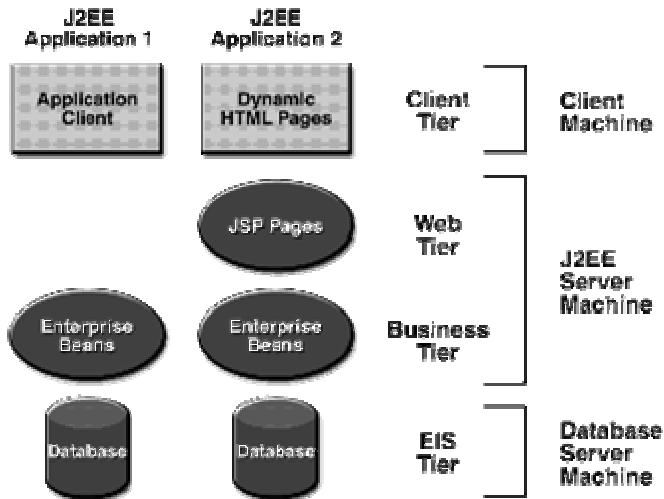
B.1 Java 2 Platform, Enterprise Edition (J2EE)

Η πλατφόρμα J2EE είναι ένα σύνολο τεχνολογιών, στο πλαίσιο της γλώσσας προγραμματισμού Java, οι οποίες χρησιμοποιούνται για την ανάπτυξη κατακεντρωμένων επιχειρησιακών εφαρμογών (distributed enterprise applications). Κάποια από τα χαρακτηριστικά του J2EE είναι :

- κατακεντρωμένο (distributed) και πολλαπλών βαθμίδων (multitiered) μοντέλο εφαρμογής
- reusable components
- unified security model
- flexible transaction control
- υποστήριξη για web services με βάση την γλώσσα XML
- platform-independent

Η πλατφόρμα J2EE χρησιμοποιεί ένα distributed και multitiered μοντέλο εφαρμογής, σύμφωνα με το οποίο η εφαρμογή χωρίζεται σε components ανάλογα με τη λειτουργία τους, και τα διάφορα components εγκαθίστανται σε διαφορετικές μηχανές ανάλογα με το tier (βαθμίδα) στο οποίο ανήκουν. Το σχήμα B.1 δείχνει δύο J2EE applications που αποτελούνται από τα παρακάτω tiers :

- Client-tier : τα components που εκτελούνται στη μηχανή του client.
- Web-tier : στην περίπτωση που ο client είναι ένας browser, στο web tier υπάρχουν τα web components που εκτελούνται στον J2EE server και παράγουν τις δυναμικές HTML σελίδες που στέλνονται στον browser.
- Business-tier : components που εκτελούνται στον the J2EE server και περιέχουν τις λειτουργίες που αφορούν το συγκεκριμένο business domain της εφαρμογής.
- Enterprise information system (EIS)-tier : το επιχειρησιακό πληροφοριακό σύστημα (EIS).



Σχήμα B.1 Multitiered J2EE Applications

Το J2EE αποτελείται από πάρα πολλές τεχνολογίες οι οποίες είναι αδύνατον να περιγραφούν εδώ. Ο αναγνώστης παραπέμπεται στη διεύθυνση <http://java.sun.com/j2ee> για περισσότερες πληροφορίες. Εδώ θα κάνουμε μόνο μια σύντομη αναφορά σε τρεις τεχνολογίες του J2EE που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος IMS Run-Time Environment : Τα *Servlets*, τις *JavaServer Pages* (JSP) και την *JSP Standard Tag Library* (JSTL).

B.1.1 Servlets

Ένα servlet είναι μια Java κλάση η οποία εκτελείται σε έναν web server και αντιστοιχεί σε κάποιο συγκεκριμένο URL. Όταν ο server λάβει ένα HTTP request για το URL ενός servlet τότε καλεί κάποια ειδική μέθοδο του servlet η οποία εξυπηρετεί το HTTP request παράγοντας ένα HTTP response που στέλνεται πίσω σε αυτόν που έκανε το request. Το HTTP response είναι πρακτικά ένα HTML ή XML έγγραφο. Ένας web server που υποστηρίζει την τεχνολογία των servlets αναφέρεται ως servlet container.

Στο σχήμα B.2 φαίνεται ένα πολύ απλό παράδειγμα ενός servlet. Η μέθοδος doGet καλείται από τον servlet container όταν λάβει ένα HTTP request με τη μέθοδο GET για το URL στο οποίο αντιστοιχεί το servlet. Ως παράμετροι της μεθόδου περνώνται δύο αντικείμενα, το request και το response, που αντιπροσωπεύουν το HTTP request και το HTTP response αντίστοιχα. Όπως βλέπουμε, το servlet γράφει στο response HTML κώδικα, ο οποίος στέλνεται στον browser του client. Το αποτέλεσμα είναι στον browser να εμφανιστεί το μήνυμα «Hello World !”.

Ακόμα και μέσα από αυτό το απλοϊκό παράδειγμα φαίνεται το βασικό –και πολύ σοβαρό- μειονέκτημα των servlets : ο HTML κώδικα είναι ενσωματωμένος μέσα στον Java κώδικα, με αποτέλεσμα (1) το πρόγραμμα να είναι δυσανάγνωστο (2) να μπλέκεται η δουλειά του web programmer με τη δουλειά του web designer μιας και οι δύο γράφουν κώδικα στο ίδιο αρχείο (3) κάθε φορά που πρέπει να αλλάξει κάτι στον HTML κώδικα, π.χ. το χρώμα του background της σελίδας, το servlet πρέπει να μεταγλωττιστεί.

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorldServlet
    extends HttpServlet
{
    public void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException
    {
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println(" <head>");
        out.println("  <title>Servlet Example</title>");
        out.println(" </head>");
        out.println(" <body>");
        out.println("  <h1>Hello World !</h1>");
        out.println(" </body>");
        out.println("</html>");
    }
}

```

Σχήμα Β.2 Παράδειγμα servlet

B.1.2 JavaServer Pages (JSP)

Τα servlets είναι Java κώδικας που (μπορεί να) περιέχει ενσωματωμένο HTML κώδικα. Οι JSP σελίδες είναι το αντίστροφο : HTML κώδικας στον οποίο υπάρχει ενσωματωμένος Java κώδικας. Ένα παράδειγμα φαίνεται στο σχήμα Β.3.

```

<%@ page session="false" %>
<%@ page import="java.util.Calendar" %>

<html>
<head>
  <title>JSP Example</title>
</head>
<body>
  <%
    out.println("Current time: " + Calendar.getInstance().getTime());
  %>
</body>
</html>

```

Σχήμα Β.3 Παράδειγμα JSP

Όπως βλέπουμε στο σχήμα Β.3, η JSP σελίδα μοιάζει με μια απλή HTML σελίδα με τη διαφορά ότι υπάρχει ενσωματωμένος Java κώδικας μεταξύ των <% %> (scriptlets). Μια JSP σελίδα μεταφράζεται αυτόματα σε ένα ισοδύναμο servlet από τον servlet container. Έτσι γλυτώνουμε από το μειονέκτημα (3) των servlets που

αναφέρθηκε πιο πάνω, όμως εξακολουθούν να ισχύουν τα δύο πρώτα μειονεκτήματα, γιατί ο HTML και ο Java κώδικας εξακολουθούν να συνυπάρχουν. Μια λύση του προβλήματος είναι να συγκεντρώσουμε το μεγαλύτερο μέρος του Java κώδικα μιας JSP σελίδας σε μια Java κλάση και μέσα στη JSP σελίδα απλά να καλούμε μεθόδους της κλάσης αυτής. Στην περίπτωση αυτή, η Java κλάση αναφέρεται ως **JSP Bean**. Με κατάλληλη χρήση των JSP Beans μπορεί να ελαχιστοποιηθεί ο Java κώδικας που βρίσκεται μέσα σε μια HTML σελίδα.

Στο σχήμα B.4 βλέπουμε ένα παράδειγμα JSP σελίδας που χρησιμοποιεί την τεχνική των JSP Beans. Πρόκειται για μια τυπική σελίδα login, όπου ο χρήστης καλείται να εισάγει το username και το password του προκειμένου να του επιτραπεί η πρόσβαση σε κάποιο τμήμα του συστήματος. Το ζεύγος (username, password) πρέπει να ελεγχτεί ως προς την εγκυρότητά του. Συνήθως αυτό γίνεται εξετάζοντας κάποιο table μιας Βάσης Δεδομένων το οποίο περιέχει όλα τα έγκυρα ζεύγη (username, password). Σε κάθε περίπτωση, ο κώδικας που κάνει τον έλεγχο της εγκυρότητας δεν υπάρχει λόγος να είναι μέσα στη JSP σελίδα, αλλά αντιθέτως υπάρχουν σοβαροί λόγοι να μην είναι, γιατί, εκτός από το θέμα της αναγνωσιμότητας και της ανεξαρτησίας της δουλειάς του web programmer από τη δουλειά του web designer, μπορεί να θέλουμε να χρησιμοποιήσουμε τον ίδιο κώδικα και μέσα σε άλλες JSP σελίδες (code reusability). Έτσι η ορθή τακτική είναι να γράψουμε ένα bean που κάνει τον έλεγχο της εγκυρότητας και να το χρησιμοποιήσουμε μέσα στη JSP σελίδα. Το bean είναι η κλάση LoginManager, η οποία φαίνεται στο σχήμα B.5.

Περισσότερες πληροφορίες για τις JSP σελίδες μπορούν να αναζητηθούν στο **[JSP 2.0]**.


```

<%@ page language="java" session="true" %>

<jsp:useBean id="LoginManager" scope="session" class="softeng.jsp_beans.LoginManager"/>
<jsp:setProperty name="LoginManager" property="*/>

<%! int i; %>

<% i = LoginManager.login();
   if (i == 0) // login successfull
   {
       session.setAttribute("loggedIn", new String("true"));
       response.sendRedirect("admin.jsp");
   }
%>

<%@ include file="../Templates/TemplateHeader.html" %>

<CENTER>
<BR><BR><H2>Login Page</H2><BR>

<%
   if (i == 4)
       out.println("<B>Login failed. The (username,password) pair you entered is incorrect.</B>");
%>

<BR>Please enter your username and password.<BR>
<BR><FORM ACTION="login.jsp" METHOD=POST>
<TABLE>
<TR>
<TD>User Name:</TD>
<TD><INPUT TYPE=TEXT NAME=username></TD>
</TR>
<TR>
<TD>Password:</TD>
<TD><INPUT TYPE=PASSWORD NAME=password></TD>
</TR>
<TR>
<TD ALIGN=RIGHT COLSPAN=2><INPUT TYPE=SUBMIT VALUE=Login></TD>
</TR>
</TABLE>
</FORM>
</CENTER>

<%@include file="../Templates/TemplateFooter.html" %>

```

Σχήμα Β.4 Παράδειγμα JSP με χρήση JSP Beans (login.jsp)

```

package softeng.jsp_beans;

// native java imports
import java.sql.Connection;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.servlet.http.HttpSession;

// project imports
import softeng.utils.StringUtil;

public class LoginManager
{
    private static final String connectionUrl = "jdbc:mysql:///softeng";
    private String username;
    private String password;

    public void setUsername(String usnm)
    {
        username = StringUtil.fixSqlFieldValue(usnm);
    }

    public void setPassword(String pwd)
    {
        password = StringUtil.fixSqlFieldValue(pwd);
    }

    /**
     * login
     *
     * @return int : <BR>
     * 0 -> success <BR>
     * 1 -> no username/password supplied <BR>
     * 2 -> database connection error <BR>
     * 3 -> SQLException <BR>
     * 4 -> incorrect (username,password)
     */
    public int login()
    {
        if (username == null || password == null)
        {
            return 1;
        }
        Connection con = open_DB_Connection();
        if (con == null)
        {
            System.out.println("ERROR - Unable to connect to database.");
            return 2;
        }

        int result = 3;
        try
        {
            Statement s = con.createStatement();
            String sql = "SELECT username FROM administrator WHERE username =\'" + username +
                "\' AND password =\'" + password + "\'";

```

```

// We don't need username and password anymore, so reset them to null
setUsername(null);
setPassword(null);
ResultSet rs = s.executeQuery(sql);
if (rs.next())
{
    result = 0;
}
else
{
    result = 4;
}
rs.close();
s.close();
con.close();
}
catch (SQLException e)
{
    System.out.println(e.toString());
}
return result;
}

public String logout(HttpSession session)
{
    try
    {
        session.invalidate();
    }
    catch (IllegalStateException ise)
    {
        return ise.toString();
    }
    return "Logout successfull.";
}

private Connection open_DB_Connection()
{
    Connection con = null;
    try
    {
        con = DriverManager.getConnection(connectionUrl);
        System.out.println("Got connection.");
    }
    catch (SQLException e)
    {
        System.out.println(e.toString());
    }
    catch (Exception e)
    {
        System.out.println(e.toString());
    }
}
return con;
}
}

```

Σχήμα Β.5 Παράδειγμα JSP με χρήση JSP Beans (LoginManager.java)

B.1.3 JSP Standard Tag Library (JSTL)

Η JSTL είναι μια βιβλιοθήκη από JSP tags, όμοια με τα HTML tags, με τα οποία μπορούν να γίνονται ενέργειες όπως π.χ. loops και if-else αποφάσεις. Το πλεονέκτημα των JSTL tags σε σύγκριση με τον ισοδύναμο Java scriptlet κώδικα είναι ότι με τα tags η JSP σελίδα είναι αρκετά πιο ευανάγνωστη. Έτσι, συνδυάζοντας JSP Beans με την JSTL πετυχαίνουμε το μέγιστο διαχωρισμό του presentation (HTML) από τη λογική (Java). Ένα παράδειγμα JSTL φαίνεται στο σχήμα B.4. Με χρήση του <c:forEach> action η σελίδα κάνει ένα loop πάνω στις παραμέτρους που έχουν περαστεί στην JSP σελίδα και με χρήση του <c:out> τις παρουσιάζει στον browser.

```
<%@ page language="java" session="true" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%-- Loop over the JSTL param implicit object,
      which is a map --%>

<c:forEach items='${param}' var='parameter'>

  <ul>

    <%-- Display the key of the current item, which
          corresponds to the name of the init param --%>
    <li>Name: <c:out value='${parameter.key}'/></li>

    <%-- Display the value of the current item, which
          corresponds to the value of the init param --%>
    <li>Value: <c:out value='${parameter.value}'/></li>

  </ul>

</c:forEach>
```

Σχήμα B.6 Παράδειγμα JSTL

Η JSTL ορίζεται στο **[JSTL]**.

B.2 XML

Θα κάνουμε τώρα μια σύντομη εισαγωγή στη γλώσσα XML. Η XML ορίζεται στο [XML SPEC], ενώ ένα καλό εισαγωγικό βιβλίο είναι το [HAROLD02].

B.2.1 Εισαγωγή

Σε ένα κόσμο όπου οι πληροφορίες παρέχονται μέσω του παγκόσμιου διαδικτύου, τα έγγραφα πρέπει να είναι εύκολα προσβάσιμα, μεταφέρσιμα και ευέλικτα. Πρέπει επίσης να είναι ανεξάρτητα οποιουδήποτε συστήματος και περιεχομένου. Οι γενικευμένες γλώσσες έχουν τέτοια χαρακτηριστικά, παρέχοντας στα έγγραφα αυτά μια δυνατότητα η οποία δεν υπάρχει σε άλλες γλώσσες περιγραφής εγγράφων. Η HTML είναι προβληματική και περιοριστική γλώσσα. Η XML έλυσε πολλά από τα προβλήματα που αντιμετώπισαν οι σχεδιαστές του web και είναι υπεύθυνη για την XHTML, μια ανασχεδιασμένη HTML. Θα χρησιμοποιείται για πολλά χρόνια επειδή προσφέρει αποτελεσματικές και δυναμικές πολυμεσικές λύσεις.

Η XML σχεδιάστηκε να ικανοποιήσει πολλές ανάγκες δίνοντας στα έγγραφα ένα μεγαλύτερο επίπεδο προσαρμοστικότητας στο στυλ και τη δομή από αυτό που υπήρχε παλαιότερα στην HTML. Η XML προσφέρει στους σχεδιαστές της HTML τη δυνατότητα να προσθέτουν περισσότερα στοιχεία στη γλώσσα. Δεν αναφέρεται μονάχα στους σχεδιαστές του web αλλά σε οποιονδήποτε ασχολείται με εκδόσεις.

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

B.2.2 Τι είναι η XML;

- Η XML είναι συντομογραφία για το EXtensible Markup Language.
- Η XML είναι **markup** γλώσσα, η οποία μοιάζει πολύ με την HTML.
- Η XML σχεδιάστηκε για να **περιγράφει δεδομένα**.
- Τα XML tags δεν είναι προκαθορισμένα. Πρέπει να ορίσουμε τα **δικά μας tags**.
- Η XML χρησιμοποιεί το **Document Type Definition (DTD)** ή το **XML Schema** για να περιγράψει τα δεδομένα.
- Ένα XML κείμενο με ένα DTD ή ένα XML Schema σχεδιάστηκε για να **περιγράφει επαρκώς τον εαυτό του**.

B.2.3 Διαφορές μεταξύ HTML και XML

- **Η XML σχεδιάστηκε για τη μεταφορά δεδομένων.**
- Η XML δεν είναι ένα υποκατάστατο της HTML.
- Η XML σχεδιάστηκε για να **περιγράφει** τα δεδομένα και εστιάζει στο **τι είναι** τα δεδομένα.

- Η HTML σχεδιάστηκε για να **παρουσιάζει** τα δεδομένα και εστιάζει στο **πώς φαίνονται** τα δεδομένα.
- Η HTML έχει να κάνει με την παρουσίαση των δεδομένων ενώ η XML έχει να κάνει με την περιγραφή των δεδομένων.

B.2.4 Η XML δεν κάνει ΤΙΠΟΤΑ

Μπορεί να είναι λίγο δύσκολο να το καταλάβει κανείς, αλλά η XML δεν κάνει ΤΙΠΟΤΑ. Η XML σχεδιάστηκε για τη δόμηση, την αποθήκευση και την αποστολή δεδομένων. Το παρακάτω παράδειγμα είναι ένα σημείωμα της Mary για τον John, αποθηκευμένο σε XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>John</to>
  <from>Mary</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Σχήμα B.7 Παράδειγμα XML

Το σημείωμα έχει μια κεφαλή (heading) και ένα σώμα (body), το οποίο περιέχει το μήνυμα. Περιέχει επίσης πληροφορίες για τον αποστολέα και τον αποδέκτη. Παρ' όλα αυτά όμως το XML κείμενο εξακολουθεί να μην κάνει ΤΙΠΟΤΑ. Περιέχει απλά πληροφορίες, τοποθετημένες μέσα σε XML tags. Κάποιος πρέπει να γράψει ένα κομμάτι software για να το στείλει, να το λάβει ή να το παρουσιάσει.

B.2.5 Η XML είναι απλή και επεκτάσιμη

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. Για παράδειγμα τα tags <to> και <from> στο παραπάνω παράδειγμα δεν είναι ορισμένα σε κανένα XML standard. Αυτά τα tags «επινοήθηκαν» από το συγγραφέα αυτού του XML κειμένου.

B.2.6 Η XML είναι ένα συμπλήρωμα της HTML

Είναι σημαντικό να καταλάβει κανείς ότι η XML δεν αποτελεί ένα υποκατάστατο της HTML. Είναι πολύ πιθανόν ότι οι μελλοντικές Web εφαρμογές θα χρησιμοποιούν την XML για να περιγράψουν τα δεδομένα, ενώ η HTML και συναφείς τεχνολογίες όπως CSS, XSL-FO θα χρησιμοποιούνται για την απεικόνιση των δεδομένων αυτών.

Για αυτό και η καλύτερη περιγραφή της XML είναι η εξής: «η XML είναι ένα διαπлатοφορμικό, ανεξάρτητο από software και hardware εργαλείο για την μετάδοση πληροφοριών».

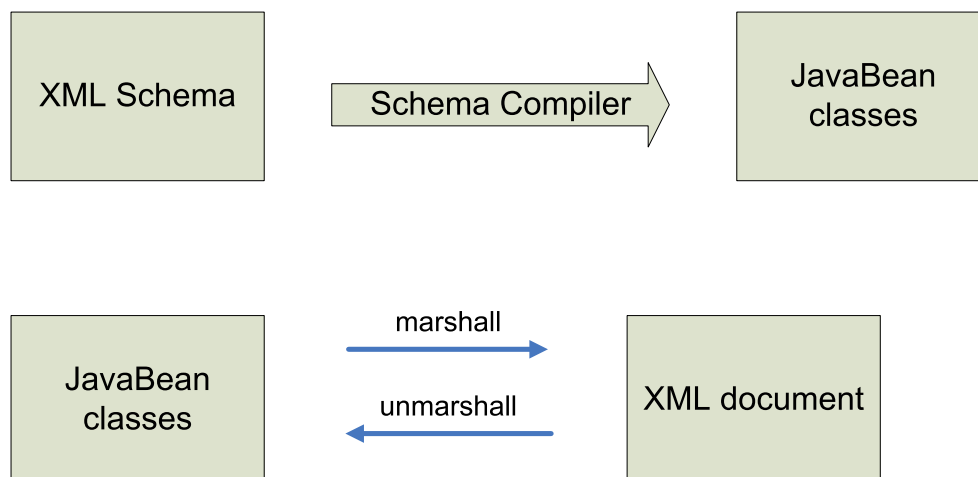
B.2.7 Η XML στην ανάπτυξη μελλοντικών Web εφαρμογών

Είναι συναρπαστικό το πόσο γρήγορα το XML standard αναπτύχθηκε και πόσο γρήγορα ένας μεγάλος αριθμός από σχεδιαστές λειτουργικών συστημάτων υιοθέτησαν το standard αυτό. Πιστεύεται ότι η XML θα είναι τόσο σημαντική για το μέλλον του Web όσο ήταν η HTML για τη θεμελίωση του. Πιστεύεται επίσης ότι η XML θα είναι το πιο κοινό εργαλείο για την επεξεργασία των δεδομένων και τη μετάδοση τους.

B.3 XMLBeans

Το XMLBeans είναι ένα open source εργαλείο για XML-Java data binding. Με τον όρο αυτό εννοούμε την προσπέλαση των περιεχομένων ενός XML εγγράφου χρησιμοποιώντας Java κλάσεις που αντικατοπτρίζουν τη δομή και «ενθυλακώνουν» τα περιεχόμενα του XML εγγράφου. Αντίθετα από τα δύο κύρια XML APIs, το DOM (Document Object Model) και το SAX (Simple API for XML), τα οποία επικεντρώνονται στη δομή ενός XML εγγράφου, τα XML databinding frameworks, όπως είναι το XMLBeans, εστιάζουν στα δεδομένα που ορίζονται μέσα στο XML έγγραφο παρέχοντας πρόσβαση σε αυτά μέσα από ένα object model το οποίο αντιπροσωπεύει τα δεδομένα.

Το object model, δηλαδή το σύνολο των Java κλάσεων μέσα από τις οποίες προσπελούνται τα δεδομένα, παράγεται από το XML Schema που ορίζει το συντακτικό του XML εγγράφου. Η διαδικασία αυτή ονομάζεται schema compilation και στην περίπτωση των XML Beans γίνεται με το εργαλείο scomp που μας παρέχουν. Οι παραχθείσες κλάσεις μπορούν στη συνέχεια να χρησιμοποιηθούν για ανάγνωση (unmarshall) ή εγγραφή (marshall) του XML εγγράφου.



Σχήμα B.8 XML-Java data binding

Τα παραπάνω μπορούν να γίνουν πιο κατανοητά με το ακόλουθο παράδειγμα. Έστω ότι έχουμε το ακόλουθο XML Schema :

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="students">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="student" type="student" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="student">
    <xs:sequence>
      <xs:element name="name" type="name"/>
    </xs:sequence>
    <xs:attribute name="gender" type="gender"/>
  </xs:complexType>

  <xs:complexType name="name">
    <xs:sequence>
      <xs:element name="firstName" type="xs:string"/>
      <xs:element name="lastName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="gender">
    <xs:restriction base="xs:string">
      <xs:enumeration value="male"/>
      <xs:enumeration value="female"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Σχήμα Β.9 XML Beans Παράδειγμα (XML Schema)

Ένα XML έγγραφο που είναι έγκυρο ως προς το παραπάνω schema είναι το εξής :

```
<?xml version="1.0" encoding="UTF-8"?>
<students xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation=" xmlbeans_example.xsd">

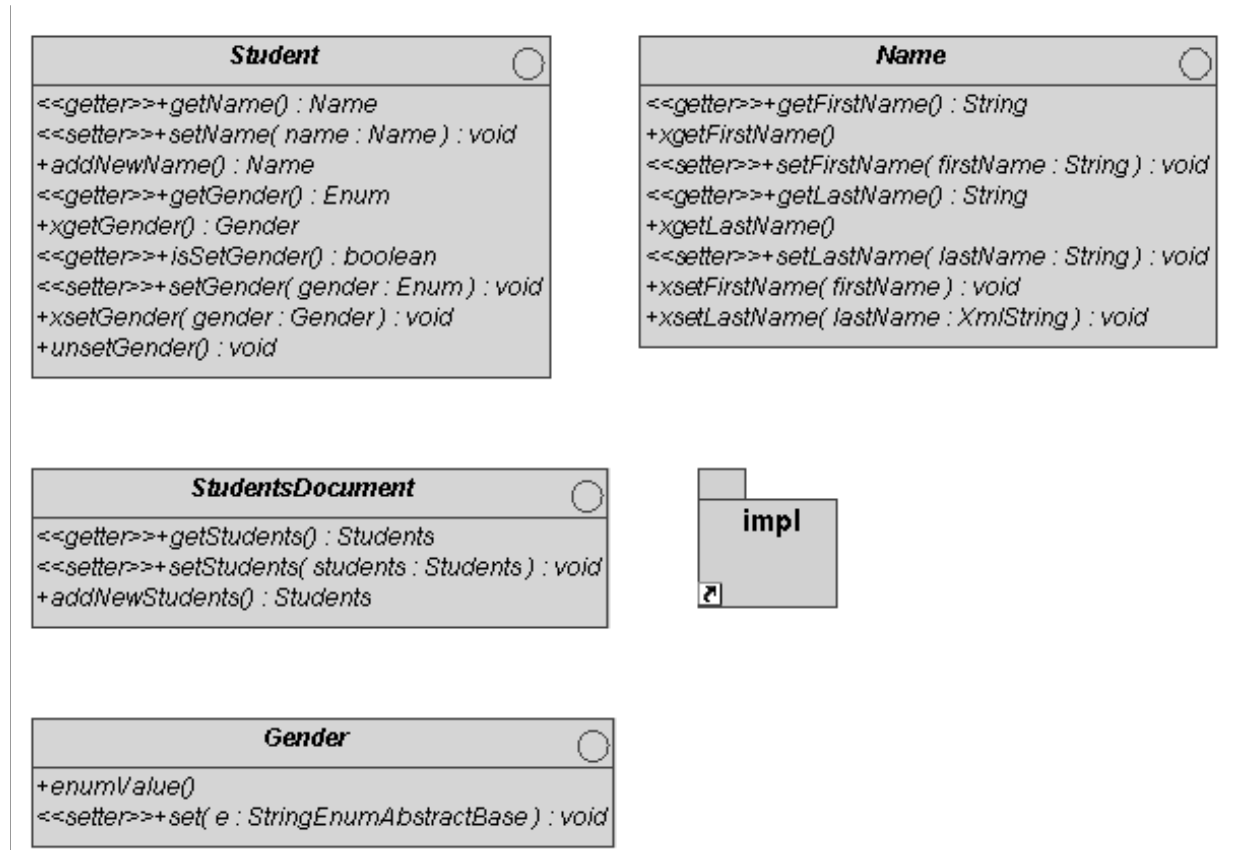
  <student gender="male">
    <name>
      <firstName>Michael</firstName>
      <lastName>Papadopoulos</lastName>
    </name>
  </student>

  <student gender="female">
    <name>
      <firstName>Maria</firstName>
      <lastName>Ioannou</lastName>
    </name>
  </student>

</students>
```

Σχήμα Β.10 XML Beans Παράδειγμα (XML document)

Μεταγλωττίζοντας το παραπάνω XML Schema με τον scomp (XML Beans Schema Compiler) λαμβάνουμε ένα σύνολο από Java interfaces και Java κλάσεις που υλοποιούν αυτά τα interfaces, τα οποία μπορούμε να χρησιμοποιήσουμε για την προσπέλαση των δεδομένων του XML εγγράφου. Τα interfaces που παράγονται φαίνονται στο παρακάτω διάγραμμα :



Σχήμα B.11 XML Beans Παράδειγμα (XML Beans generated interfaces)

Περισσότερες πληροφορίες για τα XML Beans μπορούν να βρεθούν στο site <http://xmlbeans.apache.org>

Παράρτημα Γ

Λάθη που εντοπίστηκαν στις προδιαγραφές

Κατά την υλοποίηση των προδιαγραφών διαπιστώθηκε ότι αυτές έχουν κάποια λάθη. Τα λάθη αυτά παρουσιάζονται στη συνέχεια.

Γ.1 IMS Content Packaging Specification

Στο XML Schema του IMS Content Packaging Specification (αρχείο **imscp_v1p1.xsd**) όλα τα elements δηλώνονται στο global scope, δηλαδή είναι παιδιά του root element `<xsd:schema>`. Αυτό σημαίνει ότι κάθε element μπορεί να εμφανίζεται ως root element σε ένα instance document, π.χ. σύμφωνα με το schema το παρακάτω document είναι ένα έγκυρο IMS Manifest :

```
<?xml version="1.0" encoding="UTF-8"?>
<resources xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd ">
  <resource identifier="RESOURCE1" type="webcontent">
    <file href="content.html"/>
  </resource>
  <resource identifier="RESOURCE2" href="posttest.html" type="webcontent">
    <file href="posttest.html"/>
    <file href="ssUpdate.js"/>
  </resource>
  <resource identifier="RESOURCE3" type="webcontent">
    <file href="remediation.html"/>
  </resource>
</resources>
```

Σχήμα Γ.1 – Ένα «λάθος» IMS Manifest που όμως θεωρείται έγκυρο με βάση το υπάρχον Content Packaging schema

Προφανώς το παραπάνω document ΔΕΝ είναι ένα έγκυρο IMS Manifest αφού περιλαμβάνει μόνο το `<resources>` element. Ένα έγκυρο IMS Manifest έχει πάντα ένα `<manifest>` element ως root, και μέσα σε αυτό πρέπει να βρίσκεται ακριβώς ένα `<organizations>` και ακριβώς ένα `<resources>` element.

Το λάθος αυτό στο XML Schema στο Content Packaging οφείλεται μάλλον στο ότι αυτοί που έγραψαν το schema χρησιμοποίησαν κάποιον XML editor ο οποίος αυθαίρετα τοποθέτησε τις δηλώσεις των elements στο global scope.

Το σωστό είναι στο global scope να δηλώνεται μόνο το <manifest> element (αφού μόνο αυτό μπορεί να εμφανιστεί ως root element) και όλα τα άλλα elements να δηλώνονται μέσα στο <manifest> element. Δηλαδή :

```

.....
.....

<!-- ** Element Declarations ** -->
<!-- ***** -->
<xsd:element name="manifest" type="manifestType"/>
<!-- ***** -->
<!-- ** Complex Types ** -->
<!-- ***** -->

.....
.....

<!-- ** manifest ** -->
<!-- ***** -->
<xsd:complexType name="manifestType">
  <xsd:sequence>
    <xsd:element name="metadata" type="metadataType" minOccurs="0"/>
    <xsd:element name="organizations" type="organizationsType"/>
    <xsd:element name="resources" type="resourcesType"/>
    <xsd:element name="manifest" type="manifestType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:group ref="grp.any"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="attr.identifier.req"/>
  <xsd:attributeGroup ref="attr.version"/>
  <xsd:attribute ref="xml:base"/>
  <xsd:anyAttribute namespace="##other" processContents="strict"/>
</xsd:complexType>
<!-- ***** -->

.....
.....

```

Σχήμα Γ.2 – Διόρθωση του Content Packaging schema

Γ.2 IMS Simple Sequencing Specification

Γ.2.1 Λάθος Πρώτο

Στον ψευδοκώδικα της *Choice Sequencing Request Process* (βλ. [SS, 03a] § SB.2.9) στο βήμα 12 γράφει «**Case:** Target activity is forward from the common ancestor activity». Το σωστό είναι να γράφει «**Case:** The target activity is a descendent of the common ancestor».

Γ.2.2 Λάθος Δεύτερο

Στον ψευδοκώδικα της *Flow Tree Traversal Subprocess* (βλ. [SS, 03a] § SB.2.1) στο βήμα 4.3.1 γράφει «**If Sequencing Control Forward Only** for the parent of the activity is **True Or** the activity is the root of the activity tree **Then**».

Το σωστό είναι «**If Sequencing Control Forward Only for the activity is True Then**».

Αυτό είναι ένα σοβαρό λάθος εξαιτίας του οποίου η backwards διάσχιση του activity tree δεν δουλεύει σωστά. Μάλιστα εξαιτίας της πολυπλοκότητας του αλγορίθμου με τον οποίο διασχίζεται το activity tree στο Simple Sequencing δεν είναι καθόλου εύκολο να συνειδητοποιήσει κανείς το πού ακριβώς είναι το λάθος και γιατί είναι λάθος.

Γ.2.3 Λάθος Τρίτο

Στην παράγραφο §SM.2 : *Sequencing Rule Description* του [SS, 03a], η συνθήκη (rule condition) *Objective Measure Known* ορίζεται ως εξής : «Objective Measure Known - evaluates to True if the Objective Progress Status for the objective (TM.1.1) associated with the activity (indicated by Rule Condition Referenced Objective) is True and the Objective Measure Status (TM.1.1) for the objective associated with the activity (indicated by Rule Condition Referenced Objective) is True».

Ο ορισμός αυτός είναι νοηματικά λανθασμένος, αφού το αν ο βαθμός (measure) ενός objective είναι γνωστός ή όχι υποδεικνύεται από το Objective Measure Status και μόνο – το Objective Progress Status είναι άσχετο. Άρα το σωστό είναι η συνθήκη να οριστεί ως εξής : «Objective Measure Known - evaluates to True if the Objective Measure Status (TM.1.1) for the objective associated with the activity (indicated by Rule Condition Referenced Objective) is True».

Γ.2.4 Λάθος Τέταρτο

Στον ψευδοκώδικα της *Flow Tree Traversal Subprocess* (βλ. [SS, 03a] § SB.2.1) στο βήμα 3.2.1 γράφει «**If the activity is the last activity in the activity's parent's list of Available Children (AM.1.1) Then**». Το λάθος είναι ότι θεωρεί δεδομένο ότι το activity έχει πατέρα, ενώ αυτό ισχύει μόνο αν το activity ΔΕΝ είναι το root activity. Αν το activity είναι το root activity, τότε δεν υπάρχει activity-πατέρας, το οποίο προγραμματιστικά μεταφράζεται ως parentActivity == null και θα έχει ως συνέπεια τη δημιουργία ενός NullPointerException.

Το σωστό είναι «**If the activity is not the root activity of the activity tree And is the last activity in the activity's parent's list of Available Children (AM.1.1) Then**».

Βιβλιογραφία

e-Learning Specifications :

- [CP, 03a] *IMS Content Packaging Information Model*, C.Smythe, T.Anderson, M.McKell, A.Cooper, W.Young and C.Moffatt, Version 1.1.3, IMS, June 2003.
- [CP, 03b] *IMS Content Packaging XML Binding*, C.Smythe, T.Anderson, M.McKell, A.Cooper, W.Young and C.Moffatt, Version 1.1.3, IMS, June 2003.
- [CP, 03c] *IMS Content Packaging Best Practice Guide*, C.Smythe, T.Anderson, M.McKell, A.Cooper, W.Young and C.Moffatt, Version 1.1.3, IMS, June 2003.
- [CP, 03d] *IMS Content Packaging Summary of Changes Final Specification 1A*, C.Smythe, Version 1.1.3, IMS, April 2003.
- [SS, 03a] *IMS Simple Sequencing Information and Behavior Model v1.0 Final Specification*, IMS Global Learning Consortium, Inc., March 2003.
- [SS 03b] *IMS Simple Sequencing XML Binding v1.0 Final Specification*, IMS Global Learning Consortium, Inc., March 2003
- [SS, 03c] *IMS Simple Sequencing Best Practice and Implementation Guide v1.0 Final Specification*, IMS Global Learning Consortium, Inc., March 2003.
- [MD, 01a] *IMS Learning Resource Meta-Data Information Model v.1.2.1 Final Specification*, IMS Global Learning Consortium, Inc., September 2001.
- [MD, 01b] *IMS Learning Resource Meta-Data XML Binding Specification v.1.2.1 Final Specification*, IMS Global Learning Consortium, Inc., September 2001.
- [LIP, 01a] *IMS Learner Information Package Information Model Final Specification*, R.Robson, C.Smythe and F.Tansey, Version 1.0, IMS, March 2001
- [LIP, 01b] *IMS Learner Information Package XML Binding Final Specification*, R.Robson, C.Smythe and F.Tansey, Version 1.0, IMS, March 2001
- [QTI 2.0, 05a] *IMS Question and Test Interoperability Information Model, Version 2.0*, IMS Global Learning Consortium, Inc., January 2005
- [QTI 2.0, 05b] *IMS Question and Test Interoperability XML Binding, Version 2.0*, IMS Global Learning Consortium, Inc., January 2005

[QTI 2.0, 05c] *IMS Question and Test Interoperability Integration Guide, Version 2.0*, IMS Global Learning Consortium, Inc., January 2005

[AF GLOSSARY 03] *IMS Abstract Framework: Glossary*, K.Blinco, S.Griffin, J.Merriman, C.Smythe, IMS Global Learning Consortium, Inc., Final Release, July 2003.

Όλα τα παραπάνω έγγραφα είναι διαθέσιμα στο website του IMS Global Learning Consortium: <http://www.imsglobal.org/>

[SCORM RTE] *SCORM Version 1.32004 Run-Time Environment Model Version 1.3*, Advanced Distributed Learning, January 30, 2004 Available at: <http://www.adlnet.org/>

[IEEE P1484.11.1] *IEEE P1484.11.1 Draft 3 Draft Standard for Learning Technology – Data Model for Content Object Communication*. November 10, 2003. Available at: <http://ltsc.ieee.org/>

[IEEE 1484.11.2-2003] *IEEE 1484.11.2-2003 Standard for Learning Technology – ECMAScript Application Programming Interface for Content to Runtime Services Communication*. November 10,2003 Available at: <http://ltsc.ieee.org/>

[IEEE LOM 6.1] *IEEE Learning Object Meta-Data Working Draft, Version 6.1*, available at <http://ltsc.ieee.org/wg12/index.html>

e-Learning Papers & Articles :

[APR 01] P. Avgeriou, A. Papasalouros and S. Retalis, *Learning Technology Systems: issues, trends, challenges*, proceedings of the 1st IOSTE symposium in Southern Europe, Science and Technology Education, Paralimni, Cyprus, May 2001

[AVG 03] P. Avgeriou, *A reference architecture for open Learning Management Systems*, Doctoral Dissertation, National Technical University of Athens, Department of Electrical and Computer Engineering, Division of Computer Science, Software Engineering Laboratory, January 2003

[GOODYEAR 00] Goodyear, P., *eLearning, knowledge work and working knowledge*, IST2000 Event, eLearning Futures session, Nice, November 2000

[RETAL 05] S. Retalis and A. Papasalouros, *Designing and Generating Educational Adaptive Hypermedia Applications*, Educational Technology and Society (to appear).

Java Programming Language and Java Technologies :

[JLS] *The Java Language Specification, Third Edition*, Sun Microsystems, Inc., May 2005. Available online at <http://java.sun.com/docs/books/jls/>

[JSP 2.0] *JavaServer Pages Specification, Version 2.0 - Final Release*, Sun Microsystems, Inc., November 2003. Available online at <http://jcp.org/aboutJava/communityprocess/final/jsr152/>

[SERV 2.4] *Java Servlet Specification Version 2.4 - Final Release*, Sun Microsystems, Inc., November 2003. Available online at <http://jcp.org/aboutJava/communityprocess/final/jsr154/index.html>

[JSTL] *JavaServer Pages(TM) Standard Tag Library (JSTL) Specification 1.1 Maintenance Release*, Sun Microsystems, Inc., November 2003. Available online at <http://jcp.org/aboutJava/communityprocess/final/jsr052/index2.html>

XML :

[XML SPEC] *Extensible Markup Language (XML) 1.0 (Third Edition) W3C Recommendation 04 February 2004*
Available online at <http://www.w3.org/TR/REC-xml/>

[HAROLD02] Elliott Rusty Harold and W. Scott Means, *XML in a Nutshell, 2nd Edition*, O'Reilly & Associates, 2002

Software Design :

[J2EE DESIGN] *Designing Enterprise Applications with the J2EE Platform, Second Edition*, Sun Microsystems, Inc., February 2002
Available online at http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/index.html

[ACM03] Deepak Alur, John Crupi, Dan Malks, *Core J2EE Patterns: Best Practices and Design Strategies*, Second Edition, Prentice Hall, 2003

[GoF] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison Wesley, 1994

Software Engineering :

[ΣΚΟ03] Εμμανουήλ Στ. Σκορδαλάκης, *Λογισμική Μηχανική (Software Engineering)*, Αθήνα 2003

[KRUCHTEN99] Philippe Kruchten, *The Rational Unified Process: An Introduction*, Addison-Wesley, 1999.

[JBR99] Jacobson, I., Booch, G., and Rumbaugh, J. 1999. *The Unified Software Development Process*. Reading, MA.: Addison-Wesley.

[COCKBURN01] Alistair Cockburn, Writing Effective Use Cases, Addison-Wesley, 2001.

UML :

[BOOCH] Grady Booch, James Rumbaugh, and Ivar Jacobson, "The Unified Modeling Language User Guide", Addison Wesley, 1998.

[FOWLER 03] Martin Fowler, "UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition", Addison Wesley, 2003.