

Ένα Μοντέλο Κοστολόγησης Πολύμεσου Εκπαιδευτικού Λογισμικού

Νικόλαος Παπασπύρου*
(nickie@softlab.ntua.gr)

Συμεών Ρετάλης*
(retal@softlab.ntua.gr)

Αναστάσιος Κουτουμάνος*
(tkout@softlab.ntua.gr)

Εμμανουήλ Γιακουμάκης†
(mgia@aueb.gr)

Εμμανουήλ Σκορδαλάκης*
(skordala@softlab.ntua.gr)

* Εθνικό Μετσόβιο Πολυτεχνείο, Τομέας Πληροφορικής,
Τμήμα Ηλεκτρολόγων Μηχ. και Μηχ. Υπολογιστών.

† Οικονομικό Πανεπιστήμιο Αθηνών, Τμήμα Πληροφορικής.

Περίληψη

Η εξέλιξη της τεχνολογίας πληροφοριών, ιδιαίτερα η ανάπτυξη στους τομείς των δικτύων υπολογιστών και των συστημάτων υπερμέσων, ενισχύουν το ρόλο των υπολογιστών στη διδασκαλία και τη μάθηση. Πράγματι, ένα μεγάλο μέρος της διαδικασίας διδασκαλίας μπορεί να αυτοματοποιηθεί, καθώς και η διαδικασία μάθησης μπορεί να επωφεληθεί σημαντικά από τις νέες τεχνολογίες. Το αναγκαίο λογισμικό για το σκοπό αυτό ονομάζεται “εκπαιδευτικό λογισμικό” και η ανάπτυξή του είναι κάθε άλλο παρά εύκολη. Ένα από τα προβλήματα που σχετίζονται με την ανάπτυξη εκπαιδευτικού λογισμικού είναι η ακριβής κοστολόγησή του. Στην εργασία αυτή παρουσιάζουμε ένα νέο μοντέλο για την κοστολόγηση εκπαιδευτικού λογισμικού πολύμέσων, χρησιμοποιώντας έναν αλγοριθμικό μηχανισμό κοστολόγησης βασισμένο στη δομή κατάταξης εργασίας μιας τυπικής εκπαιδευτικής εφαρμογής. Προτείνουμε τη “σελίδα”, δηλ. μια πλήρη οθόνη εκπαιδευτικού υλικού, ως βασική μονάδα στο μετρικό μας σύστημα. Παρότι το μοντέλο μας δεν έχει ακόμα επαρκώς χρησιμοποιηθεί στην πράξη, τα πρώτα αποτελέσματα είναι ιδιαίτερα ικανοποιητικά.

1 Εισαγωγή

Η εισαγωγή των υπολογιστών, και γενικά της τεχνολογίας πληροφοριών, στην εκπαίδευση έχει ως σκοπό τη μερική αυτοματοποίηση της διαδικασίας διδασκαλίας και τη διευκόλυνση της διαδικασίας μάθησης σε ένα νέο περιβάλλον διδασκαλίας και μάθησης.

Σε ένα τέτοιο περιβάλλον αναμένεται ότι κάποιες πολύπλοκες έννοιες θα εξηγούνται καλύτερα, διάφορα θέματα θα συζητούνται εκτενέστερα, λιγότερος χρόνος θα απαιτείται για την εξήγηση των ίδιων πραγμάτων, νέες μέθοδοι θα βρεθούν για να δίνονται κίνητρα στους μαθητές, περισσότερες πληροφορίες σχετικές με την πορεία των μαθητών θα λαμβάνονται. Ο υπολογιστής θα ενεργεί σαν ένας υπομονετικός, ακούραστος, αμερόληπτος υπηρέτης, που θα προσφέρει κίνητρα για μάθηση. Για την επίτευξη αυτών των στόχων είναι απαραίτητη η ανάπτυξη λογισμικού προσανατολισμένου προς αυτή την κατεύθυνση εφαρμογών. Το λογισμικό αυτό ονομάζεται *εκπαιδευτικό λογισμικό* (educational software ή courseware) [Self90].

Κατά γενική παραδοχή, τα οπτικά ερεθίσματα είναι ο σημαντικότερος τρόπος με τον οποίο το κεντρικό νευρικό σύστημα του ανθρώπου έρχεται σε επαφή με τον εξωτερικό κόσμο. Επίσης, είναι γενικά παραδεκτό ότι ο ήχος και οι εικόνες μεταφέρουν συμβάλλουν στην εύκολη απομνημόνευση [Fras90]. Λαμβάνοντας υπόψη αυτές τις παρατηρήσεις, το σύγχρονο εκπαιδευτικό λογισμικό συνδυάζει διαφορετικά μέσα στο τελικό εκπαιδευτικό προϊόν. Αυτά τα μέσα είναι συνήθως: κείμενο, γραφικά, ήχος, βίντεο και κινούμενη εικόνα.

Η χρήση πολλών μέσων είναι εξαιρετικής σημασίας για την ανάπτυξη εκπαιδευτικού λογισμικού, κυρίως λόγω της ιδιότητάς της να συνδέει ήχο και εικόνα με κείμενο και, κατ’ αυτό τον τρόπο, να καθιστά τη διαδικασία της εκπαίδευσης πιο ευχάριστη για τους μαθητές [Howe92].

Προκειμένου να αναπτυχθεί το απαιτούμενο πολύμεσο εκπαιδευτικό λογισμικό είναι αναγκαίο να χρησιμοποιηθεί ένα κατάλληλο περιβάλλον, που ονομάζεται “περιβάλλον συγγραφής” (authoring environment). Τα περιβάλλοντα συγγραφής κατατάσσονται σε τρεις κατηγορίες:

- *Γλώσσες Προγραμματισμού Γενικού Σκοπού* (π.χ. C/C++, BASIC, Common LISP, Pascal, Java). Στην περίπτωση αυτή απαιτείται σημαντική προγραμματιστική εμπειρία.
- *Γλώσσες Συγγραφής* (authoring languages, π.χ. PILOT, PODIUM, COURSEWRITER, TUTOR). Οι γλώσσες αυτές είναι ειδικά σχεδιασμένες για να διευκολύνουν την ανάπτυξη εκπαιδευτικού λογισμικού, με τον ίδιο τρόπο που οι ειδικές γλώσσες προσομοίωσης (π.χ. GPSS) διευκολύνουν την ανάπτυξη προγραμμάτων

προσομοίωσης και οι γλώσσες τεχνητής νοημοσύνης (π.χ. PLANNER) διευκολύνουν την διεξαγωγή έρευνας στην αντίστοιχη περιοχή. Παρά την καλή διάθεση των σχεδιαστών τους, με λίγες εξαιρέσεις οι γλώσσες συγγραφής είναι εξίσου πολύπλοκες και απαιτούν τον ίδιο χρόνο εκμάθησης όπως και οι γλώσσες γενικού σκοπού [Kear82].

- *Συστήματα συγγραφής* (authoring systems, π.χ. TOOLBOOK, Course of Action, Authorware, ProPi, Quest). Τα συστήματα αυτά αποσκοπούν στη διευκόλυνση της ανάπτυξης εκπαιδευτικού λογισμικού από πρόσωπα χωρίς μεγάλη προγραμματιστική εμπειρία, κατά τρόπο σχετικά ανώδυνο και χωρίς σφάλματα. Συμπεριφέρονται σαν γεννήτορες προγραμμάτων [Velj90]. Οι χρήστες μεταχειρίζονται μια διαπροσωπεία υψηλού επιπέδου για να περιγράψουν το λογισμικό που πρόκειται να αναπτυχθεί. Η περιγραφή αυτή είναι στην πραγματικότητα ο καθορισμός της προδιαγραφής του λογισμικού· οι χρήστες προδιαγράφουν, δεν προγραμματίζουν.

Παρά το γεγονός ότι το κόστος του υλικού (hardware) έχει μειωθεί σημαντικά και ότι η λειτουργικότητα των περιβαλλόντων συγγραφής έχει σαφώς βελτιωθεί, η προσπάθεια που απαιτείται για την ανάπτυξη πολύμεσου εκπαιδευτικού λογισμικού παραμένει ουσιαστική. Οι κατασκευαστές εκπαιδευτικού λογισμικού δε φαίνεται να διαθέτουν την ικανότητα να εκτιμήσουν με ακρίβεια το χρόνο και το κόστος ανάπτυξης [Mars94]. Αν και στη βιβλιογραφία που σχετίζεται με την τεχνολογία λογισμικού έχει δημοσιευθεί ένας μεγάλος αριθμός από μετρικές [Keme87, Pres94] για την πρόβλεψη του κόστους ενός έργου λογισμικού, μόνο λίγες τέτοιες μετρικές έχουν προταθεί για την κοστολόγηση εκπαιδευτικού λογισμικού και η αποτελεσματικότητά τους είναι ακόμα αβέβαια.

Ο σκοπός αυτής της εργασίας είναι να παρουσιάσει ένα μοντέλο για την κοστολόγηση εκπαιδευτικού λογισμικού, βασισμένου στην δομή κατάτμησης εργασίας (work breakdown structure) [Tans88] μιας τυπικής πολύμεσης εφαρμογής. Το μοντέλο που προτείνουμε χρησιμοποιεί τη “σελίδα”, δηλαδή μια πλήρη οθόνη πληροφοριών, ως μονάδα του μετρικού συστήματος. Στην ενότητα 2 παραθέτουμε και σχολιάζουμε διάφορες παρόμοιες προσεγγίσεις σε αυτή την ερευνητική περιοχή και παρουσιάζουμε τα πρωτότυπα χαρακτηριστικά της έρευνάς μας. Στην ενότητα 3 ορίζουμε τη μετρική μας και αναλύουμε το μοντέλο κοστολόγησης. Τα αποτελέσματα μερικών μελετών περίπτωσης που χρησιμοποιήθηκαν για τον έλεγχο του μοντέλου μας παρουσιάζονται στην ενότητα 4. Τέλος, η ενότητα 5 περιέχει τη σύνοψη της εργασίας και σχόλια σχετικά με το μέλλον της έρευνάς μας.

2 Σχετική Έρευνα

Η ανάπτυξη εκπαιδευτικού λογισμικού δεν είναι εύκολο έργο. Σύμφωνα με την εκτενή βιβλιογραφία σε αυτή την περιοχή της τεχνολογίας λογισμικού, η διαδικασία ανάπτυξης γενικά απαρτίζεται από έξι στάδια:

- *Ανάλυση*: καθορίζεται το αντικείμενο και οι στόχοι του έργου, καθορίζεται η μέθοδος ανάπτυξης και οι απαιτήσεις σε υλικό και λογισμικό.
- *Σχεδίαση*: κατασκευάζεται η σκηνοθεσία (storyboard) και ένα διάγραμμα ροής για το πρόγραμμα (μια γραφική αναπαράσταση του τί πρέπει να κάνει το πρόγραμμα, συμπεριλαμβάνοντας τη σχεδίαση των διαφόρων μέσων).
- *Προγραμματισμός*: γίνεται ο προγραμματισμός του λογισμικού και ελέγχεται ότι αυτό ανταποκρίνεται στους στόχους και στο διάγραμμα ροής.
- *Τεκμηρίωση*: γράφεται ένας οδηγός για τους μαθητές και ένα εγχειρίδιο για τους εκπαιδευτές.
- *Αξιολόγηση*: το πρόγραμμα ελέγχεται τόσο κατά τη διάρκεια όσο και μετά την ανάπτυξή του προκειμένου να διαπιστωθεί ότι είναι σωστό και ανταποκρίνεται στις απαιτήσεις.
- *Συντήρηση*: διορθώνονται σφάλματα εκ των υστέρων και κατασκευάζονται νέες εκδόσεις, που περιέχουν διορθώσεις και βελτιώσεις.

Στην πράξη η ανάπτυξη εκπαιδευτικού λογισμικού είναι μια επαναληπτική διαδικασία, όπου τα διάφορα στάδια συχνά επικαλύπτονται. Αρκετά μοντέλα για τη ανάπτυξη εκπαιδευτικού υλικού έχουν προταθεί και όλα συμπεριλαμβάνουν τα παραπάνω στάδια [Tenn93, Mars95, Perr95]. Τα περισσότερα από αυτά τα μοντέλα χρησιμοποιούνται ως βάση για την κοστολόγηση εκπαιδευτικού λογισμικού.

Πέντε κύριες μέθοδοι για την κοστολόγηση εκπαιδευτικού λογισμικού εμφανίζονται στη σχετική βιβλιογραφία [Mars94]:

- *Εμπειρικές προβλέψεις*: (educated guesses) η γνώση και η εμπειρία από προηγούμενα έργα καθώς και η ανάλυση των απαιτήσεων του πελάτη επιτρέπουν στον κατασκευαστή να κάνει μια πρόχειρη κοστολόγηση.
- *Βιομηχανικοί μέσοι όροι*: (industry averages) ο κατασκευαστής βασίζεται στην κοστολόγηση διαφορετικών έργων εκπαιδευτικού λογισμικού που καλύπτουν μια ευρεία περιοχή διαφορετικών εφαρμογών.
- *Διαδραστική μέθοδος πολεμικής αεροπορίας*: (air force interactive courseware method) το μοντέλο κοστολόγησης βασίζεται στη γνώμη ειδικών, οι οποίοι λαμβάνουν υπόψη συγκεκριμένους παράγοντες που επηρεάζουν το κόστος ανάπτυξης για διάφορους τύπους εκπαιδευτικού λογισμικού και εξαρτώνται από το επίπεδο της εκπαίδευσης [Gola93].

- *Ανάλυση παράγοντα Q*: (Q factor analysis) χρησιμοποιεί έναν ποιοτικό παράγοντα “Q” για την εκτίμηση του κόστους ανάπτυξης, ο οποίος είναι παρεμφερής με το κόστος ανά μονάδα επιφάνειας που χρησιμοποιείται στην ανέγερση κτιρίων [Berg90].
- *Αλγοριθμικές μέθοδοι κοστολόγησης*: (Algorithmic methods for courseware estimation) Δυο τέτοιες αλγοριθμικές μέθοδοι υπάρχουν:
 1. Computer Based Training (CBT) analyst, η οποία βασίζεται στη χρήση πινάκων [Kear85]. και
 2. Cost Estimation Algorithm for Courseware (CEAC), η οποία χρησιμοποιεί μια εξίσωση για τον υπολογισμό του χρόνου ανάπτυξης, προσθέτοντας τις χρονικές συνεισφορές των στοιχείων διδασκαλίας, πρακτικής, προσομοίωσης και πιστοποίησης [Scho88].

Η ανάλυση αυτών των μεθόδων έδειξε ότι υπάρχει μεγάλη απόκλιση ανάμεσα στο αναμενόμενο και στο πραγματικό κόστος, όπως αναφέρεται σαφώς στο [Mars94]. Μεταξύ των αδυναμιών ορισμένων από τις μεθόδους που προαναφέρθηκαν, αξίζει να σημειωθούν οι εξής:

- δεν υπάρχει σαφής ορισμός των βασικών μονάδων,
- η εκτίμηση βασίζεται σε ελλιπή ή αναξιόπιστα στοιχεία,
- η εκτίμηση βασίζεται σε υποκειμενικές παρατηρήσεις· δεν υπάρχει εξωτερικός έλεγχος εγκυρότητας, και
- η εκτίμηση συμπεριλαμβάνει περιττούς παράγοντες κόστους.

Μια προσπάθεια για να ξεπεραστούν αυτές οι αδυναμίες εμφανίζεται στο [Mars95]. Όμως η μέθοδος αυτή βασίζεται στο μοντέλο κύκλου ζωής του καταρράκτη (waterfall life cycle model), παρότι στην ίδια εργασία αναφέρεται ότι αυτό δεν είναι το πλέον κατάλληλο για την ανάπτυξη πολύμεσων εκπαιδευτικού λογισμικού. Επιπλέον, τα στάδια της ανάλυσης και της συντήρησης αποκλείονται από την κοστολόγηση, γιατί είναι δύσκολη η εκτίμηση της συνεισφοράς τους στο συνολικό κόστος του έργου λογισμικού.

Λαμβάνοντας υπόψη τις προαναφερθείσες μεθόδους και τις αδυναμίες τους, σε αυτή την εργασία παρουσιάζουμε μια διαφορετική προσέγγιση της κοστολόγησης πολύμεσων εκπαιδευτικών εφαρμογών. Ένα πρωτότυπο χαρακτηριστικό του μοντέλου μας είναι το γεγονός ότι δεν προϋποθέτει τη χρήση ενός συγκεκριμένου μοντέλου κύκλου ζωής. Έμφαση δίνεται στην εκπαιδευτική εφαρμογή· χρησιμοποιώντας την τεχνική της δομής κατάτμησης εργασίας [Tans88], η εφαρμογή διαιρείται αναδρομικά σε τμήματα. Η διαίρεση αυτή καταλήγει στο διαχωρισμό βασικών μονάδων, που ορίζονται σαφώς και αναγνωρίζονται εύκολα. Η χρήση της σελίδας ως βάσης του μετρικού συστήματος είναι επίσης ένα πρωτότυπο χαρακτηριστικό του μοντέλου μας. Το προτεινόμενο μοντέλο είναι αλγοριθμικό και ικανοποιεί τα περισσότερα από τα κριτήρια των Boehm and Wolverton [Boeh80], που περιγράφονται συνοπτικά στην ενότητα 4.

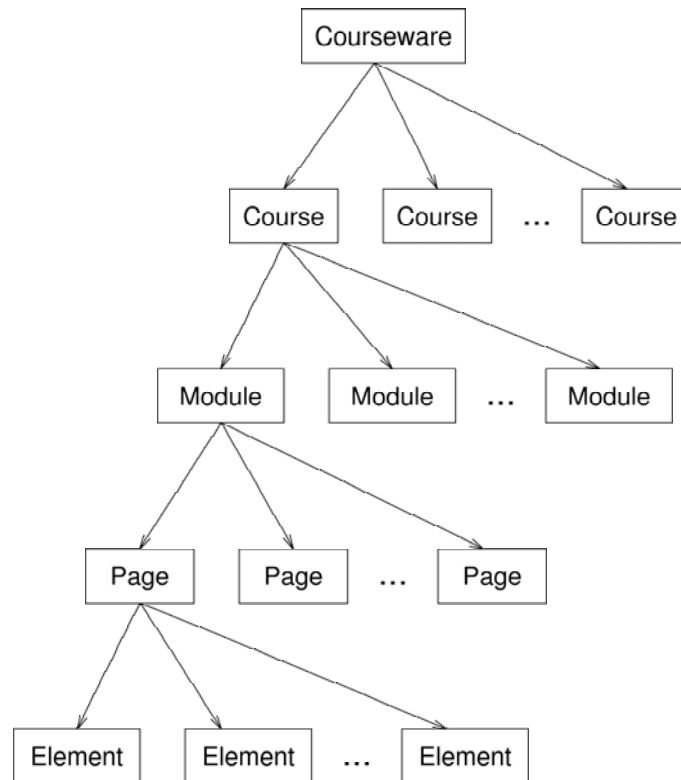
3 Εκτίμηση Κόστους

Στις ακόλουθες παραγράφους παρουσιάζουμε το μοντέλο μας για την κοστολόγηση πολύμεσων εκπαιδευτικών εφαρμογών. Στην παράγραφο 3.1 παρουσιάζεται μια σύνοψη του μοντέλου, που βασίζεται στην δομή κατάτμησης εργασίας. Στην παράγραφο 3.2 προτείνουμε ένα μέτρο μεγέθους για το εκπαιδευτικό λογισμικό, σχολιάζουμε την πρότασή μας και αναθεωρούμε τη δομή κατάτμησης εργασίας. Οι παράγραφοι 3.3, 3.4, 3.5 και 3.6 περιέχουν τη λεπτομερή περιγραφή των διαφόρων παραγόντων κόστους του μοντέλου μας. Στην παράγραφο 3.7 εξετάζουμε τη διαδικασία ρύθμισης του μοντέλου μας για κάποια συγκεκριμένη ομάδα κατασκευαστών και περιβάλλον συγγραφής. Τέλος, στην παράγραφο 3.8 επιχειρούμε μια θεωρητική αξιολόγηση του μοντέλου μας.

3.1 Σύνοψη του μοντέλου

Η μέθοδος κοστολόγησης εκπαιδευτικού λογισμικού που προτείνεται σε αυτή την εργασία βασίζεται στην τεχνική της δομής κατάτμησης εργασίας [Tans88]. Σύμφωνα με αυτή την τεχνική, μια εφαρμογή λογισμικού διαιρείται σε τμήματα και κάθε τμήμα σε άλλα υποτμήματα. Κατ’ αυτό τον τρόπο, το συνολικό κόστος είναι χοντρικά ίσο με το άθροισμα των κοστών όλων των υποτμημάτων, τα οποία πρέπει να εκτιμηθούν. Η δομή κατάτμησης εργασίας για μια τυπική εκπαιδευτική εφαρμογή τείνει να έχει μια μάλλον κοινή μορφή, που υποστηρίζεται άμεσα από την πλειοψηφία των συστημάτων συγγραφής. Η δομή αυτή παρουσιάζεται στο Σχ.1.

Η εκπαιδευτική *εφαρμογή* (application) διαιρείται σε *μάθημα* (courses). Κάθε μάθημα αντιπροσωπεύει μια ποσότητα συναφούς εκπαιδευτικού υλικού που πρέπει να διδαχθεί. Για παράδειγμα, μια εκπαιδευτική εφαρμογή που αποσκοπεί στη διδασκαλία των μαθηματικών σε φοιτητές τριτοβάθμιας εκπαίδευσης θα μπορούσε να απαρτίζεται από ένα μάθημα γραμμικής άλγεβρας, ένα μάθημα ολοκληρωτικού λογισμού, ένα μάθημα διαφορικών εξισώσεων, κ.λπ. Κάθε μάθημα χωρίζεται έπειτα σε *ενότητες* (modules), ακολουθώντας την ίδια λογική με την οποία τα τυπωμένα βιβλία χωρίζονται συνήθως σε κεφάλαια. Οι ενότητες είναι δυνατό να οργανώνονται σε περισσότερα επίπεδα, αν αυτό διευκολύνει τη δόμηση του εκπαιδευτικού λογισμικού. Όμως, χάριν απλότητας, τόσο το Σχ.1 όσο και το μοντέλο κοστολόγησης που θα παρουσιασθεί στις επόμενες παραγράφους αφορούν ένα μόνο επίπεδο ενότητων.



Σχήμα 1: Η δομή κατάτμησης εργασίας για μια τυπική εκπαιδευτική εφαρμογή.

Κάθε ενότητα είναι, με τη σειρά της, ένα σύνολο από συναφείς σελίδες (pages). Κάθε σελίδα αντιπροσωπεύει την ποσότητα πληροφορίας που βλέπει ο χρήστης στην οθόνη του υπολογιστή του. Η έννοια αυτή συναντάται συχνά ως “οθόνη” (screen) ή “κάρτα” (card) στη βιβλιογραφία και την ορολογία των διαφόρων συστημάτων συγγραφής. Μια σελίδα μπορεί να περιέχει κείμενο, γραφικά, ήχο, βίντεο, κινούμενη εικόνα και πιθανώς άλλα πολύμεσα στοιχεία (elements), ανάλογα με τις απαιτήσεις της εφαρμογής και το διαθέσιμο υλικό.

Το κόστος που συσχετίζεται με κάθε τμήμα στη δομή κατάτμησης εργασίας απαρτίζεται από δευτερεύοντα κόστος, που κάθε ένα αντιστοιχεί σε ένα από τα ακόλουθα συστατικά της ανάπτυξης εκπαιδευτικού λογισμικού:

- λειτουργικό συστατικό (operational component)
- μη λειτουργικό συστατικό (non operational component)
- τεκμηρίωση (documentation)
- παρεκκλίσεις (contingencies)

Ο τρόπος με τον οποίο αυτά τα συστατικά επηρεάζουν το συνολικό κόστος της εκπαιδευτικής εφαρμογής εξετάζεται στις επόμενες παραγράφους.

3.2 Μέτρο μεγέθους

Προκειμένου να εκτιμηθεί το κόστος ανάπτυξης μιας εφαρμογής είναι απαραίτητη η ύπαρξη ενός μέτρου μεγέθους της εφαρμογής. Η κοστολόγηση λογισμικού είναι ιδιαίτερα χρήσιμη κατά τα πρώτα στάδια της ανάπτυξης, συνήθως πριν καν ολοκληρωθούν οι φάσεις της ανάλυσης και σχεδίασης. Για το λόγο αυτό πρέπει να επισημανθεί ότι το μέτρο μεγέθους που χρησιμοποιείται στην κοστολόγηση αναφέρεται αναπόφευκτα στο αναμενόμενο μέγεθος της εφαρμογής, αφού το πραγματικό μέγεθος γενικά δεν είναι γνωστό.

Ένα μέτρο μεγέθους που θεωρείται αξιόπιστο και χρησιμοποιείται ευρέως στην ανάπτυξη λογισμικού είναι ο αριθμός γραμμών κώδικα (lines of source code, LSC) [Albr83]. Το μέτρο αυτό μπορεί να χρησιμοποιηθεί και για την ανάπτυξη εκπαιδευτικού λογισμικού όταν αυτό αναπτύσσεται χρησιμοποιώντας μια γλώσσα προγραμματισμού γενικού σκοπού, ή όταν γίνεται χρήση μεγάλων τμημάτων κώδικα, γραμμένου στην ενσωματωμένη γλώσσα κάποιου συστήματος συγγραφής. Στις περισσότερες όμως περιπτώσεις, το μέτρο αυτό δεν είναι κατάλληλο για αυτή την περιοχή εφαρμογών. Τα εμπορικά συστήματα συγγραφής διαφημίζονται κυρίως για το γεγονός ότι δεν απαιτείται προγραμματισμός προκειμένου να αναπτυχθεί μια μέση εκπαιδευτική εφαρμογή. Επιπλέον, μεγάλη προσπάθεια και χρόνος αναλύσεται στη δημιουργία γραφικών, ήχου και άλλων πολύμεσων στοιχείων που εμφανίζονται πολύ συχνά στις εκπαιδευτικές εφαρμογές. Αυτή η προσπάθεια δεν είναι δυνατό να εκφραστεί σε LSC και θεωρείται περισσότερο δουλειά ενός καλλιτέχνη, παρά ενός προγραμματιστή.

Για τους λόγους αυτούς, είναι φανερό ότι απαιτείται ένα νέο μέτρο μεγέθους για εκπαιδευτικές εφαρμογές. Το μέτρο που προτείνουμε προέρχεται άμεσα από τη δομή κατάτμησης εργασίας μιας τυπικής εκπαιδευτικής εφαρμογής, όπως παρουσιάστηκε στο Σχ.1. Δε θα ήταν καλή επιλογή να θεωρηθούν τα στοιχεία μιας σελίδας ως μονάδες μέτρου στη δομή κατάτμησης εργασίας. Αυτά τα στοιχεία είναι στενά συνδεδεμένα μεταξύ τους, προκειμένου να συνεργάζονται· η λειτουργικότητα της σελίδας βασίζεται σε αυτή τη συνεργασία. Οι απαραίτητοι σύνδεσμοι συνήθως απαιτούν σημαντική προσπάθεια για να σχεδιασθούν, προγραμματισθούν και ελεγχθούν. Αντίθετα, η σελίδα μπορεί να θεωρηθεί ως η βασική μονάδα μεγέθους μιας εκπαιδευτικής εφαρμογής, γιατί οι σελίδες μιας ενότητας μαθήματος είναι λίγο-πολύ ανεξάρτητες μεταξύ τους.

Υιοθετώντας τη σελίδα ως τη μονάδα μεγέθους για την κοστολόγηση εκπαιδευτικού λογισμικού, έμμεσα επιλέγουμε τον αριθμό των σελίδων ως το μέτρο μεγέθους για εκπαιδευτικές εφαρμογές. Στην περίπτωση αυτή, το κόστος μιας εφαρμογής θα ήταν ευθέως ανάλογο με τον αριθμό των σελίδων της. Όμως, για να ευσταθεί η προηγούμενη πρόταση, πρέπει κανείς να κάνει τις εξής υποθέσεις:

- ότι όλες οι σελίδες της εκπαιδευτικής εφαρμογής έχουν την ίδια πολυπλοκότητα, και
- ότι η ανάπτυξη μιας σελίδας είναι εντελώς ανεξάρτητη από την ανάπτυξη άλλων σελίδων.

Δυστυχώς, και οι δυο αυτές υποθέσεις είναι γενικά ψευδείς. Οι σελίδες δεν έχουν όλες την ίδια πολυπλοκότητα: υπάρχουν απλές σελίδες, π.χ. η σελίδα περιεχομένων, και πολύπλοκες σελίδες, π.χ. μια σελίδα που περιέχει έναν προσομοιωτή πτήσης σε μια υποθετικό μάθημα εναέριας πλοήγησης. Επιπλέον, τα σύγχρονα συστήματα συγγραφής συνήθως υποστηρίζουν “υπόβαθρα” (backgrounds), ούτως ώστε παρεμφερείς σελίδες να μπορούν να μοιράζονται το ίδιο υπόβαθρο, που υλοποιεί τα στοιχεία που είναι κοινά σε όλες. Η ύπαρξη των υποβάθρων καταργεί τη δεύτερη υπόθεση, γιατί το υπόβαθρο αναπτύσσεται μόνο μια φορά για την πρώτη σελίδα και μετά απλώς επαναχρησιμοποιείται σε όλες τις παρόμοιες σελίδες.

Παρ’ όλα αυτά, το γεγονός ότι οι δυο υποθέσεις δεν ευσταθούν γενικά δεν είναι αρκετός λόγος για να εγκαταλείψει κανείς τον αριθμό σελίδων ως μέτρο μεγέθους εκπαιδευτικών εφαρμογών. Προκειμένου να ξεπεραστούν τα προβλήματα, προτείνουμε:

- μια μικρή αλλαγή στη δομή κατάτμησης εργασίας μια τυπικής εφαρμογής, και
- την ύπαρξη κοστών που δεν είναι ανάλογα προς το μέτρο μεγέθους του μοντέλου.

Σε κάθε εκπαιδευτική εφαρμογή, υποστηρίζουμε ότι υπάρχει ένας αριθμός από κατηγορίες σελίδων. Όλες οι σελίδες μιας κατηγορίας έχουν την ίδια πολυπλοκότητα, που εξαρτάται από τη δυσκολία τους και το επίπεδο φινέτσας που απαιτείται. Επομένως, για κάθε κατηγορία σελίδων ξεχωριστά η πρώτη υπόθεση ευσταθεί. Αναθεωρούμε τη δομή κατάτμησης εργασίας του Σχ.1 προσθέτοντας ένα νέο επίπεδο για τις κατηγορίες σελίδων. Η νέα δομή φαίνεται στο Σχ.2. Επίσης, η δεύτερη υπόθεση είναι δυνατό να ικανοποιηθεί αν το κόστος του κοινού “υπόβαθρου” δεν συμπεριληφθεί στο κόστος κάθε σελίδας μιας κατηγορίας. Κατ’ αυτό τον τρόπο το κόστος κάθε σελίδας είναι ανεξάρτητο από το κόστος των άλλων σελίδων. Το κόστος του κοινού υπόβαθρου πρέπει φυσικά να υπολογισθεί επιπρόσθετα για κάθε κατηγορία σελίδων.

3.3 Λειτουργικό συστατικό

Το λειτουργικό συστατικό συμπεριλαμβάνει όλα τα μέρη του λογισμικού που απαρτίζουν την εκπαιδευτική εφαρμογή. Το κόστος του λειτουργικού συστατικού μπορεί να εκτιμηθεί με βάση τα αποτελέσματα της μερικής ανάλυσης και σχεδίασης. Αυτά τα αποτελέσματα οδηγούν σε μια λίγο-πολύ ακριβή εκτίμηση του αριθμού των μαθημάτων, ενότητων, κατηγοριών σελίδων και σελίδων.

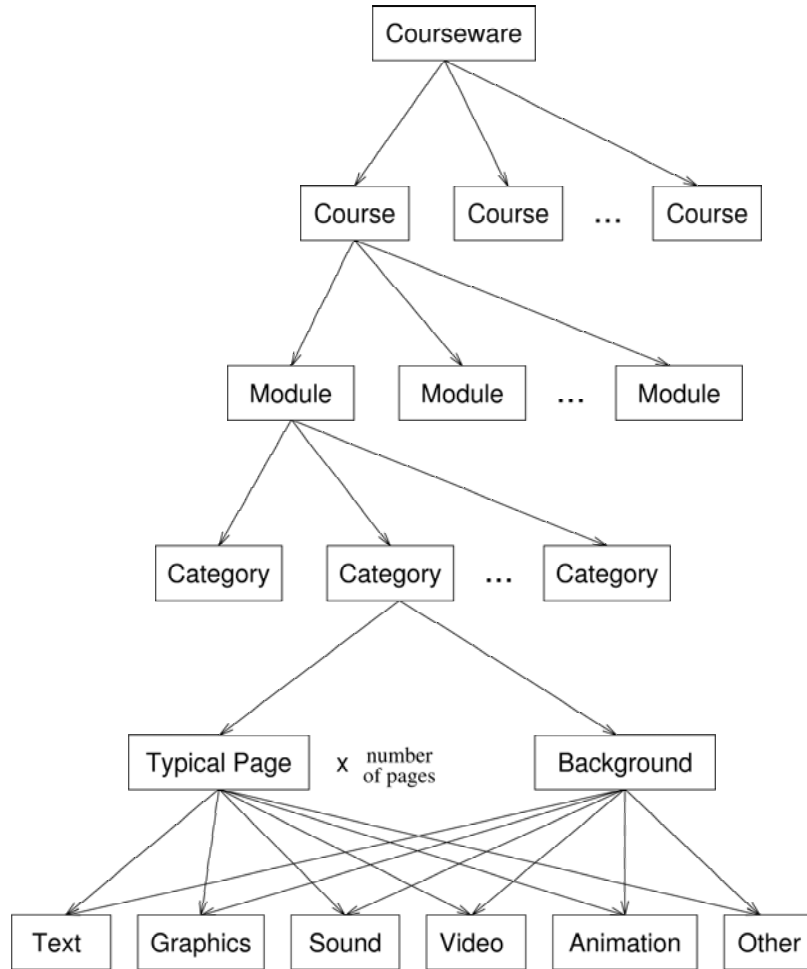
Χρησιμοποιώντας την αναθεωρημένη δομή κατάτμησης εργασίας που απεικονίζεται στο Σχ.2, το συνολικό κόστος μιας εφαρμογής μπορεί να υπολογισθεί με βάση τους παρακάτω τύπους:

$$\text{Cost}[\text{Courseware}] = \sum_{i=1}^{\text{Num}[\text{Courses}]} \text{Cost}[\text{Course}]_i \quad (1)$$

$$\text{Cost}[\text{Course}]_i = \sum_{j=1}^{\text{Num}[\text{Modules}]_i} \text{Cost}[\text{Module}]_{ij} \quad (2)$$

$$\text{Cost}[\text{Module}]_{ij} = \sum_{k=1}^{\text{Num}[\text{Categories}]_{ij}} \text{Cost}[\text{Category}]_{ijk} \quad (3)$$

όπου με $\text{Cost}[P]$ παριστάνεται το κόστος του τμήματος P της εφαρμογής και με $\text{Num}[P]$ ο αριθμός των τμημάτων τύπου P . Οι κάτω δείκτες παριστάνουν τον αύξοντα αριθμό ενός μαθήματος, μιας ενότητας ή μιας κατηγορίας σελίδων, με αυτή τη σειρά. Για παράδειγμα με $\text{Num}[\text{Categories}]_{ij}$ παριστάνεται ο αριθμός των κατηγοριών σελίδων της j -στής ενότητας του i -στού μαθήματος της εφαρμογής.



Σχήμα 2: Η αναθεωρημένη δομή κατάτμησης εργασίας για το μοντέλο κοστολόγησης.

Στο σημείο αυτό πρέπει να αποκλίνουμε ελαφρώς από τη δομή κατάτμησης εργασίας, προκειμένου να είμαστε σύμφωνοι με τη συζήτηση της προηγούμενης παραγράφου. Το κόστος μιας κατηγορίας σελίδων δεν είναι ευθέως ανάλογο με τον αριθμό των σελίδων που συμπεριλαμβάνονται σε αυτή την κατηγορία. Επιπρόσθετα, πρέπει να υπολογισθεί το κόστος του κοινού υπόβαθρου.

$$\text{Cost}[\text{Category}]_{ijk} = \text{Num}[\text{Pages}]_{ijk} \cdot \text{Cost}[\text{Page}]_{ijk} + \text{Cost}[\text{Background}]_{ij} \quad (4)$$

όπου το κόστος $\text{Cost}[\text{Page}]_{ijk}$ δεν περιλαμβάνει το κόστος των στοιχείων που ανήκουν στο κοινό υπόβαθρο.

Για μια δεδομένη κατηγορία σελίδων, το κόστος μιας τυπικής σελίδας συμπεριλαμβάνει τα κόστη όλων των πολύμεσων στοιχείων που η σελίδα περιέχει. Επιπλέον, συμπεριλαμβάνει το κόστος σχεδίασης, προγραμματισμού και ελέγχου της σελίδας.

$$\begin{aligned} \text{Cost}[\text{Page}]_{ijk} = & \text{Cost}[\text{Text}]_{ijk} + \text{Cost}[\text{Graphics}]_{ijk} + \text{Cost}[\text{Sound}]_{ijk} + \\ & + \text{Cost}[\text{Video}]_{ijk} + \text{Cost}[\text{Animation}]_{ijk} + \\ & + \text{Cost}[\text{Design}]_{ijk} + \text{Cost}[\text{Programming}]_{ijk} \end{aligned} \quad (5)$$

Ένας παρόμοιος τύπος δίνει το κόστος του κοινού υπόβαθρου.

$$\begin{aligned} \text{Cost}[\text{Background}]_{ij} = & \text{Cost}[\text{Text}]_{ij} + \text{Cost}[\text{Graphics}]_{ij} + \text{Cost}[\text{Sound}]_{ij} + \\ & + \text{Cost}[\text{Video}]_{ij} + \text{Cost}[\text{Animation}]_{ij} + \\ & + \text{Cost}[\text{Design}]_{ij} + \text{Cost}[\text{Programming}]_{ij} \end{aligned} \quad (6)$$

Το κόστος κάθε πολύμεσου στοιχείου γενικά κυμαίνεται πολύ. Επιπλέον, εξαρτάται σε μεγάλο βαθμό από την προέλευση του στοιχείου. Θα διακρίνουμε τις δυο ακραίες κατηγορίες προέλευσης ενός πολύμεσου στοιχείου:

- το στοιχείο κατασκευάζεται εξ αρχής, χρησιμοποιώντας ένα κατάλληλο εργαλείο· και

- το στοιχείο αντιγράφεται από μια υπάρχουσα βιβλιοθήκη στοιχείων προς επαναχρησιμοποίηση.

Πολλά συστήματα συγγραφής συνοδεύονται από μεγάλες βιβλιοθήκες, που ονομάζονται clip-arts και περιέχουν συνήθως πολλά συχνά χρησιμοποιούμενα γραφικά και ήχους. Η χρήση ενός ψηφιοποιητή εικόνας, ενός ψηφιοποιητή ήχου και μιας ηλεκτρονικής κάμερας απλοποιεί σημαντικά τη δημιουργία γραφικών, ήχου και βίντεο αντίστοιχα και βοηθά στη δημιουργία τέτοιων βιβλιοθηκών. Με παρόμοιο τρόπο, ηλεκτρονικό κείμενο είναι δυνατό να δημιουργηθεί χρησιμοποιώντας μια ψηφιοποιημένη εικόνα τυπωμένου κειμένου και κατάλληλο λογισμικό οπτικής αναγνώρισης χαρακτήρων (optical character recognition, OCR). Η δημιουργία ενός πολύμεσου στοιχείου εξ αρχής έχει γενικά μεγάλο κόστος. Αντίθετα, το κόστος χρησιμοποίησης ενός υπάρχοντος στοιχείου είναι συνήθως πολύ μικρότερο, αν και μερικές φορές είναι απαραίτητο να καταβληθεί κάποιο ποσό για τα δικαιώματα πνευματικής ιδιοκτησίας στον κατασκευαστή της βιβλιοθήκης.

Με βάση μόνο αυτές τις δυο ακραίες περιπτώσεις, το κόστος όλων των στοιχείων κειμένου μιας σελίδας μπορεί να υπολογισθεί χρησιμοποιώντας τον παρακάτω τύπο.

$$\text{Cost}[\text{Text}]_{ijk} = \text{Num}[\text{Text}]_{ijk} \cdot \text{Cost}_U[\text{Text}]_{ijk} \quad (7)$$

$$\text{Cost}_U[\text{Text}]_{ijk} = \mathcal{R}[\text{Text}]_{ijk} \cdot \text{Cost}_R[\text{Text}] + (1 - \mathcal{R}[\text{Text}]_{ijk}) \cdot \text{Cost}_N[\text{Text}] \quad (8)$$

όπου $\text{Num}[\text{Text}]$ είναι ένα μέτρο της ποσότητας κειμένου στη σελίδα (π.χ. ο αριθμός χαρακτήρων ή λέξεων), $\text{Cost}_U[\text{Text}]$ είναι το κόστος μιας μονάδας κειμένου, $\mathcal{R}[\text{Text}]$ είναι το κλάσμα του κειμένου που μπορεί να βρεθεί σε μια υπάρχουσα βιβλιοθήκη, $\text{Cost}_R[\text{Text}]$ είναι το κόστος χρησιμοποίησης μιας μονάδας κειμένου που βρίσκεται στη βιβλιοθήκη και $\text{Cost}_N[\text{Text}]$ το κόστος μιας μονάδας κειμένου που δημιουργείται εξ αρχής. Παρόμοιοι τύποι μπορούν να δοθούν για τα κόστη των άλλων τύπων πολύμεσων στοιχείων.

Οι δυο προαναφερθείσες περιπτώσεις προέλευσης ενός πολύμεσου στοιχείου είναι επαρκείς για το μοντέλο κοστολόγησης που προτείνουμε. Στην πράξη, νέα πολύμεσα στοιχεία δημιουργούνται συνήθως με την κατάλληλη τροποποίηση υπαρχόντων, χρησιμοποιώντας κατάλληλα εργαλεία. Το κόστος αυτών των στοιχείων γενικά εξαρτάται από το μέγεθος και τη φύση των τροποποιήσεων που απαιτούνται. Όμως, η χρήση τέτοιων στοιχείων είναι δυνατό να παρασταθεί στο μοντέλο μας δίνοντας κατάλληλη τιμή στην παράμετρο $\mathcal{R}[M]$.

Το κόστος των στοιχείων μιας σελίδας πρέπει να καλύπτει τα στάδια της ανάλυσης, σχεδίασης, προγραμματισμού και ελέγχου. Στο μοντέλο μας, αυτό γίνεται εν μέρει στα διάφορα κόστη $\text{Cost}[M]$. Παρ' όλα αυτά, πιστεύουμε ότι το κόστος της ανάλυσης και σχεδίασης μιας σελίδας δεν είναι απλά ίσο με το άθροισμα των αντίστοιχων κοστών των στοιχείων που αυτή περιέχει. Η παρατήρηση αυτή δικαιολογεί την ύπαρξη των κοστών $\text{Cost}[\text{Design}]$ στους τύπους 5 και 6, τα οποία αντιστοιχούν από κοινού στα κόστη ανάλυσης και σχεδίασης. Το ίδιο συμβαίνει και για τα κόστη προγραμματισμού και ελέγχου, που παριστάνονται στους ίδιους τύπους από τον παράγοντα $\text{Cost}[\text{Programming}]$.

Ακόμα κι όταν χρησιμοποιείται ένα σύστημα συγγραφής για την ανάπτυξη εκπαιδευτικού λογισμικού, είναι αναγκαία μερικές φορές η χρήση μιας συμβατικής γλώσσας προγραμματισμού, προκειμένου να υλοποιηθεί κάτι για το οποίο το σύστημα συγγραφής δεν επαρκεί. Αυτό συνήθως συμβαίνει στην περίπτωση εξαιρετικά πολύπλοκων σελίδων, όπως π.χ. ο προσομοιωτής πτήσης που προαναφέρθηκε. Στην περίπτωση αυτή, η τιμή του κόστους $\text{Cost}[\text{Programming}]$ πρέπει να αυξηθεί ανάλογα.

3.4 Μη λειτουργικό συστατικό

Με τον όρο μη λειτουργικό συστατικό εννοούμε οτιδήποτε δεν αποτελεί μέρος της εκπαιδευτικής εφαρμογής, πλην όμως χρειάζεται κατά την ανάπτυξή της. Για παράδειγμα, όταν μια εφαρμογή αναπτύσσεται χρησιμοποιώντας μια συμβατική γλώσσα προγραμματισμού, είναι συχνά αναγκαίο να γραφεί κώδικας για τον έλεγχο των μονάδων προγράμματος της εφαρμογής. Ο κώδικας αυτός δεν ενσωματώνεται στην εφαρμογή, είναι όμως ουσιαστικής σημασίας κατά την ανάπτυξή της. Το ίδιο συμβαίνει και με τις εκπαιδευτικές εφαρμογές. Οι προγραμματιστές συχνά αναγκάζονται να δημιουργούν στοιχεία και σελίδες προκειμένου να ελέγξουν και να διορθώσουν μεμονωμένα τμήματα της εφαρμογής.

Επιπλέον, κατά τη διάρκεια της ανάπτυξης εκπαιδευτικού λογισμικού είναι συχνά αναγκαία η δημιουργία πρωτοτύπων, προκειμένου να ελεγχθούν ή να αποσαφηνισθούν οι απαιτήσεις. Τα πρωτότυπα αυτά είναι εξαιρετικά χρήσιμα όταν οι κατασκευαστές δεν είναι βέβαιοι ποια δυνατή προσέγγιση έχει καλύτερα αποτελέσματα. Τέτοιες αβεβαιότητες παρουσιάζονται μάλλον συχνά γιατί, όσον αφορά στην εκπαίδευση, δεν υπάρχουν αντικειμενικά καλύτερες λύσεις. Τα πρωτότυπα καλύπτουν συνήθως μόνο ένα μικρό μέρος της εραφρομής και, αφού αξιολογηθούν, είτε αναθεωρούνται και ενσωματώνονται στην εφαρμογή, είτε εγκαταλείπονται. Κατά συνέπεια, το κόστος δημιουργίας των πρωτοτύπων δεν συμπεριλαμβάνεται πάντα στο λειτουργικό κόστος της εφαρμογής.

Προκειμένου να ληφθεί υπόψη το μη λειτουργικό συστατικό, είναι απαραίτητες οι ακόλουθες διορθώσεις στο μοντέλο μας:

- Ο όρος $\text{Cost}[\text{Page}]_{ijk}$ στον τύπο 4 αντικαθίσταται από

$$(1 + \alpha_{ijk}) \cdot \text{Cost}[\text{Page}]_{ijk}$$

όπου με α_{ijk} παριστάνεται το επιπρόσθετο κλάσμα για τα μη λειτουργικά στοιχεία κάθε σελίδας.

- Ο όρος $\text{Cost}[\text{Background}]_{ij}$ στον τύπο 4 αντικαθίσταται από

$$(1 + \beta_{ij}) \cdot \text{Cost}[\text{Background}]_{ij}$$

όπου με β_{ij} παριστάνεται με παρόμοιο τρόπο το επιπρόσθετο κλάσμα για τα μη λειτουργικά στοιχεία του κοινού υποβάθρου μιας κατηγορίας σελίδων και

- Ο όρος $\text{Num}[\text{Pages}]_{ijk}$ στον τύπο 4 αντικαθίσταται από

$$(1 + \gamma_{ijk}) \cdot \text{Num}[\text{Pages}]_{ijk}$$

όπου με γ_{ijk} παριστάνεται το επιπρόσθετο κλάσμα σελίδων που θα χρειασθούν κατά τη διάρκεια της ανάπτυξης αλλά δε θα συμπεριληφθούν στην τελική εφαρμογή.

Στο σημείο αυτό πρέπει να σημειωθεί ότι η χρήση κάποιου συγκεκριμένου μοντέλου κύκλου ζωής συχνά απαιτεί την ανάπτυξη ενός σημαντικού μη λειτουργικού συστατικού. Για παράδειγμα, το μοντέλο πρωτοτυποποίησης (prototyping model) απαιτεί την ανάπτυξη αρκετών πρωτοτύπων που συχνά εγκαταλείπονται αφού εξαχθούν τα απαραίτητα συμπεράσματα. Στην περίπτωση αυτή ίσως είναι σκόπιμο να αντιμετωπίζεται το μη λειτουργικό συστατικό με τον ίδιο τρόπο όπως και το λειτουργικό συστατικό, δηλαδή να γίνεται σχολαστική εκτίμηση του μεγέθους του αντί να γίνεται απλά διόρθωση με έναν πολλαπλασιαστικό παράγοντα.

3.5 Τεκμηρίωση

Η τεκμηρίωση του εκπαιδευτικού λογισμικού συνήθως αποτελείται από τρεις τύπους εγγράφων:

- Εγχειρίδια χρήσης (για εκπαιδευόμενους).
- Εγχειρίδια χρήσης (για εκπαιδευτές).
- Εσωτερικές εκθέσεις.

Η ανάγκη για δυο εκδόσεις εγχειριδίων χρήσης προκύπτει από το γεγονός ότι υπάρχουν δυο κατηγορίες ανθρώπων που εμπλέκονται στη χρήση του λογισμικού. Οι διαφορές ανάμεσα στις δυο εκδόσεις εξαρτώνται από τα χαρακτηριστικά αυτών των δυο κατηγοριών.

Οι εσωτερικές εκθέσεις γράφονται και χρησιμοποιούνται από τους κατασκευαστές της εφαρμογής κατά τη διάρκεια του κύκλου ζωής της. Συνήθως μια λεπτομερής έκθεση γράφεται με τη συμπλήρωση κάθε σταδίου της ανάπτυξης. Η ποιότητα αυτών των εκθέσεων έχει καθοριστική σημασία στο στάδιο της συντήρησης.

Στις δυο εκδόσεις των εγχειριδίων χρήσης πρέπει να συμπεριλαμβάνονται περιγραφές ή υποδείξεις για την εγκατάσταση της εφαρμογής, περιγραφή του αντικειμένου και των στόχων της εφαρμογής και λοιπές γενικές πληροφορίες. Επιπλέον, πρέπει να υπάρχουν λεπτομερείς περιγραφές σελίδων κάθε κατηγορίας, ακολουθώντας την αναθεωρημένη δομή κατάτμησης εργασίας.

Ένα μέτρο που χρησιμοποιείται συχνά για το μέγεθος της τεκμηρίωσης είναι ο αριθμός σελίδων [Pres94]. Το μοντέλο κοστολόγησης της τεκμηρίωσης εκπαιδευτικού λογισμικού δεν είναι απαραίτητο να διαφέρει από τα μοντέλα που χρησιμοποιούνται για άλλους τύπους εφαρμογών. Όμως, στην περίπτωση που η δομή κατάτμησης εργασίας κρίνεται χρήσιμη για την εκτίμηση του κόστους της τεκμηρίωσης, το μοντέλο μας προτείνει τους ακόλουθους τύπους:

$$\text{Cost}_{\text{doc}}[\text{Courseware}] = \sum_{i=1}^{\text{Num}[\text{Courses}]} \text{Cost}_{\text{doc}}[\text{Course}]_i \quad (9)$$

$$\text{Cost}_{\text{doc}}[\text{Course}]_i = \sum_{j=1}^{\text{Num}[\text{Modules}]_i} \text{Cost}_{\text{doc}}[\text{Module}]_{ij} + \text{Cost}_{\text{doc}}[\text{General}]_i \quad (10)$$

$$\text{Cost}_{\text{doc}}[\text{Module}]_{ij} = \sum_{k=1}^{\text{Num}[\text{Categories}]_{ij}} \text{Cost}_{\text{doc}}[\text{Category}]_{ijk} \quad (11)$$

3.6 Παρεκλίσεις

Οι ανακρίβειες στην εκτίμηση του κόστους λογισμικού οφείλονται κυρίως σε δυο λόγους: αφενός στην έλλειψη λεπτομερούς γνώσης τη χρονική στιγμή που γίνεται η εκτίμηση, αφετέρου σε μια γενική τάση αισιοδοξίας κάθε ομάδας κατασκευαστών. Ένας τρόπος για να ξεπεραστεί το πρόβλημα των ανακρίβειών είναι να χρησιμοποιηθούν

Πίνακας 1: Παρεκλίσεις.

Παράγοντας κόστους	Συντελεστής διόρθωσης (%)
Αριθμός σελίδων	10–20
Κόστος πολύμεσων στοιχείων	25–50
Κόστος ανάλυσης και σχεδίασης	75–150
Κόστος προγραμματισμού και ελέγχου	25–50

διορθωτικοί παράγοντες για παρεκλίσεις. Όταν το LSC χρησιμοποιείται ως το μέτρο μεγέθους του λογισμικού, η παρέκλιση εκφράζεται συνήθως ως ένα ποσοστό του εκτιμώμενου μεγέθους σε LSC.

Στο μοντέλο μας, ένας παράγοντας για παρεκλίσεις πρέπει να συμπεριληφθεί σε όλα τα μεγέθη πλήθους και κόστους που χρησιμοποιούνται στους διάφορους τύπους. Συγκεκριμένα, τα παρακάτω μεγέθη συχνά υποτιμώνται:

- Ο εκτιμώμενος αριθμός σελίδων $\text{Num}[\text{Pages}]$.
- Τα εκτιμώμενα κόστη για τους διάφορους τύπους πολύμεσων στοιχείων $\text{Cost}_U[M]$.
- Το εκτιμώμενο κόστος ανάλυσης και σχεδίασης $\text{Cost}[\text{Design}]$.
- Το εκτιμώμενο κόστος προγραμματισμού και ελέγχου $\text{Cost}[\text{Programming}]$.

Αυτή η υποτίμηση μπορεί να διορθωθεί με τον πολλαπλασιασμό των εκτιμώμενων μεγεθών με κατάλληλους σταθερούς συντελεστές. Παρόμοια μοντέλα για την εκτίμηση παρεκλίσεων συναντώνται συχνά στη βιβλιογραφία, π.χ. στο [Hump89].

Είναι δύσκολο να βρεθούν ακριβείς τιμές για τους συντελεστές παρεκλίσεων. Στον πίνακα 1 προτείνονται περιοχές τιμών που προέκυψαν από τις περιορισμένες μελέτες περίπτωσης που εξετάσαμε. Οι τιμές αυτές είναι απλώς ενδεικτικές.

3.7 Ρύθμιση

Η επιλογή τιμών για τις διάφορες παραμέτρους του μοντέλου κοστολόγησης είναι μια δύσκολη δουλειά. Μερικές από αυτές τις τιμές εξαρτώνται άμεσα από το μέγεθος και τη φύση του έργου λογισμικού και του επιπέδου γνώσης γύρω από αυτό, όπως π.χ. οι παράγοντες επαναχρησιμοποίησης $\mathcal{R}[M]$ για διάφορα πολύμεσα στοιχεία. Άλλες είναι μάλλον ανεξάρτητες από το συγκεκριμένο έργο και πρέπει να βασίζονται σε ιστορικά στοιχεία, που συγκεντρώνονται από την ανάπτυξη προηγούμενων έργων. Για παράδειγμα, οι τιμές που χρησιμοποιούνται για τους συντελεστές παρεκλίσεων γενικά εξαρτώνται από την εμπειρία των κατασκευαστών και το επίπεδο ωριμότητας.

Όσον αφορά στις τιμές που είναι ανεξάρτητες του έργου λογισμικού που πρόκειται να αναπτυχθεί, πιστεύουμε ότι ακριβείς εκτιμήσεις μπορούν να γίνουν μόνο μέσω μιας επίπονης και χρονοβόρας διαδικασίας δοκιμών και σφαλμάτων, που ονομάζεται *ρύθμιση* του μοντέλου. Στη διαδικασία αυτή, μια ομάδα κατασκευαστών ξεκινά γενικά με κάποιες αρχικές τιμές για τις παραμέτρους του συστήματος και στη συνέχεια τις αναθεωρεί διαρκώς, συγκρίνοντας τα εκτιμώμενα κόστη με τα πραγματικά.

3.8 Αξιολόγηση

Η χρησιμότητα ενός μοντέλου κοστολόγησης λογισμικού είναι δυνατό να αξιολογηθεί με βάση κάποια συγκεκριμένα κριτήρια [Boeh80, Mars94]:

1. *Ορισμός* (Definition): Είναι σαφής ο ορισμός των κοστών που εκτιμά το μοντέλο, καθώς και αυτών που δεν υπολογίζει.
2. *Πιστότητα* (Fidelity): Είναι το εκτιμώμενο κόστος κοντά στο πραγματικό.
3. *Αντικειμενικότητα* (Objectivity): Αποφεύγεται η συσχέτιση μεγάλου μέρους του κόστους με υποκειμενικούς παράγοντες που δεν είναι εύκολο να ρυθμισθούν.
4. *Αναπαραγωγή* (Constructiveness): Είναι δυνατό να εξηγηθεί στο χρήστη γιατί το μοντέλο δίνει την εκτίμηση που δίνει.
5. *Λεπτομέρεια* (Detail): Περιέχει το μοντέλο μια ακριβή κατάτμηση φάσεων και εργασιών.
6. *Σταθερότητα* (Stability): Μικρές διαφορές στην είσοδο προκαλούν μικρές διαφορές στο εκτιμώμενο κόστος.
7. *Έκταση* (Scope): Καλύπτει το μοντέλο την κατηγορία έργων λογισμικού των οποίων η κοστολόγηση απαιτείται.
8. *Ευκολία χρήσης* (Ease of use): Είναι εύκολο να καταλάβει και να χρησιμοποιήσει κανείς το μοντέλο.
9. *Προβλεψιμότητα* (Prospectiveness): Αποφεύγεται η χρήση πληροφορίας που δεν είναι γνωστή παρά μόνο μετά τη συμπλήρωση του έργου.
10. *Φειδώ* (Parsimony): Αποφεύγεται η χρήση περιττών παραγόντων κόστους ή παραγόντων που δε συνεισφέρουν σημαντικά στο τελικό αποτέλεσμα.

Το μοντέλο μας δίνει σαφή ορισμό των όρων και μέτρων που χρησιμοποιούνται με όσο το δυνατό μεγαλύτερη λεπτομέρεια. Έμφαση δίνεται στην περιγραφή ενός σαφούς, εύκολα κατανοητού αλγορίθμου που δεν περιέχει περιττά στοιχεία. Η ευκολία χρήσης απορρέει από την απλότητα του μοντέλου και τις συγκεκριμένες παραμέτρους που χρησιμοποιεί.

Όσον αφορά στο κριτήριο “λεπτομέρειας”, το μοντέλο μας εστιάζεται κυρίως στη δομή κατάταξης εργασίας και δε βασίζεται το εκτιμώμενο κόστος στις διάφορες φάσεις ανάπτυξης. Όμως, κόστη για την ανάλυση, σχεδίαση, προγραμματισμό και έλεγχο έχουν συμπεριληφθεί σε πολλά σημεία. Χρησιμοποιώντας τη δομή κατάταξης εργασίας, η εκτίμηση του συνολικού κόστους βασίζεται σε συγκεκριμένα (μή περιττά) τμήματα του λογισμικού και μπορεί εύκολα να ελεγχθεί από το χρήστη. Επομένως, τα κριτήρια της “αναπαραγωγής” και “φειδούς” ικανοποιούνται επίσης.

Επιπλέον, το κριτήριο της “σταθερότητας” ικανοποιείται για δυο λόγους. Κατ’ αρχήν χρησιμοποιείται ως βασική μονάδα μεγέθους η σελίδα, που αποτελείται από διάφορα στοιχεία. Μικρές αλλαγές στα στοιχεία που περιέχει μια σελίδα προκαλούν προβλέψιμες και μικρές αλλαγές στο εκτιμώμενο κόστος. Επίσης, δε χρησιμοποιούνται ενδείξεις εμπειρίας ή γνώσης του εκπαιδευτικού υλικού. Τέτοιες παράμετροι θα προκαλούσαν πολλά προβλήματα στον υπολογισμό και την ακριβή συνεισφορά τους. Επιπλέον, μικρές αλλαγές θα μπορούσαν να επιφέρουν μεγάλες αλλαγές στο συνολικό κόστος. Αντίθετα, στο μοντέλο μας, το κόστος κάθε σελίδας μεταβάλλεται με προβλέψιμο τρόπο, ανάλογα με την εμπειρία κάθε ομάδας κατασκευαστών.

Το μοντέλο μας αποφεύγει τη χρήση πληροφορίας που δεν είναι γνωστή παρά μόνο στα τελευταία στάδια της ανάπτυξης του έργου. Οι τιμές όλων των παραμέτρων που χρησιμοποιούνται στον αλγόριθμο μπορούν να καθορισθούν μετά την προκαταρκτική ανάλυση και σχεδίαση του εκπαιδευτικού λογισμικού. Άρα, το μοντέλο μας ικανοποιεί το κριτήριο της “έλλειψης πρωθύστερων”.

Η έκταση του μοντέλου μας καλύπτει όλα τα είδη εκπαιδευτικού λογισμικού. Η κοστολόγηση δε βασίζεται σε κάποιο συγκεκριμένο μοντέλο κύκλου ζωής, κυρίως γιατί επικρατεί ακόμα μεγάλη σύγχυση στη βιβλιογραφία σχετικά με το ποιο μοντέλο είναι το πλέον κατάλληλο για την ανάπτυξη εκπαιδευτικού λογισμικού. Όμως, το μοντέλο μας είναι σε θέση να προσαρμόζεται εύκολα σε διαφορετικά είδη πολύμεσων εκπαιδευτικών έργων λογισμικού, χρησιμοποιώντας οποιοδήποτε από τα υπάρχοντα μοντέλα κύκλου ζωής.

Περισσότερη έρευνα κρίνεται απαραίτητη προκειμένου να εξακριβωθεί κατά πόσο το μοντέλο μας ικανοποιεί τα κριτήρια της “πιστότητας” και της “αντικειμενικότητας”, όπως συζητείται εκτενέστερα στην ενότητα 4.

4 Μελέτες Περίπτωσης

Έχουμε εφαρμόσει το μοντέλο μας σε έναν περιορισμένο αριθμό μελετών περίπτωσης, που μπορούν να καταταγούν σε τέσσερις κύριες κατηγορίες, με διαφορετικά χαρακτηριστικά.

1. Εκπαιδευτικό λογισμικό για παιδιά.
2. Εκπαιδευτικό λογισμικό για τη δευτεροβάθμια εκπαίδευση.
3. Εκπαιδευτικό λογισμικό για την τριτοβάθμια εκπαίδευση.
4. Εκπαιδευτικό λογισμικό για επαγγελματική εκπαίδευση.

Στους πίνακες που ακολουθούν παρουσιάζουμε τα αποτελέσματα της εφαρμογής του μοντέλου σε τέσσερα αντιπροσωπευτικά έργα εκπαιδευτικού λογισμικού, ένα από κάθε κύρια κατηγορία. Συγκεκριμένα, τα έργα είναι:

1. *Πήγασος*: πολύμεσο εκπαιδευτικό λογισμικό που βασίζεται σε ένα μικρό (50 σελίδων) διήγημα για παιδιά, με τίτλο “Με τα φτερά του Πήγασου”, γραμμένο από τον Χρήστο Μπουλωτή.
2. *Ωρίων*: μια εισαγωγή στον προγραμματισμό για μαθητές γυμνασίου, βασισμένη στη γλώσσα προγραμματισμού BASIC, που περιέχει εκτενή περιγραφή της θεωρίας, ασκήσεις και ένα μικρό εξειδικευμένο διερμηνέα BASIC [Tsan94].
3. *Pascal*: μια εισαγωγή στην γλώσσα προγραμματισμού Pascal για πρωτοετείς φοιτητές τριτοβάθμιας εκπαίδευσης, που περιέχει θεωρία και ασκήσεις.
4. *Corel*: πολύμεσο εκπαιδευτικό λογισμικό για την επαγγελματική εκπαίδευση στη χρήση του πολύμεσου πακέτου παρουσιάσεων “Corel Show”, που περιέχει εκτενή θεωρία, παραδείγματα και ασκήσεις.

Ο πίνακας 2 περιέχει τις εκτιμώμενες τιμές για το μέγεθος κάθε έργου και το συνολικό εκτιμώμενο κόστος, χωρίς να ληφθούν υπόψη διορθώσεις οφειλόμενες σε παρεκλίσεις. Σε παρόμοια μορφή, ο πίνακας 3 περιέχει τις πραγματικές τιμές που μετρήθηκαν μετά την ανάπτυξη των έργων λογισμικού. Οι πίνακες 4 και 5 περιέχουν μερικές από τις τιμές των παραμέτρων του μοντέλου που χρησιμοποιήθηκαν για την εκτίμηση.

Τα αποτελέσματα δείχνουν ότι το πραγματικό κόστος δεν αποκλίνει σημαντικά από το εκτιμώμενο. Πρέπει να σημειωθεί και πάλι ότι δε χρησιμοποιήθηκαν διορθώσεις παρεκλίσεων, κυρίως γιατί δεν υπήρχαν επαρκή ιστορικά στοιχεία προκειμένου να ρυθμισθούν οι διορθωτικές παράμετροι. Για το λόγο αυτό, πιστεύουμε ότι η απόκλιση που παρατηρείται είναι δικαιολογημένη. Οι περιοχές τιμών για τους διορθωτικούς συντελεστές παρεκλίσεων που παρουσιάζονται στον πίνακα 1 σαφώς επαρκούν για την κάλυψη των αποκλίσεων.

Πίνακας 2: Αναμενόμενο κόστος.

Λογισμικό Τίτλος έργου	Αριθμός μαθημάτων	Αριθμός κατηγοριών	Αριθμός σελίδων	Αναμενόμενο κόστος (ανθρωπο-ώρες)
Πήγασος	1	5	75	192
Ωρίων	2	9	130	919
Pascal	1	7	150	286
Corel	1	5	100	316

Πίνακας 3: Πραγματικό κόστος.

Λογισμικό Τίτλος έργου	Αριθμός μαθημάτων	Αριθμός κατηγοριών	Αριθμός σελίδων	Πραγματικό κόστος (ανθρωπο-ώρες)	Απόκλιση (%)
Πήγασος	1	5	73	230	19.8
Ωρίων	2	8	124	850	-7.5
Pascal	1	7	147	320	11.9
Corel	1	6	113	380	20.3

Πίνακας 4: Παράμετροι μοντέλου (επαναχρησιμοποίηση).

Λογισμικό Τίτλος έργου	Επαναχρησιμοποίηση πολύμεσου στοιχείου			
	$\mathcal{R}[\text{Text}]$	$\mathcal{R}[\text{Graphics}]$	$\mathcal{R}[\text{Sound}]$	$\mathcal{R}[\text{Animation}]$
Πήγασος	0.80	0.75	0.90	–
Ωρίων	0.00	0.50	–	0.00
Pascal	0.20	0.50	–	0.00
Corel	0.25	0.50	0.10	0.00

Πίνακας 5: Παράμετροι μοντέλου (μη λειτουργικό συστατικό).

Λογισμικό Τίτλος έργου	Συντελεστές μη λειτουργικού συστατικού		
	α	β	γ
Πήγασος	0.02	0.20	0.10
Ωρίων	0.02	0.20	0.20
Pascal	0.00	0.50	0.05
Corel	0.01	0.50	0.10

Οι περιορισμένες μελέτες περίπτωσης δεν μας επιτρέπουν να ισχυρισθούμε ότι το μοντέλο μας ικανοποιεί τα κριτήρια της “πιστότητας” και της “αντικειμενικότητας”, όπως αυτά ορίζονται στην παράγραφο 3.8. Όσον αφορά στο κριτήριο της “αντικειμενικότητας”, ένας επαρκώς μεγάλος αριθμός στατιστικών στοιχείων απαιτείται προτού μπορέσει κανείς να αποφασίσει αν ικανοποιείται. Τέλος, όσον αφορά στο κριτήριο της “πιστότητας”, πιστεύουμε ότι με σωστή ρύθμιση το μοντέλο μας μπορεί να παράγει εκτιμήσεις που να πλησιάζουν πολύ κοντά στην πραγματικότητα.

5 Σύνοψη

Η ακριβής κοστολόγηση είναι ένα από τα δυσκολότερα προβλήματα της τεχνολογίας λογισμικού. Όσον αφορά στο εκπαιδευτικό λογισμικό, οι προσεγγίσεις που προτείνονται στη βιβλιογραφία έχουν γνωστές αδυναμίες, που συζητήθηκαν στην ενότητα 2. Προκειμένου να ξεπερασθούν αυτές οι αδυναμίες, προτείνουμε ένα μοντέλο που βασίζεται στη δομή κατάτμησης εργασίας μιας τυπικής εκπαιδευτικής εφαρμογής. Η πρωτοτυπία της προσέγγισής μας έγκειται αφενός στο γεγονός ότι το αλγοριθμικό μοντέλο κοστολόγησης που προτείνουμε είναι ανεξάρτητο του μοντέλου κύκλου ζωής που χρησιμοποιείται, αφετέρου στην επιλογή της μονάδας μεγέθους. Επιπλέον, λόγω της απλότητας και του σαφούς ορισμού του, το μοντέλο μας ικανοποιεί τα περισσότερα από τα κριτήρια αξιολόγησης του [Boeh80].

Δεν έχουμε ακόμα ελέγξει το μοντέλο μας με έναν επαρκή αριθμό μελετών περίπτωσης, ώστε να είμαστε πεπεισμένοι για την ακρίβειά του. Παρ'όλα αυτά, τα αποτελέσματα της περιορισμένης εφαρμογής του σε ειδικά επιλεγμένες αντιπροσωπευτικές περιπτώσεις είναι ιδιαίτερα ικανοποιητικά.

Περαιτέρω έρευνα θα εστιαστεί κυρίως στη συλλογή ιστορικών στοιχείων και τη σωστή ρύθμιση του μοντέλου μας. Αυτό θα επιχειρηθεί για τη δική μας ομάδα ανάπτυξης, καθώς και για άλλες ομάδες σε οίκους λογισμικού διαφόρων επιπέδων ωριμότητας. Σκοπός της περαιτέρω έρευνας είναι επίσης να προσδιορισθεί η συνεισφορά άλλων παραγόντων στο κόστος ανάπτυξης πολύμεσων εκπαιδευτικών εφαρμογών. Ήδη γίνεται έρευνα που αποσκοπεί στην ενσωμάτωση στο μοντέλο μας των διαφορετικών ρόλων που αναλαμβάνουν τα μέλη της ομάδας ανάπτυξης (π.χ. ειδήμονες εκπαιδευτικού υλικού, διαχειριστές, καλλιτέχνες, προγραμματιστές, κ.λπ.) κατά την ανάπτυξη λογισμικού. Με βάση τα αποτελέσματα που έχουμε ως τώρα, πιστεύουμε ότι η έρευνα βρίσκεται στο σωστό δρόμο και ότι η εργασία μας έχει τις δυνατότητες να εξελιχθεί σε ένα ακριβές μοντέλο κοστολόγησης πολύμεσων εκπαιδευτικών εφαρμογών.

Βιβλιογραφία

- [Albr83] A.J. Albrecht and J.E. Gaffney, Jr., "Software function, source lines of code, and development effort prediction: A software science validation", *IEEE Transactions in Software Engineering*, vol. SE-9, no. 6, November 1983.
- [Berg90] R.B. Bergman and T.V. Moore, *Managing interactive video/multimedia projects*, Educational Technology Publications, Englewood Cliffs, NJ, 1990.
- [Boeh80] B.W. Boehm and R.W. Wolverton, "Software cost modelling: Some lessons learned", *Journal of Systems and Software*, vol. 1, no. 3, pp. 195–201, 1980.
- [Fras90] R. Fraser et al., "Learning activities and classroom roles with and without the microcomputer", in O. Boyd-Barrett and E. Scanlon, editors, *Computers and Learning*, pp. 205–228, Addison Wesley, 1990.
- [Gola93] K.C. Golas, "Estimating time to develop interactive courseware in the 1990's", in *Proceedings of the Interservice/Industry Training and Education Conference*, 1993.
- [Howe92] G.T. Howell, *Building hypermedia applications*, McGraw Hill, 1992.
- [Hump89] W.S. Humphrey, *Managing the software process*, Addison Wesley, 1989.
- [Kear82] G. Kearsley, "Authoring systems in computer based education", *Communications of the ACM*, vol. 25, no. 7, pp. 429–437, 1982.
- [Kear85] G. Kearsley, "The CBT advisor: An expert system program for making decisions about CBT", *Performance and Instruction*, vol. 24, no. 9, pp. 15–17, 1985.
- [Keme87] C.F. Kemerec, "An empirical validation of software cost estimation models", *Communications of the ACM*, vol. 70, no. 5, May 1987.
- [Mars94] I.M. Marshall, W.B. Samson, P.I. Dugard and G.R. Lund, "Courseware: How much will it cost to develop?", in *2nd All Ireland Conference on the Teaching of Computing*, Dublin, Ireland, 1994.
- [Mars95] I.M. Marshall, W.B. Samson and P.I. Dugard, "The mythical courseware development to delivery time ratio", *Computers in Education*, vol. 25, no. 3, pp. 113–122, 1995.
- [Perr95] X. Perrot, *Production des hypermédias et des interatifs multimédias pour les musées*, Ph.D. thesis, Université Paris 8, 1995.
- [Pres94] R.S. Pressman, *Software engineering: A practitioner's approach*, McGraw Hill International, european edition, 1994.
- [Scho88] R.B. Schooley, "Computer based training (CBT) cost estimating algorithm for courseware (CEAC)", in *Proceedings of the Interservice/Industry Training Systems Conference*, pp. 319–328, 1988.
- [Self90] J. Self, "A critical evaluation of educational software: Climate", in O. Boyd-Barrett and E. Scanlon, editors, *Computers and Learning*, pp. 147–154, Addison Wesley, 1990.
- [Tans88] R.C. Tansworthe, "The work breakdown structure in software project management", *Journal of Systems and Software*, vol. 1, pp. 181–186, 1988.

- [Tenn93] R.D. Tennyson and R.L. Elmore, “Integrated courseware engineering system”, in *Automating Instructional Design, Development and Delivery*, NATO ASI, Grimstad, Norway, 1993.
- [Tsan94] P. Tsanakas and G. Papakonstantinou, “Design and implementation of educational software for the teaching of informatics”, in *Proceedings of the 2nd Panhellenic Conference in Computer Science*, Athens, 1994.
- [Velj90] M.D. Veljkov, “Managing multimedia: Authoring systems are the glue that holds multimedia applications together”, *BYTE*, pp. 227–232, August 1990.