

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ



**Τμήμα Διοικητικής Επιστήμης &
Τεχνολογίας**

**Προπτυχιακή εργασία που υποβλήθηκε στα πλαίσια του
Μαθήματος
του 8^{ου} εξαμήνου:**

***ΣΥΝΤΗΡΗΣΗ ΚΑΙ ΕΠΑΝΥΛΟΠΟΙΗΣΗ
ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ***

**Υπεύθυνος Καθηγητής:
Δ. ΣΠΙΝΕΛΛΗΣ**

ΕΡΓΟ:

***ETH LECTURE
COMMUNICATOR***

ΕΠΙΜΕΛΕΙΑ:

■ ΣΚΟΡΔΑΛΗ ΒΑΣΙΛΙΚΗ (8000119)

ΙΟΥΝΙΟΣ 2004

ΤΕΛΙΚΟ ΠΑΡΑΔΟΤΑΙΟ

1. ΕΙΣΑΓΩΓΗ	2
2. ΥΠΑΡΧΟΥΣΑ ΚΑΤΑΣΤΑΣΗ	2
Ⓜ Περιγραφή Συστήματος	2
Ⓜ Βασικά Χαρακτηριστικά	3
Ⓜ Απαιτήσεις Συστήματος	3
Ⓜ Bugs	3
3. ΕΦΑΡΜΟΓΗ	3
Ⓜ Προτάσεις για αλλαγή	3
Ⓜ Σχεδιασμός	4
Ⓜ Υλοποίηση	5
Ⓜ Εργαλεία	5
4. ΑΠΟΤΕΛΕΣΜΑΤΑ	5
Ⓜ Εμφάνιση	5
Ⓜ Συνεργασία	10
5. ΣΥΜΠΕΡΑΣΜΑΤΑ	10
Ⓜ Κριτική Έργου	10
Ⓜ Δυσκολίες	10
Ⓜ Πλεονεκτήματα	11
ΠΑΡΑΡΤΗΜΑΤΑ	12
• ΠΑΡΑΡΤΗΜΑ Α	12
• ΠΑΡΑΡΤΗΜΑ Β	36

ΤΕΛΙΚΟ ΠΑΡΑΔΟΤΑΙΟ

1. ΕΙΣΑΓΩΓΗ

Το πρόβλημα των ελληνικών πανεπιστημίων εντοπίζεται, κυρίως, στην έλλειψη πρακτικής εφαρμογής των θεωρητικών γνώσεων. Είναι επιστημονικά αποδεδειγμένο πως ο άνθρωπος μαθαίνει καλύτερα όταν έρχεται σε προσωπική επαφή με τα προβλήματα και ενθαρρύνεται ο ίδιος να βρει λύσεις. Οι καθηγητές έχουν ήδη αρχίσει να προσανατολίζονται σε πιο πρακτικές μεθόδους μάθησης, συμπεριλαμβάνοντας ασκήσεις και εργασίες στα πλαίσια των μαθημάτων τους. Ζωντανή απόδειξη είναι η συγκεκριμένη εργασία, που έγινε για το μάθημα Συντήρηση Και Επανυλοποίηση Πληροφοριακών Συστημάτων.

Στα πλαίσια της εργασίας έπρεπε να επιλεγεί ένα έργο ανοιχτού λογισμικού και να γίνει κάποια αλλαγή στον υπάρχοντα κώδικα, προσδίδοντας μια καινούρια λειτουργία στο έργο ή διορθώνοντας τυχόν προβλήματα.

2. ΥΠΑΡΧΟΥΣΑ ΚΑΤΑΣΤΑΣΗ

Το έργο που επιλέχτηκε είναι το *ETH Lecture Communicator* και η επιλογή του έγινε από τον δικτυακό τόπο : <http://sourceforge.net>.

Περιγραφή Συστήματος

Το *ETH Lecture Communicator* είναι ένα εργαλείο σχεδιασμένο για πανεπιστημιακές τάξεις με μεγάλο αριθμό σπουδαστών που επικοινωνούν μέσω υπολογιστών κατά τη διάρκεια της διάλεξης. Το εργαλείο επιτρέπει άμεση και συνεχή επικοινωνία μεταξύ καθηγητή και σπουδαστών με έναν οργανωμένο τρόπο.

Ο καθηγητής μπορεί να στείλει ερωτήσεις διαφόρων τύπων (π.χ. πολλαπλής επιλογής, σε κλίμακα) σε όλους τους συμμετέχοντες σπουδαστές και να αξιολογήσει τις απαντήσεις τους. Οι σπουδαστές μπορούν να κάνουν σχετικές με το περιεχόμενο ερωτήσεις, οι οποίες εκτιμώνται από όλους τους άλλους σπουδαστές. Ο καθηγητής μπορεί έπειτα να δώσει προτεραιότητα στις ερωτήσεις που ενδιαφέρουν τον μεγαλύτερο αριθμό σπουδαστών.

Το εργαλείο προορίζεται για να βελτιώσει τη γενική ποιότητα μιας διάλεξης με παροχή συνεχούς ανατροφοδότησης στον καθηγητή σχετικά με την κατανόηση του υλικού από τους σπουδαστές.

Διαχειριστής του έργου είναι ο Oliver Trachsel και η ηλεκτρονική του διεύθυνση είναι : oliver.trachsel@inf.ethz.ch.

Ⓢ Βασικά Χαρακτηριστικά

- Η γλώσσα προγραμματισμού, στην οποία είναι γραμμένο το έργο είναι η Java.
- Το έργο μπορεί να δουλέψει σε οποιαδήποτε πλατφόρμα (MacOS X, Windows, OS Independent, Linux), αλλά στην συγκεκριμένη περίπτωση οι εργασίες γίνανε σε Windows.
- Άδεια : GNU General Public License (GPL)
- Διεύθυνση Ιστοσελίδας :
<http://sourceforge.net/projects/lectcomm>

Ⓢ Απαιτήσεις Συστήματος

- Java Runtime Environment, Version 1.4 ή νεότερη
- Εξωτερικές Βιβλιοθήκες (log4j, version 1.2.8 ή νεότερη, JFreeChart 0.9.16 ή νεότερη, and iText 1.02b ή νεότερη)
- Ant

Ⓢ Bugs

Τα σημαντικότερα προβλήματα του έργου είναι τα εξής :

- Οι μεγάλοι μήκους ερωτήσεις συχνά είναι η αιτία να μην εμφανίζεται το πάνελ απάντησης στην οθόνη του παραλήπτη.
- Οι ερωτήσεις του καθηγητή κατά την προεπισκόπηση τους στο πάνελ δεν προσαρμόζονται σε πλάτος.
- Η εναλλαγή μεταξύ ενός / πολλαπλών πελατών(clients) ανά IP έχει πρόβλημα.

3. ΕΦΑΡΜΟΓΗ

Ⓢ Προτάσεις για αλλαγή

- Προσθήκη ενός καινούριου τύπου ερωτήσεων (π.χ. σωστό – λάθος).
- Διόρθωση του τρίτου προβλήματος που αναφέρθηκε (Η εναλλαγή μεταξύ ενός / πολλαπλών πελατών(clients) ανά IP) και οφείλεται στο ότι ένας μετρητής είναι λάθος ορισμένος.

Τελικά, λόγω έλλειψης χρόνου, αποφασίστηκε να υλοποιηθεί μόνο η προσθήκη τύπου ερώτησης Σωστό – Λάθος. Οι τύποι ερωτήσεων που ήδη υπήρχαν είναι :

- Πολλαπλής Επιλογής (Multiple Choice)
- Με Κλίμακα (Lickert Scale)
- Με Βαθμολογική Κατάταξη (Item Rating)
- Σύζευξης (Matching)
- Σύντομης Απάντησης (Short Answer)

🌐 Σχεδιασμός

Ο χρονοσχεδιασμός του έργου έγινε σύμφωνα με το χρονοδιάγραμμα που δόθηκε στην αρχή του εξαμήνου και αφορούσε τα παραδοταία του μαθήματος. Συγκεκριμένα :

- Επιλογή Έργου (μέχρι 19 Απριλίου, 2004)
- Επιλογή και Σχεδιασμός Αλλαγών (μέχρι 10 Μαΐου, 2004)
- Εξέταση και κατανόηση του κώδικα που σχετίζεται με τις προγραμματισμένες αλλαγές (μέχρι 25 Μαΐου, 2004)
- Υλοποίηση Αλλαγών (μέχρι 5 Ιουνίου)
- Εφαρμογή και Εξέταση Αποτελεσμάτων (μέχρι 6 Ιουνίου)
- Συγγραφή Τελικού Παραδοταίου (μέχρι 9 Ιουνίου, 2004)

Στο κομμάτι της Επιλογής και Σχεδιασμού των Αλλαγών αποφασίστηκε πως θα δημιουργηθούν 6 (επτά) νέα αρχεία :

- [RightWrongQuestion](#) : Περιέχει πληροφορίες για τον τρόπο που είναι δομημένη η ερώτηση και οι απαντήσεις της.
- [RightWrongEditPanel](#) : Περιέχει πληροφορίες για τη μορφή της ερώτησης στο πάνελ που τη διαμορφώνει ο καθηγητής (server).
- [RightWrongAnswer](#) : Περιέχει πληροφορίες για τον τρόπο που είναι δομημένη η απάντηση.
- [RightWrongAnswerPanel](#) : Περιέχει πληροφορίες για τον τρόπο που εμφανίζεται το πάνελ της απάντησης στους μαθητές (clients).
- [RightWrongResult](#) : Περιέχει πληροφορίες για τον τρόπο που είναι δομημένα τα αποτελέσματα.
- [RightWrongResultPanel](#) : Περιέχει πληροφορίες για τον τρόπο που εμφανίζονται οι απαντήσεις στο πάνελ του καθηγητή.

Η απόφαση για την δημιουργία των συγκεκριμένων αρχείων με τα συγκεκριμένα ονόματα δεν έγινε αυθαίρετα αλλά σε συνάρτηση με τη μορφή που έχει ο κώδικας που αφορά τους υπόλοιπους τύπους ερωτήσεων, προσπαθώντας να υπάρχει ομοιομορφία και να ακολουθηθεί η αρχική δομή του έργου.

Επιπλέον χρειάστηκε να προστεθούν κάποια στοιχεία που αφορούν τον τύπο ερώτησης Σωστό – Λάθος σε έναν πίνακα που εμπεριέχει στοιχεία για όλους τους τύπους ερωτήσεων.

Η συνολική συμβολή κώδικα παρατίθεται στο παράρτημα Α.

🌐 Υλοποίηση

Οι προγραμματισμένες αλλαγές έγιναν στην έκδοση 0.5 του έργου. Σε αυτήν την διάρκεια δημιουργήθηκαν οι εκδόσεις 0.6 και 0.6.1. Στις εκδόσεις αυτές είχαν γίνει κάποιες αλλαγές που αφορούσαν και τον τύπο ερώτησης Σωστό – Λάθος. Συγκεκριμένα χρειάστηκε να προστεθεί ένα ακόμα αρχείο :

- [RightWrongDocumentWriter](#) : Περιέχει πληροφορίες για την εξαγωγή της ερώτησης και των αποτελεσμάτων σε αρχείο pdf.

🌐 Εργαλεία

Τα εργαλεία που χρησιμοποιήθηκαν είναι τα εξής :

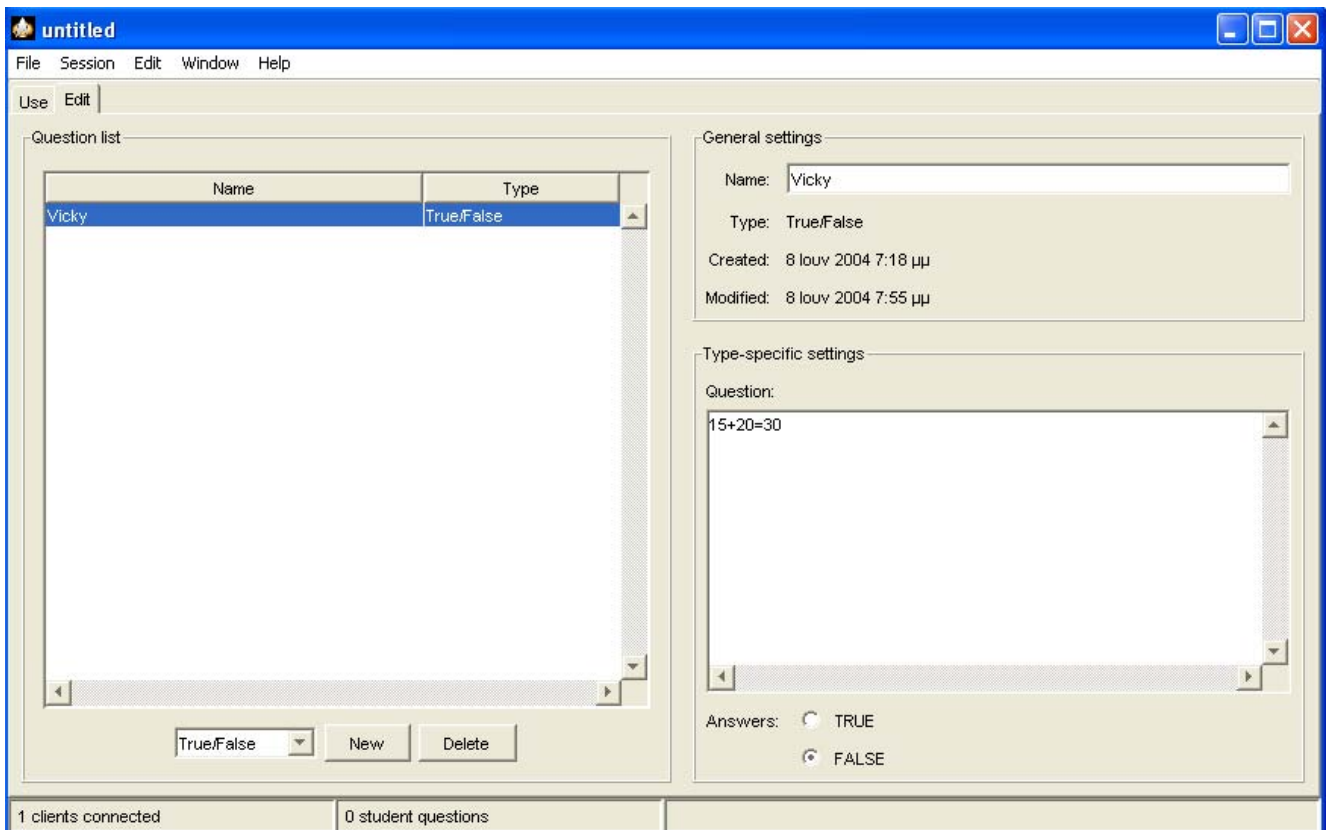
- Το Ant
- Για τις πρώτες αλλαγές το Textpad
- Για τις αλλαγές που αφορούν το τελευταίο αρχείο το Eclipse

4. ΑΠΟΤΕΛΕΣΜΑΤΑ

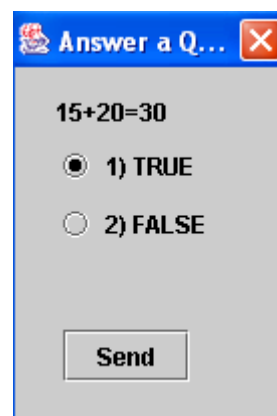
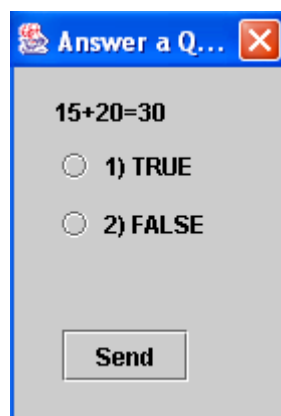
🌐 Εμφάνιση

Παρακάτω παρατίθενται κάποια παραδείγματα που δείχνουν την λειτουργία της εφαρμογής και τον τρόπο που χρησιμοποιείται ο τύπος ερώτησης Σωστό – Λάθος.

Αρχικά φαίνεται το πάνελ του server, όπου ο καθηγητής διαμορφώνει την ερώτηση που θέλει να απευθύνει. Συμπληρώνει το όνομα της ερώτησης, την ερώτηση και επιλέγει αν είναι σωστή ή λάθος.



Στη συνέχεια φαίνεται το πάνελ που εμφανίζεται στους μαθητές (clients). Στην πρώτη περίπτωση είναι πριν ο μαθητής επιλέξει την απάντηση που θεωρεί σωστή και στη δεύτερη αφού έχει επιλέξει ότι είναι σωστή η πρόταση που του δίνεται. Στην συνέχεια πατάει το send για να αποσταλούν τα αποτελέσματα στο πάνελ του server.



Παρακάτω φαίνονται τρεις περιπτώσεις αποτελεσμάτων που μπορεί να εμφανιστούν στο πάνελ του server.

Στην πρώτη περίπτωση, ο μαθητής έχει απαντήσει σωστά κι έτσι η απάντηση του φαίνεται με πράσινη ράβδο. Εδώ πρέπει να αναφερθεί ότι αν υπήρχαν περισσότεροι clients συνδεδεμένοι στον server, αυτό θα φαινόταν στο πάνελ και θα υπήρχε πράσινη ράβδος για τον αριθμό των μαθητών που απάντησαν σωστά και κόκκινη για αυτούς που απάντησαν λάθος. Κάτι τέτοιο δεν ήταν εφικτό να γίνει λόγω της έλλειψης περισσότερων του ενός υπολογιστών, αλλά η λειτουργία αυτή υποστηρίζεται. Γι ' αυτό το λόγο θα αρκεστούμε στην απάντηση του ενός μαθητή.

Στην δεύτερη περίπτωση, ο μαθητής έχει απαντήσει λάθος κι έτσι η απάντησή του φαίνεται με κόκκινη ράβδο.

Υπάρχει ακόμα μια περίπτωση όπου μια πρόταση μπορεί να μην είναι ούτε σωστή ούτε λάθος. Για παράδειγμα η πρόταση : "Η θάλασσα είναι γαλαζοπράσινη". Η θάλασσα μπορεί να είναι γαλαζοπράσινη, αλλά μπορεί να είναι και σκούρο μπλε ή απλά γαλάζια. Σε αυτές τις περιπτώσεις, όπου ο καθηγητής κατά τη διαμόρφωση της ερώτησης δεν έχει επιλέξει ούτε το σωστό ούτε το λάθος, τα στατιστικά εμφανίζονται με μπλε ράβδο.

The screenshot shows a software interface for a quiz system. The window is titled "untitled" and has a menu bar with "File", "Session", "Edit", "Window", and "Help". Below the menu bar are "Use" and "Edit" buttons. The interface is divided into several sections:

- Available questions:** A table with columns "Name" and "Type". It contains one entry: "Vicky" with type "True/False".
- Sent questions:** A table with columns "Name", "Type", and "Answers". It contains two entries for "Vicky" with type "True/False" and "1/1".
- Preview:** A section showing the question "15+20=30" and two radio button options: "1) TRUE" and "2) FALSE". The "2) FALSE" option is selected.
- Result:** A bar chart with "Students" on the y-axis (0 to 1) and "Answer" on the x-axis (1) and (2). A red bar is at (1, 0) and a green bar is at (2, 1).

At the bottom of the interface, there are status indicators: "1 clients connected" and "0 student questions".

untitled

File Session Edit Window Help

Use Edit

Available questions

Name	Type
Vicky	True/False

Send Show solution in preview area.

Sent questions

Name	Type	Answers
Vicky	True/False	1/1
Vicky	True/False	1/1

Stop Question Remove Question Clear Table

Preview

15+20=30

1) TRUE

2) FALSE

Result

Answer	Students
1)	1
2)	0

1 clients connected 0 student questions

untitled

File Session Edit Window Help

Use Edit

Available questions

Name	Type
Θάλασσα	True/False

Send Show solution in preview area.

Sent questions

Name	Type	Answers
Vicky	True/False	1/1
Vicky	True/False	1/1
Θάλασσα	True/False	1/1

Stop Question Remove Question Clear Table

Preview

Η θάλασσα είναι γαλαζοπράσινη.

1) TRUE

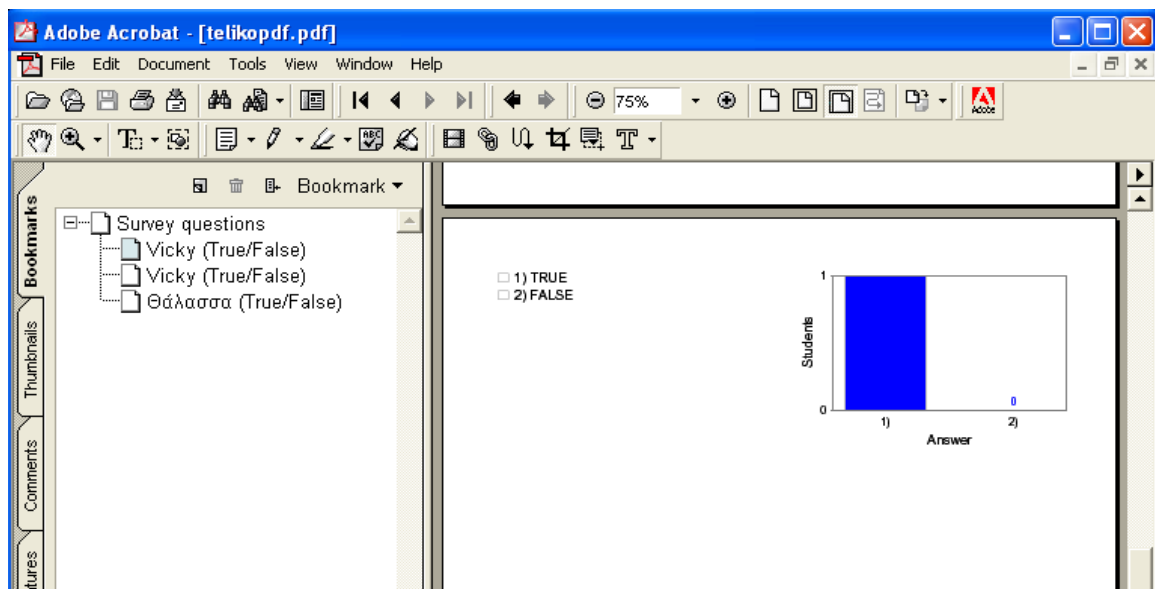
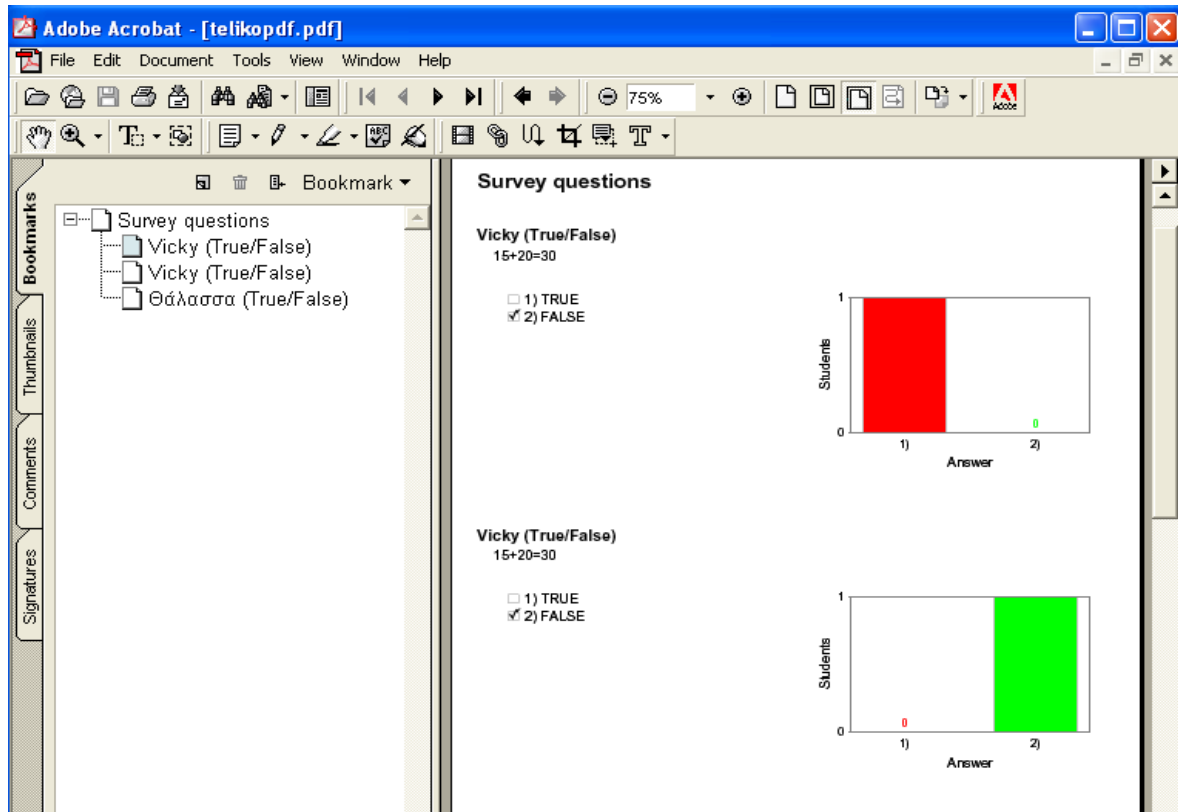
2) FALSE

Result

Answer	Students
1)	1
2)	0

1 clients connected 0 student questions

Στη συνέχεια, δίνεται η δυνατότητα στον καθηγητή να εξαγει τις ερωτήσεις και τα αποτελέσματα σε ένα pdf αρχείο.



🌐 Συνεργασία

Η συνεργασία με τον διαχειριστή του έργου, κύριο Oliver Trachsel ήταν και παραμένει άφογη. Από την πρώτη στιγμή ήταν πολύ θετικός απέναντι στις αλλαγές που πρότεινα και συνεχώς ενθάρρυνε την επικοινωνία. Ήταν πάντα πολύ πρόθυμος να απαντήσει σε τυχόν ερωτήσεις και συνήθως το έκανε στο διάστημα της ίδιας μέρας.

Του έστειλα τις αλλαγές που είχα κάνει και η απάντηση του ήταν πολύ θετική, ζητώντας μου να κάνω μόνο τρεις διορθώσεις. Επιπλέον μου επισήμανε πως οι αλλαγές μου θα υιοθετηθούν στην επόμενη έκδοση και θα προσθέσει το όνομά μου στη λίστα των ανθρώπων που έχουν συνεισφέρει στο έργο. Οι διορθώσεις που επισήμανε έγιναν και έχουν αποσταλεί.

Στο παράρτημα Β υπάρχει όλη η αλληλογραφία που ανταλλάχτηκε κατά τη διάρκεια της συνεργασίας με τον κύριο Trachsel.

5. ΣΥΜΠΕΡΑΣΜΑΤΑ

🌐 Κριτική Έργου

Το έργο χαρακτηρίζεται από μια πολύ προσεγμένη δουλειά, με συνοχή και ομοιομορφία. Είναι πολύ καλά δομημένο, διευκολύνοντας ακόμα και ανθρώπους με μικρή εμπειρία σε έργα ανοικτού λογισμικού και γενικότερα σε κώδικα να κατανοήσουν τη δομή και τη λειτουργία του.

Επιπλέον, η ιδέα του έργου είναι πολύ ενδιαφέρουσα, καθώς μπορεί να διευκολύνει καθηγητές και μαθητές στον τρόπο κατανόησης του μαθήματος, συνδυάζοντας τον παραδοσιακό τρόπο μάθησης με τα πλεονεκτήματα της χρήσης της τεχνολογίας.

Εδώ, πρέπει να αναφερθεί ότι για να μεταγλωττιστεί ο πηγαίος κώδικας θα πρέπει να δημιουργηθεί ένας φάκελος lib (ο οποίος θα περιέχει τις βιβλιοθήκες που απαιτούνται) και να μετονομαστούν δύο αρχεία (με όνομα OSXMenus.java). Επιπλέον χρειάζεται να γίνουν κάποιες αλλαγές στο αρχείο build.xml, για να δημιουργηθεί ο φάκελος dist (αλλιώς ζητάει κωδικό client). Αυτές οι παρατηρήσεις δεν αναφέρονται στο αρχείο README, κάτι που δυσκολεύει πολύ τους νέους χρήστες του έργου.

🌐 Δυσκολίες

Οι δυσκολίες που συναντήθηκαν είναι οι εξής :

- Δυσκολία στην εύρεση έργου, που οφειλόταν σε έλλειψη εμπειρίας και γνώσεων πάνω στο αντικείμενο.
- Πρόβλημα στο να μεταγλωττιστεί ο κώδικας, καθώς χρειαζόταν να γίνουν οι αλλαγές που αναφέρθηκαν παραπάνω και δεν υπήρχε πουθενά αναφορά γι ' αυτό.
- Χρειάστηκε να διαβαστεί και να κατανοηθεί μεγάλο μέρος του ήδη υπάρχον κώδικα για την υλοποίηση της αλλαγής.

- Χρειάστηκε να διαβαστεί ένα μεγάλο κομμάτι από το documentation του eclipse, καθώς δεν υπήρχε προηγούμενη εμπειρία πάνω στο συγκεκριμένο εργαλείο και επιπλέον το μέγεθος του αρχείου αυτού είναι πολύ μεγάλο, δυσκολεύοντας την αναζήτηση συγκεκριμένων εφαρμογών.
- Πρόβλημα στη δημιουργία φακέλου dist (στην έκδοση 0.6.1), καθώς ζητούσε κωδικό όνομα χρήστη και χρειαζόταν κάποια παραμετροποίηση το αρχείο build.xml.

🌐 Πλεονεκτήματα

Τα πλεονεκτήματα της συγκεκριμένης εργασίας αφορούν περισσότερο την διαδικασία σχεδιασμού και υλοποίησής της και όχι τόσο τα αποτελέσματά της. Μέσα από την ανάπτυξη αυτού του έργου αποκτήθηκε γνώση, εμπειρία και εξοικείωση γύρω από τα εξής θέματα :

- Την έννοια των έργων ανοικτού λογισμικού
- Τον τρόπο αναζήτησης έργων ανοικτού λογισμικού στο διαδίκτυο
- Την διαδικασία ανάπτυξης ενός έργου ανοικτού λογισμικού
- Τον τρόπο επικοινωνίας ανθρώπων που εργάζονται σε ένα τέτοιο έργο
- Γλώσσα προγραμματισμού Java (έννοιες όπως γραφικό περιβάλλον διασύνδεσης – GUI, γεγονότα, διαχειριστές διάταξης, κ.τ.λ.)
- Εργαλεία όπως Textpad, Ant, Eclipse, CSDiff

Η συγκεκριμένη εργασία είναι ίσως η πιο δύσκολη αλλά και πιο επικοινωνιακή εργασία στα τέσσερα χρόνια του πανεπιστημίου, καθώς χρειαζόταν πολύ δουλειά και διάβασμα αλλά επιπλέον άνοιγε την πόρτα σε έναν άγνωστο μέχρι στιγμής κόσμο, τον κόσμο της πληροφορικής και του ανοικτού λογισμικού.

Τέλος, δεν θα έπρεπε να παραληφθεί η ικανοποίηση και η χαρά του να αναγνωρίζεται η δουλειά ενός ανθρώπου και να υιοθετείται από τους συνεργάτες του, που στην συγκεκριμένη περίπτωση σημαίνει την ενσωμάτωση των αλλαγών στο κεντρικό έργο.

ΠΑΡΑΡΤΗΜΑΤΑ

- **ΠΑΡΑΡΤΗΜΑ Α**

Παρατίθενται τα αρχεία που προστέθηκαν στον κώδικα του έργου

- **RightWrongQuestion**

```
/*
 * LectComm - ETH Lecture Communicator
 *
 * Copyright (c) 2003 Oliver Trachsel <oliver.trachsel/at/inf.ethz.ch>
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */
package lectcomm.qtypes.rightwrong;

import java.io.Serializable;
import lectcomm.qtypes.*;
import org.w3c.dom.*;

/**
 * Data-container for a RightWrong question.
 */
public class RightWrongQuestion
    extends Question
    implements Serializable
{
    /** The question text. */
    private String questionText = "";

    /** The number of answers. */
    public final static int NOFANSWERS = 2;
```

```

/** A list of Strings with the answer texts. */
private String[] answers = {"TRUE", "FALSE"};

/** The indexes of the correct answer(s) */
private boolean[] correctMask = new boolean[NOFANSWERS];

/** Create a new empty RightWrong question instance. */
public RightWrongQuestion() {
    super(QuestionTypeFactory.getType("rightwrong"));
}

/**
 * @return answer number <code>index</code>.
 */
public String getAnswer(int index) {
    return answers[index];
}

/**
 * @return the correct-answer mask. The mask indicates for each
 * possible answer if it is correct or not.
 */
public boolean[] getCorrectMask() {
    return correctMask;
}

/**
 * @param set the correct-answer mask.
 */
public void setCorrectMask(boolean[] correctMask) {
    if (correctMask.length != NOFANSWERS) {
        throw new IllegalArgumentException("correctMask must
have length "+NOFANSWERS+", not "+correctMask.length);
    }
    this.correctMask = correctMask;
    dataChanged();
}

/**
 * @return the actual question text.
 */
public String getQuestionText() {
    return questionText;
}

/**
 * @param questionText the actual question text.
 */
public void setQuestionText(String questionText) {
    if (questionText == null || this.questionText == null |

```

```

        !questionText.equals(this.questionText))
    {
        this.questionText = questionText;
        dataChanged();
    }
}

/**
 * Export data to a XML DOM node.
 * @see lectcomm.qtypes.Question#exportTo(org.w3c.dom.Node)
 */
protected void exportTypeDataTo(Element rootElement) {
    Document document = rootElement.getOwnerDocument();

    /* The question element. */
    Element questionElement =
document.createElement("question");
    Text text = document.createTextNode(questionText);
    questionElement.appendChild(text);
    rootElement.appendChild(questionElement);

    /* One DOM element per answer. */
    for (int i = 0; i < NOFANSWERS; i++) {
        Element answerElement =
document.createElement("answer");
        answerElement.setAttribute("number", String.valueOf(i));
        answerElement.setAttribute("isCorrect",
String.valueOf(correctMask[i]));
        text = document.createTextNode(getAnswer(i).toString());
        answerElement.appendChild(text);
        rootElement.appendChild(answerElement);
    }
}

/**
 * Import data from XML DOM element.
 * @see
lectcomm.qtypes.Question#importFrom(org.w3c.dom.Element)
 */
protected void importTypeDataFrom(Element rootElement) throws
Exception {
    NodeList nodeList =
rootElement.getElementsByTagName("question");

    /* Must have exactly one question element. */
    if (nodeList.getLength() != 1)
        throw new Exception("Mismatched data. Wrong
DOM node must have exactly one question element.");

    Element questionElement = (Element) nodeList.item(0);

```

```

        /* Extract question. */
        if (questionElement.getFirstChild() != null)
            questionText =
questionElement.getFirstChild().getNodeValue();
        else
            questionText = "";

        /* Extract answers. */
        nodeList = rootElement.getElementsByTagName("answer");

        for (int i = 0; i < nodeList.getLength(); i++) {
            Element answer = (Element) nodeList.item(i);
            int number =
Integer.parseInt(answer.getAttribute("number"));
            /* Ignore answers with an invalid number. */
            if (number < NOFANSWERS) {
                correctMask[number] =
Boolean
.valueOf(answer.getAttribute("isCorrect"))
                .booleanValue();
            }
        }
    }
    /**
     * Make an indepened copy of this MatchingQuestion.
     */
    public Object clone() {
        RightWrongQuestion copy = (RightWrongQuestion)
super.clone();

        copy.answers = (String []) answers.clone();

        if (correctMask != null)
            copy.correctMask = (boolean[]) correctMask.clone();

        return copy;
    }

    /**
     * Make a copy of this matching question but hide the correct mask.
     * @see lectcomm.qtypes.Question#cloneWithoutSolution()
     */
    public Question cloneWithoutSolution() {
        RightWrongQuestion question = (RightWrongQuestion) clone();
        question.correctMask = null;
        return question;
    }
}

```


RightWrongEditPanel

```
/*
 * LectComm - ETH Lecture Communicator
 *
 * Copyright (c) 2003 Oliver Trachsel <oliver.trachsel/at/inf.ethz.ch>
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */
package lectcomm.qtypes.rightwrong;

import java.awt.*;
import java.awt.event.*;

import javax.swing.*;

import lectcomm.qtypes.*;
import lectcomm.util.*;
import lectcomm.util.BaseTextArea;

/** A JPanel to display and edit a RightWrongQuestion. */
public class RightWrongEditPanel
    extends JPanel
    implements QuestionEditPanel
{
    JLabel questionLabel = new JLabel();
    JTextArea questionTextArea = new BaseTextArea();
    JLabel answersLabel = new JLabel();
    JLabel rightLabel = new JLabel();
    JLabel wrongLabel = new JLabel();
    JRadioButton[] answerBoxes =
        {
            new JRadioButton(),
            new JRadioButton()};
    ButtonGroup answerBoxGroup = new ButtonGroup();

    JScrollPane questionScrollPane = new JScrollPane();

```

```

/** The currently displayed question. */
private RightWrongQuestion displayQuestion;

public RightWrongEditPanel() {
    try {
        initComponents();
        initChangeListeners();
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

/**
 * Install change listeners for all editable objects.
 */
private void initChangeListeners() {

    /* Handler for question text. */
    questionTextArea.addFocusListener(new FocusAdapter() {
        public void focusLost(FocusEvent e) {
            String text;
            if (!(text = questionTextArea.getText())

.equals(displayQuestion.getQuestionText())) {
                displayQuestion.setQuestionText(text);
            }
        }
    });

    /* Handler for answer radiobuttons. */
    ActionListener radiobuttonListener = new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            storeCorrectMask();
        }
    };

    for (int i = 0; i < answerBoxes.length; i++)
        answerBoxes[i].addActionListener(radiobuttonListener);
}

/** Stores the current answer-checkbox configuration to
 * the displayed question's data object. */
private void storeCorrectMask() {
    boolean[] mask = displayQuestion.getCorrectMask();

    if (mask == null || mask.length != answerBoxes.length) {
        mask = new boolean[answerBoxes.length];
    }
}

```

```

        for (int i = 0; i < mask.length; i++) {
            mask[i] = answerBoxes[i].isSelected();
        }

        displayQuestion.setCorrectMask(mask);
    }

/**
 * Deselect and group all answer-radiobuttons in one button group
 */
private final void groupAnswerRadiobuttons() {
    boolean deselectBoxes = false;
    for (int i = 0; i < answerBoxes.length; i++) {
        if (deselectBoxes) {
            answerBoxes[i].setSelected(false);
        } else if (answerBoxes[i].isSelected()) {
            deselectBoxes = true;
        }
        answerBoxGroup.add(answerBoxes[i]);
    }
}

/**
 * Initialize all GUI components.
 */
void initComponents() throws Exception {
    this.setLayout(new GridBagLayout());
    questionLabel.setRequestFocusEnabled(true);
    questionLabel.setText("Question:");
    questionTextArea.setForeground(Color.black);
    questionTextArea.setDebugGraphicsOptions(0);
    questionTextArea.setDoubleBuffered(false);
    questionTextArea.setText("");
    questionTextArea.setColumns(30);
    questionTextArea.setRows(5);
    questionTextArea.setTabSize(8);
    answersLabel.setText("Answers:");
    rightLabel.setRequestFocusEnabled(true);
    rightLabel.setText("TRUE");
    wrongLabel.setRequestFocusEnabled(true);
    wrongLabel.setText("FALSE");
    questionScrollPane.getViewPort().add(questionTextArea, null);
    questionScrollPane.setHorizontalScrollBarPolicy(
        JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
    questionScrollPane.setVerticalScrollBarPolicy(
        JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
    questionScrollPane.setAlignmentY((float) 0.5);
}

```

```

        this.add(
            questionLabel,
            new GridBagConstraints(
                0, 0, 1, 1, 0.0, 0.0,
                GridBagConstraints.WEST,
GridBagConstraints.NONE,
                new Insets(5, 5, 0, 5), 0, 0));

        this.add(
            questionScrollPane,
            new GridBagConstraints(
                0, 1, 3, 1, 1.0, 1.0,
                GridBagConstraints.CENTER,
GridBagConstraints.BOTH,
                new Insets(5, 5, 5, 5), 0, 0));

        this.add(
            answersLabel,
            new GridBagConstraints(
                0, 3, 1, 1, 0.0, 0.0,
                GridBagConstraints.WEST,
GridBagConstraints.NONE,
                new Insets(5, 5, 5, 5), 0, 0));

        this.add(
            rightLabel,
            new GridBagConstraints(
                2, 3, 1, 1, 0.0, 0.0,
                GridBagConstraints.WEST,
GridBagConstraints.NONE,
                new Insets(5, 5, 5, 5), 0, 0));

        this.add(
            wrongLabel,
            new GridBagConstraints(
                2, 4, 1, 1, 0.0, 0.0,
                GridBagConstraints.WEST,
GridBagConstraints.NONE,
                new Insets(5, 5, 5, 5), 0, 0));

        for (int i = 0; i < answerBoxes.length; i++) {
            this.add(
                answerBoxes[i],
                new GridBagConstraints(
                    1, i + 3, 1, 1, 0.0, 0.0,
                    GridBagConstraints.WEST,
GridBagConstraints.NONE,
                    new Insets(0, 5, 0, 0), 0, 0));
        }
    }
}

```

```

/**
 * @see
lectcomm.qtypes.QuestionEditPanel#display(lectcomm.qtypes.Question)
 */
public void display(Question question) {
    /* Set question. */
    this.displayQuestion = (RightWrongQuestion) question;
    questionTextArea.setText(displayQuestion.getQuestionText());

    /* Set radiobuttonboxes. */
    boolean[] answerMask = displayQuestion.getCorrectMask();
    for (int i = 0; i < answerBoxes.length; i++) {
        if (answerMask != null && i < answerMask.length)
            answerBoxes[i].setSelected(answerMask[i]);
        else
            answerBoxes[i].setSelected(false);
    }
}
}

```

```
/*
 * LectComm - ETH Lecture Communicator
 *
 * Copyright (c) 2003 Oliver Trachsel <oliver.trachsel/at/inf.ethz.ch>
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */
package lectcomm.qtypes.rightwrong;

import lectcomm.qtypes.Answer;

/**
 * Data-container for a RightWrong answer.
 */
public class RightWrongAnswer
    extends Answer
{
    /** The mask which indicates the chosen answer
     * (== elements set to true) */
    private boolean [] answerMask;

    /**
     * @return the given-answers mask
     */
    public boolean[] getAnswerMask() {
        return answerMask;
    }

    /**
     * @param answerMask the boolean mask indicating the chosen/not
chosen
     * answer.
     */
}
```

```
public void setAnswerMask(boolean[] answerMask) {  
    this.answerMask = answerMask;  
}  
}
```

```
/*
 * LectComm - ETH Lecture Communicator
 *
 * Copyright (c) 2003 Oliver Trachsel <oliver.trachsel/at/inf.ethz.ch>
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */
```

```
package lectcomm.qtypes.rightwrong;
```

```
import java.awt.*;
import javax.swing.*;
```

```
import lectcomm.qtypes.*;
import lectcomm.util.*;
```

```
/**
 * GUI panel to answer a RightWrong question.
 */
public class RightWrongAnswerPanel
    extends JPanel
    implements AnswerPanel
{

    /** One radiobutton per possible answer. */
    JRadioButton [] answerBoxes;

    /** @see AnswerPanel#display(Question) */
    public void display(Question qst, boolean showCorrectAnswers) {
        RightWrongQuestion question = (RightWrongQuestion) qst;

        /* Reset display. */
        this.removeAll();
    }
}
```



```

GridBagLayout layout = new GridBagLayout();
this.setLayout(layout);

/* Question area. */
CommentTextComponent questionArea = new
CommentTextComponent(question.getQuestionText());

this.add(questionArea,
    new GridBagConstraints(
        0, 0, 2, 1, 1.0, 0.0,
        GridBagConstraints.WEST,
        GridBagConstraints.HORIZONTAL,
        new Insets(14, 20, 0, 20),
        0, 0));

/* The answers with radiobuttons. */
answerBoxes = new JRadioButton[question.NOANSWERS];
ButtonGroup buttonGroup = new ButtonGroup();

boolean [] correctMask = question.getCorrectMask();

for (int i=0; i<answerBoxes.length; i++) {
    JRadioButton radiobutton = new JRadioButton();
    answerBoxes[i] = radiobutton;
    if (showCorrectAnswers) {
        answerBoxes[i].setEnabled(false);
        answerBoxes[i].setSelected(i < correctMask.length
&& correctMask[i]);
    } else {
        radiobutton.setSelected(false);
    }

    buttonGroup.add(radiobutton);
    this.add(radiobutton,
        new GridBagConstraints(
            0, i+1, 1, 1, 0.0, 0.0,
            GridBagConstraints.NORTHEAST,
            GridBagConstraints.NONE,
            new Insets(8, 20, 0, 2),
            0, 0));

    MultilineLabel textLabel = new MultilineLabel((i+1)+"
"+question.getAnswer(i), 25);
    this.add(textLabel,
        new GridBagConstraints(
            1, i+1, 1, 1, 0.0, 0.0,
            GridBagConstraints.WEST,
            GridBagConstraints.NONE,
            new Insets(8, 2, 0, 20),
            0, 0));

```

```

    }
    this.add(new JPanel(),
        new GridBagConstraints(
            0, answerBoxes.length+1, 2, 1, 0.0, 1.0,
            GridBagConstraints.WEST,
GridBagConstraints.VERTICAL,
            new Insets(12, 0, 0, 0), 0, 0));
    }

    /**
     * @see AnswerPanel#storeAnwer(Answer)
     */
    public void storeAnswer(Answer answer) {
        boolean [] answerMask = new boolean[answerBoxes.length];
        for (int i=0; i<answerMask.length; i++) {
            answerMask[i] = answerBoxes[i].isSelected();
        }
        ((RightWrongAnswer) answer).setAnswerMask(answerMask);
    }
}

```

RightWrongResult

```
/*
 * LectComm - ETH Lecture Communicator
 *
 * Copyright (c) 2003 Oliver Trachsel <oliver.trachsel/at/inf.ethz.ch>
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */
package lectcomm.qtypes.rightwrong;

import lectcomm.qtypes.*;

/**
 * Data-container and manager for the answers to a rightwrong-type question
 */
public class RightWrongResult extends Result {

    /** Counts how many times each answer has been given. */
    private int [] answerCounter;

    /** The associated SentQuestion. */
    private SentQuestion sentQuestion;

    /** Handle newly arrived answers. */
    protected void answerAdded(Answer ans) {
        boolean [] answerMask = ((RightWrongAnswer)
ans).getAnswerMask();
        for (int i=0; i<answerMask.length && i<answerCounter.length;
i++) {
            if (answerMask[i])
                answerCounter[i]++;
        }
    }

    /** Init the answer counters if not already done. */
    protected final void init(SentQuestion sentQuestion) {
```

```

        this.sentQuestion = sentQuestion;
        RightWrongQuestion question = (RightWrongQuestion)
sentQuestion.getQuestion();
        answerCounter = new int[question.NOANSWERS];
    }

    /**
     * @return an integer array with the number of times each answer was
chosen.
     */
    public int[] getAnswerCounter() {
        return answerCounter;
    }

    /**
     * Get the associated SentQuestion object.
     * @return the sent question descriptor
     */
    public SentQuestion getSentQuestion() {
        return sentQuestion;
    }
}

```

RightWrongResultPanel

```
/*
 * LectComm - ETH Lecture Communicator
 *
 * Copyright (c) 2003 Oliver Trachsel <oliver.trachsel/at/inf.ethz.ch>
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */
```

```
package lectcomm.qtypes.rightwrong;
```

```
import java.awt.*;
```

```
import org.jfree.chart.*;
import org.jfree.chart.axis.*;
import org.jfree.chart.labels.StandardCategoryItemLabelGenerator;
import org.jfree.chart.plot.*;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.renderer.BarRenderer;
import org.jfree.data.*;
```

```
import lectcomm.qtypes.*;
import lectcomm.util.CategoryColorBarRenderer;
```

```
/**
 * Panel displaying the results of a rightwrong quiz.
 */
public class RightWrongResultPanel
    extends ChartPanel
    implements ResultPanel
{
    /** The currently displayed result. */
    private RightWrongResult displayResult;

    /** Percentage-bar color for a correct answer. */
```

```

public final static Color COLOR_CORRECT = Color.green;

/** Percentage-bar color for an incorrect answer. */
public final static Color COLOR_INCORRECT = Color.red;

/** Percentage-bar color for answers where there is no correct answer
specified. */
public final static Color COLOR_NEUTRAL = Color.blue;

/** Construct a new RightWrong result panel. */
public RightWrongResultPanel() {
    super(ChartFactory.createXYLineChart(
        null, null, null, null,
        PlotOrientation.VERTICAL,
        false, false, false));
    setPopupMenu(null);
}

/** Display the given result. */
public void display(Result result) {

    if (displayResult != null) {
        displayResult.removeResultObserver(this);
    }
    displayResult = (RightWrongResult) result;

    /* Display chart. */
    JFreeChart chart = createChart((RightWrongQuestion)
displayResult.getSentQuestion().getQuestion(), displayResult);
    setChart(chart);
    result.setResultObserver(this);
}

/**
 * A chart factory to generate RightWrong result charts (using
JFreeChart).
 */
static JFreeChart createChart(RightWrongQuestion question,
RightWrongResult result) {
    /* Customize labels and colors. */
    boolean correctAnswerSpecified = false;
    boolean correctMask[] = question.getCorrectMask();
    for (int i=0; i<correctMask.length; i++) {
        if (correctMask[i]) {
            correctAnswerSpecified = true;
            break;
        }
    }

    int [] answerCounter = result.getAnswerCounter();

```

```

        Color [] categoryColors = new Color[answerCounter.length];
        for (int i=0; i<answerCounter.length; i++) {
            Color color = correctMask[i]
                ? COLOR_CORRECT
                : ((correctAnswerSpecified) ?
COLOR_INCORRECT : COLOR_NEUTRAL);
            categoryColors[i] = color;
        }

        /* Set the chart up. */
        CategoryAxis categoryAxis = new CategoryAxis("Answer");
        ValueAxis valueAxis = new NumberAxis("Students");
        int nofClients = result.getSentQuestion().getNofClients();
        valueAxis.setRange(0, Math.max(1, nofClients));

        valueAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits());

        BarRenderer renderer = new
CategoryColorBarRenderer(categoryColors, true);
        renderer.setItemLabelsVisible(Boolean.TRUE);
        renderer.setItemLabelGenerator(new
StandardCategoryItemLabelGenerator());

        /* Define the plot */
        CategoryPlot plot = new CategoryPlot(getDataset(result),
categoryAxis, valueAxis, renderer);
        plot.setOrientation(PlotOrientation.VERTICAL);
        JFreeChart chart = new JFreeChart(null,
JFreeChart.DEFAULT_TITLE_FONT, plot, false);
        return chart;
    }

    /** Generate a StatsDataseries instance that can be used for plotting.
*/
    static CategoryDataset getDataset(RightWrongResult result) {
        DefaultCategoryDataset dataset = new
DefaultCategoryDataset();
        int [] answerCounter = result.getAnswerCounter();
        for (int i=0; i<answerCounter.length; i++) {
            dataset.addValue(answerCounter[i], "", (i+1)+"");
        }
        return dataset;
    }

    /**
     * @see
lectcomm.control.ResultObserver#resultChanged(lectcomm.qtypes.Result)
*/
    public void resultChanged(Result result) {

```

```
        if (result != displayResult) {
            throw new IllegalStateException("Got informed about a
result that I'm not currently displaying.");
        }

        getChart().getCategoryPlot().setDataset(getDataset(displayResult));
    }
}
```



```
/*
 * LectComm - ETH Lecture Communicator
 *
 * Copyright (c) 2004 Oliver Trachsel <oliver.trachsel/at/inf.ethz.ch>
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */
package lectcomm.qtypes.rightwrong;

import java.awt.Color;
import java.awt.Graphics2D;
import java.awt.geom.Rectangle2D;
import java.awt.image.BufferedImage;
import java.io.IOException;

import org.jfree.chart.JFreeChart;

import com.lowagie.text.*;
import com.lowagie.text.Section;
import com.lowagie.text.pdf.*;
import com.lowagie.text.pdf.PdfContentByte;

import lectcomm.qtypes.*;
import lectcomm.qtypes.DocumentWriter;
import lectcomm.resources.*;
import lectcomm.resources.ExportDocumentProperties;

/**
 * Exporting facility for RightWrong questions to add a question with choices
 * and a result to a page-oriented document.
 */
public class RightWrongDocumentWriter
    implements DocumentWriter, ExportDocumentProperties
{
```

```

/**
 * Add the given <code>RightWrongQuestion</code> and
<code>RightWrongResult</code> to
 * the a document section.
 *
 * @see
lectcomm.qtypes.DocumentWriter#write(lectcomm.qtypes.Question,
lectcomm.qtypes.Result, com.lowagie.text.Section)
 */
public void write(Question qst, Result res, Section section,
PdfContentByte pdfContentByte)
    throws IOException, DocumentException
{
    RightWrongQuestion question = (RightWrongQuestion) qst;
    RightWrongResult result = (RightWrongResult) res;

    /* Add a parapgraph with the question text. */
    Paragraph para = new Paragraph(question.getQuestionText(),
TEXT_FONT);
    section.add(para);

    /* Add a table for the all answers & the result chart. */
    Table table = new Table(3, Math.max(1,
question.NOFANSWERS));
    table.setWidth(100);
    table.setWidths(new int[] {8, 42, 50});
    table.setTableFitsPage(true);
    table.setAlignment(Table.ALIGN_LEFT);
    table.setBorder(Table.NO_BORDER);

    boolean [] correctMask = question.getCorrectMask();
    Image emptyBoxImage =
Image.getInstance(ServerResources.EMPTY_BOX_ICON.getImage(), null);
    emptyBoxImage.scalePercent(15);
    Image tickedBoxImage =
Image.getInstance(ServerResources.TICKED_BOX_ICON.getImage(), null);
    tickedBoxImage.scalePercent(15);

    Cell cell;
    for (int i=0; i<question.NOFANSWERS; i++) {
        cell = new Cell();
        cell.setVerticalAlignment(Cell.ALIGN_TOP);
        cell.setHorizontalAlignment(Cell.ALIGN_RIGHT);
        cell.setBorder(Cell.NO_BORDER);
        Phrase phrase = new Phrase();
        if (i<correctMask.length && correctMask[i]) {
            phrase.add(new Chunk(tickedBoxImage, 0, -2));
        } else {
            phrase.add(new Chunk(emptyBoxImage, 0, -2));
        }
    }
}

```

```

        phrase.add(new Chunk(" "+(i+1)+" ", TEXT_FONT));
        cell.add(phrase);
        table.addCell(cell, i, 0);

        cell = new Cell();
        phrase = new Phrase();
        phrase.add(new Chunk(question.getAnswer(i),
TEXT_FONT));
        cell.add(phrase);
        cell.setVerticalAlignment(Cell.ALIGN_TOP);
        cell.setHorizontalAlignment(Cell.ALIGN_LEFT);
        cell.setBorder(Cell.NO_BORDER);
        table.addCell(cell, i, 1);
    }

    // construct the result chart.
    JFreeChart chart =
RightWrongResultPanel.createChart(question, result);
    chart.setBackgroundPaint(Color.white);
    Image docImage;

    /* If we're dealing with a PDF create a vectorized image using a
Graphics2D object. */
    if (pdfContentByte != null) {
        PdfTemplate template =
pdfContentByte.createTemplate(CHART_WIDTH, CHART_HEIGHT);
        Graphics2D graphics =
template.createGraphics(CHART_WIDTH, CHART_HEIGHT, new
DefaultFontMapper());
        chart.draw(graphics, new Rectangle2D.Double(0, 0,
CHART_WIDTH, CHART_HEIGHT));
        graphics.dispose();
        docImage = Image.getInstance(template);
    }

    /* If not exporting to PDF create a bitmapped image. */
    else {
        BufferedImage image =
chart.createBufferedImage(CHART_WIDTH, CHART_HEIGHT);
        docImage = Image.getInstance(image, null);
        docImage.scalePercent(IMAGE_SCALING_PERCENT);
    }

    cell = new Cell();
    cell.setVerticalAlignment(Cell.ALIGN_TOP);
    cell.setBorder(Cell.NO_BORDER);
    cell.add(docImage);
    cell.setRowspan(Math.max(1, question.NOFANSWERS));
    table.addCell(cell, 0, 2);

```

```
        /* Add the table to the output. */  
        section.add(table);  
    }  
}
```

- **ΠΑΡΑΡΤΗΜΑ Β**

Παρακάτω παρατίθεται η αλληλογραφία που ανταλλάχτηκε καθ' όλη τη διάρκεια της συνεργασίας με τον διαχειριστή του έργου.

[On May 17, 2004, Vicky Skordaly wrote:](#)

Mr Oliver Trachsel,

My name is Vicky Skordaly and i am in the 4th year of theAthens University of Economics and Business/Hellas. The name of my department is Management Science and Technology. I have chosen to modify ur project from opensource for one of my modules in the uni (Maintenance and Refactoring of Information Systems) and i was wondering if i could count in ur help when adding my feautres. I was thinking of adding a new type of questions (right/wrong) and trying to fix one of the bugs (Switch between one client/multiple clients per IP).

Thank you in advance,
Vicky Skordaly

[On May 18, 2004, Oliver Trachsel wrote:](#)

Vicky,

You are welcome to use the Lecture Communicator project as a basis for your university module. I'll be glad to have a look at your new features and bug fixes and to incorporate them if they are of good quality.

The best way to send me your changes is in the form of patch-files that can be created using the UNIX command diff or automatically from within Eclipse (if you are using Eclipse for development). But if you prefer, you can also send me the Java source files that you have changed or added.

Don't hesitate to contact me if you need further assistance.

Good luck with your project!

Best regards, Oliver

On Jun 08, 2004, Vicky Skordaly wrote :

Mr Oliver Trachsel,

My name is Vicky Skordaly and I am working on a LectComm extension. I have chosen to modify your project from opensource for one of my modules in the uni (Maintenance and Refactoring of Information Systems). We communicated about a month ago and I told you about my plans. I wanted to add a new type of Question (RightWrong). I did that, but I was working with version 0.5, so when I downloaded the release 0.6.1 I realised that I had to add or modify some things. I already did it, but I have got problem when using ant dists. It asks me for a keystore password and i don't know what to do. I am really sorry, maybe it seems ridiculous, but I would appreciate your help.

Thank you in advance,
Vicky Skordaly

On Jun 08, 2004, Oliver Trachsel wrote:

Hi Vicky,

I am glad you're still working on the Lecture Communicator and hope you don't have too much trouble getting along with my source code.

You can disable JAR-signing (and hence the password prompt) by editing line 9 of file build.xml. Currently, this reads (line 8/9):

```
<!-- do not set the following property to avoid jar signing -->
<property name="sign_client_jar" value="yes" />
```

Comment out the setting of the property (add "!--" add the beginning and "--" at the end) to get:

```
<!-- property name="sign_client_jar" value="yes" /-->
```

Now, you shouldn't be asked for a password anymore.

Explanation: The property "sign_client_jar" is now not set anymore and only target "client-jar-unsigned" will be executed, but not client-jar (because it has an if="sign_client_jar" condition).

Further information: Since version 0.6.1 the client applet of the tool is distributed as a signed JAR (i.e. the distributor of the software signs the JAR archive with a digital signature to assert that the software does indeed come from him). When a signed applet is loaded into a web browser, the user is asked if he trusts the person or company that has signed the applet. If he does, the applet has less security restrictions than an unsigned applet. In case of the ETH Lecture Communicator this enables the client to connect to a server other than the host from where the applet was loaded, which is only possible for signed applets. Up to version 0.6 (and for unsigned versions $\geq 0.6.1$) the web page with the applet must be stored on the computer where the LectComm server is ran.

Please do not hesitate to re-contact me if you need any further assistance. Also, I'd be glad if you'd let me know when you've finished your extensions so I can include them in the main distribution.

All the best,
Oliver

P.S.: If you don't mind then please just call me Oliver. Being called Mister makes me feel old. ;-) I finished my own studies just about one year ago.

On Jun 08, 2004, Vicky Skordaly wrote :

Oliver,

Firstly, I would like to thank you for being so helpful. The problem resolved immediately.

Secondly, last night I 've just finished my extensions and I send my changes to you. I would be glad if you liked them and if you could include them in the main distribution. I added a new type of Question (RightWrong).

The attachment contains the rightwrong folder (the folder I added at the location : lectcomm-0.6.1\src\lectcomm\qtypes), the file QuestionTypeFactory (in which I added a line : ,{"rightwrong", "Right Wrong", "lectcomm.qtypes.rightwrong.RightWrong"}) and two diff files showing the changes.

All the best,
Vicky Skordaly

P.S.: Thank you for the perfect coordination. I am waiting your comments on my changes.

On Jun 08, 2004, Oliver Trachsel wrote:

Vicky,

Thanks for sending me your source code. Your new question types seems well implemented and does work almost without problems.

A few remarks:

- RightWrongQuestion.correctMask should not be declared 'transient' anymore. Otherwise, a NullPointerException is thrown when the session is saved and then reloaded. (Transient fields were used before version 0.6, before the method Question.cloneWithoutSolution() was introduced.)

- It would be useful if one could deselect a previously selected checkbox/radio button in the question editing pane (i.e., if either right or wrong was selected to be able to have nothing selected). Not every right/wrong question has a correct solution and if an instructor selected one accidentally she should be able to remove the selection.

- In the user interface I would probably call the question type "True/false" (and the two choices True and False, accordingly) rather than "Right Wrong". This naming scheme is used by most other electronic learning systems (e.g. in Learning Management Systems, like Blackboard or WebCT) and is applicable to a broader range of questions. The package name could be left as is.

I will gladly incorporate your new type in the next release. If you don't have time to make the above changes, don't worry, I will do it myself.

Also, if you don't object, I will add you to the list of contributors on the web site and in the application's About dialog. Would you prefer to have your email address listed or not?

Best regards, Oliver

On Jun 08, 2004, Vicky Skordaly wrote :

Oliver,

Thank you for your remarks. I tried to do the changes you told me and if I understood correct :

The RightWrongQuestion.correctMask is not declared 'transient' anymore. This change happened in the RightWrongQuestion file.

I added a new method, so someone can deselect a previously selected radio button. This change happened in the RightWrongQuestion file.

Now in the user interface the question type's name is "True/false". This change happened in the RightWrongQuestion file, the RightWrongEditPanel file and the QuestionTypeFactory file.

I don't know if you 'd like to change the name of folder, files and classes from RightWrong.... to TrueFalse....I could do that but maybe it takes some time (because it is exam period).

I send you the 3 files I changed. I hope you like them.

I am very glad you liked my work and it would be ok for me to have my email address listed.

I am waiting for your comments on my new changes.

All the best,
Vicky Skordaly