Il pacchetto minerva*

Grazia Messineo, Salvatore Vassallo

Sommario

Vengono illustrati gli sviluppi del pacchetto minerva, già introdotto al convegno GUIT 2012, per la somministrazione di esercitazioni e prove di valutazione al PC.

Abstract

We present the developments of the package minerva, already shown in 2012 GUIT conference, to produce exercises and exams which can be given by using a PC.

1 Introduzione

Il pacchetto minerva, che abbiamo introdotto durante il convegno GUIT 2012, consente di creare prove (esercitazioni o compiti) da somministrare agli studenti al PC.

Ricordiamo brevemente le principali caratteristiche illustrate in quell'occasione:

- le prove vengono create a partire da un database di esercizi, che i docenti hanno a disposizione e possono integrare;
- si deve compilare un file master, dal quale vengono generati tutti gli output necessari per l'esecuzione della prova (file interattivo, file cartaceo, stringa delle soluzioni);
- sono definiti vari comandi matematici di semplificazione delle frazioni, degli esponenti, ecc. (già condivisi con il pacchetto esami);
- è possibile inserire parametri casuali negli esercizi, in modo da generare dallo stesso file più prove "diverse";
- sono definite le opzioni compito ed esercitazione, che permettono di scegliere se si desidera costruire una prova di verifica vera e propria oppure un'esercitazione, che può essere tracciata oppure da svolgere offline;
- è previsto un ambiente solution anche per i test a risposta multipla;
- è previsto l'ambiente MNdb, che ha un solo argomento obbligatorio, ovvero il nome di un file nel quale saranno scritti gli esercizi e all'interno del quale il comando `\selectrandomlyn` permette di selezionare n varianti da un file e di scriverle su un file esterno. La chiusura dell'ambiente fa scrivere, in ordine casuale o fissato, gli esercizi selezionati. Il codice

```
\begin{MNdb}{equazioni}
\selectrandomlyn{3}{equadue}
\selectrandomlyn{6}{disequadue}
\end{MNdb}
```

  sceglie casualmente 3 esercizi dal file `equadue`, 6 dal file `disequadue` e li manda in output. I meccanismi di scelta casuale e riordinamento da una lista sono illustrati in dettaglio nel paragrafo 3.3.

2 acrotex

Il pacchetto minerva è costruito "sopra" i pacchetti di acrotex di D.P. Story.

La nostra idea è stata quella di estendere alcune funzionalità dei pacchetti preesistenti evitando il più possibile di rinunciare a quelle esistenti e automatizzare alcune procedure.

Con il nome acrotex ci si riferisce ad un gruppo di pacchetti, elaborati di D.P. Story per la generazione di esercizi ed esami. Uno degli aspetti caratteristici di acrotex è quello di sfruttare appieno alcune potenzialità dei file PDF che vengono generati usando, quasi sempre, solo le funzionalità di LaTeX[1]

Le funzionalità più interessanti sono

1. la possibilità di rendere i file interattivi mediante l'inserimento di codice JAVASCRIPT (ADOBE SOLUTIONS NETWORK, 2001);
2. la possibilità di comunicare dati ad un server utilizzando un file di dimensioni contenute (FDF)[2] (ADOBE SOLUTIONS NETWORK, 1999).

I pacchetti principali di acrotex sono

1. exerquiz per la generazione di esercizi interattivi (Story, 2012);

*Il lavoro si inserisce nel progetto di ricerca di interesse d'Ateneo dell'Università Cattolica "Progetto M.In.E.R.Va.: Creazione di esercizi di tipo interattivo con percorsi differenziati per il recupero delle carenze e la valorizzazione delle eccellenze in matematica".

1. Per alcuni aspetti, che però non ci interessano, è invece necessario processare il file con Adobe Professional.
2. Attualmente Adobe spinge per l'utilizzo di altri tipi di file, ottenibili con un software a pagamento, ma poiché la possibilità di creare file FDF fa parte del protocollo PDF, dovrà sempre essere implementata.

2. web che gestisce gli aspetti "estetici" (pannelli, titoli, ecc.);
3. eqexam per la generazione di esami;
4. insdljs e dljslib per l'inserimento del codice JAVASCRIPT in documenti LaTeX;
5. eforms per la gestione dei *forms* PDF direttamente in LaTeX;
6. eq2db per la comunicazione PDF-server e la memorizzazione degli esiti sul server;
7. altri pacchetti minori.

Per i nostri scopi, il pacchetto più interessante è exerquiz, che definisce tre ambienti per gli esercizi:

1. problem o exercise che servono per esercizi da svolgere "su carta" perché prevedono una risoluzione estesa;
2. shortquiz che permette di scrivere esercizi a risposta aperta o chiusa, che non prevedono però la comunicazione con il server e l'attribuzione di un punteggio;
3. quiz che funziona come shortquiz, con la differenza che gli esercizi sono valutati e i dati possono essere registrati su un server.

Gli ambienti shortquiz e quiz possono contenere le stesse tipologie di esercizi:

- domande a scelta multipla (MCQ) con una sola risposta esatta;
- domande a risposta multipla (MAQ) in cui ci possono essere più risposte esatte;
- domande a risposta aperta, in cui la risposta, che deve essere scritta rispettando alcune regole, viene esaminata e valutata automaticamente utilizzando codice JAVASCRIPT.

Tutte le domande possono avere punteggi diversi e nelle domande a risposta chiusa ogni risposta può avere un punteggio diverso, anche negativo. Il codice

```
\item
 Nella frase ``Paolo legge un libro'',
 \emph{un libro} è
\begin{answers}{numero colonne}
    \bChoices[random]
\Ans[1]{1} complemento oggetto \eAns
\Ans[0]{0} complemento \eAns
\Ans[-1]{0} soggetto \eAns
      \eChoices
\end{answers}
```

produce in output

Nella frase "Paolo legge un libro", *un libro* è

1. complemento
2. complemento oggetto
3. soggetto

I numeri indicati tra parentesi quadra sono opzionali ed indicano il punteggio attribuito ad ogni risposta.

Il pacchetto prevede di fornire all'utente diversi feedback:

- nell'ambiente shortquiz viene segnalato se la risposta è giusta o sbagliata e si può inserire la soluzione. Inoltre, per le domande a risposta aperta, è possibile, mediante il comando `\tallybox`, inserire una casella accanto a quella della risposta, nella quale vengono contate le risposte sbagliate. Tutti i messaggi di feedback vengono dati immediatamente, appena si inserisce la risposta.
- Nell'ambiente quiz si ha la segnalazione della risposta giusta o sbagliata, la soluzione e il punteggio. Tutte i feedback vengono dati alla fine dell'esercitazione.

Utilizzando eq2db e l'ambiente quiz, alcuni dati possono essere inviati ad un server: i dati anagrafici, le risposte date, il punteggio, la data e l'ora di inizio e fine della prova.

acrotex è stato usato in vari progetti per la somministrazione di esercizi online, come Acroweb (Maříк), MacQTeX (Griffin), ecc..

3 Il nostro lavoro

Il progetto è rivolto a studenti delle scuole medie superiori e lo scopo principale è quello di mettere a disposizione dei percorsi di recupero delle carenze o di consolidamento delle conoscenze che lo studente può fruire in autonomia o sotto la guida del docente.

Nel citato meeting abbiamo illustrato le modalità di realizzazione della parte teorica e delle esercitazioni guidate relative a questo percorso (Messineo e Vassallo, 2012b).

Quest'anno, il nostro lavoro si è concentrato sulla modifica e integrazione di quanto già previsto in acrotex per la realizzazione delle esercitazioni.

3.1 Gli output

Il pacchetto prevede la realizzazione di diverse tipologie di output: l'esercitazione non tracciata, l'esercitazione tracciata e il compito.

3.1.1 L'esercitazione non tracciata

Si tratta di un insieme di esercizi che vengono assegnati allo studente da svolgere autonomamente, a casa o a scuola, e che non prevedono il tracciamento delle risposte come le altre due modalità.

Funzionano come le esercitazioni preparatorie agli esami ECDL. Possono essere assegnate agli

studenti in vario modo (invio per mail, su cd, ecc.). L'idea è quella di rendere disponibili agli studenti set di esercizi, con correzione automatica, che contengano le soluzioni, i richiami alla teoria e dei suggerimenti per arrivare alla risposta esatta.

Questa tipologia di output viene creata caricando il pacchetto minerva con le opzioni esercitazione e offline.

L'ambiente ideale da usare in questo tipo di esercitazione è shortquiz, presentato nel paragrafo 2, poiché è possibile somministrare domande sia a risposta aperta che chiusa, la correzione delle risposte avviene immediatamente, è possibile inserire la soluzione completa[3], si può usare, per le domande a risposta aperta, il comando `\tallybox` ed è possibile fornire dei suggerimenti (per una descrizione dettagliata, si vedano i paragrafi 2 e 3.2).

3.1.2 L'esercitazione tracciata

In questo caso, il docente somministra agli studenti un'esercitazione, ma vuole registrare sul server le risposte date dallo studente e il punteggio ottenuto.

Viene creata caricando il pacchetto minerva con le opzioni esercitazione e online, che illustreremo nel paragrafo 3.2.

In questo caso, l'ambiente migliore da usare è quiz, che, a differenza di shortquiz, prevede sia la possibilità di tracciare le risposte sia quella di avere la correzione del lavoro fatto solo alla fine dell'esercitazione e non appena data la risposta come nell'altro caso.

Anche in questo tipo di esercitazione si possono usare domande a risposta aperta o chiusa, dare suggerimenti e anche la soluzione completa.

Questa tipologia di esercitazione potrebbe essere usata anche in autonomia dagli studenti, per svolgere esercizi di recupero delle carenze o di consolidamento delle conoscenze. A tale scopo, è stato creato un eserciziario online che illustreremo nel paragrafo 3.4.

3.1.3 Il compito

In questo caso, il docente vuole preparare una prova di valutazione da somministrare ad uno o più studenti dello stesso gruppo.

Questo tipo di output viene creato caricando il pacchetto minerva con l'opzione compito, che prevede in automatico l'opzione online.

Questa opzione disabilita la correzione automatica delle risposte, la soluzione e gli eventuali suggerimenti (trattandosi di un compito, nessuno di questi elementi è utile).

Tutti i dati di questa tipologia di prova vengono registrati su un server.

3.2 I nuovi comandi

Il nostro lavoro si è concentrato su estensioni delle funzionalità di exerquiz e di eq2db e della gestione dei dati sul server. Ci siamo basati su quanto già fatto nei pacchetti esami, distribuito su CTAN (Messineo e Vassallo, 2012a), esami-online e minerva, privati (Messineo e Vassallo, 2012b).

Il pacchetto è stato pensato per esercizi di Matematica, ma può essere utilizzato facilmente anche per altre discipline.

Il nostro obiettivo era quello di estendere le caratteristiche presentate, introducendo

1. integrazione nel pacchetto di tutte le funzionalità del pacchetto acrotex;
2. inserimento di parametri e di altri elementi aleatori negli esercizi;
3. possibilità di selezione degli esercizi da un database;
4. possibilità di creazione di prove "personalizzate" (prove diverse per ciascuno studente o prove ripetute sempre diverse per lo stesso studente) in modo automatizzato usando gli elementi aleatori[4];
5. generazione di una copia cartacea delle prove (sia per backup, sia per poter somministrare la prova agli studenti in caso di indisponibilità tecnica della piattaforma);
6. automazione di alcuni meccanismi a seconda della modalità scelta (esercitazione tracciata online, esercitazione offline, compito o esame);
7. nuove funzionalità (suggerimenti);
8. comunicazione al server di un numero maggiore di dati;
9. tracciamento completo degli esercizi assegnati e degli esiti per finalità statistiche e di controllo.

Gli obiettivi citati sono stati perlopiù raggiunti utilizzando solo codice LaTeX. Non così gli obiettivi riguardanti la comunicazione con il server. Gli aspetti relativi al server sono ancora in lavorazione.

In particolare

1. sono state integrate nel pacchetto minerva alcune funzionalità di acrotex non previste nella prima versione. In particolare, abbiamo previsto

3. L'inserimento della soluzione era già previsto nel pacchetto exerquiz, ma nel nostro ci sono stati dei problemi perché il testo dell'esercizio e della sua soluzione è contenuto nel comando `\newproblem`, che, a nostra conoscenza, non poteva contenere `verbatim` come argomento. Abbiamo risolto questo problema mediante l'uso del pacchetto cprotect.

4. È ovviamente possibile anche creare una prova unica da somministrare a tutti gli studenti di una classe o di un gruppo

- la possibilità di inserire, gestire e personalizzare il pannello laterale con i comandi di navigazione;
- la possibilità di inserire domande a risposta aperta o a risposta chiusa con più di una risposta esatta[5];
- la possibilità di assegnare ad ogni risposta (per le domande a risposta chiusa) un punteggio diverso, anche negativo;
- il comando `\tallybox`, per mostrare allo studente il numero di risposte errate dato alla domanda.

Queste nuove introduzioni hanno richiesto, oltre ad una parziale modifica del codice per renderlo compatibile con quanto già inserito fino a quel momento, un attento lavoro di controllo della compatibilità e di gestione di eventuali problemi. Infatti, il nostro meccanismo di rotazione e il fatto che si voglia produrre sempre ed in automatico un file cartaceo hanno spesso generato dei problemi (di impaginazione, ecc.) che di volta in volta abbiamo dovuto risolvere.

2. le modalità di inserimento di parametri ed altri elementi aleatori è già stata ampiamente descritta in (Messineo e Vassallo, 2012a) e in (Messineo e Vassallo, 2013a);

3. la scelta di esercizi da un database costruito autonomamente è una delle caratteristiche del pacchetto probsoln di Nicola Talbot (Talbot, 2011). Il nostro codice si è ispirato molto a quello, in larga parte modificandolo, perché ci erano necessarie funzionalità diverse.

   Abbiamo già illustrato queste modifiche in (Messineo e Vassallo, 2012a) e in (Messineo e Vassallo, 2012b), riprendiamo qui alcuni punti.

   Gli esercizi possono essere contenuti anche in più file e ogni esercizio è l'unico argomento del comando `\newproblem` che compone gli esercizi. La sintassi per la scrittura degli esercizi è identica a quella usata in acrotex, mentre per l'utilizzo dei parametri aleatori e le operazioni con i parametri si rimanda a (Messineo e Vassallo, 2012a). Nell'idea originaria, ogni file è composto da più varianti di esercizi sullo stesso argomento, ma ora ciò non è più necessario. All'interno del file che viene compilato ci sono alcuni comandi che permettono di estrarre dai file degli esercizi le varianti desiderate. I principali comandi, alcuni dei quali sono stati presentati in (Messineo e Vassallo, 2012a) e in (Messineo e Vassallo, 2012b), sono

   (a) `\esercizi{file1,file2...}` che estrae una variante da ogni file, mescola gli esercizi scelti e li manda in output;

   (b) `\estraies[m]{file1,file2,...}` che sceglie aleatoriamente $n - m$ file tra gli n che ha come argomento, estrae una variante da ognuno di essi e le manda in output;

   (c) MNdb, un ambiente introdotto per estrarre m esercizi degli n presenti in un file. Ha come argomento il nome di un "database" di esercizi. In questo database gli esercizi vengono inseriti con il comando `\selectrandomlyn{m}{file}`, che sceglie casualmente m varianti di esercizi da un file. Tale comando può essere usato più volte (applicato a file diversi) all'interno dell'ambiente. La chiusura dell'ambiente manda in output gli esercizi.

   I tre comandi di scelta e l'ambiente MNdb si basano su una versione modificata dei dataset e dei comandi di scelta del pacchetto probsoln. Il meccanismo di scelta e di rimescolamento degli esercizi è stato implementato da noi.

   Questi comandi possono essere usati più volte all'interno di una prova e devono essere inseriti all'interno degli ambienti shortquiz o quiz.

4. La possibilità di avere prove "personalizzate" è stata implementata con un meccanismo già usato in esami-online (Messineo e Vassallo, 2012b). Gli elementi aleatori dipendono da due quantità: la data della prova e un identificativo dello studente (la matricola o un codice assegnato).

   Per la creazione delle prove compiliamo un file che usa il pacchetto datatool di Nicola Talbot (Talbot, 2013) per la lettura e la manipolazione dei file `csv`. Viene letto l'elenco dei partecipanti alla prova, per ogni studente viene creato un file con i dati, viene rinominato il file della prova e viene lanciata la compilazione, che avviene in più passi: si compila prima la versione cartacea, poi quella online e infine il file delle soluzioni. È possibile che questo procedimento possa essere reso più veloce utilizzando un file di formato precompilato, ma tale possibilità deve essere ancora esplorata. Se lo stesso file con gli stessi esercizi viene compilato per lo stesso studente in date diverse, si ottengono prove diverse.

   Se la prova che si vuole compilare è invece unica per tutti gli studenti o per un gruppo, il file che viene compilato non usa datatool e crea un file unico, con l'anagrafica non compilata (come invece avviene per le prove individuali).

5. L'inserimento di questo tipo di domande comporta alcuni problemi di formattazione dei dati registrati sul server, dei quali parleremo nel paragrafo 5

5. La copia cartacea della prova si è resa necessaria per poter somministrare il compito allo studente anche in caso di malfunzionamento del server. Questa esigenza ha creato alcuni problemi: infatti, anche se acrotex prevede la possibilità di creare un output cartaceo, il nostro meccanismo di rotazione degli esercizi, di scelta da un file e la volontà di avere una stringa delle soluzioni ci ha portato a modificare alcuni comandi di acrotex, per cui le funzionalità del pacchetto hanno dovuto essere tutte verificate nuovamente.

   Ad esempio, in acrotex c'è l'opzione proofing che inserisce nelle domande a risposta chiusa un simbolo accanto alla risposta esatta. Noi abbiamo legato a tale simbolo un meccanismo di `\label` e `\ref` che riesce a tenere traccia degli esercizi scelti nei file, della risposta corretta e a trascrivere questi dati nel file delle soluzioni e nel file aux. Questo comporta che l'opzione sia sempre attiva e quindi, ad esempio, che nell'ambiente shortquiz ci sia sempre il feedback immediato.

6. l'automazione dei meccanismi a seconda della modalità scelta è stata ottenuta introducendo alcune opzioni per la creazione automatica di diversi elementi. Con l'opzione esercitazione viene prevista la possibilità di avere la correzione degli esercizi, di vedere le soluzioni ed eventualmente di avere dei suggerimenti. Sono previste inoltre due ulteriori opzioni, online e offline, che permettono di salvare o non salvare i dati sul server. Inoltre, in modalità offline è possibile usare l'ambiente shortquiz nel quale lo studente ha un feedback immediato alle risposte date.

   Con l'opzione compito vengono disabilitati tutti i suggerimenti e le correzioni e i dati vengono sempre trasmessi al server.

7. La possibilità di avere dei suggerimenti è l'unica variazione di acrotex visibile allo studente (oltre a quella di avere i campi anagrafici precompilati se l'esercitazione è "personalizzata"). La generazione dei suggerimenti è ottenuta mediante il pacchetto fancytooltips di R. Mařík (Mařík, 2012) che inserisce nel file PDF delle pagine di un file PDF esterno come icone nascoste rese visibili mediante il passaggio (o il click) del mouse. I suggerimenti vengono generati in un file esterno.

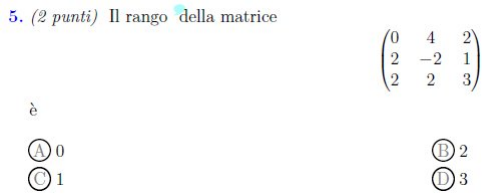
   Il codice per creare un file esterno di suggerimenti è

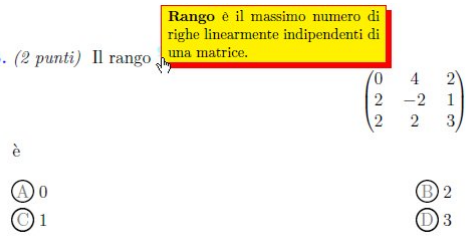
```
\documentclass{article}
\pagestyle{empty}
\usepackage[createtips]{fancytooltips}
\usepackage{fancybox}
\parindent 0 pt
\usepackage{xcolor}
\usepackage{multido,graphicx}
\usepackage[papersize={22cm,16cm},
margin=1pt]{geometry}
\long\def\suggerimento#1#2{
\setbox0=\hbox{\begin{minipage}{2in}
\fboxsep 0 pt
\color{red}
\shadowbox{
{\fboxsep 4pt\colorbox{yellow}
    {\begin{minipage}{\linewidth}
      \color{black}#2
    \end{minipage}}}}
\end{minipage}\ \ \ \ }
\pdfpagewidth=\wd0
\pdfpageheight=\ht0
\advance \pdfpageheight by \dp0
\copy0
\keytip{#1}
  \newpage
}
\begin{document}
\suggerimento{rango}{\textbf{Rango} è
il massimo numero di righe linearmente
indipendenti di una matrice.}
\end{document}
```

Il codice per richiamare i suggerimenti nel file principale è

```
\item\PTs{2} Il rango \tooltip*{ }{1}
della matrice
\[ \begin{pmatrix}0 & 4 & 2\\
2 &-2&1\\
2 & 2& 3\\
\end{pmatrix}
\]
è
\begin{answers}{2}
\bChoices
\Ans0 0 \eAns   \Ans1 2 \eAns
\Ans0 1 \eAns   \Ans0 3 \eAns
\eChoices
\end{answers}
```

Questo codice produce il seguente output





8. Per quanto riguarda le comunicazioni con il server, ci si è mossi in due direzioni: utilizzare (lato server) altri linguaggi oltre ad ASP, già implementato in acrotex, e ampliare la quantità di informazioni trasmesse.

   Il pacchetto acrotex prevede l'utilizzo di un server ASP e di librerie specifiche che la Adobe ha creato per Windows per la lettura dei file FDF. In realtà, il file FDF è un file di testo (con un'intestazione binaria) e può quindi essere letto in più modi (la Adobe stessa mette a disposizione le librerie PERL e JAVA). Per questo motivo, si sono realizzati degli script in linguaggio ASPX e PHP (si veda il paragrafo 3.4). Questo ha portato a dover modificare alcuni file del pacchetto eq2db in modo da ottenere che le comunicazioni con il server avvenissero in modo corretto.

9. Per quanto riguarda le informazioni che vengono inviate al server, abbiamo implementato con l'aiuto di D.P. Story due procedure in linguaggio ASP: la prima invia, oltre a tutte le altre informazioni, anche la risposta esatta alle domande proposte; la seconda memorizza, salvandolo in un file, il contenuto del file FDF[6]. Queste modifiche rispondono a due necessità: da un lato, poter salvare il file FDF permette, attraverso l'unione di tale file con il corrispondente PDF, di ricreare la prova come è stata svolta dallo studente[7] e ciò risponde ad ogni tipo di esigenza di conservazione delle prove di valutazione; dall'altro, questo è il primo passo verso l'obiettivo di raccolta e analisi dei dati per poter valutare l'efficacia degli esercizi proposti sia per il miglioramento degli studenti che per discriminare il loro livello di preparazione.

3.3 Una breve digressione sul riordinamento di liste e scelta casuale

Nel pacchetto minerva gli elementi casuali intervengono in molti punti (scelta degli esercizi, riordinamento delle domande, delle risposte, ecc.). Questi elementi casuali si basano su tre diversi metodi di riordinamento di una lista (e molti altri metodi sono presenti in altri pacchetti). Abbiamo deciso di mantenere tutti e tre i metodi per diversi motivi:

- una maggiore semplicità di sviluppo dei comandi per la scelta sulle diverse liste di oggetti in quanto questi fanno riferimento a quei metodi di scelta
- la possibilità di avere una maggiore variabilità nella generazione delle prove date le diverse procedure di scelta (se il metodo per la scelta degli esercizi porta ad avere lo stesso esercizio in due prove diverse, è estremamente probabile che l'ordine delle risposte sia diverso proprio per la differente procedura di scelta)
- la differenza di gestione di uno dei metodi rispetto agli altri due: infatti il metodo che viene usato (anche in acrotex) per la rotazione delle risposte nelle domande a risposta chiusa si basa su dei tag e permette di tenere degli elementi fissati.

Vorremmo qui analizzare brevemente questi tre metodi.

Il primo metodo è una nostra variazione di un codice trovato in rete. Il comando è definito usando il pacchetto xargs che permette di avere più parametri opzionali in un comando e di averli in qualunque posizione (nel nostro caso il parametro opzionale è il terzo).

```
\newcommandx{\rand@getitems}[3][3=item]
{% #1=numero totale elementi lista #2=
% nome file arrivo #3=nome file partenza
  \i@sh=#1% %\theshuf@tot
  \loop%
\expandafter\let
\csname flag\number\i@sh\endcsname a%
    \advance\i@sh by-1%
    \ifnum\i@sh > 0 \repeat%
  \i@sh=#1% %\theshuf@tot
  \loop%
    \setrannum{\j@sh}{1}{#1}%
\expandafter\ifx
\csname flag\number\j@sh\endcsname a%
\expandafter\let
\csname flag\number\j@sh\endcsname b%
 %Controllo se è già stato scelto
\FPeval\n@tmp{round(#1-\number\i@sh+1:0)}
 \expandafter\edef
 \csname #2\romannumeral\n@tmp\endcsname
 {\csname #3\romannumeral\j@sh\endcsname}
\advance\i@sh by-1%
    \fi%
    \ifnum\i@sh>0 \repeat
}
```

Il comando funziona in questo modo: con una procedura precedente è stato assegnato a ogni elemento di una lista un comando che è dato da un nome + un numero romano, di default \itemi, \itemii,

6. Un'analoga procedura è stata implementata anche in ASPX e in PHP.

7. Per ora tale unione è possibile solo manualmente usando Adobe Reader e non in via automatica con programmi come Pdftk o simili.

. . . ; la stessa procedura conta gli elementi della lista `\theshuf@tot`. Ora si sceglie casualmente un numero dell'elenco (e quindi l'elemento della lista che ha quel numero) e si assegna a una nuova variabile (ad esempio `\casoi`) l'elemento scelto e si prosegue così, controllando ad ogni passo se il numero scelto casualmente era già stato scelto (controllo sui flag).

Il secondo metodo è una nostra variazione del codice per mescolare gli elementi di una lista usato in probsoln

```
\newcommand{\shuffle}[3][]{%
\@shfctr=1\relax
\whiledo{\@shfctr < 101}%
{%
\setrannum{\@shfA}{1}{#3}
\setrannum{\@shfB}{1}{#3}
    \ifnum\@shfA=\@shfB
    \else
\edef\@@tmpA
{\csname#2\romannumeral\@shfA\endcsname}
      \let\@tmpA=\@@tmpA
\edef\@@tmpB
{\csname#2\romannumeral\@shfB\endcsname}
      \let\@tmpB=\@@tmpB
\expandafter\xdef
\csname#2\romannumeral\@shfA\endcsname
{\@tmpB}
\expandafter\xdef
\csname#2\romannumeral\@shfB\endcsname
{\@tmpA}
    \fi
    \advance\@shfctr by 1\relax
    \ifthenelse{\equal{#1}{}}{}{
  \@shfA=0%
  \loop
  \advance\@shfA by1
\expandafter\xdef
\csname#1\romannumeral\@shfA\endcsname
{\csname#2\romannumeral\@shfA\endcsname}
  \ifnum\@shfA<#3 \repeat
}
  }%
}
```

Questo comando funziona in pratica con lo stesso meccanismo del mescolamento delle carte: dati gli elementi di una lista a cui come prima è stato assegnato un "nome" e stabilito il numero di elementi della lista, si scelgono due numeri casuali nell'intervallo assegnato e, se diversi, si scambiano tra loro; la procedura viene eseguita per un certo numero di volte (nel nostro caso 100) e alla fine si ha una lista riordinata a cui vengono riassegnati gli elementi originali.

Il terzo metodo è quello usato nel pacchetto acrotex per la rotazione delle risposte nelle domande a risposta chiusa.

```
\def\aeb@randomizeChoices#1{%
    \setrannum{\aeb@ranChoice}{1}{#1}
\count0=0
\aeb@hold=\expandafter{\@temphold}
    \def\@temphold{}%
\expandafter\@tfor\expandafter
 \@temp\expandafter:
  \expandafter=\the\aeb@hold \do {%
        \advance\count0by1
  \ifnum\count0=\aeb@ranChoice
  \aeb@hold=\expandafter\expandafter
\expandafter
{\expandafter\@tempholdrandom\@temp}%
\edef\@tempholdrandom{\the\aeb@hold}%
        \else
\aeb@hold=\expandafter\expandafter
\expandafter
{\expandafter\@temphold
\expandafter{\@temp}}
\edef\@temphold{\the\aeb@hold}%
        \fi
    }%
    \aeb@numChoices=#1
 \advance\aeb@numChoices-1
\ifnum\aeb@numChoices=0\relax
\def\@next{\aeb@finishedRandomizing}
\else
\def\@next
{\aeb@randomizeChoices
{\the\aeb@numChoices}}
\fi    \@next
}
\def\aeb@finishedRandomizing{%
\aeb@hold=\expandafter\expandafter
\expandafter
{\expandafter\@tempholdrandom
\@tempholdfreeze}
\gdef\@temphold{}\gdef\@tempholdrandom{}
\gdef\@tempholdfreeze{}%
    \edef\finished@Randomizing{%
    \noexpand\@@ansChoices
    \the\aeb@hold
    \noexpand\eChoices}%
    \finished@Randomizing
}
```

Questa procedura è in un certo senso simile alla prima: si sceglie un numero casuale (e quindi il corrispondente elemento) e lo si mette in una lista "di prescelti"; dalla lista contenente i soli elementi restanti si estrae un nuovo numero casuale e quindi l'elemento corrispondente che viene spostato nella lista dei prescelti e così via. Il procedimento è complicato dal fatto che è possibile definire uno o più elementi della lista come inamovibili e quindi tali elementi devono prima essere eliminati e immessi in un'altra lista e, successivamente, reimmessi nella lista definitiva.

Sarebbe interessante verificare, magari considerando altre procedure, quali siano più efficienti sia in termini di effettiva redistribuzione casuale, sia

in termini di velocità di esecuzione.

3.4 I programmi di supporto

Come illustrato lo scorso anno, l'utilizzo dei materiali prodotti ha reso necessaria la creazione di una piattaforma web attualmente ospitata sul sito dell'IIS "Falcone-Righi" di Corsico (una delle scuole aderenti al progetto) e visibile all'indirizzo `http://minerva.falco.mi.it`. Il server web utilizzato è Microsoft IIS per poter utilizzare gli script in linguaggio ASP forniti con acrotex.

Come già accennato, una parte del progetto ha previsto la realizzazione di script in linguaggio diverso da ASP, sia per rendere la piattaforma indipendente dalle librerie della Adobe, sia per poter migrare la piattaforma su un server APACHE, in modo che l'intero progetto si serva di software Open Source.

Le due modifiche introdotte sono state

1. Un collaboratore della Direzione Sistemi Informativi dell'Università Cattolica ha implementato un parser nel linguaggio ASPX di Microsoft che funziona anch'esso su un server IIS. Tale script funziona senza l'utilizzo di librerie specifiche e salva i dati sia in un file `csv` che in un database. Inoltre, ha la possibilità di salvare il file FDF completo.

2. Uno studente dell'ITI "Lagrange" ha realizzato un parser in PHP, di cui si è parlato in (Messineo e Vassallo, 2013b), per interpretare i dati inviati mediante il file FDF. Questo script salva i dati in un file `csv`, in modo da poterli importare in un database consultabile, attraverso una pagina HTML, dai docenti o dall'amministratore della piattaforma. Lo script riconosce in automatico il tipo di file FDF e la tipologia di dati inviati e li registra di conseguenza. Anche questo script è in grado di salvare il file FDF completo.

I due parser elaborati non dipendono (come invece quello originale di acrotex) da librerie esterne e sono al momento "statici", ovvero è necessario definire all'interno dello script quali campi del file FDF devono essere salvati.

Lo studente dell'ITI "Lagrange" ha creato anche un *eserciziario online*: si tratta di una piattaforma (scritta anch'essa in linguaggio PHP) che consente ai docenti di caricare gli esercizi da loro prodotti, classificandoli secondo alcuni criteri prestabiliti, e di scegliere tra gli esercizi già caricati quelli necessari per la creazione della prova che desiderano somministrare.

In questo eserciziario, un docente può caricare esercizi nuovi. Tali esercizi devono essere scritti in LaTeX, seguendo le istruzioni già descritte in (Messineo e Vassallo, 2012a). Occorre caricare anche uno o due file PDF generati con il file di controllo, `totale-versioni`. Tali file contengono il testo di tutte le varianti degli esercizi in formato numerico e, se possibile, anche in formato parametrico. Si deve anche indicare il numero di varianti, il grado di difficoltà, la classe e la scuola a cui è rivolto.

È possibile scaricare gli esercizi e generare tutti i file necessari per la preparazione di una prova. Una funzione di preview permette di vedere gli esercizi scelti e di selezionarli; l'inserimento di alcuni dati permette la creazione dei file necessari per la compilazione. La piattaforma permette lo scaricamento di un file compresso, protetto da password, il contenuto dovrebbe permettere a chiunque abbia a disposizione i pacchetti necessari di compilare la prova, senza la necessità di intervenire sui file `tex`. Questo è il primo passo verso la possibilità di una compilazione online delle prove per tutti gli studenti da parte del docente e delle prove di recupero o esercitazione individuale da parte del singolo studente.

C'è infine da notare che lo studente ha fatto in modo che l'eserciziario sia visibile e, almeno in parte, utilizzabile con qualsiasi dispositivo, dallo smartphone al desktop.

4 La sperimentazione

Come accennato, il pacchetto è stato creato nell'ambito di un progetto di ricerca volto a sperimentare una nuova modalità di recupero delle carenze in Matematica e può essere usato anche per altre discipline.

Il pacchetto, le tipologie di esercizi e la piattaforma sono stati sperimentati in alcune classi dell'IIS "Falcone-Righi" di Corsico, sia mediante la somministrazione di esercitazioni tracciate ed altro materiale di supporto, sia mediante il coinvolgimento diretto degli studenti nella creazione degli esercizi.

La prima parte della sperimentazione è stata presentata lo scorso anno al convegno GUIT (si veda (Messineo e Vassallo, 2012b)). Alcuni risultati preliminari sono stati presentati al convegno Didamatica 2013 (Messineo e Vassallo, 2013b) e in due articoli successivi, (Messineo e Vassallo, 2013c) e (Messineo e Vassallo, 2013d).

5 Questioni aperte

Rimangono ancora alcuni problemi aperti e funzionalità da implementare.

Una funzionalità che vorremmo implementare sulla piattaforma web è la possibilità di modificare online i file con un editor LaTeX e, successivamente, di poter compilare su richiesta direttamente i file PDF sul server.

I problemi aperti riguardano soprattutto la gestione dei file FDF che vengono inviati al server.

Il primo problema è la gestione da parte del codice JAVASCRIPT degli elenchi di risposte nelle domande a risposta multipla: il file PDF manda

al server la lista delle risposte date come elenco separato da virgole; se tra le domande ve ne sono a risposta multipla, la stringa delle risposte a tale domanda è un altro elenco separato da virgole: la gestione di questo elenco all'interno dell'altro elenco non crea problemi per l'attribuzione dei punteggi e l'esito finale (gestito localmente dal codice JAVASCRIPT), ma determina un output scarsamente leggibile. Ad esempio l'output *a,a,b,c,b* riferito a 3 domande a risposta singola e una a risposta multipla può essere dato da *a,{a,b},c,b* o da *a,a,{b,c},b* e per distinguere i due casi è necessario confrontare la stringa con l'elenco delle domande (la situazione si complica se non vengono date delle risposte).

Il secondo problema è relativo all'archiviazione dei dati: come si è detto la coppia "file PDF" (testo dell'esame) e "file FDF" (risposte date) può essere unita in un unico file PDF che contiene così la prova completa dello studente. Questa procedura può essere automatizzata anche su più file e può essere eseguita sul server tramite il software PDFTK: purtroppo al momento questo non è possibile per qualche problema di compatibilità del file FDF, e la procedura può essere eseguita solo utilizzando Adobe Reader su ogni singolo file: si tratta quindi di vedere quale sia il problema che impedisce l'uso della procedura automatizzata.

Il terzo problema riguarda il tracciamento dei dati a fini statistici: la nostra idea è quella di poter utilizzare il database in cui vengono salvati gli esiti a fini statistici per analizzare sia i progressi degli studenti su ogni singolo argomento, sia per verificare se le domande dei test siano o meno efficaci a discriminare il livello degli studenti. Attualmente noi salviamo solo gli esiti di ogni singola prova per ogni studente, quindi possiamo analizzare i progressi globali dello studente (o della classe). Alcuni di questi dati sono salvati nel file `aux` proprio per il meccanismo di `\label` e `\ref` adottato e da lì riportati nel file `tex` delle soluzioni

```
\item
\ref{a-e:20130507-file:equadue-NPE-q:xix}
\item
\ref{a-e:20130507-file:equadue-NPE-q:vii}
item
\ref{a-e:20130507-file:prova-aperto-q:i}
```

dove *a* sta per *answer*, *e:20130507* indica l'ID dello studente, *file:equadue-NPE* indica il file degli esercizi da cui si estrae la domanda, *q:xix* indica il numero della domanda nel file. Si tratta quindi di riuscire a riportare questi dati sul server (se possibile con il file FDF) e immetterli nel database in modo da poter sapere a quale domanda lo studente XY ha risposto esattamente, quanti studenti hanno risposto a una certa domanda ecc. Lo scopo finale è quello di riuscire a realizzare un'analisi dettagliata dell'efficacia della domanda nel testare le abilità degli studenti, usando una metodologia, analoga a quella che viene utilizzata per l'analisi delle risposte delle prove INVALSI, basata sul modello di Rasch (Bond e Fox, 2007).

6 Conclusioni

Il progetto M.In.E.R.Va., di cui il pacchetto minerva fa parte, pur essendo ancora in fase sperimentale, si è dimostrato un ausilio molto efficace per la didattica della Matematica nelle classi in cui è stato usato.

L'obiettivo finale è quello di realizzare una piattaforma che possa essere usata con facilità sia dai docenti con nessuna conoscenza di LaTeX (che quindi fruiscono dei percorsi già predisposti o scelgono gli esercizi dal database e devono solo compilare e caricare la prova), sia da docenti "esperti" (che possono anche inserire nuovo materiale o personalizzare quello esistente), sia da studenti, che possono scegliere in autonomia i percorsi o gli esercizi da eseguire. Inoltre, lo strumento deve mettere a disposizione di tutti gli utenti alcune statistiche significative.

Il nostro auspicio è quello di avere a disposizione una piattaforma sufficientemente robusta da consentire anche la compilazione "al volo" delle prove, sul modello, ad esempio, di MacQTeX (Griffin).

I file necessari per l'utilizzo del pacchetto e le comunicazioni con il server, con una minima documentazione, sono al momento disponibili al link `https://www.dropbox.com/sh/jf4miydsrd4ymub/LqqutgJk5S`, in attesa di una versione definitiva che sarà rilasciata su CTAN.

I file necessari per la comunicazione con il server sono disponibili in ASP (estratti dal pacchetto acrotex con alcune nostre modifiche), in ASPX (opera di Bruno Cibelli, che ha collaborato con noi alla realizzazione di parte del progetto), in PHP (opera di Mohamed Abedalla, studente dell'ITI "Lagrange" di Milano), non completamente testato.

La piattaforma usata presso l'IIS "Falcone-Righi" è visibile all'indirizzo `http://minerva.falco.mi.it`. È possibile visualizzare un percorso demo o chiedere la registrazione come docenti di prova per sperimentarne l'uso.

Riferimenti bibliografici

ADOBE SOLUTIONS NETWORK (1999). *FDF Toolkit Overview Technical Note # 5194*. ADOBE SYSTEMS INC.

— (2001). *Acrobat Forms JavaScript Object Specification, Version 5.0.5 Technical Note # 5186*. ADOBE SYSTEMS INC.

Bond, T. G. e Fox, C. M. (2007). *Applying the Rasch Model: Fundamental Measurement in the Human Sciences*. Psychology Pr.

Griffin, F. «MacQTeX». URL `http://rutherglen.ics.mq.edu.au/~macqtex/`.

Maříк, R. «Acroweb». URL http://user.mendelu.cz/marik/index.php?item=41.

Maříк, R. (2012). «Il pacchetto fancytooltips». CTAN:/macros/latex/contrib/fancytooltips/.

Messineo, G. e Vassallo, S. (2012a). «Il pacchetto esami per la creazione di prove scritte». ArsTEXnica, **14**, pp. 95–103.

— (2012b). «Test online di matematica: il progetto M.In.E.R.Va». ArsTEXnica, **14**, pp. 104–112.

— (2013a). «The esami package for examinations». *TUGboat*, **34** (1), pp. 40–46.

— (2013b). «Il progetto M.In.E.R.Va». In *Atti del Convegno Didamatica 2013*. pp. 689 – 698.

— (2013c). «Il progetto M.In.E.R.Va». *Nuova secondaria*, pp. 93–95.

— (2013d). «Il progetto M.In.E.R.Va per il recupero e la valutazione». *Bricks*. URL http://bricks.maieutiche.economia.unitn.it/?cat=239.

Story, D. P. (2012). «Exerquiz & AcroTEX». URL http://www.acrotex.net/.

Talbot, N. L. (2011). «Il pacchetto probsoln». CTAN:macros/latex/contrib/probsoln.

— (2013). «Il pacchetto datatool». CTAN:macros/latex/contrib/datatool.

▷ Grazia Messineo
Università Cattolica Milano – IIS "Falcone-Righi" Corsico
grazia dot messineo at unicatt dot it

▷ Salvatore Vassallo
Università Cattolica Milano
salvatore dot vassallo at unicatt dot it