

UDK 004.5:371.31:004.85

# ONLINE PROVJERAVANJE PROGRAMSKIH ZADATAKA UNUTAR SUSTAVA ZA UČENJE AHyCo ONLINE ASSESMENT OF PROGRAMMING ASSIGNMENTS IN A LEARNING MANAGMENT SYSTEM AHyCo

Nataša HOIĆ-BOŽIĆ – Ivan BUDIŠČAK – Ivica BOTIČKI

**Sažetak:** Automatizirane online provjere znanja najčešće su korištene vrste provjeravanja znanja kod sustava za e-obrazovanje, posebno važne kada je riječ o velikim grupama studenata kod kojih nastavnici provode dosta vremena ispravljajući zadaće i ispite. Kako je kod studenata računarstva i informatike važno da se osposobe za praktično znanje programiranja, potrebna je autorska okolina za online testiranje programskih zadataka. U ovome se radu opisuje sustav za automatsko vrednovanje programskih zadataka implementiran u okviru vlastitog sustava za učenje AHyCo (Adaptive Hypermedia Courseware). Prezentirana je arhitektura sustava te novi pristup kreiranju, dostavljanju i evaluaciji tih pitanja.

**Ključne riječi:** – online provjere znanja  
– sustav za učenje AHyCo  
– programski zadaci

**Abstract:** This paper considers automatic online tests that represent the most frequently used knowledge evaluation method in e learning, especially with large groups of students for which teachers spend much time correcting homework and tests. Since it is of the great importance that students of computing and informatics get some practical knowledge in programming, an authoring environment for online tests is required. This paper describes a system for automatic evaluation of programming assignments implemented in the AHyCo system (Adaptive Hypermedia Courseware). System architecture as well as a new approach towards creating, delivering and evaluating programming assignments is presented.

**Keywords:** – online testing  
– learning menagment system AHyCo  
– programming assignments

## 1. UVOD

Mnoge visokoškolske institucije danas mijenjaju obrazovni proces uvođenjem e-obrazovanja, tj. korištenjem informacijsko-komunikacijske tehnologije upotpunjene odgovarajućim pedagoškim metodama, oblicima i principima poučavanja [1].

Jedna je od najvažnijih karakteristika sustava za e-obrazovanje mogućnost ocjenjivanja i provjeravanja znanja studenata. *Online* kvizovi i testovi među najšire su korištenim tipovima za procjenjivanje znanja i imaju dvije osnovne uloge: ocijeniti napredak studenata te im pomoći pri učenju. Testovi za samoprovjeravanje znanja ili kvizovi važan su element interaktivnosti za studente [2]. Nastavnik može koristiti testove da ustanovi koje su komponente tečaja studenti razumjeli. Može koristiti spremljene rezultate testova kako bi ustanovio koliko su studenti dobri u usvajanju znanja te za ocjenjivanje studenata. Mogu se spremati i analizirati pogrešni odgovori te odrediti priroda pogrešnog shvaćanja.

## 1. INTRODUCTION

Today many higher education institutions transform their educational processes through e-learning or the use of information and communication technology (ICT) complemented with the appropriate pedagogical methods, forms and principles of teaching [1].

One of the most important characteristics of e-learning systems is their ability to evaluate or assess the knowledge of a student. Online quizzes and assessments are among the most widely used types of evaluation with two main purposes: they evaluate students' progress and help students in learning. The assessments for self-evaluation or quizzes are an important interactive element for students [2]. Teachers can use assessments to find out whether components of a course have been clearly understood by students. Assessment results can be stored and analyzed to determine the level of student performance and for grading. Wrong answers can be used to determine the cause of errors.

Uvođenje automatiziranih *online* provjera posebno je važno kada je riječ o velikim grupama studenata, što je čest slučaj na mnogim fakultetima u Hrvatskoj. Upravljanje velikim brojem studenata uvijek je predstavljalo teškoću u odnosu na okoline za učenje koje su više orijentirane na studente kao pojedince. Zbog nepovoljnog omjera nastavnika i studenata, čak i kod znatno manjih grupa studenata, koje su uvedene reformom studija u Hrvatskoj prema Bolonjskom procesu, nastavnici troše znatnu količinu vremena na ocjenjivanje studentskih zadataka umjesto da se posvete poboljšanju kvalitete nastave [3].

Za studente računarstva i informatike važno je da se osposobe za programiranje [4-6] te za rad s bazama podataka. U tom je smislu potrebna autorska okolina za *online* testiranje programa i SQL upita nad bazama. Glavni cilj predmeta o programiranju treba biti ne samo učenje sintakse programskih jezika nego, kao još važnije, učenje načina primjene novog znanja za rješavanje problema [7]. Iako se to može ostvariti jedino putem praktičnih vježbi iz programiranja, takve praktične zadatke teško je pravodobno ručno ocijeniti [8]. Prednosti automatskog ocjenjivanja programskih zadataka su: brzina, lakša dostupnost, dosljednost u ocjenjivanju i trenutna povratna informacija studentima [2].

Međutim najvrednija je prednost za studente korištenje okoline poput one u kojoj se zaista i programira umjesto metode temeljene na papiru i olovci [9]. Nekada je većina studenata bila suviše lijena da bi se koristila okolinom za programiranje za pisanje i ispravljanje kôda jer su pretpostavljali da neke manje pogreške nastavnik neće uočiti ili će ih ignorirati [8]. Uvođenje automatskog provjeravanja programskih zadataka može nastavniku omogućiti da se više posveti izravnoj komunikaciji sa studentima te da aktivnosti koje treba često ponavljati i podložne su pogreškama prepusti računalu [3].

U ovome radu opisuje se sustav za *online* ocjenjivanje znanja koji je sastavni dio sustava za upravljanje učenjem AHyCo (*Adaptive Hypermedia Courseware*). Vlastiti web-sustav za upravljanje učenjem AHyCo razvijen je na Fakultetu za elektrotehniku i računarstvo Sveučilišta u Zagrebu u suradnji s Odjelom za informatiku Sveučilišta u Rijeci. Teorijska osnova sustava AHyCo i implementacija prilagodljive navigacije opisani su u [10,11]. Kao i ostali standardni *online* sustavi za učenje, AHyCo ima nekoliko grupa alata koje koriste studenti i/ili nastavnici. Najvažniji su alati za izradu sadržaja za učenje te njegovo dostavljanje studentima. Ostale skupine alata služe za provjeravanje znanja studenata, podršku suradničkim aktivnostima i upravljanje grupama studenata te upravljanje podacima o studentima.

Sustav AHyCo razvio se od jednostavnog alata za provjeravanje znanja putem pitanja višestrukog izbora do kompleksnog sustava za automatsko ocjenjivanje programskih zadataka. Ovaj rad predstavlja osnovne aspekte procesa kreiranja, dostavljanja i ocjenjivanja pitanja za programske zadatke.

The introduction of automatic online assessments is especially important when large groups of students, as it is the case in Croatian universities, are concerned.

Dealing with a large number of students has always presented an obstacle towards a more student-oriented learning environment. Due to an inappropriate teacher-to-student ratio, even with significantly smaller groups introduced by the Bologna process reform in Croatia, teachers spend a significant amount of time on grading students' assignments instead of working on the quality of the teaching and learning process [3].

It is especially important for students of computing and informatics to master programming [4-6] and databases. Therefore, there is a need for an authoring component for online programming assignments and online SQL assignments. The main objective of programming courses should not be learning programming syntax, but, more importantly, learning the way to apply new knowledge to solving problems [7]. Although that can be obtained only through practical programming exercises, programming assignments are difficult to manually grade in a timely manner [8]. Therefore, the advantages of an automatic evaluation of programming assignments are speed, availability, grading consistency and instant feedback to students [2].

Nevertheless, the most valuable advantage for the student is the use of a programming environment instead of paper-and-pencil-based methods [9]. In the past, most students were too lazy to use a programming environment to write and debug code, because they assumed some minor mistakes would not be perceived or would be intentionally ignored by the teacher [8]. Introducing automatic evaluation of programming assignments would enable teachers to focus more on direct communication with students and to leave repetitive and error-prone activities to computers [3].

This paper describes a system for online knowledge assessment as a component of a learning management system (LMS) AHyCo. A proprietary Web-based LMS AHyCo (*Adaptive Hypermedia Courseware*) was developed at the Faculty of Electrical Engineering and Computing (FER), University of Zagreb in collaboration with Department of Information Science, University of Rijeka. The theoretical background of AHyCo and implementation of adaptive navigation are described in [10,11]. As with other standard LMS, AHyCo has several groups of tools that are used by students and/or instructors. The most important group of tools is the one for learning content authoring and delivering. The other sets of tools are for student assessment, collaboration support for student groups and class management.

The AHyCo system evolved from a simple multiple choice assessment tool to a more complex assessment system for automatic evaluation of programming assignments. This paper presents the main aspects of the process of creation, delivery, and evaluation of questions for programming assignments.

## 2. PITANJA I PROVJERE U AHyCo-u

Nastavni sadržaji u AHyCo-u podijeljeni su na lekcije ili cjeline koje su grupirane u module. Više modula čini predmet za učenje. Provjeravanje znanja usvojenog kroz modul za učenje sustava AHyCo provodi se putem različitih vrsta pitanja i provjera. Provjerama se utvrđuje razumijevanje sadržaja cjelina pomoću niza pitanja na koja student odgovara [10].

Za provjeru se navode cjeline čija će pripadajuća pitanja provjera sadržavati, ukupan broj pitanja u provjeri te skup konfiguracijskih pravila koja određuju koliko će najmanje, a koliko najviše pitanja iz svake cjeline biti uključeno u provjeru. Tako nije potrebno navesti konkretna pitanja provjere jer će se prilikom svakoga novog pisanja odabirati novi skup pitanja koja će je činiti. Pored toga se i odgovori na pitanja permutiraju, čime se otežava prepisivanje, čak i kada se odgovara na dva ista pitanja.

Pitanje u provjeri može biti:

- 1) Klasično pitanje tipa višestrukog odabira s jednim ili više točnih odgovora. Razlikuju se obična hipermedijska pitanja i parametarski zadana pitanja [1].
- 2) SQL pitanje čiji je odgovor SQL upit. Ocjenjivanje se provodi automatski putem modula za automatsko ocjenjivanje SQL zadataka.
- 3) Programsko pitanje čiji je odgovor program pisan u programskom jeziku C ili C++. Ocjenjivanje se provodi automatski putem modula za automatsko ocjenjivanje programskih zadataka [12].

AHyCo-ov sustav za automatsko ocjenjivanje programskih zadataka podržava nekoliko programskih jezika (C, C++, C#, SQL). Moguće ga je proširiti dodavanjem novih jezika (uvođenjem vanjskih kompajlera). Student piše i učitava u računalo izvorni kôd računalnog programa koji zatim sustav ocjenjuje automatski tako što ga pokrene i uspoređuje izlaz izvršenog programa s unaprijed definiranim testnim slučajevima.

U nastavku rada objasniti će se proces izrade, prezentiranja i ocjenjivanja programskih pitanja i testova.

## 3. ARHITEKTURA SUSTAVA

Sustav za provjeravanje znanja sastavni je dio sustava za učenje AHyCo. AHyCo je web-aplikacija kojoj se pristupa uporabom web-preglednika preko interneta ili intraneta i implementiran je u ASP.NET tehnologiji. Sustav se sastoji od web-poslužitelja, poslužitelja baze podataka te klijentskih računala putem kojih korisnici pristupaju web-aplikaciji AHyCo-a, kako je prikazano na slici 1.

## 2. QUESTIONS AND ASSESSMENTS IN AHyCo

Courseware in AHyCo is structured in lessons (concepts) grouped in modules. Modules can be grouped in courses. Knowledge assessment in AHyCo is conducted through assessments containing various types of questions. Assessments are used to determine concept understanding depending on the questions that students answer [10].

To create an assessment, the following parameters have to be determined: concepts with questions for the assessment, the numbers of questions in an assessment, and configuration rules that determine the number of questions to be included in the assessment from each concept. Therefore, it is not necessary to include questions one by one since they are automatically and randomly generated prior to every new assessment. In addition to that, question answers are permuted thereby making plagiarism difficult.

The types of assessment questions are:

- 1) Classical questions – Multiple choice questions with one or more correct answers. There are common hypermedia classical questions and parameterized classical questions [1].
- 2) SQL questions – Questions for which the answers are SQL queries. Evaluation is done automatically through the module for automatic evaluation of SQL assignments.
- 3) programming questions – Questions where the answer is a program written in a programming language. Evaluation is done automatically through the module for automatic evaluation of programming assignments [12].

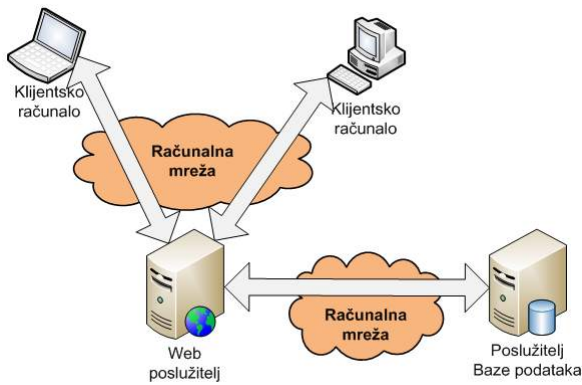
AHyCo's automated grading of programming assignments supports several programming languages (C, C++, C#, SQL). It is possible to extend the feature with other languages (with external compilers).

In a common use case, a student writes and uploads the source of a computer program, which is then graded automatically by the system, according to the output obtained by running the program against predefined test cases.

The following chapters explain in detail the process of creation, presentation and evaluation of automatically evaluated programming questions and assignments.

## 3. SYSTEM ARCHITECTURE

The student knowledge assessment module is a part of the learning management system AHyCo. AHyCo is a web based learning management system that can be accessed using a web browser through the internet or the intranet. AHyCo is implemented in Microsoft's ASP.Net technology, consists of a database server, runs on a web server, and is accessed by users as depicted in Figure 1.



Slika 1. Arhitektura AHyCo sustava

Sustav za provjeru znanja implementiran je kao troslojna arhitektura sa sljedećim slojevima:

- Sloj pristupa bazi podataka
- Sloj poslovne logike
- Sloj korisničkog sučelja
- Prezentacijski sloj

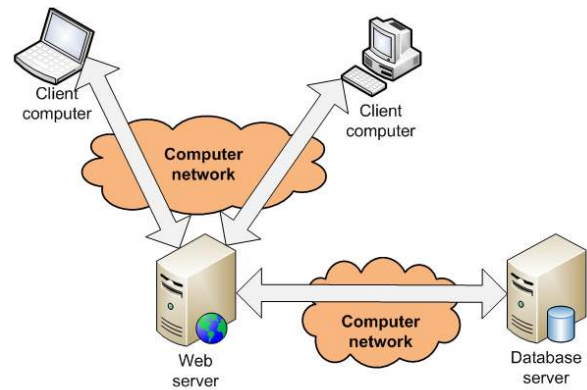
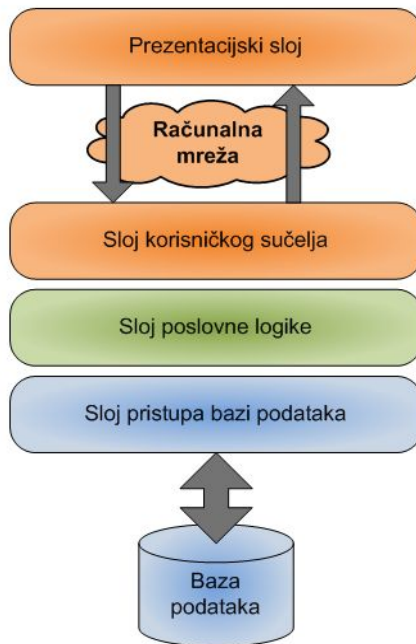


Figure 1. AHyCo system architecture

The knowledge assessment module is based on three-tiered architecture with the following layers (Figure 2):

- Data Access Layer
- Business Logic Layer
- User interface layer
- Presentation layer



Slika 2. Prikaz troslojne arhitekture sustava za provjeravanje znanja

Sloj pristupa bazi podataka ostvaren je upotrebom tehnologije *ADO.NET* koja omogućuje jednostavnu integraciju sa SQL Server-om koji je korišten kao sustav za upravljanje bazom podataka. Razredi koji sadrže metode za pohranu, dohvat, izmjenu i brisanje podataka u bazi nalaze se u prostoru imena *ProvjereLib.DataAccess*. Ideja je sloja pristupa bazi podataka da on ni na koji način ne ovisi o višim

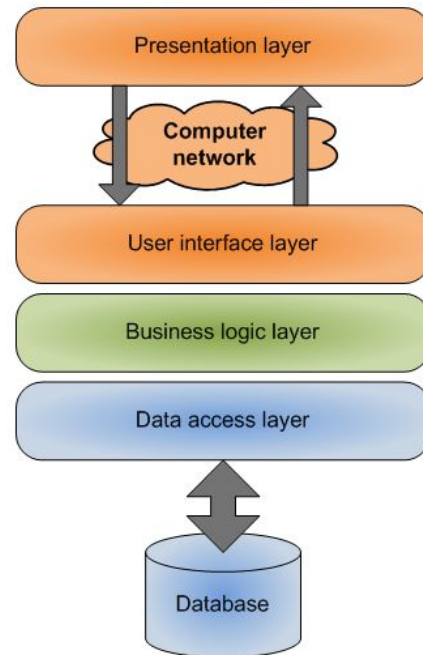


Figure 2. Three-tiered architecture of AHyCo 's module for knowledge assessment

The data access layer is realized with Microsoft's *ADO.NET* technology that enables simple integration with a database management system, namely an SQL Server. The namespace *ProvjereLib.DataAccess* contains classes for storing, retrieval, modification, and deletion of data. The main idea behind the data access layer is that it does not depend on the higher, more abstract layers, i.e. the business logic layer. All class

slojevima, kao npr. o sloju poslovne logike. U skladu s time metode razreda u prostoru imena `ProvjereLib.DataAccess` koriste isključivo razrede koje pruža ADO.NET.

Sloj poslovne logike modeliran je na temelju ER modela podataka. Razredi toga sloja nalaze se u prostoru imena `ProvjereLib.BusinessObjects`. Ideja je sloja poslovne logike upravljanje podacima na višoj, apstraktnoj razini u odnosu na sloj pristupa bazi podataka. Razredi toga sloja tako npr. implementiraju metode koje obavljaju ocjenjivanje pitanja ili pak cijele provjere, kao i validaciju podataka prilikom njihova unosa.

Sloj grafičkoga sučelja, zbog specifičnosti web-aplikacija, proteže se i na klijentskoj strani i na poslužiteljskoj strani. Na poslužiteljskoj strani razredi sloja grafičkoga sučelja nalaze se u prostoru imena `ProvjereLib.GUIObjects` te sadrže isključivo podatke koji se prezentiraju na klijentskoj strani u obliku HTML web-stranice [12].

#### 4. PROVJERAVANJE ZNANJA U SUSTAVU AHyCo

Prema Brusilovsky [2] „životni ciklus“ pitanja u *online* testovima podijeljen je u tri faze: priprema (prije aktiviranja ili aktivne faze), dostavljanje (aktivna faza) i vrednovanje (nakon aktivne faze). U nastavku rada će se opisati te faze s naglaskom na programskim pitanjima odnosno na automatskom ocjenjivanju programskih zadataka.

##### 4.1. Faza pripreme

Faza pripreme uključuje izradu, spremanje i odabir pitanja od kojih će se sastojati *online* provjera [1,2].

Sustav za provjeravanje znanja omogućuje unos pitanja kroz modul za pitanja. Moguće je unijeti pitanje bilo koje vrste, a unos se obavlja putem web-sučelja. Modul za pitanja omogućuje i izmjene na već pohranjenim pitanjima i to izmjene teksta pitanja, teksta odgovora (ako pitanje sadrži predefinirane odgovore) te ostalih parametara pitanja.

Kod unosa pitanja vrlo je važna mogućnost jednostavnog unosa složenog teksta. Uređivač teksta mora omogućiti unos slika, tablični prikaz, formatiranje teksta itd. Tehnologija .NET omogućila je implementaciju kontrole kojom je moguće uređivati tekst koristeći *Microsoft Word* no omogućen je i unos teksta pomoću HTML uređivača teksta.

Unos programskog pitanja može se podijeliti u dva dijela: unos teksta pitanja s točnim rješenjem i unos testova za pitanje. Sljedeći je korak unos točnog rješenja zadatka koji se sastoji se od tri dijela: prefiksa, točnog odgovora i sufiksa. Te je dijelove potrebno unijeti u odgovarajuća polja na web-stranici.

methods in the namespace `ProvjereLib.DataAccess` depend only on the methods provided by ADO.NET.

The business logic layer models all entities participating in the module's business logic. Its classes are stored in the `ProvjereLib.BusinessObjects` namespace. The main idea is to provide an abstraction from the data access layer, which deals with primitive data operations. BLL's classes therefore model assignment evaluation, question test case evaluation, and validate input data, etc. The UI layer runs on the server side and its classes are stored in the `ProvjereLib.GUIObjects` namespace. The web application's user interface represents the module's presentation layer providing simple validation prior to the server request processing [12].

#### 4. KNOWLEDGE ASSESSMENT IN AHYCO


According to Brusilovsky [2], the "life cycle" of a question in online assessments has been divided into three stages: preparation (before active life), delivery (active life), and assessment (after active life). The remaining chapters deal with these phases with special respect to the automatic evaluation of programming assignments.

##### 4.1. The preparation stage

The preparation stage includes authoring, storing and selecting a question or forming an online assessment [1,2].

The knowledge assessment system enables creation of questions through the module for question authoring. Using AHyCo's web interface, it is possible to create different types of questions. Already stored questions can be altered—question text, answer text and other question parameters can be changed.

In the process of question authoring, the ability of creating complex question texts is very important. The text editor should support authoring of questions that consist of images, tables, formatted text etc. .NET technology has enabled implementation of control by which text can be edited using Microsoft Word text editor. Text entry using HTML text editor is also enabled. The authoring of programming questions can be divided into two important parts: question text with specified answers and question test authoring. In addition to that, a program consisting of a prefix, suffix and the correct assignment solution must be stored in the AHyCo's web interface.



**Unos AOP programskog pitanja**

Botički, Ivica  
 05.04.2008  
 AHyCo  
 Upute za trenutnu stranicu

> Povratak u glavni izbornik
> Povratak na pregled pitanja po cjelinama
> Unos novog pitanja

Parametri predmeta	
Akademaska godina	2007./2008. ljetni semestar
Predmet	Algoritmi i strukture podataka (FER 2)
Modul	ASP - I Ciklus

Parametri pitanja							
Pitanje	5139						
Programski jezik	Cplusplus						
Aktivno pitanje	<input checked="" type="checkbox"/> Pitanje se koristi u provjerama						
Cjeline	<table border="1" style="display: inline-table; border-collapse: collapse;"> <thead> <tr> <th>Rbr</th> <th>Cjelina</th> <th>Brisanje</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>ASP - AOP - Datoteke</td> <td>Brisanje</td> </tr> </tbody> </table>	Rbr	Cjelina	Brisanje	1	ASP - AOP - Datoteke	Brisanje
Rbr	Cjelina	Brisanje					
1	ASP - AOP - Datoteke	Brisanje					

> Pehrana izmjena
> Testiranje izvođenja

> Unos/izmjena teksta pitanja

Tekst pitanja

Neka je zadana slijedna neformatirana datoteka u kojoj se nalaze zapisi oblika:

```
struct album{
char naziv[30+1];
char autor[20+1];
int prodano;
};
```

Napisati funkciju koja će za zadanog autora vratiti broj prodanih primjeraka albuma (prodano).  
Prototip funkcije mora biti:

```
int br_prodanih(FILE *f, char *autor);
```

Prefiks

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

struct album
{
char naziv[30+1];
char autor[20+1];
int prodano;
};

void prebaci(FILE *f, FILE *n) /* iz formatirane u neformatiranu */

int br_prodanih(FILE *f, char *autor)
{
struct album a;
int br = 0;
while (fread (&a, sizeof(a), 1, f)
{
if (strcmp(a.autor,autor) == 0) br = br + a.prodano;
}
return br;
}

int main()
{
FILE *f, *n;
char autor[20+1];
f=fopen
("C:\\Inetpub\\wwwroot\\aHyCo\\AOPDatoteke\\IN005_albumi_f.txt","r");
n=fopen("neformatirana.txt","wb");

if (f == NULL || n == NULL) {
printf("pogreška prilikom otvaranja datoteka");
}
else
{
prebaci(f,n);
}
fclose(f);
fclose(n);
}
```

Točan odgovor

Sufiks

Slika 3. Forma za unos programskog pitanja

Figure 3. AHyCo's web interface for programming assignment authoring

U polje *Točan odgovor* unosi se točno rješenje zadatka. Polje *Sufiks* koristi se u zadacima u kojima se traži implementacija funkcije. Tu je potrebno definirati funkciju *main()* koja će učitati ulazne parametre testova sa standardnog ulaza (funkcijom *scanf(...)*), pozivati funkciju koja je zadana zadatkom te ispisati rezultate izvođenja funkcije na standardni izlaz (funkcijom

Field *Correct answer* contains the correct assignment solution provided by the teacher, while the *Suffix* field is used in assignments that require a function implementation from students. In the case of programming assignments in programming language C, this field is used to define the *main()* function that reads input test parameters from the standard input using the

*printf(...)*). Povratni tip funkcije *main()* mora biti *int* a funkcija mora vraćati povratnu vrijednost naredbom *return*.

Definiranje provjera i uvid u obavljene provjere moguće je obaviti pomoću modula za provjere. Unos provjere može se podijeliti u dva dijela: definiranje parametara provjere te odabir sastava provjere. Provjera se sastoji od jedne ili više cjelina pri čemu u provjeru ulaze pitanja iz svake od odabranih cjelina.

Nakon što su uneseni pitanje i točan odgovor te su definirani testovi, moguće je obaviti testiranje tijekom faze dostavljanja.

#### 4.2. Faza dostavljanja – pisanje provjere

Faza dostavljanja uključuje prikazivanje pitanja, osiguravanje sučelja pomoću kojega će student odgovoriti te prihvaćanje odgovora za kasnije vrednovanje [1,2]. Ta faza ovisi o tehnologiji koja se koristi za sustav za učenje. Kod faze dostavljanja za AHyCo najvažniji dio je okolina za učenje temeljena na webu. Pisanje provjere pokreće se nakon što se student prijavi u AHyCo LMS te upiše lozinku u za to određeno polje na glavnoj stranici sustava AHyCo.

Web-obrazac za pisanje provjere može prikazati bilo koju vrstu pitanja. Ovisno o vrsti pitanja mijenja se prikaz web-stranice a time i funkcionalnosti specifične za pojedinu vrstu pitanja. Programsko pitanje u provjeri prikazano je na slici 4.

Provjeru je moguće predati na ocjenjivanje u bilo kojem trenutku klikom na poveznicu *Ocjenjivanje provjere*. Student je najprije preusmjeren na stranicu na kojoj su prikazani odgovori koje je odabrao (za klasična pitanja) ili sintaksna točnost (za programska pitanja).

Na toj stranici student ima mogućnost nastavka pisanja provjere ili predaje provjere na ocjenjivanje.

#### 4.3. Faza vrednovanja

U fazi vrednovanja sustav treba ocijeniti odgovor kao točan, netočan ili djelomično točan, dostaviti studentu povratnu informaciju, bodovati pitanje i spremati rezultate studenta [1,2].

Predajom AHyCo provjere na ocjenjivanje provjera se smatra obavljenom i student joj više ne može pristupiti. Nakon ocjenjivanja prikazuje mu se stranica sa sumarnim rezultatima provjere koja sadrži popis svih pitanja u provjeri s ostvarenim brojem bodova za svako pitanje te ukupni ostvareni broj bodova za provjeru. Student može pregledati i svako pitanje posebno.

Slika 5 prikazuje rezultate provjere, tj. kako izgleda programsko pitanje koje je vrednovano kao točno.

Pregled rezultata provjere za nastavnika razlikuje se od

*scanf()* function, calls the function specified in the assignment (one that students submit as a solution), and outputs the results by using the *printf()* function.

The test definition in addition to the completed assessment details can be accessed through the module for knowledge assessment as well. To create an assessment, its parameters as well as the concepts it covers have to be specified. Assessments are composed of one or more concepts from which the assessment questions are drawn.

As soon as questions and correct answers are defined and tests cases are supplied, it is possible to perform the assessment in the delivery stage.

#### 4.2. The delivery stage – assessment taking

The delivery stage includes a question presentation, an interface for the student to answer the question and the retrieval of the answer for evaluation [1,2]. This stage depends on the technology used for the learning system.

For the delivery stage of an assessment in AHyCo, the most important part is a web-based learning environment. An assessment starts as soon as a student has logged on to the AHyCo learning management system and has entered an assessment password.

The module for the presentation of an assessment question displays different question types differently but preserves the unique assessment page presentational characteristics. A displayed programming question is depicted in Figure 4.

An assessment can be automatically evaluated at any moment by following the link named *Assessment evaluation*. A student is first transferred onto a page that shows chosen answers (classical questions) or syntax correctness (programming questions).

After the check, the student can submit the assessment for automatic evaluation.

#### 4.3. The assessment stage

At the assessment stage, the system should evaluate answers as correct, incorrect, or partly correct, deliver feedback to student, grade the question, and record the students' performance [1,2].

When an assessment is submitted, students cannot access it anymore. After the automatic assessment evaluation, students are provided with summary assessment results containing all the assessment questions together with the number of points acquired question by question and for the whole assessment. The student can browse through the evaluated questions and get an idea of the correct answers.

Figure 5 shows the automatically evaluated assessment results when programming assignments are considered.

The screenshot shows the AHyCo online assessment interface. At the top, the logo 'AHyCo' is on the left, and the user information 'Budišćak, Ivan', '06.04.2008', and 'AHyCo' is on the right. The page title is 'Provjera: ASP AOP 1 0708'. Below the header, there is a navigation bar with 'Pitanje' and a progress indicator '1 2'. A green timer shows '02 d 23 h 59 m 44 s'. There are buttons for 'Privremeni prekid pisanja' and 'Ocjnjivanje provjere'. The main content area contains a programming question in C: 'Neka je zadana slijedna formatirana datoteka u kojoj se nalaze podaci o albumima: naziv (char[30+1]), autor (char[20+1]), prodano (int). (Podaci o jednom albumu zapisani su u jednom retku, odvojeni jednim razmakom. Pretpostaviti da naziv i autor ne sadrže praznine.) Napisati funkciju koja će vratiti broj albuma za koje je prodano granica ili više primjeraka. Prototip funkcije mora biti: int veci od(FILE \*f, int granica);'. A code editor shows the function signature: 'float \*KreirajMatricu(int n, float \*p, int m)'. A button 'Obavi sintaksnu provjeru' is at the bottom.

Slika 4. Prikaz programskog pitanja u provjeri

Figure 4. A programming assignment displayed to a student during the assessment

The screenshot shows the AHyCo online assessment interface displaying the results of a programming test. At the top, the logo 'AHyCo' is on the left, and the user information 'Budišćak, Ivan', '06.04.2008', and 'AHyCo' is on the right. The page title is 'Pregledavanje rezultata provjere: Mix provjera test'. Below the header, there is a navigation bar with 'Pitanje' and a progress indicator '1 2 3'. The score information shows 'Mogući broj bodova: 1,00' and 'Ostvareni broj bodova: 1,00'. There are buttons for 'Kraj pregledavanja'. The main content area contains the question: 'Napišite funkciju koja kvadrira zadani cijeli broj. Prototip funkcije je: int Kvadrat(int a);'. The results are: 'Sintaksna točnost: 😊' and 'Rezultat ocjenjivanja: 😊'. A table at the bottom shows the input and output values:

Ulaz	Očekivani izlaz	Dobiveni izlaz	Rezultat ocjenjivanja
4	16	16	😊

Slika 5. Pregled rezultata programskog testa

Figure 5. Results of a programming assessment displayed to a student



onog za studente jer je moguće da nastavnik neke vrste provjere ocijeni ručno. Uvidom u rezultate provjere nastavnik može vidjeti odgovor studenta na pojedino pitanje te prema potrebi, ako je npr. prilikom pisanja provjere došlo do tehničkih poteškoća u radu sustava AHyCo, izmijeniti i/ili priznati odgovor.

U nastavku je ukratko opisan način bodovanja programskih zadataka [3]. Modul spaja kôd koji je upisao student s malim dijelovima kôda koje je generirao nastavnik, kompajlira ga i izvršava nad posebno pripremljenim testnim primjerima. Rezultat je vrednovanja svakoga testnog primjera jedno od triju stanja: nije se kompajlirao ili nije uspio, djelomično je prošao te potpuno je prošao, a studentu se dodjeljuju bodovi prema formuli:

$$AR = \frac{AP}{\sum_i TP_i} \sum_i TR_i * TP_i \quad (1)$$

Oznake u formuli su:

- $AR$  – rezultat vrednovanja zadatka
- $AP$  – maksimalni broj bodova za zadatak
- $TP_i$  – maksimalni broj bodova za testni slučaj  $i$  ( $0 < i < n$ , gdje je  $n$  broj svih testnih slučajeva za zadatak)
- $TR_i$  – dodijeljeni bodovi za testni slučaj  $i$

Pri sadašnjoj implementaciji različita stanja iznose različite postotke od maksimalnih bodova pridruženih testnom slučaju: nije se preveo ili testiranje nije uspjelo (0%), testiranje je djelomično uspjelo (50%), testiranje je u potpunosti uspješno (100%).

## 5. ZAKLJUČAK I BUDUĆI PLANOVI

U ovome radu predstavljen je modul za provjeravanje znanja sustava za učenje AHyCo. Opisan je novi vlastiti pristup procesu izrade, dostavljanja i vrednovanja *online* provjera s naglaskom na poboljšanje načina provjeravanja znanja u predmetima iz područja programiranja uz pomoć automatskog vrednovanja programskih zadataka.

Sustav je u svakodnevnoj upotrebi na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu, te se u skladu sa zahtjevima korisnika, studenata i nastavnika stalno unapređuje. Njegovim unapređenjem podiže se i kvaliteta učenja jer omogućuje kontinuirane provjere znanja koje se automatski ocjenjuju i ne zahtijevaju dodatno opterećenje nastavnog osoblja.

Budući planovi razvoja AHyCo-ova modula za vrednovanje programskih zadataka uključuju poboljšanje sustava na osnovi analize podataka o studentskim rezultatima. Također je planirano proširenje modula novim programskim metodama, nadogradnja mehanizama testiranja te uporaba mobilnih uređaja u svrhu unapređenja mehanizama ispitivanja studenata.

The results overview for the teacher is different since it includes the possibility for repeating the process of automatic evaluation. Teachers can browse students' answers and repeat the evaluation process if they suspect an error of any kind has occurred.

The remainder of the paper describes the scoring scheme for programming assignments in AHyCo [3]. The module merges the student code with the teacher-generated base code snippets, compiles and runs it using the specified test cases. Each test case results in one of the three states: did not compile or failed, partially passed, and passed, and the student is awarded points specified by the following formula:

Brief explanation of the notation:

- $AR$  – assignment evaluation result
- $AP$  – maximum assignment points
- $TP_i$  – maximum points for the test case  $i$  ( $0 < i < n$ , where  $n$  is the number of test cases for an assignment)
- $TR_i$  – awarded points for the test case  $i$

In the current implementation, different states amount to a different percentage of maximum points assigned to a test case: did not compile and failed (0%), partially passed or output partially matched (50%), passed (100%).

## 5. CONCLUSION AND FUTURE PLANS

In this paper, the assessment module of the AHyCo learning management system has been presented. A new approach to the process of creation, delivery and evaluation of online assessments is described as well as an effort made in enhancing assessment mechanisms in programming courses by using automatic evaluation of programming assignments.

The system is used on a daily basis at the Faculty of electrical engineering and computing, University of Zagreb and is, depending on the requirements set by the teachers and students, constantly being upgraded. By automatizing assessment tasks, the overall quality of learning has been raised since teachers have more time left to work with students.

Future activities concerning the module for evaluation of programming assignments include the analysis of acquired student related data and its use in system's enhancement. Furthermore, the extension of the module with new programming methods and an upgrade of the assessment mechanism together with the use of mobile devices as a way of enhancing the assessment process are planned.

## 6. POPIS OZNAKA

rezultat vrednovanja zadatka	AR
maksimalan broj bodova za zadatak	AP
maksimalni broj bodova za testni slučaj	TP <sub>i</sub>
dodijeljeni bodovi za testni slučaj	TR <sub>i</sub>

## 6. LIST OF SYMBOLS

assignment evaluation result
maximum assignment points
maximum points for the test case
awarded points for the test case <i>i</i>

## LITERATURA REFERENCES

- [1] Hoić-Božić, N., Mornar, V., Pukljak Zoković, D., *The Model for Testing in Adaptive Hypermedia Courseware*, Proceedings of the ITI 2003 Conference, Cavtat, Croatia, 2003, p. 255-260
- [2] Brusilovsky, P., Miller, P., *Web-based Testing for Distance Education*, Proceedings of WebNet 99, Honolulu, Hawaii, United States, 1999
- [3] Botički, I., Pukljak-Zoković, D., Nizetić, I., *The analysis of student activity by the automatic evaluation of programming assignments in an online learning environment*, Proceedings of the ICAT 2007 Conference, Sarajevo, Bosnia and Herzegovina, 2007, CD-ROM edition
- [4] Kurt, A., Kubat, C., Oztemel, E., *Web-Based Virtual Testing and Learning in Material Science and Engineering*, International Journal of Engineering Education, Vol. 22 (2006), No. 5, p. 986-992
- [5] Kerer, C. et al. *ShareMe: Running a Distributed Systems Lab for 600 Students With Three Faculty Members*, IEEE Transactions on Education, Vol. 48 (2005), No. 3, p. 430-437
- [6] Garcia-Beltran, A., Martinez, R. *Web Assisted Self-assessment in Computer Programming Learning Using AulaWeb*, International Journal of Engineering Education, Vol. 22 (2006), No. 5, p. 1063-1069
- [7] Cheang, B, Kurnia, A., Lim, A., Oon W., *On automated grading of programming assignments in an academic institution*, Computers and Education, Vol. 41 (2003), No. 2, p. 121-131
- [8] Jones, E., *Grading student programs - a software testing approach*, Proceedings of the fourteenth annual consortium on Small Colleges Southeastern conference, Roanoke College, Salem, Virginia, United States, 2000, p. 185 – 192
- [9] Edwards, S., *Using test-driven development in the classroom: Providing students with concrete feedback on performance*, Proceedings of the International Conference on Education and Information Systems: Technologies and Applications (EISTA'03), International Institute of Informatics and Systemics, 2003, p. 421–426
- [10] Hoić-Božić, N., Mornar V., *AHyCo: a Web-Based Adaptive Hypermedia Courseware System*. Journal of Computing and Information Technology – CIT, Vol. 13 (2005), No. 3, p. 165-176
- [11] Hoić-Božić, N., Mornar, V., *Authoring of Adaptive Hypermedia Courseware Using AHyCo System*. Advances in Web-Based Education: Personalized Learning Environments, Hershey, Idea Group, 2005, p. 253-276
- [12] Budiščak I., *Knowledge testing in AHyCo system*. Graduation thesis, Faculty of electrical engineering and computing, University of Zagreb, Croatia, 2008

Primljeno / Received: 30.2.2008

Prihvaćeno / Accepted: 20.6.2008

Izvorno znanstveni članak

Original scientific paper

Adrese autora / Authors' address

Nataša Hoić-Božić

Odjel za informatiku Sveučilišta u Rijeci

Omladinska 14

51000 Rijeka, HRVATSKA

natasa.hoic@ri.t-com.hr

Ivan Budiščak

Ivica Botički

Fakultet elektrotehnike i računarstva

Sveučilište u Zagrebu

Unska 3

10000 Zagreb, HRVATSKA

ivan.budiscak@fer.hr

ivica.boticki@fer.hr