

S T U . .  
. . . . .  
F I I T .  
. . . . .

**Slovenská technická univerzita v Bratislave**  
**FAKULTA INFORMATIKY**  
**A INFORMA NÝCH TECHNOLOGIÍ**

---

# **Podpora vzdelávania v predmete Opis digitálnych systémov**

**(Tímový projekt)**

**Akademický rok:** 2009/2010  
**Študijný program:** PKSS  
**Vedúci projektu:** Ing. Elena Tomalová

Bc. Michal Majzlík  
Bc. Juraj Bleho  
Bc. Martin Boroš  
Bc. Štefan Olejník  
Bc. Jozef Szadvári  
Bc. Slavomír Žiak

# Obsah

---

|  |    |
|--|----|
| Obsah.....   | 2  |
| Zadanie .....  | 4  |
| Podpora vzdelávania v predmete Opis digitálnych systémov .....             | 4  |
| Úvod .....   | 5  |
| Motivácia .....  | 5  |
| Ciele práce .....  | 5  |
| Štruktúra dokumentu .....  | 6  |
| Použité skratky .....  | 6  |
| Analýza.....   | 8  |
| Podpora vzdelávania v predmete Špecifikačné a opisné jazyky .....          | 8  |
| Podpora vzdelávania v predmete Špecifikačné a opisné jazyky .....          | 11 |
| Podpora dištančného vzdelávania v predmete Opis digitálnych systémov ..... | 12 |
| Podpora vzdelávania v predmete špecifické a opisné jazyky .....            | 14 |
| Podpora dištančného vzdelávania v predmete Opis digitálnych systémov ..... | 16 |
| Zhodnotenie analýz.....  | 18 |
| Špecifikácia .....   | 19 |
| Špecifikácia systému .....   | 19 |
| Privilégia pre entitu Učiteľ .....   | 20 |
| Privilégia pre entitu Študent .....  | 21 |
| Rozšírenie modulu Test .....   | 22 |
| Návrh.....   | 25 |
| Vyhodnotenie otázky typu Petriho sieť .....                                | 25 |
| Podpora vzdelávania v predmete Opis digitálnych systémov .....             | 27 |
| (LETNÝ SEMESTER).....  | 27 |
| Návrh.....   | 28 |
| Architektúra systému .....   | 28 |
| Fyzický model údajov .....   | 29 |
| Podpora vzdelávania v predmete Opis digitálnych systémov .....             | 32 |
| (ZIMNÝ SEMESTER).....  | 32 |
| Implementácia .....  | 33 |
| Editor Petriho sietí.....  | 33 |
| Prototyp .....   | 34 |

|  |    |
|--|----|
| asový plán na zimný semester .....   | 35 |
| Zhodnotenie za zimný semester .....  | 36 |
| Podpora vzdelávania v predmete Opis digitálnych systémov .....             | 37 |
| (LETNÝ SEMESTER).....  | 37 |
| Implementácia .....  | 38 |
| Implementácia Petriho siete.....   | 38 |
| Implementácia „chy aposunovky“ .....                                       | 39 |
| Testovanie .....   | 41 |
| Výsledky testovania.....   | 41 |
| Možné vylepšenia.....  | 42 |
| Návrhy na vylepšenie .....   | 42 |
| Funkcionalita .....  | 42 |
| Ovládanie aplikácie .....  | 42 |
| Grafické rozhranie .....   | 42 |
| Inštalácia.....  | 43 |
| Offline mód.....   | 44 |
| Online mód .....   | 44 |
| Záver.....   | 45 |
| Zdroje .....   | 46 |
| Podpora vzdelávania v predmete Opis digitálnych systémov .....             | 47 |
| (ZIMNÝ SEMESTER).....  | 47 |
| Prílohy .....  | 48 |
| Príloha A - Ukážka HTML kódu s Java Appletom Petrinet editora.....         | 48 |
| Príloha B - Ukážka XMLDB dátového modelu pre otázku typu Petriho sie ..... | 49 |
| Príloha C - Používateľská príručka prototypu .....                         | 50 |
| Podpora vzdelávania v predmete Opis digitálnych systémov .....             | 54 |
| (LETNÝ SEMESTER).....  | 54 |
| Príloha D – Používateľská príručka.....                                    | 55 |

# Zadanie

---

## Podpora vzdelávania v predmete Opis digitálnych systémov

Analyzujte existujúce aplikácie a moduly, vytvorené pre podporu predmetu Špecifika né a opisné jazyky. Analyzujte tiež dostupné vzdelávacie aplikácie s podobným zameraním. Pri analýze sa zamerajte najmä na podporu správa zadaní (pride ovanie zadaní, zber, overovanie správnosti ich vypracovania a automatické ohodnotenie) a overovania praktických zru ností študentov.

Na základe analýzy navrhните a implementujte e-learningové moduly (prípadne externé aplikácie) pre správu výu by predmetu Opis digitálnych systémov (náhrada predmetu Špecifika né a opisné jazyky), ktoré budú podporova správu a vyhodnocovanie zadaní a zabezpe ova overovanie získaných praktických zru ností študentov v rámci predmetu.

Odporú aná literatúra:

- Matej Mayer, Bc., Podpora dištan ného vzdelávania v predmete Opis digitálnych systémov, Diplomová práca, FIIT STU Bratislava, máj 2009

- Andrej Letkovský, Bc., Podpora dištan ného vzdelávania v predmete Opis digitálnych systémov, Diplomová práca, FIIT STU Bratislava, máj 2009

- Izsák Peter, Automatické vyhodnocovanie programov vo VHDL, Závere ný projekt, FEI STU Bratislava, máj 2000

- Tím 3PSS, Podpora vzdelávania v predmete Špecifika né a opisné jazyky, Tímový projekt, FIIT STU Bratislava, máj 2008

- Tím 4PSS, Podpora vzdelávania v predmete Špecifika né a opisné jazyky, Tímový projekt, FIIT STU Bratislava, máj 2008

# Úvod

---

Účelom tejto práce je zjednodušiť a zatriktívniť, výučbu a následne aj testovanie znalostí študentov z predmetu Opis digitálnych systémov, ktorý sa využíva na Fakulte Informatiky a Informačných technológií STU.

## Motivácia

Zo všetkých zverejnených tém nás najviac oslovila téma s názvom „Podpora vzdelávania v predmete Opis digitálnych systémov“. Nakoľko veľa z nášho tímu už v minulosti pracovala s touto problematikou a už pri práci na bakalárskom projekte, alebo na iných projektoch v rámci zadaní v predmetoch bakalárskeho štúdia. Na základe doteraz nadobudnutých poznatkov a z dôvodu silnej motivácie pre ďalšiu prácu v oblasti elektronického vzdelávania má náš tím podnet a rozvíjať a riešiť túto problematiku využitím vedomostí a zručností. Naším cieľom je zefektívniť a zjednodušiť proces vzdelávania a už ako v predmete Špecifické a opisné jazyky alebo ako elektronického vzdelávania vo všeobecnosti. Vzhľadom na tieto skutočnosti si myslíme, že bol výber témy opodstatnený a naša motivácia je väčšia z presvedčenia, že naša práca bude prínosom v elektronickom vzdelávaní.

## Ciele práce

Cieľom práce je oboznámiť sa s rôznymi systémami elektronického vzdelávania a zistiť, ktorý sa pre náš projekt najviac prospešný (zatiaľ ako cieľ vývoja predpokladáme systém Moodle, pre jeho relatívne ľahkú rozšíriteľnosť a jeho nasadenia v učenom procese na našej fakulte).

Ďalším cieľom je analyzovať existujúce riešenia, s dôrazom na správu zadaní, ktoré študenti vypracúvajú a odovzdávajú, následne na to vytvoriť systém, ktorý by mal umožňovať plánovanie a správu rôznych akcií pre študentov a to:

- termíny odovzdania priebežných častí projektov,
- termíny prezentácií hotových projektov,
- elektronické odovzdávanie v stanovených časoch
- automatizované vyhodnocovanie zadaní

- zobrazí úlohy v prehľadnom harmonograme

Cieľom je vytvoriť modul schopný automatizovane hodnotiť študentské projekty, taktiež uľahčiť prístup ku kvalitným študijným materiálom, umožniť študentom pridávať vlastné poznámky, riešenia príkladov a zostavenie dostatočného počtu testovacích otázok na generovanie testov s možnosťou okamžitého vyhodnotenia otázky, potvrdenia zadanej možnosti, alebo zamietnutia s príslušným vysvetlením prečo je odpoveď nesprávna.

## Štruktúra dokumentu

Tento dokument je štruktúrovaný nasledovne:

### Analýza

V tejto stati sa náš tím sústredil na analýzu už existujúcich riešení. Analyzované boli diplomové práce a tímové projekty z predošlých akademických rokov, ktoré boli riešené na fakulte Informatiky a informačných technológií STU.

### Špecifikácia

V tejto stati je bližšie špecifikované, ktoré vlastnosti a požiadavky by mal mať náš výsledný produkt.

### Hrubý návrh

V tejto stati je načrtnutý výsledný systém.

## Použité skratky

**PKSS** – Počítačové a komunikačné systémy a siete

**PSS** – študijný odbor Počítačové systémy a siete

**VHDL** – (VHSIC Hardware Description Language) programovací opisný jazyk slúžiaci pre opis hardvéru. Používa sa pre návrh a simuláciu digitálnych integrovaných obvodov

**HTML** – Hypertextový značkový jazyk (HyperText Markup Language; HTML) je značkový jazyk určený na vytváranie webových stránok a iných informácií zobraziteľných vo webovom

prehliada i. HTML kladie dôraz skôr na prezentáciu informácií (odseky, fonty, váha písma, tabuľky atď.) ako na sémantiku (význam slov).

**PHP** – [2]PHP: Hypertext Preprocessor) je populárny open source skriptovací programovací jazyk, ktorý sa používa najmä na programovanie klient-server aplikácií (na strane servera) a pre vývoj dynamických webových stránok..

**SQL** – (Structured Query Language) – jazyk na manipuláciu s relačnou databázou

**MySQL** – [3]MySQL je slobodný a otvorený viacvláknový, viacúčítateľský SQL relačný databázový server. MySQL je podporovaný na viacerých platformách (ako Linux, Windows a Solaris) a je implementovaný vo viacerých programovacích jazykoch ako PHP, C++ a Perl. Databázový systém je relačný typu DBMS (database management system).

# Analýza

---

Pri analýze prác a projektov, ktoré nám boli odporúčané v zadaní projektu sa kládol dôraz hlavne na zadania s témou „Podpora vzdelávania v predmete Špecifické a opisné jazyky“ a zadania „Podpora dištančného vzdelávania v predmete Opis digitálnych systémov“. Krátkou analýzou prešli aj iné projekty, s podobnou tematikou, ktoré sa však odlišujú od nášho zadania, preto neboli podrobnejšie študované a nie sú uvedené v tomto dokumente.

## Podpora vzdelávania v predmete Špecifické a opisné jazyky

(Tím 4 PSS, Tímový projekt 2009)

### Ciele projektu

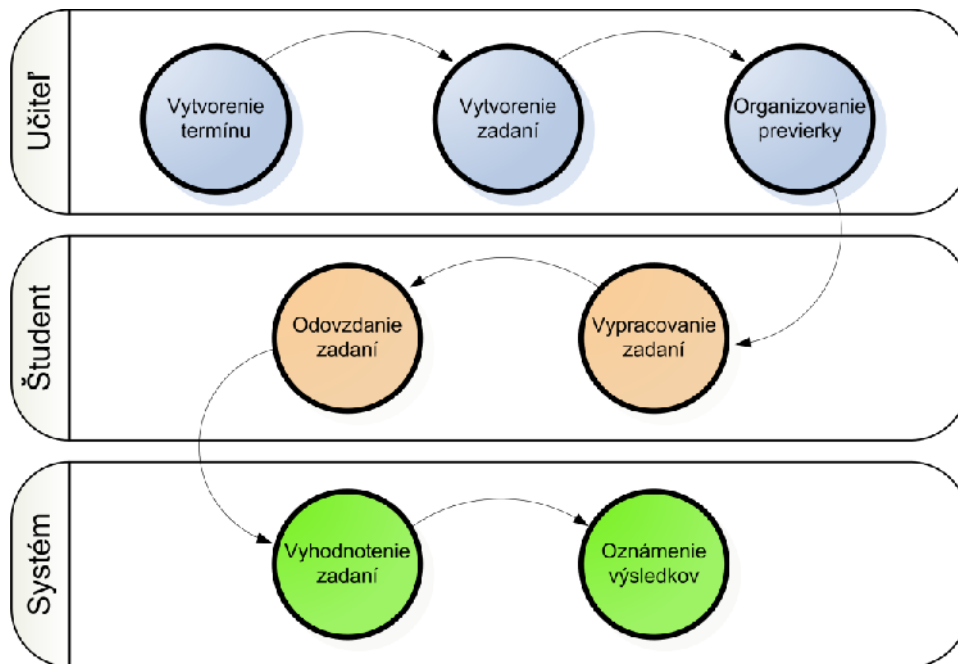
Tím sa zamerával na overovanie praktických znalostí v predmete Špecifické a opisné jazyky v oblasti problematiky Petriho sietí. Ako cieľ projektu si tím stanovil vytvorenie aplikácie na modelovanie, simuláciu a analýzu Petriho sietí, ktorá bude schopná vyhodnocovať správnosť študentmi namodelovanej siete a tým poskytovať spätnú väzbu na ich vedomosti. Cieľom vytvorenia aplikácie, bolo aj použiť nosič pri testovaní študentov a tým odľahčiť náročnosť kontrolovania správnosti vykonávaných prác a znížiť náročnosť.

### Špecifikácia požiadaviek

Na základe analyzovaných dostupných zdrojov podobných projektov sa tím rozhodol špecifikovať nasledujúce požiadavky.

Vytvorenie aplikácie schopnej automaticky vyhodnocovať zadania z tematiky Petriho sietí. Zameranie na teoretické a praktické vedomosti študentov a tomu prispôsobenie aplikácie. Iže aplikácia má podporovať tieto dva typy zadaní. Zjednodušený priebeh fungovania možno nájsť na nasledujúcom obrázku, ktorý je prevzatý z tejto práce.





Obr. .1 Priebeh testovania a opravy

- Systém má podporovať tieto činnosti:
  - vytvorenie termínu
  - vytvorenie zadanií
  - organizovanie previerky
  - vypracovanie zadanií
  - odovzdanie zadanií
  - vyhodnotenie zadanií
  - oznámenie výsledkov
- V systéme vystupujú dve entity:
  - Učiteľ
  - Študent

Entita učiteľ má právo na nasledujúce činnosti uvedené na obrázku.



Obr. .2 prípady použitia pre používate a U ite

- Entita Študent ma právo na nasledujúce innosti:
  - výber termínu
  - výber príkladu
  - vypracovanie príkladu
  - odovzdanie testu
  - zobrazenie výsledkov hodnotenia

## Návrh, implementácia a testovanie

Autori navrhli architektúru systému, ako aplikáciu Klient-Server. Jednotlivé údaje boli ukladané do databázy s použitím technológie MySQL. Hlavná časť systému bola implementovaná v jazyku JAVA. Dizajn aplikácie je postavený na princípe návrhového vzoru MVC (Model-View-Controll) – podrobnejšie opísaný v dokumente. Testovaním tejto aplikácie tím odhalil, pomerne veľa množstvo chýb, ktoré boli následne doladené.

## Záver

Výsledkom projektu je funkčná aplikácia na preverovanie teoretických a praktických znalostí v modelovaní, simulácii a analýze Petriho sietí. Základnú myšlienku a jadro aplikácie by sme mohli využiť pri realizácii nášho tímového projektu. Táto práca obsahuje návrh vhodnej architektúry systému, ktorá by mohla byť použitá pri implementácii našej aplikácie.

## Podpora vzdelávania v predmete Špecifikačné a opisné jazyky

(Tím 4 PSS, Tímový projekt 2008)

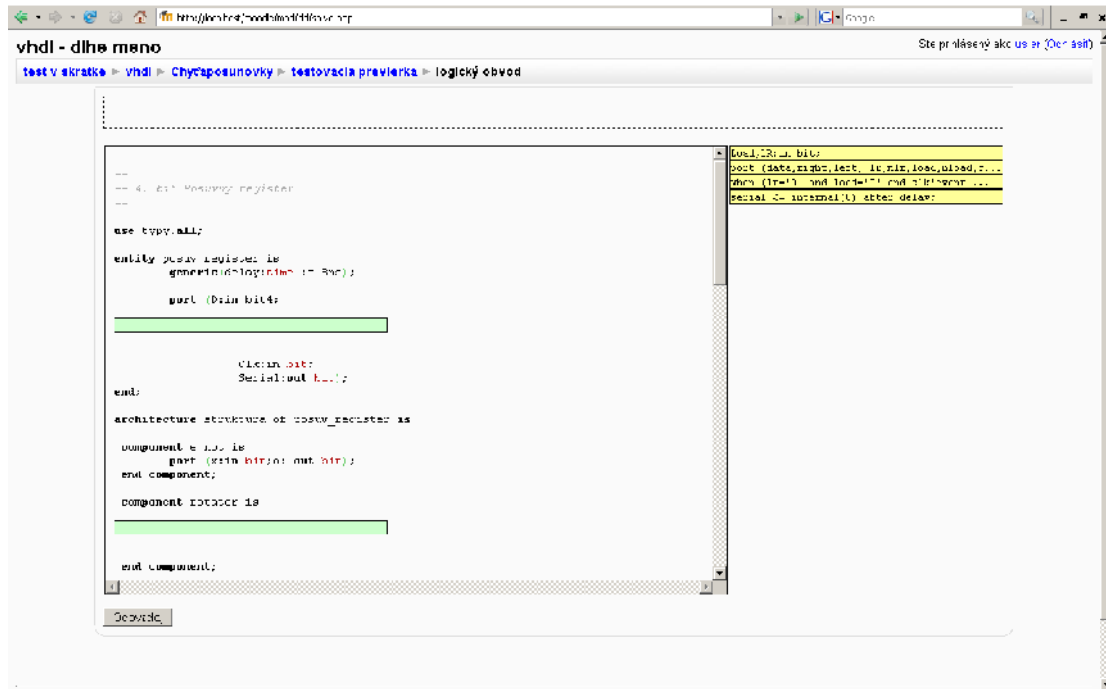
## Analýza

Po analýze niekoľkých riešení sa tím rozhodol vytvoriť samostatný modul do vývojového modulu Moodle.

Tím vytvoril modul, ktorý pedagógom umožní zostavenie, plánovanie a organizovanie previerok pre študentov. Previerky sa skladajú z niekoľkých zadaní týkajúcich sa opisu digitálnych obvodov v jazyku VHDL. Vyplnené zadanie pripraví, má možnosť zmeniť, alebo priradiť k previerkam. Študent v stanovenom čase zadanie vyplní a odošle na spracovanie. Systém zadanie ohodnotí. Zadania a previerky sú automaticky skopírované do archívnej časti modulu.

Osobitnou časťou je samotné zadanie - **Chybový posunok**. Ide o zaujímavý spôsob ako otestovať vedomosti študentov, formou doplnenia chýbajúcich kusov VHDL kódu do existujúcej kostry programu. V pravej časti obrazovky sa nachádza kostra VHDL programu a v ľavej časti sú fragmenty kódu, ktoré je možné systémom drag&drop (chybový a posunok)

umiestni na svoje miesto v kostre programu. Jednotlivé fragmenty majú svoju váhu a na základe tejto váhy a po tu správne umiestnených fragmentov dostane študent ohodnotenie. Na obrázku 3. je obrazovka s rozpracovanou Chy aposunovkou.



Obr. 3 Ukážka aplikácie Chy aposunovka

## Záver

Autori tejto práce identifikovali nieko ko návrhov na zlepšenie, náš tím sa rozhodol preskúma možnosti prvého návrhu - zakomponova Chy aposunovku do testovacieho modulu systému Moodle - Test.

## Podpora dištan ného vzdelávania v predmete Opis digitálnych systémov

(Bc. Matej Mayer, Diplomový projekt 2009)

## Analýza

Diplomová práca Bc. Mateja Majera sa zaoberá vytvorením modelu na podporu výu bu vzdelávania jazyku VHDL, ktorý sa vyu uje na predmete opis digitálnych systémov. V práci analyzuje viacero druhov vzdelávacích systémov, z ktorých si nakoniec na implementáciu vybral systém Moodle. Ten sa totiž už v danom ase využíval na výu bu v danom predmete. Do systému Moodle implementoval modul, ktorý sa delí na tri asti,

konkrétne as študent, as u íte a as systém. Každá z týchto astí podporuje rôzne funkcie zabezpe ujúce požiadavky na výu bu VHDL jazyka.

as študent slúži na odskúšanie študentov, ktorý študujú predmet Opis digitálnych systémov. Študentom je umožnené editova , kompilova a spusti simuláciu kódu VHDL. Ke sa študent rozhodne, môže odovzda zadanie ako finálne, toto sa porovná s referen nou vzorkou a zároveň sa automaticky vyhodnotí. Po finálnom odovzdaní si budú študenti môc automaticky prezrie výsledok skúšky.

as u íte slúži vy u ujúcim na vytváranie zadaní, úpravu výsledkov študentov a zároveň prezeranie zadaní. Pri vytváraní zadaní môže u íte stanovi termín prístupnosti modulu pre študentov, d ťku testu, maximálny po et bodov, bodovú zrážku pri nezhode a základný po et bodov pri úspešnom skompilovaní. Ak sa kód nepodarí skompilova zadanie sa musí ohodnoti vy u ujúcim manuálne. Táto as ešte podporuje vkladanie zadaní, obrázkov, kostry kódu, referen nej testovacej vzorky a vzorového riešenia. alej má u íte k dispozícii as prezeranie zadaní, ktoré mu umož ťuje export zadaní do cvs a zip formátov a prezeranie odovzdaných zadaní a výsledkov automatického vyhodnotenia.

Poslednou as ou vytváraného modulu je as systém, ktorá zabezpe uje chod a funk nos vytváraného modulu. Táto as sa delí na pod asti obsluhy VHDL kódu, odpo ítavania asu, uloženia a ohodnotenia odoslaných kódov. Pri ohodnocovaní kódov študent vytvoril ohodnotenie na základe porovnania referen nej vzorky a vzorky odovzdanej daným študentom. Porovnanie zdrojových kódov sa mu implementova nepodarilo. Pod as obsluhy VHDL kódu slúži na kompiláciu a simuláciu kódu a nahradenie testovacích vzoriek.

## Záver

Na kompiláciu a simuláciu študent použil dve asti Modelsimu a to kompilátor VCOM a simulátor VSIM, ktoré integroval do systému Moodle. Toto riešenie je celkom jednoduché a preh adné, a preto sme sa ho rozhodli použi aj v našej práci. Komunikácia VCOM-u a VSIM-u sa deje prostredníctvom konzoly. Porovnanie vzoriek sa deje tak, že sa odsimuluje odovzdaná vzorka, alej sa odsimuluje referen ná vzorka, ktoré sú následne uložené vo wlf súboroch. Simulátor VSIM podporuje porovnanie dvoch takýchto súborov, pri om výsledok porovnania je text. Problémom, s ktorým sa študent nevedel popasova je grafické znázornenie daného porovnania, o by mohlo by zjednodušenie práce pre u íte a

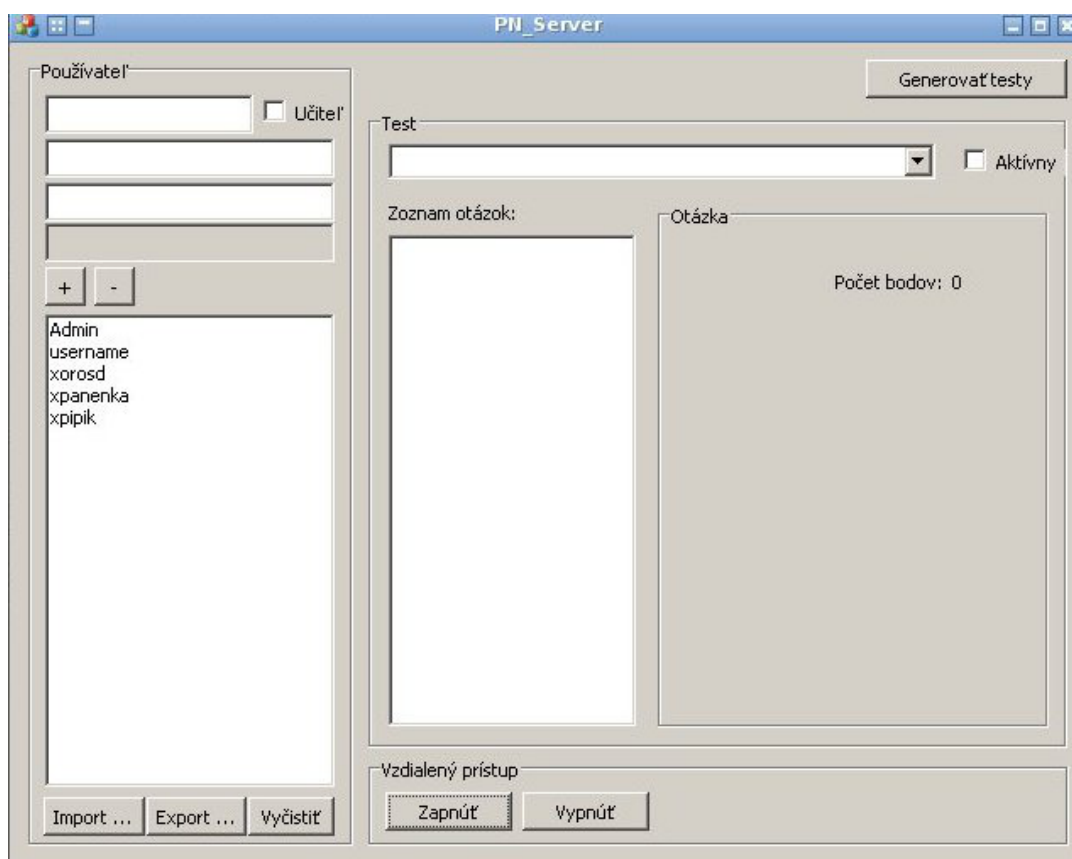
kontrolujúceho výsledky skúšky. Preto by sa o riešenie grafického zobrazenia mohol pokúsi náš tím.

## Podpora vzdelávania v predmete špecifické a opisné jazyky (Tím Surfin'birds, Tímový projekt 2009)

### Analýza

V rámci predmetu Tímový projekt sa tím SURFIN'BIRDS, v zložení Bc. Peter Koine, Bc. Šimon Hupka, Bc. Ján Turo , Bc. Michal Choleva, Bc. Emil Mo ovský pod vedením Ing Tomalovej, zaoberal Podporou vzdelávania v predmete Špecifické a opisné jazyky. Vo svojom projekte sa venovali Petriho sie am. Rozhodli sa vytvoriť externú aplikáciu v jazyku C++. Aplikácia je typ klient-server.

Na serveri sa nachádza databáza študentov a vyu učiacich. K serveru majú priamy prístup len vyu učiaci. Vyu učiaci tu môže vytvárať testové úlohy, z ktorých sa po as testu náhodne niektoré vyberú a zašlú študentom na riešenie. Zasielanie zadaní po as skúšky je zašifrované ke že spolu so zadaním sa zasielajú aj odpovede - testové úlohy teda vyhodnocuje klientska aplikácia.



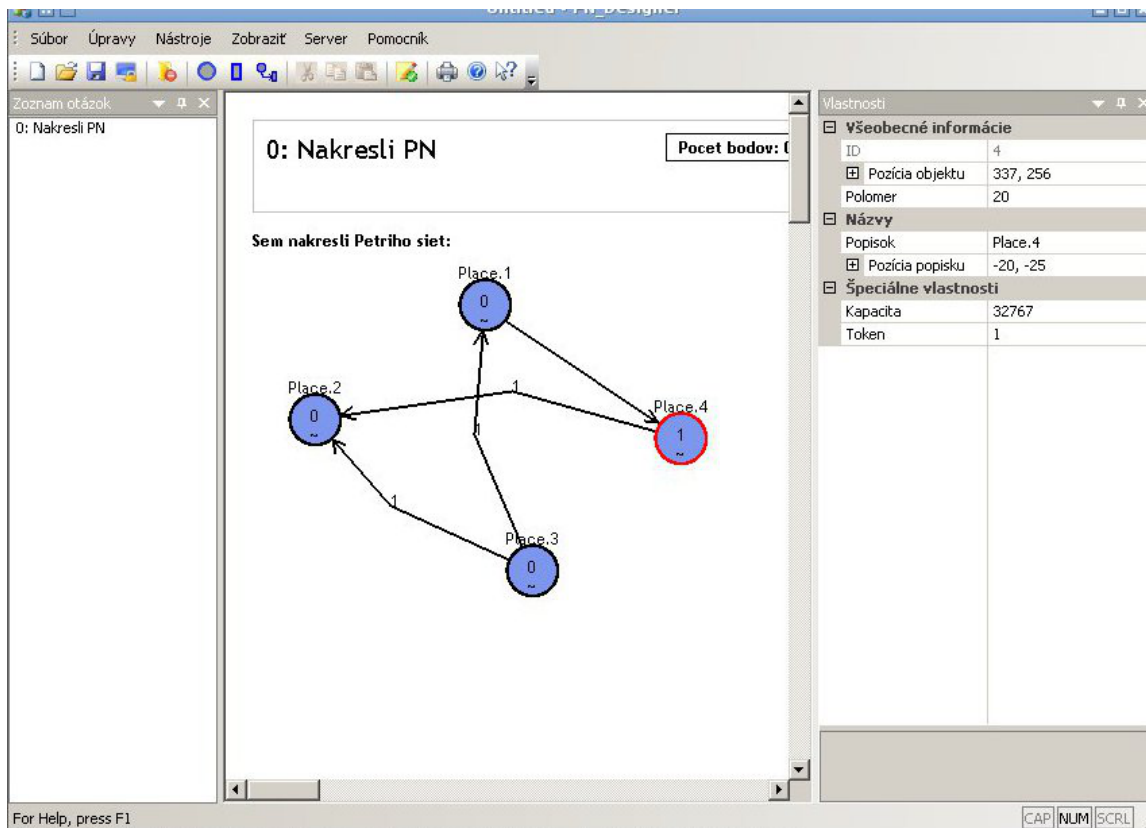
Obr. .4 ukážka aplikácie PN.Server

Klient má dve odlišné úlohy. Prvou je jeho použitie pri skúškach. Študenti sa pomocou neho prihlásia na server, dostanú úlohy na riešenie. Tie vyriešia v klientskej aplikácii. Spolu so zadaním úloh aplikácia prijme aj ich riešenia, ktoré potom následne vyhodnotí a výsledok odošle na server. Z dôvodu bezpečnosti, aplikácia sleduje spustené procesy aby sa dala odhaliť prípadná komunikácia medzi študentmi, snaha o zistenie odpovedí zo zašifrovanej komunikácie klientskej aplikácie so serverom ale podvádzanie pomocou internetu. Počas testu je teda povolené mať spustenú len klientsku aplikáciu.

Druhou úlohou aplikácie je jej možnosť bežať samostatne bez servera. Spolu s aplikáciou sa distribuujú kolekcia úloh na riešenie doma pre štúdium a precvičovanie sa.

V projekte sú 2 typy úloh. Prvý je doplnenie neúplnej Petriho siete podľa zadania. Druhým je kompletný návrh Petriho siete. Pri prvom type je riešenie len jedno a jednoznačné. Pri druhom type aplikácia vyhodnocuje riešenie podľa vlastností študentom navrhnutej siete, ktoré sa dajú overiť algoritmom.

Pre vyhodnocovanie Petriho sietí prebrali algoritmus na vyhodnotenie vlastností Petriho sietí z diplomovej práce Bc. Ľuboša Heribana - Diagnostika porúch založená na PS modeloch. Algoritmus na porovnávanie Petriho sietí pomocou incidencej matice.



Obr. .5 Ukážka modelu Petriho siete

Ovládanie programu má bežné pracovné prostredie, hlavnou časťou je pracovná plocha kde sa kreslia Petriho siete. Tá je rozdelená do mriežky, pričom v každom bode mriežky sa môže nachádzať jeden objekt - miesto alebo prechod siete. Aplikácia sa ovláda pomocou ponuky akcií a panelom nástrojov. Je potrebné len jedno tlačítko myši aby tým zabezpečila bezproblémovosť použitia aj na všetkých Tablet PC zariadeniach. Tým sa dostávame k ďalšej úlohe, ktorú Tím 3 riešil a to je podpora stylusu pre Tablet PC. Okrem gest zabudovaných v operačnom systéme sa rozhodli študenti implementovať aj vlastné riešenie. Aplikácia pozná tri typy akcií stylusom (prípadne myšou). Prvou je používateľ nakreslí akýkoľvek súvislý obrázok, ktorý sa pretne v jednom bode, vyhodnotí sa to ako miesto v Petriho sieti, ak sa tento obrázok nepretne tak sa jedná o prechod. V prípade spojenia miest a prechodov je potrebné kliknúť na jeden aj druhý objekt a tým sa medzi nimi spojí - hrana.

## **Podpora diaľneho vzdelávania v predmete Opis digitálnych systémov**

(Diplomový projekt)

### **Analýza**

Cieľom tohto projektu, bolo vytvorenie multimedialného nástroja pre podporu výučby v predmete Opis digitálnych systémov. Vytvorený nástroj, by mal umožniť získavanie a overovanie praktických zručností z jazyka SystemC. Ing. Letkovský sa rozhodol použiť systém moodle a rozšíriť ho o ďalšie moduly. Rozhodol sa tak z nasledovných dôvodov: Je šírený pod licenciou GNU, čo znamená, že jeho zdrojový kód môže byť ľahko upravovaný. S tým súvisí aj fakt, že je modulárny a dobre zdokumentovaný. Ďalšou výhodou je to, že spĺňa štandardy e-learningu a štandardy W3C. Dôvodom je aj jeho používanie na Fakulte Informatiky a Informačných technológií. Ako web server bol zvolený Apache HTTP Server a ako databázový server MySQL. Ide teda o aplikáciu typu server-klient. Na strane servera je systém moodle s pripojeným modulom a na strane klienta webový prehliadač.

Vytvorené moduly slúžia dvom skupinám: učiteľom a študentom. Študentom umožňuje:

- Tvorbu zdrojového kódu so zvýrazňovaním syntaxe
- Overenie syntaktickej správnosti vytvoreného zdrojového kódu



- Simuláciu návrhu
- Odovzdanie vytvoreného zadania
- Priebežné uloženie

Uite ovi systém umož uje:

- Tvorbu zadaní
- Nastavenie asového limitu pre vypracovanie
- Prezeranie odovzdaných zadaní
- Úprava automatického hodnotenia
- Archivácia odovzdaných riešení

Na overenie syntaktickej správnosti bol použitý externý kompilátor ModelSim. Výhodou ModelSimu je možnosť volania jeho súastí, kompilátora VCOM a simulátora VSIM z príkazového riadku. Pre kompiláciu zdrojového kódu v jazyku SystemC je však potrebné použiť kompilátor SSCOM.

Systém je tvorený dvomi modulmi: skuskasc a vysledkycsc. Modul skuskasc je navrhnutý tak, aby študentovi neumožoval žiadne neoprávnené zásahy. Študent je pred začatím riešenia informovaný o tom ako test vykona a aké zadanie má riešiť. Študent programuje svoje riešenie v textovom editore, ktorý umožňuje priebežné uloženie, kompiláciu, odovzdanie a simuláciu. Výsledky kompilácie a simulácie sa zobrazujú pod textovým editorom. Editor umožňuje spustenie zvýrazňovania syntaxe a číslovanie riadkov. Modul poskytuje uite ovi formulár, v ktorom do modulu vloží všetky potrebné informácie o jednom teste. Modul je implementovaný tak, že jedna jeho inštancia obsahuje iba jedno zadanie. Pre iné zadanie je potrebné vytvorenie novej inštancie. Toto riešenie prináša uite ovi uah enie práce s tvorbou skupín.

Modul vysledkycsc je navrhnutý tak aby pomohol uite ovi vyh adáva výsledky konkrétneho študenta alebo zálohovať riešenia všetkých študentov, ktorí vypracovávali vybraný test. Tento modul poskytuje uite ovi dva formuláre. Prvý formulár Vyh adávanie riešení slúži na zobrazenie riešenia konkrétneho študenta pre vybraný test. V prípade potreby má uite možnosť upraviť hodnotenie. Druhý formulár Archivácia riešení ponúka možnosť uloženia všetkých výsledkov alebo riešení v dvoch formátoch: zip archív, csv.

## Záver

Tento diplomový projekt rozdeľuje privilégia na ovládanie systému, medzi role: u študent. Podobné rozdelenie práv by sme mohli zohľadniť pri návrhu a implementácii nášho diplomového projektu.

## Zhodnotenie analýz

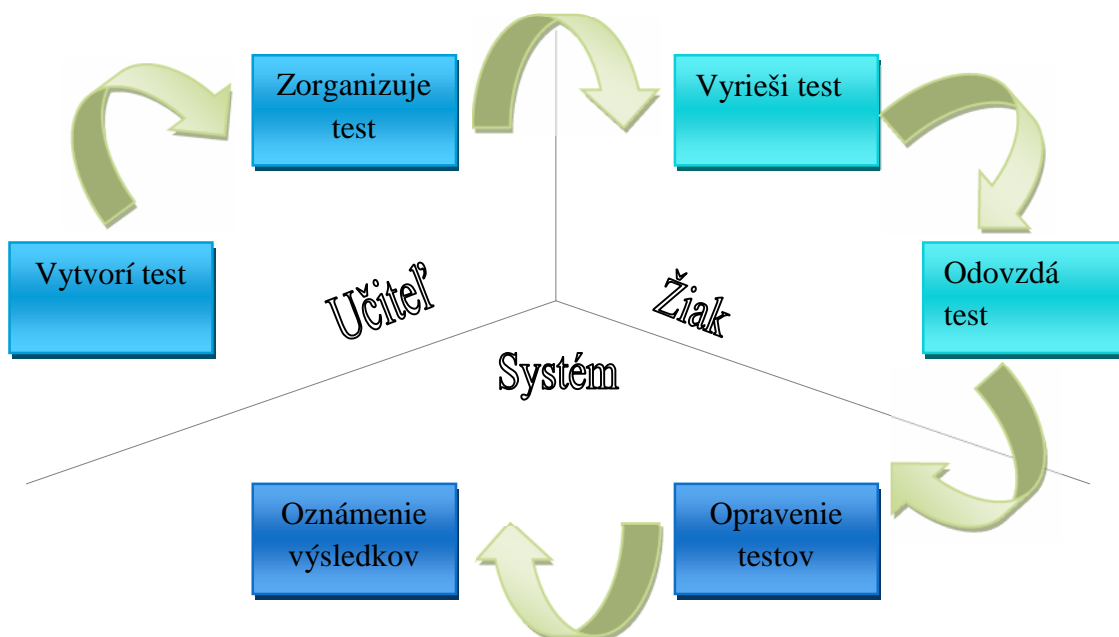
Každý jeden analyzovaný projekt obsahuje niekoľko hlavných bodov, ktorými by sme sa mohli inšpirovať pri riešení nášho projektu. Nad hlavnou kostrou systému sa stále pracuje. Na základe analyzovaných projektov, by náš tím mohol uvažovať nad zostavením aplikácie klient-server, využiť rozdelenie privilégií medzi entity študent-u študent, alebo prípadne navrhnuť architektúru systému podobnú už predtým analyzovaných a implementovaných projektov. Pri realizácii nášho projektu tím uvažuje nad využitím a vylepšením už vytvoreného zdrojového kódu z práce: „Podpora vzdelávania v predmete špecifické a opisné jazyky“, ktorú naimplementoval tím 4. z odboru PSS v akademickom roku 2008/2009.

# Špecifikácia

V tejto stati náš tím vytvoril špecifikáciu požiadaviek pre výslednú aplikáciu. alej je bližšie ur ené aké funkcie bude náš systém obsahova . V špecifikácií sú rozpracované jednotlivé entity a ich privilégia v e-learningovanom systéme.

## Špecifikácia systému

Systém pre podporu vzdelávania v predmete Opis digitálnych systémov by mal zabezpe ova automatizované vytváranie a vyhodnocovanie testov. Tieto testy sú hlavne z oboru Petriho sietí a z oboru VHDL kódu. Pri probléme Petriho sieti ako aj problému VHDL kódu sme vchádzali z riešenia už existujúcich projektov. Naším cie om je hlavne ich vylepšenie a zintegrovanie týchto viacerých projektov do jednej aplikácie. Systém má napomáha pri vytváraní a vyhodnocovaní otázok testovej formy, vytváraní zadaní Petriho sietí a vytvori o možno najlepšie vyhodnotenie týchto zadaní. alej náš tím vychádza z programu „Chy aposunovka“, pri ktorom sa dop ajú asti kódu a následne po odovzdaní ho systém vyhodnocuje. Táto as je zameraná na problém VHDL kódov. Naším hlavným cie om je vylepši tieto už existujúce riešenie, doladi , prípadne opravi nefungujúce asti a pokúsi sa zhrnú tieto asti do jednej aplikácie. Predpokladanú interakciu systému medzi u ite mi a žiakmi môžeme vidie na nasledujúcom obrázku.



## Obr. 6 Interakcia v systéme

Z obrázka môžeme vidieť jednotlivú interakciu. Systém by mal pracovať tak, že učiteli si budú môcť vytvoriť zadania, následne budú môcť určiť termín testu. Na to ho žiak vyplní a odovzdá. Systém to vyhodnotí a odovzdá výsledky žiakovi a učiteli.

## Privilégia pre entitu Učiteľ

### Správa testov

Táto funkcia zahŕňa všetky možnosti, ktoré sa týkajú testov. Obsahuje všetky dostupné funkcie a to vytvorenie, mazanie a modifikáciu testov. Príklady môžu byť tvorené buď len pre jeden problém ako napríklad Petriho siete, alebo všeobecný test v rámci celého predmetu.

### Správa študentov

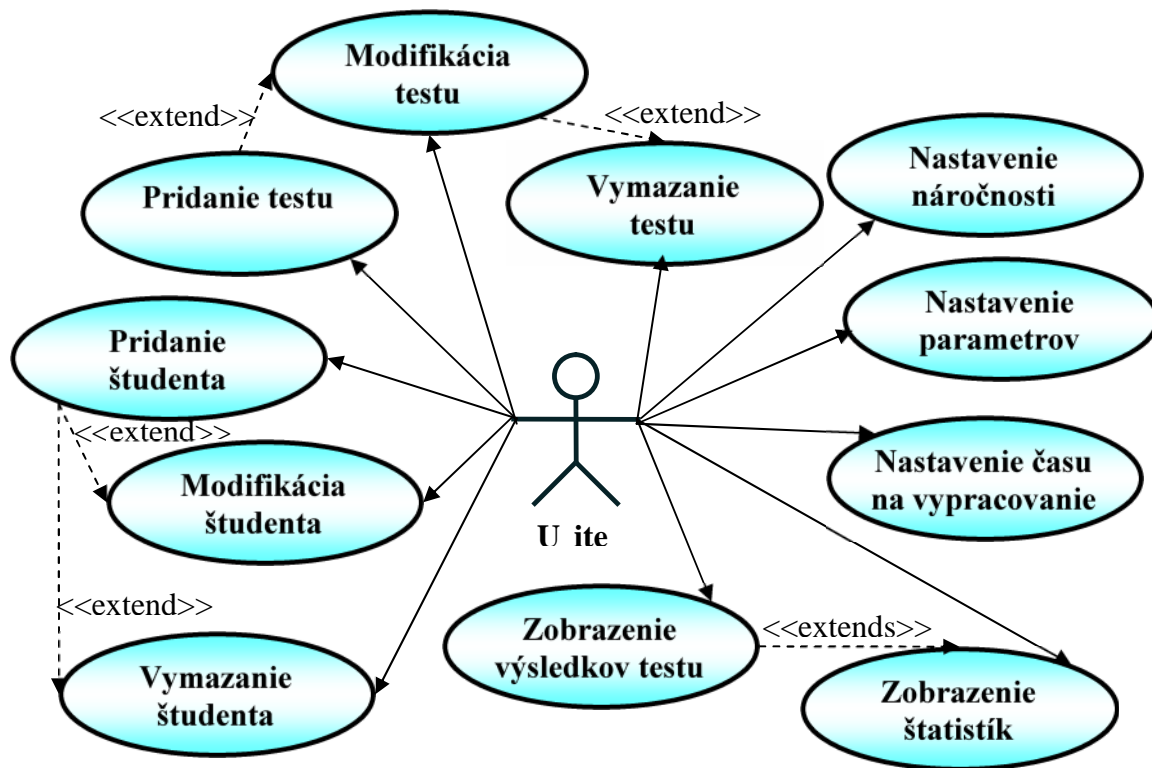
Táto funkcia slúži na vytvorenie úloh pre jednotlivých študentov, ich vytvorenie, mazanie, úprava, pridelenie jednotlivých študentov do skupín a podobne.

### Nastavenie parametrov testovania

Učiteľ bude mať možnosť určiť a nastaviť spôsob vyberania testov. Bude môcť sprístupniť prípadne zablokovať testy pre vybranú skupinu študentov. Taktiež by systém mal obsahovať možnosť generovania testov podľa oboru, alebo podľa náročnosti. V systéme by taktiež nemalo chýbať stanovenie časových limitov pre test.

### Zobrazenie výsledkov testov a štatistika

Učiteľ bude mať právo zobraziť informácie o celkovej ako aj o úspešnosti každého z jednotlivcov. Učiteľ by mal mať taktiež právo modifikovať a upravovať výsledky testov ako aj nazrieť do vyriešených príkladov.



Obr. .7 Prípady použitia pre používateľa a U ite

## Priviléžia pre entitu Študent

### Výber a vypracovanie príkladu

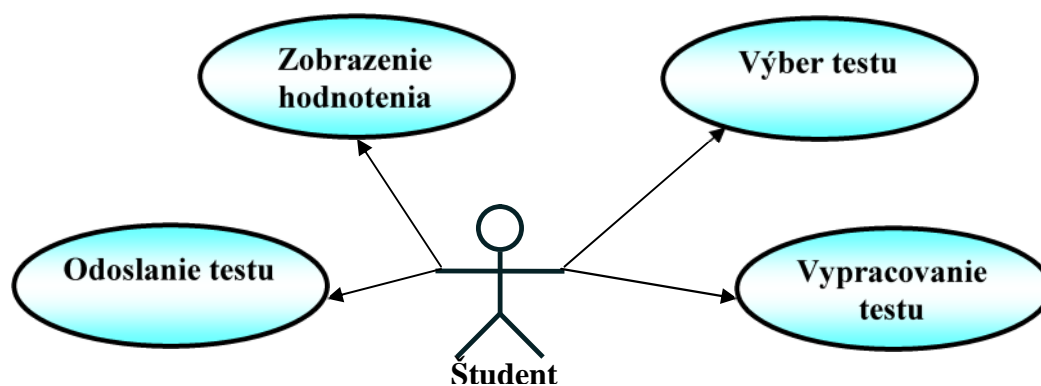
Na vypracovanie u ite povolí vyberanie testov, žiak si bude môc vybra test z danej problematiky, ktorú považuje za lepšiu vo bu, prípadne si bude môc vybra z dvoch podobných testov. Po za atí testovania systém ponúkne študentovi test na vyplnenie, ktorý obsahuje viacero zadaní rôznej náro nosti. Každý študent dostane jedno z niekoľkých zadaní patriacich do jednej skupiny náro nosti oho cie om je zabráni študentom opisovaniu pri skúšaní. K dispozícii bude ma študent informáciu o asovom limite.

## Odovzdanie testu

Študent má možnosť kedykoľvek odovzdať test. V prípade vypršania časového limitu sa test odovzdá automaticky.

## Zobrazenie výsledkov a hodnotenia

Po odovzdaní testu a jeho vyhodnotení systémom bude študent informovaný o bodovom ohodnotení svojho testu. Nebude mať však prístup k informáciám o hodnotení testov ostatných študentov. V prípade, že užívateľ nastaví funkciu zobrazovania a porovnávania správnych riešení, študent bude mať právo pozrieť si kde urobil chybu a správne riešenie, v prípade že nezíska maximálny počet bodov.



Obr. 8 Prípady použitia pre používateľa ŠtudentHrubý návrh

Táto kapitola sa podrobnejšie zaoberá návrhom modulov na preskúšanie praktických vedomostí.

## Rozšírenie modulu Test

Tento modul slúži na vytváranie a správu nových testov. Je súčasťou jadra vývojového systému Moodle.

Má veľa možností pridávania vytvorených testov študentom. Ten, kto test vytvára, môže vyhraďovať čas na riešenie v hodinách alebo v dňoch. Zadá sa okamih spustenia testu, o

znamená čas, kedy bude test sprístupnený pre študentov a môže zadať konečný čas, dokedy si študenti daný test môžu otvoriť.

Učiteľ môže test zostaviť tak, že tam bude len jedna úloha, alebo viacero otázok. Má na výber z veľkej a rôznych typov otázok:

- výpočtová
- vpísanie textu podľa inštrukcií
- porovnávací
- výber z možností / viac výberový
- áno/nie

Tieto typy otázok nie sú pre náš projekt kľúčové. Vytvoríme dva nové, ktoré implementujeme do modulu Test. Tieto nové typy budú z oboru VHDL programov a Petriho sietí. Pri ich implementácii budeme vychádzať z už existujúcich projektov a ich aplikácií:

#### 1. Chyba posunovka

Budeme vychádzať z projektu **Podpora vzdelávania v predmete Špecifické a opisné jazyky**, ktorý bol vypracovaný tímom PSS4 v roku 2008. V tejto práci sa zaoberali vytvorením nového modulu Chyba posunovka, ktorý slúžil podobne ako modul Zadanie. Pri ich module však vložili funkcionality a rozhrania pre ľahké vytvorenie úlohy pre VHDL kód. My zoberieme zdrojové kódy tohto modulu a prispôbíme ho tak, aby sme vytvorili nový modul, ktorý zaradíme medzi moduly typov otázok, ktoré ovláda modul Test.

#### 2. Petriho siete

Tu budeme vychádzať z dvoch prác: **Podpora vzdelávania v predmete Špecifické a opisné jazyky**, vypracovaný tímom PSS3 a tímom PSS4 z roku 2008.

Tím **PSS3** vypracoval samostatný program v C++ na riešenie a testovanie Petriho sietí. Z tohto chceme použiť algoritmus na porovnávanie Petriho sietí - vyhodnotenie samotných otázok typu Petriho sieť. V projekte tímu **PSS4** bola vytvorená Java aplikácia na kreslenie Petriho sietí. Túto prepracujeme na Java Applet a doplníme algoritmus na porovnávanie Petriho sietí z projektu tímu PSS3. Študent tak po časovom teste dostane zadanie úlohy kde bude musieť vytvoriť Petriho sieť, ktorá bude spĺňať požiadavky podľa zadania. Túto

vytvorenú sie odovzdá, Applet sie vyhodnotí a urí i je navrhnutá správne alebo nie a výsledok odošle do databázy.

Pridaním týchto dvoch otázok do modulu Test sa zna ne rozšíria možnosti používania systému Moodle pre predmet Opis digitálnych systémov. Otázky bude možné kombinova s klasickými testovými otázkami a takto bude možné vytvori semestrálne zadania (ak u ite nastaví požadovaný as na riešenie nieko ko týžd ov), priebežný, alebo závere ný test (skúšku).



# Návrh

---

Ako už bolo spomínané naša aplikácia vychádza z už existujúcich riešení, ktoré sú založené na type klient-server. Databáza bude taktiež uložená na serveri. Databáza obsahuje všetky informácie o používateľoch, testoch a ostatné dôležité informácie. Prístup ku všetkým informáciám, má iba správca systému. Pri návrhu tohto systému, náš tím plánuje používať šifrovacie algoritmy, aby sa zabezpečila väčšia bezpečnosť aplikácie. Samotná komunikácia prebieha tak, že študent zadá svoje prihlasovacie údaje, ktoré získava od administrátora systému. Tie zadá do internetového prehliadača, po odoslaní sa pošlú na server, ten tieto vstupné dáta spracuje a ak vyhodnotí ich autentifikáciu ako správnu, umožní študentovi prihlásiť sa do systému. Ten mu následne umožní získať prístup k vypracovaniu testov, ktoré mu prideli vyučujúci pedagóg.

## Vyhodnotenie otázky typu Petriho siete

V prípade typu otázky vytvorenia Petriho siete študent v Java aplete vytvára Petriho sieť na základe požiadaviek v znení otázky. Toto znenie zadáva učiteľ pri tvorbe testu. Učiteľ tiež túto sieť vytvorí. Keď študent dokončí tvorbu siete tak klikne na tlačítko Odošli. Vtedy applet vyhodnotí zhodnosť študentovej siete s tou, ktorú vytvoril skúšajúci.

Vyhodnocovanie je spravené porovnávaním matic, ktoré reprezentujú porovnávané Petriho siete. Pri porovnávaní sa použijú tri matice:

1. incidenčná matica
2. matica kapacít miest
3. matica inicializačných hodnôt miest

### Incidenčná matica

Jej rozmery sú  $m \times n$ , kde  $m$  (počet sčítaných) je počet prechodov a  $n$  je počet miest danej Petriho siete. Postupne sa prechádza celá Petriho sieť a hľadajú sa najprv hrany  $P \rightarrow T$  (miesto  $P$  -> prechod) pričom ak máme hranu medzi miestom číslo 5 a prechodom číslo 8 s váhou 3 tak sa do incidencej matice v stĺpci 8 a riadku 5 nastaví hodnota -3. Ďalej sa hľadajú hrany typu

T->P (prechod -> miesto) no narozdiel od P->T sa teraz pripočítava kladná hodnota (nie záporná ako v predchádzajúcom príklade).

### **Matica inicializačných hodnôt**

Jednorozmerná matica (vektor) s veľkosťou  $n$  je počet miest v sieti. Obsahuje inicializačné hodnoty miest.

### **Matica kapacít miest**

Takisto je to jednorozmerná matica s veľkosťou  $n$  no obsahuje kapacity jednotlivých miest.

### **Algoritmus**

Porovnanie za nezáistením, i incidentné matice porovnávajúcich petriho sietí majú rovnaký rozmer. Ak nie tak nemajú tieto dve siete rovnaký počet miest a prechodov – vyhlásime siete za nezhodné.

Potom nasleduje porovnanie P->T/T->P hrán. Toto sa vykonáva v niekoľkých pokusoch - čím viac tým sú siete menej pravdepodobne zhodné. Najprv sa porovnávajú matice so všetkými parametrami (incidentné matice, inicializačné hodnoty, kapacity miest) a postupne až k porovnaniu len incidentných matíc. Algoritmus porovnávania týchto matíc v princípe hľadá možnosti preusporiadania jednej matice tak aby sme dostali druhú maticu – tým pádom sú siete zhodné len sú trocha inak navrhnuté, resp. Nakreslené v inom poradí ich objektov.

Informácia o úspešnosti sa uloží do JavaScriptu a neskôr pri odovzdaní celého testu aj do databázy spolu so štruktúrou petriho siete nakreslenej študentom pre neskoršiu kontrolu alebo konzultácie.

**TÍMOVÝ PROJEKT II.**  
**Podpora vzdelávania v predmete Opis**  
**digitálnych systémov**  
**(LETNÝ SEMESTER)**

---

# Návrh

---

## Architektúra systému

[6]Pri realizácii návrhu náš tím vychádzal z architektúry typu klient-server. Model klient-server rozlišuje dva systémy. A to systém klienta a systémov servera, ktoré komunikujú cez počítačovú sieť. Pri klient-server architektúre je používaný systém, ktorý sa skladá zo softvéru klienta a zo softvéru servera. Softvér alebo proces klienta môže iniciovať komunikačné spojenie, zatiaľ čo server čaká na požiadavku od klienta. Server v Internete je vždy aktívny, nedáva však o sebe vedieť a čaká na požiadavky od klienta. Napríklad server, ktorý spravuje multimediálne a hypertextové dokumenty. Taktiež má informácie, kde sa nachádzajú tieto informácie a aké majú vlastnosti. Na druhú stranu počítač je potom možné spustiť klienta (teda i na tom istom, kde je server) a vyžiadať si od servera nejakú službu, napríklad konkrétny dokument. Server od klienta požiadavku prijme a spracuje ju. Klienta zvyčajne aktivuje používateľ, ktorý vstupuje do siete Internet. Ten podľa vlastnej vôle zadáva požiadavky na server. V sieti je tento model vyhovujúcim spôsobom na efektívne prepojenie programov, ktoré sú distribuované po rôznych miestach v sieti.

Ďalej je v tejto kapitole opísané správanie sa nášho systému. V architektúre servera je opísaná štruktúra na strane servera. V architektúre klienta je opísané, ako je riešená grafická stránka našej aplikácie, pripojenie k databáze a bezpečnosť systému.

### Server

Všetky zdrojové súbory sú uložené na strane servera. Taktiež všetky dôležité záznamy o výsledkoch, menách užívateľov, ako aj prihlasovacích hesiach sú uložené na strane servera v databáze MySQL. Prístup k správe celej databázy má len administrátor, ktorý má prístupové údaje. Ostatní používatelia, majú prístup iba k obmedzenej správe a prístup iba k určitým údajom, ktoré im poskytne pridelené privilégium a grafické rozhranie aplikácie.

## Klient

Klientská aplikácia bude prídavný modul v aplikácií Moodle. Pri vývoji tohto systému boli použité technológie PHP, MySQL, javascript a JAVA. Grafické používateľské rozhranie sa skladá z viacerých strán. Hlavnou stránkou je tvorená štýlmi použitými v systéme Moodle. Aplikácia Petriho sietí je vytvorená ako java-applet.

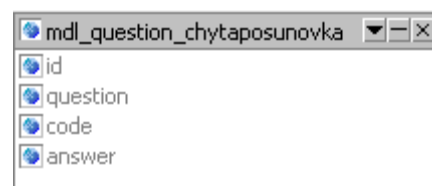
Konečný produkt bude webová aplikácia, ktorá je dostupná na serveri labbs2.fiit.stuba.sk. Všetky zdrojové súbory sú taktiež priložené aj na CD-médium priloženému k dokumentácii. To znamená že túto aplikáciu bude možné rozbehnúť aj lokálne, na každom počítači, na ktorom je nainštalovaný Apache, alebo jemu podobný server a taktiež MySQL server.

## Fyzický model údajov

Naša aplikácia bude obsahovať celkovo 231 tabuliek. Medzi tabuľkami sú uplatnené vzťahy, ktoré sú opísané v nasledujúcich podkapitolách. Rozpísané sú iba tie tabuľky, ktoré využil náš tím pri vývoji tejto aplikácie.

### MDL\_QUESTION\_CHYTAPOSUNOVKA

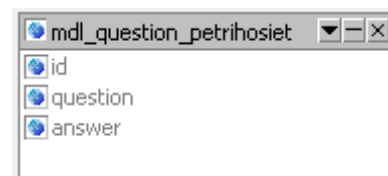
Tabuľka obsahuje informácie pre modul chyta posunovky. Má uložené informácie o špecifickom ID, čísle otázky, vkladanej stránke kódu a čísle odpovede.



Obr. 9.9 mdl\_question\_chytaaposunovka

### MDL\_QUESTION\_PETRIHOSIET

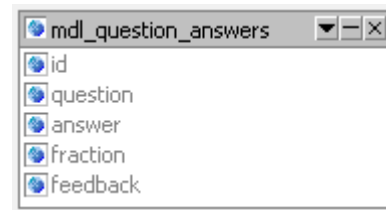
Tabuľka obsahuje informácie o špecifickom ID záznamu, vkladanej stránke kódu a čísle odpovede.



Obr. 9.10 mdl\_question\_petrihosiet

### MDL\_QUESTION\_ANSWERS

Tabu ka obsahuje informácie o špecifickom ID záznamu, vkladanej asti kódu a ísle odpovede, zlomku a spätnej väzbe



Obr. .11 mdl\_question\_answer

### MDL\_QUESTION\_ATTEMPTS

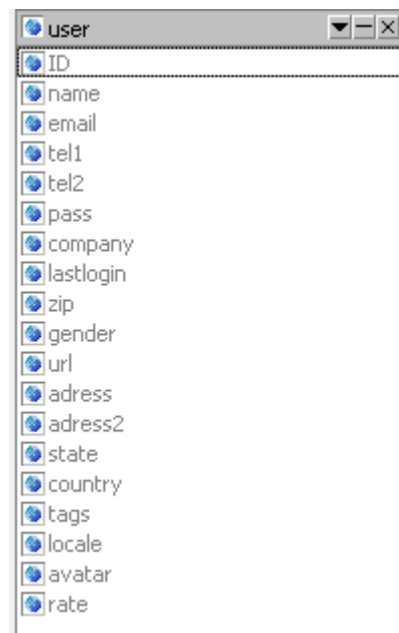
Táto entita obsahuje informácie o špecifickom ID záznamu a module, ktorý spája odpove so správou aplikáciou



Obr. .12 mdl\_question\_attempts

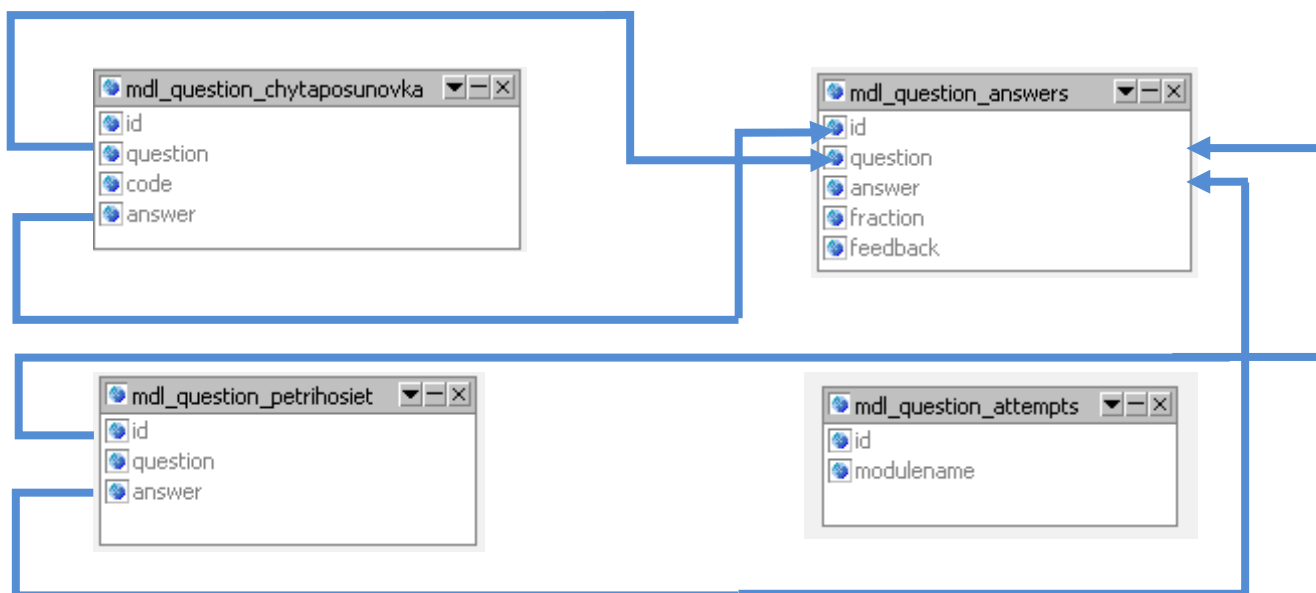
### USER

Tabu ka obsahuje informácie o špecifickom ID záznamu , mene používate a, jeho e-maile, výsledku testov, jeho prihlasovacie heslo do systému, alšími údajmi sú skupina, posledné prihlásenie, PS , pohlavie, medzi údajmi je taktiež ve a nepovinných položiek. Napríklad webová adresa používate a, Adresa bydliska, rodinný status, krajina, skratka rodnej krajiny, obrázok používate a, a hodnotenie. V našom systéme sa s niektorými položkami nepracuje a sú ur ené pre možný budúci vývoj tejto aplikácie.



Obr. .13 user

Celá aplikácia pracuje tak, že po vypracovaní testu „chy asposunovka“ a odoslani na vyhodnotenie sa uloží do tabu ky MDL\_QUESTION\_CHYTAPOSUNOVKA íslo otázky a íslo odpovede. alšou informáciou je meno študenta. Následne sa do záznamu ID a „question“ uloží do tabu ky MDL\_QUESTION\_ANSWERS íslo odpovede a otázky z tabu ky MDL\_QUESTION\_CHYTAPOSUNOVKA. Na takom istom princípe prebieha tok dát z tabu ky MDL\_QUESTION\_PETRIHOSIET. Túto interakciu môžeme názorne vidie na nasledujúcom obrázku.



Obr. .14 diagram údajov

**TÍMOVÝ PROJEKT I.**  
**Podpora vzdelávania v predmete Opis**  
**digitálnych systémov**  
**(ZIMNÝ SEMESTER)**

---



# Implementácia

---

Nami implementovaný projekt pre podporu vzdelávania v predmete Opis digitálnych systémov pozostáva z jednotlivých modulov, ktoré sú vyvíjané v jazyku Java a z aplikácie Moodle, ktorá je vyvíjaná pomocou webových technológií. Na konci zimného semestra sme zdokumentovali a odovzdali funkčný prototyp systému. Študent má možnosť vytvárať Petriho siete podľa zadania. V prototypu sú implementované tieto funkcie:

## Editor Petriho sietí

Editor Petriho sietí výtok funkcionality Java aplikácie PetriTester, ktorú sme prebrali z práce: Podpora vzdelávania v predmete Špecifické a opisné jazyky, vypracovanej tímom PSS4 2008. Z tejto aplikácie teda preberáme iba možnosti editovania a simulovania Petriho siete.

Keďže náš cieľový systém, Moodle, je webová aplikácia, bolo nutné upraviť aplikáciu PetriTester, aby sa dala zobraziť a používať v html stránke. Upravili sme ju na Java Applet. Úprava si vyžiadala nasledovné kroky:

1. analýza aplikácie, zistenie vstupných a výstupných bodov aplikácie
  - a. výsledok: vstupná trieda je `petritool.Main`
2. analýza aplikácie, komunikácia s aplikáciou, výstup a vstup dát z/do aplikácie
  - a. výsledok:
    - i. export Petriho siete je implementovaný iba do súboru na harddisk, čo pre naše potreby nie je dostačujúce, import je taktiež umožnený iba zo súboru, formát exportovanej Petriho siete je XML
    - ii. na komunikáciu s applet-om využijeme možnosť parametrov v html tagu `APPLET` a JavaScript
3. implementácia import a export funkcií
  - a. existujúca funkcionality bola upravená nasledovne: aby bolo možné exportovať z aplikácie pomocou volania JavaScriptovej funkcie z prehliadača: metóda `getNet()`, ktorá vracia hodnotu typu `String` a importovať pomocou parametra v html stránke: funkcia `setNet()`, ktorá na seba berie obsah parametra `netxml`, ktorý sa nachádza v tagu `APPLET` v html stránke.

- b. formát dát bol upravený, aby bolo možné sa vložiť do html stránky, je nutné XML, ktorým je sie reprezentovaná zakódova pomocou Base64 kódovania [ref. 6]
4. prerobenie triedy `petritool.Main`, aby bola zdedená z triedy `java.applet.Applet`, pridanie metódy `init()`, kde je implementované na ítanie Petriho siete pomocou metódy `setNet()`.

Ukážka php súboru, ktorý obsahuje Editor Petriho sietí sa nachádza v prílohe A.

## Prototyp

V tejto kapitole je opísaná implementácia najdôležitejších logických a funkčných blokov, ktoré tvoria aplikáciu. V prototypu nášho systému sa pokúšame otestovať a overiť nový modul do aplikácie Moodle. V ňom je obsiahnutá hlavná časť používateľského rozhrania umožňujúca vytváranie Petriho siete pomocou Java Applet-u, jej zobrazenia a simulácie.

Samotný prototyp predstavuje jednoduchý test, kde vystupuje učiteľ a študent. Učiteľ pripraví test s otázkou typu Petriho sieť a študent má za úlohu vyriešiť tento test. Má k dispozícii zadanie a Java Applet, ktorý slúži ako editor a simulátor Petriho siete. Po vyriešení si učiteľ bude môcť pozrieť študentove riešenie.

## asový plán na zimný semester

---

- ✓ vybranie témy, zhotovenie ponuka a prezentácie (vybratá téma Podpora vzdelávania v predmete Opis digitálnych systémov)
- ✓ zhotovenie informa nej stránky
- ✓ písanie zápisníc zo stretnutí
- ✓ analýza podobných prác
- ✓ na základe analýzy vybratie pre nás najvhodnejšej témy z predmetu Opis digitálnych systémov, pre ktorú budeme náš e-learningový systém tvori
- ✓ h bková analýza danej témy v dostupných prác, vybratie jednotlivých astí a modulov, ktoré môžeme použiť
- ✓ návrh možností a funkcionalít, ktoré bude výsledný e-learningový modul obsahova
- ✓ vybratie programovacích jazykov, ktoré budeme používa
- ✓ návrh možností a funkcionalít, ktoré bude výsledný e-learningový modul obsahova
- ✓ špecifikácia požiadaviek a hrubý návrh riešenia
- ✓ vytvorenie dokumentácie analýzy problému, špecifikácie požiadaviek a hrubý návrh
- ✓ implementácia prototypu vybraných astí

Legenda:      × - nesplnené

                 v - splnené

# Zhodnotenie za zimný semester

---

Výsledkom tohto dokumentu, je zanalyzovaná, navrhnutá a v štádiu implementácie tvorby aplikácie na preverovanie teoretických a praktických znalostí v modelovaní, simulácii a analýze Petriho sietí. Dokument je len prvou časťou výsledného dokumentu, ktorý bude výsledkom finálnej práce nášho tímu na predmete Tímový projekt II. Dokument bude výsledkom práce piatich študentov za dva semestre inžinierskeho štúdia.

V úvode dokumentu sú spracované a zanalyzované existujúce riešenia. Tie boli neskôr rozdiskutované počas stretnutí členov nášho tímu. Náš tím sa zamerával na už zdokumentované myšlienky a algoritmy, ktoré považoval za najvýhodnejšie v našom projekte. Počas zimného semestra sme sa snažili zamerať na moduly ktoré ešte neboli dostatočne sfunkčnené. Zamerali sme sa hlavne na aplikáciu Petriho sietí.

V ďalšej časti dokumentu boli špecifikované požiadavky systému a vytvorený hrubý návrh. Bol navrhnutý model prípadov použitia pre daný systém.

Ďalšia časť dokumentu bude riešená počas letného semestra, ako práca na predmete Tímový projekt II. Táto časť bude obsahovať Implementáciu, testovanie, prílohy a finálnu používateľskú príručku k danému produktu.

Táto dokumentácia slúži ako východiskový dokument pre vytvorenie funkčnej aplikácie pre podporu vzdelávania na predmete Opis digitálnych systémov.

**TÍMOVÝ PROJEKT II.**  
**Podpora vzdelávania v predmete Opis**  
**digitálnych systémov**  
**(LETNÝ SEMESTER)**

---

# Implementácia

---

Nami vyvíjaný projekt na automatické vyhodnocovanie vedomostí a praktických zručností študentov v oblasti Petriho sietí je implementovaný technológiou PHP, MySQL, HTML a JAVA. Pri vývoji bola použitá tieto verzie: PHP 5.2.6, MySQL 5.0.51a, Moodle 1.9.6. Pri aplikácii Petriho siete viete používať pomocou myšky modelovať vlastnú sieť a túto následne odsimulovať. alej po vypracovaní danej úlohy môže používať danú úlohu odovzdať na automatické hodnotenie. V ďalších kapitolách je rozvinutý aj spôsob implementácie aplikácie „chybových“, ktorá taktiež slúži na testovanie znalostí študentov.

## Implementácia Petriho siete

### Zmena nasadzovania appletu

Kvôli pretrvávajúcim problémom s chodom appletu korektne vo viacerých prehliadačoch, sme zmenili spôsob akým integrujeme applet do HTML stránky. Pomocou technológie [8]Java Network Launch Protocol sme docielili rovnaké správanie appletu na všetkých prehliadačoch.

### Vylepšenie vytvárania Petriho siete

Spôsob akým sa vytvárali hrany v aplikácii PetriDesigner považujeme za nedostatočný, rozhodli sme sa preto implementovať iný. Podľa starého bolo nutné označiť vrchol a prechod a stlačiť tlačidlo spojenia, my sme vytvorili spôsob "potiahni a pusti", ktorým sa hrany vytvárajú oveľa pohodlnejšie.

### Zapracovanie funkcionality vyhodnotenia siete

Do aplikácie PetriDesigner sme zapracovali porovnávací algoritmus, ako je uvedené v kapitole Návrh. Keďže algoritmus je napísaný v jazyku C++ a Applety sú programované v jazyku Java bolo nutné algoritmus prepísať, čo nebola až taká ťažká úloha. Po naštudovaní princípu fungovania algoritmu a syntaktickej úprave do jazyka Java sme vykonali "black box" testovanie (viac v kapitole Testovanie \*este bude pridané). Navyše tento algoritmus je stále

vo vývoji, iže bolo nutné zahrnúť aj viaceré zmeny, ktoré nám dodal autor tesne pred ukončením práce na tomto projekte.

### Vyhodnotenie siete

Applet funguje v dvoch, módoch:

1. keď robí študent vypracovanie úlohy (kontrola siete sa nerobí)
2. keď si u siete prezerá výsledky testu, vtedy sa appletu nastavujú parametre:
  - a. `doNetCorrection=true`
  - b. `netxml=Base64` zakódované XML so sieťou, ktorú narkeslil študent
  - c. `baseNetXML= Base64` zakódované XML so sieťou, ktorú zadal u siete ako referenčnú sieť

V druhom móde sa porovnanie sietí vykoná hneď po štarte appletu. U siete ovi sa zobrazí okno s percentuálnou zhodou študentovej siete s referenčnou sieťou zadanou pri vytváraní testu. Spolu s týmto sa zobrazia aj vlastnosti jeho siete (dopísať aké). Po prezrení a opravení testu u siete om je možné výsledky porovnania sietí uložiť do databázy, výsledky z porovnania je možné z appletu dostať volaním metódy `getResults()` z JavaScriptu v HTML stránke.

Zlepšenie: v druhom móde je nutné poslať appletom študentovu sieť dva krát, toto sa nemuselo diať, ak by applety medzi sebou komunikovali.

Na implementáciu tohto algoritmu bol potrebné implementovať niečo ako maticových operácií, za týmto účelom sme rozšírili triedu z voľne dostupného balíka JAMA: `Jama.Matrix`

## Implementácia „chyby aposunovky“

Modul chyby aposunovka je naimplementovaný technológiou Question v systéme Moodle. Jeho hlavné súbory sú:

- `edit_chybyaposunovka_form.php`
- `questiontype.php`

### **edit\_chyaposunovka\_form.php**

Tento súbor obsahuje triedu *question\_edit\_chyaposunovka\_form*, ktorá je dedičom všeobecnej triedy systému Moodle – *question\_edit\_form*.

Táto trieda obsahuje niekoľko metód, ktoré systém Moodle potrebuje pre modul Question:

- *definition\_inner(&\$mform)* – definuje štruktúru formulára pre vytváranie novej otázky. Ten obsahuje input pre upload súboru so zadaním vo formáte VHDL s predpripravenými tagmi označujúce miesta, ktoré študent bude vyberať a umiestňovať na správne miesta
- *set\_data(\$question)*
- *qtype()* - názov typu Question

### **questiontype.php**

Aj tu máme triedu (*question\_chyaposunovka\_qtype*), ktorá je dedená zo všeobecnej triedy systému Moodle – *default\_questiontype*.

Trieda obsahuje niekoľko metód:

- *name()* - vráti názov typu otázky
- *save\_question\_options(\$question)* – táto metóda sa stará o uloženie otázky do databázy. Uploadnutý súbor sa nahradí do premennej a s použitím časti zdrojových kódov od tímu 1.4 z roku(2007/2008) sa vyhodávajú v súbore tagy vyznačujúce miesta zdrojového kódu VHDL, ktoré budú študentovi ponúknuté pri riešení. Do databázy sa ukladajú riešenia ako aj VHDL kód (použitá je tabuľka *mdl\_question\_chyaposunovka*).
- *print\_question\_formulation\_and\_controls(&\$question, &\$state, \$cmoptions, \$options)* – metóda sa stará o zobrazenie otázky typu „chyaposunovka“ v prípade zobrazenia pri vypínaní ako aj v prípade zobrazenia výsledku otázky.
- *grade\_responses(&\$question, &\$state, \$cmoptions)* -



# Testovanie

---

V tejto kapitole je opísaná metodika testovania naimplementovanej aplikácie. Pri takto rozsiahlom systéme je nutné venovať dostatočne dlhý čas testovaniu a pokryť testovaním všetky možné situácie, ktoré by mohli nastať pri behu tohto programu a pri zadávaní rôznych vstupov od používateľov danej aplikácie. Keďže táto aplikácia má slúžiť na hodnotenie študentov, musí byť zabezpečená a overená konzistentnosť celého systému.

Otestované boli dve rôzne časti a to dátová a funkčná časť. Pri dátovej časti bolo overované, či údaje sú správne zapísané v databáze. Pri funkčnej časti bolo overované, či dané údaje sa správajú podľa nami špecifikovaných vlastností.

Testovanie bolo uskutočnené jednak počas vývoja aplikácie, po jednotlivých častiach, ale aj po dokončení celého projektu. Testovanie prebiehalo na počítačoch vybavených rôznymi verziami operačného systému MSWindows a na počítačoch s rôznymi hardvérovými konfiguráciami. Aplikácia bola testovaná na systémoch: MS Windows XP, Vista, 7 a to pri 32bit. architektúre. Funkčná časť systému sa testovala na webových prehliadačoch: Mozilla Firefox, Google Chrome a Internet Explorer, všetko doteraz v najnovšej verzii. Počas testovania bolo odhalené veľké množstvo nedostatkov, ktoré boli následne odstránené a opravené na požadovanú úroveň. Ďalším testovaním sme overili stabilitu a základnú bezpečnosť našej aplikácie a jej pripravenosť na ďalšie vyvíjanie.

## Výsledky testovania

Testovanie nám odhalilo pomerne veľa chýb, čo je pre nás pozitívny výsledok, pretože sme mohli zistené nedostatky opraviť a skvalitniť tak úroveň systému.

# Možné vylepšenia

---

Aplikácia je funkčná a otestovaná. alej je možné do aplikácie pridať neobmedzené množstvo funkcií a vylepšení.

## Návrhy na vylepšenie

### Funkcionalita

- Pridanie ďalších funkcií pre rozšírenie automatizácie
- Zvýšenie bezpečnosti pridaním ďalších bezpečnostných protokolov
- Zefektívnenie algoritmov pre rýchlejšie spracovanie

### Ovládanie aplikácie

- Viac zjednodušiť ovládanie. Prispôbiť ho tak, aby bolo ešte viac intuitívne.
- Pridať viac ovládacích tlačítkov.

### Grafické rozhranie

- alej možno rozvíjať grafické rozhranie, prispôbovať ho požiadavkám a nastaviť ho tak, aby táto aplikácia bola ešte viac atraktívnejšia.

# Inštalácia

---

Aplikácia je upgrade do open-source systému Moodle. Najprv sa nainštaluje aplikácia moodle, pod a inštaluje sa príručka systému Moodle. Všetky potrebné informácie je možné nájsť na internetovej adrese: <http://www.moodle.sk/> alebo <http://www.moodle.org>. Vyžadovaná je verzia čísla 1.9.8 a vyššie.

Po úspešnom nainštalovaní modulu sa do domovského adresára nakopírujú adresáre *petrisiet* a *chytaposunovka* do adresára moodle: *moodle/question/type/*, ktoré sú uložené na CD médiu priloženom k tejto dokumentácii. Alej je potrebné skopírovať adresár *geshi* (prevzatý od tímu 4 zo š.r. (2007/2008), ktorý zabezpečuje zobrazovanie VHDL kódu) do *moodle/filter/*.

Pre potreby petriho sietí je potrebné postupovať takto:

1. Do spoločného adresára nakopírovať súbory:
  - a. PetriTool.jar
  - b. petritool.jnlp
  - c. lib\Jama-1.0.2.jar
  - d. lib\jgraph.jar
2. Vzorové použitie je taktiež súčasťou inštalovateľného balíka:
  - a. súbor applet.php

alej sa pomocou systému Moodle nainštalujú tieto moduly kde sa automaticky aj vytvoria potrebné tabuľky v databáze.

## Offline mód

Pre naštartovanie aplikácie v offline móde je potrebné nainštalovať aplikáciu, ktorá dokáže vytvoriť virtuálny lokálny server. Napríklad aplikácia XAMPP, ktorá je dostupná na internete <http://www.apachefriends.org/en/xampp-windows.html>.

## Online mód

Spustenie aplikácie v online móde vyžaduje mať k dispozícii MySQL databázu s danou štruktúrou tabuliek. SQL. Inštalácia je takmer identická s inštaláciou v offline móde, s tým rozdielom, že pre on-line mód, je potrebné mať k dispozícii webový server s podporou aspoň PHP 5.0.

## Záver

---

Tento projekt je výsledkom práce na predmetoch Tímový projekt I. a Tímový projekt II. V zimnom semestri sa náš tím zamerlal hlavne na tvorbu analýzy, hrubého návrhu a tvorby prototypu Petriho sietí na preverovanie teoretických a praktických znalostí v modelovaní, a simulácii Petriho sietí. Za úlohu do letného semestra si náš tím zadal vytvoriť plne funkčný modul Petriho sietí a taktiež pridať modul „Chybový systém“ do aplikácie Moodle. Všetky potrebné informácie z ktorých sme vychádzali sú uvedené v analýze tejto dokumentácie. ale bolo v dokumentácii podrobne opísaný návrh a implementácia prototypu. V letnom semestri náš tím pokračoval v konkrétnom návrhu a podrobne zdokumentovanej implementácii. Obidva moduly, ktoré sme si zobrali za úlohu vytvoriť a spojiť sa pri finálnom testovaní nám overili ako funkčné. Túto úlohu považujeme za splnenú. Tento systém môže slúžiť na ďalšie vyvíjanie a zdokonaľovanie a to v oblasti bezpečnosti a v oblasti podrobného prispôbovania k aktuálnym osnovám predmetu Špecifické a opisné jazyky. V budúcnosti môže byť táto aplikácia nasadená do reálnej prevádzky pri podpore vzdelávania a testovania teoretických a praktických vedomostí študentov.

## Zdroje

---

1. Matej Mayer, Bc., Podpora dištan ného vzdelávania v predmete Opis digitálnych systémov, Diplomová práca, FIIT STU Bratislava, máj 2009
2. Andrej Letkovský, Bc., Podpora dištan ného vzdelávania v predmete Opis digitálnych systémov, Diplomová práca, FIIT STU Bratislava, máj 2009
3. Izsák Peter, Automatické vyhodnocovanie programov vo VHDL, Závere ný projekt, FEI STU Bratislava, máj 2000
4. Tím 3PSS, Podpora vzdelávania v predmete Špecifika né a opisné jazyky, Tímový projekt, FIIT STU Bratislava, máj 2008
5. Tím 4PSS, Podpora vzdelávania v predmete Špecifika né a opisné jazyky, Tímový projekt, FIIT STU Bratislava, máj 2008
6. Špecifikácia BASE64 kódovania, The Base16, Base32, and Base64 Data Encodings, S. Josefsson, Network Working Group, október 2006, <http://tools.ietf.org/html/rfc4648>
7. Klient – Server Dostupné na internete: <http://www.efirma.sk/klient-server.htm>, 22. Apríl 2010
8. Java™ Rich Internet Applications Deployment Advice, [http://java.sun.com/javase/6/docs/technotes/guides/jweb/deployment\\_advice.html#usingJnlp](http://java.sun.com/javase/6/docs/technotes/guides/jweb/deployment_advice.html#usingJnlp), 22. Apríl 2010

**TÍMOVÝ PROJEKT I.**  
**Podpora vzdelávania v predmete Opis**  
**digitálnych systémov**  
**(ZIMNÝ SEMESTER)**

---

# Prílohy

## Príloha A - Ukážka HTML kódu s Java Appletom Petrinet editora

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<SCRIPT>
function storePNxmlInHiddenField()
{
    var applt = document.getElementById('myApplet');
    if (applt)
    {
        var field = document.getElementById('pnxmlfield');
        field.value=applt.getNet();
    }
    else
    {
        alert('No applet found on page!');
        field.value='Error. No data!';
    }
}
</SCRIPT>
</head>
<body>

<applet id="myApplet"
    code="petritool.Main.class"
codebase="http://labss2.fiit.stuba.sk/TeamProject/2009/team09pss/applet/pn"
width="500" height="500"
archive="./lib/jgraph.jar"
archive="./lib/Jama-1.0.2.jar"
>
    <param name="netxml" value="<?php echo $_POST['pnxml']; ?>" />
    If you see this, you dont have java plugin installed, please install
proper plugin.
</applet>

<form action="applet.php" method="post" >
    <input type="submit" value="Poslat"
onClick="storePNxmlInHiddenField();" />
    <input type="hidden" name="pnxml" id="pnxmlfield" />
</form>
```

This is petrinet saved as xml and base64 encoded:

```
<?php
    echo $_POST['pnxml'];
?>

</body>
</html>
```



## Príloha B - Ukážka XMLDB dátového modelu pre otázku typu Petriho sie

```
<?xml version="1.0" encoding="UTF-8" ?>
<XMLDB PATH="question/type/petrihosiet/db" VERSION="20060812"
COMMENT="XMLDB file for Moodle question/type/petrihosiet">
  <TABLES>
    <TABLE NAME="question_petrihosiet" COMMENT="Options for petrihosiet
questions">
      <FIELDS>
        <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true"
UNSIGNED="true" SEQUENCE="true" ENUM="false" NEXT="question"/>
        <FIELD NAME="question" TYPE="int" LENGTH="10"
NOTNULL="true" UNSIGNED="true" DEFAULT="0" SEQUENCE="false" ENUM="false"
PREVIOUS="id" NEXT="answer"/>
        <FIELD NAME="answer" TYPE="int" LENGTH="10" NOTNULL="true"
UNSIGNED="true" DEFAULT="0" SEQUENCE="false" ENUM="false"
PREVIOUS="question" NEXT="tolerance"/>
      </FIELDS>
      <KEYS>
        <KEY NAME="primary" TYPE="primary" FIELDS="id" COMMENT="Primary key
for question_petrihosiet" NEXT="question"/>
        <KEY NAME="question" TYPE="foreign" FIELDS="question"
REFTABLE="questions" REFFIELDS="id" PREVIOUS="primary"/>
      </KEYS>
      <INDEXES>
        <INDEX NAME="answer" UNIQUE="false" FIELDS="answer"/>
      </INDEXES>
    </TABLE>
  </TABLES>
</XMLDB>
```

## Príloha C - Používateľská príručka prototypu

Používateľská príručka je zložená z troch častí, každá pre konkrétnu používateľskú rolu. Prvá časť je určená pre administrátora a popisuje postup inštalácie a nastavenie systému. Druhá časť je pre učiteľa a tretia časť sa venuje študentovi.

### Časť pre Administrátora (inštalácia príručky)

Ako prvý krok je potrebné nainštalovať Moodle, ktorý sa nachádza na CD pod názvom moodle.zip. Po nainštalovaní Moodle je potrebné do adresára /question/type/ nakopírovať modul petrihosiet. Ten sa nachádza na CD pod názvom petrihosiet.zip. alej už len nainštalovať Moodle podľa inštrukcií v jednotlivých krokoch.

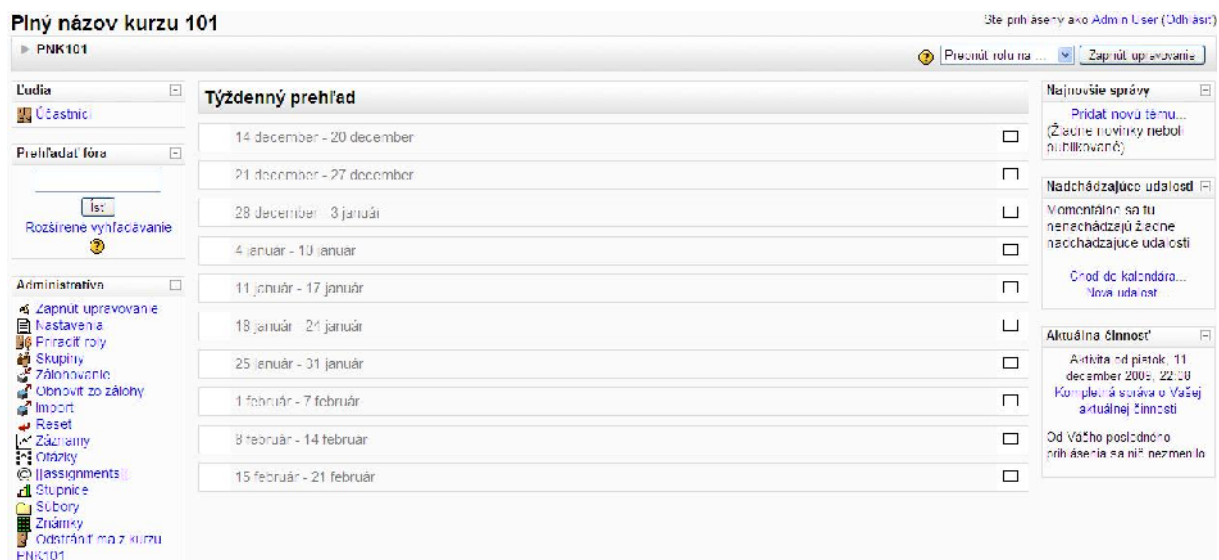
Inštalácia Petritester appletu:

1. rozbalenie archívu PetriTesterApplet.zip do adresára *petritesterapplet* na serveri
  - a. editovanie súboru <moodle\_root>/question/type/petrihosiet/display.html
  - b. nahradenie atribútu codebase vo všetkých APPPLET tagoch za:
 

```
codebase="http://servername/petritesterapplet_directory"
```

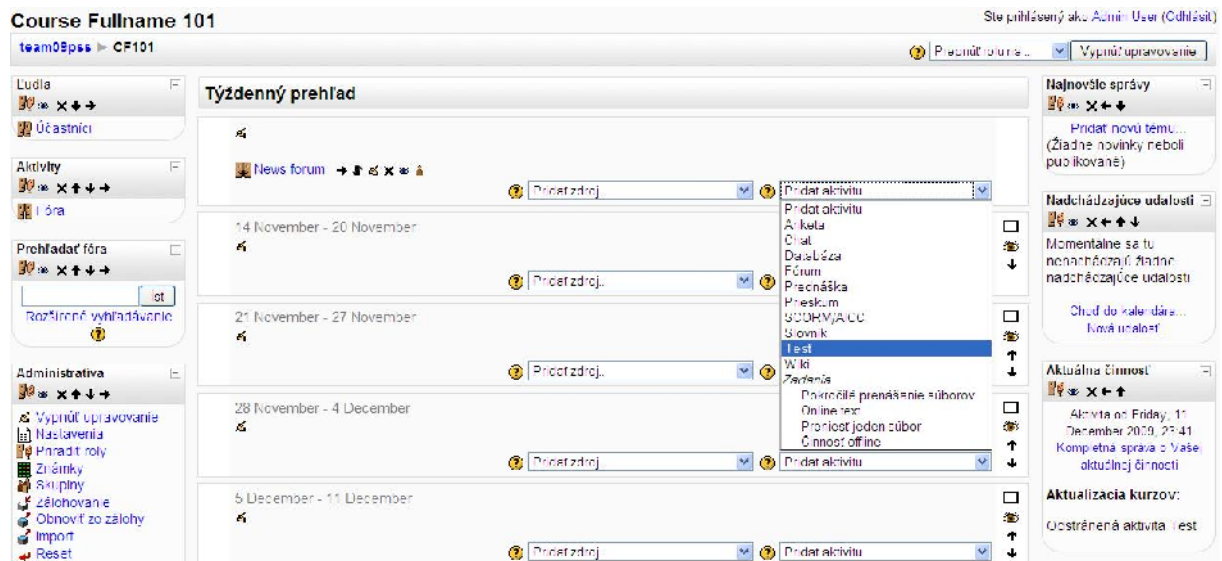
### Časť pre učiteľa

Konto pre učiteľa môže vytvoriť len administrátor stránky. Po prihlásení do systému je v hlavnej časti stránky nachádza tlačidlo pre vytvorenie nového kurzu, ak ešte žiaden kurz nie je vytvorený. Na ťa sa nám stránka ako na Obrázku C-1.



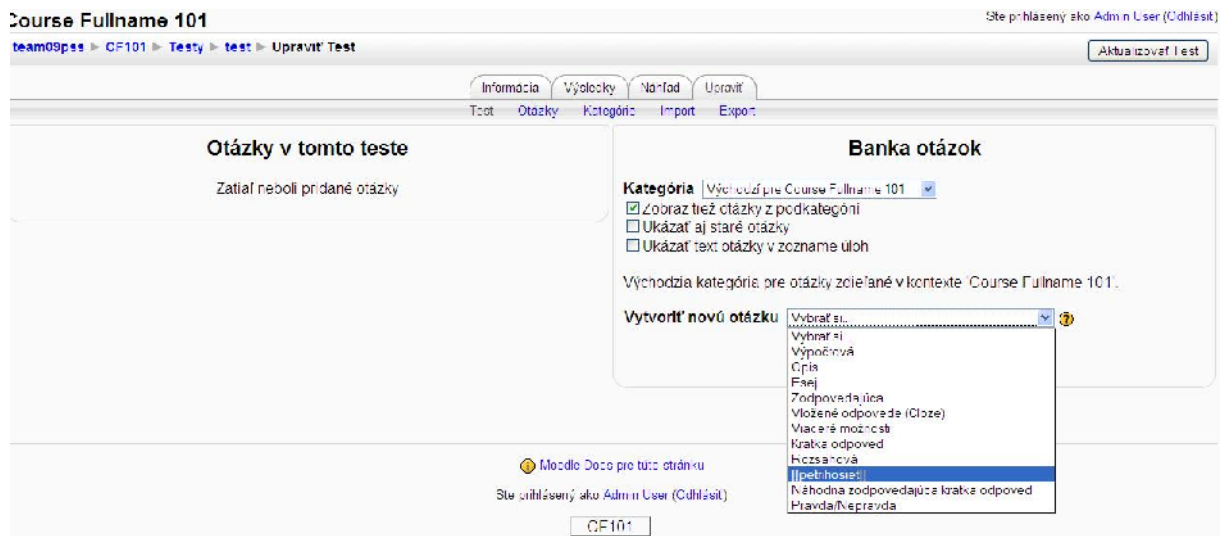
Obrázok C-1 Úvodná obrazovka systému Moodle.

Potom je potrebné prepnúť sa do editovateľného režimu. V tomto režime sa nám objavia ďalšie možnosti.



Obrázok C-2 Týždenný prehľad

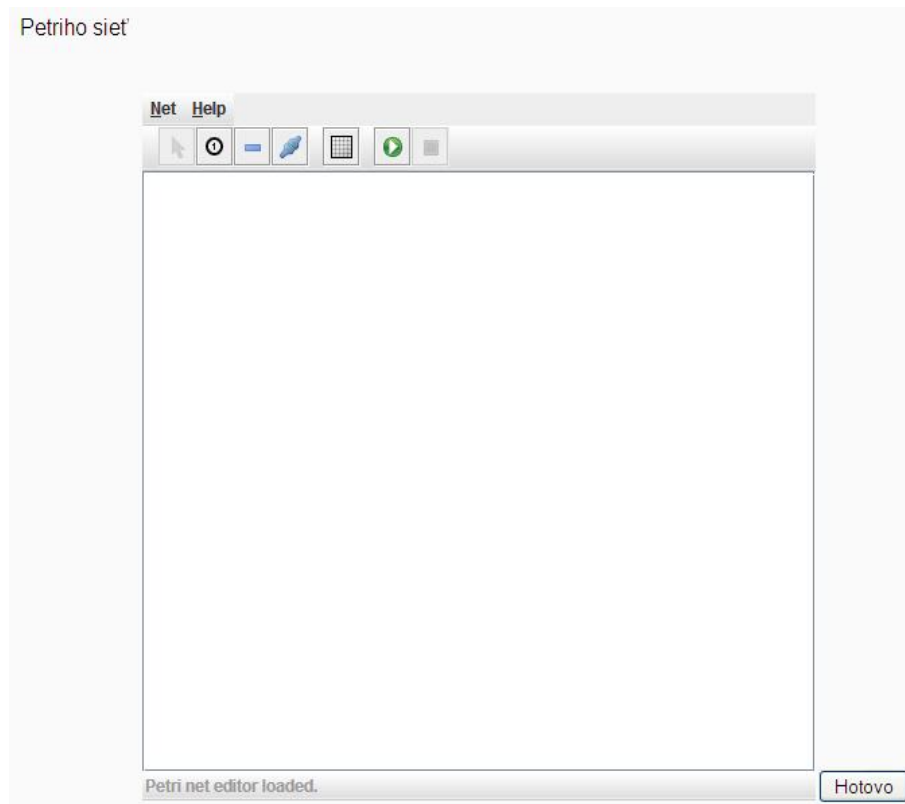
V týždennom prehľade (Obrázok C-2) si všimneme Pridať aktivitu, kde po kliknutí na ňu zvolíme možnosť Test. Zobrazí sa nám stránka, ako na obrázku C-3, kde sa dájú vytvárať



otázky do testu.

Obrázok C-3

Máme na výber z mnoho druhov typov otázok. Každá má osobitné nastavenia a nakonfigurovať sa dajú jasne z inštrukcií. Prikladám ešte obrázok C-4 z typu otázky *petrihosiet*. Je tu vidieť Java applet, kam učiť nakreslí sieť a potvrdí nakreslenie tlačidlom Hotovo.



Obrázok C-4

as pre študentov

Po prihlásení do systému si študent vyberie kurz, v ktorom chce test rieši . Po vybratí kurzu sa zobrazí obrazovka s jednotlivými sekciami, ktoré daný kurz obsahuje.

The screenshot displays the Moodle interface for a course titled "Course Fullname 101". The main content area is a "Týždenný prehľad" (Weekly Overview) table. The table has columns for dates and checkboxes. The current date, 12 December, is highlighted in orange. The table lists weekly intervals from 14 November to 22 January. The left sidebar contains navigation menus for "Ludia", "Aktivity", "Prehľad fóra", and "Administrácia". The right sidebar shows "Najnovšie správy" (Latest News) and "Aktuálna činnosť" (Current Activity).

| Dates                     | Checkboxes                          |
|---------------------------|-------------------------------------|
| 14 November - 20 November | <input type="checkbox"/>            |
| 21 November - 27 November | <input type="checkbox"/>            |
| 28 November - 4 December  | <input type="checkbox"/>            |
| 5 December - 11 December  | <input type="checkbox"/>            |
| 12 December - 18 December | <input checked="" type="checkbox"/> |
| 19 December - 25 December | <input type="checkbox"/>            |
| 26 December - 1 January   | <input type="checkbox"/>            |
| 2 January - 8 January     | <input type="checkbox"/>            |
| 9 January - 15 January    | <input type="checkbox"/>            |
| 16 January - 22 January   | <input type="checkbox"/>            |

Obrázok C-5

Vyberie si test, ktorý chce vypracovať a po kliknutí na sa zobrazí stránka, kde študent potvrdí zaatie testu.

**TÍMOVÝ PROJEKT II.**  
**Podpora vzdelávania v predmete Opis**  
**digitálnych systémov**  
**(LETNÝ SEMESTER)**

---

## Príloha D – Používateľská príručka

Náš projekt je určený na overenie teoretických a praktických znalostí študentov z predmetu Špecifické a opisné jazyky. Grafické prostredie je urobené tak, aby bolo najjednoduchším a najintuitívnejším spôsobom uľahčovalo orientovanie ako aj u študentov tak aj študentom pri zadávaní, vypracovávaní a vyhodnocovaní testov. V našej aplikácii sú dva funkčné moduly a to modul petriho sietí a modul „chytáposunovky“. Rozlišujú sa dva rôzne druhy používateľov s dvoma rôznymi právomocami. A to entita študent a entita učitelia.

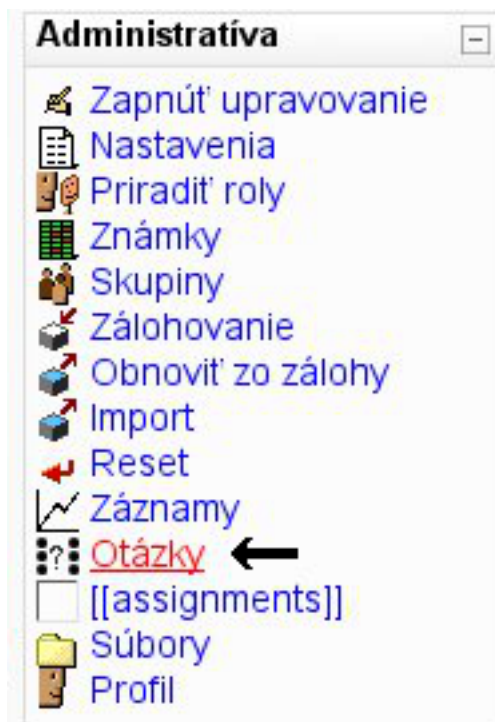
Základnou požiadavkou pre správne nainštalovanie modulu chytáposunovka je korektné nainštalovaný Moodle vo verzii aspoň 1.9.8. Ešte pred samotnou inštaláciou treba nakopírovať adresáre petrisiet, chytáposunovka do adresára moodle/question/type/. Je to potrebné preto pred inštaláciou, z dôvodu, že sú tam moodlovské skripty pre inštaláciu databázových tabuliek. alej treba skopírovať adresár filtra pre zobrazovanie vhdí kódu: adresár geshi do adresára modle/filter.

### Príručka pre vyučujúceho

Predpokladáme, že vyučujúci je už so systémom Moodle oboznámený a preto sa budeme venovať len tvorbe testu.

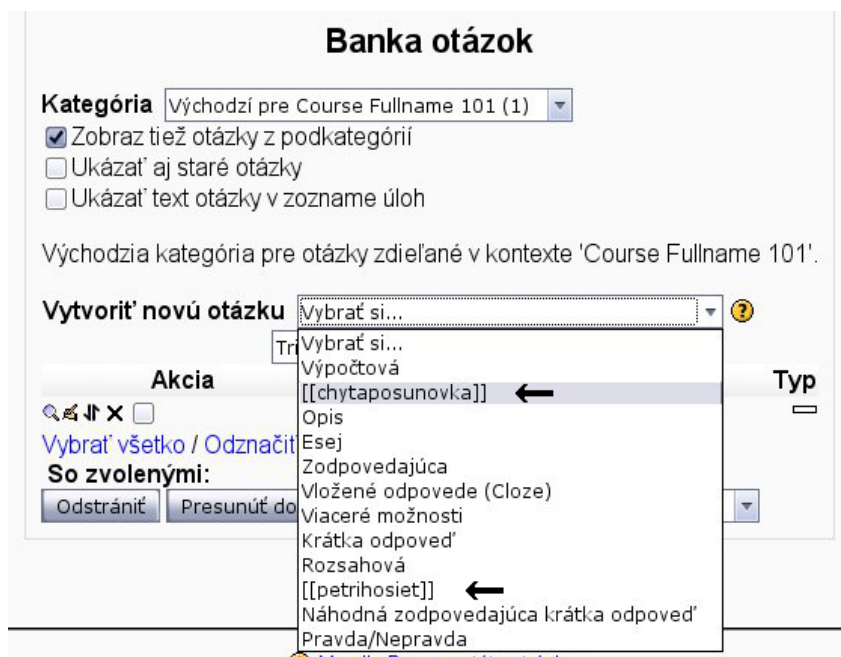
Vyučujúci najprv vytvorí otázky typu „chytáposunovka“ a „petrihosiet“.

Najprv je treba v menu Administrácia zvoliť pridávanie otázok.



Obr. D-1 administratíva

Potom si zvoli typ otázky, bu chytaposunovka alebo petrihosiet.



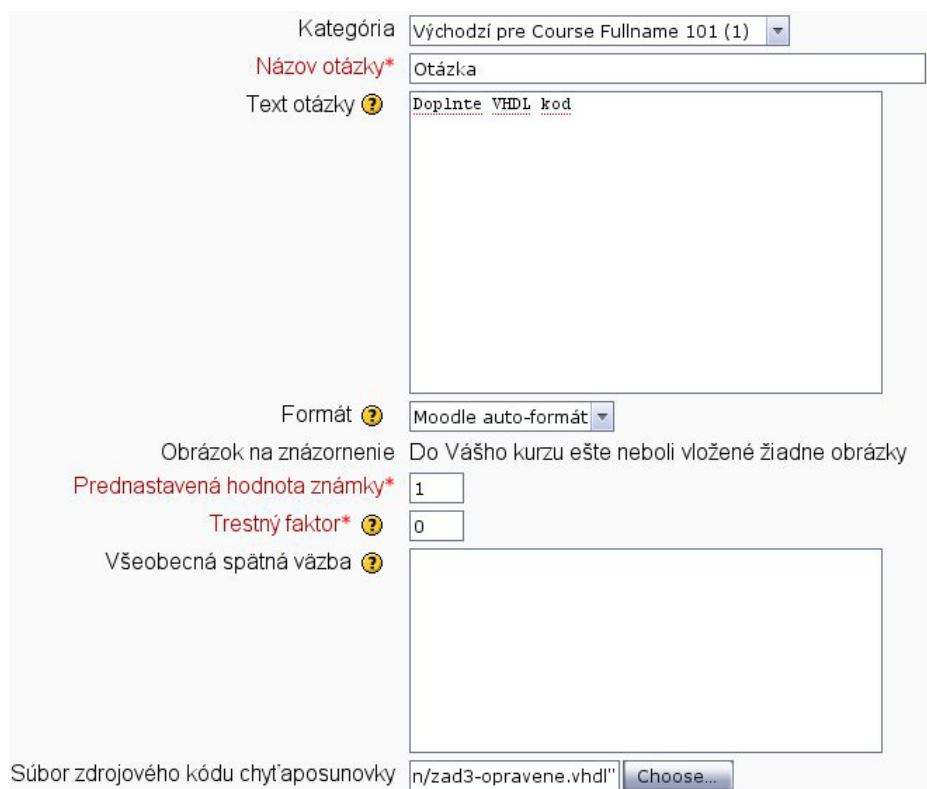
Obr. D-2 banka otázok



V prípade „chytaposunovka“ treba následne vyplni Názov otázky, znenie zadania a súbor s pripravenými tagmi na doplnenie.

Tagy treba do súboru dať ešte pred jeho uploadom do systému Moodle a sú v tvare:

<droper> as zdrojového kódu</droper>



The screenshot shows the Moodle question creation interface. The 'Kategória' dropdown is set to 'Východzí pre Course Fullname 101 (1)'. The 'Názov otázky\*' field contains 'Otázka'. The 'Text otázky' field contains 'Doplnite VHDL kód'. The 'Formát' dropdown is set to 'Moodle auto-formát'. The 'Obrázok na znázornenie' field contains the text 'Do Vášho kurzu ešte neboli vložené žiadne obrázky'. The 'Prednastavená hodnota známky\*' field contains '1'. The 'Trestný faktor\*' field contains '0'. The 'Všeobecná spätná väzba' field is empty. At the bottom, the 'Súbor zdrojového kódu chytaposunovky' field contains 'n/zad3-opravene.vhdl' and a 'Choose...' button.

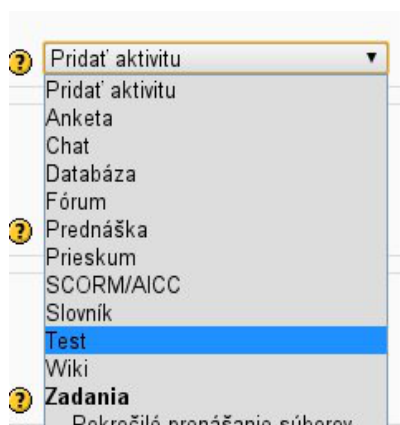
Obr. D-3 formulár

V prípade otázky typu „petrihosiet“ je treba takisto vyplni názov otázky a znenie zadanie, pod a ktorého bude študent vedieť aké parametre má petriho sieť spísať.

alej je treba v Java applete navrhnúť petriho sieť, ktorá bude spísať dané zadanie. Táto sieť sa potom použije ako referenčná pri kontrole, či študent úspešne zvládol návrh petriho siete pod a zadania.

Pokyny na prácu s appletom sú uvedené v samostatnej časti tejto príručky.

Po vytvorení dostatočného množstva otázok je potrebné vytvoriť Test. Na hlavnej stránke kurzu sa dá do daného týždňa vložiť „Aktivita“, v našom prípade pojde o aktivitu typu „Test“.



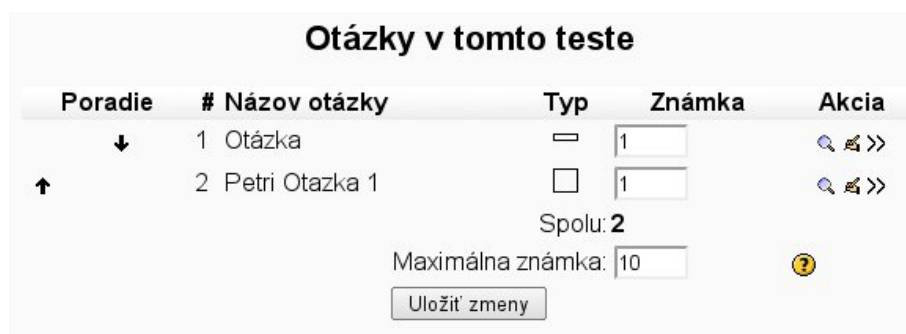
Obr. D-4 aktivita

Využívajúci vyplní potrebné údaje k testu ako je jeho názov, ktorý uvidia študenti. Po odoslaní je potrebné da Test upraví kde z banky otázok pridáme výber jednotlivých otázok do testu. Bu kliknutím na znak „<<“ ved a otázky, alebo hromadne oznaíme otázky, ktoré v teste chceme a klikneme na tlačítko „Prida do testu“.



Obr. D-5 otázky

Otázky sa zobrazia potom na svojej obrazovke v „Otázky v tomto teste“.

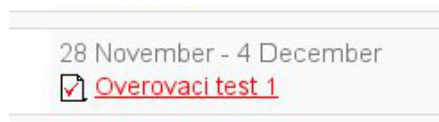


Obr. D-6 otázky 2

Potvrdíme zmenu kliknutím na tlačítko „Ulož zmeny“ a následne už je možné aby študenti zali test vyp a .

### Príruka pre študenta

Študent sa prihlási do systému Moodle a zvolí si daný kurz. Na hlavnej stránke uvidí prehľad celého semestra kurzu. Tu nájde test, ktorý má vypracovať. Klikne na neho a môže začať vypracovávať test.



Obr. D-7 test

Každú otázku testu je potrebné odoslať kliknutím na tlačítko pod každou otázkou.

Otázku typu „chytaposunovka“ študent rieši tak, že myšou presúva možné odpovede pod zobrazeným zdrojovým kódom na šedé mieste v kóde, tak aby výsledkový zdrojový kód bol skompilovateľný a funkčný.

A screenshot of a Moodle question interface. The question is titled "Doplnite VHDL kod". Below the title, there is a text area for the answer. The text area contains VHDL code with several lines highlighted in grey boxes, indicating where the student can drag and drop options. The code is: 

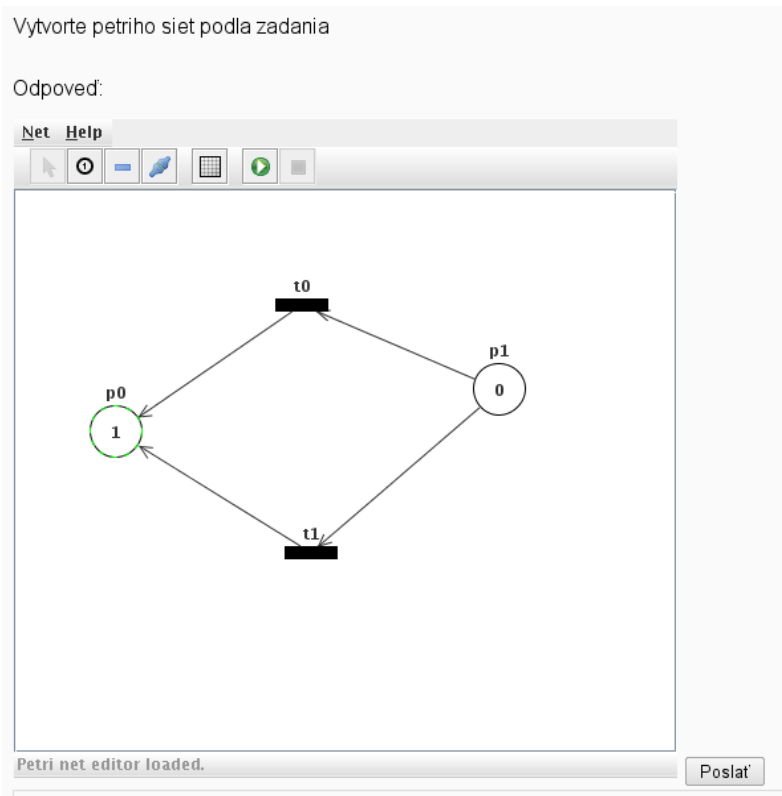
```
or out <=
or_inA or or_inB
; end; architecture Spravanie_AND3 of Krabicka_AND3 is
and3_out <= and3_inA and and3_inB and and3_inC; end; -- OR 3-vstupovy entity Krabicka_
begin
; end;
```

 To the right of the text area, there is a list of options: "begin" and "kabel\_in". Below the options, there is a button labeled "Uložit otázku".

Obr. D-8 VHDL-kod

Otázku typu „petrihosiet“ rieši študent v java applete. Podľa zadania navrhne petriho sieť, pričom si ju môže aj otestovať simuláciou. Po dokončení práce s appletom je potrebné kliknúť na tlačítko odoslať pod appletom.

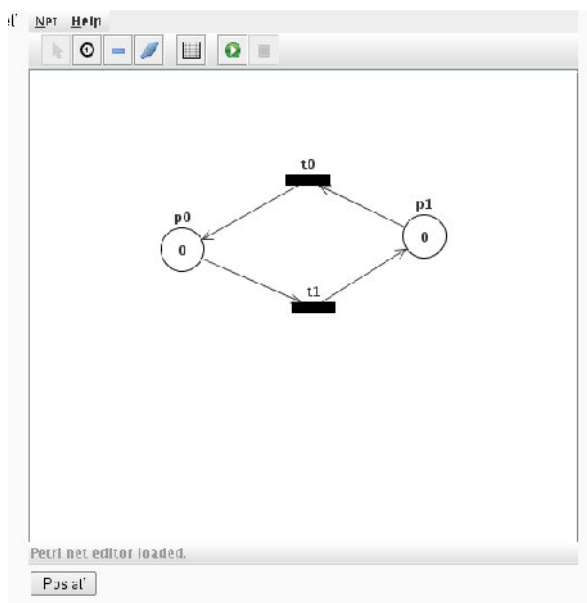
Následne po ukončení vypracovávania testu, klikne v spodnej časti na „Odoslať stránku“, čím sa test odovzdá, vyhodnotí a uloží do databázy.



Obr. D-9 Petriho sieť

### Všeobecná príručka použitia Java appletu

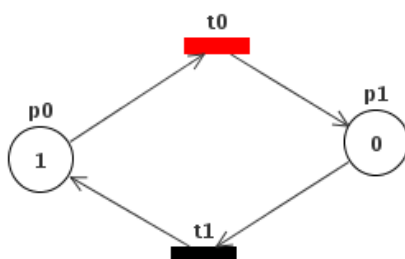
Java applet má veľmi jednoduché ovládanie. Vidíme na obrázku jeho hlavné okno, kde najviac miesta vyplnia a priestor pre kreslenie petriho siete. Najprv nad týmto priestorom je niekoľko tlačítkov na tvorbu siete. Prvé tlačidlo prepne kurzor do označovacieho módu. Takto je možné napríklad označiť 5 objektov a všetky posunúť zároveň. Druhé tlačidlo slúži na vkladanie miest, nasleduje tlačidlo na vkladanie prechodov a ďalšie je pre tvorbu hrán medzi miestami a prechodmi. Hrany sa môžu vkladať tak, že sa označí hrana a prechod a kliknutím na tlačidlo pre tvorbu hrán sa vytvorí hrana medzi nimi. Potom je tlačidlo pre zobrazenie/schovanie mriežky a dve tlačidlá pre spustenie a zastavenie simulácie siete.



Obr. D-10 petriho sie II.

Dvojklik na jednotlivé objekty (v označovacom móde) zobrazí okno kde je možné nastaviť kapacitu miest, váhu hrán a ďalšie parametre siete.

Používateľ má možnosť si navrhnutú sieť aj odsimulovať a zistiť či naozaj robí to čo od nej žiada. Slúžia na to dve posledné tlačítka (úplne vpravo). Prvé (trojuholník v krúžku) zapne simuláciu. V tomto sa začína kroková simulácia, pričom každý ďalší krok je potrebné spustiť kliknutím na vysvietené prechody (vysvietia sa tie, cez ktoré je možné prejsť do ďalšieho stavu). Zastavenie simulácie je možné kliknutím na tlačítko so štvorcikom.



Obr. D-11 petriho graf

Pre odoslanie navrhnutej siete (či už pri tvorbe referenčnej siete alebo pri riešení testu študentom) je potrebné po skončení práce kliknúť na tlačítko Poslať pod appletom.