

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra informatiky a výpočtovej techniky

Študijný odbor odbor: INFORMATIKA

Počítačová podpora programátorskej súťaže

Tímový projekt

Tím c. 1: Bc. Michal Bigoš
 Bc. Szabolcs Bugár
 Bc. Vladimír Grlický
 Bc. Ladislav Gyokeres
 Bc. Juraj Hájek
 Bc. Attila Kúr
Vedúca projektu: prom . mat. Lubica Hanulová
Školský rok: 2001/2002

Zadanie

Pocítacová podpora programátorskej súťaže

FEI STU sa už tradícne zúčastňuje medzinárodnej ACM súťaže. Je to súťaž tímov, ktoré riešia problémy v niektorom z povolených programovacích jazykov v zadanom case. Každý tím má k dispozícii jeden počítač. Tímy svoje riešenia odovzdávajú prostredníctvom počítačovej siete, pokiaľ neodovzdajú správne riešenie problémov alebo pokiaľ neuplynie vyhradený čas. Víťazný tím je ten, ktorý vyrieši najviac problémov (v najkratšom case). Lokálnemu kolu ACM súťaže na FEI STU predchádza korešpondenčné kolo, ktoré má mierne zmenené pravidlá.

Cielom tohto projektu je vytvoriť systém na podporu programátorských súťaží typu ACM súťaže, ktorý umožňuje:

- konfigurovanie parametrov systému (počet problémov, maximálny počet odovzdaní problému, čas na riešenie problémov, povolené programovacie jazyky ...) tak, aby sa mohol použiť aj na podobné súťaže s obmednenými pravidlami,
- jednoduché odovzdávanie problémov, potvrdzovanie prijatia riešení a oboznamovanie tímov s priebežnými výsledkami,
- automatické testovanie a vyhodnocovanie problémov,
- kontrolu, či tímy používajú len povolené prostriedky (programy).

Pri riešení treba zvážiť, či je vhodné vytvoriť systém, ktorý by sa dal použiť aj pre korešpondenčné kolo programátorskej súťaže.

Úvod

Tento dokument slúži ako projektová dokumentácia k predmetu tímový projekt, ktorý sa vyučuje na inžinierskom stupni štúdia na Fakulte elektrotechniky a informatiky v Bratislave.

Úlohou tohoto predmetu je oboznámit študentov s metódami práce v tíme, osvojit a prehľbit si znalosti, ktoré sú potrebné pre efektívnu prácu v tíme. Zadania pre tento predmet sú koncipované tak, aby projekt nemohol zrealizovať jeden, príp. dvaja nadaní jednotlivci, ale aby bola potrebná skutocná tímová spolupráca jednotlivých členov.

Predmet má aj vysoký prínos pre prax, nakoľko rozsah softvérových projektov v súčasnosti vyžaduje, aby ich neriešili jednotlivci, ale aby sa riešili v rámci väčších riešiteľských kolektívov.

Náš tím v rámci tímového projektu rieši problematiku podpory programátorských súťaží typu ACM, čo je celosvetová programátorská súťaž usporiadaná americkou organizáciou Association for Computing Machinery. Takéto súťaže zohrávajú dôležitú úlohu pri propagácii počítačových vied, aj pri odhalovaní perspektívnych talentov v danom odbore

Chceli by sme však, aby sa výsledný produkt dal v budúcnosti používať aj v rámci cvicení na našej fakulte, čím by sa naše úsilie, ktoré sme vynaložili na tvorbu systému, ďalej zhodnotilo.

Dokumentácia k projektu pozostáva z dvoch častí. Prvá časť obsahuje informácie, ktoré súvisia s riadením projektu, ako je napr. plán projektu, rozdelenie úloh medzi členov tímu a zápisy z jednotlivých stretnutí. Druhá časť obsahuje dokumentáciu k jednotlivým fázam riešenia projektu.

Riadenie projektu

Obsah

| | |
|--|------------|
| ZADANIE..... | I |
| ÚVOD | II |
| OBSAH..... | III |
| 1 PONUKA | 1 |
| 1.1 ÚVOD..... | 1 |
| 1.2 ZLOŽENIE TÍMU..... | 1 |
| <i>Bc. Michal Bigoš.....</i> | <i>1</i> |
| <i>Bc. Szabolcs Bugár.....</i> | <i>1</i> |
| <i>Bc. Vladimír Grlický.....</i> | <i>2</i> |
| <i>Bc. Ladislav Gyökeres.....</i> | <i>2</i> |
| <i>Bc. Juraj Hájek.....</i> | <i>2</i> |
| <i>Bc. Attila Kúr.....</i> | <i>3</i> |
| 1.3 MOTIVÁCIA..... | 3 |
| 1.4 CO MÔŽEME POSKYTNÚT | 3 |
| 1.4.1 <i>Funkcie systému.....</i> | <i>4</i> |
| 1.4.2 <i>Server.....</i> | <i>4</i> |
| 1.4.3 <i>Klient.....</i> | <i>4</i> |
| 1.4.4 <i>Používatelia systému.....</i> | <i>4</i> |
| 1.5 PREDPOKLADANÉ ZDROJE..... | 8 |
| 1.5.1 <i>Softvérové požiadavky.....</i> | <i>8</i> |
| 1.5.2 <i>Hardvérové požiadavky.....</i> | <i>9</i> |
| 1.6 ZORADENIE TÉM PODĽA PRIORITY..... | 9 |
| 1.7 NÁVRHY ZMIEN ORGANIZÁCIE PREDMETU..... | 9 |
| 2 PREZENTÁCIA PONUKY | 10 |
| 3 PLÁN PROJEKTU | 11 |
| 4 ÚLOHY V TÍME..... | 15 |
| ZÁPISY ZO STRETNUTÍ..... | 17 |
| PRÍLOHA A..... | 40 |

1 Ponuka

1.1 Úvod

Vzhľadom na pokročilú technologickú úroveň súčasných výpočtových systémov sa nám ponúka možnosť otestovania vedomostí študentov a odovzdávania školských zadaní cez Internet. Takýto prístup uľahčuje vo významnej miere aj prácu cvičiacich a skúšajúcich, ako aj učiteľov, ktorí usporadúvajú korešpondenčné súťaže formou zasielania vyriešených zadaní cez prehliadacie prístupné pod každým operacným systémom. Týmto je zabezpečené, že súťažiaci na svojich počítačoch nepotrebuju spúšťať nijaký špeciálny softvér navyše.

1.2 Zloženie tímu

Riešiteľský tím tvorí šesť študentov prvého ročníka inžinierskeho štúdia na Fakulte elektrotechniky a informatiky Slovenskej technickej univerzity v Bratislave. V nasledujúcich odstavcoch sú stručne uvedené ich odborné profily.

Bc. Michal Bigoš

Bakalárske štúdium absolvoval na FEI STU v Bratislave, kde navštevoval odbor Informatika, špecializácia Softvérové inžinierstvo. Pol roka pracoval ako inžinier kvality vo firme WebMessenger. Tu sa podieľal aj na tvorbe programov v jazyku Java. V súčasnosti pracuje vo firme mgate na vývoji internetových aplikácií, kde ako nástroje používa hlavne HTML, DHTML, PHP, JavaScript a SQL. Okrem vyššie spomenutých jazykov aktívne ovláda jazyky C, C++, Visual C++, Java, Lisp a Prolog. S databázovými prostrediami sú mu blízke MS SQL Server a Oracle. S prehĺbovaním vedomostí v oblasti informacných systémov získaných počas bakalárskeho štúdia chce pokračovať aj v inžinierskom štúdiu, comu zodpovedá i výber predmetov Architektúra softvérových systémov, Databázové systémy II. a Manažment v softvérovom inžinierstve. Nadobudnuté skúsenosti s prácou v tíme by chcel uplatniť aj pri riešení tohto projektu.

Bc. Szabolcs Bugár

Už viac ako 10 rokov sa zaoberá výpočtovou technikou, preto má dlhoročné skúsenosti s viacerými architektúrami a platformami. Pomáhal napláňovať a vybudovať počítačovú sieť pre základnú a strednú školu vo Velkom Mederi. Je absolventom bakalárskeho štúdia na FEI STU v odbore Informatika, špecializácia Softvérové inžinierstvo. Ovláda rôzne programovacie jazyky, medzi ktoré patrí C, C++, Microsoft Visual C++, MFC, Visual Basic, dBase IV, Lisp, Prolog. Vytvoril internetovskú stránku rodinnej firmy, kde pomáha svojou vedomosťou v oblasti počítačovej techniky. V tomto roku začal študovať aj kurz sieťových technológií Cisco Certificate Networking Academy a absolvoval prvý semester. Podieľal sa na výstavbe laboratória pre tento kurz na Katedre informatiky a výpočtovej techniky. V posledných troch rokoch sa viackrát zúčastnil na lokálnom a korešpondenčnom kole súťaže ACM, kde jeho tím získal veľmi dobré výsledky. Má veľmi dobré skúsenosti s použitím

vyhodnocovacích systémov pre súťaže (aj zo strany ich používateľov), čo môže byť prínosom pri riešení tohto tímového projektu. V rámci predmetu Odborné praktikum pomáha vyberať úlohy pre lokálne a korešpondenčné kolo súťaže ACM.

Bc. Vladimír Grlický

Absolvoval bakalárske štúdium na FEI STU v odbore Informatika, špecializácia Softvérové inžinierstvo. So súťažou podobného typu sa stretol už na strednej škole, kde sa zúčastnil Korešpondenčného seminára s programovania (KSP) a programátorskej súťaže ZENIT. Má praktické skúsenosti s programovaním v jazykoch Assembler, Pascal, C, C++, Visual C++, LISP, Prolog, Java. Ovláda prácu vo vývojových prostrediach Microsoft Visual Studio, Together Control Center, Borland JBuilder. V súčasnosti pracuje vo firme mgate na vývoji internetových aplikácií (pomocou technológií PHP, SQL, JavaScript, HTML, DHTML) v pozícii programátor, návrhár. V rámci záverečného projektu „Objektovo-orientovaný simulátor“ získal skúsenosti s problematikou objektovo-orientovanej analýzy (OOA) a návrhu (OOD) informacných systémov. K jeho súčasným záujmom patrí prehĺbovanie znalostí o technológiách UML, XML a JSP. Takisto sa zaujíma o architektúru stredne veľkých a veľkých softvérových systémov, vyššie programovacie jazyky a objektovo-orientované programovanie.

Bc. Ladislav Gyökeres

Je absolventom bakalárskeho štúdia na FEI STU, odbor Informatika, špecializácia Softvérové inžinierstvo. Zaoberá sa hlavne vývojom aplikácií v prostredí Microsoft Visual C++. Počas vysokoškolského štúdia sa zúčastňoval na viacerých programovacích súťažiach, kde pracoval v tíme. Má skúsenosti s databázovým systémom MS Access a ovláda jazyky C/C++, Java, Pascal, Basic, Lisp, Prolog, SQL. Má skúsenosti aj so tvorbou statických WWW stránok. Vybrané predmety v rámci inžinierskeho štúdia ako Databázové systémy II a Manažment v softvérovom inžinierstve znamenajú ďalšie prínosy pre tento tím. Súčasné oblasti jeho záujmu predstavujú neurónové siete a databázové systémy.

Bc. Juraj Hájek

Je absolventom bakalárskeho štúdia na FEI STU, odbor Informatika, špecializácia Softvérové inžinierstvo. Má 3 roky programátorskej praxe, zameranej najmä na databázové technológie, ale je aj autorom niekoľkých menších aplikácií pre Windows CE. Ovláda nasledovné programovacie jazyky: Visual Basic - na veľmi dobrej úrovni, ďalej C/C++, Visual C++, eMbedded Visual C++, Pascal, Java, čiastočne aj PHP. Ovláda databázové technológie DAO a ADO a databázové prostredia MS Access a MS SQL Server. Má skúsenosti s lokalizáciou väčších softvérových projektov. Jeho záverečná práca bola zameraná na spracovanie videa. Problém mu nerobí ani práca pod operačnými systémami na báze UNIX-u.

Bc. Attila Kúr

V posledných rokoch sa viackrát úspešne zúčastnil lokálneho a korešpondenčného kola programátorskej súťaže ACM. Preto veľmi dobre pozná problematiku, ktorá sa v tomto projekte má riešiť a to aj zo strany riešiteľov. Má praktické skúsenosti s prácou v prostrediach Microsoft Visual C++ a MS Access. Z programovacích jazykov, ktoré môžu byť užitočné pri riešení tohto problému ovláda C/C++ (MFC), MS Jet-SQL a Pascal. Najnovšie taktiež vytvára webové aplikácie v prostredí Macromedia FLASH. Tieto jeho vedomosti môžu pomôcť pri vytváraní príjemného používateľského rozhrania pre riešený problém a podobnú úlohu si dokáže predstaviť aj v tíme. Pri realizácii sietovej časti aplikácie mu dobre poslúži kurz sieťových technológií Regional Cisco Networking Academy.

1.3 Motivácia

Jedným z dôvodov, prečo sme si vybrali práve tento projekt, bola niekoľkonásobná účasť viacerých členov tímu na lokálnom a korešpondenčnom kole programátorskej súťaže ACM, ktorá sa už tradície koná každý rok na našej fakulte, preto je táto problematika nášmu tímu blízka. Táto súťaž každý rok využívala iné softvérové a hardvérové prostriedky, ale z našich skúseností vieme, že tieto prostriedky neboli vždy užívateľsky príjemné alebo plne funkčné. Skúsenosti, ktoré sme získali počas používania existujúceho systému, sú pre nás hlavnou motiváciou pre vypracovanie tohto projektu. Poznáme prostredie, v ktorom podobné súťaže prebiehajú, pričom predpokladáme, že táto znalosť nám pomôže lepšie identifikovať prípadné ďalšie problémy, ku ktorým by na takejto súťaži mohlo dôjsť. Tiež si uvedomujeme, že bezproblémový, a v čo najväčšej možnej miere automatizovaný priebeh celej súťaže, je nielen vynikajúcou vizitkou usporiadateľskej fakulty, ale výrazne uľahčí prácu ako súťažiacim (počas súťaže), tak aj samotným jej usporiadateľom (pred, počas aj po súťaži). Preto by sme chceli vytvoriť jednoduchú, platformovo nezávislú aplikáciu, ktorá by bola modulárna, ľahko modifikovateľná, a tiež by sa dala využiť v pedagogickom procese ako pomôcka alebo nástroj pre overenie programátorskej zručnosti študentov na fakulte.

1.4 Co môžeme poskytnúť

Chceme vyvinúť komplexný produkt, ktorý by sa dal využiť nielen pri organizovaní programátorských súťaží, ale aj pri bežnej výučbe predmetov na fakulte.

Počas nasledujúcich dvoch semestrov budeme analyzovať podobné riešenia a pokúsime sa vytvoriť ucelený systém, ktorý v prvom rade splní kritériá zadania a nekladie veľké nároky na prostredie, v ktorom bude prevádzkovaný.

Budeme vychádzať z možností našej fakulty, lebo predpokladáme, že systém sa bude využívať práve tu. Z dlhodobého hľadiska je tiež dôležité, aby aplikácia bola ľahko modifikovateľná a čo najviac platformovo nezávislá.

Pri návrhu systému budeme klásť zvýšený dôraz na jeho modulárnosť a prehľadnú konfigurovateľnosť.

1.4.1 Funkcie systému

Nami navrhovaný systém na podporu programátorskej súťaže bude poskytovať:

- zabezpečený prístup viacerých typov používateľov,
- komfortné používateľské rozhranie nezávislé od platformy,
- vysokú konfigurovateľnosť jeho parametrov,
- prehľadné zobrazovanie všetkých potrebných informácií o práve prebiehajúcej súťaži,
- možnosť komunikácie medzi jednotlivými typmi používateľov,
- automatické testovanie a vyhodnocovanie problémov,
- monitorovanie a vhodná prezentácia všetkých vykonaných akcií v systéme,
- podpora viacjazyčného používateľského rozhrania

V závislosti od týchto funkcií sme ako najvhodnejšiu architektúru vybrali centrálny riadený systém typu klient - server.

1.4.2 Server

Server bude plniť nasledovné funkcie:

- uchovávanie databázy riešených príkladov,
- uchovávanie čiastočných výsledkov jednotlivých tímov,
- monitorovanie činností klientov počas celej súťaže,
- vyhodnocovanie zadaní (v prípadnej súčinnosti s rozhodcami)

1.4.3 Klient

Úlohou klienta je sprístupňovanie služieb servera (pomocou internetového rozhrania), pričom možnosti klienta sú dané jednotlivými typmi používateľov.

1.4.4 Používatelia systému

V systéme sme identifikovali nasledovné typy používateľov:

- administrátor,
- rozhodca,

- súťažiaci,
- pozorovateľ

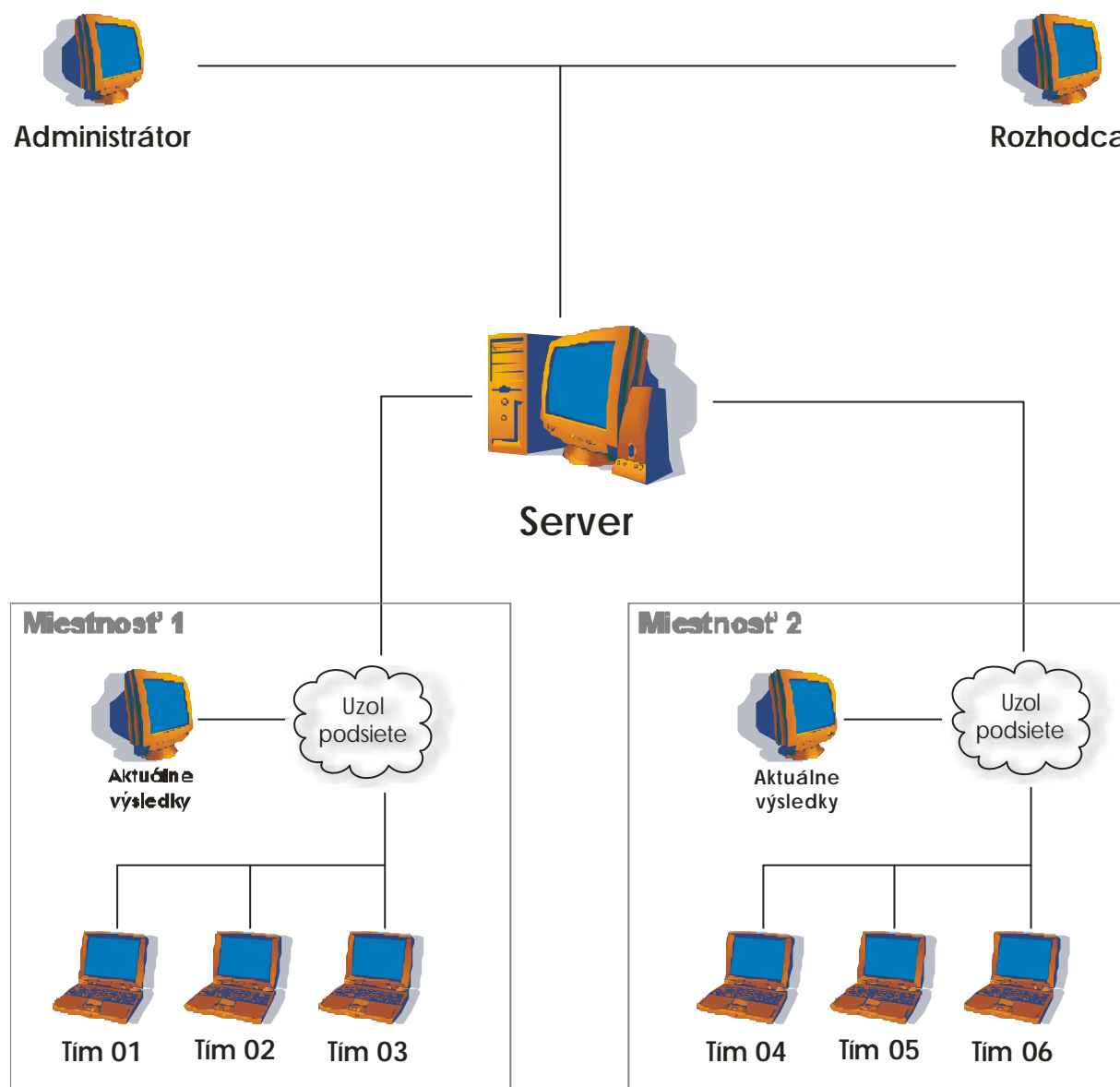
Administrátor - jeho úlohou je prvotná inštalácia a konfigurácia parametrov systému a správa ostatných používateľských účtov.

Rozhodca - pri úlohách, ktoré sa nedajú vyhodnotiť automaticky, ohodnocuje správnosť a kvalitu odovzdaných zadaní. Má možnosť odpovedať na všeobecné alebo problémovo-orientované otázky súťažiacich.

Súťažiaci - má možnosť prevziať si zo servera jednotlivé zadania, odovzdať riešenia a prehliadať čiastkové alebo konečné výsledky.

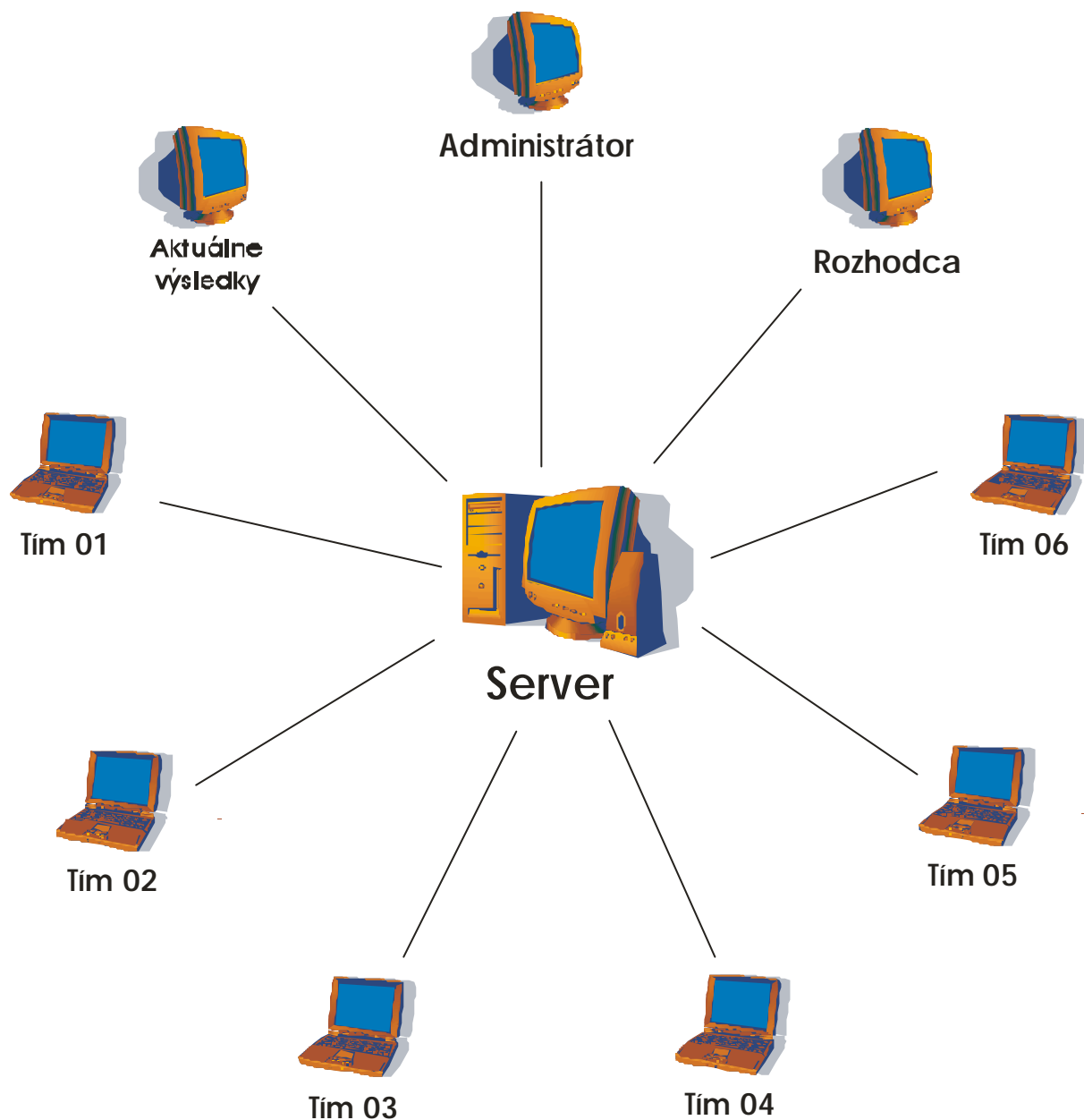
Pozorovateľ - pri práve prebiehajúcich súťažiach si môže prezerať čiastkové výsledky a priebežné poradie súťažiacich tímov, avšak nemôže sa aktívne zúčastňovať súťaže.

–



Obr. c. 1: Príklad štruktúry navrhovaného systému pre lokálne kolo súťaže

Na Obr. c. 1 je znázornený príklad možnej štruktúry a použitia navrhovaného systému pre lokálne kolo súťaže. Navrhnutý systém má tú výhodu, že účastníci súťaže nemusia byť fyzicky prítomní v jednej miestnosti. Táto vlastnosť systému zabezpečuje lepšiu flexibilitu a rozširuje oblasť použitia systému (napríklad môžu byť do súťaže zapojené aj ďalšie tímy z iných fakúlt).



Obr. c. 2: Príklad štruktúry navrhovaného systému pre korešpondenčné kolo súťaže

V korešpondenčnom kole (Obr. c. 2) môžu súťažiaci okrem rozhrania použitého v lokálnom kole využívať aj výhody e-mailovej schránky. Pri tomto kole súťaže nepotrebujeme také obmedzenie ako pri lokálnom kole, a je potrebné len zabrániť vniknúť do databázy hlavného servera. Pre korešpondenčné kolo nepotrebujeme zaradiť komunikačné moduly do štruktúry systému, tzn. klienti budú napojení „priamo“ na hlavný server systému.

Pri výbere implementačného prostredia prichádzajú do úvahy dve alternatívy:

- a) použitie Java technológie a vytvorenie modulov, ktoré budú navzájom medzi sebou komunikovať,

- b) použitie jednej z technológií PHP, JSP, ASP a vytvorenie online verzie systému prístupného pomocou internetového prehliadaca

Preferujeme alternatívu PHP, a to kvôli jeho platformovej nezávislosti, dostupnosti zdarma a našim skúsenostiam s týmto jazykom.

Nevýhodou riešenia na báze Javy je nutnosť inštalácie na každom klientovi, výhodou je lepšie zabezpečenie systému a vzájomná komunikácia medzi jednotlivými modulmi.

Kvôli zámeru čo najviac spríjemniť prácu obidvoch strán počítačového systému budeme na strane klienta predpokladať grafické používateľské prostredie. V prípade špecifických požiadaviek však nebude problém systém doplniť o textové rozhranie.

Chceme vyvinúť komplexný produkt, ktorý by sa dal využiť nielen pri organizovaní programátorských súťaží, ale aj pri bežnej výučbe v rámci fakulty.

Z našej strany môžeme poskytnúť (ako už bolo uvedené) skúsenosti z podobných súťaží, ako aj vedomostnú základňu ďalších našich dvoch členov, ktorí sa zaoberajú tvorbou internetových aplikácií komerčne.

Zosilnením hardvérovej konfigurácie hlavného serveru by sme mohli dostať systém doplnený o spracovanie viacerých súťaží. Princíp takéhoto „multikontestového“ systému (MCS – Multicontest System) spočíva v evidencii (tvorba, monitorovanie, archivácia, vyhodnocovanie) viacerých nezávislých súťaží ovládaných rovnakým spôsobom. Pritom každá zo súťaží môže mať nastavené vlastné parametre (napríklad korešpondenčné kolo jednej súťaže môže prebiehať súčasne s lokálnym kolom inej súťaže).

1.5 Predpokladané zdroje

1.5.1 Softvérové požiadavky

1.5.1.1 Server

Na strane servera predpokladáme použitie HTTP serveru Apache s využitím programovacieho jazyka PHP a niektorej z databázových technológií MySQL alebo PostgreSQL. Uvedené nástroje pracujú pod viacerými OS, preto otázku výberu konkrétneho OS, pod ktorým budeme systém implementovať, nechávame zatiaľ otvorenú.

1.5.1.2 Klient

Jedinou požiadavkou na strane klienta je existencia internetového prehliadaca, pričom potrebu použitia skriptov na strane klienta ešte zvažujeme.

1.5.2 Hardvérové požiadavky

1.5.2.1 Server

Hardvérové požiadavky servera sa odvíjajú od počtu klientských staníc.

1.5.2.2 Klient

Vzhľadom na to, že jedinou podmienkou fungovania klienta je prítomnosť HTML prehliadaca sú hardvérové požiadavky minimálne.

1.6 Zoradenie tém podľa priority

1. Pocítacová podpora programátorskej súťaže (**PROG**)
2. Softvérová podpora organizovania vedeckej konferencie (**KONF**)
3. RoboCup – inteligencia hráča (**RoboCup**)

1.7 Návrhy zmien organizácie predmetu

Väčšia účasť študentov na výbere tém, príp. na ich tvorbe. Pravdepodobne by došlo k vyššej úrovni vypracovanosti projektov vďaka väčšej zainteresovanosti samotných študentov na projektoch.

2 Prezentácia ponuky

Slovenská technická univerzita v Bratislave
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra informatiky a výpočtovej techniky
Oddelenie INFORMATIKA

PCINKA

POČÍTAČOVÁ PODPORA PROGRAMÁTORskej SÚTAŽE

Timový projekt:

Bc. Michal Bigoš Bc. Ladislav Gyökeres
Bc. Szabolcs Bugár Bc. Juraj Hájek
Bc. Mladomir Grlický Bc. Attila Kúr

Motivácia

- > účasť na programátorskej súťaži ACM
- > skúsenosť s používaním existujúceho systému
- > pomoc budúcim súťažiacim a organizátorom súťaže

Čo môžeme poskytnúť

- > kompletný produkt využiteľný aj pri bežnej výučbe
- > ľahko modifikovateľné a platformovo nezávislé aplikácie
- > interaktívna práca so systémom pomocou www rozhrania
- > systém pre podporu viacerých súťaží - Multioctest system (MCS)
- > nenáročnosť systému na prostredie, v ktorom bude prevádzkovaný

Úlohy systému

- > podpora niekoľkých typov používateľov
- > používateľské rozhranie nezávislé od platformy
- > vysoká konfigurovateľnosť parametrov systému
- > prezentácia informácií o práve prebiehajúcej súťaži
- > on-line konzultácie a rozhodzami
- > automatické testovanie a vyhodnocovanie problémov
- > monitorovanie akcií systémov
- > viacjazyčné používateľské rozhranie

Príklad štruktúry navrhovaného systému

Používatelia systému

- > administrátor
- > rozhodca
- > súťažiaci
- > pozorovateľ

3 Plán projektu

Zimný semester

| Týžden Hrubý plán projektu na zimný semester (Tímový projekt) | |
|--|---|
| 1. | ponuka (pridelenie študentov do tímov, zverejnenie tém a požiadaviek na vypracovanie ponuky) |
| 2. | ponuka (spracovanie ponuky – každý tím spracuje minimálne jednu ponuku, uskutočnenie stretnutí pre jednotlivé témy) |
| 3. | odovzdanie ponúk vyhodnotenie ponúk, urcenie rozvrhu a učiteľa pre tím |
| 4. | rozdelenie úloh, vytvorenie plánu projektu, analýza problému (štúdium problematiky) |
| 5. | analýza problému, špecifikácia požiadaviek a hrubý návrh riešenia |
| 6. | analýza problému, špecifikácia požiadaviek a hrubý návrh riešenia |
| 7. | analýza problému, špecifikácia požiadaviek a hrubý návrh riešenia |
| 8. | odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek riešenia spolu s hrubým návrhom |
| 9. | odovzdanie posudku analýzy, špecifikácie a hrubého návrhu iného tímu |
| 10. | dopracovanie zistených nedostatkov a návrh prototypu vybraných častí |
| 11. | implementácia prototypu vybraných častí |
| 12. | odovzdanie prototypu vybraných častí systému spolu s dokumentáciou a používateľská prezentácia prototypu |

| Týžden Zjemnený plán projektu na zimný semester pre tím c. 1 | |
|---|--|
| 1. | ponuka (pridelenie študentov do tímov, zverejnenie tém a požiadaviek na vypracovanie ponuky) |
| 2. | ponuka (spracovanie ponuky - každý tím spracuje minimálne jednu ponuku, uskutočnenie stretnutí pre jednotlivé témy) <ul style="list-style-type: none"> - vypracovanie spoločného rozvrhu členov tímu - výber témy - analýza základných požiadaviek zákazníka - vypracovanie ponuky - príprava prezentácie |

| Týžden Zjemnený plán projektu na zimný semester pre tím c. 1 | |
|---|---|
| 3. | odovzdanie ponúk <ul style="list-style-type: none"> - prezentácia - vyhodnotenie ponúk - urcenie rozvrhu a ucitela pre tím |
| 4. | rozdelenie úloh, vytvorenie plánu projektu, analýza problému (štúdium problematiky) <ul style="list-style-type: none"> - zhrnutie nedostatkov vytvorenej ponuky - rozdelenie úloh členov tímu - analýza problému - vytvorenie plánu projektu - vytvorenie základnej štruktúry pre web – prezentáciu - zaciatok práce na tímovom diskusnom fóre |
| 5. | analýza problému, špecifikácia požiadaviek a hrubý návrh riešenia <ul style="list-style-type: none"> - podrobnejšia analýza problému z hladiska súťažiacich na ACM - súťaž ACM – prieskum - analýza požiadaviek na systém z hladiska predmetov vyucovaných na FEI STU |
| 6. | analýza problému, špecifikácia požiadaviek a hrubý návrh riešenia <ul style="list-style-type: none"> - zhrnutie poznatkov o požiadavkách na systém - integrácia požiadaviek ACM – cvicenia na FEI STU - bližšia špecifikácia architektúry systému |
| 7. | analýza problému, špecifikácia požiadaviek a hrubý návrh riešenia <ul style="list-style-type: none"> - závererečná analýza - dokoncenie špecifikácie - dokoncenie návrhu - príprava dokumentácie - diskusia o tom, ktoré casti systému sa budú implementovat v rámci prototypu |
| 8. | odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek riešenia spolu s hrubým návrhom <ul style="list-style-type: none"> - vypracovanie posudku analýzy, špecifikácie a hrubého návrhu iného tímu - diskusia o dokumentácii konkurenčného tímu vzhľadom na prínos pre náš tím - rozdelenie úloh medzi členov tímu ohľadne implementácie prototypu - návrh prototypu vybraných castí - zaciatok implementácie prototypu - vytvorenie databázy a kostry aplikácie |
| 9. | odovzdanie posudku analýzy, špecifikácie a hrubého návrhu iného tímu <ul style="list-style-type: none"> - vytvorenie užívateľského rozhrania prototypu - registrácia súťažiacich - konfigurácia základných parametrov na strane administrátora systému |
| 10. | dopracovanie zistených nedostatkov vybraných castí |

| Týžden Zjemnený plán projektu na zimný semester pre tím c. 1 | |
|---|--|
| | <ul style="list-style-type: none"> - prípadné modifikácia rozpracovaného prototypu, vzladom na vypracovaný posudok druhého tímu - implementácia vnútornej logiky aplikácie vytvorených častí používateľského rozhrania |
| 11. | implementácia prototypu vybraných častí a jeho testovanie <ul style="list-style-type: none"> - testovanie prototypu - vytvorenie používateľskej príručky |
| 12. | odovzdanie prototypu vybraných častí systému spolu s dokumentáciou a používateľská prezentácia prototypu <ul style="list-style-type: none"> - zhodnotenie doterajšej spolupráce za zimný semester - identifikácia najkritickejších častí v doterajšej tímovej spolupráci - stanovenie cieľov na letný semester, diskusia o možných zlepšeniach - oficiálna tímová oslava ukončenia etapy analýzy a návrhu |

4 Úlohy v tíme

Dlhodobé úlohy na zimný semester

| Meno člena | Dlhodobá úloha na zimný semester |
|-------------------|--|
| Michal Bigoš | Prezentácia stavu projektu na webe. |
| Szabolcs Bugár | Analýza projektu |
| Vladimír Grlický | Správa tímovej konferencie |
| Ladislav Gyökeres | Analýza projektu |
| Juraj Hájek | Vedúci tímu, integrácia dokumentácie |
| Attila Kúr | Zástupca vedúceho projektu, analýza projektu |

Krátkodobé úlohy na zimný semester

Plán k 14.11.2001:

| Týžden | Krátkodobá úloha | Zodpovedný |
|--------|---|--------------------------------------|
| 1. | Urcenie preferencií študentov na zaradenie do tímov | všetci |
| 2. | Vytvorenie tímu, prvé neoficiálne stretnutie tímu Konzultácia jednotlivých tém so zákazníkmi, výber prioritnej témy Vypracovanie ponuky | všetci |
| 3. | Tlač a zviazanie vypracovanej ponuky | V. Grlický |
| | Odovzdanie ponuky | všetci |
| | Prezentácia ponuky | A. Kúr |
| 4. | Vytvorenie hlbšej analýzy požiadaviek na projekt | Sz. Bugár, A. Kúr |
| | Vytvorenie web-stránky tímu | M. Bigoš |
| | Vytvorenie diskusnej konferencie pre potreby tímu | V. Grlický |
| | Vytvorenie šablóny pre zápisy stretnutí | L. Gyökeres |
| 5. | Dokoncenie hlbšej analýzy požiadaviek na projekt, zaciatoč práce na špecifikácii projektu | Sz. Bugár, L. Gyökeres, A. Kúr |
| | Získanie poznatkov (informácií od súťažiacich aj usporiadateľov) z prípravného a lokálneho kola práve prebiehajúceho na FEI STU | Sz. Bugár, L. Gyökeres, A. Kúr |
| | Získanie poznatkov o systéme odovzdávania zadaní na predmete OS od Ing. Steinmullera. Konzultácia vhodnosti nami navrhovaného riešenia pre potreby cvicení. | J. Hájek |
| | Vytvorenie plánu tímového projektu na zimný semester | |
| | Dokoncenie analýzy existujúcich riešení volne prístupných cez pavucinu | V. Grlický |
| | Zozbieranie informácií týkajúcich sa zabezpečenia celého systému | M. Bigoš |
| 6. | Doplňiť obsah stránky tímu | M. Bigoš |
| | Analýza bezpečnosti odovzdávania pomocou PHP | M. Bigoš |
| | Doplňenie analýzy a hrubého návrhu | Sz. Bugár, L. Gyökeres, A. Kúr |
| | Štúdium možnosti realizácie klienta | J. Hájek |

| Týžden | Krátkodobá úloha | Zodpovedný |
|--------|---|--------------------------|
| | Dokončenie analýzy ďalších existujúcich systémov, ktoré sú prístupné | V. Grlický |
| 7. | Prerobiť stránku tímu a doplniť jej obsah Analýza bezpečnosti odovzdávania pomocou PHP Napísanie častí dokumentácie: návrh architektúry systému výber technológie | Bigoš |
| | Finálne dopracovanie jednotlivých častí dokumentácie: charakteristické typy prog. súťaží – typ ACM, testy, korešpondencné kolo/odovzdávanie na cvičeniach pravidlá ACM, podrobný priebeh ACM dátový model – logický | Bugár Gyökeres Kúr |
| | Kompletizácia dokumentácie Vytvorenie šablóny dokumentácie Finálne dopracovanie jednotlivých častí dokumentácie: dátový model – logický prílohy – ICFP, ICF analýza existujúcich systémov – ICFP, FLP charakteristické typy prog. súťaží – typ OS | Hájek |
| | Finálne dopracovanie jednotlivých častí dokumentácie: dátový model – logický | Grlický |

Zápisy zo stretnutí

Zápis z 1. stretnutia tímu c. 1

| | |
|-------------------------|---|
| Dátum: | 18.10.2001 |
| Miestnosť: | D-103 (FEI STU) |
| Prítomní: | |
| Pedagóg: | prom. mat. L. Hanulová |
| Clenovia tímu: | M. Bigoš, Sz. Bugár, L. Gyökeres, V. Grlický, J. Hájek, A. Kúr |
| Zapisovateľ: | J. Hájek |
| Overil: | L. Gyökeres, |
| Téma stretnutia: | Organizačné pokyny, rozdelenie úloh, analýza problému |

Popis stretnutia

1. Na začiatku stretnutia L. Hanulová zhrnula pripomienky k ponuke tímu. Upozornila na potrebu hlbšej analýzy projektu z pohľadu viacerých typov užívateľov.
2. L. Hanulová upozornila na výhody vytvorenia konferencie s aktuálnym stavom riešených úloh a problémami, s ktorými sme sa počas riešenia stretli. Otázku vytvorenia konferencie sme prediskutovali a schválili.
3. Prediskutovali sme otázku, či je množstvo informácií o podobných projektoch, ktoré sme získali počas vypracovania ponuky, postačujúce. Rozhodli sme sa vyhľadať na internete ďalšie podobné riešenia, kvôli lepšiemu prehľadu v danej oblasti.
4. Sz. Bugár navrhol, že vykoná podrobnú analýzu požiadaviek na systém, nakoľko má so súťažou ACM viacero skúseností. Jeho ponuku sme prijali.
5. Treba analyzovať potreby používateľov systému z hľadiska predmetov vyučovaných na FEI STU.
6. Treba hlbšie analyzovať a sformulovať požiadavky na systém z hľadiska bezpečnosti.
7. Pri analýze sa treba rozhodnúť, ktoré časti systému budeme implementovať v rámci prototypu.
8. Medzi jednotlivých členov tímu sme rozdelili dlhodobé úlohy a to nasledovne:

J. Hájek - vedúci tímu, integrácia dokumentácie.

A. Kúr - zástupca vedúceho tímu.

M. Bigoš - prezentácia stavu projektu na webe.

V. Grlický - správa tímovej konferencie.

Úlohou členov tímu, ktorí majú so súťažou ACM vlastné skúsenosti (Sz. Bugár, L. Gyökeres, A. Kúr), bude analýza projektu.

Úlohy do ďalšieho stretnutia

1. Vytvorenie hlbšej analýzy požiadaviek na projekt (zodp. Sz. Bugár, A. Kúr).
2. Vytvorenie web-stránky tímu (zodp. M. Bigoš).
3. Vytvorenie diskusnej konferencie pre potreby tímu (zodp. V. Grlický).
4. Vytvorenie šablóny pre zápisy stretnutí (zodp. L. Gyökeres).

Zápis z 2. stretnutia tímu c. 1

| | |
|-------------------------|---|
| Dátum: | 25.10.2001 |
| Miestnosť: | Softvérové štúdio 2 (KIVT – FEI STU) |
| Prítomní: | |
| Pedagóg: | prom. mat. L. Hanulová |
| Clenovia tímu: | M. Bigoš, Sz. Bugár, V. Grlický, L. Gyökeres, J. Hájek, A. Kúr |
| Zapisovateľ: | V. Grlický |
| Overil: | J. Hájek |
| Téma stretnutia: | Podrobnejšia analýza problému, špecifikácia požiadaviek |

Popis stretnutia

1. Sz. Bugár, A. Kúr a L. Gyökeres prezentovali svoje poznatky z korešpondenčných a lokálnych kôl organizovaných na FEI STU. Ako najväčšie nevýhody súčasných systémov používaných našou fakultou sa javia:
 - veľká platformová závislosť,
 - náročná údržba a konfigurovateľnosť,
 - nízka automatizácia pred súťažou (prípravná fáza) aj po súťaži (vyhodnotenie výsledkov, ich vhodná prezentácia, štatistika).
2. V. Grlický bližšie oboznámil ostatných členov tímu z doteraz nazbieranými poznatkami o súčasných systémoch na podporu programátorských súťaží. Hlavný dôraz bol kladený na systém PC² (**P**rogramming **C**ontest **C**ontrol system), keďže tento systém je dobre zdokumentovaný, voľne prístupný z pavuciny a je vo veľkej miere používaný na lokálnych a regionálnych kolách v USA alebo Číne.

Medzi najväčšie výhody spomenutého systému patrí najmä:

- niekoľko typov používateľov systému,
- výborná konfigurovateľnosť celého systému (hlavne z pohľadu administrátora),
- prehľadné zobrazovanie informácií o práve prebiehajúcej súťaži,
- platformová nezávislosť (systém je implementovaný v jazyku Java).

K jeho nedostatkom (z pohľadu nami navrhovaného systému) patrí:

- nutnosť inštalácie klientskeho programu na každý súťažný počítač,

- nepoužitelnosť v korešpondencnom kole súťaže (vyplývajúca z predchádzajúceho bodu).
- 3. M. Bigoš umiestnil pripravené stránky tímu do pavuciny (prístupné na adrese <http://www2.dcs.elf.stuba.sk/~team01>) a všetkých zúčastnených oboznámil so štruktúrou týchto stránok.
- 4. Na stretnutí sme jednoznačne odsúhlasili myšlienku „multikontestového“ systému, tzn. celý systém budeme ďalej vyvíjať pre potreby viacerých súťaží.

Úlohy do ďalšieho stretnutia:

1. Dokončenie hlbšej analýzy požiadaviek na projekt, začiatok práce na špecifikácii k projektu (**zodp. Sz. Bugár, L. Gyökeres, A. Kúr**)
2. Získanie poznatkov (informácií od súťažiakov aj usporiadateľov) z prípravného a lokálneho kola práve prebiehajúceho na FEI STU (**zodp. Sz. Bugár, L. Gyökeres, A. Kúr**)
3. Vytvorenie plánu tímového projektu na zimný semester (**zodp. Sz. Bugár, L. Gyökeres, A. Kúr**)
4. Dokončenie analýzy existujúcich riešení voľne prístupných cez pavucinu (**zodp. V. Grlický**)
5. Umiestnenie vytvorenej tímovej konferencie do pavuciny (**zodp. V. Grlický**)
6. Zozbieranie informácií týkajúcich sa zabezpečenia celého systému (2 týždne) (**zodp. M. Bigoš**)

Poznámky:

Zápis zo stretnutia v ďalšom týždni vyhotoví A. Kúr

Zápis z 3. stretnutia tímu c. 1

| | |
|-------------------------|---|
| Dátum a čas: | 30. 10. 2001 (utorok), 12:00 |
| Miestnosť: | Softvérové štúdio 2 (FEI STU) |
| Prítomní: | |
| Pedagóg: | prom. mat. L. Hanulová |
| Clenovia tímu: | M. Bigoš, Sz. Bugár, L. Gyökeres, V. Grlický, J. Hájek, A. Kúr |
| Zapisovateľ: | A. Kúr |
| Overil: | V. Grlický |
| Téma stretnutia: | Analýza súčasného stavu |

Zhodnotenie úloh z minulého stretnutia:

| Osoba | Splnená úloha |
|--------------------------|---|
| Bugár Hájek | konzultácia s Ing. B. Steinmüllerom o možnostiach použitia systému pre programovacie predmety |
| Bugár Gyökeres Kúr | rozšírenie analýzy požiadaviek a zaciatok hrubého návrhu |
| Grlický | správa o analýze existujúcich systémov a dokončenie konferencie na stránke tímu |
| Bigoš | aktualizácia stránky tímu (pridané zápisy a konferencia) |

Opis stretnutia

1. Prediskutovali sme poznatky získané pri konzultáciách s Ing. Branislavom Steinmüllerom, ktoré sa týkali najmä odovzdávania zadaní – zdrojových kódov na predmete Operacné systémy.
2. Clenovia Bugár, Gyökeres a Kúr oboznámili tím s najnovšími bodmi v analýze požiadaviek (rozdiely medzi lokálnym akorešpondenčným kolom a podrobnejší opis používateľa typu pozorovateľ), resp. Grlický a Hájek nám prezentovali vlastnosti ďalších existujúcich systémov.
3. Grlický prezentoval aj konferenciu na stránke tímu a taktiež jej bezpečnosť, pričom priebežne oboznamoval členov tímu so spôsobom jej použitia.
4. S vedúcim pedagógom pani Hanulovou sme sa dohodli, že priebežné výsledky všetkých vytváraných dokumentov jej pošleme den pred naším stretnutím formou elektronickej pošty. Cieľom tejto aktivity je zefektívnenie priebehu tímových stretnutí, keďže sa pre nu vytvorí možnosť predbežného preštudovania materiálov, o ktorých bude rec na nasledujúcom stretnutí.
5. Pani Hanulová nám ešte pripomenula nedostatky našej stránky v porovnaní so stránkami ostatných tímov (my nemáme tabuľku splnených, resp. zadaných úloh a ani plán projektu nie

je na našej stránke). Preto sme sa dohodli na castejšej komunikácii medzi členmi tímu (nielen formou e-mailu, ale hlavne využitím konferencie na našej stránke), aby obsah stránky bol vždy aktuálny.

Úlohy do ďalšieho stretnutia:

| Osoba | Zadaná úloha | Termín |
|--------------------------|---|-----------|
| Bigoš | doplniť obsah stránky tímu a analýza bezpečnosti odovzdávania pomocou PHP | 8.11.2001 |
| Bugár Gyökeres Kúr | doplnenie analýzy a hrubého návrhu | 7.11.2001 |
| Hájek | štúdium možnosti realizácie klienta | 8.11.2001 |
| Grlický | dokončenie analýzy ďalších existujúcich systémov, ktoré sú prístupné | 8.11.2001 |

Poznámky:

Ďalšie stretnutie bude vo štvrtok 8.11.2001 o 12:00 v Softvérovom štúdiu 2.

Na budúcom stretnutí bude zapisovať: L. Gyökeres.

Zápis zo 4. stretnutia tímu c. 1

| | |
|-------------------------|---|
| Dátum a čas: | 08. 11. 2001 (štvrtok), 12:00 |
| Miestnosť: | Softvérové štúdio 2 (FEI STU) |
| Prítomní: | |
| Pedagóg: | prom. mat. L. Hanulová |
| Clenovia tímu: | M. Bigoš, Sz. Bugár, L. Gyökeres, V. Grlický, J. Hájek, A. Kúr |
| Zapisovateľ: | M. Bigoš |
| Overil: | J. Hájek |
| Téma stretnutia: | Analýza súčasného stavu projektu pre odovzdávaním dokumentácie. |

Zhodnotenie úloh z minulého stretnutia:

| Osoba | Úloha | Zadané | Stav |
|--------------------------|--|--|--|
| Hájek | štúdium možnosti realizácie klienta | Stretnutie 3 30/10/2001 | splnené k 8/11/2001 |
| Bugár Gyökeres Kúr | doplnenie analýzy a hrubého návrhu | Stretnutie 3 30/10/2001 | splnené k 8/11/2001 |
| Grlický | dokoncenie analýzy ďalších existujúcich systémov, ktoré sú prístupné | Stretnutie 3 30/10/2001 | splnené k 8/11/2001 |
| Bigoš | aktualizácia stránky tímu analýza bezpečnosti odovzdávania pomocou PHP | Stretnutie 3 30/10/2001 Stretnutie 2 30/10/2001 | splnené k 8/11/2001 nesplnené k 8/11/2001 |

Opis stretnutia

1. Clenovia Grlický a Hájek boli počas stretnutia na konzultácií upána Trabalku, aby sa oboznámili s jeho požiadavkami na systém pre fakultný predmet FLP, ako aj s systémom, doposiaľ používaným na odovzdávanie na tomto predmete. Po návrate z konzultácie oboznámili ostatných členov tímu o získané informácie.
2. V náväznosti na bod jedna sa rozpútala diskusia, ktorej témou bol hlavne návrh architektúry systému, vzhľadom na možnosti odovzdávania zadaní. Všetci členovia tímu nakoniec dospeli k názoru, že daný problém si vyžaduje hlbšie rozobratie možných riešení, ktoré dostal za úlohu spraviť Bigoš.
3. Vedúca pani Hanulová nám priniesla ukázať, príklady dokumentácií minuloročných prác. Vyvarovala nás voci chybám, ktoré sa v nich vyskytli. Dalej spripomenkovala tímový web stránku. Jej výhrady sa týkali hlavne aktuálnosti, hlavne čo sa týka zápisov zo stretnutí, ďalej lepšieho – prehľadnejšieho zobrazenia úloh jednotlivých členov tímu aj s dátumami zadania a ukončenia úlohy, ako aj prepracovania podrobnejšieho plánu projektu.

4. Všetci sme sa dohodli na štruktúre dokumentácie a rozdelili sme si spísanie jednotlivých častí. Rozdelenie je zahrnuté vrámci úloh do ďalšieho stretnutia. Taktiež sme sa dohodli na výnimocnom stretnutí na utorok 13/11/2001 z dôvodu kontroly dokumentácie pred jej odovzdaním.

Úlohy do ďalšieho stretnutia:

| Oscba | Zadaná úloha | Termín |
|--------------------------|---|------------|
| Bigoš | Prerobiť stránku tímu a doplniť jej obsah Analýza bezpečnosti odovzdávania pomocou PHP Napísanie častí dokumentácie: návrh architektúry systému výber technológie | 13/11/2001 |
| Bugár Gyökeres Kúr | Finálne dopracovanie jednotlivých častí dokumentácie: charakteristické typy prog. súťaží – typ ACM, testy, korešpondenčné kolo/odovzdávanie na cvičeniach pravidlá ACM, podrobný priebeh ACM dátový model – logický | 13/11/2001 |
| Hájek | Kompletizácia dokumentácie Vytvorenie šablóny dokumentácie Finálne dopracovanie jednotlivých častí dokumentácie: dátový model – logický prílohy – ICFP, ICF analýza existujúcich systémov – ICFP, FLP charakteristické typy prog. súťaží – typ OS | 13/11/2001 |
| Grlický | Finálne dopracovanie jednotlivých častí dokumentácie: dátový model – logický analýza existujúcich systémov – PC ² , Ural, zhrnutie výhod a nevýhod, porovnanie riešení | 13/11/2001 |

Poznámky:

V utorok 12/11/2001 o 13:00 sa uskutoční neoficiálne stretnutie členov za účelom zhodnotenia stavu dokumentácie pred finálnym odovzdaním. Členovia sa stretnú u vedúcej projektu L. Hanulovej.

Ďalšie stretnutie bude vo štvrtok 15/11/2001 o 12:00 v Softvérovom štúdiu 2.

Na budúcom stretnutí bude zapisovať: Sz. Bugár.

Zápis z 5. stretnutia tímu c. 1

| | |
|---------------------|-------------------------------|
| Dátum a čas: | 15. 11. 2001 (štvrtok), 12:00 |
| Miestnosť: | Softvérové štúdio 2 (FEI STU) |

| | |
|------------------|---|
| Prítomní: | |
| Pedagóg: | prom. mat. L. Hanulová |
| Clenovia tímu: | M. Bigoš, Bugár Sz., L. Gyökeres, V. Grlický, J. Hájek, A. Kúr |

| | |
|---------------------|-----------|
| Zapisovateľ: | Bugár Sz. |
| Overil: | J. Hájek |

| | |
|-------------------------|--|
| Téma stretnutia: | Odovzdávanie dokumentácie a urcenie ďalších úloh |
|-------------------------|--|

Zhodnotenie úloh z minulého stretnutia

| Osoba | Úloha | Zadané | Stav |
|--------------------------|---|----------------------------|-------------------------|
| Bigoš | Prerobiť stránku tímu a doplniť jej obsah Analýza bezpečnosti odovzdávania pomocou PHP Napísanie častí dokumentácie: návrh architektúry systému výber technológie | Stretnutie 4 8/11/2001 | splnené k 13/11/2001 |
| Bugár Gyökeres Kúr | Finálne dopracovanie jednotlivých častí dokumentácie: charakteristické typy prog. súťaží – typ ACM, testy, korešpondenčné kolo/odovzdávanie na cvičeniach pravidlá ACM, podrobný priebeh ACM dátový model – logický | Stretnutie 4 8/10/2001 | splnené k 13/11/2001 |
| Hájek | Kompletizácia dokumentácie Vytvorenie šablóny dokumentácie Finálne dopracovanie jednotlivých častí dokumentácie: dátový model – logický prílohy – ICFP, ICF analýza existujúcich systémov – ICFP, FLP charakteristické typy prog. súťaží – typ OS | Stretnutie 4 8/10/2001 | splnené k 13/11/2001 |
| Grlický | Finálne dopracovanie jednotlivých častí dokumentácie: dátový model – logický analýza existujúcich systémov – PC ² , Ural, zhmutie výhod a nevýhod, porovnanie riešení | Stretnutie 4 08/10/2001 | splnené k 13/11/2001 |

Opis stretnutia

1. Na zaciatku stretnutia tím nacas odovzdal dokumentáciu vedúcej projektu (pani Hanulovej) a druhému tímu (tím c. 6) podla vopred stanovených podmienok.
2. Zhrnuli sme poznatky a cenné skúsenosti, ktoré sme získali pocas prvej fázy na predmete Tímový projekt. Dohodli sme sa, že integráciu výslednej dokumentácie treba robiť skôr, než sme teraz robili. Diskutovali sme aj o tom, čo sme urobili dobre a čo zle počas tejto doby.
3. Dokumentáciu, ktorú sme dostali od konkurenčného tímu, so sebou odniesol vedúci tímu, Juraj Hájek. Ostatní členovia tímu preštudujú elektronickú podobu dokumentácie, ktorá sa nachádza na stránke konkurenčného tímu (tím c. 6).
4. Dohodli sme sa na tom, že najprv budeme implementovať používateľské rozhranie systému a časti, ktoré zabezpečujú vnútornú logiku aplikácie budú implementované až v letnom semestri.
5. Náš vedúci projektu (Juraj Hájek) navrhol, aby sa vedúcim tímu stal Vladimír Grlický, lebo sa lepšie vyzná v tejto problematike. Tento návrh sme aj schválili.
6. Dohodli sme sa na neformálnom stretnutí, ktoré sa bude konať 20.11.2001 vutorok pred predmetom RPvI, kde prediskutujeme, čo a ako ďalej spravíme v projekte.
7. Michal Bigoš požičal svoju knihu o PHP dvom členom tímu, ktorí bývajú na spoločnej izbe (B.Sz., K.A.), aby mohli rýchlejšie osvojiť vedomosti v oblasti PHP.

Úlohy do ďalšieho stretnutia

| Osoba | Zadaná úloha | Termín |
|--------|--|------------|
| všetci | dohodnúť sa o rozdelenie podielov na vykonanej práci a úsilia v predchádzajúcej obdobe tímovej práci | 20/11/2001 |

Poznámky:

Ďalšie stretnutie bude vo štvrtok 22/11/2001 o 12:00 v Softvérovom štúdiu 2.

Na budúcom stretnutí bude zapisovať: Ladislav Gyökeres.

Zápis zo 7. stretnutia tímu c. 1

| | |
|-------------------------|--|
| Dátum a čas: | 22. 11. 2001 (štvrtok), 12:00 |
| Miestnosť: | Softvérové štúdio 2 (FEI STU) |
| Prítomní: | |
| Pedagóg: | prom. Mat. L. Hanulová |
| Clenovia tímu: | Bugár Sz., L. Gyökeres, J. Hájek, A. Kúr |
| Zapisovateľ: | L. Gyökeres |
| Overil: | J. Hájek |
| Téma stretnutia: | Vyhodnotenie dokumentácie a určenie ďalších úloh |

Zhodnotenie úloh z minulého stretnutia

| Osoba | Úloha | Zadané | Stav |
|------------------|---|---|-----------------------------------|
| Všetci | Dohodnúť sa o rozdelení podielov na vykonanej práci | Stretnutie 5 15/11/2001 | nesplnené |
| Bugár Kúr | Návrh používateľského rozhrania pre súťažiach | Stretnutie 6 29/11/2001 | splnené k 29/11/2001 |
| Gyökeres | Návrh používateľského rozhrania na tvorbu súťaží | Stretnutie 6 29/11/2001 | splnené k 29/11/2001 |
| Bigoš Grlický | Finalizácia databázy | Stretnutie 6 29/11/2001 | ??? neprítomní na stretnutí |
| Hájek | Štúdium možnosti odosielania mailu cez WWW | Stretnutie 6 29/11/2001 | úloha bola zmenená |
| Hájek | Implementácia kontroly zdrojových textov vzhľadom na povolené funkcie pre jazyk C | Neoficiálne stretnutie 26/11/2001 | splnené |

Opis stretnutia

1. Vedúca projektu vyjadrila svoje pripomienky k aktuálnosti prezentácie tímu na webe, ďalej k štruktúre a obsahu zápisov zo stretnutí, ako aj k forme prezentácie pridelených úloh pre jednotlivých členov tímu. Keďže otázkou kvality zápisov a pridelených úloh sme sa zaoberali už na viacerých stretnutiach v minulosti, bude potrebné sa na tento problém bližšie zamerať.
2. S vedúcou projektu sme prediskutovali mieru súcinnosti členov a mieru spolupráce tímu ako celku. Bližšie sme jej popísali spoluprácu členov tímu medzi pravidelnými oficiálnymi stretnutiami.
3. Podobne ako na stretnutí c. 6 sme zhrnuli naše názory na dokumentáciu konkurenčného tímu. Vyjadrili sme sa k posudku, ktorý na náš projekt vypracoval tím

c. 6, ďalej sme sa zaoberali nedostatkami nášho vlastného posudku (chýbajúce komplexné záverečné hodnotenie kvality vypracovaného dokumentu, nedostatočné zdôraznenie dobrých stránok dokumentu).

4. Prítomní členovia tímu postupne prezentovali riešenia úloh, ktoré im boli pridelené na minulom stretnutí.

A. Kúr a L. Gyökeres prezentovali svoje návrhy pridelených častí používateľského rozhrania.

J. Hájek prezentoval program na kontrolu používania zakázaných knižníc a funkcií pre jazyk C a oboznámil prítomných s použitím programu.

5. Zhrnuli sme množstvo práce, ktoré vykonali členovia tímu pri analýze a návrhu systému a vypracovaní dokumentácie a dohodli sme sa na rozdelení bodov za odovzdanú dokumentáciu:

18b S. Bugár, J. Hájek, A. Kúr

16b L. Gyökeres, V. Grlický

14b. M. Bigoš

Úlohy do ďalšieho stretnutia

| Osoba | Zadaná úloha | Termín |
|------------------|--|-----------|
| Bugár Kúr | Implementácia narhnutého používateľského rozhrania pre súťažiach | 6/12/2001 |
| Gyökeres | Implementácia navrhnutého používateľského rozhrania na tvorbu súťaží | 6/12/2001 |
| Hájek | Skontaktovať sa s neprítomnými členmi, zistiť stav v ktorom sa nachádza implementovaná databáza. | 6/12/2001 |
| Hájek | Realizácia odovzdávania zadaní cez web a ich archivácia. | 6/12/2001 |
| Bigoš Grlický | Na pridelení úloh sa dohodneme po skontaktovaní sa a povzajomnej konzultácii. | |

Poznámky:

Program na kontrolu používaných funkcií jazyka C sa nachádza na servri labss2 v adresári /home/users/team01/team01/sources/code_checkers/c_checker.pl.

Ďalšie stretnutie bude vo štvrtok 29/11/2001 o 12:00 v Softvérovom štúdiu 2.

Na budúcom stretnutí bude zapisovať: V. Grlický.

Zápis z 8. stretnutia tímu c. 1

| | |
|---------------------|-------------------------------|
| Dátum a čas: | 6. 12. 2001 (štvrtok), 12:00 |
| Miestnosť: | Softvérové štúdio 2 (FEI STU) |

| | |
|------------------|---|
| Prítomní: | |
| Pedagóg: | prom. Mat. L. Hanulová |
| Clenovia tímu: | Sz. Bugár, V. Grlický, L. Gyökeres, J. Hájek, A. Kúr |

| | |
|---------------------|-------------|
| Zapisovateľ: | V. Grlický |
| Overil: | L. Gyökeres |

| | |
|-------------------------|--|
| Téma stretnutia: | Rozdelenie úloh pre úspešné dokončenie prototypu |
|-------------------------|--|

Zhodnotenie úloh z minulého stretnutia

| Osoba | Úloha | Zadané | Stav |
|----------------|--|------------|--------------|
| Bugár, Kúr | Implementácia navrhnutého používateľského rozhrania pre súťažiach | 29/11/2001 | rozpracované |
| Gyökeres | Implementácia navrhnutého používateľského rozhrania na tvorbu súťaží | 29/11/2001 | rozpracované |
| Hájek | Realizácia odovzdávania zadaní cez web a ich archivácia | 29/11/2001 | splnené |
| Bigoš, Grlický | Implementácia tried pracujúcich nad databázou | 22/11/2001 | rozpracované |

Opis stretnutia

1. Vedúca projektu znovu vyjadrila svoje pripomienky k aktuálnosti prezentácie tímu na webe.
2. Prítomní členovia tímu postupne prezentovali riešenia úloh, ktoré im boli pridelené na minulom stretnutí.

J. Hájek prezentoval skript v jazyku PHP umožňujúci prebratie súboru od súťažiaceho a jeho prenesenie na server („upload“ riešenia cez webovské rozhranie).

V. Grlický predstavil prvotnú verziu dokumentu obsahujúceho tzv. štábnu kultúru pre členov tímu c. 1. Najdôležitejšie časti tohto dokumentu boli aj rozobraté. Členovia tímu vytvorené predpisy nachádzajúce sa v tomto dokumente odsúhlasili.

V. Grlický oboznámil ostatných členov tímu mailom, ktorý mu po vzájomnej korešpondencii poslal jeden z tvorcov systému IPSC (systém podporujúci programátorské súťaže). Mail obsahuje skúsenosti tvorcov IPSC implementáciou tohto systému, pričom poukazuje na výhody, resp. nevýhody niektorých použitých riešení.

Všetci zúčastnení členovia sa dohodli na spôsobe ukladania súborov prijatých od súťažiacich a zo zvolených alternatív vybrali hierarchickú architektúru (druhou alternatívou bola „flat“ architektúra).

Úlohy do ďalšieho stretnutia

| Osoba | Zadaná úloha | Termín |
|------------------|---|------------|
| Bugár Kúr | Zjemnenie doteraz navrhnutého používateľského rozhrania pre súťažiach | 14/12/2001 |
| Gyökerez | Zjemnenie doteraz navrhnutého používateľského rozhrania na tvorbu súťaží | 14/12/2001 |
| Hájek | Rozpracovanie odovzdávania zadaní pomocou webovského rozhrania a ich archivácie | 13/12/2001 |
| Bigoš Grlický | Integrácia používateľských rozhraní ostatných členov tímu | 16/12/2001 |

Poznámky:

Ďalšie stretnutie bude vo štvrtok 13.12.2001 o 12:00 v Softvérovom štúdiu 2

Na budúcom stretnutí bude zapisovať **A. Kúr**

Výsledný produkt
Softvérový systém

Obsah

| | | |
|----------|---|------------|
| 1 | ANALÝZA | 2 |
| 1.1 | CHARAKTERISTICKÉ PRÍPADY POUŽITIA | 2 |
| 1.1.1 | <i>Programátorská súťaž ACM</i> | 2 |
| 1.1.2 | <i>Korešpondenčné kolo ACM</i> | 3 |
| 1.1.3 | <i>Testy</i> | 5 |
| 1.2 | ZAKOMPONOVANIE SYSTÉMU DO VYUCOVACIEHO PROCESU | 6 |
| 1.2.1 | <i>Online overovanie znalostí</i> | 6 |
| 1.2.2 | <i>Korešpondenčné odovzdávanie zadaní</i> | 6 |
| 1.2.3 | <i>Testy</i> | 7 |
| 1.3 | POPIS EXISTUJÚCICH SYSTÉMOV | 8 |
| 1.3.1 | <i>Súčasný spôsob vyhodnocovania súťaže ACM</i> | 8 |
| 1.3.2 | <i>FLP – systém na odovzdávanie zadaní</i> | 9 |
| 1.3.3 | <i>PC² -The Programming Contest Control System</i> | 10 |
| 1.3.4 | <i>UPC2001 Internet Contest (Ural Programming Contest)</i> | 11 |
| 1.3.5 | <i>Ural State University Problem Set Archive with Online Judge System</i> | 11 |
| 1.3.6 | <i>IPC – RoboJudge</i> | 14 |
| 1.4 | ŠPECIFIKÁCIA POŽIADAVIEK | 16 |
| 1.4.1 | <i>Popis používateľov</i> | 16 |
| 1.4.2 | <i>Špecifikácia funkcií systému</i> | 17 |
| 1.4.3 | <i>Priorita funkcií</i> | 18 |
| 2 | HRUBÝ NÁVRH | 20 |
| 2.1 | ARCHITEKTÚRA SYSTÉMU | 20 |
| 2.1.1 | <i>Architektúra systému z pohľadu použitej technológie</i> | 20 |
| 2.1.2 | <i>Architektúra systému z pohľadu komunikácie klienta so serverom</i> | 24 |
| 2.1.3 | <i>Architektúra systému z pohľadu modulárnosti a škálovateľnosti</i> | 25 |
| 2.2 | VÝBER TECHNOLÓGIE | 26 |
| 2.2.1 | <i>Zhrnutie požiadaviek</i> | 27 |
| 2.2.2 | <i>Možnosti realizácie</i> | 27 |
| 2.2.3 | <i>Možné technológie</i> | 28 |
| 2.2.4 | <i>Výber technológie pre implementáciu systému</i> | 28 |
| 2.3 | LOGICKÝ MODEL ÚDAJOV | 31 |
| 2.3.1 | <i>Diagram modelu údajov</i> | 31 |
| 2.3.2 | <i>Popis entít</i> | 31 |
| 2.3.3 | <i>Opis vzťahov</i> | 36 |
| | LITERATÚRA | 38 |
| | PRÍLOHY | 408 |

1 Analýza

V tejto časti sa nachádza analýza systému. Pri analýze sme sa zamerali nielen na už existujúcu súťaž ACM, ale zamýšľali sme sa aj nad tým, aké požiadavky na systém môžu v budúcnosti vzniknúť v rámci iných súťaží, resp. pri použití na cvičeniach na FEI STU.

1.1 Charakteristické prípady použitia

1.1.1 Programátorská súťaž ACM

Programátorská súťaž ACM (ACM Programming Contest) je celosvetová súťaž usporiadaná americkou organizáciou Association for Computing Machinery, počítačovou organizáciou, ktorá existuje od roku 1947. Súťaž je určená 1-3 členným tímom študentov vysokých škôl po celom svete a prebieha v niekoľkých kolách. Pravidlá súťaže možno nájsť na stránke <http://contest.felk.cvut.cz/rules.html>. K dispozícii sú aj skrátené pravidlá v slovenskom jazyku na adrese <http://www.dcs.elf.stuba.sk/acm/rulesacm.html>, ktoré uvádzame v prílohe A.

Pocas súťaže je ohrančený počet odovzdaní riešenia toho istého problému. Tento počet je obvykle štyri odovzdávania, ale v niektorých prípadoch môže byť aj zmenený.

Pocas súťaže je riešiteľským tímom sprístupnené ich predbežné poradie spolu s priebežnými výsledkami ako sú celkový absolútny čas, počet odovzdaných zadaní, počet akceptovaných problémov. V poslednej hodine súťaže je však táto možnosť zrušená z dôvodu urobiť súťaž ešte napínavejšou.

Cielom lokálneho kola ACM súťaže je určiť tie najlepšie tímy spomedzi všetkých prihlásených. Tímy sa môžu prihlasovať z každej fakulty univerzity (aj keď v prípade STU v Bratislave sa naša fakulta prihlásila vždy ako jediná), a na stredoeurópske kola sa dostanú iba tí najlepší z celej univerzity.

1.1.1.1 Vyhodnocovanie zdrojových programov

Možné výstupy testovaného programu sú nasledovné:

- **Accepted** = program dal správny výstup
- **Wrong Answer** = program nedodal správny výstup
- **Presentation Error** = výstup je v podstate správny, ale chybné formátovaný
- **Time Limit Exceeded** = program neskončil v časovom limite
- **Run-time Error** = program sa zrútil alebo skončil s nenulovým návratovým kódom
- **Compile Error** = program sa nepodarilo skompilovať
- **Invalid Library Call** = program používa nepovolené operácie

Aby súťažiaci mohli čo najrýchlejšie lokalizovať chyby v svojom programe, je potrebné presne zadefinovať prioritu, resp. poradie, v ktorom sa jednotlivé odpovede generujú počas vyhodnocovania.

Cím je číslo odpovede vyššie, tým sú súťažiaci pravdepodobne bližšie k riešeniu. Poradové číslo im naznačuje aj to, že predchádzajúce chyby sú opravené, resp. že ich riešenie prešlo cez kontrolu prechádzajúcich chybových hlásení. Napríklad, ak dostanú odpoveď s číslom 5 (**Wrong Answer**), z toho súťažiaci vedia, že program skončil v danom časovom limite, ale nedával správny výstup. Ale táto správa naznačuje aj to, že ich výstupný formát ešte nemusí byť správny. Cieľom je, aby číslo odpovede bolo 7 (**Accepted**). V tomto prípade ďalšie zjemňovanie riešenia nemá zmysel, lebo program bol akceptovaný, prijatý.

Kvôli lepšiemu pochopeniu uvedieme príklad:

Pri správnom výstupe programu je jednoznačná odpoveď **Accepted**. Toto naznačuje, že dané riešenie problému vyhovuje všetkým podmienkam stanoveným sudcami. Ak je odpoveďou **Presentation Error**, znamená to, že riešenie nespĺňa požiadavky na formátovanie výstupu. V tomto prípade musia riešitelia skontrolovať také podrobnosti, či ich program nevynecháva napríklad bodku (.), ciarku (,), medzeru () atď. Pritom ale údaje, ktoré sa na výstupe objavujú, sú správne, len majú nesprávny formát. Ak odpoveď je **Wrong Answer**, tak na výstupe sú chybné údaje. Po odstránení tejto chyby ešte môžu dostať ako odpoveď **Presentation Error**, lebo odpoveď typu **Wrong Answer** skontroluje iba chybu vo výstupe, ale neberie do úvahy chybné formátovanie. V tomto procese sa postupuje ďalej takým istým spôsobom, čiže ak odpoveď je **Time Limit Exceeded**, tak po oprave tejto chyby sa ešte môže vyskytnúť aj chybové hlásenie **Wrong Answer, Presentation Error**. Ak odpoveď je **Invalid Library Call**, to znamená, že program chce používať nepovolené knižnice, ale neznamená zas, že zdrojový program je skompilovateľný. Takto môžeme ďalej postupovať.

Možno najväčším ohrančením lokálnych kôl súťaží ACM je práve to, že každý tím má k dispozícii iba jeden počítač, nezávisle na tom, koľko má členov. Aby sa zefektívnila paralelná práca všetkých členov tímu, na lokálnych kolách súťaže ACM na FEI STU bývala v posledných rokoch umožnená tlač vytváraných programov počas riešenia problémov. Riešitelia mali k dispozícii skript, ktorým posielali zdrojový kód (súbor so zadaným menom) na sieťovú tlačiarňu HP LaserJet 2200. Potom kým jeden z tímu sa zaoberal doladovaním vytlačenej kódu na papier, k počítaču mohol prísť ďalší člen, ktorý už mohol programovať riešenie niektorého iného problému. Na zabezpečenie toho, aby vytlačený zdrojový kód sa vždy dostal do správnych rúk (t.j. k tímu, ktorý daný kód dal vytlačiť), bola nevyhnutná prítomnosť dozorca – organizátora pri tlačiarni. Hoci to znamená prídavnú úlohu pre organizátora, z vlastných skúseností vieme, že možnosť tlače výrazne uľahčuje prácu súťažiacich. V duchu pomôcť riešiteľom súťaží ACM sme sa preto rozhodli, že túto funkciu musí podporovať aj náš systém.

1.1.2 Korešpondenčné kolo ACM

Najdôležitejšie dôvody, prečo bolo na našej fakulte usporiadané korešpondenčné kolo ACM boli aby sa existencia súťaže dostala do povedomia viacerým študentom a aby sa zabezpečila príprava na ďalšie lokálne kolo, ktoré sa plánovalo na jeseň.

Študenti, resp. ďalšie osoby, ktoré sa chceli zúčastniť na korešpondenčnom kole, mohli na oficiálnej stránke FEI STU vyplniť formulár, kde zadali meno, odbor a e-mailovú schránku členov, kam boli neskoršie posielané potrebné informácie pre súťažiacich. Po odštartovaní súťaže sa jednotlivé príklady dali získať na tej istej strane v rôznych formátoch (.html, .doc atd.). Z dôvodu rôzneho charakteru korešpondenčného a lokálneho kola boli vykonané viaceré modifikácie v pravidlách vzhľadom na lokálne kolo.

Tímy mohli riešiť príklady kdekolvek. Riešenia sa potom mali odovzdať na vopred určenú e-mailovú adresu. Zdrojové súbory museli poslať ako príloha listu a ako predmet (mail subject) bolo treba napísať SUBMITXY, kde X sa rovná číslu tímu a Y je označenie príkladu, ktorý sa odovzdáva. Napríklad ak tím s číslom 1 chcel odovzdať príklad F, tak predmet listu mal byť „SUBMIT1F“. Keď na druhej strane rozhodca zistil príchod nového e-mailu, dostal tím odpoveď, že jeho odovzdanie bolo zaregistrované. Po uplynutí ďalšieho času dostal tím odpoveď na otestovaný program (Accepted, Wrong Answer atd.).

Ak mali členovia tímu nejaké nejasnosti, mohli napísať svoje otázky na tú istú adresu, na ktorú odosielajú zadania. Odpoveď dostali od toho istého rozhodcu. Priebežné výsledky mohli sledovať tiež na stránke súťaže. Keďže prístup na túto stránku nebol chránený heslom, priebeh súťaže mohli sledovať aj osoby, ktoré sa na nej priamo nezúčastnili.

Približne týždeň pred skončením súťaže sa prestali zverejňovať aktuálne výsledky, aby súťažiaci boli motivovaní, že treba pridať, aby ich ostatné tímy ich nepredbehli, resp. aby oni predbehli ostatných.

Po skončení súťaže boli zverejnené oficiálne výsledky a informácie o slávnostnom odovzdávaní diplomov, resp. vyhlásení výsledkov.

Toto kolo bolo v poslednej chvíli ešte predĺžené, lebo viacerí študenti sa prihlásili tesne pred skončením pôvodného termínu a nemali by dostatok času na vyriešenia príkladov. Súťažiaci sa mohli prihlasovať aj v priebehu súťaže.

Vítaný tím korešpondenčného kola, postupoval do stredoeurópskeho kola v ďalšom semestri a mal pomôcť pri organizácii lokálneho kola, ktoré sa uskutočnilo v nasledujúcom semestri v októbri.

Medzi klady tejto súťaže patrí najmä:

- pohodlné odovzdávanie príkladov
- pohodlné sledovanie priebežných výsledkov
- ľubovoľne vybrané implementačné prostredie
- užívateľsky príjemné prihlasovanie tímov

Záporné skúsenosti:

- niekedy veľmi pomalá odozva na odovzdané riešenia
- cez víkend žiadne sa nekonalo žiadne vyhodnocovanie

- drobné problémy s novým systémom

1.1.3 Testy

Príkladom otestovania vedomostí pomocou výberu alternatív na celosvetovej „platforme“ je školiaci program firmy Cisco, ktorý sa vo svete používa na kurze Regional Cisco Networking Academy Program.

Priebeh testovania

Každý študent má zriadené konto na serveri Cisca. Kto chce preštudovať látku, môže sa prihlásiť (prístupové meno a heslo) na webovskú adresu kurzu. Potom môže vybrať aj kapitolu, ktorú chce preštudovať. Ak je aktivované aj testovanie, môže taktiež otestovať svoje vedomosti formou výberového testu (musí vybrať, koľko otázok chce vidieť na obrazovke naraz). Na každú otázku sú poskytnuté štyri možné odpovede, z ktorých je vždy jedna správna. Študent nemusí vybrať žiadnu odpoveď zponúknutých – „body sa pridávajú“. Jeho vedomosti sú potom ohodnotené programom (počet získaných bodov, resp. sú vyjadrené v percentách). Správne alebo nesprávne odpovede nie sú naznačené ani počas testu, ani po jeho ukončení. V centrálnej databáze je evidovaný každý účastník, študent akadémie. Podľa získaných výsledkov je možnosť vytlačiť vysvedčenie pomocou systému, resp. diplom, ktorý je dôkazom toho, že daná osoba absolvovala daný semester.

1.2 Zakomponovanie systému do vyučovacieho procesu

V tejto časti si bližšie popíšeme, ako sa dá navrhovaný systém zakomponovať do vyučovacieho procesu na FEI STU. Predpokladáme, že zmeny v návrhu systému tak, aby bol použiteľný pre potreby fakulty, budú minimálne, pričom sa jeho použiteľnosť výrazne zvýši.

1.2.1 Online overovanie znalostí

Ekvivalentom programátorskej súťaže, tak, ako je popísaná v na str. 2, je spôsob testovania znalostí študentov, ako je v súčasnosti realizovaný napr. na predmete Operacné systémy (OS).

V rámci tohoto predmetu majú študenti v priebehu dopredu stanoveného času za úlohu vypracovať program podľa zadaných požiadaviek. Systém je teda do istej miery podobný ako pri súťaži ACM. Hlavný rozdiel je v spôsobe odovzdávania – kým na ACM majú možnosť viacerých odovzdaní a okamžitú odozvu o správnosti svojich riešení, na OS (ani ďalších predmetoch) zatiaľ nemajú. Cviiaci by zas pravdepodobne uvítali možnosť jednoduchým spôsobom sledovať priebeh odovzdávania zadaní.

Charakteristickým znakom týchto testov je, že študenti pracujú pod psychickým a časovým tlakom, teda spôsob práce študentov so systémom musí byť čo najrýchlejší a najjednoduchší.

V súčasnosti je odovzdávanie realizované ako sústava UNIX-ovo orientovaných skriptov, čo zapríčiniuje, že sa tento spôsob odovzdávania nedá jednoduchým spôsobom prispôbiť aj pre potreby predmetov, ktoré sa vyucujú pod inými ako UNIX-ovými operacnými systémami.

Po konzultácii s Ing. B. Steinmüllerom sme sa dohodli na tom, že súťažiaci by mali mať ponechanú možnosť odovzdávať vyriešené zadania z príkazového riadku, nakoľko sa jedná o najrýchlejší a najjednoduchší spôsob komunikácie, i keď menej príjemný z hľadiska používateľa.

1.2.2 Korešpondenčné odovzdávanie zadaní

Ekvivalentom ku korešpondenčnej súťaži je v podmienkach cvicení na fakulte systém, keď študenti samostatne doma pracujú na pridelených zadaniach a do určitého termínu musia odovzdať výsledky svojej práce cviciacemu.

Z pedagogických dôvodov je potrebné, aby hodnotenie zadaní nebolo úplne automatizované, ale aby hodnotenie bolo ponechané na cviciaceho – napr. stupeň, ako zodpovedá programátorský štýl vyučovanej paradigmy programovania atď.

Systém tu teda zohráva len úlohu podporného nástroja, ktorý cviciacemu pomáha automatizovať niektoré rutinné činnosti, archivovať odovzdané zadania a študentom uľahčuje komunikáciu s cviciacim.

1.2.3 Testy

Rozšírenou metódou overovania znalostí študentov sú rôzne testy, keď študenti musia označiť jednu, alebo viac správnych odpovedí na danú otázku.

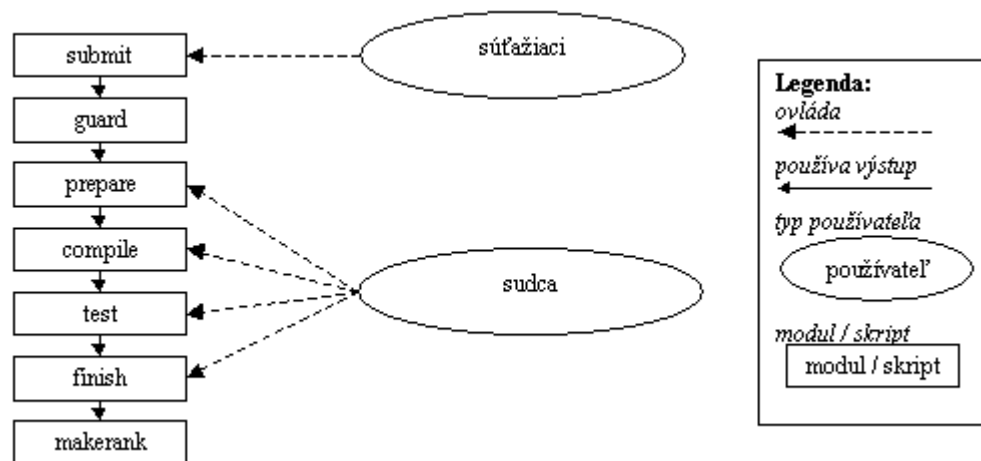
Na našej fakulte už bol používaný testovací program na skúške z predmetu *Úvod do trhovej ekonomiky* (prednášateľ: doc. Sládková). Študenti pomocou počítačového programu odpovedali na otázky výberom zo štyroch alternatív. Podľa odpovede potom program prideloval body k výslednému počtu bodov.

S podobným systémom sme sa stretli aj na cvičeniach z predmetov *Elektronické obvody I, II*. Na každom cvičení študenti postupne vypracovávali test na počítači. Systém fungoval skoro tak isto, ako na predmete *Úvod do trhovej ekonomiky*.

1.3 Popis existujúcich systémov

1.3.1 Súčasný spôsob vyhodnocovania súťaže ACM

V súčasnosti sa na FEI STU používa systém zdedený z predchádzajúcich rokov, ktorý tím organizátorov doplnil vlastnými funkciami. Štruktúra celého systému je znázornená na Obr. c. 3.



Obr. c. 3: Štruktúra súčasného systému na odovzdávanie a vyhodnocovanie riešení

Práca so systémom

Súťažiaci pracujú na jednotlivých pracovných staniciach. Používajú rôzne editory na editáciu zdrojových programov. Zdrojový program môžu odovzdať, z príkazového riadku príkazom *submit file* (pritom *file* musí byť meno odovzdávaného súboru). Tieto názvy sú dané oznaceniami zadania. *submit* je meno skriptu, ktorý daný súbor skopíruje do vopred zadefinovaného adresára tým, že meno súboru je premenované podľa nasledujúcich príznakov: dátum, čas, oznacenie problému, identifikačný znak tímu, programovací jazyk. Tento skript tiež oznámi tímu, či odovzdávanie prebehlo úspešne, alebo nastala nejaká chyba.

V adresári, kam sa jednotlivé programy skopírovali skriptom *submit*, beží skript s menom *guard*. Beží cyklicky, neustále robí listing daného adresára (*ls*) a sleduje zmeny. Ak nastala nejaká zmena, zvukovým aj vizuálnym signálom oznamuje, že niektorý tím odovzdal úlohu. Na terminál vypíše meno premenovaného súboru, ktoré obsahuje predtým vymenované parametre.

Teraz začína úloha sudcu. Najprv spustí skript *prepare*, ktorý nastaví rôzne parametre, premenné, vzorové vstupy, výstupy a programovací jazyk na ďalšie spracovanie. Potom je spustený skript *compile*, ktorý podľa parametrov daných skriptom *prepare*, zdrojový kód skompiluje do špeciálneho adresára. Nasleduje testovanie, ktoré je zabezpečené skriptom *test*. On najprv vykoná hrubý test programu, čo znamená, že ho spustí, a ak program neskončí v definovanom časovom limite, tak ho zruší (*kill*). Ak však program skončil vlastnoprávne (t.j. nebolo treba ho zrušiť), potom tento skript porovnáva výstup programu s daným vzorovým výstupom. Podľa výsledku testovania môže byť potrebný aj zásah sudcu. Sudca teraz spustí skript *finish*, ktorý úplne vymaže testovací adresár (on je vlastne *clean*) a hotový zdrojový program skopíruje do adresára *done*. Tým je naznačené, že daná úloha je otestovaná.

Existuje aj ďalší skript, ktorý podobne skriptu *guard*, beží cyklicky na pozadí. Jeho meno je *makerank* a má za úlohu vytvorenie HTML stránky, ktorá slúži na prezentovanie súčasného stavu. Stránka obsahuje tabuľku s poradovým číslom, číslom tímu, jednotlivými označeniami úloh a celkovým casom. Do tabuľky sú zapísané aj jednotlivé odovzdávania úloh. Ak bolo odovzdané riešenie správne, tak sa do príslušnej bunky zapíše čas. Ak je však riešenie nesprávne, aktualizuje sa iba počet neúspešných odovzdaní v hornom indexe. Ako príklad môže poslužiť Tab. c. 1.

| Poradie | Tím | A | B | C | D | E | F | Celkový cas |
|---------|-----|-------------------|--------------|------|-------------------|--------------|---|-------------|
| 1. | 3. | ¹ | 0:32 | | 1:46 ² | | | 8:26 |
| 2. | 2. | ⁵ | 0:59 | 2:48 | | ² | | 4:38 |
| 3. | 4. | 4:50 ² | | | | | | 5:20 |
| 4. | 1. | | ¹ | | | | | |

Tab. c. 1: Prezentácia výsledkov – príklad výstupu skriptu *makerank*

Aby mali sudcovia ľahšiu úlohu, jednotlivé skripty (ale iba skripty sudcov) sú ocíslované podľa poradia, v ktorom sa majú spúšťať, čiže: *1_prepare*, *2_compile*, *3_test*, *4_finish*. Toto im umožňuje, aby postupovali vždy správne a nevynechali ani jednu z dôležitých fáz vyhodnocovania.

Problematika zistovania časového limitu na beh programu je zaujímavá. Vzorový program sa spúšťa cyklicky, napríklad 10-krát. Z týchto údajov získajú priemerný čas jedného behu programu a tento čas vynásobia tromi, aby podmienky neboli také prísne (vzorové programy sú totiž optimalizované a tiež veľmi efektívne).

Potrebu nového systému naznačuje aj to, že počas prípravného kola nastala situácia, v ktorej testovací systém ohodnotil odovzdané zdrojové programy nesprávne, hoci riešenia boli správne. Príčinu chyby môžeme pravdepodobne hľadať v komplexnosti, zložitosti celého odovzdávacieho systému, ktorý nie je jednotný. Počas konzultácie s organizátormi sme zistili, že aj oni sami by potrebovali spoľahlivejší a inteligentnejší systém, ktorý by im umožňoval rýchlejšiu, spoľahlivejšiu a teda lepšiu prácu.

1.3.2 FLP – systém na odovzdávanie zadaní

V rámci predmetu Funkcionálne a logické programovanie sa od tohoto roku na FEI STU využíva nový systém na spracovanie odovzdávaných zadaní. Autorom systému je M. Trabalka.

Systém je vytvorený v jazyku Perl a je podľa vyjadrenia autora veľmi jednoduchý.

Princíp znovupoužitia existujúcich softvérových súčiastok automaticky evokuje myšlienku, či by nebolo vhodné naviazať na existujúci systém a zamerať sa na implementáciu nových funkcií, pričom by sme túto skutočnosť konzultovali so zadávateľom projektu a s konkurenčným tímom. Avšak je osadený do frameworku, bez ktorého je systém

nepožiteľný, pričom autor projektu nie je z pochopiteľných dôvodov ochotný poskytnúť existujúci framework voľne k dispozícii.

Komunikácia so systémom je riešená pomocou web-rozhrania, pričom existujú dva typy používateľov – študent a cvičiaci.

Práca so systémom

Každý študent má pridelené meno a heslo, ktoré si môže pomocou web-rozhrania zmeniť. Po prihlásení do systému sa mu zobrazí HTML stránka, na ktorej vidí svoje odovzdané programy, čas ich odovzdania, hodnotenie a poznámky k programu od cvičiaceho. K informáciám o ostatných študentoch nemá prístup. Po kliknutí na príslušné tlačítko môže interaktívnym spôsobom zadať cestu na nový odovzdávaný súbor a odoslať ho.

Každý študent má viacero možností na odovzdanie, pričom sa akceptuje posledné odovzdanie v rámci daného termínu. Nevýhodou súčasného systému je, že zatiaľ musí cvičiaci určiť posledné zadanie v termíne podľa zaregistrovaného dátumu a času, táto činnosť nie je žiadnym spôsobom automatizovaná.

Systém podporuje viacerých cvičiacich, pričom každý cvičiaci spravuje vlastnú skupinu študentov.

Systém obsahuje aj niektoré ďalšie zaujímavé funkcie, napr. automatické skomprimovanie archivovaných zadaní do formátu *.tgz a download tohoto súboru.

1.3.3 PC² -The Programming Contest Control System

Zo všetkých analyzovaných systémov má tento systém najdlhšiu históriu (je vyvíjaný od roku 1988 na California State University, Sacramento, ďalej CSUS) Systém je používaný na lokálnych, regionálnych (USA) aj medzinárodných kolách súťaží typu ACM.

Celý systém je napísaný v jazyku Java a používateľovi (administrátorovi, rozhodcovi, súťažiacemu, ...) poskytuje:

- podpora viacerých miest pri organizovaní lokálnych kôl (ďalej budeme rozlišovať iba lokálne a korešpondenčné kolá, k lokálnym budú patriť aj regionálne a medzinárodné kolá)
- poskytuje mechanizmy pre administrátorov súťaže – konfigurácia rôznych parametrov samotnej súťaže, pre súťažiacich – možnosť odoslať riešenia problémov, pre rozhodcov – získať a vyhodnotiť odoslané riešenia a vrátiť výsledky späť súťažiacim, vypočítavanie a zobrazovanie poradia súťaže, a množstvo ďalších vlastností.
- archivuje odoslané riešenia (správne aj nesprávne), udržiava a zobrazuje aktuálne poradie súťažiacich, umožňuje rozhodcom spätne vybrať a „spustiť“ odoslané riešenie,

- podporuje „multi-site“ súťaže, tzn. súťaže konajúce sa na viac ako jednom mieste,
- heterogénne prostredie (možnosť kombinácie prostredí Windows, Unix)

Systém je „šitý na mieru“ lokálnym kolám ACM súťaží, pre korešpondenčné kolá sa však nehodí, keďže by bolo potrebné stiahnuť a nainštalovať klientský program a celá komunikácia by prebiehala cez tento program. PC² má však veľa funkcií, ktoré môžeme implementovať aj v našom systéme.

Domovská stránka: <http://www.ecs.csus.edu/pc2/>

1.3.4 UPC2001 Internet Contest (Ural Programming Contest)

Toto je príklad tzv. „ad hoc“ systémov, ktoré sa vytvárajú len na jedno použitie – na jednu súťaž.

Práca so systémom:

Pred samotnou súťažou si súťažiaci musia stiahnuť klientský program (ClientSoftware.exe) pre MS Windows, ktorý so serverom komunikuje pomocou TCP/IP protokolu (porty 4305, 4306).

Program má veľmi intuitívne ovládanie, avšak jeho jedinou úlohou je odoslania problému či otázky pre rozhodcov a zobrazenia aktuálnych výsledkov neposkytuje nič viac.

Problémom použitia tohto typu prístupu (klientský program) je aj fakt, že TCP porty sú často kvôli bezpečnosti blokové, čo môže znemožniť zapojenie sa do prebiehajúcej súťaže.

Domovská stránka: <http://contest.psu.ru/Inet/>

1.3.5 Ural State University Problem Set Archive with Online Judge System

Ako už názov napovedá, ide o systém ruskej produkcie, ktorý je akosi zbierkou problémov. Jeho súčasťou je však aj prepracovaný online vyhodnocovací systém, ktorý spolu s intuitívnym ovládaním a prehľadným spracovaním celého súboru patrí k najprepracovanejším online vyhodnocovacím systémom zanalyzovaných systémov pre podporu programovacích súťaží voľne prístupných v pavucine. Keďže je systém otvorený pre širokú verejnosť, samozrejmosťou je komunikácia s jeho používateľmi v anglickom a ruskom jazyku (podľa výberu). Systém používateľom poskytuje množstvo funkcií, ktoré sú opísané v nasledujúcej časti.

Štruktúra systému je rozdelená do troch častí:

- Online Judge - časť týkajúca sa samotného riešenia problémov, ktoré sa nachádzajú v archíve – odoslanie riešenia problému, odoslanie otázky do fóra (konferencie), odpovedanie na otázku vo fóre.
- Authors - podsystem, ktorého úlohou je evidencia používateľov – súťažiacich, tzn. umožňuje vytvorenie jednotlivca, jeho voliteľné zaradenie do tímu, zobrazenie informácií o ostatných používateľoch či aktualizáciu vlastných informácií alebo vyhľadanie súťažiaceho podľa jeho mena.
- Online Contests - tento podsystem poskytuje informácie o jednotlivých súťažiach (už ukončených, práve prebiehajúcich i naplánovaných na najbližšie časové obdobie). Takisto tu možno nájsť celkové poradie autorov – súťažiacich, zoradených podľa ich úspešnosti v predchádzajúcich súťažiach.

Pro stručnom načrtnutí systému nasleduje podrobnejší popis niektorých jeho častí:

- **Online Judge**

o *Submit A Problem*

V prípade, že používateľ systému je zaregistrovaný, môže odoslať riešenie vybraného problému na vyhodnotenie. Pre odosielanie riešenia systém poskytuje dve ekvivalentné možnosti. Prvou je poslanie riešenia mailom na presne špecifikovanú adresu, druhý spôsob je komfortnejší – riešenie je možné odoslať pomocou webovského formulára. V oboch prípadoch je však potrebné uviesť identifikačné číslo súťažiaceho, číslo riešeného problému, programovací jazyk, v ktorom bolo riešenie napísané a samotný zdrojový kód riešenia.

Pri odosielaní riešenia si súťažiaci môže vybrať aj spôsob jeho oboznámenia s výsledkom riešenia. Na výber má tri možnosti, a to odpoveď mailom na vlastnú adresu (uvádzanú v jeho osobných údajoch), odpoveď mailom na špecifikovanú adresu (túto musí uviesť do textového poľa vo vstupnom formulári) alebo použitie akejkoľvek monitorovacej tabule nazývanej *Online Status* (popísanej nižšie v tejto časti).

o *Problem Set*

Táto sekcia obsahuje množstvo samostatných problémov, ako aj problémov rôznych súťaží, ktoré sa konali v prechádzajúcich rokoch a súťaživému záujemcovi poskytuje vynikajúcu príležitosť vybrať si primerane náročný problém, riešiť ho, a týmto sa pripraviť na programátorskú súťaž

- *Online Status*

V prípade, že používateľ si ako odpoveď na odoslané riešenie vyberie *Online Status*, takmer okamžite môže výsledok svojho riešenia vidieť na online tabuli, ktorá sa automaticky aktualizuje každých 10, 30, 60, alebo 120 sekúnd (interval obnovovania stavu je nastaviteľný používateľom).

- *WebBoard*

Žiadny systém podobného typu by nemal postrádať konferenciu (fórum), kde si môžu používatelia vzájomne klásť otázky a odpovedať na ne. WebBoard je práve týmto fórom. Správy tu možno

- **Authors**

- *Register*

Ak sa chce používateľ systému zapojiť do riešenia problémov, musí byť v systéme evidovaný. Zaregistrovať sa môže práve v tejto časti. Po vložení základných osobných údajov (meno, e-mail, domovská stránka, štát), hesla, motta a výberu tímu (voliteľne) sa používateľovi prideli jedinečné číslo, ktorým bude identifikovateľný počas celej práce v systéme.

- *Authors Ranklist*

Kvôli zvýšeniu súťaživosti systém poskytuje aj menný zoznam všetkých súťažiacich (registrovaných používateľov) zoradený podľa počtu vyriešených problémov. O každom súťažiacom je možné si vyhľadať bližšie informácie o problémoch, ktoré úspešne vyriešil ako aj informácie o počte a výsledkoch odoslaných riešení.

- *Update Your Info*

V prípade zmeny osobných údajov si každý používateľ môže svoje údaje aktualizovať aj v tomto systéme.

- **Online Contests**

- *Current Contest*

Táto časť podáva informácie o práve bežiacich súťažiach alebo o súťaži najbližšie naplánovanej.

- *Scheduled Contests*

- *Archive*

- *Summary Ranklist*

Z ďalších zaujímavých funkcií treba spomenúť aj možnosť pridávania vlastných problémov (tá však nie je plne automatizovaná, je tu potreba overenia administrátormi systému).

Domovská stránka: <http://acm.timus.ru/>

1.3.6 IPC – RoboJudge

Pri súťaži Internet Programming Contest (IPC), ktorý organizuje Duke Univerzity sa používa automatizovaný systém spracovania odovzdaných zadaní s názvom *RoboJudge*.

Systém je naprogramovaný v jazyku Perl a je veľmi zviazaný z operacným systémom UNIX.

Výhodou takéhoto riešenia je, že operacné systémy typu UNIX obsahujú veľa hotových nástrojov na spracovanie textov, čiže implementácia je veľmi jednoduchá. Nevýhodou je, že systém je platformovo závislý, aj keď mnohé z týchto nástrojov sú k dispozícii aj pre iné OS.

Práca so systémom

Z hľadiska súťažiacich systém pozostáva z troch nástrojov:

- register – zaregistrovanie sa do súťaže
- submit – odoslanie vyriešenej úlohy rozhodcom
- clarify – odoslanie otázky na rozhodcov

Tieto nástroje sú veľmi jednoduché a slúžia len na vytvorenie e-mailu s pevným formátom a odoslanie rozhodcom.

Proces na strane rozhodcov je plne automatizovaný. Záznam činností a tvorba prehľadov o stave sú taktiež plne automatizované. Rozhodca zasahuje len v prípade, že nie je odovzdanie riešenia nie je v poriadku.

Po príchode e-mailu na stranu serveru sa automaticky vyvolá program, ktorý analyzuje hlavičku a telo e-mailu a podľa toho posielajú mail na spracovanie ďalším programom, ktoré odovzdaný program preložia, otestujú na predpripravovaných dátach a výsledok odošlú naspäť súťažiacemu.

Odobzdania môžu byť však aj ďalej distribuované medzi rozhodcov.

Tento systém sa v praxi osvedčil, pričom v priebehu 3 hodinovej súťaže dokáže 5 rozhodcov spracovať rádovo stovky odovzdaní. Typická dĺžka spätnej odozvy súťažiacemu je do 2minút, čo je pomerne krátky čas.

Domovská stránka: www.cs.duke.edu/~ola/ipc.html.

1.4 Špecifikácia požiadaviek

Pri vytváraní špecifikácie požiadaviek na systém, sme vychádzali z vlastných skúseností so súťažou ACM, ako aj so skúsenostím, ktoré sme získali v minulých rokoch s odovzdávaním zadaní na rôznych cvičeniach.

Požiadavky na systém sme konzultovali s Mgr. G. Polcicovou, ktorá je hlavnou organizátorkou súťaží ACM na FEI STU a s R. Kováčom, ktorý organizoval korešpondenčné kolo súťaže ACM. Ďalej sme požiadavky konzultovali so študentmi, ktorí pripravovali lokálne kolo súťaže ACM v posledných dvoch rokoch – konkrétne M. Horváth, P. Lacko a P. Trebatický.

Z hľadiska využitia systému na cvičeniach bola prínosom konzultácia požiadaviek na systém s Ing. B. Steinmüllerom (cvičenia s predmetu Operacné systémy) a s Ing. M. Trabalkom (predmet Funkcionálne a logické programovanie).

1.4.1 Popis používateľov

V systéme sme identifikovali nasledovné typy používateľov:

- administrátor systému
- administrátor súťaže
- rozhodca
- súťažiaci
- pozorovateľ

Administrátor systému - jeho úlohou je prvotná inštalácia a konfigurácia parametrov systému, ktoré priamo nesúvisia s konfiguráciou jednotlivých súťaží. Jeho druhou úlohou je správa účtov administrátorov súťaží.

Administrátor súťaže – nastavuje parametre súťaže a užívateľské účty rozhodcov, súťažiacich, príp. pozorovateľov. Každá súťaž má jedného administrátora.

Rozhodca - pri úlohách, ktoré sa nedajú vyhodnotiť automaticky, ohodnocuje správnosť a kvalitu odovzdaných zadaní. Má možnosť odpovedať na všeobecné alebo problémovo-orientované otázky súťažiacich. Súťaž môže mať viacerých rozhodcov.

Súťažiaci - má možnosť prevziať si zo servera jednotlivé zadania, odovzdávať riešenia a prehliadať čiastkové alebo konečné výsledky.

Pozorovateľ - pri práve prebiehajúcich súťažiach si môže prezerať čiastkové výsledky a priebežné poradie súťažiacich tímov, avšak nemôže sa aktívne zúčastňovať súťaže.

1.4.2 Špecifikácia funkcií systému

Systém na podporu programátorskej súťaže by mal poskytovať nasledovné funkcie:

- konfigurácia užívateľských účtov
- zabezpečený prístup užívateľov do systému
- podpora hierarchických súťaží – prenos údajov medzi jednotlivými kolami
- nastavenie parametrov súťaže
 - časové obmedzenie na registráciu účastníkov a odovzdanie zadaní
 - povolené programovacie jazyky
 - maximálny počet odovzdaní
- registrácia účastníkov súťaže, a to:
 - administrátorom súťaže (v prípade obmedzenia účastníkov na konkrétne osoby)
 - samotnými účastníkmi (otvorená súťaž)
- distribúcia zadaní medzi súťažiacich
- odovzdávanie vyriešených zadaní
- generovanie prehľadov o priebehu súťaže
- komunikácia rozhodca – súťažiaci, pričom je vhodné, aby bola táto komunikácia transparentne oddelená od ostatnej komunikácie
- možnosť diskvalifikácie súťažiaceho tímu, resp. jednotlivca
- distribúcia riešení medzi rozhodcov v prípade rucného vyhodnocovania
- kontrola povolených programovacích konštrukcií pre jednotlivé prog. jazyky
- v prípade automatického vyhodnocovania preklad zadaní a testovanie korektnosti ich výstupov pre množinu vstupných testovacích dát
- monitorovanie a vhodná prezentácia všetkých vykonaných akcií v systéme

Na tieto funkcie sú kladené nasledovné požiadavky:

- nezávislosť komunikácie od platformy klienta

- co najjednoduchšie a najrýchlejšie používateľské rozhranie na odovzdávanie zadaní, pričom je vhodné ponechať súťažiacim možnosť komunikácie so systémom s príkazového riadku
- rýchla spätná odozva o správnosti riešenia
- možnosť voľby jazyka používateľského rozhrania
- možnosť voľby automatického, alebo ručného vyhodnocovania
- možnosť použitia viacerých programovacích jazykov, ľahké rozšírenie o ďalšie jazyky

Kontrolu toho, či všetci užívatelia používajú povolené prostriedky ponechávame na administrátora OS na ktorom bude systém prevádzkovaný, nakoľko je táto časť silne platformovo závislá. Keďže predpokladáme heterogénne prostredie s rôznymi OS, je táto časť vhodná ponechať mimo nášho systému.

Taktiež kvalita nástrojov na monitorovanie bezpečnosti, ktoré sú k dispozícii, je výraznejšie vyššia, než akú sme schopní realizovať v rámci nášho tímového projektu.

Z hľadiska bezpečnosti je teda potrebné zamerať sa najmä na bezpečnosť komunikácie medzi softvérom na strane používateľov a naším systémom. Ďalej je potrebné zamerať sa na minimalizáciu možnosti prieniku do systému vyhodnocovania a archivácie odovzdaných riešení.

1.4.3 Priorita funkcií

Funkcie, ktoré by mal systém poskytovať sme rozdelili do troch skupín:

- I. Funkcie, ktoré poskytujú systému základnú funkčnosť, resp. bez ktorých by nami implementovaný systém nedosahoval ani kvalít už dostupných systémov.
- II. Funkcie, ktoré nie sú nevyhnutné na prevádzkovanie systému, ale dôležitým spôsobom ovplyvňujú jeho použiteľnosť.
- III. Drobné vylepšenia systému.

Nasleduje zatriedenie funkcií systému do jednotlivých skupín:

- I. VYSOKÁ
 - konfigurácia užívateľských účtov
 - nastavenie parametrov súťaže
 - registrácia účastníkov súťaže
 - odovzdávanie vyriešených zadaní

- generovanie prehľadov o priebehu súťaže
- kontrola povolených programovacích konštrukcií pre jednotlivé prog. jazyky
- v prípade automatického vyhodnocovania preklad zadaní a testovanie korektnosti ich výstupov pre množinu vstupných testovacích dát
- monitorovanie a vhodná prezentácia všetkých vykonaných akcií v systéme
- zabezpečený prístup užívateľov do systému

II. STREDNÁ

- možnosť alternatívnej komunikácie súťažiacich so systémom s príkazového riadku
- distribúcia riešení medzi rozhodcov v prípade ručného vyhodnocovania
- komunikácia rozhodca – súťažiaci

III. NÍZKA

- viacjazyčné používateľské rozhranie

2 Hrubý návrh

V tejto kapitole sa nachádza hrubý návrh systému, obsahuje návrh architektúry systému, model údajov a jeho popis. Obsahuje aj popis možných alternatív riešenia.

2.1 Architektúra systému

Vzhľadom na zozitost akompexnost problému, ktorý má systém realizovať je základným problémom správny výber architektúry celého systému. Architektúra musí zohľadňovať všetky požiadavky kladené na systém a účinne ich sklbiť s technológiou, na ktorej mienime systém vybudovať (viac čast Výber technológie). Našou predlohou pri hľadaní vhodného riešenia boli hlavne systémy používané vo svete. Hlavnými faktormi ovplyvňujúcimi architektúru systému sú:

1. technológia
2. komunikácia klienta so serverom – forma odovzdávania zadaní
3. modulárnosť a škálovateľnosť

2.1.1 Architektúra systému z pohľadu použitej technológie

Architektúra systému z pohľadu použitej technológie sa skladá z piatich hlavných častí:

1. prezentčná/užívateľská časť
2. komunikačná časť pre platformovo závislých klientov
3. vyhodnocovacia časť
4. údajová časť
5. samotní platformovo závislí klienti

Niektoré časti majú jednoznačné umiestnenie a reprezentáciu v systéme:

1. prezentčná/užívateľská časť – web server
2. údajová časť – databázový server a štruktúra súborov

Ostatné časti sa dajú realizovať rôznymi spôsobmi. V nasledovných oddieloch popíšeme jednotlivé možnosti realizácie systému vzhľadom na vybrané technológie aj s diagramom štruktúry systému.

2.1.1.1 Vyhodnocovacia a komunikacná časť ako súčasť web servera

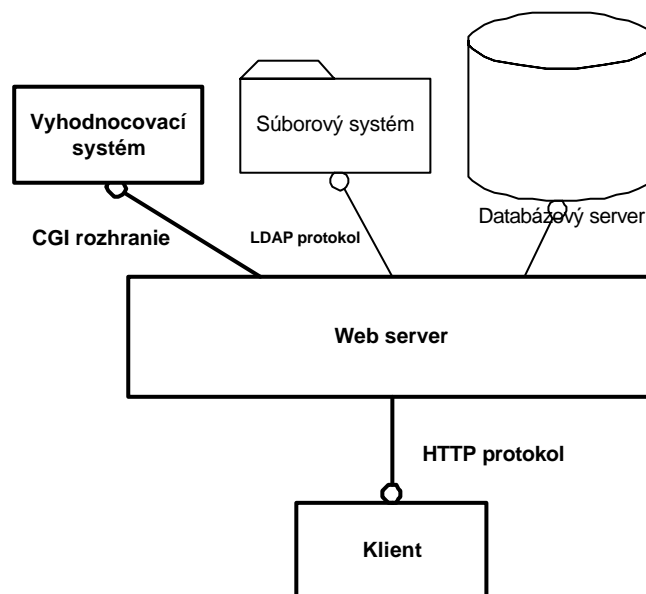
Toto riešenie sa opiera o dve hlavné časti systému: databázový server a web-server. Ostatné časti systému sú realizované v rámci web servera, pomocou CGI rozhrania web-servera. Vyhodnocovacia časť systému beží ako CGI script web-servera a tak isto komunikacná časť pre platformovo závislých klientov je realizovaná v rámci CGI rozhrania web servera cez protokol http.

Výhody realizácie:

- zapúzdrenosť systému v rámci web servera
- prenositeľnosť systému
- bezpečnosť systému
- modulárnosť systému

Nevýhody realizácie:

- nutnosť implementácie CGI rozhrania
- úzka väzba vyhodnocovacej časti na nesúvisiacom CGI rozhraní
- nutnosť implementácie CGI rozhrania do platformovo závislých klientov a simulácie komunikácie pomocou protokolu http s web serverom



Obr. c. 4: Vyhodnocovacia a komunikacná časť ako súčasť web-servera

2.1.1.2 Systém so samostatnou vyhodnocovacou castou

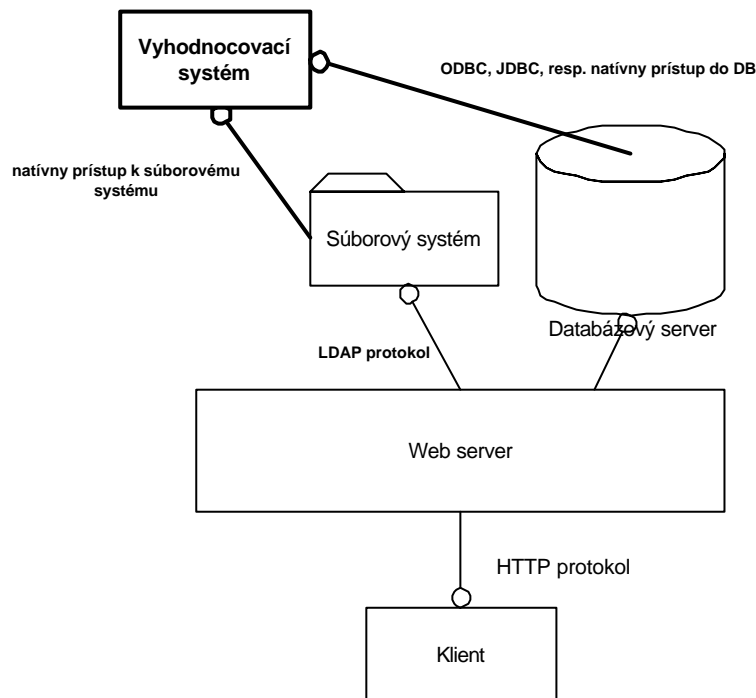
Pri tomto spôsobe riešenia je vyhodnocovacia časť systému samostatná a je tvorená rôznymi programami na strane servera, ktoré komunikujú s web-serverom cez údaje v databáze. Komunikácia je naďalej tvorená klientmi simulujúcimi http protokol.

Výhody realizácie:

- oddelenie vyhodnocovacej časti a tým získanie možnosti lepšej škálovateľnosti
- ostatné vlastnosti predchádzajúceho riešenia

Nevýhody:

- nutnosť implementácie CGI rozhrania do platformovo závislých klientov a simulácie komunikácie pomocou protokolu http s web serverom



Obr. c. 5: Systém so samostatnou vyhodnocovacou časťou

2.1.1.3 Samostatná vyhodnocovacia časť, podpora klientov s iným rozhraním ako http

V tomto riešení problému je vyhodnocovacia časť systému samostatná tvorená rôznymi programami na serveri, ktoré komunikujú s webserverom cez údaje v databáze. Komunikácia je naďalej tvorená klientmi simulujúcimi http protokol.

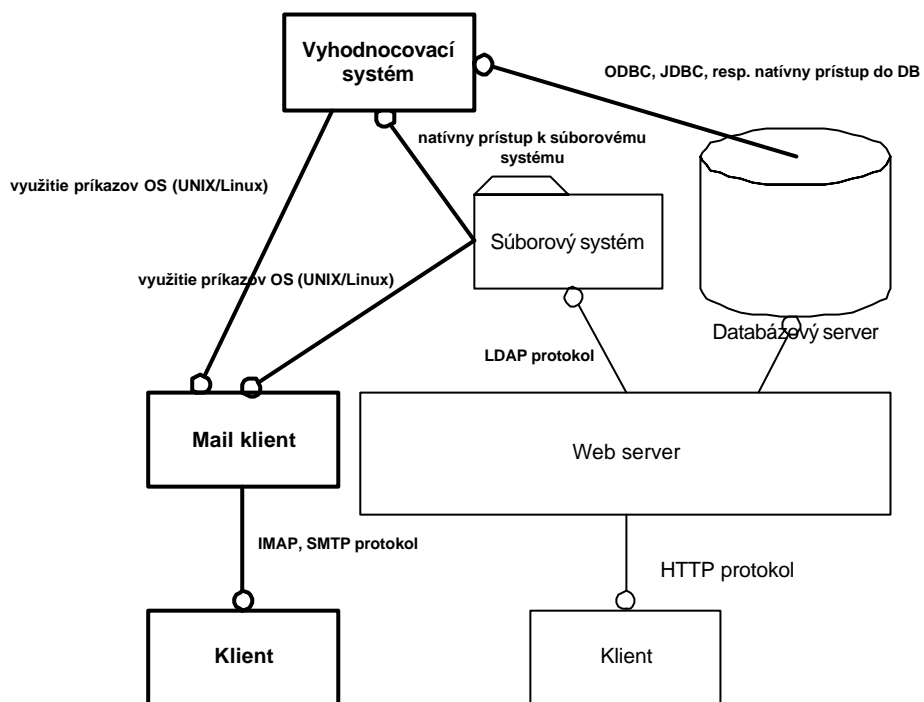
Výhody realizácie:

- oddelenie vyhodnocovacej časti a tým získanie možnosti lepšej škálovateľnosti

- ostatné vlastnosti predchádzajúceho riešenia

Nevýhody:

- nutnosť implementácie CGI rozhrania do platformovo závislých klientov a simulácie komunikácie pomocou protokolu http s web serverom



Obr. c. 6:

Systém so samostatnou vyhodnocovacou časťou a s podporou klientov s iným rozhraním ako http

2.1.2 Architektúra systému z pohľadu komunikácie klienta so serverom

Vzhľadom na širšie použitie systému nielen na ACM súťaže, ale ako aj vyhodnocovací systém na cvičenia, musí byť umožnená podpora čo najväčšieho počtu špecifických klientov, resp. komunikácie na určitých protokoloch. Týmto požiadavkám vyhovuje najviac systém číslo 3 z predchádzajúcej časti.

Táto architektúra systému nám umožňuje tvorbu väčšieho množstva klientov ako napr.: formulár v internetovom prehliadaci, skript posielajúci zadanie mailom, samotné poslanie zadania mailom (vhodné pre korešpondenčné kolá).

Z tohto rozdelenia nám vyplývajú základné moduly systému orientované smerom k užívateľovi, a sú to:

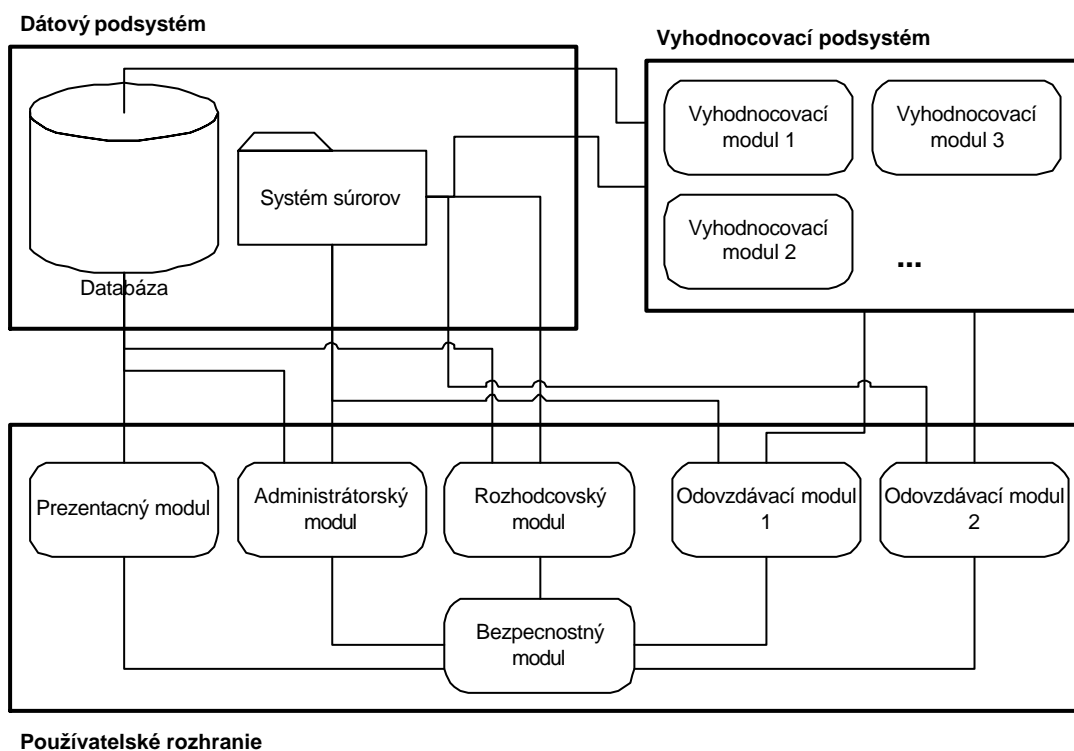
1. prezentčný modul – web prehliadac
2. odovzdávacie moduly – web prehliadac, mailový klient, ...
3. administrátorský modul – web prehliadac
4. rozhodcovský modul

2.1.3 Architektúra systému z pohľadu modulárnosti a škálovateľnosti

2.1.3.1 Vzhľadom na logickú štruktúru systému

Ide i rozdelenie do troch hlavných posystémov, ktoré sú navzájom poprepájané podľa potreby zdieľania dát navzájom, resp. spúšťania jednotlivých akcií (spúšťanie vyhodnocovania po odovzdaní). Na diagrame ja naznacena modulárnosť číslovaním. Ide hlavne o modulárnosť vyhodnocovacieho podsystému a odovzdávacích modulov, kde množstvo odovzdávacích modulov je závislé na množstve povolených foriem odovzávania a množstvo vyhodnocovacích modulov na počte rôznych súťaží, ktoré budú na serveri uskutočňované. V stručnosti k vysvetleniu jednotlivých vzťahov. K systému súborov majú prístup nasledujúce moduly uvádzané aj s odôvodnením:

1. Administrátorský modul – tvorba štruktúry adresárov pre jednotlivé súťaže, príprava zadaní a testovacích vstupov.
2. Rozhodcovský modul – prístup k uloženým výstupom po vyhodnotení a odovzdaným zdrojovým kódom
3. Odovzdávacie moduly – pre uloženie prijatého zdrojového kódu riešenia na disk
4. Vyhodnocovacie moduly – pre samotné vyhodnotenie zdrojových kódov a zápisu ich výstupov

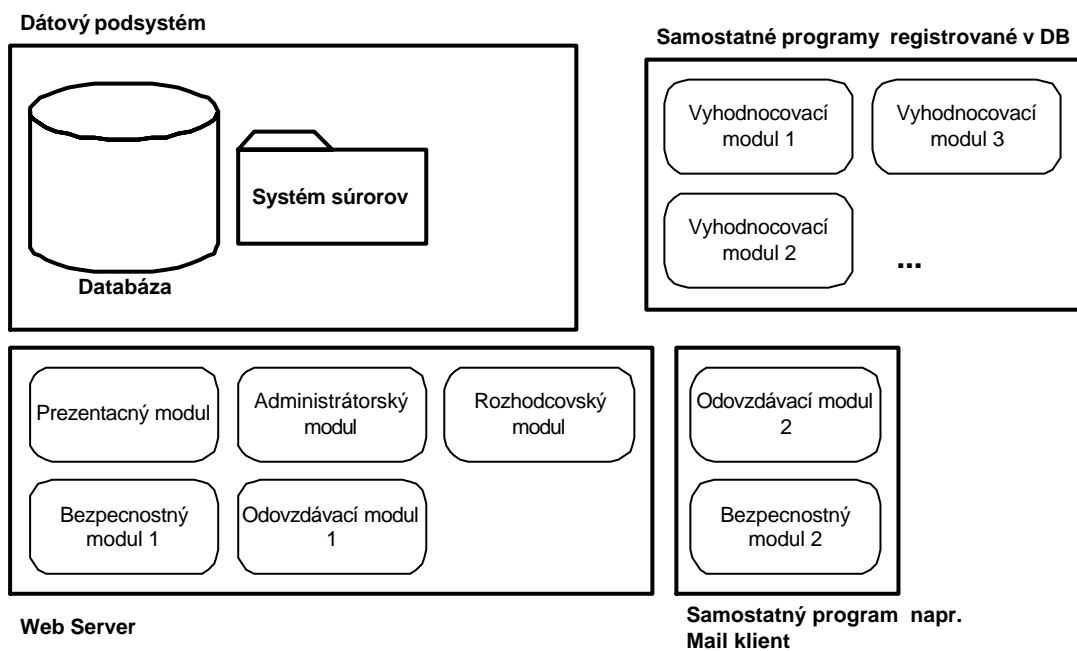


Obr. c. 7: Architektúra systému vzhľadom na jeho logickú štruktúru

Prístup k databáze je umožnený nasledujúcim modulom s nasledujúcim odôvodnením:

1. Bezpečnostný modul – pre overenie používateľa pri prihlasovaní sa do systému
2. Prezentacný modul – na prezentáciu výsledkov uložených v DB
3. Administrátorský modul – na registráciu súťažiacich, pridelovanie zadaní, registráciu zadaní aj s vyhodnocovacími modulmi
4. Rozhodcovský modul – na prístup k výsledkom
5. Odovzdávacie moduly - na registráciu odovzdaných riešení
6. Vyhodnocovacie moduly – na registráciu vyhodnotených výstupov daných riešení ako aj zápis výsledkov

2.1.3.2 Vzhľadom na vzájomú väzbu logickej štruktúry a použitých technológií



Obr. c. 8: Arch. systému vzhľadom na vzájomnú väzbu log. štruktúry a použitých technológií

Na Obr. c. 8 je znázornená fyzická realizácia dekompozície do modulov rámci použitých technológií. Je tu znázornené umiestnenie jednotlivých modulov v nami použitých prostriedkoch.

2.2 Výber technológie

Vzhľadom na vlastnosti požadované od systému sa núka množstvo riešení, ako po stránke architektúry systému, tak aj z pohľadu použitej technológie dostupnej na realizáciu projektu.

2.2.1 Zhrnutie požiadaviek

Pre výberom technológie zhrnme znova požiadavky, ktoré sú kladené na systém, tak ako boli uvedené pri špecifikácii.

1. odovzdávanie výsledkov po sieti, resp. internete
2. bezpečnosť prístupu a vyclenenia jednotlivých kategórií užívateľov
3. komunikácia so systémom nezávislá na platforme
4. evidencia výsledkov, vstupov, výstupov pre sudcov
5. automatizácia vyhodnocovania
6. pohodlná správa systému
7. rýchla spätná odozva systému
8. modulárnosť a škálovateľnosť systému
9. príjemné používateľské rozhranie
10. čo najjednoduchšie a najrýchlejšie odovzdávanie vyriešených problémov

2.2.2 Možnosti realizácie

Dôležitým obmedzním, ktoré musíme zohľadniť vo výbere technológie je aj jej dostupnosť v prostredí v našej fakulte, resp. dostupnosť danej technológie v softvérovom štúdiu, kde budeme systém implementovať.

K vyššie uvedeným bodom sme sa snažili nájsť riešenie, ktoré by pokrývalo požiadavky kladené na systém. Ide vlastne o vyhodnotenie možností realizácie.

1. Systém musí mať architektúru klient – server, pracovať na protokole TCP/IP na jednom, najrozšírenejších operačných systémov, t.j. niektorý z operačných systémov firmy Microsoft, alebo niektorý UNIX OS.
2. Príjemné užívateľské rozhranie možno zabezpečiť napríklad použitím internetového prehliadača ako aj množinou platformovo závislých klientov, kľúčovou myšlienkou je hlavne použitie web servera, ktorý môže plniť aj prezentacnú časť systému.
3. Pri použití web-servera na realizáciu prístupu k systému sa javí opodstatnené realizovať bezpečnostné politiky systému cez rozhranie web servera (väčšinou už implementované v rámci samotného servera na dostatočnej úrovni), ako aj na úrovni prístupu k dátam v databáze
4. Problém rýchlej odozvy systému je najviac rizikovou časťou, ide o kombináciu rôznych faktorov, ktoré majú vplyv na celkovú rýchlosť odozvy, návrh realizácie

vyhodnocovacej časti systému, ako aj rýchlosť sieťového spojenia, rýchlosť spracovania požiadaviek, atď.

5. Evidenciu výsledkov možno najvhodnejšie realizovať výberom databázového systému a kvalitným návrhom dátovej časti systému ako aj štruktúry a správy súborov s odovzdanými zadaniami
6. Automatizácia vyhodnocovania by mala byť realizovaná sadou modulov slúžiacich na vyhodnocovanie jednotlivých typov súťaží, úzko komunikujúcich s databázou ako aj komunikacnou časťou systému realizovanou samotným web serverom, alebo iným serverom (napr. e mailovým, viac vid. časť analýza architektúry systému)
7. použitím web rozhrania sa núka implementácia administrátorskej časti formou správy systému v internetovom prehliadači, je to vynikajúce riešenie aj z pohľadu vzdialenej správy systému
8. modulárnosť a škálovateľnosť systému závisí hlavne od rozdelenia jednotlivých modulov systému (vid. návrh systému)

2.2.3 Možné technológie

Nasleduje zoznam možných technologických riešení vhodných pre nami realizovaný projekt (prípadajúcich do úvahy):

1. Operacný systém – Microsoft Windows 2000/NT alebo Unix/Linux. Operacné systémy MS Windows 9x sme dopredu vylúčili kvôli ich nízkej bezpečnosti a stabilite.
2. Web server – Apache. Iná bezpečná a nám dostupná alternatíva podľa nášho názoru žiaľ neexistuje
3. Databázový server – MS SQL Server, PostgreSQL, MySQL.
4. Implementačné prostredie prezentačnej časti, webového rozhrania – PHP, JSP + Java Servlety
5. Implementačné prostredie vyhodnocovacej časti – C, C++, Perl, atď. (ľubovoľne spojitelné s web serverom)
6. Implementačné prostredie platformovo závislých klientov – C, C++, C Shell, atď (podľa potreby a typu súťaže resp. predmetu)

2.2.4 Výber technológie pre implementáciu systému

Náš výber sa riadil hlavne dostupnosťou jednotlivých technológií ako aj dostatočnou vyspelosťou, jednoduchosťou použitia a vhodnosťou pokrytia šírky problému jednotlivých technológií na zámer projektu.

Na základe spomenutých technológií sme sa rozhodli pre nasledovné riešenie systému:

Operacný systém - Unix/Linux

Pre operacný systém Unix/Linux hovoria naše dobré skúsenosti s touto platformou, jeho stabilita, výkon a bezpečnosť v sieťovom prostredí. V softvérovom štúdiu, kde budeme pracovať, je dostupný systém RedHat Linux 7.1, členovia tímu nebudú mať problémy pracovať na rovnakom, alebo podobnom systéme aj mimo softvérového štúdia, nakoľko sa jedná o voľne šíriteľný softvér.

V operacných systémoch typu Unix sa už tradície nachádza veľké množstvo nástrojov na spracovanie textov, ktoré nám môžu uľahčiť vývoj nášho systému. Tiež rôzne distribúcie OS Linux štandardne obsahujú veľké množstvo voľne šíriteľných prekladacov a interpreterov rôznych programovacích jazykov, napr. C, C++, Java, Pascal, Lisp, atd., potrebných pri vyhodnocovaní odovzdaných programov.

Prezentácia – web server Apache a PHP ako ISAPI modul servera Apache

Prezentácia systému (registrácia užívateľov, odosielanie zadání, výsledkové listiny) bude riešená pomocou web rozhrania, aby sa zabezpečila platformová nezávislosť, na ktorú kládli zákazníci veľký dôraz.

Vybrali sme web server Apache – je dostupný v softvérovom štúdiu, máme s ním pozitívne pracovné skúsenosti a vyhovuje našim potrebám aj po bezpečnostnej stránke. Taktiež z jazykom PHP už majú viacerí členovia nášho tímu pozitívne skúsenosti.

Databázový server PostgreSQL

Vzhľadom na výber ostatných častí systému a dostupnosť posudzovaných serverov v softvérovom štúdiu, rozhodovali sme sa medzi MySQL a PostgreSQL. Nakoľko PostgreSQL sa javí oproti MySQL ako vyspelejšia platforma, alternatívu MySQL sme zamietli.

Implementačné prostredie vyhodnocovacej časti – C, C++, Perl

Časť systému, ktorá bude mať za úlohu vyhodnocovanie a testovanie odovzdaných problémov, bude implementovaná ako sada samostatných programov, ktorá bude oddelená od prezentácie časti.

Vzhľadom na povahu systému - prevažne spracovanie textov, práca s ďalšími externými nástrojmi, sa javí výhodné použitie jazyka Perl, pričom rizikom je nízka znalosť tohoto jazyka v prostredí nášho tímu.

Predbežne sme nezavrhlí ani alternatívu C/C++.

Implementačné prostredie platformovo závislých klientov – C/C++

Keďže chceme súťažiacim poskytnúť aj alternatívny - rýchlejší prístup k službám systému, rozhodli sme sa aj pre implemetáciu jednoduchých klientov, ktorí budú spustiteľní s príkazového riadka, a budú komunikovať so systémom pomocou http protokolu.

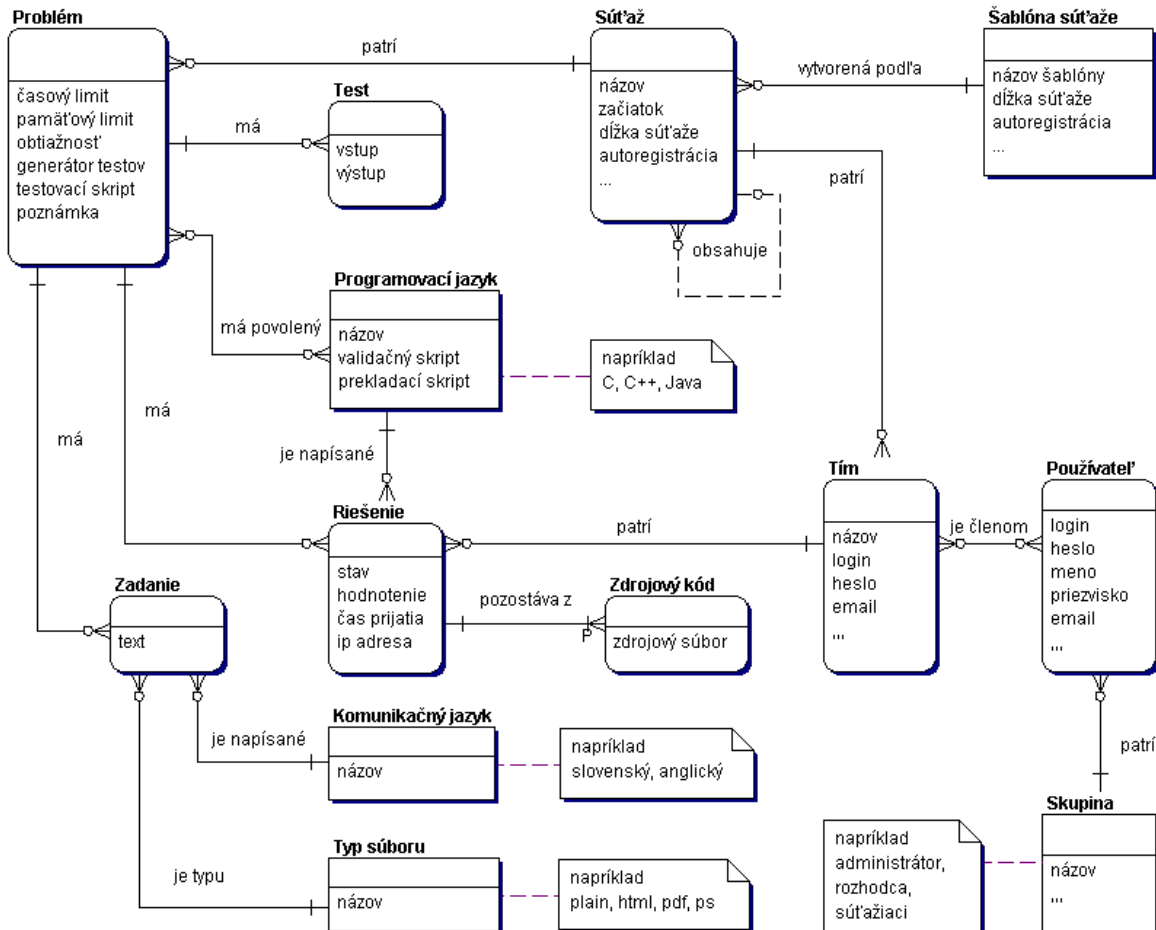
Títo klienti budú implementovaní v jazyku C/C++.

2.3 Logický model údajov

2.3.1 Diagram modelu údajov

Na Obr. c. 9 uvádzame kompetný diagram logického modelu údajov.

Dalších častiach dokumentu bližšie popíšeme jeho štruktúru, jednotlivé entity a vzťahy medzi nimi.



Obr. c. 9: Diagram logického modelu údajov

2.3.2 Popis entít

Používateľ

Obsahuje základné informácie o používateľovi systému a to:

- *Meno* používateľa
- *Prizvisko* používateľa
- *Login*, pod ktorým sa prihlasuje do systému

- *Heslo*, pod ktorým sa prihlasuje do systému
- *Kontaktná informácia* –napr. jeho e-mailova schránka, doplnit do DMD
- *Najvyšší stupeň dosiahnutého vzdelania*
- *Rocník*
- *Úcast na súťaži v minulosti*
- *Prídavná informácia* – lubovolná doplňujúca inf. o používateľovi

Atribúty *Meno*, *Login* a *Heslo* sú povinné, ostatné atribúty sa týkajú iba súťažiacich.

Skupina

Vyjadruje práva užívateľa v danom systéme.

Má atribúty:

- *Názov*
- *Práva*

Pričom atribút *Názov* popisuje množinu funkcií, ku ktorým má *Používateľ* prístup. Rôzne práva popisujú rôzne používateľské skupiny, napr. administrátor systému, administrátor súťaže, rozhodca, súťažiaci, pozorovateľ.

Tím

Reprezentuje abstraktného používateľa, ktorý môže odovzdávať riešenia úloh. Združuje viacerých súťažiacich do spoločného tímu.

Má atribúty:

- *Názov*
- *Login*
- *Heslo*
- *Stav*

Každý tím má svoj *názov*, pričom členovia tímu odovzdávajú zadania po samostatnej autentifikácii, pričom tieto autentifikacie informácie sú spoločné pre celý tím.

Členom tímu môže byť len *používateľ*, ktorý patrí do *skupiny* „súťažiaci“.

Tím ako abstrakcia používateľa, ktorý môže odovzdávať zadania, môže obsahovať aj jediného súťažiaceho. Keďže je neefektívne a neprirozené, aby mal pri súťaži jednotlivcov každý súťažiaci dve autentifikacie informácie (evidované v entitách *tím* a *používateľ*), v tomto prípade nie sú autentifikacie informácie *Tímu* vyplnené a akceptujú sa autentifikacie informácie entity *Používateľ*, ktorá je priradená k danému *Tímu*.

Alternatívou riešenia tohto problému by bolo, aby zadanie mohli odovzdávať samostatne *tím* (ak by sa jednalo o súťaž skutočných tímov, alebo *používateľ*, ktorý patrí do *skupiny* súťažiacich (v prípade, že by sa jednalo o súťaž jednotlivcov).

Atribút *Stav* nadobúda hodnoty v závislosti od toho, či je sa *tím* zúčastňuje príslušného kola súťaže, už sa daného nezúčastňuje – neprešiel „pavúkom“ súťaže, resp. či nie je diskvalifikovaný.

Sút až

Má atribúty:

- *Názov*
- *Zaciatok registrácie*
- *Koniec registrácie*
- *Je povolená autoregistrácia súťažiacich*
- *Zaciatok súťaže*
- *Dĺžka súťaže*

Autoregistráciou sa myslí, že používatelia sa môžu prihlásiť do súťaže sami, pričom v takomto prípade majú určený čas, od kedy sa môžu zaregistrovať.

Ak nie je autoregistrácia povolená, súťažiacich do súťaže pridáva administrátor danej súťaže.

Význam ostatných atribútov je zrejmý z ich pomenovania.

Šablóna súťaže

Obsahuje prednastavenia parametrov pre určitý typ súťaže, nakoľko je pravdepodobné, že sa súťaž z rovnakými pravidlami bude opakovať viackrát.

Atribúty, ktoré obsahuje, sú popísané v rámci iných entít, pre zaujímavosť, dá sa napr. predstaviť dĺžka súťaže, štandardné povolené prog. jazyky, pre jednotlivé problémy atď.

Problém

Popisuje informácie o súťažnom probléme, a to:

- *Obtiažnosť* – pre potreby cvicení môže reprezentovať napr. bodový zisk
- *Casový limit výsledného programu*
- *Pamäťový limit výsledného programu*
- *Odkaz na generátor testov*
- *Odkaz na testovací skript*
- *Povolené jazyky*

V tejto entite existujú viaceré alternatívy. Napr. atribút *Povolené jazyky* by sme mohli priradiť priamo súťaži. V praxi sa však môže vyskytnúť situácia, keď sú na konkrétne úlohy ohľadne programovacích jazykov kladené rôzne obmedzenia.

Testovací skript vykonáva testovanie už preloženého programu na množine testovacích dát, pričom tieto dáta môžu byť generované dynamicky *generátorom testov*.

Zadanie

Má atribút:

- *Text*

Text je reprezentovaný ako odkaz na súbor (cesta k súboru), ktorý obsahuje samotný text zadania. Text zadania sa v cieľovom súbore nachádza v jazyku špecifikovanom prepojením na entitu Komunikacný jazyk, typ tohto súboru je daný prepojením s entitou Typ súboru.

Typ súboru

Atribút:

- *Názov* – pomenovanie typu súboru (napr. plain, html, pdf, ps)

Komunikacný jazyk

Atribút:

- *Názov*

Názov reprezentuje pomenovanie jazyka, v ktorom sa špecifikovaný text nachádza. Výhodou použitia komunikacného jazyka ako samostatnej entity je lepšia kontrola prekladov textov, ako aj väčšia konzistentnosť údajov nachádzajúcich sa v databáze.

Programovací jazyk

Popisuje informácie, ktoré sú potrebné na preklad riešenia.

Má atribúty:

- *Názov* – také pomenovanie programovacieho jazyka, aby čo najlepšie vystihovalo nielen samotný jazyk, ale aj prípadné špecifické nastavenia
- *Validačný skript* – odkaz na skript, ktorého úlohou je kontrola, či sa v prijatom riešení nevyskytujú zakázané funkcie alebo neboli použité iné nedovoľené prostriedky (tie, ktoré možno odhaliť priamo v zdrojovom kóde),
- *Prekladací skript* – odkaz na skript prakiladajúci prijaté riešenie do spustiteľného tvaru.

V praxi sa môže vyskytnúť prípad, že pre rovnaký programovací jazyk sú potrebné rôzne kontrolné skripty pre rôzne súťaže. Aby zostal model dát čo najjednoduchší, dá sa tento problém jednoducho obísť – vytvorenie názov typu C++_ACM, C++_CVICENIE_OS atd.

Riešenie

Riešenie reprezentuje odovzdanú súťažnú úlohu.

Má atribúty:

- *Stav* – akceptované, naskompilovateľné, nesprávny výstup. Ak nie je táto položka vyplnená, znamená to, že riešenie na vyhodnotenie ešte len čaká.
- *Hodnotenie* – bodové ohodnotenie riešenia
- *Cas prijatia* – presný dátum a čas, kedy bolo zadanie prijaté systémom
- *IP adresa* – IP adresa, odkiaľ bolo zadanie odoslané

Uchovávať IP adresu odosielateľa riešenia môže byť užitočné z hľadiska bezpečnosti, avšak jej použitie z hľadiska overenia pravosti užívateľa môže byť problematické, čiže je to skôr orientačná položka

Zdrojový kód

Má atribút:

- *Zdrojový súbor* – odkaz na súbor (cesta k súboru), ktorý obsahuje zdrojový kód riešenia problému, alebo jeho časti (v prípade viacerých súborov).

2.3.3 Opis vzťahov

Teraz bližšie popíšeme vzťahy medzi entitami v diagrame logického modelu údajov.

Každý *používateľ* patrí do práve jednej *skupiny* používateľov, ktorá popisuje jeho práva v systéme, pričom používateľov môže byť v skupine viac.

Ak *používateľ* patrí do *skupiny* súťažiacich, tak patrí do určitého *tímu*.

Pri vzťahu *tím* – *používateľ* existujú dve základné alternatívy. Každý tím má aspoň jedného súťažiaceho, pričom *tím* vzniká zaregistrovaním prvého súťažiaceho pre daný tím, alebo umožnenie existencie tímu bez súťažiacich, pričom by sa súťažiaci do tímu postupne zaregistrovali.

Tím sa zúčastňuje na určitej *súťaži*, príp. na viacerých kolách. Každý *tím* musí patriť k určitej súťaži, pričom je pochopiteľne možná aj existencia *súťaže*, ku ktorej ešte nepatrí žiaden *tím*.

Na Obr. c. 9 je pri entite *súťaž* znázornený rekurzívny vzťah. Súťaž môže mať len jedno kolo, alebo pozostávať z ďalších *súťaží* – ďalších kôl.

Takto je možné podporovať hierarchické *súťaže*, kde každé kolo *súťaže* má vlastný začiatok a koniec, vlastné *súťažné problémy*, pričom množina súťažiacich ostáva uzavretá a hodnotenie jednotlivých tímov sa dá medzi *súťažami* prenášať.

Z hľadiska cvicení môže napr. celý semester reprezentovať jednu *súťaž*, pričom zadania, ktoré postupne študenti v priebehu semestra riešia, reprezentujú jednotlivé kolá tejto súťaže.

Keďže je neefektívne, aby administrátor *súťaže*, pred začiatkom *súťaže* vyplnil všetky parametre týkajúce sa danej *súťaže*, zaviedli sme entitu *šablóna súťaže*, ktorá obsahuje prednastavenia parametrov pre určité typy súťaží.

Pre každú súťaž sú priradené konkrétne *problémy*, pričom k jednému problému je priradené práve jedno, alebo viac zadaní problému (v prípade viacjazyčného prostredia).

Odovzdanie problému reprezentuje entita *riešenie*, pričom riešenie sa vzťahuje k práve k jednému *problému* a k práve jednému tímu. *Riešenie*, ktoré sa viaže k tímu a k *problému* môže byť viac, a to v závislosti od pravidiel súťaže.

Riešenie je ďalej prepojené s entitou *zdrojový kód*, ktorá umožňuje pracovať s *riešením* ako s množinou *zdrojových súborov*. Odkaz na *tím* presne špecifikuje odosielateľa riešenia, odkaz na *problém* určuje problém, ktorý dané *riešenie* rieši a použitý programovací jazyk je daný odkazom na entitu *programovací jazyk*.

3 Používateľský prototyp

3.1 Ciele prototypovania

Cielom nami vytvoreného prototypu bolo vytvorenie používateľského rozhrania, aby mal zákazník čo najskôr predstavu o vzhľade a správaní plánovanej aplikácie. Takýto prototyp umožňuje aj overiť konzistentnosť predstáv zákazníka o správaní sa aplikácie s predstavami vývojárov.

Zamerali sme sa teda na vytvorenie najdôležitejších častí používateľského rozhrania a vytvorenie základu budúcej databázy. Jednotlivé funkcie, ktoré implementujú kontrolu odoslaných zdrojových textov a ich ďalšie spracovanie budú implementované až v ďalšej fáze vývoja.

Z pohľadu implementátorov prototyp slúži na vyjasnenie predstáv o aplikácii medzi jednotlivými členmi tímu a mal by predstavovať aj kostru budúcej aplikácie.

3.2 Softvér potrebný na prevádzku prototypu

Aplikácia je implementovaná ako web – aplikácia v jazyku PHP, pričom dáta sú uložené v databáze MySQL (finálna aplikácia bude používať databázu PostgreSQL).

Server, na ktorom sa prototyp prevádzkuje by mal obsahovať nasledovný softvér:

- Web-server Apache (verzia)
- Databázový server MySQL (verzia)
- PHP (verzia)

3.3 Zhodnotenie prototypu

Prototyp sme vytvárali kvôli vyjasneniu predstavy o budúcej aplikácii pre zákazníka, aj z hľadiska vývojového tímu. Túto úlohu prototyp splnil.

Literatúra

- [1] PC² -The Programming Contest Control System, <http://www.ecs.csus.edu/pc2/>
- [2] UPC2001 Internet Contest, <http://contest.psu.ru/Inet/>
- [3] Ural State University Problem Set Archive with Online Judge System, <http://acm.timus.ru/>
- [4] Internet Programing Contest Of Duke Univerzity, www.cs.duke.edu/~ola/ipc.html.
- [5] Skrátené pravidlá ACM, <http://www.dcs.elf.stuba.sk/acm/rulesacm.html>

Prílohy

Príloha A.

Pravidlá súťaže ACM v skratke

- **NAJSTRUCNEJŠIE POVEDANÉ: 6-10 problémov - 5 hodín času - tím 3 ľudí - iba 1 počítač.**
- **Problémy** majú presne špecifikovaný formát vstupu a výstupu (cítajú text zo štandardného vstupu a zapisujú do štandardného výstupu), dôležité je teda vnútorné spracovanie - správne a podľa možností aj rýchle algoritmy.
- Zadania a komunikácia s usporiadateľmi sú v angličtine.
- Programovať možno v Pascale, C a C++.
- Problém treba naprogramovať, odľadiť a odovzdať zdrojový text programu. Rozhodcovia si ho skompilujú a skúšajú ho na neznámych vstupných údajoch. Keď program pracuje so všetkými (rozhodcovskými) vstupmi správne (t.j. dodá presne očakávané výstupy) a v určenom časovom limite (ktorý je tiež neznámy), riešenie je akceptované (Accepted).
- V prípade neakceptovaného riešenia sa udáva jeden z týchto dôvodov:
 - **Wrong Answer** = program nedodal správny výstup
 - **Presentation Error** = výstup je v podstate správny, ale chybné formátovaný
 - **Time Limit Exceeded** = program neskončil v časovom limite
 - **Run-time Error** = program počas behu spadol alebo skončil s nenulovým návratovým kódom
 - **Compile Error** = program sa nepodarilo skompilovať
 - **Invalid Library Call** = program používa nepovolené operácie
- Poradie vyhodnocovania odpovedí je nasledovné:
 1. Invalid Library Call
 2. Compile Error
 3. Run-time Error
 4. Time Limit Exceeded
 5. Wrong Answer
 6. Presentation Error

7. Accepted

- Programy môžu používať iba štandardné operácie knižníc ANSI. Ide o funkcie definované v hlavičkových súboroch

stdio.h, stdlib.h, ctype.h, string.h, math.h, limits.h, complex.h,
iostream.h

z ktorých povolené sú funkcie

abs atoi bsearch bzero calloc ceil drem exit exp fflush fgetc fgets floor
fmod free getc getchar gets hypot isalnum isalpha isdigit ispunct
isspace log malloc memcmp memcpy memmove memset pow printf
puts qsort realloc scanf setbuf sprintf sqrt sscanf strcasecmp strcat
strchr strcmp strcpy strlen strncasecmp strncmp strncpy strchr strstr
strtok strtol tolower toupper ungetc

a zakázané sú

accept bind fopen fork gethostbyaddr perror listen perror raise signal
sleep socket socketgetstr sockread sockwrite time times vfork waitpid

(Pre Pascal platia zodpovedajúce ekvivalenty.)

- Nie je dovolené akokoľvek sa pokúšať narušiť systém odovzdávania a hodnotenia riešení problémov, alebo hľadať jeho slabiny.
- **Súťažný tím** je trojčlenný a skladá sa zo študentov vysokej školy, z ktorých najviac jeden môže už mať ukončený nejaký stupeň vysokoškolského štúdia (napr. bakalár) a ani jeden neukončil štúdium pred viac ako 2 rokmi, ani nemá titul PhD.
- **Okrem 1 počítača pre tím** nie je povolené používať žiadne elektronické počítače, kalkulčky ani komunikačné zariadenia. Akákoľvek papierová literatúra (vrátane vytlačených listingov apod.) sa môže používať.
- **Hodnotí sa** v prvom rade počet vyriešených problémov a v druhom rade súčet časov (vždy meraných od začiatku súťaže!) úspešných odovzdaní príkladov. Za odovzdanie, ktoré nie je akceptované, je pokuta 20 minút, ale táto sa do celkového času započíta, iba ak nakoniec problém akceptovaný bol. (Z toho vyplýva, že treba ako prvé riešiť najľahšie príklady a opraviť neakceptovaný program sa väčšinou oplatí.)
- S výnimkou poslednej hodiny súťaže sa priebežne zverejňuje, ktoré problémy jednotlivé tímy úspešne odovzdali, a priebežné poradie.
- Po minulé 2 roky (a veľmi pravdepodobne aj tento) sa v stredoeurópskom kole pracovalo pod Linuxom na textových termináloch, s použitím kompilátora GNU.

