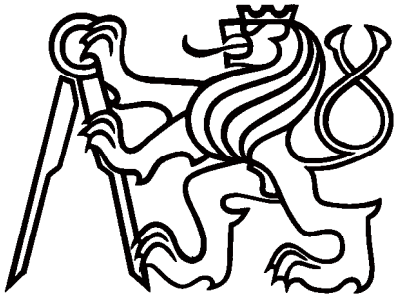


České vysoké učení technické
v Praze
Fakulta stavební

Katedra mapování a kartografie

Jan Pražák

Tvorba testovacího modulu internetového výukového kurzu

DIPLOMOVÁ PRÁCE

Vedoucí diplomové práce: Ing. Petr Soukup, Ph.D.

Praha, prosinec 2004

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně s použitím uvedených zdrojů a literatury.

V Praze dne 1.prosince 2004

Jan Pražák

Touto cestou bych rád poděkoval všem, kteří významnou měrou přispěli ke vzniku této diplomové práce, byť jen malou radou či připomínkou. Zejména Ing. Petrovi Soukupovi Ph.D. za vedení diplomové práce, cenné rady a připomínky.

Jan Pražák

Obsah:

Anotace

Diplomová práce se zabývá tvorbou konfiguračního modulu testovacího systému, který bude používán k výuce a testování studentů formou e-learningu. V programové části byl použit převážně programovací jazyk PHP, datová struktura souboru je XML. Součástí diplomové práce jsou ukázkové sady otázek pro výukový kurz předmětu Informatika 1 vyučovaný na katedře mapování a kartografie. Zdrojové kódy programů jsou obsahem webové prezentace diplomové práce dostupné na adrese http://gama.fsv.cvut.cz/~soukup/dip/prazak/index.html.

1. Cíl diplomové práce

Úkolem diplomové práce bylo vytvoření konfiguračního modulu testovacího systému a jeho propojení s vlastním testovacím systémem. Tento modul je jedním ze základních stavebních kamenů celého testovacího systému, který bude důležitým pomocníkem a zároveň zatraktivní a zefektivní výuku a testování znalostí studentů na katedře mapování a kartografie. Celý systém je založen na jednotné datové struktuře a programovacím jazyku. Je vyvíjen tak, aby jednotlivé moduly bylo možno na sebe jednoduše napojovat. Vedle klasické papírové podoby diplomové práce jejíž součástí je i datový nosič s digitální verzí, zdrojovými soubory a dalšími potřebnými programy a utilitami bylo úkolem i vytvoření webových stránek, které obsahují plné znění diplomové práce včetně výpisu zdrojových kódů.

Součástí úkolu bylo také vytvoření několika sad vzorových testovacích otázek z vybraných kapitol předmětu Informatika 1, jehož absolvování je jednou z podmínek bakalářského studia na katedře mapování a kartografie.

Upozornění: V diplomové práci jsou použity názvy produktů a technologií, na něž se vztahují příslušné autorské a patentové zákony.

2. E-Learning

2.1. Jak vznikl E-Learning

E-learning má své kořeny v distančním vzdělávání (DiV) [1],[2].

Distanční vzdělávání je specifická multimediální forma řízeného samostudia, která je vhodná pro průběžné vzdělávání dospělých. Potřeba DiV vychází z nároků informační společnosti požadující, aby se lidé průběžně vzdělávali. Učení se tak, více než kdy předtím, stává klíčem k úspěchu. Rychlý vývoj však způsobuje, že klasická podoba vzdělávání přestává být efektivní. Proto společnost hledá nové, netradiční a mnohem efektivnější formy vzdělávání. Příležitostí pro účinnou realizaci celoživotního vzdělávání je využití informačních a komunikačních technologií (ICT).

Na rozdíl od prezenční formy studia jsou účastníci distančního studia od sebe odděleni (v čase, v prostoru). Průběh studia koordinuje příslušná vzdělávací instituce a učivo je předáváno prostřednictvím rozmanitých multimediálních prostředků. Úspěšná realizace DiV vyžaduje vybudování komplexního systému podpory studia, který stojí na čtyřech základních pilířích:

- organizace studia
- studijní materiály
- tutor[1)]
- osobní kontakt mezi tutorem a studentem

Ve všech těchto aspektech musí být promyšleně uplatňován princip multimediality.

Využití ICT přispívá ke zkvalitnění organizace studia, ke zvýšení hodnoty vzdělávacího procesu a usnadňuje dosahování pedagogických cílů.

2.2. Historie distančního vzdělávání

U vzniku distančního vzdělávání stála snaha o výuku těsnopisu. Vše započalo v roce 1837, kdy Angličan Isaac Pitman zahájil výuku těsnopisu formou korespondenčních kurzů rozesílaných poštou. Sestrojení prvního bezdrátového telegrafu (1895) pak odstartovalo vývoj elektronického vzdělávání. Následoval vynález elektroniky a televize, čímž byl přenos informací rozšířen o vizuální složku. V 80.letech 20.století probíhá výuka již za pomocí počítače. Bylo dosaženo maxima po stránce

[1)] tutor je označení lektora (konzultanta) v distančním kurzu

multimediální. Po stránce komunikace a přístupu k informacím jsme dosáhli vrcholu vybudováním celosvětové sítě. Pojem e-learning se objevuje až od roku 1999. V tomto roce začaly vznikat na internetu vzdělávací portály nabízející nové typy zázemí v oblasti vzdělávání.

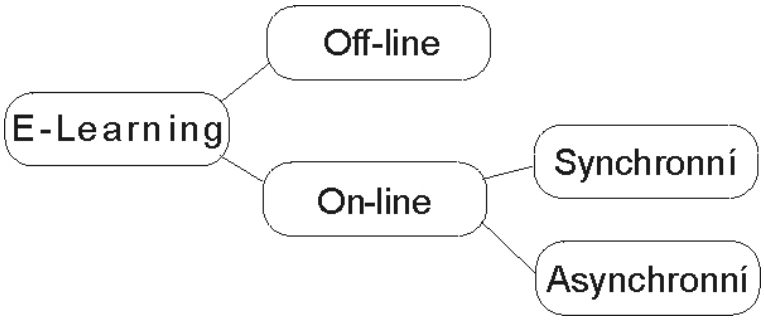
Ve vývoji DiV můžeme rozlišit tři generace distančního vzdělávání:

- I.generace – korespondenční kurzy
- II.generace – DiV + audio, video a PC technologie
- III.generace – DiV + síť (internet, intranet)

Síť je „všudypřítomné médium“, které usnadňuje zejména navázání kontaktů a čerpání informací z alternativních zdrojů a navzájem propojuje klíčové účastníky DiV. Přítomnost sítě umožňuje studujícím pružně komunikovat nejen se svým tutorem, ale i ostatními distančními studujícími. Studující již není odkázán jen na odborné materiály poskytnuté vzdělávací institucí.

E-learning je pouze podpora vzdělávacího procesu spojená s moderními informačními a komunikačními technologiemi, jejichž aplikace má vést ke zkvalitnění vzdělávání. V oblasti elektronického vzdělávání se můžeme setkat s pojmy jako jsou: vzdělávání podporované technologiemi (TBL), vzdělávání podporované počítači (CBT), vzdělávání podporované webovými technologiemi (WBT) a on-line vzdělávání (OL) (podrobněji vysvětleno v kapitole 2.3.). Ačkoliv e-learning představuje především podporu distančního vzdělávání, nalézá také efektivní uplatnění jako doplněk prezenční formy výuky. Takovémuto smíšenému způsobu vzdělávání říkáme „blended learning“.

Dle využití ICT dělíme e-learning na off-line výuku (tj. počítač nemusí být napojen na síť) a on-line výuku (vyžaduje zapojení počítače do sítě). Dnes již považujeme připojení na síť za nezbytné a pod pojmem e-learning si představujeme pouze on-line formu výuky. Tu podle způsobu připojení a možnosti komunikace rozdělujeme na formu asynchronní (komunikace mezi účastníky probíhá v rozdílném čase) a synchronní (komunikace mezi účastníky se uskutečňuje v reálném čase).



Obr. 2.1: Rozdělení e-learningu

Stejně jako každý prostředek i e-learning má své silné a slabé stránky. Mezi výhody lze zařadit především vyšší efektivnost výuky, nižší finanční náklady a časovou nezávislost. Hlavními nevýhodami jsou vysoká náročnost na tvorbu kurzů a závislost na funkčnosti ICT. Aby takto náročně vytvořené kurzy byly kompatibilní s různými systémy pro řízení výuky, musí být tvořeny v souladu s danými standardy (např. SCORM[1)]).

2.3. Tři úrovně elektronického vzdělávání:

CBT (Computer-Based Training) neboli „vzdělávání za podpory počítačů“ je první úrovní vzdělávání. Většinou je tato forma považována pouze za off-line vzdělávání. Veškeré programy a kurzy jsou distribuovány na datových nosičích (např. disketa, CD, DVD). Další uvedené úrovně sice již budou vyžadovat připojení k internetu, ale přesto se bude stále jednat o využívání počítačů (ikdyž již připojených k síti), tudíž i tyto následující úrovně budou spadat pod úroveň první.

WBT (Web-Baset Training) - Druhá úroveň elektronického vzdělávání je založena na podpoře webu. Jde o vzdělávání pomocí webových technologií. Jedná se již o on-line formu, programy a kurzy jsou distribuovány přes internet a intranet. Toto připojení je, kromě distribuce kurzů, důležité především pro navázání komunikace mezi studentem a tutorem a mezi studenty navzájem, a to jak synchronně (např. chat), tak asynchronně (např. e-mail). Výhodou kurzů podporovaných webovými technologiemi je možnost okamžité aktualizace informací. Další nespornou výhodou WBT oproti CBT bez připojení na síť je skutečnost, že aktualizace informací, které jsou součástí nějakého modulu v kurzu, může být provedena téměř okamžitě a bez jakýchkoliv nově vzniklých finančních nákladů. Při aktualizaci dat v kurzech vedených formou CBT bez připojení k síti je potřeba provést redistribuci nosičů, na kterých je kurz distribuován, a to stojí jak

[1)] Shareable Content Object Reference Model

peníze, tak i čas. I přes nesporné výhody WBT je i tento stále jen způsob využívání počítačů při vzdělávání (i když již v mnohem rozšířenější formě).

LMS (Learning Management Systém) – tzv. systém pro řízení výuky, v současné době nejdokonalejší úroveň elektronického vzdělávání. Studující k takto řízeným kurzům přistupují stejně jako při WBT. Podstatný rozdíl je zde v podpoře především pro tutory, vývojáře a autory. I elektronické vzdělávání na bázi LMS je způsob vzdělávání využívající počítače, a tudíž i tento způsob řízení výuky spadá pod CBT. Můžeme říci, že řízení výuky přes LMS je vlastně v dnešní době nejvyšší forma WBT, neboť všichni účastníci do tohoto systému vstupují přes internetový prohlížeč.

2.4. Základní charakteristika a funkce LMS

Zkratka LMS pochází z anglického názvu Learning Management Systém, což česky znamená Systém pro řízení výuky. Jedná se o speciální software instalovaný na serveru poskytující mnohostrannou podporu výuky. Je to soubor nástrojů, které umožňují tvorbu, správu a užívání kurzů v elektronickém prostředí. Tyto kurzy jsou rozšiřovány prostřednictvím internetu nebo intranetu, takže je možné do nich odkudkoliv vstoupit pomocí běžného internetového prohlížeče. Kromě nástrojů pro tvorbu, správu a distribuci kurzů obsahuje LMS nástroje pro komunikaci, a to jak pro komunikaci mezi studujícími a tutorem, tak pro komunikaci mezi studenty navzájem. Dalším důležitým znakem LMS jsou nástroje pro hodnocení studijních výsledků a zpětnou vazbu. Kromě systémů LMS existují ještě systémy LCMS (Learning Content Management System), které mají na starosti obsahové stránky kurzů. Jelikož systémy LMS jsou stále dost nákladné a pro vzdělávací instituce poměrně velkou investicí. Je potřeba při výběru LMS zvolit co nejlepší strategii, aby vybraný systém splňoval požadovaná kritéria úměrně k ceně.

Základní kritéria, která by měl LMS splňovat:

- rozhranní umožňující vytvářet a prezentovat kurz
- soubor výukových nástrojů, které usnadňují samostudium, komunikaci a spolupráci
- soubor administrativních nástrojů, které pomáhají realizačnímu týmu a tutorům v procesu správy, vedení a zlepšování kurzu
- příznivá cena (úměrně ke kvalitě, možnostem a servisu)

2.5. E-learning na akademické půdě

Rozvoj e-learningu kráčí ruku v ruce s rozvojem informačních a komunikačních technologií. Managementy škol i pedagogická veřejnost si stále silněji uvědomují, jaké příležitosti se díky novým technologiím otevírají. Již v 90.letech minulého století začaly české vysoké školy, za podpory státu, budovat nezbytnou technickou a technologickou infrastrukturu – počítačové sítě, připojené na internet. Jako první vybudovalo svou síť České vysoké učení technické v Praze již v roce 1990. Fenomén e-learningu v rámci pedagogického procesu začíná působit o pár let později, současně s rozvojem distanční formy vzdělávání. E-learning je od roku 1999 podporován Národním centrem distančního vzdělávání v Praze [13]. Hlavní důraz je přitom kladen právě na profesní rozvoj pedagogů v této oblasti. Za podpory programu Phare byl realizován distanční kurz LOLA (Learning about Open Learning) s e-learningovou podporou. Cílem kurzu bylo ukázat, co mohou nebo naopak nemohou pedagogové očekávat od vzdělávacích platforem pro e-learning.

Zavádění e-learningu na půdu vysokých škol je výrazně podporováno Ministerstvem školství, mládeže a tělovýchovy ČR (MŠMT) [12], respektive státem. Kromě morální podpory a ocenění se školám dostává zajímavá finanční podpora prostřednictvím podpůrných programů. Právě v těchto měsících vyhlásilo MŠMT ČR témata tzv. rozvojových programů na rok 2005. Tyto programy jsou zaměřeny na rozvoj informačních a komunikačních technologií podporujících akreditované studijní programy a tvorbu multimediálních vzdělávacích pomůcek. Dále MŠMT ČR podpoří projekty zaměřené na další vzdělávání akademických pracovníků v oblasti informačních technologií a moderních výukových postupů. V rámci podpory rozvoje struktury je preferováno rozšíření nabídky akreditovaných studijních programů v kombinované a distanční formě.

Ruku v ruce s finanční podporou ministerstvo deklaruje také konkrétní požadavky na charakter distančního, respektive kombinovaného studia. V těchto studijních programech musí být jednoznačně popsán způsob výuky v distanční části, včetně začlenění e-learningu. Nezbytně nutné je zajištění možnosti komunikace studenta s vyučujícím přes internet. Informačních a komunikačních technologií má být využíváno také pro distribuci studijní literatury a zejména pro zkvalitňování jejich pedagogické úrovně, vlivem multimediálních prvků.

Vysoké školy se v tomto ohledu vyrovnávají s náročným úkolem. Zavedení e-learningu vyžaduje nejen značné finanční investice do technologií a technického vybavení, ale je nezbytné provést kvalitní odbornou přípravu akademických pracovníků a připravit takové motivační nástroje, které přesvědčí o atraktivitě e-learningu.

Školy využívající e-learning v ČR:

- Univerzita Karlova v Praze
- Univerzita Hradec Králové
- Virtuální univerzita VIRTUNIV
- Univerzita Palackého v Olomouci

3. Programové nástroje

3.1. HTML

HTML (Hypertext Markup Language) je základním jazykem pro vytváření webových stránek [5]a jeho zvládnutí je nutné i pro používání složitějších skriptovacích jazyků, jako je PHP či ASP.

„Oficiální“ verzi jazyka HTML utváří konsorcium W3C [14]. Slovíčko oficiální jsem zde použil hlavně proto, že dva hlavní výrobci prohlížečů, Netscape [6], ale hlavně Microsoft [9], neustále rozšiřují HTML o svoje příkazy a až v momentě, kdy "naučí" svůj prohlížeč je podporovat, se snaží, aby je konsorcium W3C uznalo za oficiální standard HTML.

HTML je interpretovaným jazykem, což znamená, že zdrojový kód je přímo předložen prohlížeči, není tedy překládán do strojového kódu. To také znamená, že žádné vnější elementy, které jsou přeloženy do strojového kódu (obrázky, animace), se nenachází uvnitř HTML-dokumentu, ale je na ně z něj pouze odkazováno. HTML je zároveň jazyk velmi pružný, pokud totiž prohlížeč narazí na chybu, nebo na příkaz, který nezná, tak ho ignoruje a přesune se na další příkaz.

3.2 DHTML

Na rozdíl od podrobných specifikací (jako je např. HTML 4.01 či XHTML 1.0) není dynamické HTML (dále jen DHTML) žádným standardem či normativním předpisem; neexistuje nic jako DHTML 1.0. Termínem DHTML [8] se pouze označuje souhrn nástrojů a postupů umožňujících dynamické změny HTML dokumentů. Tedy postupy, jak přímo v prohlížeči na jinak statické stránce zobrazit či skrýt část jejího obsahu, provést určitou akci podle činnosti uživatele (pohyb myši, kliknutí, stisk klávesy, načtení stránky atd.), animovat pohyb prvků na stránce či automatizovat funkce prohlížeče.

3.3. XHTML

V další fázi bylo potřeba definovat HTML jako podmnožinu jazyka XML (eXtensible Markup Language - rozšiřitelný značkovací jazyk), který vychází ze SGML, avšak je „ořezán“ o některé zbytečné vlastnosti a některé nové vlastnosti naopak přidává. W3C prosazuje XML jako hlavní a jediný značkovací jazyk nejen pro web,

proto byla reformulace HTML do XML logickým vyústěním této snahy. Nově vytvořený jazyk dostal jméno XHTML (eXtensible HyperText Markup Language - rozšiřitelný hypertextový značkovací jazyk) a jeho první specifikace se označuje XHTML 1.0. Tato specifikace je stejná jako specifikace HTML 4.01, jsou zde pouze integrována pravidla XML.

W3C chce XHTML prosazovat jako jediný jazyk pro definici webových stránek, bez ohledu na typ zařízení, které je zpracovává. Expanze webu na různá alternativní zařízení ukázala, že tato zařízení kvůli svým omezeným možnostem nemohou podporovat všechny vlastnosti XHTML, že podporují jen nějakou jeho část (podmnožinu). Tyto podmnožiny bylo třeba definovat a standardizovat, proto vznikla specifikace „Modularization of XHTML“. Tato specifikace rozděluje všechny prvky XHTML 1.0 do modulů, ze kterých se následně skládají značkovací jazyky. Skupiny zabývající se webem v alternativních zařízeních mohou definovat nové moduly s prvky specifickými pro dané zařízení, mohou stávající moduly modifikovat, ale hlavně mohou z modulů skládat nové kompletní značkovací jazyky, které vyhovují potřebám a možnostem interpretace webu na těchto zařízeních. Z těchto jazyků zatím W3C uznalo XHTML 1.1 (až na několik detailů se shoduje s XHTML 1.0 Strict, hlavní rozdíl spočívá v tom, že XHTML 1.1 je definováno pomocí modulů) a XHTML Basic (skládá se pouze ze základních modulů, k použití hlavně na mobilních telefonech, PDA[1)] a podobně).

3.4. CSS

CSS - Cascading Style Sheets - kaskádovací styly, poprvé implementovala společnost Microsoft v roce 1996 do Internet Exploreru 3.0. Tato technologie, zcela nahrazuje tag[2)] <font> a uvádí v provoz tag <style>. Pomocí CSS můžeme definovat kromě barvy, písma a velikosti spoustu dalších věcí (rámeček, podtržení, tučnost, vlnitost, zobrazení, odrážky, okraje..) Třídy a identifikátory umožňují tvorbu stylu jediným atributem a nemusíme tedy ve zdrojovém kódu opakovat stejné tagy vícekrát. Kromě toho můžete také definovat styl tagů např. tak <input> bude mít vždy červený text - to je možné udělat jediným řádkem.

[1)] Personal Digital Assistent, v překladu osobní digitální pomocník, nebo-li počítač do dlaně. Má operační systém a uživatelské či systémové programy. Jsou zažité i jiné= názvy například handheld, palmPC, palmtop

[2)] .tag je kód ohraničený dvěma znaménky < > ; nejsou zobrazovány, ale určují, co má prohlížeč „udělat“

3.5. PHP

PHP [11] urazilo v posledních několika málo letech dlouhou cestu. Růst v jeden z nejprominentnějších jazyků ovládajích Web nebyl snadný. V následujících odstavcích je stručně popsáno, jak PHP vyrostlo do dnešní podoby [7]. Tento jazyk byl zároveň z důvodu jeho naprosté univerzálnosti v použití a podpory ze strany serveru katedry mapování a kartografie vybrán a použit k naprogramování celého modulu a je základním jazykem celého testovacího systému.

3.5.1. PHP/FI

PHP je nástupcem staršího produktu, nazvaného PHP/FI. PHP/FI vytvořil Rasmus Lerdorf v roce 1995, na počátku jako jednoduchou sadu skriptů v jazyce Perl pro zpracování záznamů o přístupech k jeho webu. Tuto sadu nazval 'Personal Home Page Tools'. Protože byla třeba větší funkčnost, napsal Rasmus mnohem rozsáhlejší implementaci v C, která byla schopna komunikovat s databázemi a umožňovala uživatelům vyvíjet jednoduché dynamické aplikace pro Web. Rasmus se rozhodl uvolnit zdrojový kód PHP/FI pro všechny, takže kdokoli ho může používat, stejně jako opravovat chyby a vylepšovat kód.

PHP/FI, což znamená Personal Home Page / Forms Interpreter, obsahovalo něco ze základní funkcionality PHP, jak ho známe dnes. Mělo proměnné Perlovského typu, automatickou interpretaci formulářových proměnných a syntaxi vloženou do HTML. Syntaxe samotná byla podobná jazyku Perl, přestože mnohem omezenější, jednodušší a v něčem nekonzistentní.

V roce 1997 se PHP/FI 2.0, druhá implementace psaná v C, stala kultovní záležitostí pro (odhadem) tisíce uživatelů po celém světě, a s přibližně 50.000 doménami oznamujícími nainstalované PHP/FI, což čítalo zhruba 1 % všech domén na Internetu. I když do projektu začalo svými kusy kódu přispívat více lidí, stále to byl velký projekt jednoho muže.

PHP/FI 2.0 bylo oficiálně uvolněno až v listopadu 1997, poté co strávilo většinu svého života v betaverzích. Krátce nato bylo následováno první alfaverzí PHP 3.0.

3.5.2. PHP 3

PHP 3.0 byla první verze, která se velmi blížila takovému PHP, jak ho známe dnes. Vytvořili ho Andi Gutmans a Zeev Suraski v roce 1997 jako kompletně přepsaný

celek, poté co shledali PHP/FI 2.0 výrazně "poddimenzované" pro vývoj svých aplikací pro e-komerci. Ve snaze spolupracovat a zahájit budování nad existující uživatelskou základnou PHP/FI, rozhodli se Andi, Rasmus a Zeev pracovat společně a prohlásit PHP 3.0 za oficiálního nástupce PHP/FI 2.0, a vývoj PHP/FI 2.0 byl v podstatě zastaven.

Jednou z nejsilnějších zbraní PHP 3.0 byly jeho obrovské možnosti rozšíření. K poskytnutí pevné infrastruktury pro mnoho různých databází, protokolů a API koncovým uživatelům, přilákaly možnosti rozšíření PHP 3.0 také tucty vývojářů, kteří se připojili a vytvořili nové rozšiřující moduly. Toto byl nesporně klíč k obrovskému úspěchu PHP 3.0. Jiným klíčovým prvkem v PHP 3.0 byla podpora objektově orientované syntaxe a mnohem silnější a konzistentnější syntaxe jazyka.

Nový jazyk byl uvolněn pod novým názvem, který odstranil implikaci omezeného osobního použití, kterou neslo označení PHP/FI 2.0. Byl nazván pouze 'PHP', což je rekurzívní akronym - PHP: Hypertext Preprocessor.

Na konci roku 1998 vyrostlo PHP do rozsahu instalací v řádu (odhadem) desítek tisíc uživatelů a stovek tisíc Webů. V době svého vrcholu bylo PHP 3.0 instalováno na přibližně 10% všech WWW serverů na Internetu.

PHP 3.0 bylo oficiálně uvolněno v červnu 1998, poté co strávilo cca 9 měsíců ve veřejném testování.

3.5.2. PHP 4

V zimě 1998, krátce po oficiálním uvolnění PHP 3.0, začali Andi Gutmans a Zeev Suraski pracovat na přepsání jádra PHP. Cílem návrhu bylo zvýšit výkon pro složité aplikace a zlepšit modularitu kódové báze PHP. Takové aplikace byly schopny pracovat s PHP 3.0 (díky novým možnostem a podpoře široké škály databází a API od jiných tvůrců), ale PHP 3.0 nebylo navrženo pro efektivní práci tak náročných aplikací.

Nový engine[1)], nazvaný 'Zend Engine' (sestaven z jejich křestních jmen, Zeev a Andi), úspěšně splnil cíle návrhu a byl uveden v polovině roku 1999. PHP 4.0, založené na tomto enginu a doplněné širokou škálou nových prvků, bylo oficiálně uvolněno v květnu 2000, necelé dva roky po svém předchůdci, PHP 3.0. K podstatně zvýšenému výkonu této verze, přidává PHP 4.0 další klíčové prvky, jako je podpora pro mnoho WWW serverů, HTTP sessions, buffering výstupu, bezpečnější způsoby zpracování vstupů uživatele a mnoho nových jazykových konstruktů.

[1)] engine = programový základ; jádro

3.5.3 PHP 5

Nedávno vydaná verze PHP 5 je momentálně poslední uvolněnou verzí PHP. Oproti verzi PHP 4 došlo k vylepšení jádra Zend Engine k integraci nových prvků, které byly navrženy nově pro PHP 5.0.

Dnes používají PHP (odhadem) stovky tisíc vývojářů a nainstalované PHP hlásí několik milionů serverů - tj. přes 20 % domén na Internetu.

Vývojový tým PHP zahrnuje tucty vývojářů, stejně tak jako tucty dalších lidí, kteří pracují na projektech spojených s PHP, jako je PEAR a dokumentační projekt.

3.6. ASP

Zkratka z anglického Active Server Pages. Jde o tenchnologii, která generuje stránky na straně serveru. ASP je dítětem společnosti Microsoft – ta tuto technologii představila v prosinci 1996 jako součást svého internetového serveru Internet Information Server 3.0 (IIS). Žádný programovací jazyk ASP ale neexistuje – to je jen obecný název, zastřešující několik technik. Díky nim lze snadno a poměrně jednoduše integrovat databázové systémy a podnikové aplikace do webových stránek a zpřístupňovat tak důležitá data velkému okruhu uživatelů. Základem ASP je programovací jazyk. K nejznámějším patří např. JavaScript nebo VBScript (podrobněji v následujících kapitolách).

3.7. JavaScript

JavaScript je jednoduchý programovací jazyk, vyvinutý pro zvýšení komfortu při prohlížení internetových stránek. Můžete ho zapisovat přímo do HTML kódu. Tento jazyk vyvinula firma Netscape, ale v současné době je podporován Internet Explorerem i řadou dalších prohlížečů. Navzdory názvu nevychází JavaScript z programovacího jazyka Java. Ten je produktem jiné firmy (Sun Microsystems).

3.8. VB Script

VBScript je skriptovací jazyk, který se odehrává na straně prohlížeče. Proto skripty napsané ve VBScriptu jsou funkční i v režimu offline. Možnosti VBScriptu jsou takřka totožné jako schopnosti JavaScriptu (viz kap. 3.7).

JavaScript a VBScript jsou velmi podobné jazyk, co do cílů a možností. Liší se především syntaxí a samozřejmě některé drobné rozdíly (jeden umí něco, co druhý

neumí), ale tyto rozdíly jsou zanedbatelné. Hlavní rozdíl však plyne z historie obou jazyků. Zatímco JavaScript vytvořila společnost Netscape a tím podpořila schopnosti svého prohlížeče, Microsoft reagoval podporou JavaScriptu, ale vymyslel i svůj vlastní, který má nahradit JavaScript, ale tento jazyk – VBScript zůstává výhradou pouze prohlížečů Microsoftu. VBScript je tedy převážně funkční pouze v Internet Exploreru, což zcela jednoznačně odpovídá na otázku, který skriptovací jazyk použít. Pokus použijeme VBScript, uživatelé jiných prohlížečů (Opera, Mozilla) budou ochuzeni, případně omezeni, což je rozhodně nežádoucí. VBScript bychom tedy mohli použít v našem testovacím systému, protože na všech počítačích je použit prohlížeč od společnosti Microsoft. Ale pro zachování budoucí plné kompatibility není tento skriptovací jazyk pro náš případ příliš vhodný.

4. Instalace

4.1. Průvodce instalací

Většinu času při vývoji aplikací strávíme jejich psaním a laděním. Není samozřejmě nutné pokaždé se připojovat na vzdálený server s nainstalovanou podporou PHP (a příp. i dalších skriptů), ale můžeme si na svém počítači vytvořit vlastní server (tzv. localhost), který se bude tvářit stejně jako při připojování na vzdálený server. Pro moji práci jsem zvolil server Apache, který vznikl na stejné bázi jako PHP a je zároveň používán na téměř 50% všech WWW-serverů. Systém psaný jazykem PHP však není vázaný jen na Apache HTTP Server, protože jazyk PHP může obecně běžet na libovolném serveru (Personal Web Serve, Internet Information Server v.3.0 nebo 4.0 atd.) a libovolné platformě (Windows, Linux atd.). Pro nakonfigurování systému tak, aby na něm správně fungovaly stránky a aplikace psané ve skriptovacím jazyku PHP, je nutné provést několik úkonů, které jsou popsány v následujících odstavcích v pořadí, v jakém by měly následovat. Postup je popsán pro operační systém Microsoft Windows 9x/NT/2000/XP [9], postup pro instalaci pod jinými operačními systémy je popsán v příslušné literatuře [7].

4.2. Instalace Apache Server

Nejjednodušší je Apache HTTP Server [10] nainstalovat pomocí instalátoru, který je součástí binární verze souboru.

Obr. 4.1: Instalátor Apache HTTP Serveru

Vývoj programu se stále vyvíjí, momentálně nejnovější verze je 2.0.52, která je také nedílnou součástí této diplomové práce na datovém nosiči. Instalátor nás pohodlně provede instalací. Po nastavení složek programu a požadovaných součástí se dostaneme na stránku Server Information, kde do prvních dvou polí (Network Domain, Server Name) vepíšeme místo původního textu jen localhost. Do posledního pole (Administrator's Email Address) napíšeme admin@localhost.

Obr. 4.2: Konfigurační okno Apache HTTP Serveru

Po kontrole cesty k instalaci, která by měla být C:/Apache se opět proklikáme celým procesem instalace.

4.3. Instalace PHP 4

Nyní již máme nainstalován a nezkonfigurován Apache, doinstalujeme tedy PHP. I jazyk PHP samozřejmě prochází dlouhým vývojem, na datovém nosiči je uložena verze PHP 4.3.9 [11]. Celý obsah souboru .zip, který jsme si stáhli, rozbalíme do zvolené složky (např. C:/PHP). Mezi rozbalenými soubory je i soubor php4-dist.ini, který zkopírujeme do adresáře, ve kterém máme uložen operační systém Windows (např. C:\Windows). Tento soubor přejmenujeme na php4.ini a nastavíme v něm cestu k DLL-knihovnám jazyka PHP (tzn. že direktivě extension_dir nastavíme hodnotu adresáře, do kterého jsme knihovny nakopírovali (tzn. např. C:\PHP). Dále je nutné nastavit adresář, který bude brán jako kořenový adresář dokumentů webového serveru (např. C:\Skript). Nyní je PHP na našem systému nainstalováno, zbývá přidat podporu do Apache Serveru.

4.4. Nastavení podpory PHP na Apache Serveru

Konfigurace se provádí upravením souboru httpd.conf, který nalezneme v adresáři Conf nainstalovaného Apache Serveru (např. C:\Program Files\Apache Group\Apache\Conf). Do souboru je nutné upravit následující direktivity:

- ScriptAlias /php/ “c:/php/”
- AddType application/x-httpd-php .php .phtml .html .htm
- Action application/x-httpd-php “/php/php.exe”

Těmito direktivami určujeme cestu ke knihovnám jazyka PHP a podporované přípony souborů, které bude server uvažovat jako možné PHP skripty.

Po uložení a zavření souboru je nutné celý Apache Server restartovat, aby došlo k načtení nových konfiguračních souborů.

Nyní pro kontrolu funkčnosti serveru si napíšeme tento jednoduchý skript (např. v libovolném textovém editoru):

```
<?php
 PHPInfo ();
?>
```

Tento skript uložíme s libovolným názvem s příponou .php (např. test.php) do kořenového adresáře webového serveru (C:\Skript). Poté spustíme libovolný internetový prohlížeč a zadáme URL našeho skriptu tj. http://localhost/test.php. Pokud bylo vše v pořádku nakonfigurováno, měla by se zobrazit stránka, která obsahuje informace o používané verzi jazyka PHP.

Obr. 4.3: Výpis konfigurace PHP v prohlížeči

Tímto je počítač připraven ke spouštění skriptů PHP bez nutnosti připojení k externímu serveru (např. přes internet).

4.5. Hypertext Transfer Protokol

Protokol HTTP (Hypertext Transfer Protokol) je jedním ze stavebních kamenů celé služby WWW (World-Wide Web). Používá se pro komunikaci mezi prohlížečem a serverem. Klient (v našem případě prohlížeč) se spojí se serverem a pošle mu požadavek. Server jako reakci na klientův požadavek zasílá odpověď. Aby si spolu klient a server rozuměly, musí používat jistá pravidla komunikace, kterým se říká protokol. Přesný formát požadavku a odpovědi je v tomto případě definován ve

specifikaci protokolu HTTP. Celou situaci mírně komplikuje skutečnost, že dnes existují tři verze protokolu – 0.9,1.0 a 1.1. Formát požadavku a odpovědi se v jednotlivých verzích odlišuje.

Protokol HTTP je aplikačním protokolem, který pro vlastní přenos dat sítí používá protokol nižší úrovně, jež zajišťuje spolehlivé datové spojení. Tímto protokolem je nejčastěji TCP. Klient se nejčastěji připojuje k WWW-serveru na port 80, i když je možno v konfiguraci serveru vybrat libovolný jiný port.

4.6. Cookies

Vlastní aplikace napsané v PHP mají jistá nepříjemná omezení vyplývající z principu protokolu HTTP. Protokol HTTP je nestavový, tzn. že pro přenos každé stránky se otevírá nové zvláštní HTTP spojení, které se ihned po přenosu uzavře. Částečnou možností řešení tohoto problému je ukládání pomocných stavových informací do skrytých polí formuláře anebo tyto informace přidávat do cesty v URL za jméno skriptu. Tyto techniky jsou využívány poměrně často a mnoho problémů uspokojivě vyřeší. Ovšem ani tyto dva způsoby neřeší problém trvalého uložení nějakých informací. Další možností, jak tento problém vyřešit jsou tzv. cookies, což je rozšíření protokolu HTTP pocházející z dílny firmy Netscape. Dnes je toto rozšíření podporováno snad všemi prohlížeči.

4.7. Zabezpečení

Přístup k vlastnímu nastavení testu a vyhodnocení výsledků by mělo být logicky zabezpečeno před zraky neoprávněných uživatelů. Tento přístup je podmíněn v našem případě přiděleným závazným uživatelským jménem a heslem, přiděleným oprávněnému uživateli správcem systému.

V našem sytému program si pomocí jednoduché stránky s formuláři vyžádá od uživatele zadání uživatelského jména a hesla. Toto jméno a heslo je vráceno zpět skriptu, který má v proměnných $PHP_AUTH_USER, $PHP_AUTH_PW a $PHP_AUTH_TYPE dostupné uživatelské jméno, heslo a typ autentifikace (zatím je skriptovacím jazykem PHP podporována pouze autentifikace typu basic). Obsah proměnných skript dále porovnává s tabulkou uživatelů, kteří mají přidělené uživatelské jméno a heslo a tím i oprávněný přístup do systému.

Prohlížeče si během svého spuštění pamatují jméno a heslo pro danou aplikaci (určenou pomocí parametru realm). Uživateli tedy stačí zadat jméno a heslo pouze jednou, dále je již automaticky posílá prohlížeč.

Zabezpečení se samozřejmě netýká pouze první (přihlašovací) stránky, ale také je ochráněn přístup na ostatní stránky skriptu, aby tak bylo znemožněno prostým napsáním jiné stránky do příkazového řádku vstup na tyto zabezpečené stránky bez hesla.

4.8. Instalace konfiguračního modulu Testovacího Systému

4.8.1. Programové soubory

Nyní když máme plně funkční server s podporou skriptů PHP na našem počítači, můžeme přikročit k vlastní instalaci souborů konfiguračního modulu. Z CD nosiče z adresáře „/Instalace/soubory“ překopírujeme do libovolného adresáře na našem webserveru překopírujeme soubory potřebné pro test tj. *index.html, login.php, hlavni.php, nastaveni.php, otazky.php, fce.php, logout.php, protection.php, test.css.*

4.8.2. Testy a konfigurační soubory

Soubory hotových testů a konfiguračních souborů (tzn. soubory *.xml) nakopírujeme do podadresáře .\testy příp. .\cfg.

4.8.3. Nastavení práv

Pro adresáře kde máme uloženy testy, nebo kam chcete ukládat výsledky je nutné zkontrolovat nastavení práv plného čtení a zápisu.

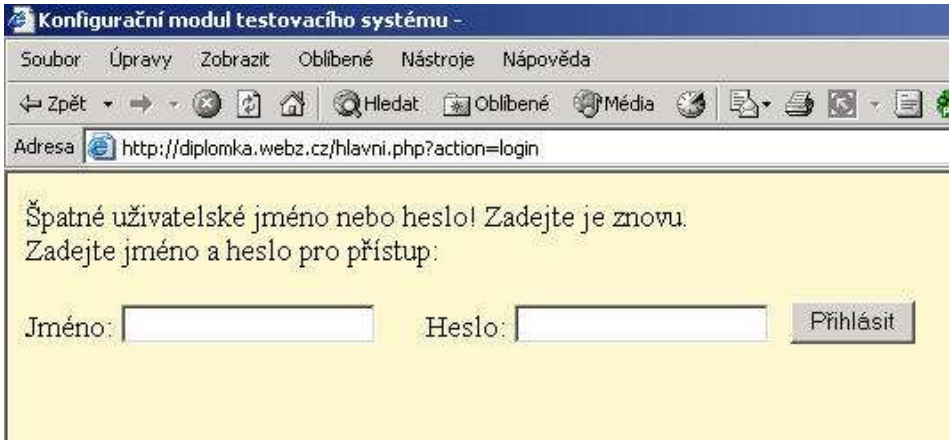
4.8.4. Uložení hotových testů a konfiguračních souborů

K ukládání souborů dochází do stejného adresáře jako jsou uložené původní soubory tzn. .\testy příp. .\cfg.

5. Průvodce konfiguračním modulem

5.1 Přihlášovací okno

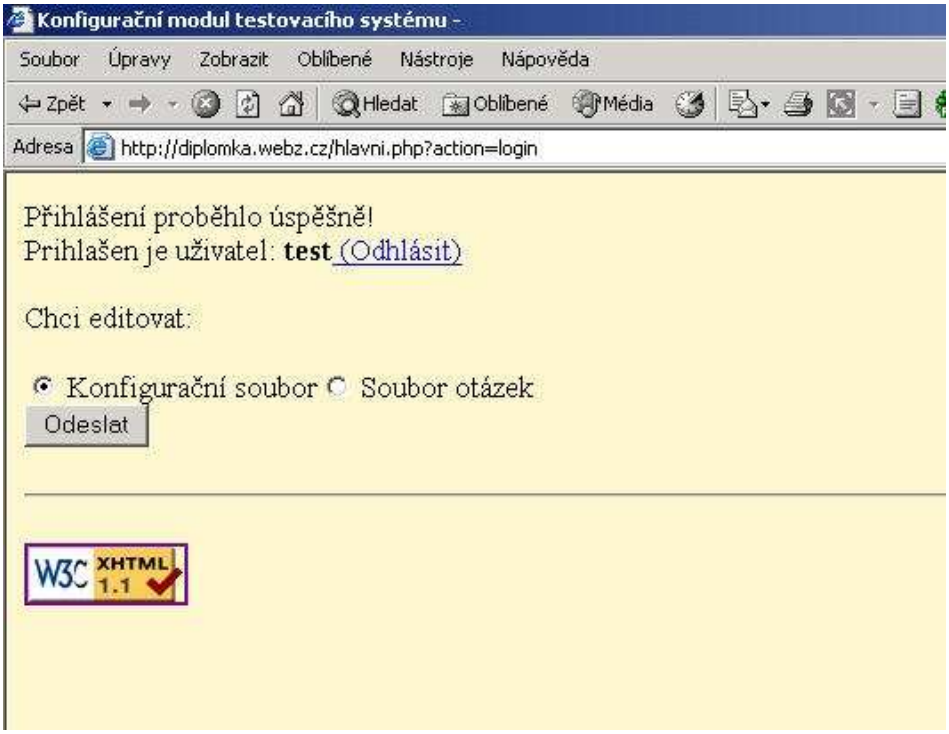
Vstup do vlastního konfiguračního modulu je zabezpečen pomocí uživatelského jména a hesla. Uživatel je vyzván k zapsání svého uživatelského jména a hesla, které mu bylo přiděleno správcem systému, do příslušných formulářů. Pokud některý z údajů nezadá, či pokud údaje nebudou vyplněny správně, skript ohlásí chybu a vyzve uživatele k opětovnému zadání.



Obr. 5.1: Chyba v zadání uživatelského jména nebo hesla

V případě správného vyplnění svých identifikačních údajů, se uživateli objeví první zabezpečená stránka vlastního konfiguračního modulu.

5.2 Úvodní stránka



Obr. 5.2: Úvodní stránka

V tomto dialogu má uživatel možnost vybrat si, zdali chce editovat či vytvořit vlastní konfiguraci testu, či zda chce upravovat jednotlivé otázky a odpovědi testu.

Dalším důležitým zadávaným parametrem je jméno souboru .xml, jehož obsah chceme dále upravovat. Pokud zadáme systému úplně nový název souboru otázek příp. konfigurace, který v kořenovém adresáři neexistuje, program vytvoří nový soubor podle zadaného názvu.

5.3 Konfigurace testu

Tato stránka umožňuje uživateli jednoduchou editaci parametrů potřebných pro konfiguraci vlastního testu, které jsou uloženy v konfiguračním souboru s názvem, který zadáme v předchozím dialogu (např. cfg.xml).

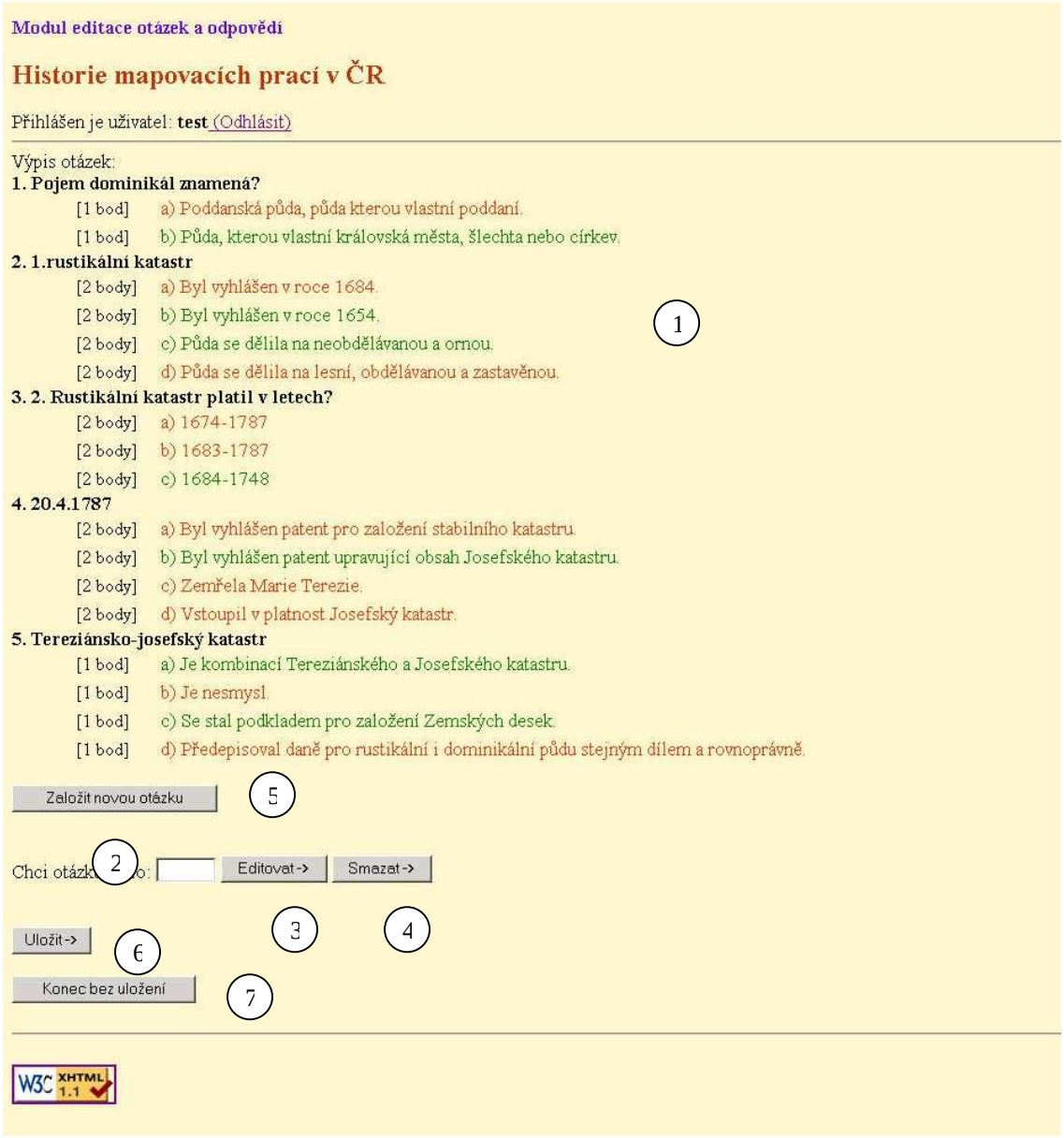
Při spuštění dojde k načtení parametrů ze zadaného souboru, které je možno dále editovat (v případě nového souboru se načtou parametry přednastavené jako výchozí). Před vlastním uložením, které se provádí stiskem tlačítka **Uložit** je kontrolováno vyplnění všech požadovaných polí, příp. je uživatel vyzván k doplnění chybějících údajů. V tomto případě dojde k uložení pod stejným názvem tzn. původní soubor bude přepsán. Pokud budeme chtít nastavené parametry uložit pod jiným názvem, zvolíme

tlačítko **Uložit jako…**. Systém vyvolá dialog, který požaduje po uživateli zadání nového názvu souboru

Pomocí tlačítka **Zrušit** může uživatel opustit konfigurační okno bez uložení vyplněných údajů, přičemž dojde k přesměrování na úvodní stránku.

5.4 Editace otázek

Modul editace otázek a odpovědí je ve své podstatě daleko propracovanější a komplikovanější, ikdyž na první pohled vypadá velice jednoduše. V první fázi se uživateli zobrazí výpis otázek (1) a k nim odpovídajících odpovědí ze souboru formátu XML, jehož jméno bylo skriptu zadáno v předchozím okně. Otázky jsou řazeny a číslovány postupně, jak byly uloženy v datovém souboru. Každá odpověď je barevně odlišena podle správnosti (správně=zeleně ; špatně = červeně) a v hranatých závorkách je zobrazeno i její bodové ohodnocení.



Obr. 5.3: Výpis otázek

Formulář (2) je vstupním formulářem pro zadání čísla otázky, kterou chceme dále upravovat případně smazat. Správné vyplnění tohoto formuláře je kontrolováno skriptem při stisku tlačítka umožňujícího **Editaci** (3) nebo **Smazání** (4) otázek.

Obr. 5.4: Zobrazení dialogů při nesprávném zadání čísla otázek

Kromě výše zmíněných tlačítek jsou na stránce tlačítka **Přidat otázku** (5), **Uložit** (6) a **Konec bez uložení** (7) o jejichž funkčnosti se zmiňuji více.v kapitole 5.6 příp. 5.8.

5.5 Mazání otázek/odpovědí

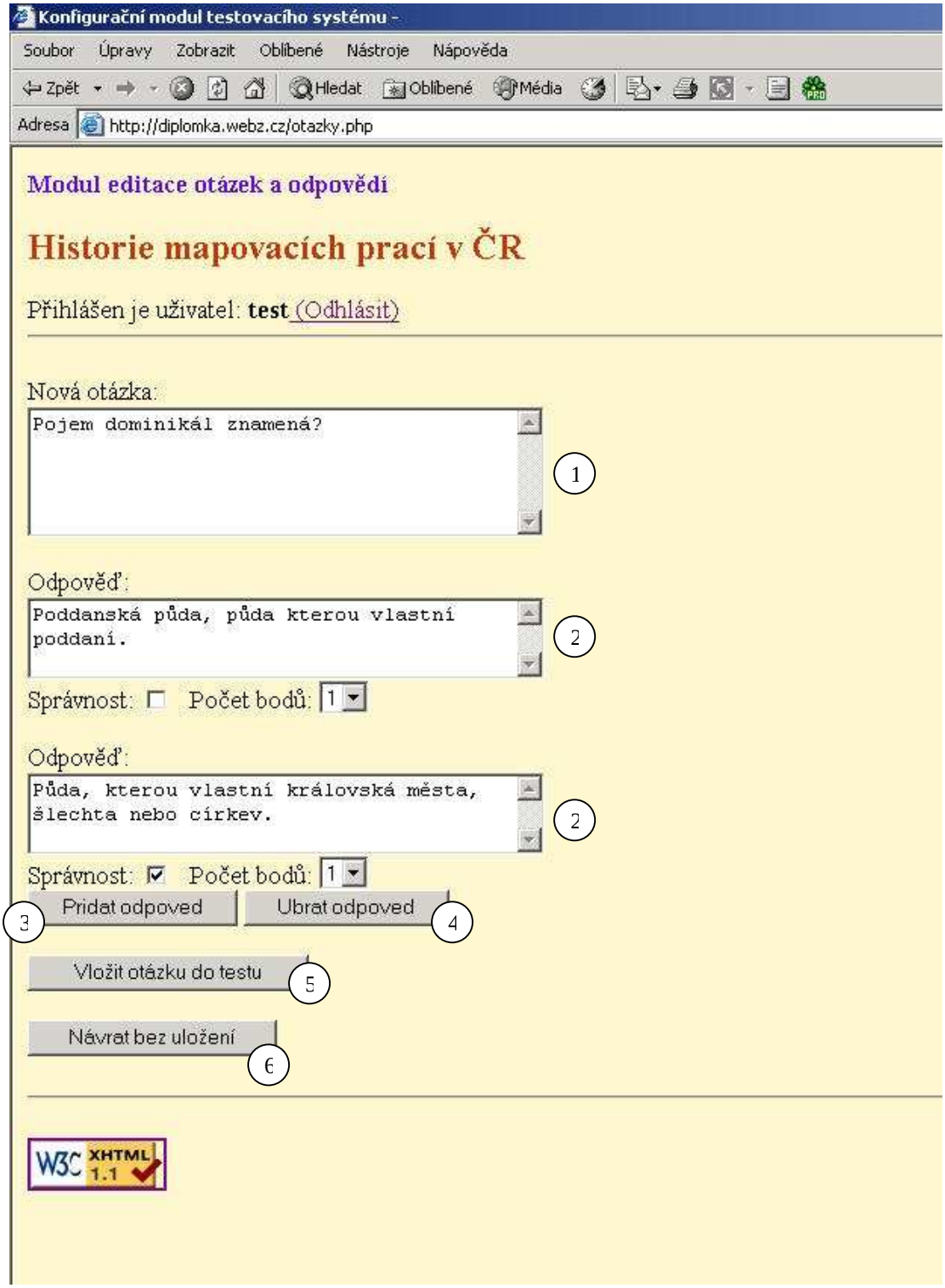
Při zadání čísla otázky a stisku tlačítka Smazat dochází pomocí skriptu k vyvolání potvrzovacího formuláře. Při kladném potvrzení dojde k okamžitému smazání otázky. Při negativní odpovědi dojde ke zrušení příkazu a ke smazání nedojde.

Pozn.: Takto smazanou otázku lze ještě zachránit.[1)]

5.6 Přidávání otázek/odpovědí

Při stisku tlačítka **Přidat otázku** v okně výpisu otázek se v prohlížeči otevře nová stránka, která v první fázi zobrazí jeden formulář pro zadání nové otázky (1). Teprve po zadání otázky, jejíž vyplnění je kontrolováno při stisku tlačítka **Přidej odpověď** (3)se zobrazí další dva formuláře (2) pro zadání odpovědí a jejich správnosti a bodového ohodnocení.

[1)] Pokud neuložíme v okně výpisu otázek (viz. 5.4) celý soubor, smazané otázky se zachovají.



Obr.:5.6: Okno přidávání nových otázek/odpovědí

Pomocí tlačítek **Přidat odpověď** (3) a **Ubrat odpověď** (4) je možné přidávat a ubírat odpovědi s tím, že skript kontroluje, aby byly vždy zobrazeny minimálně dvě odpovědi (logický předpoklad testu). Zároveň s textem odpovědi zadáváme i bodové hodnocení a správnost jednotlivých odpovědí.

Při stisku tlačítka **Vložit otázku do testu** (5) dojde ke kontrole vyplnění všech formulářů a teprve při kladném výsledku dojde k uložení do vnitřní paměti počítače.[1)]

Pomocí tlačítka **Návrat bez uložení** (6) je umožněn návrat na výpis všech otázek bez uložení změn.

5.7 Úprava otázek/odpovědí

Při stisku tlačítka **Editovat otázku** se v okně do jednotlivých formulářů vypíše právě ta jedna otázka i s odpověďmi, kterou jsme si vybrali. Podobně jako při přidávání nových otázek je možno přidávat a ubírat odpovědi (vždy se smaže poslední odpověď).

Tlačítkem **Změnit** uložíme upravovanou otázku do vnitřní paměti programu.

Pomocí tlačítka **Návrat bez uložení** lze úpravu přerušit bez uložení.

5.8 Uložení

Při stisku tlačítka **Uložit test** dojde k definitivnímu přepsání původního souboru s otázkami, v pořadí a znění, které uživatel vidí ve výpisu otázek. Pokud chceme upravený test uložit do jiného souboru, než byl zdrojový, zvolíme tlačítko **Uložit test jako…** Návrat na úvodní stránku bez jakéhokoliv uložení lze provést tlačítkem **Konec bez uložení**.

5.9 Odhlášení

Každý uživatel má možnost se v kterékoliv fázi běhu programu odhlásit pomocí tlačítka **Odhlásit**, které se nalézá napravo od jména přihlášeného uživatele. K automatickému odhlášení dojde také při zavření okna prohlížeče.

[1)] Stejně jako v případě smazání nedojde zatím k zapsání do souboru, toto je provedeno až při závěrečném uložení celého testu.

6. Závěr

Výsledkem této diplomové práce je funkční modul umožňující konfiguraci vstupních souborů testovacího systému. Tento modul je díky dodržení uznávaných standardů W3C plně kompatibilní se všemi internetovými prohlížeči. Zároveň je možný další vývoj celého testovacího systému. Modul byl v návaznosti na celý systém testován v praxi a byla prokázána jeho plná funkčnost. Jediným omezujícím nárokem je potřeba instalace jazyka PHP na straně serveru min. ve verzi 4. Tuto podmínku však server katedry mapování a kartografie splňuje.

Celá diplomová práce je dostupná nejen jako tištěná verze, ale zároveň i digitálně na přiloženém datovém nosiči a formou webové prezentace včetně zdrojových kódů (http://gama.fsv.cvut.cz/~soukup/dip/prazak/index.html).

Tato diplomová práce mi přinesla a rozšířila mnoho nových znalostí v oblasti informatiky a programování. Budu velice potěšen, pokud má práce přispěje ke zkvalitnění výuky studentů na katedře mapování a kartografie příp. dalších katedrách.

7. Výpis funkcí

V této kapitole je vypsán zdrojový kód souboru fce.php, který obsahuje funkce nutné pro vlastní běh konfiguračního modulu. K jednotlivým funkcím je připsán jednoduchý komentář vyjadřující jejich účel. Funkce jsou řazeny abecedně.

define()

- substituce dlouhých nebo často používaných řetězců za jiný. Hodí se pro zjednodušení zápisu.

"BR" = tag
 a zalomení řádku

"NL" = zalomení řádku

"ZAHLAVI" = celá hlavička XHTML dokumentu až po open tag </body>

"ZAPATI" = záhlaví XHTML dokumentu: </body></html>

"VALID" = odkaz a obrázek na validační stránky W3C pro kontrolu správnosti dokumentu XHTML1 http://validator.w3.org/

"FILE_ZAHL" = definuje hlavičku souboru XML, do kterého se zapisuje (kódování a otevírací tagy souboru)

"FILE_ZAP" = definuje zápatí souboru XML, do kterého se zapisuje (uzavírací tagy)

```
define("BR", "<br />\n");

define("NR", "\n");

define("NL", "<hr />");

define("ZAHLAVI", '
<?xml version="1.1" encoding="windows-1250"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.1 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head> <meta http-equiv="Content-type" content="text/html;
charset=windows-1250" />
<link rel="stylesheet" href="test.css" type="text/css" />
```

```
<title>Konfigurační modul testovacího systému</title>
</head>
<body>' );

define("ZAPATI", '</body></html>');

define("VALID",'<hr /><p><a
href="http://validator.w3.org/check?uri=referer">
<img src="http://www.w3.org/Icons/valid-xhtml11" alt="Valid XHTML
1.1!" height="31" width="88" />
</a> </p>');

define("FILE_ZAHL" ,
'<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE test SYSTEM "test5.dtd">
<test>');

define("FILE_ZAP" , '
</test>
');

define("FILE_ZAHL2" ,
'<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE cfg SYSTEM "cfg5.dtd">
<cfg>
');

define("FILE_ZAP2" , '
</cfg>
');

function is_implicit_n($in) {if ($in!="a") $in="n"; return($in);}
function is_implicit_a($in) {if ($in!="n") $in="a"; return($in);}
```

Edit_ot()

zobrazí otázku $cislo z pole $arr_in do formulářů a umožňuje její editaci; vrací nové pole $arr_in

```
function Edit_ot($arr_in,$cislo, $Poc_Odp2)
```

```
{
$otazka = $arr_in[$cislo][0][0];
$Poc_Odp2 = $_SESSION[s_Poc_Odp2];
Print '
<br />
<form action="otazky.php" method="get" onSubmit="return
kontrola(this)" />
<p>
Chci otázku číslo:
<input type="text" name="Edit_Ot" value="" size="5" />
<input type="submit" name="Editovat" value="Editovat ->" />
<input type="submit" name="Del" value="Smazat ->" />
</p>
<input type="hidden" name="Edit" value="true" />
<input type="hidden" name="C" value="'.$Cis.'" />
<input type="hidden" name="q" value="1" />
</form>
';*/
Print '<form action="otazky.php">';
Print 'Otázka číslo '.$cislo.': '.BR;
Print '<textarea name="Otazka_Upr" rows="5"
cols="40">'.$otazka.'</textarea><br />';


if (isset ($Edit2))
{
$Poc_Odp2++;
if ($Poc_Odp < 2) $Poc_Odp=2;
unset ($Edit2);
}

if (isset ($Edit3))
{
$Poc_Odp2 = $Poc_Odp2-1;
if ($Poc_Odp >10) $Poc_Odp=10;
unset ($Edit3);
}

$_SESSION[s_Poc_Odp2] = $Poc_Odp2;
for($j=1;  $j<=$Poc_Odp2; $j++)    //edit odpovedi k dane otazce
{
```

```
$znac = Array(1=>"a)", "b)", "c)", "d)", "e)", "f)", "g)", "h)", "i)",
"j)", "k)", "l)", "m)", "n)");
if ($arr_in[$cislo][$j][1]=="a") $check="checked";
Print BR.'Odpověď '.$znac[$j].': '.BR;
Print '<textarea name="Odp_Upr['.$j.']" rows="3"
cols="40">'.$arr_in[$cislo][$j][0].'</textarea><br />';
Print 'Správnost: '.'<input type="checkbox" name="Odp_Spr['.$j.']"
value="a" '.$check.'>'.'    ';
// bodove hodnoceni
Print 'Body:'.'
<select name="Bod['.$j.']" size="1">';
for ($k=1; $k<=5; $k++)
{
if ($arr_in[$cislo][$j][2]== $k) $select="selected";
Print '<option value="'.$k.'" '.$select.'>'.$k;
unset ($select);
}
Print '</option>

<br />';

Print 'Typ odpovědi:'.'
<select name="Typ_Odp['.$j.']" SIZE="1">';
if ($arr_in[$cislo][$j][2]== "Normal") $select1="selected";
if ($arr_in[$cislo][$j][2]== "URL") $select2="selected";
Print '<option value="Normal" '.$select1.'>Normal';
Print '<option value="URL" '.$select2.'>URL';
unset ($select1);
unset ($select2);
Print '</select>
<br />';
unset ($check);
}
Print '  '.'<input type="submit" value="Zmenit">';
Print '<input type="hidden" name="Cis" value="'.$cislo.'">';
Print '<input type="hidden" name="Change" value="true">';
$_SESSION[s_Poc_Odp] = $Poc_Odp;
Print '</form>';
Print '<form action="otazky.php">';
Print '  '.'<input type="submit" name="Edit2" value="Přidat
odpověď">';
```

```
Print '  '.'<input type="submit" name="Edit3" value="Ubrat
odpověď"><br /><br />';
Print '<input type="hidden" name=Edit value="true">';
Print '<input type="hidden" name="Edit_Ot" value="'.$cislo.'">';
Print '<input type="hidden" name="Editovat" value="X">';
Print '<input type="hidden" name="Poc_Odp2" value="'.$Poc_Odp2++.'">';
Print '<input type="hidden" name="q" value="2">';
Print '</form>';
// navrat bez ulozeni
Print '<form action="otazky.php">';
Print '<input type="submit" value=">> Návrat bez uložení <<">';
Print '<input type="hidden" name="Change" value="false">';
Print '</form>';

Return $otazka;
Return $arr_in;
Return $Poc_Odp2;
 }
```

Nactifile()

nacte obsah souboru $jmeno do proudu

```
function nactifile($jmeno)
{
 $fp = fopen($jmeno,'r');
 $data = fread($fp, filesize($jmeno));
 fclose($fp);
 return ($data);
}
```

Parse2arr()

z datoveho proudu xml souboru preparseruje a udela pole, které vrátí

```
function parse2arr($proud)
{
$parser = xml_parser_create();                    //vytvoreni parseru
xml_parser_set_option($parser, XML_OPTION_SKIP_WHITE, 1);
//preskakovani bilych znaku
xml_parser_set_option($parser,XML_OPTION_CASE_FOLDING,0);
```

```
// nerozlisuje mala velka pismena
//xml_set_character_data_handler($parser,$proud);
xml_parse_into_struct($parser,$proud,$vals,$index);
xml_parser_free($parser);
return ($vals);
}
```

Prepis_cfg()

zjednoduší a přepíše veliké pole s konfiguračními daty z parseru a vybere potřebné hodnoty, kterým přiřadí jednotlivé indexy

```
function prepis_cfg($arr_in)
{
for($i=0;$i<=sizeof($arr_in);$i++)
{
switch ($arr_in[$i][tag]):     /* porovnavani tagu xml */
case "zadani":{
$arr_out[z_ad]=$arr_in[$i][attributes][adresa];
$arr_out[z_ot]=intval($arr_in[$i][attributes][otazek]);
$arr_out[z_ca]=intval($arr_in[$i][attributes][cas]);
$arr_out[z_lo]=is_implicit_n($arr_in[$i][attributes][login]);
} break;
case "michani":{
$arr_out[m_ot]=is_implicit_n($arr_in[$i][attributes][otazek]);
$arr_out[m_od]=is_implicit_n($arr_in[$i][attributes][odpovedi]);
} break;
case "vysledky":{
$arr_out[v_po]=intval($arr_in[$i][attributes][pocitani]);
$arr_out[v_zo]=is_implicit_a($arr_in[$i][attributes][zobrazovat]);
$arr_out[v_uk]=is_implicit_n($arr_in[$i][attributes][ukladat]);
$arr_out[v_ad]=$arr_in[$i][attributes][adresar];
} break;

endswitch;
}

return($arr_out);
}
```

Prepis_test()

zjednoduší a přepíše veliké pole s otázkami z parseru a vybere potřebné hodnoty, kterým přiřadí jednotlivé indexy

```
function prepis_test($arr_in)
{
$ot=1; $odp=1;
for($i=0;$i<=sizeof($arr_in);$i++)
{
switch ($arr_in[$i][tag]):
case "okruh": $arr_out[0][0]=$arr_in[$i][value];
break;
case "otazka":{
switch($arr_in[$i][type]):
case "open": $odp=1;
break;
case "close": $ot++;
break;
endswitch;
}
break;
case "zadani": { if ($arr_in[$i][attributes][typ]=="url"):
$arr_out[$ot][0][0]='<a href="'.$arr_in[$i][value].'" target="new"
class="test_url">'.$arr_in[$i][value].'</a>';
else:  $arr_out[$ot][0][0]=$arr_in[$i][value];
endif;
}
break;
case "odpoved":{ if ($arr_in[$i][attributes][typ]=="url"):
$arr_out[$ot][$odp][0]='<a href="'.$arr_in[$i][value].'" target="new"
class="test_url">'.$arr_in[$i][value].'</a>';
else:  $arr_out[$ot][$odp][0]=$arr_in[$i][value];
endif;
}

$arr_out[$ot][$odp][1]=$arr_in[$i][attributes][spravnost];
$arr_out[$ot][$odp][2]=intval($arr_in[$i][attributes][body]);
```

```
$arr_out[$ot][0][1]=$odp;
$odp++;
break;
endswitch;
}
$arr_out[0][1]=$ot-1;

return ($arr_out);
}
```

Smaz_Ot()

smaže vybranou otázku včetně odpovědí a zkrátí pole $arr_in

```
function Smaz_Ot($arr_in, $cislo)
{
$vymaz = array_splice($arr_in, $cislo, 1);
$arr_in[0][1]--;
return $arr_in;

}
```

Znacky()

vrací písmenný index odpovědi odpovídající pořadovému číslu odpovědi

```
function Znacky()
{
$znac = Array(1=>"a)", "b)", "c)", "d)", "e)", "f)", "g)", "h)", "i)",
"j)", "k)", "l)", "m)", "n)");
return $znac;
}
```

Zobr_ot_all()

zobrazí v okně výpis otázek s odpověďmi, bodovým hodnocením a barevným odlišením správnosti.

```
function Zobr_ot_all($arr_in)
{
$prvku = $arr_in[0][1];
```

```
for($i=1; $i<=$prvku; $i++)
{
if($arr_in[$i]!=""):
Print '<font class="test_ota">'.$i.'.
'.$arr_in[$i][0][0].'</font><BR>';   /* vypisuji otazky */
for($j=1; $j<=$arr_in[$i][0][1]; $j++)
{
$znac = Array(1=>"a)", "b)", "c)", "d)", "e)", "f)", "g)", "h)", "i)",
"j)", "k)", "l)", "m)", "n)");    // jednotlive odpovedi razeny pomoci
pismen
Print '<table>';
$text_bod = ' body';
if ($arr_in[$i][$j][2]==1):
$text_bod =' bod';
elseif ($arr_in[$i][$j][2]>=5):
$text_bod =' bodů';
endif;
// vypisuji otazky s bodovym hodnocenim a barevne odlisenou spravnosti
switch($arr_in[$i][$j][1])
{
case "a":
Print '<tr><td width="50"></td><td width="70"><font
class=test_bod>['.$arr_in[$i][$j][2].$text_bod.'] </td><td><font
class="cfg_odp_a">'.$znac[$j].' '.$arr_in[$i][$j][0].'</font><br
/></td></tr>';
break;
case "n":
Print '<tr><td width="50"></td><td width="70"><font
class=test_bod>['.$arr_in[$i][$j][2].$text_bod.'] </td><td><font
class="cfg_odp_n">'.$znac[$j].' '.$arr_in[$i][$j][0].'</font><br
/></td></tr>';
break;
default:
}
Print '</table>';
}
else:
endif;
}
}
```

Zobr_Zahl()

definuje záhlaví modulu obsahující login a jméno testu

```
function Zobr_Zahl($Login,$Test_Name)
{
Print '<h2 class="test">Modul editace otázek a odpovědí</h2>'.NR;
Print '<h1 class="test">'.$Test_Name.'</h1>'.NR.NR;
Print 'Přihlášen je uživatel:  <b>'.$Login.'</b>';
Print '<a href="'.$PHP_SELF.'?action=logout">
(Odhlásit)</a>'.NR.NL.NR.NR;
}
```

8. Seznam obrázků

9. Použitá literatura

1) ELLA.cz, společnost pro elektronické vzdělávání [on-line], http://www.ella.cz
2) Telnarová, Z. E-Learning. Ostrava: OU, 2003 ISBN 80-7042-874-0
3) Molnár, Z. Efektivnost informačních systémů. Druhé, rozšířené vydání. Praha: Grada Publishing, 2001 ISBN 80-247-0087-5
4) Mužík, J., Fiala, B. Vzdělávání dospělých v informační společnosti, http://econ.cz/konference/fiala.doc
5) Kosek, J.: HTML – Tvorba dokonalých WWW stránek – podrobný průvodce. Praha: Grada Publishing 1998. ISBN 80-7169-608-0 http://www.kosek.cz/html/
6) Netscape Communication Corp.: http://www.netscape.com
7) Kosek. J.: PHP – Tvorba interaktivních internetových aplikací. Praha: Grada Publishing, 1999 ISBN 80-7169-373-1
8) Schurman M. Eric, Pardi J. William: Dynamické HTML v akci. Praha: Computer Press, 2000 ISBN 80-7226-401-X
9) Microsoft Corporation Ltd., http://www.microsoft.com, http://www.microsoft.cz
10) The Apache Software Foundation, http://www.apache.org , http://httpd.apache.org
11) The PHP Group, http://www.php.net , http://www.php.cz
12) Webové stránky Ministerstva školství, mládeže a tělovýchovy ČR, http://www.msmt.cz
13) Webové stránky Národního centra distančního vzdělávání, http://www.csvs.cz/struktura/ncdiv/ncdiv.html
14) World Wide Web Consortium, http://www.w3c.org
15) Kosek. J.: PHP – XML pro každého, 1.vydání,. Praha: Grada Publishing, 2000
16) Castagneto, a kolektiv, Programujeme PHP profesionálně, 2.vydání, Praha Computer Press, 2002
17) Štrimpfl M.: Active Server Pages pro úplné začátečníky, 1.vydání, Praha, Computer Press, 2000 ISBN 80-7226-347-1

18) Morkes D.: Java Script – Praktické příklady, Praha, Grada Publishing, 2002 ISBN 80-247-0258-4

19) Bradley N.: XML – kompletní průvodce, Praha, Grada Publishing, 2000 ISBN 80-716-9949-7

Příloha A - Obsah přiloženého CD

- Diplomová práce
  - tento adresář obsahuje text diplomové práce ve dvou formátech
    - ❖ soubor *diplomka.pdf*
    - ❖ soubor *diplomka.ps*
- Instalace
  - tento adresář obsahuje vlastní zdrojové kódy a další nezbytné soubory konfiguračního modulu
- Programy
  - tento adresář obsahuje programy použité pro testování, příp. ladění vlastního programu
- Testy
  - tento adresář obsahuje několik vzorových testů z předmětu Informatika 1

Podrobný popis jednotlivých adresářů a souborů a jejich použití je uložen v souboru ctime.txt, který je uložen v kořenovém adresáři CD.