

# **Využití moderních nástrojů pro modifikaci informačního systému CAAC na WWW stránkách**

Lukáš Veselý

---

Bakalářská práce  
2006



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav aplikované informatiky  
akademický rok: 2005/2006

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Lukáš VESELÝ**  
Studijní program: **B 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**

Téma práce: **Využití moderních nástrojů pro modifikaci  
informačního systému CAAC na WWW stránkách**

### Zásady pro vypracování:

1. Zpracujte literární rešerši pro oblast zabývající se tvorbou informačních systémů na Internetu.
2. Navrhněte postup pro modifikaci informačního systému CAAC na WWW stránkách, při současném dodržení již existujících pravidel uvnitř tohoto informačního systému.
3. Pro zvolený postup provedte úpravu informačního systému CAAC na WWW stránkách.
4. Vytvořte dokumentaci k upravené verzi informačního systému CAAC.
5. Zprovozněte modifikovaný informační systém CAAC na WWW stránkách Internetu.

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

[1] MySQL[online]. c1995 [cit. 2006-01-18]. Dostupný z WWW:

<http://dev.mysql.com>;

[2] PHP[online]. c2001 [cit. 2006-01-18]. Dostupný z WWW: <http://php.net>;

[3] KOSEK, J. Téměř vše o WWW[online]. c1999, 2005/11/07 [cit. 2006-01-18].

Dostupný z WWW: <http://kosek.cz>;

[4] WELLING, L., THOMSON, L.: PHP a MySQL - rozvoj webových aplikací. Praha: SoftPress, 2002.

Vedoucí bakalářské práce:

**Ing. Pavel Navrátil, Ph.D.**

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

**14. února 2006**

Termín odevzdání bakalářské práce:

**16. června 2006**

Ve Zlíně dne 14. února 2006

prof. Ing. Vladimír Vašek, CSc.  
*pověřený děkan*



doc. Ing. Ivan Zelinka, Ph.D.  
*ředitel ústavu*

## **ABSTRAKT**

Informační systém CAAC (Computer Aided Automatic Control – počítačová podpora automatického řízení) využívá internetové stránky k podpoře výuky předmětů zaměřených na automatické řízení. Podstatou mé práce bylo zpracování literární rešerše pro oblast zabývající se tvorbou informačních systémů na Internetu a dále pak modifikace již existujícího informačního systému CAAC, vytvořeného v jazyce HTML. Ta spočívala ve využití moderních nástrojů – skriptovacího jazyka PHP, kaskádových stylů CSS, Javascriptu a databázového systému MySQL při úpravě systému. Přitom bylo důležité dodržet již existující pravidla, hlavně strukturu systému, strukturu adresářů a souborů a celkové grafické provedení. Další částí pak bylo vytvoření dokumentace k modifikované verzi.

Klíčová slova: CAAC, HTML, CSS, Javascript, PHP, MySQL.

## **ABSTRACT**

CAAC Information System (Computer Aided Automatic Control) uses the Internet pages as a support of subjects orient to the automatic control. The bases of my work were processing the literary material of the Information Systems on the Internet area and modification the previous version of CAAC Information System, which was made in HTML language. These modern tools – scripting language PHP, cascading style sheets CSS, Javascript and database system MySQL, were used in this work. The information system structure, the directory structure and the graphic appearance have to be kept. Made of the documentation to modified version was the next part of my work.

Keywords: CAAC, HTML, CSS, Javascript, PHP, MySQL.

## Poděkování

Chtěl bych vyjádřit poděkování vedoucímu mé bakalářské práce Ing. Pavlovi Navrátilovi, Ph.D. za jeho podnětné připomínky, náměty k realizaci a neocenitelnou pomoc během tvorby bakalářské práce.

Prohlašuji, že na celé bakalářské práci jsem pracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně 12. 6. 2006

.....

Lukáš Veselý

# OBSAH

<b>ÚVOD</b> .....	<b>8</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 INFORMAČNÍ SYSTÉMY NA INTERNETU</b> .....	<b>11</b>
1.1 MOODLE.....	11
1.1.1 Základní vlastnosti Moodlu.....	11
1.2 REDAKČNÍ A INFORMAČNÍ SYSTÉM PHPRS.....	13
1.2.1 Základní vlastnosti systému phpRS .....	13
<b>2 INFORMAČNÍ SYSTÉM CAAC</b> .....	<b>15</b>
2.1 STRUKTURA INFORMAČNÍHO SYSTÉMU CAAC .....	15
2.2 STRUKTURA ADRESÁŘŮ A SOUBORŮ.....	16
2.3 OSNOVY WWW STRÁNEK.....	16
2.4 INFORMAČNÍ SEZNAMY O STAVU ŘEŠENÍ JEDNOTLIVÝCH ČÁSTÍ INFORMAČNÍHO SYSTÉMU CAAC .....	16
Seznam 1. typu:.....	17
Seznam 2. typu:.....	18
<b>3 MODIFIKACE INFORMAČNÍHO SYSTÉMU CAAC</b> .....	<b>19</b>
3.1 POUŽITÉ PROGRAMOVACÍ JAZYKY A TECHNOLOGIE.....	19
3.1.1 Databázový systém MySQL a jazyk SQL .....	19
3.1.2 PHP .....	22
3.1.3 HTML.....	25
3.1.4 CSS – kaskádové styly .....	26
3.1.5 Javascript.....	27
3.1.6 Instalace programu Easy-PHP (Apache, PHP, MySQL).....	27
3.2 POSTUP PŘI MODIFIKACI INFORMAČNÍHO SYSTÉMU CAAC .....	28
3.2.1 Vytvoření databáze MySQL.....	28
3.2.2 Modifikace informačního systému CAAC.....	28
3.2.3 Jedny z možných návrhů na vylepšení informačního systému CAAC.....	29
<b>II PRAKTICKÁ ČÁST</b> .....	<b>30</b>
<b>4 ÚPRAVA INFORMAČNÍHO SYSTÉMU CAAC</b> .....	<b>31</b>
4.1 NÁVRH DATABÁZE .....	31
4.2 ZMĚNA ÚVODNÍ STRÁNKY .....	31
4.3 VYTVOŘENÍ SKRIPTU NA PŘEVOD HTML SOUBORŮ DO DATABÁZE .....	32
4.4 ČTENÍ DAT Z DATABÁZE .....	32
4.4.1 Zobrazení subsystémů.....	33
4.4.2 Zobrazení základních submodulů.....	34
4.4.3 Další skripty – tisk, dotaz na email a vyhledávání .....	34
4.4.4 Diskuze.....	36
<b>5 ADMINISTRACE INFORMAČNÍHO SYSTÉMU CAAC</b> .....	<b>37</b>

5.1.1	Přihlášení k administraci .....	37
5.1.2	Hlavní strana administrace .....	37
5.1.3	Úprava subsystémů.....	37
5.1.4	Úprava základních submodulů .....	38
5.1.5	Úprava submodulů knihovny hotových programů .....	39
5.1.6	Úprava stránky literatury .....	39
5.1.7	Správa diskuze .....	40
5.1.8	Ostatní nastavení .....	40
5.1.9	Odhlášení.....	40
<b>6</b>	<b>MODIFIKOVANÝ INFORMAČNÍ SYSTÉM CAAC .....</b>	<b>41</b>
	<b>ZÁVĚR .....</b>	<b>44</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>45</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>46</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>47</b>
	<b>SEZNAM TABULEK.....</b>	<b>48</b>

## ÚVOD

Informační systém CAAC (Computer Aided Automatic Control) by měl sloužit jako počítačová podpora k získávání informací v oblasti automatického řízení a pomáhat studentům, případně i dalším uživatelům při zpracování úloh a projektů z oblasti automatického řízení. Využívá přitom internetové služby WWW stránek. Jedná se svým způsobem o určitou formu eLearningu, který je poslední dobou velmi rozšiřován.

Původní verze informačního systému CAAC (před úpravou) využívala pouze klasických postupů, a proto byla navržena její úprava, aby bylo možno využít nových vlastností moderních metod. Je tím myšleno například převedení celé stávající struktury a obsahu do databáze MySQL z důvodu pozdějšího využití při vyhledávání na stránkách, ukládání příspěvků z diskuze a celkového zjednodušení při vytváření nového obsahu stránek (subsystémů, základních submodulů). Vytváření nového obsahu a celá správa stránek je přitom umožněna přes administrátorské prostředí, ve kterém je dále možno také spravovat diskuzi. Při modifikaci stávajícího informačního systému CAAC jsem využil získané znalosti z předmětů „Tvorba WWW stránek“ a „Databázové systémy“. Dále byly také využity i informace získané z jiných zdrojů.

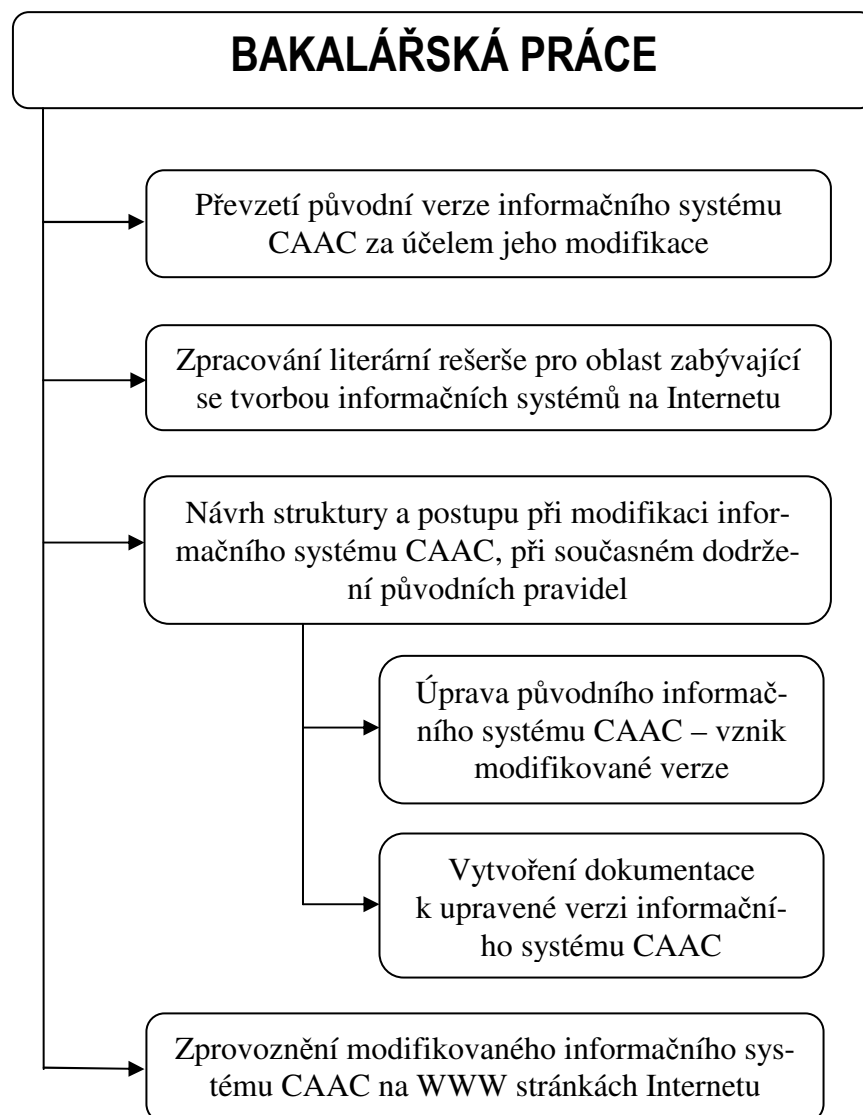
V praktické části jsem využíval při psaní zdrojového kódu hlavně skriptovacího jazyka PHP, HTML jazyka, kaskádových stylů CSS, Javascriptu a jazyka SQL. Při návrhu databáze jsem využil již dříve zmíněné znalosti z předmětu „Databázové systémy“ a použil jsem přitom databázový systém MySQL.

Mým dalším úkolem v praktické části mé bakalářské práce bylo tuto upravenou verzi informačního systému CAAC zprovoznit na WWW stránkách Internetu. Při vytváření a ladění této upravené verze jsem využíval lokální server (localhost). Pro instalaci lokálního serveru byl použit volně šiřitelný program Easy-PHP, který obsahuje vše potřebné pro provozování tohoto lokálního serveru (obsahuje server Apache, podporu pro PHP a MySQL).

Součástí této práce bylo také mimo zpracování dokumentace k nové verzi informačního systému CAAC, také zpracovat literární rešerši pro oblast zabývající se tvorbou informačních systémů na Internetu. V této části jsem popsal informační systém Moodle – softwarový balíček pro tvorbu výukových systémů a elektronických kurzů na internetu, který využívá naše fakulta jako podporu při výuce některých předmětů. Dále jsem popsal redakční a informační systém phpRS.



## Struktura bakalářské práce:



## **I. TEORETICKÁ ČÁST**

## 1 INFORMAČNÍ SYSTÉMY NA INTERNETU

Pro popis informačních systémů na Internetu jsem zvolil systém Moodle, který naše univerzita využívá jako podporu při výuce některých předmětů. Dále jsem popsal redakční a informační systém phpRS.

### 1.1 Moodle

Moodle je softwarový balíček pro tvorbu výukových systémů a elektronických kurzů na internetu. Jedná se o neustále se vyvíjející projekt. Moodle je poskytován zdarma jako Open Source software (spadající pod obecnou veřejnou licenci GNU). To v zásadě znamená, že je chráněn autorskými právy, ale poskytuje přitom uživatelům značnou svobodu. Moodle můžete kopírovat, používat i upravovat, pokud souhlasíte s tím, že budete tento zdroj poskytovat ostatním, nebudete měnit ani odstraňovat původní údaje o licencích a autorských právech a uplatníte stejné licenční podmínky i u jakýchkoliv odvozených produktů.

Moodle lze použít na jakémkoliv počítači s fungujícím PHP, podporuje řadu typů databází (především MySQL).

Slovo Moodle znamená Modular Object-Oriented Dynamic Learning Environment (Modulární objektově orientované dynamické prostředí pro výuku). V tomto smyslu se vztahuje jak k samotnému zrodu Moodlu, tak k přístupu studenta či učitele k výuce v on-line kurzech. [7]

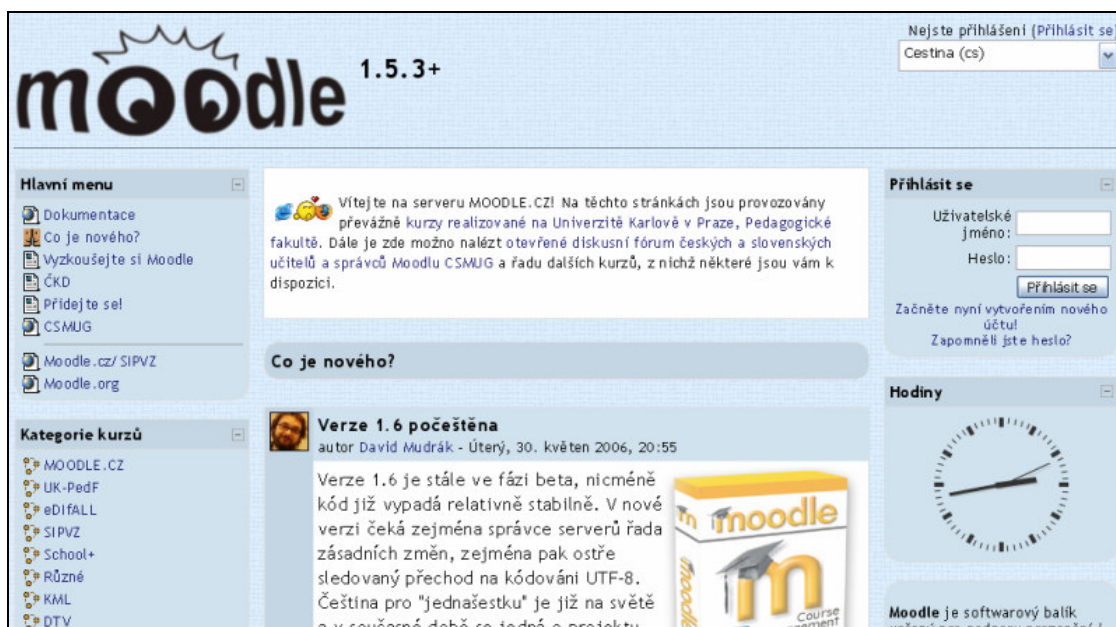
#### 1.1.1 Základní vlastnosti Moodlu

Moodle je produkt, který je stále v pohybu a vyvíjí se. Zde je seznam některých jeho vlastností a funkcí, které podporuje:

- je vhodný pro plně distanční internetovou výuku i jako doplněk kontaktní výuky,
- jednoduché, efektivní, široce kompatibilní, technicky nenáročné a intuitivní uživatelské rozhraní,
- snadná instalace na téměř všechny platformy, které podporují PHP. Vyžaduje pouze jednu databázi (a tu může sdílet),
- plně nezávislý na konkrétní databázi – podporuje všechny hlavní typy databází (kromě úvodního vytvoření tabulek),

- seznam kurzů nabízí popis každého kurzu i informaci, zda do něj mají přístup návštěvníci,
- kurzy lze třídit do kategorií, kategorie lze prohledávat – každý server s Moodle může podporovat tisíce kurzů,
- velký důraz na zabezpečení: data ze všech formulářů jsou kontrolována, cookies jsou šifrovány,
- většinu oblastí pro vkládání textu (zdroje, příspěvky do fór, záznamy do deníku atd.) lze editovat pomocí vestavěného WYSIWYG editoru HTML. [7]

### Vzhled Moodle:



Obr. 1 Vzhled informačního systému Moodle

Důležitou součástí projektu Moodle je webová stránka <http://www.moodle.org>. Je zdrojem informací a místem pro diskusi a spolupráci uživatelů, mezi něž patří správci systémů, pedagogové, vědci, a samozřejmě vývojáři. Tato stránka se, stejně jako Moodle sám, neustále vyvíjí, aby vyhovovala potřebám uživatelů, a stejně jako Moodle bude vždy zdarma.

## 1.2 Redakční a informační systém phpRS

Redakční systém představuje aplikaci, která spravuje informace různého charakteru a zároveň se stará i o jejich efektivní využití a zobrazení na některém z předpřipravených výstupů. Hlavním výstupem webového redakčního systému phpRS jsou dynamicky generované HTML stránky, ke kterým čtenáři (uživatelé) přistupují prostřednictvím internetového prohlížeče.

phpRS systém podporuje také řadu čistě čtenářsky orientovaných služeb, mezi které například patří možnost vytištění speciálně upravené verze článku, známkování kvality článku, stránkování hlavní stránky nebo možnost odeslání informačního e-mailu na zadanou adresu s hypertextovým odkazem na vybraný článek. [6]

Redakční a informační systém phpRS je chráněn GNU GPL licencí a ta je laicky řečeno velmi "mírná" a přiznává uživatelům velké množství práv. Např. každý uživatel má právo si tento redakční systém sám před vlastním použitím jakkoliv upravit. Na druhou stranu i GNU GPL licence má svá omezení, které musí každý uživatel, chce-li tento software používat legálně – bezpodmínečně dodržet. Například musí být jasně a zřetelně vidět, že vámi používaný redakční systém je právě redakční systém phpRS. [6]

### 1.2.1 Základní vlastnosti systému phpRS

- inteligentní správu uživatelů, která umožňuje rozlišovat mezi administrátorem phpRS systému, redaktory a autory článků,
- komplexní článkový management včetně možnosti zadání data vydání a stažení článku a komentářového subsystému, jenž umožňuje čtenářům reagovat na vydané články,
- propracovaný vzhledový generátor (tzv. layout engine), celkový vzhled je vytvářen prostřednictvím šablon, které lze různě kombinovat,
- download management, který umožňuje zpřístupnění souborů na vašem informačním webu a sledování statistiky jejich "stahování" (existuje možnost tématického dělení)
- anketní subsystém (obsahuje interní ochrany, které se snaží zabránit vícenásobnému hlasování jedné osoby, uzavřené anekty lze uzamknout proti dalšímu hlasování),

- rozsáhlý statistický modul, který umožňuje sledovat celkovou návštěvnost webu a čtenost jednotlivých článků,
  - speciální "Novinkový subsystem", který umožňuje velmi rychle a efektivně publikovat krátké informační zprávy,
  - komplexní vyhledávací subsystem (kromě standardních funkcí umožňuje tento subsystem vyhledávat i v databázi "klíčových slov", která umožňuje značně zefektivnit výsledek vyhledávání,
  - speciální plug-in subsystem, jenž umožňuje velice jednoduše rozšířit základní verzi phpRS o další nové funkce,
  - vlastní interní zálohovací subsystem umožňuje zálohovat celou databázi a následně ji prostřednictvím speciálního importního subsystemu v případě potřeby obnovit.
- [6]

### Základní vzhled redakčního systému phpRS:

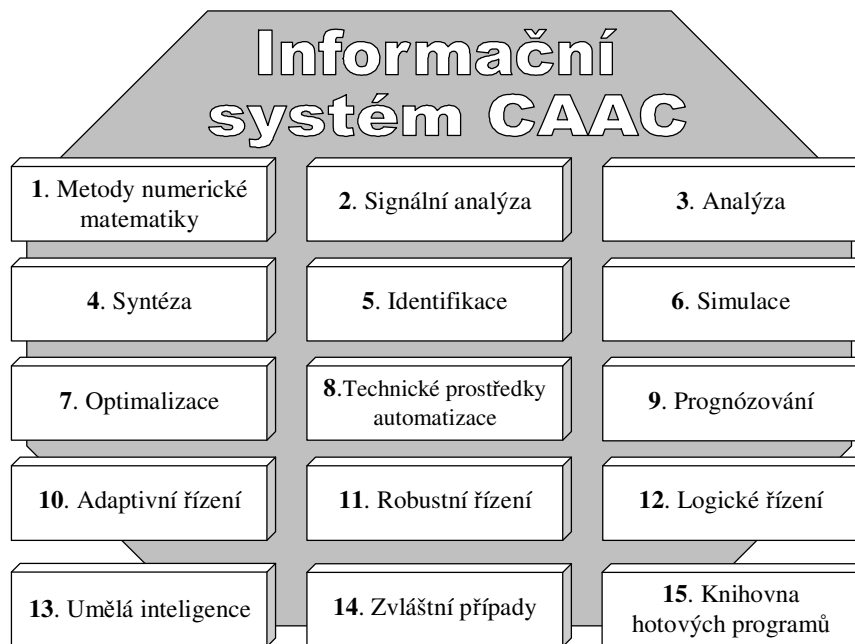


Obr. 2. Vzhled redakčního systému phpRS

Domovskou stránkou redakčního a informačního systému phpRS je <http://www.phprs.cz>.

## 2 INFORMAČNÍ SYSTÉM CAAC

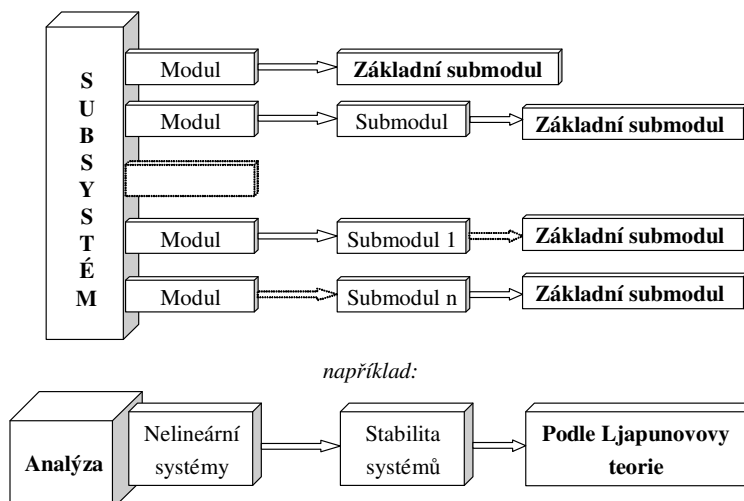
Informační systém CAAC (Computer Aided Automatic Control) je synonymní s termínem CADCS (Computer Aided Design in Control System), který je více rozšířený. Informační systém CAAC představuje průběžně tvořený otevřený systém dílčích tematických okruhů, tzv. subsystémů pro počítačovou podporu teorie automatického řízení. Informační systém CAAC je prozatím rozložen do 15-ti subsystémů, které zahrnují teorii automatického řízení.



Obr. 3. Výčet názvů tematických okruhů informačního systému CAAC

### 2.1 Struktura informačního systému CAAC

Ve všech subsystémech informačního systému CAAC je formulována tzv. problémová specifikace. Každý subsystém je rozčleněn na moduly, které se dále člení na submoduly až do základních submodulů. Základní submodul je nejnižší úroveň hierarchického uspořádání informačního systému CAAC, který řeší konkrétní problém (např. stabilita systému podle Ljapunovy teorie) daného subsystému (např. Analýza) a měl by obsahovat odborný popis, ukázkový příklad, výpočetní program a použitou případně doporučenou literaturu. Výjimku tvoří pouze subsystém „Knihovna hotových programů“, kde základní submodul by měl obsahovat výpočetní program vytvořený pro vybrané části konkrétního subsystému nebo subsystémů a dále pak popis tohoto programu.



Obr. 4. Struktura informačního systému CAAC

## 2.2 Struktura adresářů a souborů

Pro správné a bezpečné upravení nebo nahrazení některého souboru informačního systému CAAC je nutno, aby bylo dodrženo určité umístění a názvy adresářů a souborů. Proto byly vytvořeny 3 typy struktur adresářů a navrženo umístění souborů v jednotlivých adresářích. Tyto 3 typy pokrývají celou problematiku týkající se struktur adresářů a umístění souborů v informačním systému CAAC na WWW stránkách.

## 2.3 Osnovy WWW stránek

Každá osnova WWW stránky obsahuje určitou část struktury informačního systému CAAC, kde každý subsystém je rozčleněn na moduly, které se dále člení na submoduly až do základních submodulů. Byly navrženy 3 osnovy WWW stránek, a to osnova WWW stránky subsystému, osnova informační WWW stránky subsystému „Knihovna hotových programů“ a osnova WWW stránky základního submodulu. Každá z těchto osnov má pevně navrženu vnitřní strukturu a to použitím tagů <h1> až <h6> a nedefinováním vlastních CSS stylů.

## 2.4 Informační seznamy o stavu řešení jednotlivých částí informačního systému CAAC

Pro získání přehlednějších informací o stavu řešení jednotlivých částí informačního systému CAAC byly vytvořeny dvě základní struktury seznamů. Pomocí těchto seznamů by mělo být jednodušší zjišťovat informace o skutečných stavech řešení daných problémů a na základě těchto získaných informací pak provádět zásahy do informačního systému CAAC.



První druh seznamu bude obsahovat informace o základních submodulech daného subsystému. Tento seznam bude používán pro subsystémy 1–14. Pro subsystém 15, tzn. subsystém „Knihovna hotových programů“ je vytvořen druhý typ seznamu, neboť u tohoto subsystému není potřeba tolik informací.

### Seznam 1. typu:

Tento typ seznamu, který se zabývá subsystémy 1-14 bude obsahovat:

- název základního modulu (*základní submodul*)
- název HTML souboru (*HTML soubor*)
- stupeň naplnění (*naplněno*)
- datum zápisu stránky, autor stránky

Pokud daný základní submodul obsahuje i odkaz na skriptový program, pak v seznamu budou naplněny tyto další body:

- název HTML souboru se skriptovým programem (*HTML soubor (info + program)*)
- datum zápisu stránky, autor programu, autor stránky

Položka „stupeň naplnění“ je párová a značí do jaké míry je daný základní submodul naplněn. První část položky se týká textu, který popisuje daný problém a může nabývat těchto 3 hodnot (1, X, -). Druhá část položky se týká ukázkového příkladu a může nabývat těchto 3 hodnot (2, X, -). Jednotlivé uvedené hodnoty znamenají:

- 1 > základní submodul obsahuje text, který popisuje daný problém
- 2 > základní submodul obsahuje ukázkový příklad
- X > základní submodul nebude obsahovat danou část
- > základní submodul zatím neobsahuje danou část

Položka „název HTML souboru se skriptovým programem“ značí, zda daný základní submodul obsahuje nebo neobsahuje program ve skriptu. Tato položka může nabývat těchto hodnot (konkrétní název HTML souboru s programem, X, -), kde

název HTML souboru s programem > základní submodul obsahuje program ve skriptu

X > základní submodul nebude obsahovat program ve skriptu

– > základní submodul zatím neobsahuje program ve skriptu

Tab. 1. Příklad seznamu typu 1

	základní submodul	HTML soubor	naplněno	datum	autor stránky
1	Spojité verze Ziegler Nicholsovi metody kritického zesílení regulátoru	s_ziegler.htm	1,2	5.9.2001	Navrátil Pavel

HTML soubor (info+program)	datum	autor programu	autor stránky
s_ziegler_pj.htm	5.9.2001	Michael Hyža	Navrátil Pavel

**Seznam 2. typu:**

Tento typ seznamu, který se zabývá subsystémem 15, bude obsahovat:

- název subsystému, kterým se zabývá daný program (*subsystém*)
- název řešeného problému (*řešený problém*)
- název informačního HTML souboru (*info HTML soubor*)
- datum zápisu stránky, autor stránky
- název vytvořeného programu (*program*)
- datum vytvoření programu, autor programu

Položka „název vytvořeného programu“ značí, zda daný program je součástí informačního systému CAAC nebo je součástí jiných WWW stránek. Tato položka může nabývat těchto hodnot (jméno souboru daného programu, odkaz na daný program), kde

jméno souboru daného program > program je přímou součástí informačního systému CAAC

odkaz na daný program > odkaz na WWW stránky, které daný program obsahují, ale nejsou součástí WWW stránek informačního systému CAAC

Tab. 2. Příklad seznamu typu 2

	subsystém	řešený problém	info HTML soubor	datum	autor stránky
1	Analýza-Syntéza	Analýza, syntéza jednorozměrového regulačního obvodu	anasyn_p1.htm	1.9.2001	Navrátil Pavel
2	Adaptivní řízení	Adaptivní mnohorozměrové řízení pomocí delta modelů	adaptiv_p1.htm	18.2.2003	Navrátil Pavel

program	datum	autor programu
anasyn_p1.zip	1.6.2000	Navrátil Pavel
adaptiv_p1.zip	6.6.2001	Navrátil Petr

**Poznámka:**

V předcházející tabulce (Tab. 2.), je uvedena ukázka informačního seznamu zabývajícím se subsystémem 15, tj. subsystémem „Knihovna hotových programů“, konkrétně je zde uvedena ukázka zápisu informačního seznamu pro vytvořený multiokruhový program (pozice 1) a jednookruhový program (pozice 2).

Pokud bude v české verzi informačního systému CAAC v subsystému „Knihovna hotových programů“ zařazen program v anglickém jazyce, bude v položce „program“, a to buď za názvem vytvořeného programu nebo odkazem na WWW stránky, které daný program obsahují, ale nejsou součástí WWW stránek informačního systému CAAC, uvedena v závorce zkratka ENG, např. robriez\_p1.zip (ENG) nebo http://www.utb.cz/stctool (ENG).

### 3 MODIFIKACE INFORMAČNÍHO SYSTÉMU CAAC

Stávající informační systém CAAC byl vytvořen pomocí HTML jazyka, modifikace spočívala ve využití skriptovacího jazyka PHP společně s databázovým systémem MySQL. Ke grafickým úpravám a rozvržení jsem využil kaskádových stylů CSS.

#### 3.1 Použité programovací jazyky a technologie

##### 3.1.1 Databázový systém MySQL a jazyk SQL

Vznikl z výzkumného projektu o relačních databázích vedeného v 70. letech firmou IBM. Byl přijat jako oficiální standard v roce 1986 (označován jako SQL86). Opravená verze je z roku 1992 a je označována jako SQL92. Zkratka SQL značí *Structured Query Language*. Ten je v oblasti relačních databází standardem dodnes. Jazyk v sobě zahrnuje nástroje pro tvorbu databází (tabulek) a dále nástroje na manipulaci s daty (vkládání dat, aktualizace, mazání a vyhledávání informací).

Pro řešení problémů různých syntaxi SQL (používaných před zavedením standardu), Microsoft formalizoval CLI – *Common Language Interface* pro pracovní stanice a oznámil, že jeho produkty by měli používat toto rozhraní – nazývá ho ODBC – *Open Database Connectivity*.

SQL patří do kategorie tzv. deklarativních programovacích jazyků – kód jazyka SQL nepíšeme v žádném samostatném programu, ale vkládáme jej do jiného programovacího jazyka, který je již procedurální. [1]

##### **SQL se skládá z několika částí:**

- 1) jazyk DDL – Data Definition Language – jedná se o jazyk pro vytváření databázových schémat a katalogů.
- 2) jazyk SDL – Storage Definition Language – definuje způsob ukládání tabulek.
- 3) jazyk VDL – View Definition Language, určující vytváření pohledů (pohled si lze představit jako virtuální tabulku složenou z různých jiných tabulek – pro návrháře a správce).
- 4) jazyk DML – Data Manipulation Language – obsahuje základní příkazy INSERT, UPDATE, DELETE a nejpoužívanější příkaz SELECT. S jazykem DML pracují nejvíce koncoví uživatelé a programátoři aplikací.

**Datová struktura MySQL:**

Data jsou na disku organizována ve formě souborů. Pro každou tabulku jsou vytvořeny tři soubory a to: \*.frm, \*.myd a \*.myi. Soubor \*.frm je formátovací soubor a obsahuje datovou strukturu. Soubor \*.myd obsahuje data a soubor \*.myi obsahuje informace o klíčích a jiných vnitřních křížových odkazech. Tyto soubory se nacházejí většinou v adresáři /var/lib/mysql/Jmeno\_Databaze.

**Návrh tabulek:**

Základem každé databáze jsou tabulky, které nám popisují nějakou entitu. Tabulka se skládá ze sloupců (polí), kterým říkáme také atributy a volíme takové vlastnosti, které nás o dané entitě zajímají. Podle pravidel používaných v MySQL se k pojmenování sloupců používají alfanumerické znaky a podtržítka.

**Datové typy sloupců:**

V MySQL existují 3 hlavní datové typy – čísla, text a datum a čas. Správná volba datového typu ovlivňuje celkový výkon databáze.

**1. Číselné datové typy:**

Tab. 3. Číselné datové typy v MySQL

Název typu	Paměťový prostor (B)	Interval	Bez znaménka
TINYINT	1	-128 do 127	0 – 255
SMALLINT	2	-23768 do 32767	0 – 56535
MEDIUMINT	3	-8388608 do 8388607	0 – 16777215
INT	4	-2147483648 do 2147483647	0 – 4294967295
BIGINT	8	-9223372036854775808 do 9223372036854775807	0 – 18446744073709550615
FLOAT(M, D)	4	Liší se podle použitých hodnot.	-
DOUBLE (M, D)	8	Liší se podle použitých hodnot.	-
DECIMAL(M, D)	Hodnota M + 2B	Liší se podle použitých hodnot.	-

## 2. Textové datové typy

Tab. 4. Textové datové typy v MySQL

Název datového typu	Maximální velikost	Přidělená paměť
CHAR (X)	255 B	X B
VARCHAR (X)	255 B	X+1 B
TINYTEXT	255 B	X+1 B
TINYBLOB	255 B	X+2 B
TEXT	65535 B	X+2 B
BLOB	65535 B	X+2 B
MEDIUMTEXT	1,6 MB	X+3 B
MEDIUMBLOB	1,6 MB	X+3 B
LONGTEXT	4,2 GB	X+4 B
LOB	4,2 GB	X+4 B

## 3. Datum a čas

Tab. 5. Datum a čas v MYSQL

Typ	Velikost	Popis
DATE	3 bajty	datum ve formátu YYYY-MM-DD
DATETIME	8 bajtů	datum ve formátu YYYY-MM-DD HH:MM:SS
TIME	3 bajty	čas ve formátu HH:MM:SS
TIMESTAMP	4 bajty	ve formátu YYYYMMDDHHMMSS (hodnoty končí v roce 2037)

### Modifikátory:

**AUTO\_INCREMENT** – (automatické číslo) definuje datový typ pole, který při přidání nového záznamu do tabulky automaticky uloží jedinečné číslo o jedničku větší než předchozí. Nedovolí uložit už jednou použitou hodnotu, i když záznam s touto hodnotou byl již z databáze odstraněn. Modifikátor **AUTO\_INCREMENT** lze použít pouze u polí s celočíselnými hodnotami.

**ZEROFILL** – používá se k zobrazení vedoucích nul u čísel, která jsou založena na šířce zobrazení, tj. vyžadují určitý počet číslic. Chceme-li číslo 26 zobrazit jako 000026, musíme pole deklarovat jako **INT(6) ZEROFILL**.

**BINARY** – umožňuje ukládání textových informací jako binárních řetězců. Řetězce uložené v binárním formátu rozlišují velikost písmen. To se projeví také na třídění těchto polí.

**DEFAULT** – umožňuje určit hodnotu pole, jehož hodnota nebude při uložení záznamu (řádku) známa. Výchozí hodnotou používanou v systému MySQL je NULL. Hodnota NULL je dosazována do všech prázdných polí, kromě polí typu ENUM.

### **Integritní omezení:**

**NULL a NOT NULL** – určují, zda je zadání hodnoty do příslušného pole povinné.

**PRIMARY KEY** – primární klíč je pole, jehož hodnota jedinečným způsobem identifikuje každý záznam tabulky.

**UNIQUE** – omezující podmínka jedinečnosti vynucuje integritu entity klíčem, který není primární. Zajišťuje, aby do sloupců s omezující podmínkou jedinečnosti nebyly vloženy duplicitní hodnoty.

**UNSIGNED** – označuje neznaménkový číselný typ. [1]

### **3.1.2 PHP**

**PHP:** Hypertext Preprocessor (dřívější zkratka – Professional Home Pages) je široce používaný mnohoúčelový skriptovací jazyk, zvláště vhodný pro vývoj WWW aplikací a způsobilý pro vkládání do HTML dokumentů. PHP čerpá především z jazyků C, Java a Perl. Provádí se na straně serveru, kde běží jeho interpreter, který skripty vykonává. Cílem tohoto jazyka je umožnit webovým vývojářům rychle psát dynamicky generované WWW stránky. Jeho výhodou spočívá v podpoře více platform. Aplikace tak můžeme vyvíjet třeba ve Windows a pak je zpřístupnit na výkonném unixovém serveru. To je tedy velkou výhodou oproti ASP, které nás omezuje na použití pouze pod Windows. Další výhodou je, že ke zdrojovým kódům skriptů se nedostane nikdo jiný než autor. Možnosti PHP ještě umocňuje spolupráce s mnoha databázovými systémy (MySQL, PostgreSQL, MS SQL server, ...). Výsledkem běhu PHP skriptu je obyčejný HTML dokument, který by měl být schopen zobrazit každý prohlížeč, odpadají tedy problémy s kompatibilitou. PHP je nezávislé i na platformě serveru, tzn., že může běžet pod libovolným webovským serverem pracujícím v operačním systému Windows nebo Unix a je zcela zdarma.

[2]

**Syntaxe PHP:**

Soubor je tvořen kousky HTML a PHP skriptů, přičemž HTML kód se posílá přímo prohlížeči, část v PHP kódu se interpretuje na straně serveru, a do uživatelského prohlížeče se posílá pouze výstup skriptu – tj. to, co skript vypisuje na standardní výstup (např. příkazem echo).

Zdrojový kód PHP skriptu se píše mezi tagy `<? a ?>`.

```
<?php echo "Tato část kódu se vypíše na obrazovku"; ?>
<? echo "zkrácená verze"; ?>
```

Příkaz echo slouží k výpisům na standardní výstup – co se má vypsat, je uvedeno za příkazem.

Příkazy PHP příkazy musí být od sebe odděleny – pomocí středníků nebo tagů. Přímo do skriptů si můžeme psát poznámky – text mezi `//` a koncem aktuálního řádku je považován za poznámku, stejně tak text mezi `/*` a `*/`. Při psaní proměnných musíme rozlišovat malá a velká písmena, ale při psaní příkazů můžeme používat malá i velká písmena libovolně.

**Proměnné:**

Proměnné se nedeklarují předem, k zápisu se používá znak `$` a za ním je název proměnné.

```
$promenna = "text";
$promenna_2 = 56;
```

V proměnné může být uloženo cokoliv – PHP si samo určí typ hodnoty.

**Globální proměnné:**

`$GLOBALS` – obsahuje odkaz na každou proměnnou, která je momentálně dostupná v globálním kontextu skriptu, klíči tohoto pole jsou názvy globálních proměnných

`$_SERVER` – proměnné nastavované WWW serveru nebo jinak přímo spjaté s prováděcím prostředím aktuálního skriptu

`$_GET` – proměnné poskytované skriptu přes HTTP GET

`$_POST` – proměnné poskytované skriptu přes HTTP POST

`$_FILES` – proměnné poskytované skriptu přes HTTP post uploady souborů

`$_COOKIE` – proměnné poskytované skriptu přes HTTP cookies

`$_SESSION` – proměnné, které jsou momentálně registrovány v aktuální relaci skriptu

**Podmínky a cykly:**

**if** – slouží k podmíněnému provedení příkazu, pokud je splněna určitá podmínka, tu musíme zadat jako výraz, který vrací logickou hodnotu

```
if ($x! = 0) $vysledek = $y/$x;
```

Pokud se proměnná x nerovná nule, výsledek se rovná podílu proměnných y a x.

**if – else** – příkaz uvedený za else se provede v případě nesplnění podmínky.

```
if ($x > $y) echo "Správně";  
else echo "Špatně";
```

Pokud je proměnná x větší než y, vypíše se „Správně“, jinak se vypíše „Špatně“.

**switch** – na základě hodnoty jednoho výrazu provedeme jednu z několika větví skriptu. Nejprve je vyhodnocen výraz, poté jsou postupně procházeny hodnoty uvedené za klíčovým slovem case, dokud se nenalezne hodnota shodná s hodnotou výrazu, následně jsou vykonávány příkazy, dokud nenarazí na break nebo konec příkazu switch.

```
switch ($i) {  
case 0: echo "i se rovná 0"; break;  
case 1: echo "i se rovná 1"; break;  
case 2: echo "i se rovná 2"; break;  
default: echo "i se rovná 2"; break;  
}
```

**while** – provádí zadaný příkaz, dokud platí určitá podmínka. Patří mezi nejjednodušší příkazy.

```
$i = 0;  
while($i < 10){  
    echo "$i<BR>";  
    $i++;  
}
```

Vypíše čísla 0 až 9.

**do – while:** obdobný jako while, avšak podmínka je až na konci cyklu. Příkazy v tomto cyklu jsou vykonány aspoň jednou, oproti cyklu while, kde nemusí být provedeny ani jednou.

```
$i = 0;  
do {  
    echo "$i";  
    $i++;  
}  
while ($i < 10);
```

Vypíše čísla 0 až 9.



**for** – před začátkem provádění cyklu je vyhodnocen výraz1, poté výraz2 – pokud je jeho hodnota true, provede se tělo cyklu, na konci se vyhodnotí výraz3, tělo se provádí tak dlouho, dokud má výraz2 hodnotu true.

```
for ($i = 0; $i < 10; $i++)  
echo ("{$i}");
```

Vypíše čísla 0 až 9.

### 3.1.3 HTML

HyperText Markup Language (HTML) označuje jazyk pro popis dokumentu. Dokumenty WWW musí dodržovat pravidla HTML, aby je klienti mohli zobrazit. Vzhled dokumentu je jednoznačně v kompetenci klienta. Na klientovi záleží, jak bude formátovat jednotlivé prvky dokumentu. Jazyk HTML se v průběhu let několikrát změnil v souladu s technologickým vývojem a požadavky producentů softwaru a jeho uživatelů. Poslední verzí jazyka HTML, jehož konečnou podobu schválilo konsorcium W3C ([www.w3c.org](http://www.w3c.org)), je verze 4.01.

V současné době se již od jazyku HTML přechází k jazyku novému, pojmenovanému XHTML (eXtensible HyperText Markup Language), jehož tvůrcem je opět konsorcium W3C. Jazyk XHTML je vlastně HTML verze 4.01 upravená jako aplikace XML (eXtensible Markup Language), díky čemuž je možno přenášet data mezi všemi platformami a technologiemi (např. mobilní telefony, palmtopy, ...). Jazyk XHTML má na rozdíl od jazyka HTML mnohem přísnější pravidla a klade větší důraz na čistotu a správnost zdrojového kódu (např. všechny tagy musí být psány malými písmeny, musí se ukončovat, ...).

#### **Základní struktura HTML dokumentu:**

Při tvorbě HTML dokumentu je třeba dodržovat určitou pevnou strukturu. Každý HTML zahrnuje několik značek, vymezujících jeho základní části.

Struktura HTML dokumentu:

```
<HTML>  
<HEAD>  
hlavička dokumentu  
</HEAD>  
<BODY>  
tělo dokumentu  
</BODY>  
</HTML>
```

Celý dokument je ohraničen značkami <HTML> a </HTML>. Jeho text je rozdělen na dvě části: hlavičku (tu vymezují značky <HEAD> a </HEAD>) a tělo (uzavřeno mezi tagy <BODY> a </BODY>). Hlavička zpravidla bývá velmi jednoduchá. Obsahuje především režijní informace, určené klientskému programu (k tomu slouží nepárový element <META>) a dále pak i titulek dokumentu uzavřený mezi značky <TITLE> a </TITLE>. Tělo dokumentu obsahuje vlastní text, který bude zobrazen uživateli. Součástí textu mohou být tabulky, obrázky, odkazy, text, .... [5]

### 3.1.4 CSS – kaskádové styly

HTML je značkovací jazyk, ve kterém by se pomocí značek měl vyznačovat význam jednotlivých částí textu. V HTML existuje několik atributů a elementů, které ovlivňují pouze grafický vzhled. Použitím těchto atributů a elementů je sice možno získat graficky atraktivní stránku, má to však řadu nevýhod. Text stránky je mnohdy špatně strukturován, protože jednotlivé elementy jsou využívány účelově k dosažení určitých grafických efektů. Druhou velkou nevýhodou je velká pracnost, protože většina z vizuálních atributů musí být nastavována opakovaně u všech elementů. Obě tyto nevýhody odstraňují kaskádové styly – Cascading Style Sheets (CSS).

Kaskádové styly umožňují definovat způsob zobrazení (druh a velikost písma, barvu, zarovnání, velikost okraje, barvu pozadí, apod.) každého elementu (existujícího, např.: „*HI*“, případně nově vytvořeného, např.: „*nadpis*“) na stránce. Styl však není přímo součástí textu stránky a tak může být zápis stránky přehlednější a dobře strukturovaný. Navíc styly umožňují definovat jednotný vzhled určitého elementu v celém dokumentu jedním zápisem, tzn., že není potřeba jej opakovat u každého elementu. Styly je možné definovat buď přímo v záhlaví HTML dokumentu nebo v externím souboru (soubor ve tvaru „\*.css“), což je nejčastější případ. Výhodou definice stylů v externím souboru je to, že tento jeden externí soubor může být využíván více WWW stránkami. Při požadavku na změnu vzhledu WWW stránek stačí pouze upravit daný styl v tomto externím souboru a změny se pak automaticky promítnou do všech WWW stránek, které využívají daný externí soubor.

Velkou předností kaskádových stylů je i podpora jiných platforem. Pomocí kaskádových stylů lze totiž nadefinovat rozdílný vzhled např. pro výstup na tiskárnu, pro mobilní telefony, atd. [5]

### 3.1.5 Javascript

Skriptovací jazyk Javascript (Java skript) se používá pro vytváření malých a jednoduchých aplikací, které jsou spuštěny prostřednictvím prohlížeče WWW stránek. Je to jazyk interpretovaný – nemusí se kompilovat. Syntaxí se podobá programovacímu jazyku C a Javě, ale je podstatně jednodušší. Java skript je jazykem objektově orientovaným, má již vestavěny určité objekty, které umožňují programátorovi pracovat s datem a časem, oknem prohlížeče, zapisovat na plochu prohlížeče nebo provádět různé matematické operace. Jeho nevýhodou je, že každý uživatel, prohlížející si WWW stránky si může Javascript vypnout přímo ve svém prohlížeči, např. z bezpečnostních důvodů.

Zdrojový kód v Javascriptu se píše mezi tagy `<SCRIPT>` a `</SCRIPT>`, popřípadě ho lze načíst z externího souboru `<SCRIPT language="JavaScript" src="soubor.js"></SCRIPT>`.  
[5]

### 3.1.6 Instalace programu Easy-PHP (Apache, PHP, MySQL)

Jedná se o samoinstalační balíček, celou instalaci provedete spuštěním souboru „setup.exe“ a výběrem umístění programu.

Webový server **apache** se konfiguruje pomocí souboru `httpd.conf`, který se nachází v adresáři „`apache/conf/`“. Zde je důležité nastavit cestu (DocumentRoot) k našim skriptům.

**MySQL** server se konfiguruje pomocí souboru „`my.ini`“ (nachází se v adresáři „`mysql`“) kde můžeme změnit cestu k naší databázi.

**PHP** má konfigurační soubor „`php.ini`“, který se nachází v adresáři „`php`“.

Součástí programu Easy-PHP je taky správce databází `phpMyAdmin`, kterého jsem využil při návrhu a vytvoření databáze. Program `phpMyAdmin` je napsán v jazyce PHP, je to vlastně webové rozhraní, které umožňuje uživatelům snadnou tvorbu a správu databází MySQL.

## 3.2 Postup při modifikaci informačního systému CAAC

Hlavním úkolem bylo převést již existující data ze souborů HTML do databáze. Zvolil jsem databázový systém MySQL a to hned z několika důvodů. V současné době se jedná o nejoblíbenější databázový systém, který je hodně rozšířený a používán při tvorbě moderních WWW stránek, a zaujal mě již při výuce v předmětu „Databázové systémy“. Jeho hlavní výhodou je, že je poskytován zcela zdarma.

Při programování WWW stránek jsem používal programovací jazyk PHP, HTML jazyk, Javascript, kaskádové styly CSS pro formátování a grafickou úpravu stránek a samozřejmě jazyk SQL pro databázové příkazy.

### 3.2.1 Vytvoření databáze MySQL

Při vytváření databáze a jednotlivých tabulek jsem používal aplikaci phpMyAdmin. Je to vlastně webové rozhraní, které umožňuje uživatelům snadnou správu databází MySQL.

Vytvořil jsem základní databázi, kterou jsem nazval „caac\_db“. Poté jsem v této databázi vytvořil tabulku „subsystemy“, do které se ukládá nadpis, zdrojový kód, autor každého nově vytvořeného subsystému. Při vytvoření nového subsystému přes administraci systému CAAC, se v databázi vytvoří nová tabulka pod názvem „subsystem\_data“, kde subsystem je název tohoto nově vytvořeného subsystému. Zatím existují 3 subsystémy, a to Analýza (analiza\_data), Syntéza (synteza\_data) a Knihovna hotových programů (hotprogr\_data). Příponu „\_data“ jsem použil kvůli odlišení tabulek subsystémů od ostatních tabulek.

Celá správa databáze probíhá přes webové rozhraní – administraci systému CAAC. Je zde možné vytvářet, upravovat a mazat nové subsystémy a základní submoduly. Probíhá zde taky správa „diskuze“ a „literatury“ a další nastavení – změna emailu příjemce dotazů a nastavení přihlašovacího jména a hesla k administraci.

### 3.2.2 Modifikace informačního systému CAAC

Při modifikaci stávajícího informačního systému CAAC bylo důležité zachovat adresářovou strukturu a celkové grafické provedení. Modifikaci jsem provedl tak, že jsem všechny existující subsystémy, základní submoduly a stránku s literaturou převedl do databáze. Při převodu základních submodulů jsem vytvořil PHP skript, který přímo načtený HTML soubor rozdělí na části – nadpis, HTML text, literatura, autor a datum vypracování. Tím jsem si hodně ulehčil práci a taky jsem později využil tento skript při vytváření nových základ-

ních submodulů přes webové rozhraní. Jakmile byla všechna data převedena v databázi, vytvořil jsem skripty na zobrazování jednotlivých subsystémů a základních submodulů, které načítají potřebná data z databáze. Dále jsem vytvořil skripty pro tisk stránky, odesílání dotazu na email a stránku s diskuzí k jednotlivým základním submodulům. Potom jsem vytvořil administrátorské prostředí pro celkovou správu informačního systému CAAC.

### 3.2.3 Jedny z možných návrhů na vylepšení informačního systému CAAC

Jedny z možných návrhů na vylepšení informačního systému CAAC jsou například:

- začlenění vybraného WYSIWYG editoru do administrace informačního systému CAAC za účelem snadnější úpravy již existujících, případně nově vytvořených subsystémů a základních submodulů,
- rozšíření možnosti administrace o další správce s jinými právy než administrátor (mohli by např. pouze vytvářet nové základní submoduly),
- vytvoření podrobnějších statistik přístupů k jednotlivým stránkám, za účelem větší informovanosti o využití informačního systému CAAC uživateli,
- vytvoření modulu pro otestování znalostí z vybraných částí informačního systému CAAC.

## **II. PRAKTICKÁ ČÁST**

## 4 ÚPRAVA INFORMAČNÍHO SYSTÉMU CAAC

### 4.1 Návrh databáze

Struktura databáze vznikala postupně s vývojem celých stránek, ale základní návrh je dodržen. Název databáze je „caac\_db“, obsahuje tabulku „subsystemy“, ve které jsou uloženy úvodní stránky subsystémů. Každý subsystém má vlastní tabulku, jejíž název je shodný s názvem adresáře subsystému s příponou „\_data“. Do této tabulky subsystému se ukládají data (nadpis, text, literatura, autor,...) jednotlivých základních submodulů. Dále je zde tabulka „login“, která obsahuje jméno a heslo administrátora stránek, tabulka „diskuze“ pro ukládání příspěvků z diskuze na stránkách a tabulka „ostatní\_nastavení“, v níž je uložen email administrátora, na který budou zasílány dotazy.

### 4.2 Změna úvodní stránky

Jako první jsem přepracoval úvodní stránku, na které jsem změnil logo informačního systému CAAC, tlačítko start a hlavně jsem vytvořil počítadlo přístupů. Celkový počet přístupů se ukládá do databáze, aby nedocházelo k navyšování počítadla při každé aktualizaci stránky, nebo při návratu zpět na úvodní stránku, využil jsem relací (session). Pokud existuje relace „pocitadlo“ při vstupu na stránku, nedojde k inkrementaci počítadla.

```
session_start();
//pokud session existuje, vypíše se počet přístupů//
if (isset($_SESSION['pocitadlo'])) {
    $dotaz = mysql_query("select value from ostatni_nastaveni where
text_id='pocitadlo'");
    while ($pole = mysql_fetch_array($dotaz)) {
        $pristupu = $pole['value'];
    }
    //pokud session neexistuje, zaregistrujeme ji a přičte se 1 přístup//
    else {
        $_SESSION['pocitadlo'] = '1';
        $dotaz = mysql_query("select value from ostatni_nastaveni where
text_id='pocitadlo'");
        while ($pole = mysql_fetch_array($dotaz)) {
            $pristupu = $pole['value']+1;
        }
        mysql_query("update ostatni_nastaveni set value='$pristupu' where
text_id='pocitadlo'");}
}
```

### 4.3 Vytvoření skriptu na převod HTML souborů do databáze

Již existující HTML soubory základních submodulů jsou vytvořeny podle předem daných pravidel, proto bylo možné vytvořit skript, který načte námi zvolený HTML soubor. Přitom jej rozdělí na nadpis, HTML kód, literaturu, výpočetní program, autora, pracoviště a datum vypracování. Pracuje přitom následujícím způsobem. Máme předem nadefinované oddělovače začátku a konce potřebné části řetězce, kterou chceme rozpoznat, do proměnné se uloží část textu pouze mezi těmito oddělovači. Přitom ještě odstraní případné mezery na konci řetězce. Každou takto rozpoznanou část HTML souboru uloží do databáze.

Například oddělovač pro „nadpis“:

```
$oddel_nadpis_1 = "<h1>";
$oddel_nadpis_2 = "</h1>";
$nadpis_zac= strpos($obsah,$oddel_nadpis_1)+strlen($oddel_nadpis_1);
$nadpis_end = strpos($obsah,$oddel_nadpis_2,$nadpis_zac);
$nadpis = trim(substr($obsah,$nadpis_zac,$nadpis_end-$nadpis_zac));
```

Začátek nadpisu je řetězec „<h1>“, konec je „</h1>“, zjistíme pozici prvního a posledního znaku řetězce nadpisu a do proměnné „\$nadpis“ uložíme text mezi těmito dvěma pozicemi. PHP funkce „trim()“ odstraní přebytečné mezery z konce řetězce.

Vložení do databáze provedeme pomocí SQL příkazu INSERT.

```
$dotaz = "INSERT INTO $subsystem VALUES
(\"\", \"$nadpis\", \"$text\", \"$literatura\", \"$autor\", \"$pracoviste\",
\", \"$datum_vyprac\");
MySQL_Query($dotaz);
```

Tento skript jsem později upravil a využil v administraci při vytváření nových základních submodulů.

### 4.4 Čtení dat z databáze

Jakmile jsem měl všechna data v databázi, vytvořil jsem skripty pro zobrazování jednotlivých subsystémů a základních submodulů. Předtím jsem vytvořil 2 skripty pro připojení a odpojení od databáze.

#### **Připojení k databázi – soubor connect\_db.php :**

V tomto souboru je definice názvu databáze, názvu serveru, na kterém máme naši databázi a přihlašovací jméno a heslo k databázi. Při volání tohoto skriptu dojde ke spojení



s databází a k nastavení znakové sady databáze (kvůli správnému zobrazení českých znaků).

```
<?php
$server="localhost";
$uzivatel="root";
$heslo="";
$db="caac_db";
//----- spojení s databází -----//
    $spojeni = mysql_connect($server,$uzivatel,$heslo);
    if (!$spojeni) { echo "nepřipojen k databázi"; }
    if (!mysql_select_db($db,$spojeni)) { echo "chyba v připojení k
databázi"; }
    //----- nastavení znakové sady databáze -----//
    mysql_query("SET CHARACTER SET cp1250", $spojeni) or
die(mysql_error());
    mysql_query("SET collation_connection=cp1250_general_ci", $spojeni)
or die(mysql_error());
    mysql_query("SET collation_connection=cp1250_czech_cs", $spojeni)
or die(mysql_error());
?>
```

### Odpojení od databáze – soubor close\_db.php :

Při volání tohoto skriptu k uvolnění výsledku z paměti a k odpojení databáze.

```
<?php
    mysql_free_result($dotaz); //uvolnění výsledku
    mysql_close($spojeni); //odpojení od databáze
?>
```

#### 4.4.1 Zobrazení subsystémů

Každému ze subsystémů náleží PHP soubor, ve kterém je uložen název tohoto subsystému a následně je volán skript pro zobrazení subsystémů – „zobraz\_subsystem.php“. Tento skript pracuje tak, že načte z databáze potřebná data a potom je zobrazí v předem určené podobě jako HTML stránku. Přitom využívá jako vstupní parametr název subsystému.

Příklad:

Soubor „analyza.php“:

```
<?php
$system = "analyza";
include ("zobraz_subsystem.php");
?>
```

Soubor „zobraz\_subsystem.php“ nejprve načte z databáze potřebná data z tabulky „analyza“ a poté je zobrazí.

```
$dotaz = mysql_query("Select * FROM subsystemy WHERE subsystem='<!--> $subsystem'");  
while ($pole = MySQL_Fetch_Array($dotaz)) {  
    $nadpis = $pole['nadpis'];  
    $text = $pole['text'];  
    $autor = $pole['autor'];  
}
```

Načtení dat z databáze.

```
<h1><?php echo $nadpis;?></h1>  
<br>  
<?php echo $text;?>
```

Zobrazení jako HTML soubor – použití HTML tagů pro formátování textu.

#### 4.4.2 Zobrazení základních submodulů

Tento skript pracuje stejným způsobem jako „zobraz\_subsystem.php“, má však 2 vstupní parametry, a to subsystem, do kterého daný základní submodul patří a název tohoto základního submodulu.

Příklad:

Soubor základního submodulu „aproxtd.php“:

```
<?php  
    $subsystem = "analyza";  
    $zakl_submodul = "aproxtd";  
    include ("../..<!--> /zobraz_zakl_submodul.php");  
?>
```

#### 4.4.3 Další skripty – tisk, dotaz na email a vyhledávání

**Tisk stránek – náhled před tiskem:**

Jde o skript „tisk.php“ který je volán, pokud chceme vytisknout stránku některého základního submodulu. Při jeho spuštění se otevře nové okno, ve kterém se zobrazí text daného základního submodulu s jiným formátováním než je původní stránka (formátování textu při tisku je dáno nastavením kaskádových stylů v souboru „tisk.css“ v adresáři „styles“), stránka je bez tlačítek a je pouze černobílá. Při načtení stránky (nebo při kliknutí myší na stránku) dojde k volání funkce „window.print();“ a k vlastnímu tisku.

**Dotaz na email:**

Pokud máme k danému učivu nějaký dotaz, můžeme autorovi přímo zaslat email. K tomu slouží skript „dotaz.php“, který obsahuje formulář, a skript „dotaz\_odeslat.php“, v němž je využita PHP funkce „mail“ k odeslání dotazu.

```
if (mail($prijemce, $predmet, $text, $hlavicka)){
    echo ("E-mail byl úspěšně odeslán...");
}
```

**Vyhledávání na stránkách:**

Při vyhledávání na stránkách jsem využil fulltextového vyhledávání a indexování v MySQL. To spočívá ve vytvoření FULLTEXT indexu nad sloupci, ve kterých chceme vyhledávat – můžeme potom zvolit kategorii vyhledávání (např. vyhledávání v nadpisech, textu, klíčových slovech,...). Samotné vyhledávání probíhá pomocí funkcí MATCH() a AGAINST(), které vrací nulu, pokud hledané slovo nebylo nalezeno, nebo kladné číslo podle počtu nalezených záznamů. Dále můžeme vyhledávat podle kritérií vyhledávání, a to AND, OR a hledat přesně zadaný řetězec.

**Vytvoření FULLTEXT indexu:**

```
if(mysql_query("ALTER TABLE $kategorie ADD FULLTEXT search
($indexes)"));
```

**Vytvoření dotazu pro vyhledávání:**

```
$query="SELECT *,MATCH(".$indexes.")
AGAINST('".$_POST["searchtext"]." IN BOOLEAN MODE) as score FROM
$kategorie
WHERE MATCH(".$indexes.") AGAINST('".$_POST["searchtext"]." IN
BOOLEAN MODE)";
// doplnění o sestupné řadění výsledků hledání podle score
$query.=" ORDER BY score DESC";
$result = mysql_query($query); // vykonání dotazu
```

\*pozn. – v proměnné \$indexes jsou uloženy názvy sloupců, ve kterých se vyhledává

Poté vypíšeme nalezené záznamy například do tabulky.

```
echo "<table>
      <tr>
          <td><b>nadpis</b></td>
          <td><b>autor</b></td>
          <td><b>poslední modifikace</b></td>
      </tr>";
while($pole=mysql_fetch_array($result)){
    $nadpis = $pole['nadpis'];
    $autor = $pole['autor'];
    $datum_vyprac = $pole['datum_vyprac'];
echo "<tr>
      <td>$nadpis</td>
      <td>$autor</td>
      <td>$datum_vyprac</td>
    </tr>";}
echo "</table>";
```

Nakonec zrušíme FULLTEXT index:

```
if(mysql_query("ALTER TABLE $kategorie DROP INDEX search"));
```

#### 4.4.4 Diskuze

Diskuze slouží uživatelům stránek jako možnost dotazu k danému tématu. Úvodní stránka je jakýmsi výpisem všech subsystémů a jejich základních submodulů. Ke každému základnímu submodulu je možno vložit dotaz – příspěvek. Po kliknutí myší na vybraný základní submodul dojde k zobrazení příspěvků k danému tématu, pokud nikdo zatím žádný příspěvek nevložil, zobrazí se formulář pro vložení nového příspěvku. Příspěvky jsou řazeny podle data sestupně a je zajištěno stránkování (po 10 ti příspěvcích). Administrátor odpovídá na dotazy přímo přes stránku diskuze, nebo přes administraci. V administraci má navíc možnost tyto příspěvky řadit podle datumu, příslušného subsystému, nebo základního submodulu, nebo zobrazit všechny příspěvky, u kterých prozatím není vložena odpověď.

## 5 ADMINISTRACE INFORMAČNÍHO SYSTÉMU CAAC

Všechny soubory, které jsou používány při administraci informačního systému CAAC se nachází v adresáři „admin“. Názvy těchto souborů jsou voleny tak, že například soubor, pomocí kterého dojde k vytvoření nového subsystému se nazývá „novy\_subsystem.php“, soubor pro smazání základního submodulu „smazat\_zakl\_submodul.php“, ....

### 5.1.1 Přihlášení k administraci

Kliknutím myši v pravém horním rohu stránky „Menu CAAC“ dojde k zobrazení přihlašovací stránky v novém okně. Pro vstup do administrace je nutné zadat správné jméno a heslo, která jsou uložena v databázi. Při přihlašování dojde k ověření jejich správnosti.

### 5.1.2 Hlavní strana administrace

V horní části stránky je menu, ve kterém máme možnost vybrat mezi úpravou subsystémů, základních submodulů a submodulů knihovny hotových programů. Dále zde můžeme upravit stránku literatury, spravovat diskuzi a pod položkou ostatní nastavení máme možnost nastavit email pro příchozí dotazy, upravovat menu CAAC, zobrazit statistiky zobrazení jednotlivých základních submodulů a upravovat seznam naplnění jednotlivých subsystémů. Pod tímto menu je výpis všech subsystémů a jejich základních submodulů.

### 5.1.3 Úprava subsystémů

Při výběru položky menu „úprava subsystémů“ dojde k zobrazení nabídky pro úpravu subsystémů. Zde můžeme změnit existující subsystém, vytvořit nový subsystém, nebo smazat subsystém.

#### **Změnit subsystém:**

Na této stránce vybereme subsystém, který chceme upravit, při jeho výběru dojde k automatickému zobrazení nadpisu, HTML kódu stránky, autora s možností jejich úpravy. Upravenou stránku subsystému uložíme kliknutím na tlačítko „uložit změny“.

#### **Nový subsystém:**

V horní části stránky je výpis již existujících subsystémů. Do pole „název/adresář“ zadáme název nového subsystému (stejně tak se bude jmenovat adresář nového subsystému a tabulka v databázi s příponou „\_data“). Podle původních pravidel by tento název neměl obsahovat diakritiku, a měl by mít méně než 8 znaků. Nadpis už diakritiku obsahovat může a

může mít libovolnou délku. Dále vyplníme HTML kód stránky a autora. Nový subsystém vytvoříme kliknutím na tlačítko „vytvořit“. Při vytvoření nového subsystému se vytvoří nová tabulka v databázi („název subsystému“\_data), do které se budou později ukládat data jednotlivých základních submodulů náležících do tohoto nového subsystému. Potom se vloží do tabulky „subsystemy“ námi zadaný nadpis, HTML kód stránky a autor nového subsystému a vytvoří se taky adresář „název subsystému“ a soubor „název subsystému“.php, který obsahuje název nového subsystému a odkaz na soubor „zobraz\_subsystem.php“. Zavoláním tohoto souboru („název subsystému“.php) dojde k zobrazení nového subsystému. Dále je nutné upravit menu informačního systému CAAC, a to následujícím způsobem. Menu upravíme kliknutím na položku „ostatní nastavení“ a výběrem „změna menu“. U námi vytvořeného subsystému přepíšeme u odkazu (tag <a></a>) atribut „href“, aby byl odkaz aktivní. Tím je nový subsystém vytvořen a je přímo dostupný přes menu informačního systému CAAC.

#### **Smazat subsystém:**

Před smazáním subsystému je nutné smazat celý obsah adresáře, neboli odstranit všechny jeho základní submoduly, jinak dojde pouze k odstranění subsystému z databáze a neodstraní se všechny soubory z disku. Subsystém, který chceme smazat, vybereme z nabídky a klikneme na tlačítko „smazat“. Budeme upozorněni na to, že dojde ke smazání námi vybraného subsystému, při potvrzení dojde k jeho odstranění – smaže se tabulka „subsystém“\_data z databáze a z tabulky „subsystemy“ se odstraní řádek se subsystémem. Pokud je adresář se subsystémem prázdný, odstraní se taky.

#### **5.1.4 Úprava základních submodulů**

Při výběru položky menu „úprava zákl. submodulů“ dojde k zobrazení nabídky pro úpravu základních submodulů. Zde můžeme změnit, vytvořit nový, nebo smazat základní submodul.

#### **Změna základního submodulu:**

Výběrem možnosti „změnit zákl. submodul“ se nám zobrazí stránka, kde máme na výběr subsystém a jeho základní submodul. Jejich zvolením dojde k automatickému zobrazení nadpisu, autora, data vypracování, názvu základního submodulu, vlastního HTML kódu stránky, literatury, výpočetního programu, pracoviště a klíčových slov, které můžeme upravit. Provedené úpravy uložíme kliknutím na tlačítko „uložit změny“ ve spodní části obra-

zovky. Máme zde taky možnost zobrazit náhled stránky, vrátit změny a zobrazit a vložit do textu obrázky dostupné v adresáři se základním submodulem.

#### **Nový základní submodul:**

Nový základní submodul je možno vytvořit dvěma způsoby. Vpisováním hodnot do daných formulářových polí, nebo přímým načtením a rozdělením HTML souboru na požadované části. K tomu slouží tlačítko „načíst soubor“, na zobrazené stránce vybereme subsystem, do kterého patří nově vytvářený základní submodul, klikneme na „Procházet“ a vybereme HTML soubor z pevného disku. Kliknutím na „načíst soubor“ dojde k načtení a rozpoznání jednotlivých částí (nadpisu, obsahu, literatury, ...). Potom můžeme přímo vytvořit základní submodul kliknutím na tlačítko „vytvořit nový základní submodul“, nebo načtená data vložit zpět do předchozího formuláře a zkontrolovat tak jednotlivá pole formuláře před vložením do databáze a vytvořením nového základního submodulu. Nový základní submodul vytvoříme kliknutím na tlačítko „vytvořit“ v dolní části obrazovky. Pokud se v základním submodulu vyskytují také obrázky, je nutné je zkopírovat do správného adresáře. Název tohoto adresáře se vypíše na obrazovku při vytvoření nového základního submodulu.

#### **Smazat základní submodul:**

Výběrem subsystemu, jeho základního submodulu a kliknutím na tlačítko „smazat“ dojde po potvrzení dotazu o smazání základního submodulu ke smazání tohoto námi vybraného základního submodulu. Dojde k odstranění dat z databáze, souboru základního submodulu s příponou „.php“ a adresáře základního submodulu, včetně případných obrázků.

### **5.1.5 Úprava submodulů knihovny hotových programů**

Ovládání „úpravy submodulů knihovny hotových programů“ je naprosto stejné jako úprava subsystemů [5.1.3] a úprava základních submodulů [5.1.4].

### **5.1.6 Úprava stránky literatury**

Stránka literatury se skládá prozatím ze čtyř částí, a to „Analýza“, „Syntéza“, „Logické řízení“ a „Další literatura“. Výběrem jedné z možností se zobrazí HTML kód této vybrané části, kterou můžeme upravit a uložit kliknutím na tlačítko „uložit změny“. Pokud klikneme na tlačítko „autor a datum“, můžeme upravit autora a datum, který se zobrazí v dolní části stránky s literaturou. Při vytvoření nového subsystemu se automaticky vytvoří další část literatury, jejíž nadpis je shodný s nadpisem subsystemu.

### 5.1.7 Správa diskuze

V informačním systému CAAC je vytvořena „diskuze“, kde mohou návštěvníci stránek psát připomínky a dotazy k probíranému tématu. Ve správě diskuze jsou zobrazeny všechny příspěvky v diskuzi, řazené podle data, subsystému, základního submodulu, jména příspěvovatele a je zde možnost taky zobrazit pouze příspěvky bez odpovědi. U každého příspěvku můžeme vložit, upravit nebo smazat odpověď, popřípadě smazat celý příspěvek.

### 5.1.8 Ostatní nastavení

#### **Nastavení emailu:**

Zde nastavíme email administrátora, na který mají být zasílány dotazy k danému tématu. Aktuálně používaný email je zobrazen v poli „Aktuální email“. Pokud chceme změnit tento email, napíšeme jej do pole „Nový email“ klikneme na tlačítko „uložit“.

#### **Datum aktualizace:**

Zde můžeme nastavit datum aktualizace, který se zobrazuje na úvodní stránce informačního systému CAAC.

#### **Změna menu:**

Jedná se o změnu HTML kódu souboru „menu.html“. Můžete zde přidávat, mazat nebo upravovat jednotlivé položky menu informačního systému CAAC.

#### **Statistiky:**

Na této stránce je zobrazen vybraný subsystém a „počet shlédnutí“ každého jeho základního submodulu.

#### **Seznam naplnění:**

Jedná se o seznam základních submodulů, kde je zobrazeno, z jaké části jsou naplněny. Toto naplnění je zobrazeno ve sloupci „naplnění“ tabulky. V horní části máme na výběr řazení podle subsystému – výběrem daného subsystému zobrazíme všechny jeho základní submoduly a stav naplnění. Aktualizace záznamů se provede automaticky při úvodním zobrazení stránky. Význam jednotlivých položek a jejich hodnot naleznete zde [2.4].

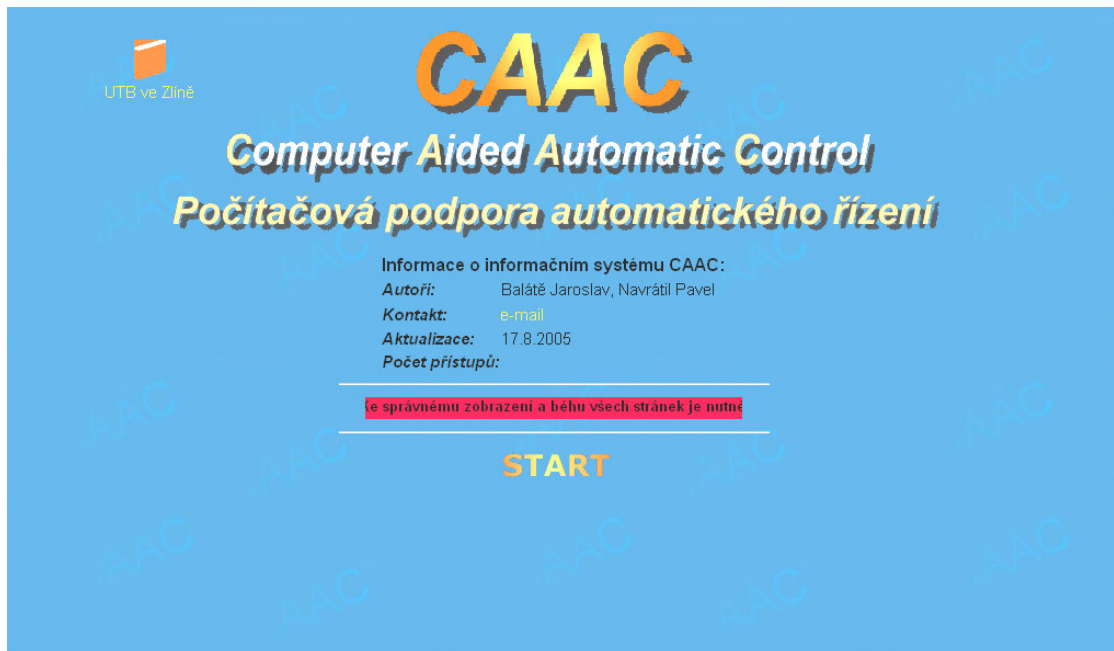
### 5.1.9 Odhlášení

Po celou dobu přihlášení v administraci systému CAAC je v pravém horním rohu obrazovky zobrazeno jméno přihlášeného uživatele (administrátora). Odhlášení provedeme kliknutím na odkaz „odhlásit se ...“ vedle jména přihlášeného administrátora. Přitom dojde k odhlášení z administrace a k automatickému uzavření okna prohlížeče.

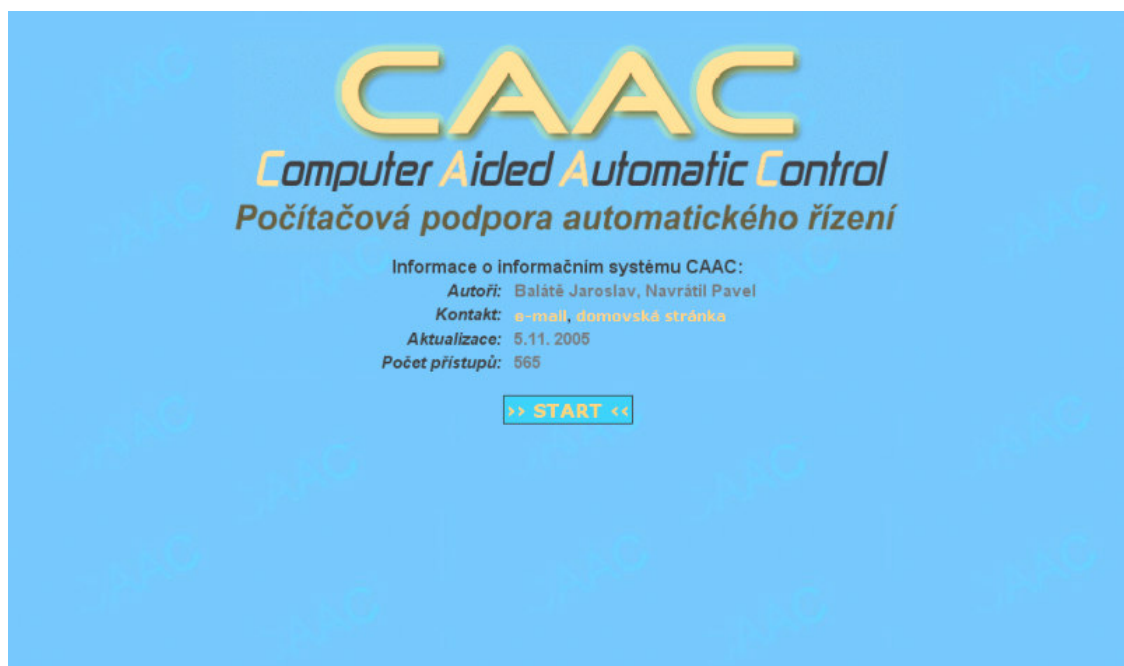


## 6 MODIFIKOVANÝ INFORMAČNÍ SYSTÉM CAAC

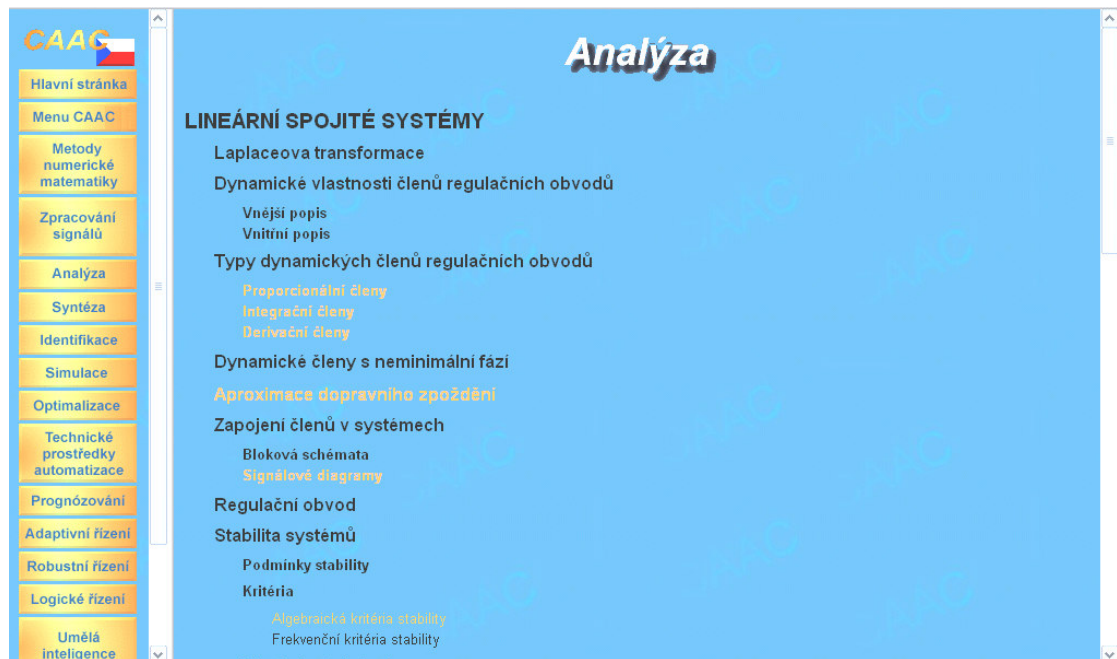
Na následujících stránkách jsou zobrazeny náhledy vybraných stránek původního informačního systému CAAC a jeho modifikované verze.



Obr. 5. Vzhled původní úvodní strany



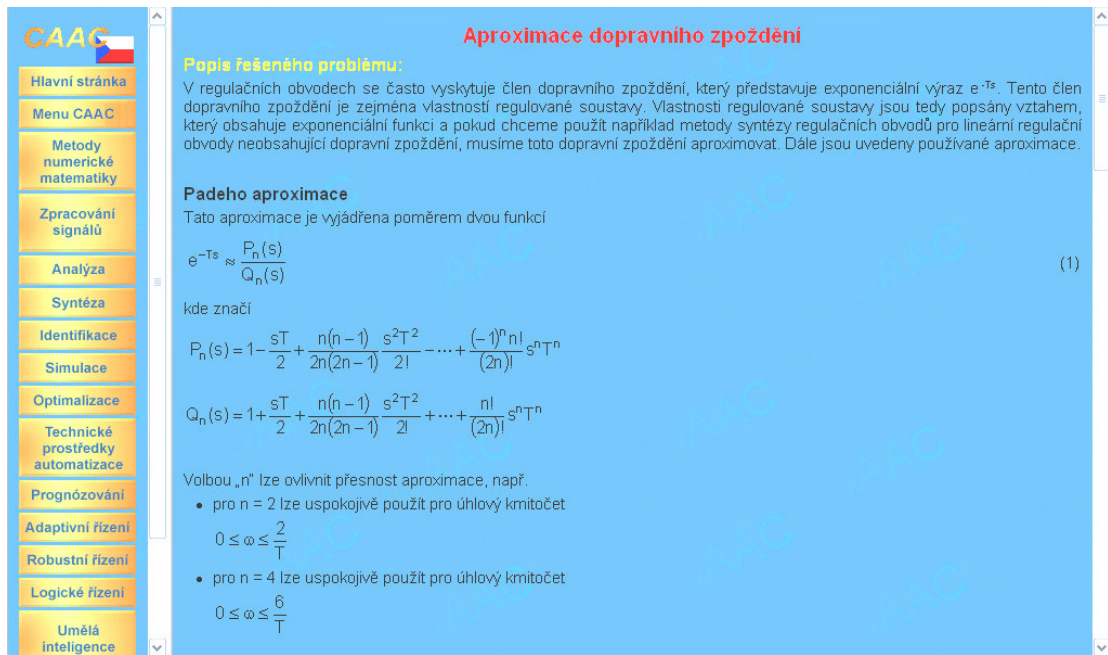
Obr. 6. Vzhled úvodní strany modifikované verze



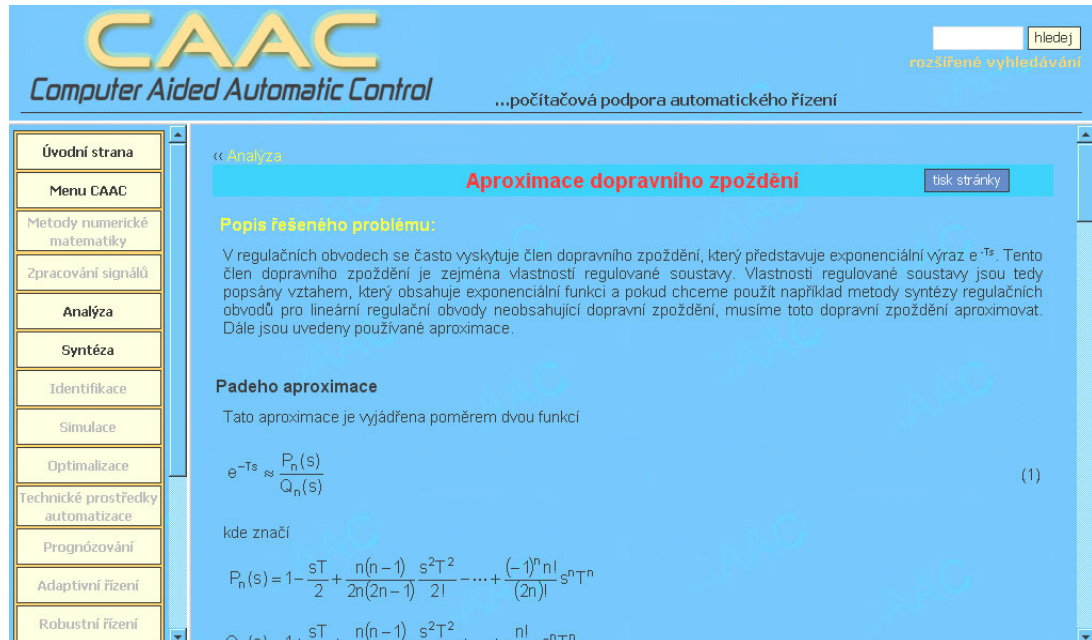
Obr. 7. Vzhled původní stránky subsystému



Obr. 8. Vzhled stránky subsystému modifikované verze



Obr. 9. Vzhled původní stránky základního submodulu



Obr. 10. Vzhled stránky základního submodulu modifikované verze

## ZÁVĚR

Mým úkolem bylo navrhnout postup a modifikovat informační systém CAAC využitím moderních nástrojů při zachování stávajících pravidel (struktura informačního systému, struktura adresářů, umístění a názvy souborů, celkové grafické provedení, ...). Vždycky je lepší tvořit WWW stránky od základu, než měnit již existující, kde je navíc nutné dodržet určitá pravidla. Má práce spočívala v modernizaci, což se doufám podařilo – je vytvořena celková administrace stránek, vyhledávání na stránkách, diskusní fórum, zjednodušený tisk stránek s náhledem před vytištěním.

Zvolil jsem databázový systém MySQL a to hned z několika důvodů. V současné době se jedná o nejoblíbenější databázový systém, který je hodně rozšířený a používaný při tvorbě moderních WWW stránek, a zaujal mě již při výuce v předmětu „Databázové systémy“. Jeho hlavní výhodou oproti ostatním databázovým systémům je, že je poskytován zcela zdarma. Při programování WWW stránek jsem používal programovací jazyk PHP, jehož výhodou je, že běží na většině dnes používaných operačních systémech (Windows, Linux, ...) a je taky zcela zdarma. Dále jsem samozřejmě využíval HTML jazyk, Javacript, kaskádové styly CSS pro formátování a grafickou úpravu stránek a jazyk SQL pro databázové příkazy.

Informační systém CAAC by měl sloužit jako podpora výuky předmětů zaměřených na automatické řízení, jeho modernizací by měla být tato podpora kvalitnější (možnost využít diskuzi k probíranému tématu, dotazy zasílané na email, vytvořený tisk jednotlivých stránek s náhledem) a jednodušší by mělo být také samotné naplňování jednotlivých subsystémů a základních submodulů. Celková aktualizace a správa internetových stránek je nyní zajištěna pomocí administrátorského prostředí.

Součástí této práce bylo také mimo zpracování dokumentace k nové verzi informačního systému CAAC, také zpracovat literární rešerši pro oblast zabývající se tvorbou informačních systémů na Internetu. V této části jsem popsal informační systém Moodle – softwarový balíček pro tvorbu výukových systémů a elektronických kurzů na internetu, který využívá naše fakulta jako podporu při výuce některých předmětů a dále jsem popsal redakční a informační systém phpRS.

**SEZNAM POUŽITÉ LITERATURY**

- [1] *MySQL* [online]. c1995 [cit. 2006–01-18]. Dostupný z WWW: <<http://dev.mysql.com>>.
- [2] *PHP* [online]. c2001 [cit. 2006–01-18]. Dostupný z WWW: <<http://php.net>>.
- [3] KOSEK, J. *Téměř vše o WWW* [online] c1999 [cit. 2006–01-18]. Dostupný z WWW: <<http://www.kosek.cz>>.
- [4] WELLING, L., THOMSON, L.: *PHP a MySQL – rozvoj webových aplikací*. Praha: SoftPress, 2002.
- [5] JANOVSKEÝ D. *Jak psát web* [online]. c2005 [cit. 2006–06-07]. Dostupný z WWW: <<http://www.jakpsatweb.cz>>.
- [6] *phpRS* [online]. c2001 [cit. 2006–06-07]. Dostupný z WWW: <<http://www.phprs.cz>>.
- [7] *Moodle* [online]. c2003 [cit. 2006–06-07]. Dostupný z WWW: <<http://www.moodle.cz>>.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

- CAAC Computer Aided Automatic Control; počítačová podpora automatického řízení.
- Moodle Softwarový balíček pro tvorbu výukových systémů a elektronických kurzů na internetu.
- phpRS Redakční a informační systém.
- HTML HyperText Markup Language; jazyk pro tvorbu WWW stránek.
- CSS Cascading Style Sheets; kolekce metod pro grafickou úpravu WWW stránek.
- PHP PHP: Hypertext preprocessor (dříve Professional Home Pages); skriptovací jazyk.
- MySQL Databázový systém.
- SQL Structured Query Language; jazyk pro práci s databází MySQL.

**SEZNAM OBRÁZKŮ**

<i>Obr. 1. Vzhled informačního systému Moodle</i> .....	12
<i>Obr. 2. Vzhled redakčního systému phpRS</i> .....	14
<i>Obr. 3. Výčet názvů tematických okruhů informačního systému CAAC</i> .....	15
<i>Obr. 4. Struktura informačního systému CAAC</i> .....	16
<i>Obr. 5. Vzhled původní úvodní strany</i> .....	41
<i>Obr. 6. Vzhled úvodní strany modifikované verze</i> .....	41
<i>Obr. 7. Vzhled původní stránky subsystému</i> .....	42
<i>Obr. 8. Vzhled stránky subsystému modifikované verze</i> .....	42
<i>Obr. 9. Vzhled původní stránky základního submodulu</i> .....	43
<i>Obr. 10. Vzhled stránky základního submodulu modifikované verze</i> .....	43

**SEZNAM TABULEK**

<i>Tab. 1. Příklad seznamu typu 1.....</i>	17
<i>Tab. 2. Příklad seznamu typu 2.....</i>	18
<i>Tab. 3. Číselné datové typy v MySQL.....</i>	20
<i>Tab. 4. Textové datové typy v MySQL.....</i>	21
<i>Tab. 5. Datum a čas v MYSQL.....</i>	21