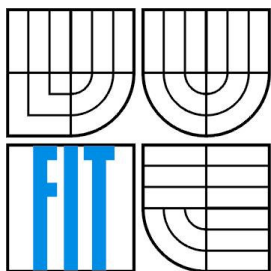


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

UDRŽOVATELNÁ VERZE INFORMAČNÍHO SYSTÉMU AKADEMICKÉHO CENTRA

MAINTAINABLE VERSION OF THE ACSA INFORMATION SYSTEM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Michal Šramka

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Ivana Rudolfová

BRNO 2008

Zadání diplomové práce

Řešitel: **Šramka Michal, Bc.**

Obor: Informační systémy

Téma: **Udržitelná verze informačního systému Akademického centra**

Kategorie: Softwarové inženýrství

Pokyny:

1. Seznamte se s problematikou udržitelnosti software a existujícími technikami jejího zajištění.
2. Seznamte se s architektonickými vzory a rámci vhodnými pro implementaci středně velkých informačních systémů.
3. Analyzujte stávající moduly informačního systému ACSA s cílem identifikovat jejich slabá místa z pohledu udržitelnosti.
4. Navrhněte vhodnou architekturu pro novou verzi informačního systému ACSA. Hlavním cílem je zvýšení udržitelnosti oproti stávající verzi.
5. Po dohodě s vedoucím diplomové práce implementujte jádro a vybrané moduly nové verze informačního systému (MySQL, PHP).
6. Funkčnost ověřte na reálném vzorku dat.
7. Zhodnoťte výsledky a diskutujte možné pokračování projektu.

Literatura:

- o Maciaszek, L.: Practical Software Engineering: A Case Study Approach. PEARSON, 2005. ISBN 0321204654
- o Fowler, M.: Refactoring - zlepšení existujícího kódu, Grada, 2003, ISBN 80-247-0299-1

Při obhajobě semestrální části diplomového projektu je požadováno:

- o 1 až 4.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVR-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Rudolfová Ivana, Ing., UIFS FIT VUT**

Konzultant: Švec Jaroslav, Ing.

Datum zadání: 24. září 2007

Datum odevzdání: 19. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2



doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Bc. Michal Šramka**

Id studenta: 89872

Bytem: Dvoutletky 706, 285 22 Zruč nad Sázavou

Narozen: 15. 11. 1982, Kutná Hora

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií

se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....

(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
diplomová práce

Název VŠKP: Udržovatelná verze informačního systému Akademického centra

Vedoucí/školicel VŠKP: Rudolfová Ivana, Ing.

Ústav: Ústav informačních systémů

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1

elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ☐ ihned po uzavření této smlouvy
 - ☐ 1 rok po uzavření této smlouvy
 - ☐ 3 roky po uzavření této smlouvy
 - ☐ 5 let po uzavření této smlouvy
 - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel

.....
Autor

Abstrakt

Cílem této diplomové práce je vytvořit základ nového informačního systému Akademického centra studentských aktivit. Nový informační systém Akademického centra by měl být dobře dokumentovaný a udržitelný. V této práci jde především o návrh jádra systému a několika modulů a jejich následnou implementaci.

První část projektu se zabývá otázkami udržitelnosti softwarových produktů a návrhem jejich architektury. Ve druhé části je provedena analýza současného informačního systému Akademického centra studentských aktivit a vytvořen návrh architektury pro systém nový. Dále je proveden návrh jádra a vybraných modulů. Následuje implementace navržených částí a jejich otestování na reálných datech.

Klíčová slova

Udržitelnost software, architektura software, architektonické rámce, architektonické vzory, PHP, MySQL, Akademické centrum studentských aktivit, ACSA, model-pohled-řadič, MVC.

Abstract

The goal of this master's thesis is creating a concept of architecture for new information system of the Academic Center for Students Activities. New information system of the Academic Center should be well documented and better maintainable.

The first part deals with software maintenance and software architecture design. The second part contains analysis of current information system of Academic Center for Students Activities and also the architecture of new system is designed. The next parts are design of framework for the new system and also design of particular modules. Then there are implementation of designed parts and testing on real data.

Keywords

Software maintenace, software architectures, architectural frameworks, architectural patterns, PHP, MySQL, Academic Center for Student Activities, ACSA, model-view-controller, MVC.

Citace

Šramka Michal: Udržitelná verze informačního systému Akademického centra. Brno, 2008, diplomová práce, FIT VUT v Brně.

Udržitelná verze informačního systému Akademického centra

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Ivany Rudolfové. Další informace mi poskytl externí konzultant Ing. Jaroslav Švec. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jméno Příjmení
19. 5. 2008

Poděkování

Děkuji vedoucí práce Ing. Ivaně Rudolfové a externímu konzultantovi Ing. Jaroslavu Šveci za profesionální vedení, rady a mnoho důležitých podnětů, které mi v průběhu zpracování práce poskytovali.

© Michal Šramka, 2008.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	7
1 Úvod.....	10
2 Udržovatelnost software	11
2.1 Procesy údržby software dle ISO/IEC 14764.....	11
2.2 Údržba software z pohledu životního cyklu.....	11
2.3 Různé typy údržby software.....	12
2.3.1 Opravná údržba.....	12
2.3.2 Adaptivní údržba.....	12
2.3.3 Zlepšovací údržba	13
2.3.4 Preventivní údržba	13
2.4 Modely provádění údržby	13
2.5 Problémy při provádění údržby	14
2.5.1 Některé prostředky eliminace problémů při údržbě.....	15
2.6 Refaktoring.....	15
3 Architektura software, rámce a vzory	17
3.1 Architektura software.....	17
3.2 Architektonické rámce	18
3.2.1 Model-View-Controller	18
3.2.2 Presentation-Control-Mediator-Entity-Foundation.....	19
3.2.3 Další rámce	20
3.3 Architektonické vzory	20
4 Analýza současného IS ACSA.....	21
4.1 Základní informace o ACSA.....	21
4.1.1 Hlavní projekty ACSA.....	21
4.1.2 Organizace působící pod záštitou ACSA.....	21
4.1.3 Využití IS ACSA	21
4.2 Základní informace o IS ACSA	22
4.2.1 Popis jednotlivých součástí.....	22
4.2.2 Technické informace.....	23
4.3 Podrobná analýza IS ACSA	23
4.3.1 Dokumentace a zdrojové kódy.....	23
4.3.2 Rozdělení uživatelů.....	24
4.3.3 Rozhraní systému.....	25
4.3.4 Modul Správa uživatelů	26

4.3.5	Modul Překlady.....	27
4.3.6	Modul Správa webové prezentace	28
4.3.7	Modul Akce	29
4.3.8	Portál studentských organizací.....	29
4.3.9	Modul Redakční a publikační systém	30
4.3.10	Celkové zhodnocení IS ACSA	31
5	Návrh architektury nového IS ACSA	33
5.1	Požadavky kladené na novou architekturu	33
5.2	Výběr vhodného architektonického rámce.....	33
5.2.1	MVC z pohledu IS ACSA.....	34
5.3	Privátní a veřejná část systému	35
5.4	Integrace původních modulů IS ACSA.....	35
5.5	Návrh jádra nového IS ACSA	35
5.5.1	Komponenta Model	36
5.5.2	Komponenta View	38
5.5.3	Komponenta Controller	39
5.6	Moduly nového IS ACSA	39
5.6.1	Datová komunikace mezi moduly.....	40
5.6.2	Návrh modulu pro nový IS ACSA	40
5.6.3	Modul Správa informačního systému	42
5.6.4	Modul Správa uživatelů	42
5.6.5	Modul Správa webové prezentace	43
5.6.6	Modul Redakční a publikační systém	44
5.6.7	Modul Překlady.....	45
5.7	Uživatelské rozhraní nového IS ACSA.....	46
5.8	Pravidla pro psaní zdrojového kódu a dokumentace	47
6	Implementace	48
6.1	Plán implementace nového IS ACSA	48
6.1.1	První fáze	48
6.1.2	Druhá fáze.....	48
6.1.3	Třetí fáze	48
6.1.4	Další fáze	48
6.2	Adresářová struktura nového IS ACSA	49
6.3	Implementace jádra	50
6.3.1	Model	50
6.3.2	View	51
6.3.3	Controller	52

6.4	Implementace modulů	52
6.4.1	Modul Správa informačního systému	52
6.4.2	Modul Správa uživatelů	53
6.4.3	Modul Překlady.....	53
6.4.4	Modul Správa webové prezentace	55
6.4.5	Modul Redakční a publikační systém	55
7	Ověření systému na reálných datech.....	57
7.1	Úprava dat původního systému	57
7.2	Testování nového systému	57
8	Možná rozšíření	59
8.1	Nové uživatelské rozhraní	59
8.2	Překlad dynamických textů	59
8.3	Redakční systém založený na AJAXu.....	60
8.4	Automatizované komponenty.....	60
8.5	Personalizace systému.....	60
9	Závěr	61
	Seznam obrázků.....	62
	Literatura	63
	Přílohy	65
1	Společná část databáze.....	65
2	Jazykové soubory	66
2.1	Formát původního jazykového souboru	66
2.2	Formát nového jazykového souboru	66
3	Zdrojové kódy	67
4	Databázové SQL skripty	68
5	Dokumentace IS ACSA	69
6	Testovací verze	70

1 Úvod

Udržitelná verze informačního systému Akademického centra je projekt, jehož hlavním cílem je vytvoření nové verze informačního systému Akademického centra¹, který bude ve srovnání se současným systémem především lépe udržitelný a rozšiřitelný.

Největší motivací pro výběr tohoto tématu byl můj zájem o internetové technologie a vývoj informačních systémů. Dalším důvodem byla má předchozí spolupráce s Akademickým centrem při zpracování bakalářské práce, která pokračovala i po jejím skončení. A v neposlední řadě podpořila mé rozhodnutí i potřeba vytvoření nového IS ACSA.

Akademické centrum studentských aktivit² je organizace působící na Vysokém učení technickém v Brně, která poskytuje podporu studentům v oblastech vysokoškolské legislativy, projektového řízení, komunikace a při řešení různých problémů spojených s aktivní činností v akademických samosprávách vysokých škol. Více o činnosti ACSA lze nalézt na webových stránkách³ organizace. Stručný popis činností ACSA je uveden také v kapitole 4.1 této práce.

Tato diplomová práce navazuje na semestrální projekt [16] a rozšiřuje jej především v oblastech detailního návrhu nového IS ACSA a implementace některých modulů. První část diplomové práce se zabývá teoretickými otázkami. Ve druhé části je provedena analýza současného IS ACSA, návrh systému nového a jeho implementace.

Kapitola 2. obsahuje informace týkající se udržitelnosti software. Náplní 3. kapitoly jsou některé architektonické rámce a vzory, které jsou používány při návrhu informačních systémů. Ve 4. kapitole je uvedena podrobná analýza současného IS ACSA. Kapitola 5. obsahuje návrh jádra a modulů nového systému. V 6. kapitole jsou uvedeny podrobnosti k implementaci. Kapitola 7. zmiňuje úpravy dat potřebné k přechodu od původní verze k verzi nové. Předposlední 8. kapitola se zabývá možnostmi rozšíření a pokračování ve vývoji nového IS ACSA. Dále následuje závěr a několik příloh.

¹ informační systém Akademického centra – dále jen zkratka IS ACSA

² Akademické centrum studentských aktivit – dále jen zkratka ACSA

³ WWW stránky ACSA: <http://www.acsa.vutbr.cz/>

2 Udržovatelnost software

Stejně tak jako všechny ostatní produkty, tak i software potřebuje v průběhu svého používání určitým způsobem udržovat. Na rozdíl od klasických produktů (např.: automobil – opotřebovaná součástka) zahrnuje údržba software i další činnosti, které nesouvisí přímo s opravami. V této kapitole jsou rozebrány činnosti týkající se údržby software a nastíněna některá rizika s ní související. Při psaní této kapitoly čerpal autor informace ze zdrojů [1], [2], [3], [4] a [5].

2.1 Procesy údržby software dle ISO/IEC 14764

Mezinárodní standard ISO/IEC 14764 definuje údržbu software jako modifikaci softwarového produktu po jeho dodání za účelem opravy chyb, zlepšení výkonu a dalších vlastností a adaptace produktu v novém prostředí. Standard definuje šest procesů při údržbě software:

1. Implementační proces, který řeší vytvoření koncepce a plánu údržby.
2. Proces analýzy problémů a změn, který nastává při nasazení systému do provozu.
3. Proces zhodnocení dopadů implementace změn samotných.
4. Proces ověření správnosti provedených změn v součinnosti s tím, kdo si změny vyžádal.
5. Proces migrace do nového prostředí.
6. Vyřazení softwarového produktu nebo jeho části z provozu.

2.2 Údržba software z pohledu životního cyklu

Životní cyklus softwarového produktu má několik fází. Jejich počet a názvy se mohou lišit v závislosti na použitém modelu. V souladu s [1] lze uvažovat následující fáze:

- analýza požadavků,
- návrh systému,
- implementace,
- integrace a nasazení,
- provoz a údržba.

Fáze pokrývající provoz a údržbu představuje poslední a nejdelší fázi životního cyklu. V této fázi je produkt aktivně používán v reálném prostředí. Při provozu systému vznikají požadavky na údržbu systému. Jedná se o opravy chyb, přizpůsobení systému změnám okolního prostředí a reakce na nové požadavky uživatelů.

Fáze životního cyklu předcházející fázi provoz a údržba musí být realizovány tak, aby následné provádění údržby bylo co nejjednodušeji uskutečnitelné. Při analýze požadavků je nutné zaměřit se na

důkladné zjištění všech požadavků tak, aby následných změn po nasazení do provozu bylo co nejméně. Při návrhu systému je nutné brát ohled na co nejjednodušší provádění změn v navrhovaném systému. Výsledkem implementace by měl být srozumitelný a strukturovaný kód podpořený kvalitní dokumentací, která umožní snadné provádění změn.

Z pohledu nákladů na vývoj systému je údržba velmi významnou položkou. Platí, že čím starší systém je, tím jsou náklady na jeho údržbu vyšší. Hlavním důvodem je postupné rozrůstání systému často způsobené průběžnou údržbou, což zvyšuje složitost systému a narušuje jeho původní architekturu. Příkladem údržby, na kterou mnoho organizací vynaložilo nemalé prostředky, je například problém roku 2000⁴.

2.3 Různé typy údržby software

Údržba software se podle ISO/IEC 14764 dělí do čtyř skupin:

- opravná,
- adaptivní,
- zlepšovací,
- preventivní.

2.3.1 Opravná údržba

Tento typ údržby se zabývá opravou chyb nalezených při provozu systému. Může se jednat o chyby:

- chyby logické,
- chyby v návrhu,
- chyby v kódu.

Logické chyby vznikají například při špatně provedených testech a nesprávné implementaci návrhu. Chyby návrhu pramení ze špatně pochopených požadavků. Chyby v kódu bývají způsobeny špatnou implementací detailů návrhu. Další chyby mohou být také způsobeny zpracováním dat a výkonem systému. Chyby degradují požadovanou funkčnost systému a jsou obvykle reportovány koncovými uživateli.

2.3.2 Adaptivní údržba

Adaptivní údržba reaguje na změny okolního prostředí systému. Udržuje systém aktuální a funkční v souladu s novými požadavky přicházejícími z okolí systému. Požadavky na adaptaci systému mohou vznikat například z těchto důvodů (výčet není zcela jistě úplný):

- změna hardware,

⁴ anglicky „Y2K problem (bug)“, více informací lze nalézt např. v [17]

- změna legislativy,
- změna způsobu provádění určité činnosti.

Požadavky na tento druh údržby vznikají pouze v případě, že je okolí systému monitorováno. Příkladem může být účetní systém – i bez reakce na změnu legislativy týkající se účetnictví bude systém stále fungovat, problém se projeví až při komunikaci s finančním úřadem. Proto je nutné sledovat změny okolí systému a promítat je do jeho fungování.

2.3.3 Zlepšovací údržba

Tento typ reaguje především na nové požadavky uživatelů systému. Jedná se o funkční zlepšování systému, které přináší zákazníkovi určitou přidanou hodnotu. Může se jednat například o zlepšení výkonu a urychlení některých operací nebo o vytvoření přívětivějšího uživatelského rozhraní. Při použití iterativního životního cyklu lze chápat další iterace jako procesy zlepšovací údržby.

2.3.4 Preventivní údržba

Preventivní údržba je prováděna z důvodu jednoduššího provádění ostatních typů údržby. Nerozšiřuje funkčnost systému, nezabývá se opravou chyb a ani adaptací systému. Jedná se především o následující činnosti:

- aktualizace dokumentace,
- přidávání dalších komentářů,
- zlepšování struktury systému a kódu.

Ostatní typy údržby degradují původní systém – zvyšují jeho složitost a narušují původní strukturu. Preventivní údržba redukuje tento efekt a je prováděna v zájmu toho, kdo systém udržuje. Snahou je vytvoření srozumitelnějšího programu, který je možno lépe udržovat.

2.4 Modely provádění údržby

Modely provádění údržby definují, jakým způsobem je údržba prováděna. Nejčastěji používanými modely jsou:

- Model rychlých oprav (Quick Fix model),
- Boehmův model (Boehm's model),
- Osbornův model (Osborn's model),
- Iterativní zlepšovací model,
- Model orientovaný na znovupoužití (Reuse-oriented model).

Model rychlých oprav představuje ad-hoc přístup. Soustředí se na rychlé vyhledání chyby a její opravení. Nezabývá se dlouhodobým dopadem této změny na systém. Výhodou je rychlá reakce na objevené chyby. Jedná se o tradiční model, který se nezabývá dalším rozvojem systému. Je vhodný

například v případech, kdy údržbu provádí člověk, který má dostatečné znalosti o systému a nemusí se zabývat studiem a tvorbou detailní dokumentace.

Boehmův model je založen na ekonomických modelech a principech. Použití těchto modelů pomáhá lépe porozumět problému a zvýšit produktivitu údržby.

Osbornův model definuje při opravě chyb čtyři hlavní úkoly. Nejprve je nutné zahrnout požadavky na údržbu do specifikace změn. Dále je nutné specifikovat požadavky na kvalitu. Potom je potřeba stanovit metriky, podle kterých je možné rozhodnout, jestli byly cíle údržby splněny. Posledním úkolem je zajištění zpětné vazby s manažery kvůli zhodnocení výsledku.

Iterativní model byl vytvořen na základě iterativního životního cyklu. Model má tři fáze: analýza, návrh a hodnocení změn a implementace. Model předpokládá úplnou dokumentaci systému.

Model orientovaný na znovupoužití předpokládá existenci komponent, které mohou být opětovně použity. Postup je následující: nejprve jsou ve starém systému vyhledány komponenty vhodné pro znovupoužití, následně provedena jejich modifikace v souladu s novými požadavky a nakonec jsou komponenty integrovány do nového systému.

2.5 Problémy při provádění údržby

Při údržbě software může nastat velké množství různých problémů. Některé z nich mají kořeny v předcházejících fázích životního cyklu, jiné jsou způsobeny při vlastní údržbě systému. Největším problémem je, když se systém dostane do stavu, ve kterém ho již nelze dále udržovat.

Častými problémy při údržbě jsou:

- špatná dokumentace,
- provádění „ad-hoc“ změn,
- nedokumentované změny,
- velikost softwarového systému,
- stárnutí softwarového systému.

Pro správné provádění údržby software je nutné jeho dobré pochopení. Pro pochopení fungování software je velmi často potřebná kvalitní dokumentace, která tuto činnost výrazně urychlí. V případě, že vývoj i údržbu provádí jedna a tatáž osoba, dochází často k zanedbávání dokumentační činnosti. Tento fakt může způsobit značné problémy v případě, že má systém udržovat někdo jiný. Bez kvalitní dokumentace je nutné provádět zpětné inženýrství (snaha o pochopení fungování systému analyzováním jeho kódu), což může být značně náročné a nákladné.

Při provádění změn v systému je vhodné uvažovat dopad změn v celém systému a neprovádět změny „ad-hoc“. Tyto změny často narušují původní architekturu systému a postupem času znemožňují efektivní provádění další údržby.

Po provedení jakékoli změny je nutné tuto změnu zdokumentovat, aby v budoucnu bylo možné zjistit, co přesně se v systému změnilo a proč se systém chová daným způsobem.

Platí, že s rostoucí velikostí softwarového produktu se stává jeho údržba obtížnější. Hlavním důvodem je určení dopadů prováděných změn na ostatní části systému.

Stárí software hraje při údržbě také velkou roli. I kvalitně navržený software může být po určité době provozu a údržby dále již jen obtížně udržovatelný. Bývá to způsobeno především změnami, které byly provedeny v průběhu používání systému – tj. samotnou údržbou.

2.5.1 Některé prostředky eliminace problémů při údržbě

Mezi vhodné prostředky zajišťující efektivní údržbu patří například:

- volba vhodného modelu životního cyklu a modelu pro údržbu software,
- kvalitní návrh architektury systému,
- psaní strukturovaného a dokumentovaného kódu,
- tvorba podrobné dokumentace a dokumentace změn,
- plánování údržby (neprovádění změn ad-hoc).

2.6 Refaktoring

Refaktoring reprezentuje proces provádění změn v softwarovém systému takovým způsobem, že nemají vliv na jeho vnější chování, ale zlepšují vnitřní strukturu programového kódu systému. Tento koncept se poprvé objevil ve Smalltalku a začal se prosazovat až s obecným rozšířením objektově orientovaného programování. V dnešní době již existují osvědčené postupy refaktORIZACE jejichž popis lze nalézt například v [5]. Z pohledu údržby software představuje refaktoring proces preventivní údržby software.

Důvody pro používání refaktoringu mohou být různé. Například refaktoring může

- zlepšit původní návrh,
- vést ke zlepšení srozumitelnosti,
- pomoci při hledání chyb,
- zrychlit další vývoj softwaru.

Dalším důležitým krokem při refaktoringu je určit čas, kdy začít refaktorovat. Je možné řídit se například následujícími pravidly:

- pravidlo „Do třetice“ – v případě, že by měl být stejný kód napsán potřetí,
- před přidáním další funkcionality,
- když je nutné opravit chybu,
- při provádění revize kódu.

Refaktoring jistě neřeší všechny problémy spojené se špatným kódem programu, ale určitě představuje užitečnou techniku při jeho zlepšování. Existují momenty, kdy refaktoring není vhodný.

Důležitou podmínkou je, že refaktorovaný systém by neměl obsahovat vážnější chyby. V opačném případě může být mnohdy výhodnější celý systém nebo jeho část přeprogramovat.

3 Architektura software, rámce a vzory

Navržení vhodné architektury patří k nejobtížnějším a zároveň nejdůležitějším činnostem při vývoji softwarového systému. V této kapitole je uveden přehled některých používaných architektur, rámců a vzorů. Informace k těmto tématům lze nalézt v [1], [6], [11], [12], [13], [14] a [15].

3.1 Architektura software

Architektura je základem, ze kterého vychází všechny součásti výsledného softwarového systému. Definuje návrh systému na nejvyšší úrovni a řeší strategická rozhodnutí. Určuje strukturu systému složenou z jeho jednotlivých částí, jejich externích vlastností a vztahů mezi těmito částmi. Návrh architektury vychází především z následujících informací:

- nefunkční požadavky,
- závislosti součástí systému,
- strategická rozhodnutí týkající se celého systému.

Architektura systému má zásadní vliv na celkové vlastnosti systému, jako jsou:

- výkon,
- bezpečnost,
- dostupnost,
- udržitelnost,
- škálovatelnost.

Existují různé pohledy na architekturu software. UML zachycuje strategické aspekty architektury systému pohledem 4 + 1, který definuje:

- logický pohled,
- pohled implementace,
- pohled procesů,
- pohled nasazení,
- a pátým zastřešujícím je pohled případů použití.

Mezi běžně používané typy architektur patří:

- klient/server,
- třívrstvá architektura,
- vícevrstvá architektura,
- architektura orientovaná na služby,
- architektura bod-bod (peer-to-peer),
- a mnoho dalších problémově orientovaných architektur.

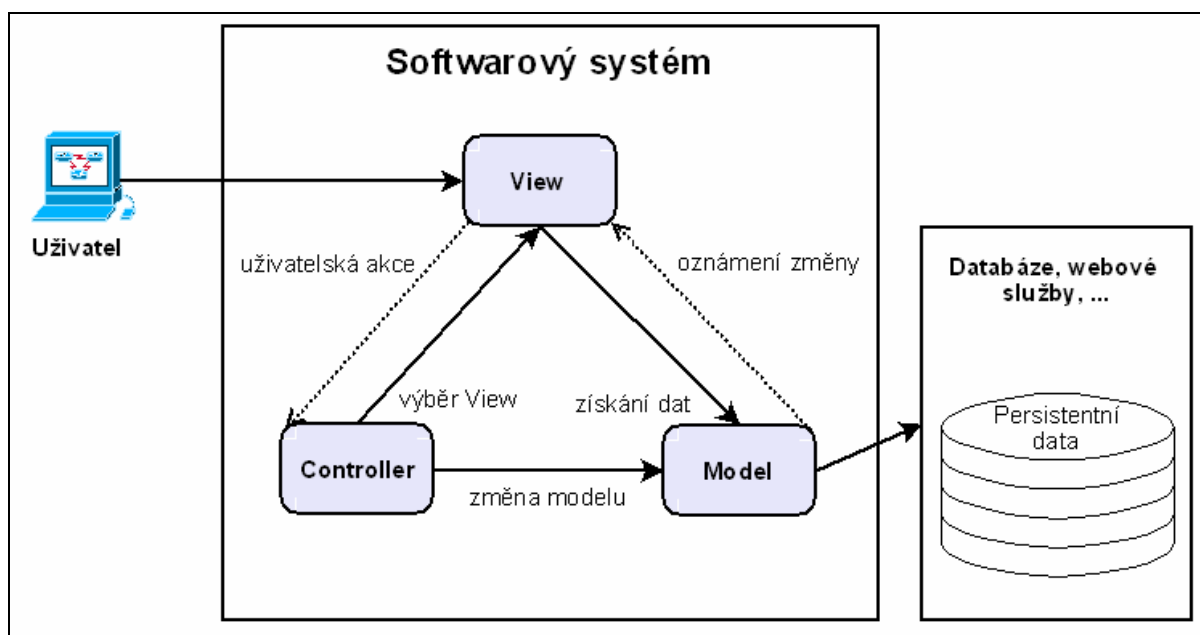
3.2 Architektonické rámce

Architektonický rámec určuje konstrukce, pomocí nichž je tvořeno řešení problému. Poskytuje kostru pro řešení problému, kterou je možné vhodně upravit a rozšířit v mezích stanovených daným rámcem. Jedná se o technologii znovupoužití týkající se návrhu systému. Rámec určuje architektonický návrh systému. Může se například jednat o stanovení způsobu realizace určitých činností anebo třeba o hotový softwarový systém umožňující vývoj pomocí připravených komponent.

3.2.1 Model-View-Controller⁵

Jedná se pravděpodobně o nejznámější architektonický rámec. Byl vyvinut v rámci programovacího jazyka Smalltalk-80 a rozšířil se i do dalších objektově orientovaných jazyků. Jeho hlavní myšlenkou je oddělení zodpovědnosti (zájmů) jednotlivých částí. Programátor musí rozdělit třídy aplikace do tří skupin: model (data), view (pohled) a controller (řadič, řízení činnosti).

Model reprezentuje datové objekty dané domény a jejich obsluhu. View prezentuje stav modelu ve formátu požadovaném uživatelem. Controller řídí zpracování interaktivních požadavků a může vyvolat změnu modelu. Situaci ilustruje Obrázek 1.



Obrázek 1: Schéma rámce MVC

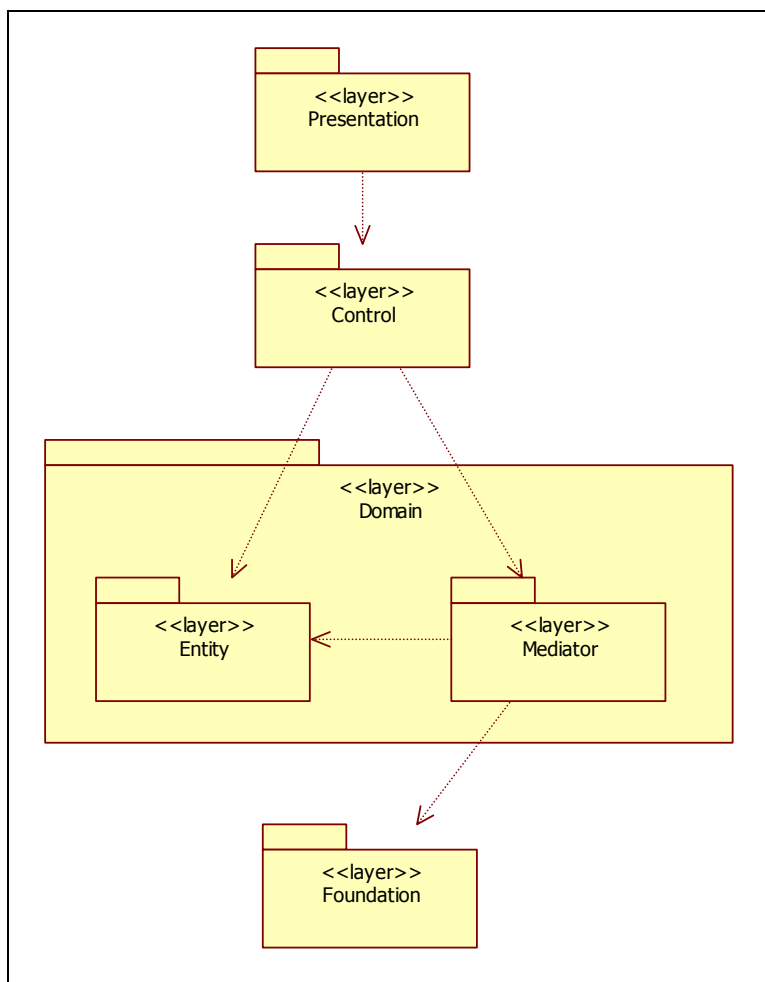
Mezi výhody MVC, které přináší rozdělení zájmů jednotlivých částí, patří například:

- možnost samostatného vývoje uživatelského rozhraní, řídicí logiky a logiky dat,
- změny v jednotlivých částech neovlivňují ostatní části,
- možnost použití modelu s různými uživatelskými rozhraními (nebo bez uživatelského rozhraní – vhodné např. pro testování).

⁵ Dále jen zkratka MVC.

3.2.2 Presentation-Control-Mediator-Entity-Foundation⁶

PCMEF je představitelem rámce založeného na rozdělení zodpovědnosti do vrstev. Přináší hierarchické vertikální členění vrstev. Každá vrstva může obsahovat další balíčky a může být vložena v jiné vrstvě. Rámec PCMEF obsahuje čtyři hlavní vrstvy, přičemž vrstva Domain spojuje vrstvy Entity a Mediator, viz Obrázek 2. Uživatel komunikuje s vrstvou Presentation. Přístup k databázi zajišťuje vrstva Foundation. Více informací o PCMEF lze nalézt v [1].



Obrázek 2: Schéma rámce PCMEF

Hlavními principy PCMEF jsou:

- závislost vyšší vrstvy na nižší,
- informace o změnách ve směru od nižší vrstvy k vyšší,
- komunikace pouze mezi sousedními vrstvami,
- eliminace kruhových závislostí,
- pojmenování tříd podle příslušnosti k vrstvě,
- komunikace nesousedních vrstev pomocí rozhraní.

⁶ Dále jen zkratka PCMEF.

3.2.3 Další rámce

Různých architektonických rámců je k dispozici velké množství a jejich popis je možné najít v uvedených zdrojích, především pak v [6] a [11]. Jedná se například o následující:

- Broker,
- Presentation-Abstraction-Control,
- Pipes and filters,
- Peer-to-peer.

3.3 Architektonické vzory

Při návrhu architektury (a nejen jí) je vhodné používat osvědčené techniky řešení obvyklých problémů. Návrhové vzory představují soubor nejlepších metod pro řešení známých a běžně se opakujících problémů. Zajišťují lepší srozumitelnost a udržitelnost programu.

Mezi architektonické návrhové vzory lze zařadit následující vzory:

- Fasáda,
- Abstraktní továrna,
- Řetěz odpovědnosti,
- Pozorovatel,
- Prostředník.

Fasáda (anglicky „Facade“) řeší problém společného rozhraní nesourodých subsystémů. Definuje bod kontaktu se systémem, který je zodpovědný za komunikaci s jednotlivými subsystémy. Umožňuje soustředit závislosti mezi komunikujícími systémy do jednoho místa.

Abstraktní továrna (anglicky „Abstract Factory“) řeší problém jak vytvořit rodiny souvisejících tříd, aniž by bylo předem nutné specifikovat konkrétní třídu.

Řetěz odpovědnosti (anglicky „Chain of Responsibility“) řeší problém spojení odesílatele žádosti s příjemcem tím, že zřetězí objekty příjemců a prostřednictvím řetězu předává žádost až do objektu, který ji zpracuje.

Pozorovatel (anglicky „Observer“) řeší problém jak informovat všechny závislé objekty o tom, že objekt změnil svůj stav.

Prostředník (anglicky „Mediator“) zapouzdřuje interakci sady objektů a umožňuje jejich volné spojení tím, že brání vytvářet explicitní vzájemné odkazy.

Podrobný popis nejen těchto vzorů je obsažen v [15] a [18].

4 Analýza současného IS ACSA

Od svého vzniku v roce 2005 poskytuje IS ACSA podporu činnostem ACSA. Od té doby se značně rozrostl a získal velké množství uživatelů. V současné době je registrováno přibližně 1000 uživatelů a na 70 studentských organizací. Cílem této kapitoly je důkladná analýza IS ACSA s cílem určení silných a odhalení slabých míst tohoto informačního systému.

4.1 Základní informace o ACSA

ACSA je organizace působící na Vysokém učení technickém v Brně, která se věnuje podpoře v různých oblastech života studentů vysokých škol.

Jedním z hlavních cílů ACSA je už od založení v roce 2002 rozvíjet kvalitu akademické samosprávy, zejména v oblastech dotýkajících se studentů. Mezi další cíle patří systematizace studentského hodnocení kvality vysokých škol a pomoc studentům v tísní. Od roku 2002 se ACSA značně rozrostlo a dnes se věnuje mnoha dalším projektům. Pod záštitou ACSA působí také několik dalších organizací a pracovních skupin.

4.1.1 Hlavní projekty ACSA

Mezi hlavní projekty patří:

- pořádání celostátní studentské konference,
- pořádání seminářů, konferencí a workshopů,
- provoz Portálu studentských organizací,
- projekt Zelená univerzita (ZUni),
- zavádění projektového řízení na vysoké školy (PŘoVŠe),
- projekt Národní šetření studentů (Na6).

4.1.2 Organizace působící pod záštitou ACSA

Pod záštitou ACSA působí:

- YPMG – Skupina mladých projektových manažerů,
- BEST – Rada evropských studentů techniky (Board of European Students of Technology).

4.1.3 Využití IS ACSA

Systém slouží k prezentaci ACSA a obou zde působících organizací. Dále je využíván pro podporu všech hlavních projektů, kterými se ACSA zabývá. Například přihlašování na akce, stránky studentských organizací nebo dotazník Na6.

4.2 Základní informace o IS ACSA

IS ACSA je přístupný pomocí internetového prohlížeče a prezentuje informace o ACSA a jeho projektech. Pro registrované uživatele je IS ACSA vstupní branou na všechny pořádané akce a umožňuje jim také registraci do studentských organizací, případně registraci organizací samotných. Oprávnění uživatelé mohou měnit a vytvářet informace prezentované systémem – jedná se především o zaměstnance ACSA, správce jednotlivých organizací a některé další vybrané uživatele systému.

4.2.1 Popis jednotlivých součástí

IS ACSA je složen ze šesti hlavních modulů, jimiž jsou:

- modul Akce,
- modul Správa webové prezentace,
- modul Překlady,
- modul Portál studentských organizací,
- modul Redakční a publikační systém,
- modul Správa uživatelů.

Modul Akcí shromažďuje informace o akcích pořádaných ACSA. Uživatelům systému umožňuje přihlašování na akce, eviduje platby, vystavuje certifikáty za úspěšně absolvované akce. Zaměstnancům ACSA poskytuje přehled o pořádaných akcích, přihlášených uživateli a umožňuje provádět správu těchto akcí.

Modul Správa webové prezentace umožňuje zaměstnancům ACSA měnit webovou prezentaci přímo v prostředí internetového prohlížeče pomocí jednoduchého editoru.

Modul Překladů slouží pro vytváření překladů statických textů ostatních modulů. Správci modulu definují zdroje s uloženými texty. Oprávnění uživatelé (překladaři) mohou vkládat překlady jednotlivých textových frází.

Portál studentských organizací poskytuje studentským organizacím prostor pro vlastní prezentaci a nabízí jim využití některých funkcí Redakčního a publikačního systému. Studentská organizace se může zaregistrovat a vytvořit v systému vlastní webovou prezentaci, ve které může využívat některých služeb poskytovaných Redakčním a publikačním systémem. Uživatelé IS ACSA mohou správce dané organizace požádat o přidělení členství v organizaci.

Modul Redakčního a publikačního systému seskupuje dohromady několik modulů, jsou to:

- Ankety,
- Aktuality,
- Články,
- Emaily,
- Soubory ke stažení,

- Galerie fotografií,
- Diskusní fóra,
- Dotazníková šetření.

Činnost jednotlivých modulů je zřejmá z jejich názvu. Všechny tyto moduly umožňují oprávněným uživatelům vytvářet a měnit konkrétní data. Ostatním uživatelům pak tyto data zpřístupňují.

Modulu Správa uživatelů poskytuje ostatním modulům informace o uživatelích. Každému uživateli IS ACSA je umožněno měnit informace patřící k jeho účtu. Správce tohoto modulu, kromě editace dat uživatelů, také nastavuje práva přístupu uživatelů k jednotlivým modulům IS ACSA.

4.2.2 Technické informace

Technologie použité pro vytvoření a provoz systému:

- PHP 4,
- původně MySQL 3.23, nyní MySQL 4,
- původně Apache 1.3, nyní Apache 2.0,
- HTML,
- CSS,
- JavaScript.

Práce se systémem probíhá v prostředí internetového prohlížeče. Kompatibilita je zajištěna pouze s prohlížeči Internet Explorer a Mozilla Firefox. U některých ostatních prohlížečů dochází k větší či menší degradaci zobrazení poskytovaných informací.

4.3 Podrobná analýza IS ACSA

IS ACSA byl vytvořen v roce 2005 týmem skládajícím se převážně ze studentů Fakulty informačních technologií VUT. Jeden z členů týmu vytvořil rozhraní systému a ostatní členové pracovali na přidělených modulech. Někteří ze studentů pracovali na vývoji IS ACSA v rámci svých diplomových nebo bakalářských prací. V první fázi vzniklo rozhraní systému, modul Správa uživatelů, modul Překladů, modul Akcí, Portál studentských organizací a nepříliš úspěšně byl realizován i Redakční a publikační systém. V roce 2006 byl IS ACSA rozšířen o fungující Redakční a publikační systém a modul Správa webové prezentace. V roce 2007 probíhaly pouze úpravy stávajících modulů a běžná údržba.

4.3.1 Dokumentace a zdrojové kódy

Z pohledu údržby IS ACSA je dokumentace jednou z jeho největších slabin. K modulům vytvářeným v rámci diplomových nebo bakalářských prací existuje dokumentace v podobě těchto prací. Jedná se o následující moduly:

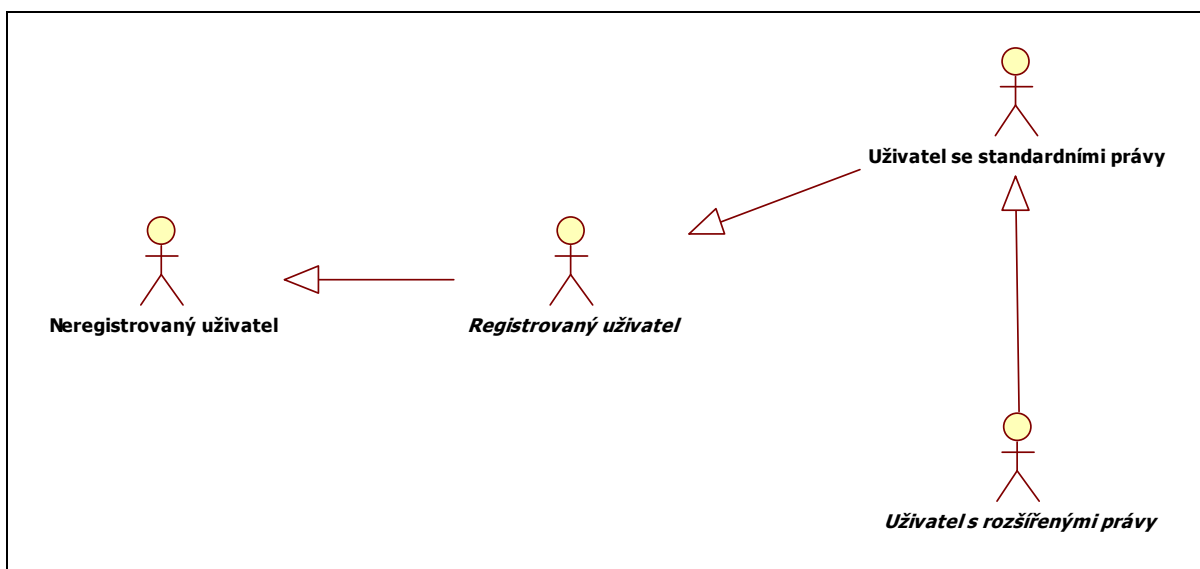
- Rozhraní systému – bakalářská práce [7],
- Redakční a publikační systém – bakalářské práce [8] a [9],
- Portál studentských organizací – diplomová práce [10],

Problémem však je, že tyto práce dokumentují systém v době, kdy byly dokončeny, a nereflektují následně provedené změny a úpravy. Těchto změn a úprav proběhlo velké množství a v současné době již tyto práce nejsou pro pochopení fungování systému dostatečným zdrojem. K ostatním modulům systému neexistuje dokumentace žádná.

Vzhledem k neexistenci aktuální dokumentace je nutné procházet zdrojový kód systému a snažit se o pochopení jednotlivých pasáží. Tuto činnost by značně urychlila a zjednodušila přítomnost vhodných komentářů ve zdrojových textech. Ani tato forma dokumentace však nemá valnou kvalitu a komentáře jsou přítomny velmi zřídka. Kromě několika výjimek, kdy je použito objektově orientovaného přístupu, jsou zdrojové kódy psány klasickým procedurálním přístupem. Kód prezentující data je psán společně s kódem, který s daty pracuje. Strukturovanost kódu se liší modul od modulu v závislosti na konkrétním autorovi a jeho přístupu.

4.3.2 Rozdělení uživatelů

Uživatelé (viz Obrázek 3) IS ACSA jsou rozděleni do dvou hlavních skupin. Jedná se neregistrované a registrované uživatele. Registrovaní uživatelé mohou přistupovat do privátní části, kde mají nastavena přístupová práva k jednotlivým modulům. Z tohoto pohledu lze registrované uživatele rozdělit na uživatele se standardními právy a uživatele s právy rozšířenými. Zaměstnanci ACSA patří mezi registrované uživatele s rozšířenými právy.

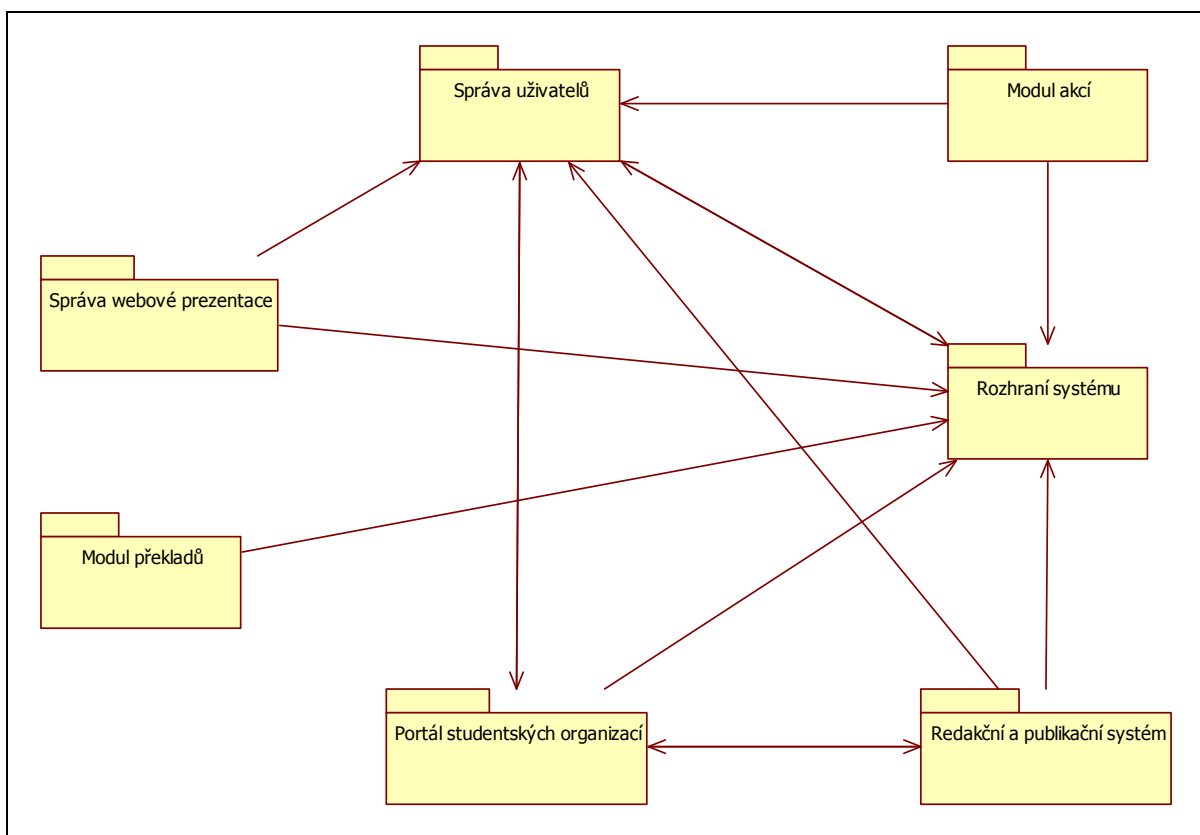


Obrázek 3: Rozdělení uživatelů podle přístupových práv

Vzhledem k tomu, že registrovaný uživatel může být členem některých studentských organizací, lze provést dělení i podle členství v organizacích. K tomuto dělení je však nutná spolupráce s Portálem studentských organizací.

4.3.3 Rozhraní systému

Rozhraní systému je jednou z dokumentovaných součástí systému (viz [7]). Představuje vlastní jádro systému (viz Obrázek 4). Jeho hlavní činností je poskytnout ostatním modulům zázemí pro jejich fungování. Rozhraní nabízí ostatním modulům použití jednotného uživatelského rozhraní a spravuje tyto moduly. Správa modulů se týká jejich registrace v systému, přístupových práv pro uživatele a přihlašování uživatelů do systému (viz Obrázek 5). Rozhraní také zajišťuje funkce společné pro všechny moduly, jako je připojení k databázi, práce se společnou částí databáze a další často potřebné funkce. Dále se stará o společná nastavení globálních proměnných, jako je volba jazykové verze, identifikace přihlášeného uživatele, identifikace aktivního modulu a dalších.



Obrázek 4: Spolupráce mezi jednotlivými moduly IS ACSA

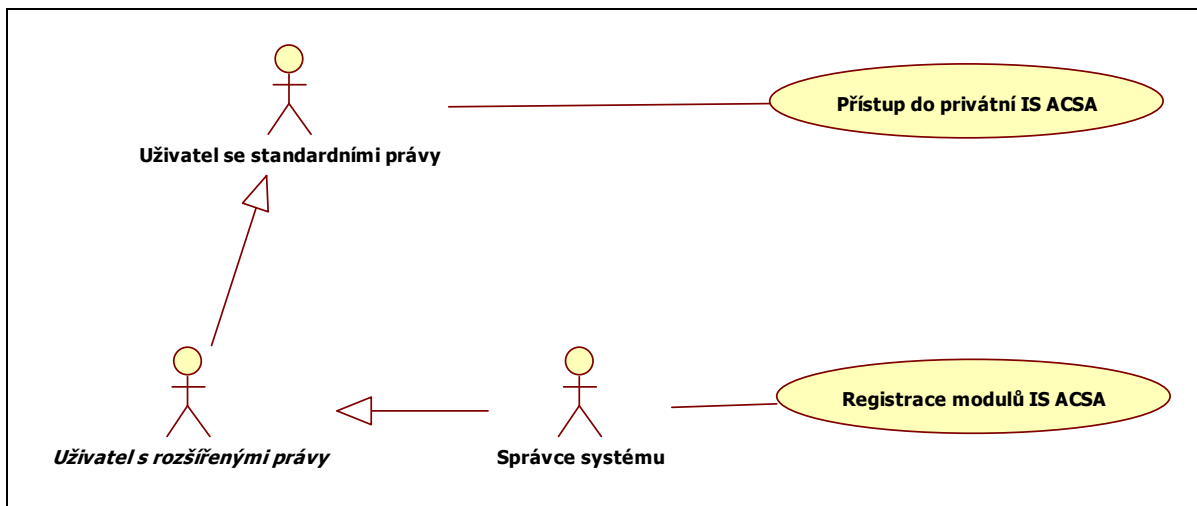
Rozhraní také definuje způsob spolupráce mezi jednotlivými moduly, který by měl být autory jednotlivých modulů striktně dodržován. Každý modul může pracovat pouze se svými databázovými tabulkami. K ostatním datům může přistupovat pouze pomocí exportních funkcí modulu, který daná data spravuje. Tímto je zajištěna nezávislost implementací jednotlivých modulů. Exportní funkce vytváří rozhraní pro komunikaci a eliminují vznik nežádoucích závislostí.

4.3.3.1 Kladné vlastnosti

Mezi kladné vlastnosti Rozhraní IS ACSA patří:

- snaha zabránit vzniku duplicitního kódu centralizací užitečných funkcí,

- zajišťuje jednotné uživatelské rozhraní,
- stanovuje podmínky komunikace mezi moduly,
- spravuje společné informace o modulech a obecně využitelné informace.



Obrázek 5: Hlavní činnosti a uživatelé Rozhraní IS ACSA

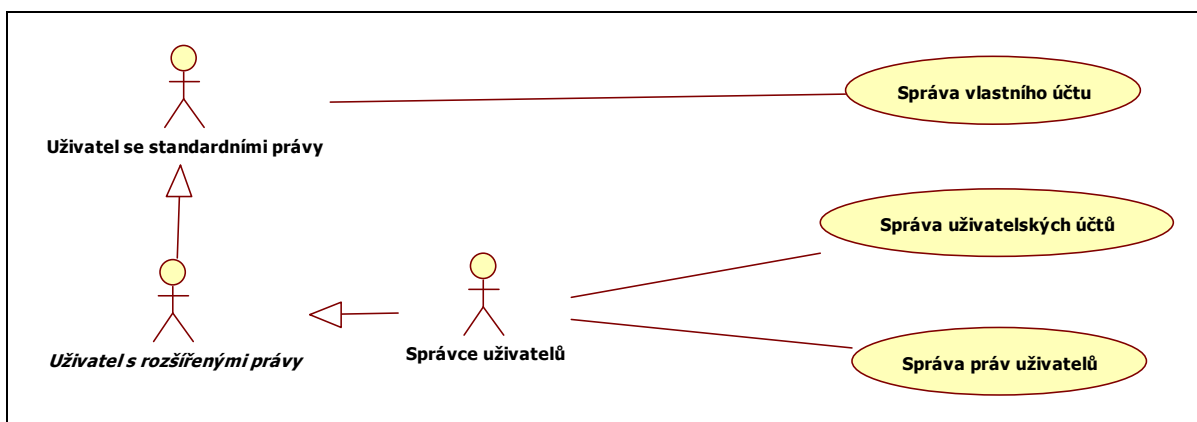
4.3.3.2 Záporné vlastnosti

Mezi záporné vlastnosti IS ACSA patří:

- dokumentace zdrojového kódu a realizovatelnost změn,
- neodděluje kód prezentující data od kódu, který data zpracovává.

4.3.4 Modul Správa uživatelů

Modul Správa uživatelů umožňuje přístup k uživatelským účtům. Každý uživatel systému má možnost spravovat svůj účet. Správce uživatelů nastavuje práva jednotlivých uživatelů a má možnost spravovat účty všech uživatelů systému (viz Obrázek 6).



Obrázek 6: Hlavní činnosti a uživatelé modulu Správa uživatelů

Ostatním modulům jsou poskytovány informace o uživateli a jejich právech. Toto je zajištěno prostřednictvím exportních funkcí modulu.

4.3.4.1 Kladné vlastnosti

Mezi kladné vlastnosti patří:

- možnost přidělit uživateli práva ke konkrétnímu modulu.

4.3.4.2 Záporné vlastnosti

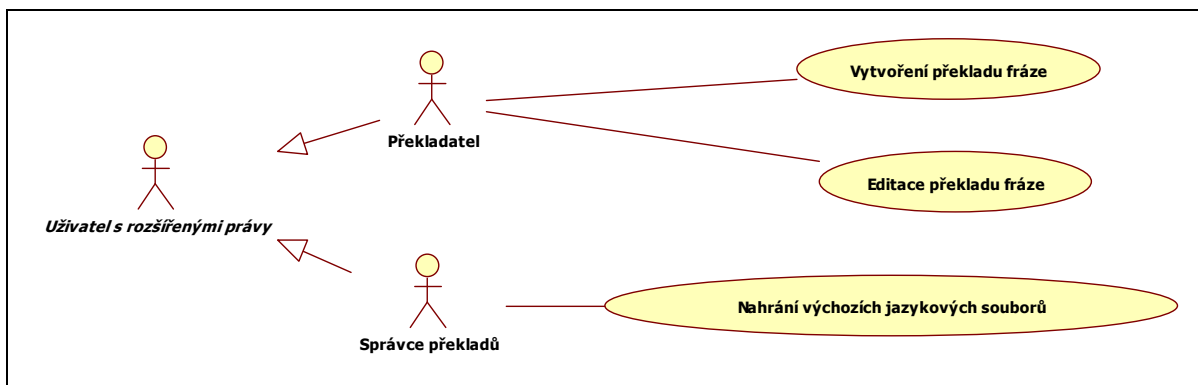
Mezi záporné vlastnosti patří:

- neexistující dokumentace,
- porušení pravidel komunikace mezi moduly,
- neoddělení prezentace dat od jejich zpracování.

4.3.5 Modul Překlady

Vzhledem k mezinárodní působnosti ACSA bylo již od začátku nutné vytvářet vícejazyčný informační systém. Celý systém používá kódování UTF-8. Každý modul obsahuje soubor lang_0.php, ve kterém jsou uloženy konstanty se všemi texty daného modulu. Na základě hodnot tohoto souboru jsou připraveny soubory pro ostatní jazyky. Při tvorbě překladů jsou přeložené texty ukládány do odpovídajících jazykových souborů (lang_1.php, ...). Označení jazyků čísly je dáno hodnotami v databázi.

Před vlastním překladem je nutné nastavit cesty k souborům lang_0.php a tyto soubory nahrát do systému. Překlad provádí překladatel uvnitř informačního systému (viz Obrázek 7). Největším problémem při překladech je práce se soubory. Při práci s velkými soubory je čas odezvy systému příliš dlouhý a byly reportovány i případy, kdy došlo ke ztrátě dat.



Obrázek 7: Hlavní činnosti a uživatelé modulu Překlady

4.3.5.1 Kladné vlastnosti

Mezi kladné vlastnosti modulu patří:

- poskytuje funkční způsob pro překládání textů,

- obsahuje velké množství přeložených textů.

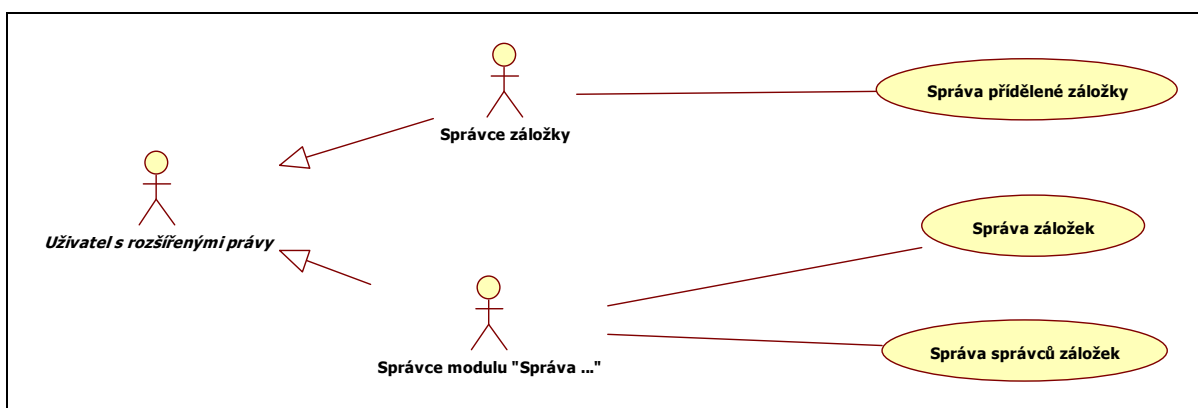
4.3.5.2 Záporné vlastnosti

Mezi záporné vlastnosti modulu patří:

- dlouhá odezva při práci s velkými soubory,
- reportovány případy, kdy došlo ke ztrátě dat,
- neexistence dokumentace.

4.3.6 Modul Správa webové prezentace

Tento modul umožňuje oprávněným uživatelům editovat veřejnou část webové prezentace IS ACSA (viz Obrázek 8). Umožňuje měnit obsah ve všech úrovních menu a vytvářet příslušné stránky pomocí WYSIWIG⁷ editoru umístěného přímo v systému.



Obrázek 8: Hlavní činnosti a uživatelé modulu Správa webové prezentace

4.3.6.1 Kladné vlastnosti

Mezi kladné vlastnosti patří:

- ACSA může jednoduše měnit obsah webové prezentace,
- uživatelsky příjemné a jednoduché provádění změn stránek.

4.3.6.2 Záporné vlastnosti

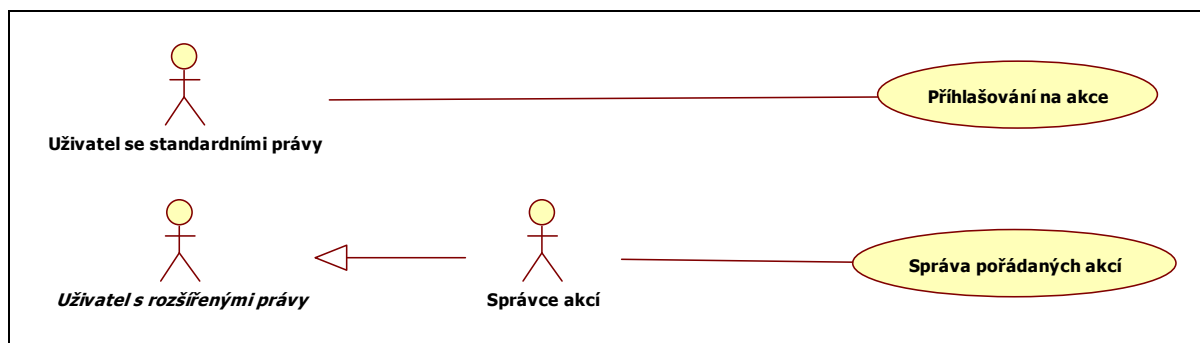
Mezi záporné vlastnosti patří:

- pomocí WYSIWIG editoru vzniká nepříliš kvalitní HTML kód,
- neexistuje dokumentace,
- není vazba s Redakčním a publikačním systémem.

⁷ WYSIWIG – více informací lze nalézt v [19]

4.3.7 Modul Akce

Modul Akcí zprostředkovává přihlašování uživatelů IS ACSA na pořádané akce. Zaměstnanci ACSA vytvářejí akce a potvrzují přijetí na danou akci (viz Obrázek 9).



Obrázek 9: Hlavní činnosti a uživatelé modulu Akce

4.3.7.1 Kladné vlastnosti

Mezi kladné vlastnosti modulu patří:

- poskytuje kvalitní podporu pro pořádané akce,
- velmi dobře strukturovaný modul,
- velký význam pro ACSA.

4.3.7.2 Záporné vlastnosti

Mezi záporné vlastnosti patří:

- nedodržuje pravidla o komunikaci mezi moduly (pracuje s tabulkami uživatelů),
- nemá se zbytkem IS ACSA plně konzistentní uživatelské rozhraní,
- neexistuje dokumentace.

4.3.8 Portál studentských organizací

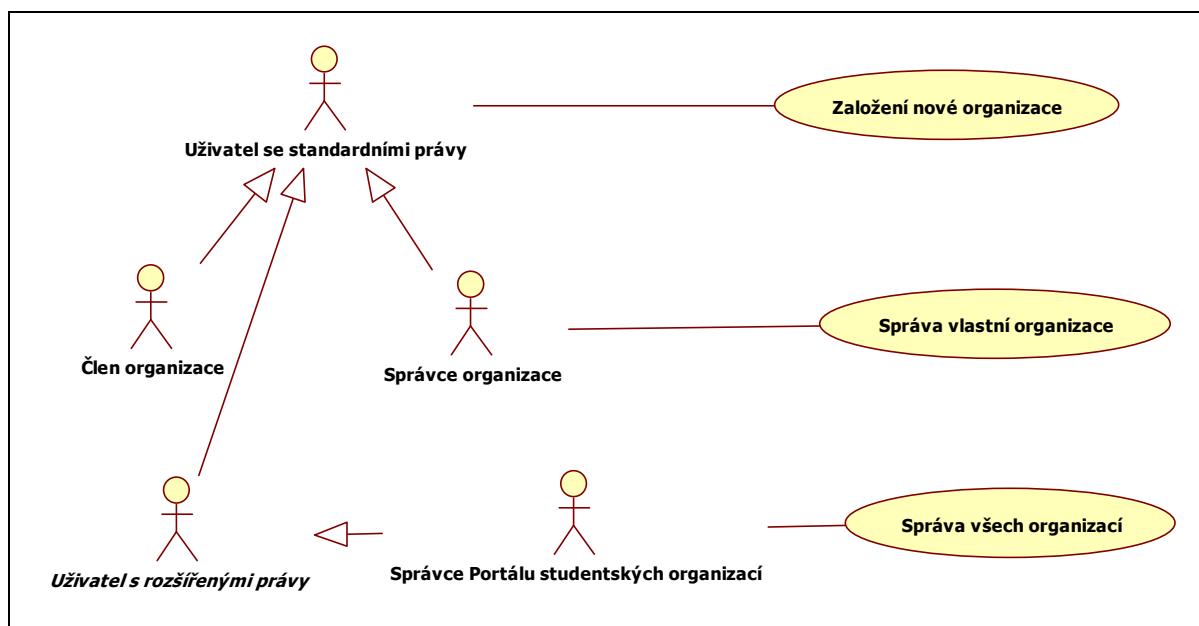
Portál studentských organizací (uživatelé modulu viz Obrázek 10) je velmi rozsáhlý informační systém, který vznikl s cílem shromáždit informace o studentských organizacích a poskytnout jim prostor pro vlastní prezentaci. Bohužel většina organizací ho chápe pouze jako adresář a nemá příliš velký zájem jej využívat jiným způsobem. Toto je možná způsobeno nepříliš přívětivým a složitým uživatelským rozhraním. Kromě modulu Správa uživatelů nejsou ostatní části IS ACSA na tomto modulu závislé. Modul Správa uživatelů využívá adresář vysokých škol a informace o členství v organizacích.

4.3.8.1 Kladné vlastnosti

Mezi kladné vlastnosti patří:

- rozsáhlá dokumentace (viz [10]),

- kvalitně strukturovaný kód.



Obrázek 10: Hlavní činnosti a uživatelé Portálu studentských organizací

4.3.8.2 Záporné vlastnosti

Mezi záporné vlastnosti patří:

- komplikované uživatelské rozhraní,
- nedokončená integrace funkcí Redakčního a publikačního systému,
- malá přidaná hodnota pro studentské organizace (chápou to spíše jako seznam)
- v současné době není důvod být členem organizace – neexistuje privátní část.

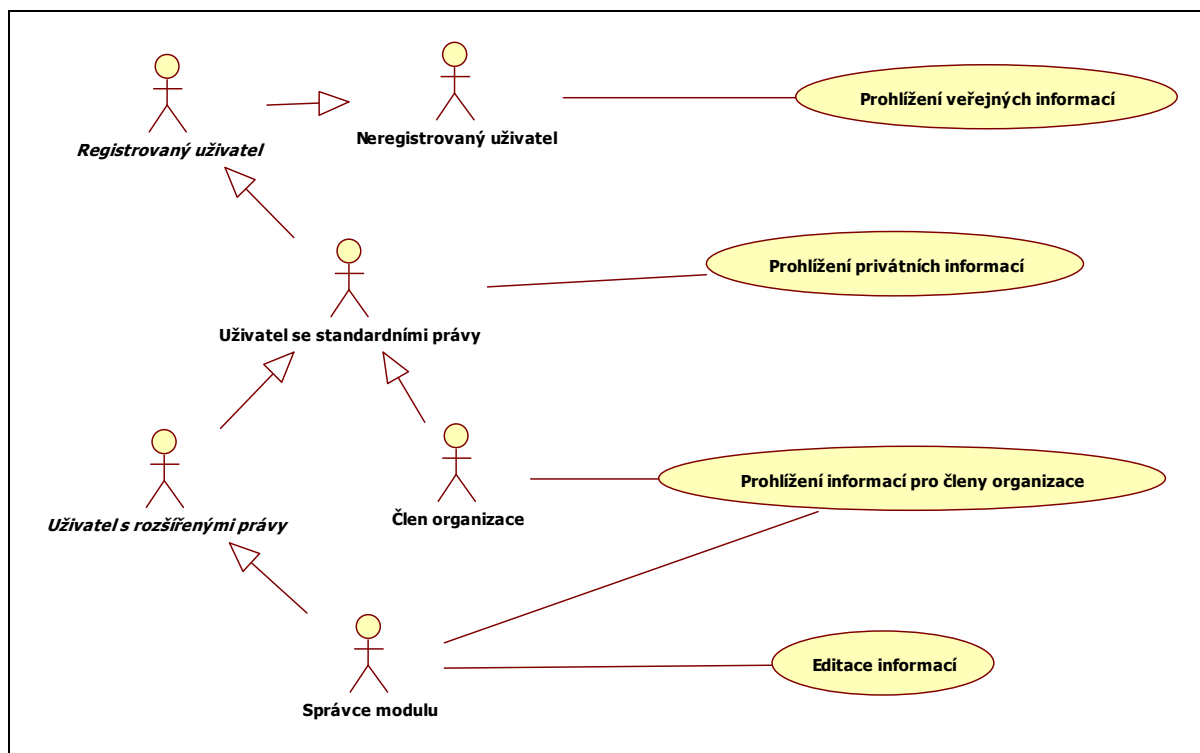
4.3.9 Modul Redakční a publikační systém

Redakční a publikační systém⁸ seskupuje dohromady moduly umožňující vytvářet aktuality, články, ankety, dotazníková šetření, diskusní fóra, posílat emaily a vkládat fotky a soubory ke stažení. Důležitým požadavkem na tento modul byla možnost využití služeb v kterékoli části IS ACSA a možnost jejich datového oddělení. Každá služba modulu RPS má vlastní adresář, kde jsou uloženy zaregistrované instance této služby. Přístup k dané instanci služby probíhá pomocí unikátních 32 bitových řetězců, které jsou odlišné pro různé skupiny uživatelů. Uživatelským rozhraním určeným pro práci se službou je klasická HTML stránka. V IS ACSA je služba vložena na libovolné místo pomocí HTML prvku „<iframe>“.

Uživatelé a jimi prováděné činnosti jsou v jednotlivých modulech RPS velmi podobné (viz Obrázek 11). Obecně by měly být rozlišeny čtyři skupiny uživatelů (4 typy přístupových řetězců) –

⁸ dále jen RPS

neregistrovaný, registrovaný, člen organizace a správce. Ne všechny moduly RPS však toto pravidlo dodržují.



Obrázek 11: Hlavní činnosti a uživatelé modulu Redakčního a publikačního systému

4.3.9.1 Kladné vlastnosti

Mezi kladné vlastnosti modulu patří:

- lze ho používat v libovolné části IS ACSA,
- velká nezávislost na ostatních částech.

4.3.9.2 Záporné vlastnosti

Mezi záporné vlastnosti modulu patří:

- chybí propojení do ostatních částí IS ACSA – registrace a použití služeb,
- každý modul má vlastní způsob použití služby – neexistuje společné rozhraní,
- některé moduly nejsou v IS ACSA vůbec používány,
- nedostatek komentářů a často nevhodně strukturovaný kód,
- obtížné stanovení přesné vertikální velikosti prvku „<iframe>“ v různých prohlížečích,
- nemožnost použití tlačítek zpět a vpřed v internetovém prohlížeči.

4.3.10 Celkové zhodnocení IS ACSA

IS ACSA je úspěšně používaný informační systém. Pro ACSA představuje nenahraditelný nástroj pro každodenní práci. Během používání byl IS ACSA průběžně měněn s cílem zlepšit jeho jednotlivé součásti. V současné době jsou technologie a postupy použité pro vývoj systému již značně zastaralé.

Podpora pro PHP4 skončila 31. 12. 2007. V nové verzi PHP5 se již plně prosadil objektově orientovaný přístup k programování a databáze MySQL5 také přináší mnoho užitečných zlepšení dříve dostupných pouze v komerčních produktech. Proto nastává vhodná doba pro vytvoření nové verze systému reflektující současnou situaci.

Dalším důvodem je, že vědomosti spojené s údržbou systému jsou převážně soustředěny mezi dvěma lidmi, kteří pracují na jeho rozvoji. Tento stav je nadále neudržitelný a je nutné tyto vědomosti přesunout do dokumentace systému.

4.3.10.1 Kladné vlastnosti

Mezi hlavní kladné vlastnosti systému patří:

- funguje a je ověřen více než třemi lety provozu,
- poskytuje ACSA prostředek pro komunikaci s uživateli,
- je možné snadno přidávat nové moduly.

4.3.10.2 Záporné vlastnosti

Mezi hlavní záporné vlastnosti systému patří:

- špatná dokumentace,
- nevhodně strukturovaný kód některých částí,
- není oddělena prezentace dat od jejich zpracování,
- nedůsledné dodržování pravidel stanovených rozhraním,
- využívá zastaralé technologie a postupy.

5 Návrh architektury nového IS ACSA

V této kapitole jsou aplikovány znalosti uvedené v předchozích kapitolách s cílem vytvořit návrh architektury nového IS ACSA. Jak již z názvu této práce plyne, hlavním cílem je vytvoření dobře udržitelného informačního systému.

5.1 Požadavky kladené na novou architekturu

Jak bylo uvedeno v kapitole 3.1, při návrhu architektury jsou důležité především nefunkční požadavky. Funkční požadavky nemají v této fázi návrhu tak velký význam. Dalším faktem je, že odlišnost funkčních požadavků od toho, co poskytuje současný IS ACSA, je z pohledu architektury zanedbatelná. Funkčnost současného IS ACSA je probrána v kapitole 4.3, a proto není třeba funkční požadavky opětovně uvádět.

Hlavními nefunkčními požadavky na novou architekturu IS ACSA jsou:

- podpora více jazyků,
- snadná rozšiřitelnost o nové moduly,
- stanovení pravidel pro tvorbu dokumentace,
- automatizovaná tvorba dokumentace ze zdrojových textů,
- důsledné oddělení prezentace dat od jejich zpracování,
- použití objektově orientovaného přístupu v PHP5,
- databázový server MySQL5.

Část těchto požadavků splňuje i současný IS ACSA. Jedná se o podporu více jazyků a relativně snadnou rozšiřitelnost o nové moduly. Ostatní požadavky se týkají především lepší udržitelnosti a nových technologií.

5.2 Výběr vhodného architektonického rámce

Výběr architektonického rámce je ovlivněn především požadavky uvedenými v kapitole 5.1. Těmto požadavkům velmi dobře vyhovuje rámec MVC (model-view-controller), a to z těchto důvodů:

- odděluje prezentaci dat od jejich zpracování,
- je ve spojení s PHP úspěšně používán, což dokazuje velké množství Open Source implementací tohoto rámce⁹.

Mimo tento rámec se nachází požadavky na dokumentaci. Tyto budou splněny stanovením pravidel pro psaní zdrojového kódu a tvorbu dokumentace.

⁹ více informací lze nalézt v [20]

5.2.1 MVC z pohledu IS ACSA

5.2.1.1 Model

Model zajišťuje práci s daty IS ACSA. Vytváří logiku dat nad databází a poskytuje tento pohled na data komponentě View. Dále také zpracovává požadavky komponenty Controller.

Z pohledu programového kódu se v této části vyskytuje pouze kód jazyka PHP a případně SQL dotazy pracující s databází.

5.2.1.2 View

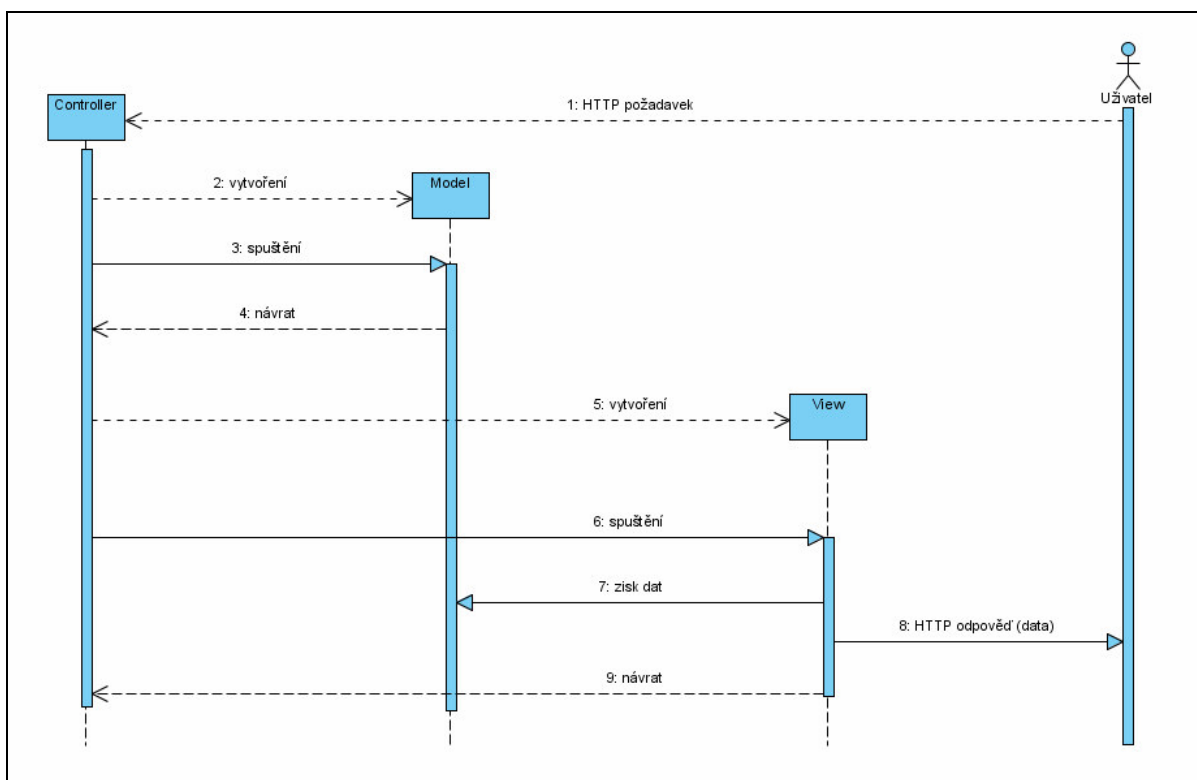
Tato část obsahuje uživatelské rozhraní IS ACSA, které zobrazuje požadovaná data a nabízí možnost interakce se systémem (odeslání formuláře, kliknutí na odkaz). V případě HTML stránky je s pomocí prohlížeče vytvořen a odeslán nový HTTP požadavek.

Z pohledu programového kódu se jedná o dokumenty HTML (doplněné o CSS, JavaScript) obsahující malé množství kódu PHP sloužícího pouze ke generování dalšího HTML výstupu.

5.2.1.3 Controller

Komponenta Controller (řadič) reaguje na požadavky uživatele a předává je komponentě Model, která na jejich základě mění svůj stav. Řadič také vybírá jaký pohled definovaný komponentou View bude použit pro prezentaci.

Z pohledu programového kódu se jedná čistě o PHP.



Obrázek 12: Nástin komunikace mezi vrstvami MVC a uživatelem IS ACSA

5.3 Privátní a veřejná část systému

Nový systém bude mít, stejně tak jako původní, privátní a veřejnou část. Veřejná část bude prezentovat data modulu Správa webové prezentace. V privátní části, přístupné po přihlášení, bude přístup k jednotlivým modulům nového IS ACSA. Model práv přístupu zůstane zachován v souladu s původním systémem. Toto by mělo zjednodušit následnou integraci některých původních modulů do nového systému.

V systému jsou definovány projekty, které jsou ekvivalentem pro jednotlivé moduly IS ACSA (Správa uživatelů apod.). Tyto projekty obsahují přístupové moduly (nelze však zaměňovat s moduly IS ACSA), ke kterým je možné udělit přístup konkrétnímu uživateli. Přístupové moduly jsou obvykle dvou typů – uživatelské a administrátorské.

5.4 Integrace původních modulů IS ACSA

Ačkoli není náplní této práce integrace původních modulů IS ACSA, je nutné se touto problematikou zabývat. Protože celý systém nebude přestavěn současně, bude nutné po určitou dobu umožnit fungování těchto původních modulů tak, aby počet nutných změn v těchto modulech byl co nejmenší.

Problémem v tomto případě jsou především změny v databázi, které se týkají společných částí. Jde především o tabulky modulu Rozhraní systému a společná data. V případě, že to bude možné, budou tyto tabulky zachovány v původním stavu. V modulech, ve kterých budou tabulky změněny, musí být změněny i odpovídající rozhraní původních modulů.

5.5 Návrh jádra nového IS ACSA

Základem nového IS ACSA bude jádro založené na rámci MVC, které bude využíváno jednotlivými moduly. Jádro systému bude zajišťovat činnosti důležité z pohledu celého systému. Především se jedná o správu sezení (session), tj. uchování stavu při práci s IS ACSA, a bude poskytovat jednotné uživatelské rozhraní společných částí systému. Mezi činnosti jádra patří například:

- výběr jazyka,
- přihlášení/odhlášení uživatele,
- zajištění přístupu podle práv uživatele k jednotlivým modulům,
- nastavení kontextu pro moduly,
- zajištění připojení k databázi a unifikovaný způsob použití databáze moduly,
- výběr vhodného typu šablony

Ačkoli se přímo nabízí použití některé z volně dostupných implementací rámce MVC, bylo přistoupeno k implementaci rámce vlastního, který se stane jádrem IS ACSA. Hlavním důvodem pro

tento přístup je zájem o vytvoření vlastního řešení na míru IS ACSA. Při návrhu jádra bylo mnoho cenných zkušeností a informací čerpáno především z [21], [22], [23], [24].

Jádro systému nebude přímo přistupovat k žádným datům v databázi. Pro realizaci činností, které vyžadují databázová data, bude využito rozhraní odpovídajících modulů. Moduly nutné pro správné fungování jádra jsou:

- modul Správa uživatelů, který poskytuje informace nutné pro přihlašování k systému,
- modul Správa informačního systému, který poskytuje informace o přístupu k jednotlivým částem privátní části IS ACSA a datům ve společných tabulkách.

Dále bude pro získání informací zobrazovaných ve veřejné části IS ACSA využíván modul Správa webové prezentace, který tato data spravuje.

5.5.1 Komponenta Model

Komponenta Model poskytne modulům zázemí pro provádění jejich specifických akcí týkajících se zpracování informací. Z pohledu jádra IS ACSA lze rozlišit dva druhy – model, který se stará o fungování veřejné části IS ACSA, a model zajišťující privátní část.

Model poskytující zázemí pro privátní část zajišťuje přihlašování uživatelů a ověřuje jejich práva pro přístup ke konkrétním modulům privátní části IS ACSA. Dále tato komponenta provádí požadovanou akci vybraného modulu. Vlastní přihlašování probíhá pomocí fiktivního modulu privátní části IS ACSA, který zajistí ověření uživatele.

Diagram tříd komponenty Model je na Obrázku 13. Třídy a balíčky nepatřící přímo komponentě Model nejsou úplně specifikovány a lze je nalézt podrobně v dalších diagramech.

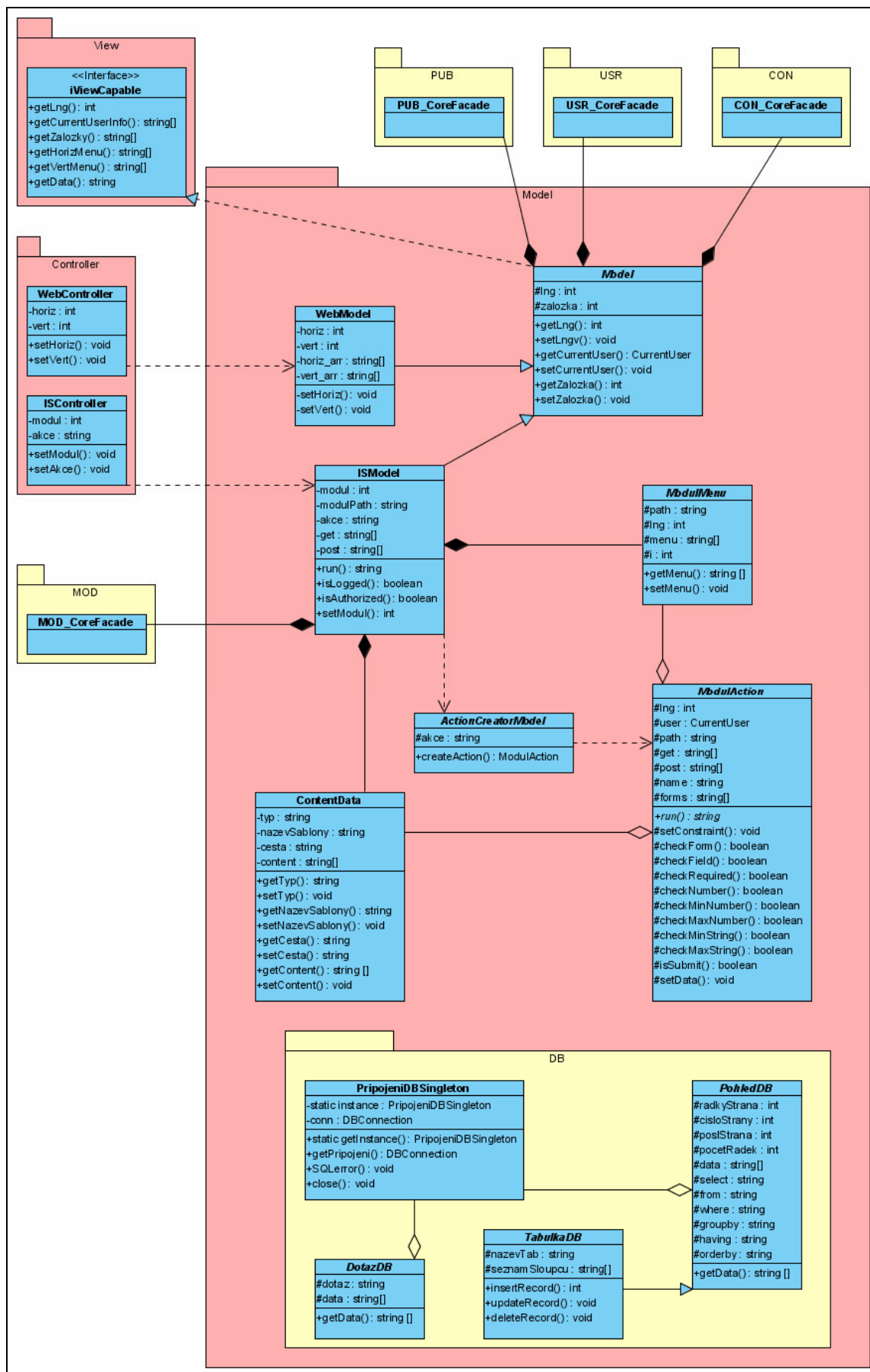
5.5.1.1 Rozšíření modelu o vlastní databázovou vrstvu

Protože by bylo vhodné, aby jádro poskytovalo ostatním modulům jednotný přístup k databázi, byly navrženy odpovídající třídy, které tuto činnost zajišťují. Jsou to třídy Tabulka, Pohled a Dotaz umístěné v balíčku DB, který je součástí komponenty Model. Jednotlivé moduly rozšiřují tyto třídy pro použití na konkrétních tabulkách a pohledech.

Třída Pohled umožňuje získávat data z databázových tabulek, pohledů a různých spojení tabulek. Třída Tabulka je rozšířením třídy Pohled o možnost vkládání, změnu a mazání dat. Třída Dotaz reprezentuje libovolný databázový dotaz, který předchozí dvě třídy nedokáží realizovat.

5.5.1.2 Komunikace jádra s moduly

Jediná komponenta, která získává data z jednotlivých modulů je právě Model. Komunikace probíhá s využitím návrhového vzoru Fasáda. Tímto je zajištěno, že všechny závislosti mezi jádrem a každým modulem jsou soustředěny právě na jediném konkrétním místě. Příklad tohoto řešení pro komunikaci mezi moduly je na Obrázku 16.



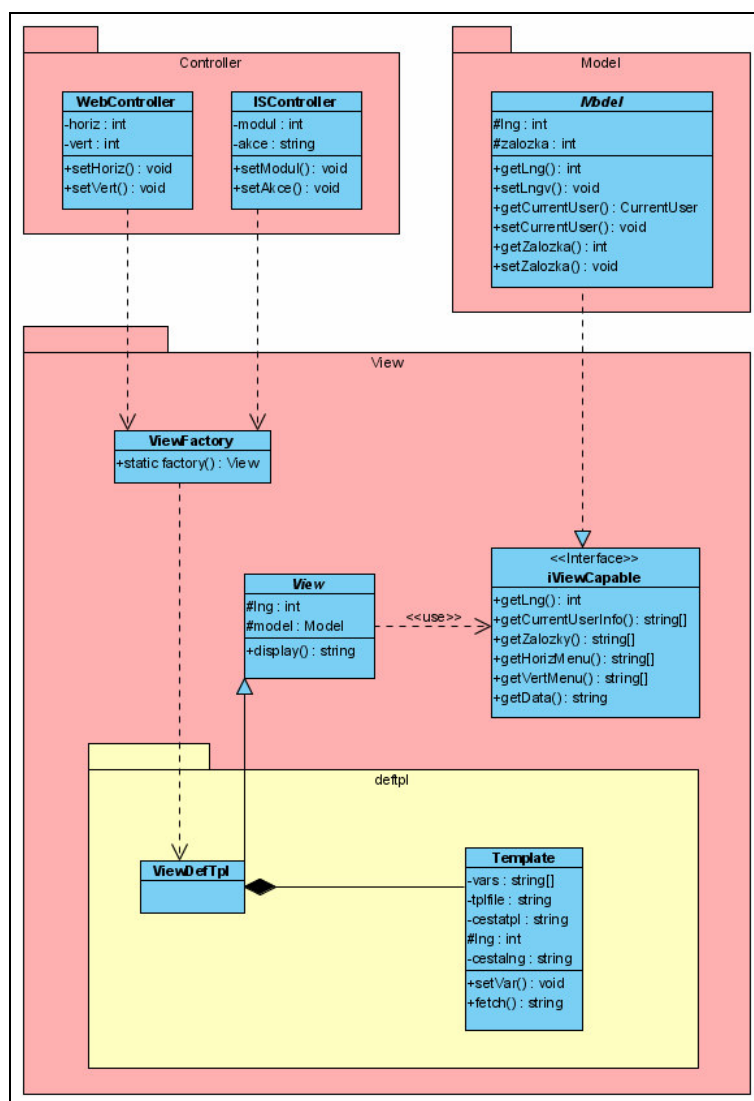
Obrázek 13: Diagram tříd komponenty Model

5.5.2 Komponenta View

Hlavním účelem této komponenty je prezentace dat, které jsou získány z komponenty Model. Způsob prezentace závisí na konkrétním typu View. Mělo by být možné implementovat libovolné množství různých pohledů na daný model. V novém IS ACSA bude vytvořen jediný výchozí pohled, ale nic nebrání vytvoření dalších. Komunikace s modelem probíhá pomocí rozhraní `iViewCapable`, které model implementuje.

Diagram tříd komponenty View je na Obrázku 14. Třídy a balíčky nepatřící přímo komponentě View nejsou úplně specifikovány a lze je nalézt podrobně v dalších diagramech.

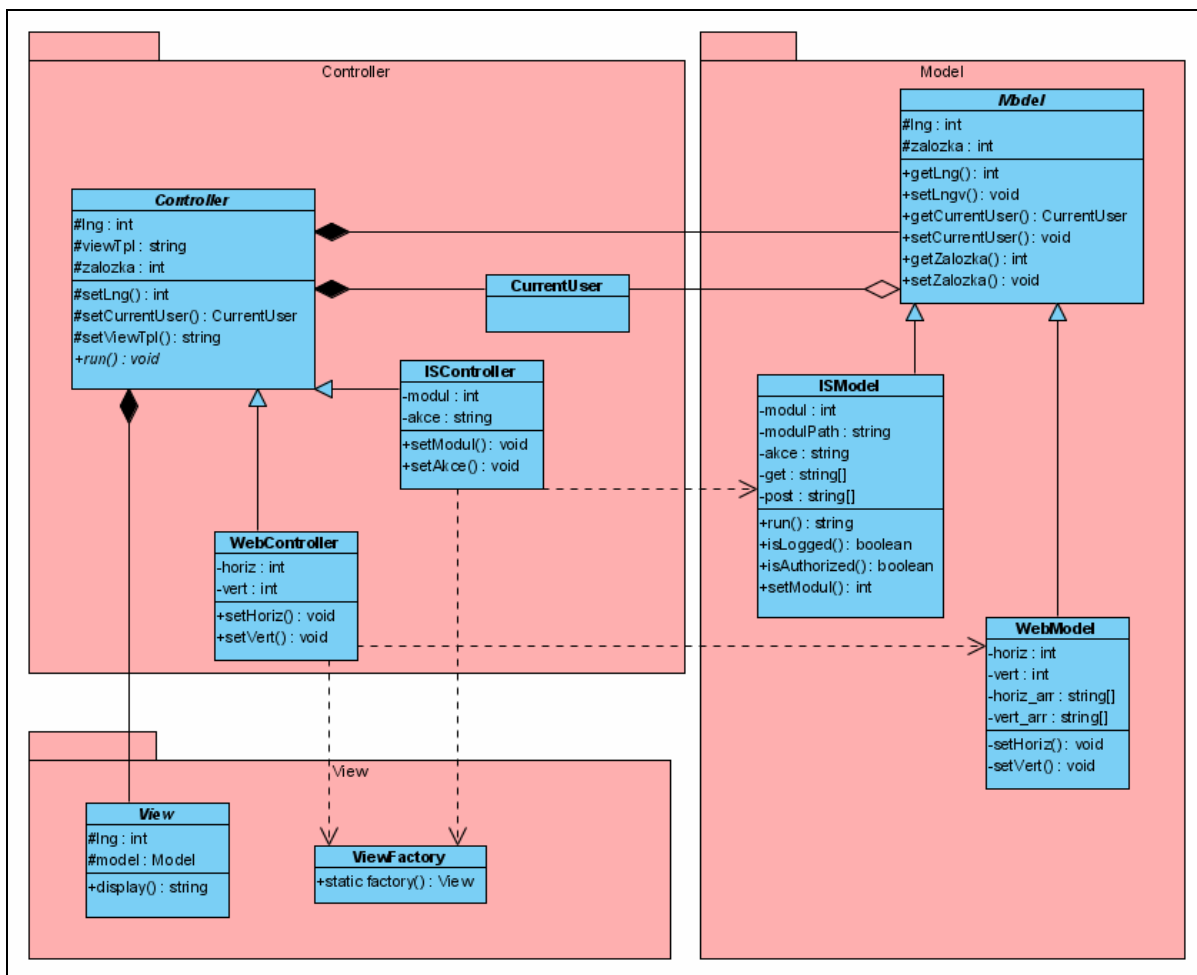
V diagramu je vidět balíček `DefTpl`, který představuje standardní pohled na model. V případě potřeby je možné vytvořit další podobné balíčky, které budou poskytovat pohled jiný. Tímto způsobem je možné docílit prezentace systému v různých formátech vhodných například pro zobrazení na mobilních zařízeních nebo ve formátu vhodném pro tisk apod.



Obrázek 14: Diagram tříd komponenty View

5.5.3 Komponenta Controller

Komponenta Controller řídí komunikaci mezi uživatelem a IS ACSA. Na základě požadavku uživatele vytvoří patřičný model a předá mu data ke zpracování. Dále pak vybere uživatelem požadovaný pohled. Starostí Controlleru je také správa sezení – stavu komunikace mezi uživatelem a systémem. Diagram tříd této komponenty je na Obrázku 15.



Obrázek 15: Diagram tříd komponenty Controller

5.6 Moduly nového IS ACSA

Nový IS ACSA bude obsahovat obdobně pojmenované, a co se funkčnosti týká také velmi podobné, moduly jako současná verze tohoto systému (budou provedeny změny, které jsou v současné době požadovány).

V rámci diplomové práce bude proveden návrh následujících modulů:

- modul Správa informačního systému,
- modul Správa uživatelů,
- modul Překlady,
- modul Správa webové prezentace,

- modul Redakční a publikační systém.

U modulu Redakčního a publikačního systému se bude jednat o návrh společného rozhraní jednotlivých existujících modulů tak, aby bylo možné tyto moduly jednotně používat v celém systému IS ACSA.

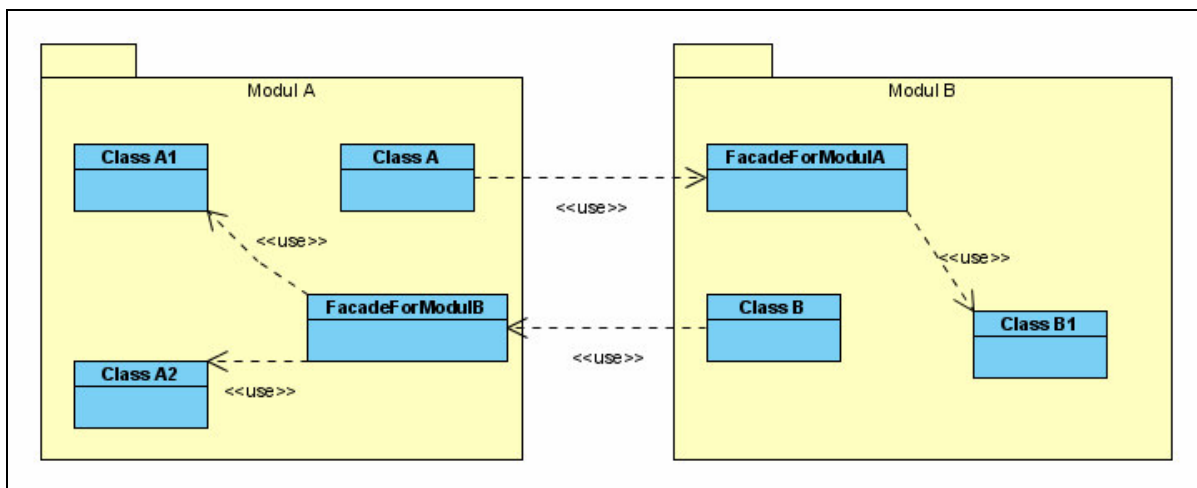
Zbývající dva současné moduly (modul Akcí, Portál studentských organizací) nebudou v rámci diplomové práce řešeny, neboť se jedná o značně rozsáhlé informační systémy.

Dále bude nový IS ACSA obsahovat dva fiktivní moduly, které neoperují s vlastními daty. Jedná se o modul pro přihlašování a modul s výchozí stránkou uživatele v privátní části systému.

Všechny dostupné moduly budou využívat podporu poskytovanou jádrem systému, a to především v oblastech prezentace dat a komunikace s databází. Moduly komunikují přímo pouze s komponentou Model, která využívá jimi implementovaných akcí k provádění všech činností nad daty informačního systému. Každý modul má část komunikující s vlastními tabulkami pomocí tříd odvozených od tříd z balíčku DB v komponentě Model.

5.6.1 Datová komunikace mezi moduly

Každý modul přistupuje přímo pouze ke svým datům. Z důvodu eliminace závislostí mezi moduly je nutné vytvořit rozhraní pro komunikaci. Pro tento účel lze velmi dobře využít návrhového vzoru Fasáda. Tímto je zajištěno, že všechny závislosti mezi moduly jsou soustředěny pouze v konkrétních fasádách těchto modulů. Princip je vidět na Obrázku 16.

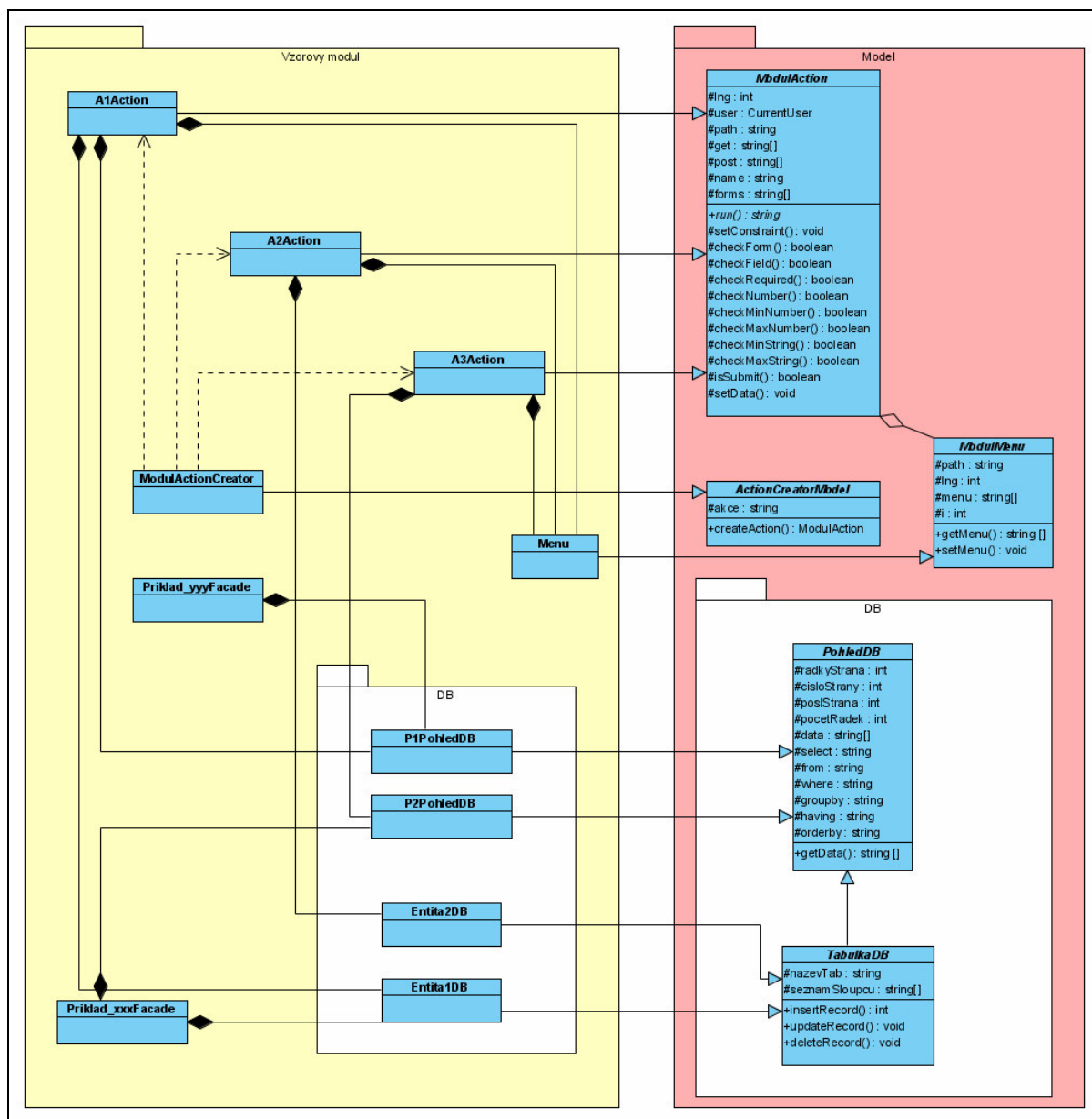


Obrázek 16: Příklad rozhraní s využitím vzoru Fasáda

5.6.2 Návrh modulu pro nový IS ACSA

Moduly nového IS ACSA budou zodpovědné za provádění konkrétních akcí nad svými daty. Od jádra převezmou data zasláná uživatelem a podle obsahu těchto dat provedou odpovídající změny v systému.

Každý modul zajistí práci s vlastními daty a v případě potřeby poskytne potřebná rozhraní pro ostatní moduly. Modul bude obsahovat akce, které je možné nad jeho daty provádět. Rozlišení jednotlivých akcí a jejich spuštění zajistí speciální třída každého modulu. Dále musí každý modul obsahovat třídu pro vlastní navigační menu. Na Obrázku 17 je diagram tříd, který ukazuje vzorový modul pro nový IS ACSA. Všechny moduly nového systému budou navrženy a implementovány v souladu s tímto příkladem.



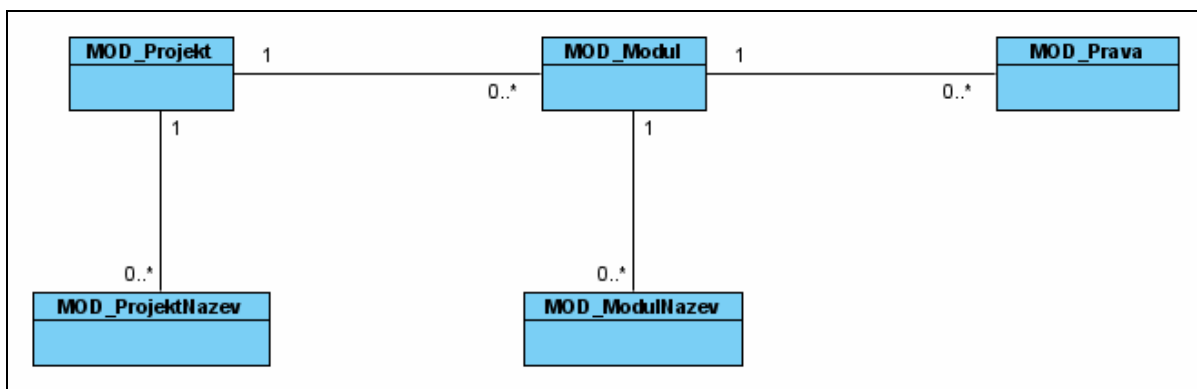
Obrázek 17: Diagram tříd vzorového modulu nového IS ACSA

5.6.3 Modul Správa informačního systému

Tento modul spravuje informace o ostatních modulech a přístupových právech uživatelů k nim. Z pohledu původního systému pracuje nad tabulkami modulu Rozhraní systému. Zkratka označení modulu je zachována v souladu s původní verzí, a to MOD. Všechny třídy a databázové tabulky tohoto modulu nesou tuto předponu.

Tento modul definuje projekty, které odpovídají jednotlivým modulům IS ACSA. Dále pak definuje přístupové moduly pro různé uživatele, které přísluší daným projektům. Přístupové moduly jsou obvykle dvou typů, a to administrátorské a uživatelské. Není to však vyžadováno a lze vytvořit i další moduly v závislosti na potřebách daného projektu.

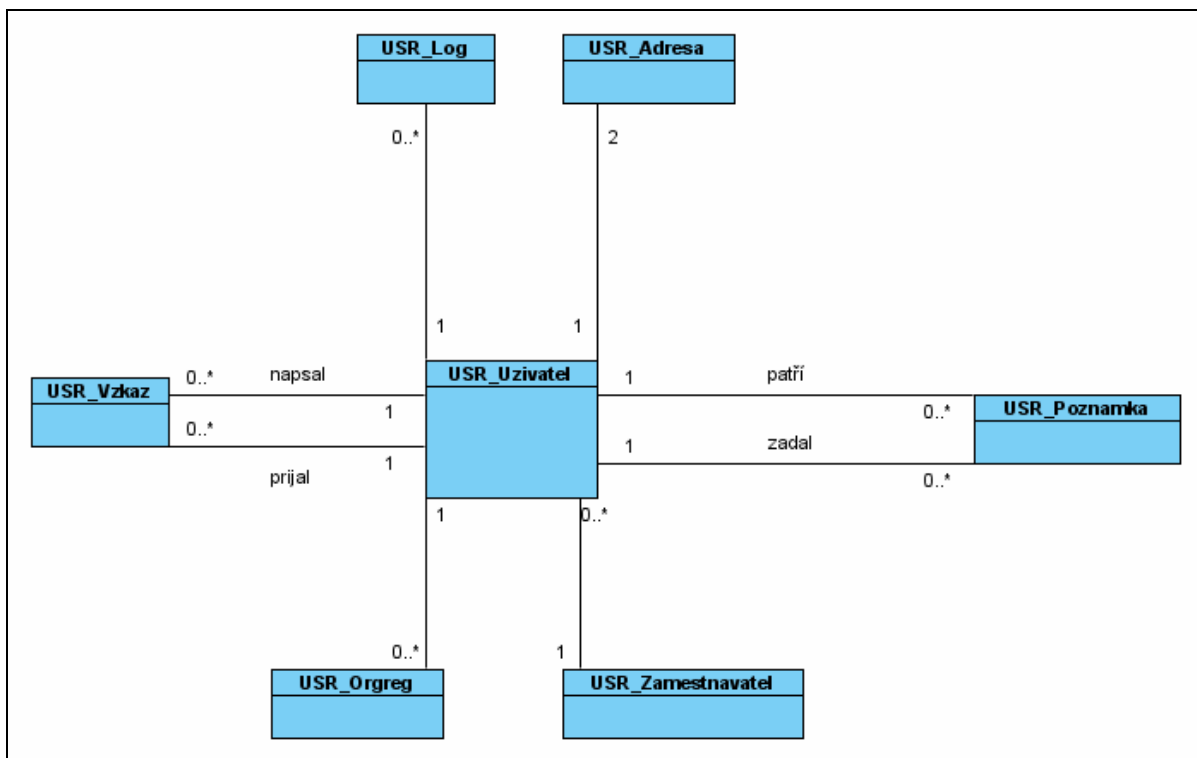
Konceptuální diagram tříd tohoto modulu je na Obrázku 18 – atributy jednotlivých entit budou uvedeny až u schématu databáze. Stejně tak i vazby na ostatní moduly. Tento přístup bude použit i u dalších diagramů.



Obrázek 18: Konceptuální diagram tříd modulu Správa informačního systému

5.6.4 Modul Správa uživatelů

Modul Správa uživatelů má na starost uživatelské účty a data s nimi spojená. Ostatním modulům poskytuje informace o uživateli. Zkratka modulu je USR. Jeho funkčnost se v zásadě shoduje s modulem v původní verzi IS ACSA. K tomuto modulu budou existovat v privátní části dva přístupové moduly. Jednak administrátorský umožňující správu všech uživatelů a další, který bude zpřístupněn běžnému uživateli a umožní mu změnu dat spojených s jeho vlastním účtem. Na Obrázku 19 je konceptuální diagram tříd tohoto modulu.



Obrázek 19: Konceptuální diagram tříd modulu Správa uživatelů

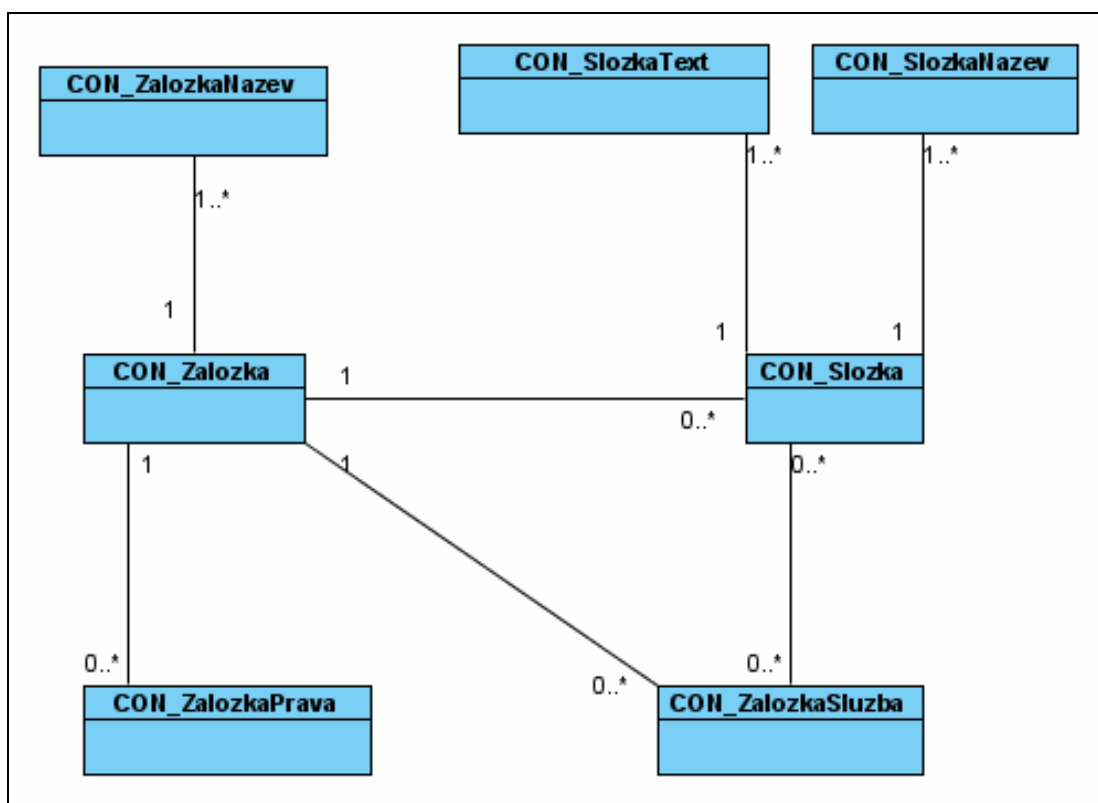
5.6.5 Modul Správa webové prezentace

Na rozdíl od modulu Správa uživatelů bude modul Správy webové prezentace v novém IS ACSA podstatně vylepšen. Je nutné jej především více spojit s modulem Redakčního a publikačního systému tak, aby bylo možné jednoduše využívat poskytovaných služeb. Zkratka tohoto modulu je CON a platí stejná pravidla jako u ostatních modulů, že všechny třídy a tabulky nesou tuto předponu.

Uživatele tohoto modulu lze rozdělit do tří skupin:

- správce modulu,
- správce záložky,
- standardní uživatel.

Správce modulu vytváří nové záložky, určuje jejich správce a povoluje používání služeb Redakčního a publikačního systému na dané záložce. Správce záložky provádí editaci obsahu záložky. Standardní uživatel má právo prohlížet obsah záložek, které jsou součástí webové prezentace IS ACSA. Konceptuální diagram tříd je na Obrázku 20.



Obrázek 20: Konceptuální diagram tříd modulu Správa webové prezentace

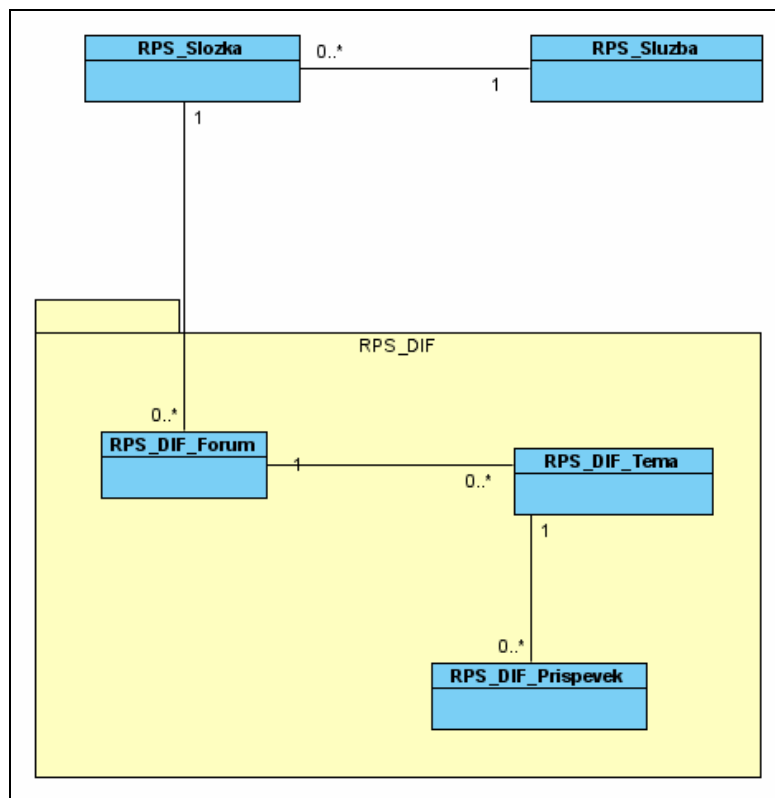
5.6.6 Modul Redakční a publikační systém

Největším problémem v původním modulu je, že neexistuje jednotné rozhraní pro přístup k poskytovaným službám. Modul poskytuje několik služeb, které nelze jednotně používat.

Řešením uvedeného problému je vytvoření společného adresáře, který bude sjednocovat přístup ke všem poskytovaným službám. Kromě adresáře bude také existovat společná složka, ve které budou uloženy všechny zaregistrované instance konkrétních služeb.

Na Obrázku 21 je konceptuální diagram tříd tohoto modulu. Tento diagram obsahuje pouze modul DIF (Diskusní fóra) jako příklad použití společného adresáře služeb. Ostatní služby tohoto modulu budou využívat adresář stejným způsobem.

Umístění služeb v informačním systému prostřednictvím vloženého rámce v HTML stránce zůstane zachováno. Pro fungování s novým rozhraním bude nutné každou službu upravit tak, aby dokázala využít tohoto rozhraní místo svého původního rozhraní. Vnitřní fungování služeb zůstane zachováno ze současné verze systému. Možnosti transformace fungování služeb uvnitř vloženého rámu je věnována kapitola 8.3.



Obrázek 21: Konceptuální diagram tříd modulu Redakční a publikační systém

5.6.7 Modul Překlady

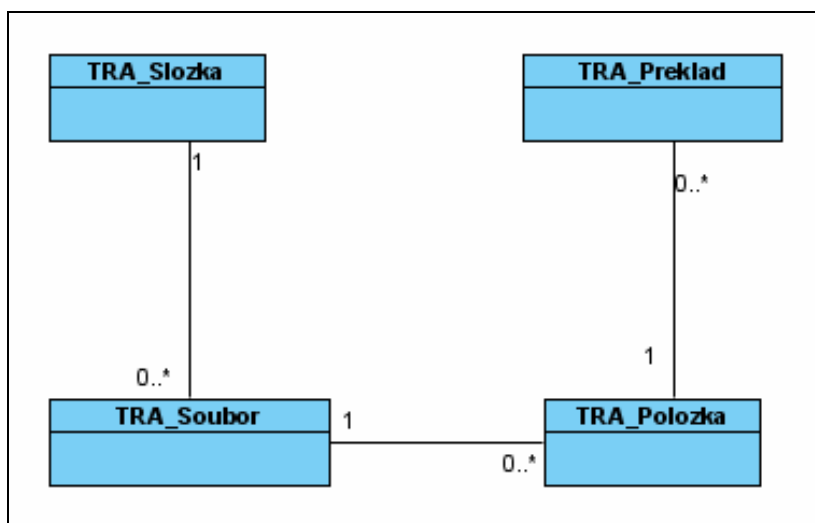
Modul Překlady je v současném IS ACSA velice důležitou součástí. Jeho problémem je časová náročnost zpracování velkých souborů, která je způsobena centralizací všech textů pro daný modul do jediného souboru. Použité kódování textů v celém IS ACSA je UTF-8.

V současné době spravuje modul velké množství textu, které je nutné v novém modulu dále využívat (nelze provádět překlad textů znovu). Pro překlad jsou texty uloženy v databázových tabulkách a jsou rozlišovány jednoznačnými textovými identifikátory. Pro zobrazení jsou tyto texty načítány ze souborů, které jsou po překladu generovány.

Modul spravuje statické texty, které jsou vkládány do stránek prezentovaných uživatelům IS ACSA. O prezentaci se v novém IS ACSA stará komponenta View, která určuje způsob zobrazení. Použití textů tedy velmi úzce souvisí se způsobem prezentace dat uživatelům. Uživatelské rozhraní a způsob použití přeložených textů je uveden v následující kapitole 5.7. Texty jsou používány nezávisle na tomto modulu.

Při vytvoření nového textu je tento text načten z výchozího jazykového souboru do databáze. Modul Překlady bude mít dva typy uživatelů, a to správce a překladatele. Správci nastavují cesty k jednotlivým výchozím jazykovým souborům. Překladatelé provádějí překlad textů.

Řešením problému s velkými soubory je rozdělení těchto souborů na menší, které se vztahují k právě zobrazované stránce. Toto řešení se přímo nabízí ve spojení se způsobem prezentace dat popsaným v již zmíněné kapitole 5.7. Struktura modulu Překlady je vidět na Obrázku 22.



Obrázek 22: Konceptuální diagram tříd modulu Překlady

5.7 Uživatelské rozhraní nového IS ACSA

Nový IS ACSA umožňuje vytvořit různá uživatelská rozhraní. Komponenta View jádra systému bude standardně prezentovat systém ve formě HTML stránek. Pro tvorbu těchto stránek bude použit jednoduchý šablonovací systém, který sestaví výslednou stránku z předem připravených šablon. Vzhled nového uživatelského rozhraní bude velmi podobný současnému IS ACSA, avšak způsob jeho vytváření se bude od současného systému velice lišit.

Šablonovací systém bude využívat dvou typů šablon. První typ šablony zajistí zobrazení informací společných pro všechny části systému – hlavní šablona. Druhý typ pak bude mít každá stránka svůj – šablona stránky.

Hlavní šablona zajistí zobrazení následujících částí stránky:

- hlavička stránky,
- menu se záložkami,
- horizontální menu,
- vertikální menu,
- patička stránky,
- poskytne prostor pro vložení obsahu vytvořeného šablonou stránky.

Šablona stránky bude specifická pro konkrétní zobrazovaná data. V určitých případech bude místo šablony stránky vložen HTML kód dodaný modulem Správa webové prezentace, který umožňuje uživatelům měnit webovou prezentaci pomocí WYSIWYG editoru.

Šablony budou tvořeny v jazyce HTML s vloženými pasážemi v jazyce v PHP. Tento kód v PHP bude sloužit k zobrazování dat obsažených v proměnných šablony. Každá šablona bude mít svůj vlastní jazykový soubor, který bude možné pomocí modulu Překlady upravovat.

Všechny šablony budou dodržovat pravidla stanovená pro formát dokumentů v jazyce XHTML. Kódování všech znaků bude UTF-8.

5.8 Pravidla pro psaní zdrojového kódu a dokumentace

Kvalitní dokumentace je základem pro provádění jednoduché údržby systému. V této části jsou navržena pravidla, při jejichž dodržování bude nový IS ACSA z pohledu údržby lépe srozumitelný a přehledný.

Navrhovaná pravidla pro psaní zdrojového kódu jsou:

- každý zdrojový soubor obsahuje pouze jednu třídu,
- patří-li třída konkrétnímu modulu, začíná její název zkratkou tohoto modulu,
- dokumentace tříd je ve zdrojovém textu podle pravidel pro PHPDoc¹⁰,
- dodržovat pravidla pro názvy identifikátorů:
 - třída View: *NázevView*,
 - třída Controller: *NázevController*,
 - třída Model: *NázevModel*,
 - třída databázové vrstvy: *NázevDB*,
 - konstanta: *NÁZEVKONSTANTY*,
 - parametry funkcí, procedur a metod: *_název*,
- vytvářet validní HTML a CSS.

Pravidla pro tvorbu dokumentace jsou navržena takto:

- pro shromažďování informací o systému použít Wiki¹¹ systém,
- při tvorbě modulu vytvořit i odpovídající dokumentaci:
 - stručný popis modulu,
 - schéma databáze,
 - diagram tříd,
 - dokumentace tříd pomocí PHPDoc,
 - uživatelská příručka pro modul,
- při provádění změn reflektovat tyto změny do dokumentace,
- zaznamenávat nalezené chyby a způsob jejich opravy,
- zaznamenávat požadavky na úpravy systému a způsob jejich řešení.

¹⁰ PHPDoc je program pro automatizovanou tvorbu dokumentace ze zdrojových textů, více informací viz [28]

¹¹ Wiki je označení pro webový systém, který umožňuje měnit vlastní obsah.

6 Implementace

Implementace nového IS ACSA vychází z návrhů provedených v předchozí kapitole. Vzhledem k velkému rozsahu řešeného problému je v této kapitole navržen plán postupu implementace, která bude pokračovat i po dokončení této diplomové práce. Samotná implementace bude provedena v jazyce PHP a pro uchování dat bude použita databáze MySQL.

6.1 Plán implementace nového IS ACSA

Z důvodu velkého rozsahu je vhodné rozdělit implementaci nového IS ACSA do několika fází. Některé z těchto fází budou provedeny v rámci této diplomové práce, další budou následovat po jejím dokončení. V rámci diplomové práce bude provedena implementace prvních dvou fází a v případě dostatku času i části třetí fáze.

6.1.1 První fáze

V této fázi bude provedena implementace následujících částí nového IS ACSA:

- jádro systému,
- modul sloužící pro přihlašování k systému,
- část modulu Správa webové prezentace, poskytující data pro prezentaci na webu ACSA,
- modul Správa informačního systému.

6.1.2 Druhá fáze

V této fázi bude implementace pokračovat těmito částmi:

- modul Správa uživatelů,
- dokončení modulu Správa obsahu,
- společné rozhraní modulu Redakčního a publikačního systému.

6.1.3 Třetí fáze

V této fázi proběhne implementace:

- modul Překlady,
- přestavba a integrace jednotlivých modulů Redakčního a publikačního systému.

6.1.4 Další fáze

V dalších fázích bude nutné provést zejména tyto činnosti:

- integrace modulu Akcí,
- integrace modulu Portál studentských organizací,
- provedení změn vzniklých v průběhu akceptačního testování.

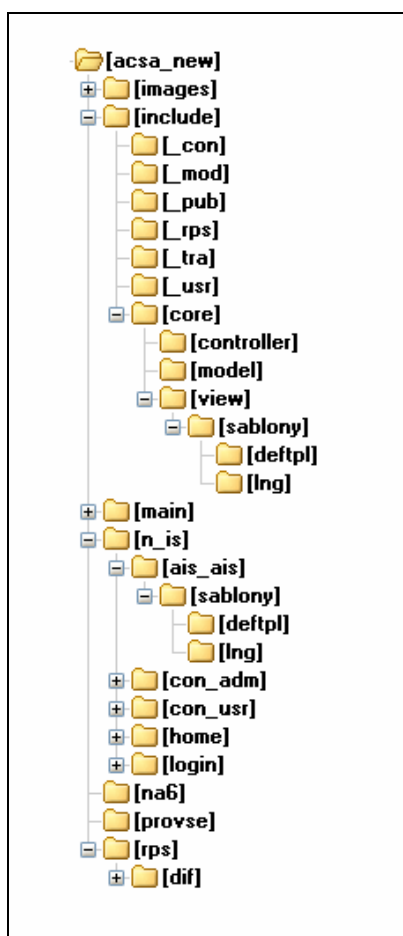
Dále je pak možné pokračovat přidáváním nových modulů a zlepšováním modulů současných.

6.2 Adresářová struktura nového IS ACSA

Adresářová struktura nového systému se v některých částech značně odlišuje od původní verze. Z důvodu přehlednosti jsou z Obrázku 23 odstraněny některé adresáře, které by neměly pro lepší pochopení uspořádání systému žádný význam.

Zcela novým adresářem je „include“, který obsahuje jednak soubory tříd implementující jádro systému (adresář „core“). Pak také obsahuje soubory tříd sloužících pro práci s databází jednotlivých modulů (adresáře začínající „_“).

Jádro je uloženo v adresáři „include/core“ a obsahuje své tři komponenty Model, View a Controller v odpovídajících adresářích. V adresáři komponenty View je dále adresář obsahující adresáře implementovaných šablon a adresář s jazykovými soubory.



Obrázek 23: Adresářová struktura nového IS ACSA

V adresářích „main“, „na6“, „provse“ apod. jsou uloženy soubory patřící k dané záložce, která patří do modulu Správa webové prezentace.

Adresář „n_is“ představuje privátní část IS ACSA. Jsou zde uloženy adresáře se soubory tříd jednotlivých modulů. Jejich názvy jsou dány nastavením prefixů v modulu Správa informačního systému. První prefix patří projektu (jde fakticky o modul, např. Správa webové prezentace – CON). Druhý prefix náleží přístupovému modulu, ke kterému je přidělován přístup uživatelům. Dále tento adresář obsahuje dva adresáře s fiktivními moduly – jde o modul sloužící pro přihlašování a modul s výchozí stránkou privátní části.

Posledním zajímavým adresářem je adresář „rps“, který obsahuje jednotlivé moduly Redakčního a publikačního systému.

6.3 Implementace jádra

Jádro nového IS ACSA bylo implementováno v souladu s návrhovými diagramy tříd uvedenými v kapitole 5.5. V této části jsou uvedeny některé implementační detaily týkající se jednotlivých komponent jádra.

Pro správné fungování vyžaduje jádro korektní nastavení hodnot následujících globálních proměnných:

- `BASE_PATH` – relativní cesta ze současného umístění do kořenového adresáře systému,
- `CURRENT_DIR` – relativní cesta vedoucí do aktuálního adresáře.

Nastavení několika dalších proměnných je v souboru „./include/config.inc.php“. Jejich hodnoty je nutné měnit pouze při zásadních změnách IS ACSA.

Pro práci s databází je nutné provést konfiguraci připojení, která se provádí v souboru „./include/configDB.inc.php“.

6.3.1 Model

Součástí této komponenty jsou dvě hlavní třídy – `WebModel` a `ISModel`. První má zodpovědnost za práci s veřejnou částí IS ACSA – webové stránky, druhá se stará o privátní část. Tyto třídy jsou příjemcem dat od komponenty `Controller` a zdrojem dat pro komponentu `View`.

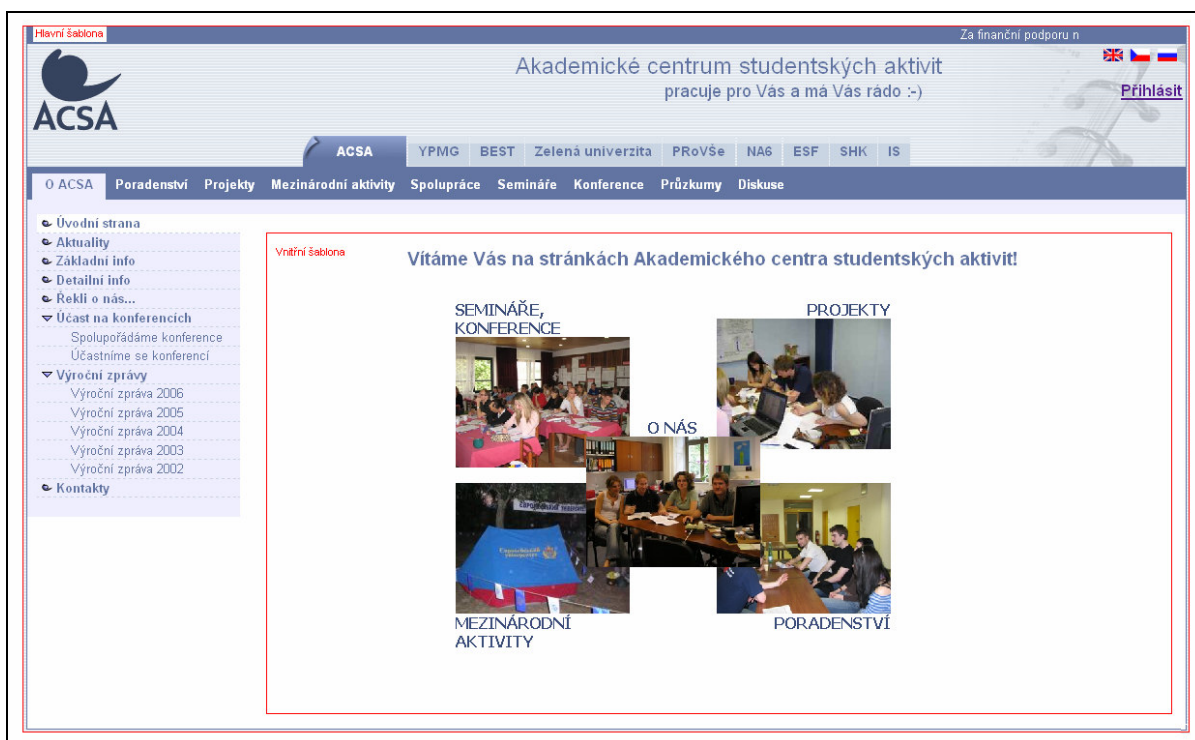
Třída `ISModel` se stará o volání konkrétních akcí modulů IS ACSA a získává od nich data, která jsou následně zobrazována komponentou `View` uživateli.

Dále stojí za zmínku způsob provádění přesměrování. V případě, že třída `ISModel` dojde k závěru, že by měla být provedena jiná akce, vrací třídě `ISController` název této akce a tato akce je následně spuštěna. Tímto je zajištěno provedení přechodů iniciovaných komponentou `Model` (z pohledu uživatele jde o přechod na jinou obrazovku např. po uložení dat apod.).

6.3.2 View

Komponenta View je implementována tak, aby bylo možné využívat více způsobů prezentace systému. Standardně je implementován jednoduchý šablonovací systém pojmenovaný „deftpl“. Tento systém vytváří výstup formou HTML stránek a využívá dva druhy šablon. Hlavní šablona vytváří společné části webové stránky a vnitřní se stará o konkrétní obsah (viz Obrázek 24). Implementace „deftpl“ je v adresáři „/include/core/view/sablony/deftpl“.

Šablony pro systém „deftpl“ obsahují kód jazyka HTML a vložené úseky kódu PHP. V těchto šablonách jsou místo přímého vkládání textu používány konstanty z jazykových souborů – tím je zajištěna jejich schopnost přizpůsobit se více jazykům.



Obrázek 24: Šablony použité pro zobrazení IS ACSA

Vnitřní šablony se liší podle prováděné akce, přísluší ke konkrétním modulům a jsou umístěny uvnitř těchto modulů v adresářích nazvaných „sablony“. Tyto adresáře obsahují kromě adresáře šablon i adresář „lng“ s jazykovými soubory.

Jazykové soubory jsou společné pro všechny typy zobrazení – dojde-li k rozšíření o nový systém zobrazování, bude i tento používat původní jazykové soubory. Výběr jazyka je řízen požadavkem uživatele, který je následně schvalován komponentou Model. V případě, že je nastavena globální proměnná LANG_0 na hodnotu „true“, jsou používány výchozí jazykové soubory definované programátorem. Toto nastavení je vhodné pouze pro vývojáře, kteří doplňují do jazykových souborů nové texty a potřebují je obratem zobrazovat.

6.3.3 Controller

Tato část jádra je vstupní branou do systému, obsahuje dvě hlavní třídy WebController a ISController. Třída WebController spravuje veřejnou část systému a třída ISController část privátní. Starostí této komponenty je správa sezení a nastavení prostředí pro komponenty Model a View, které vytváří a spouští jejich činnost.

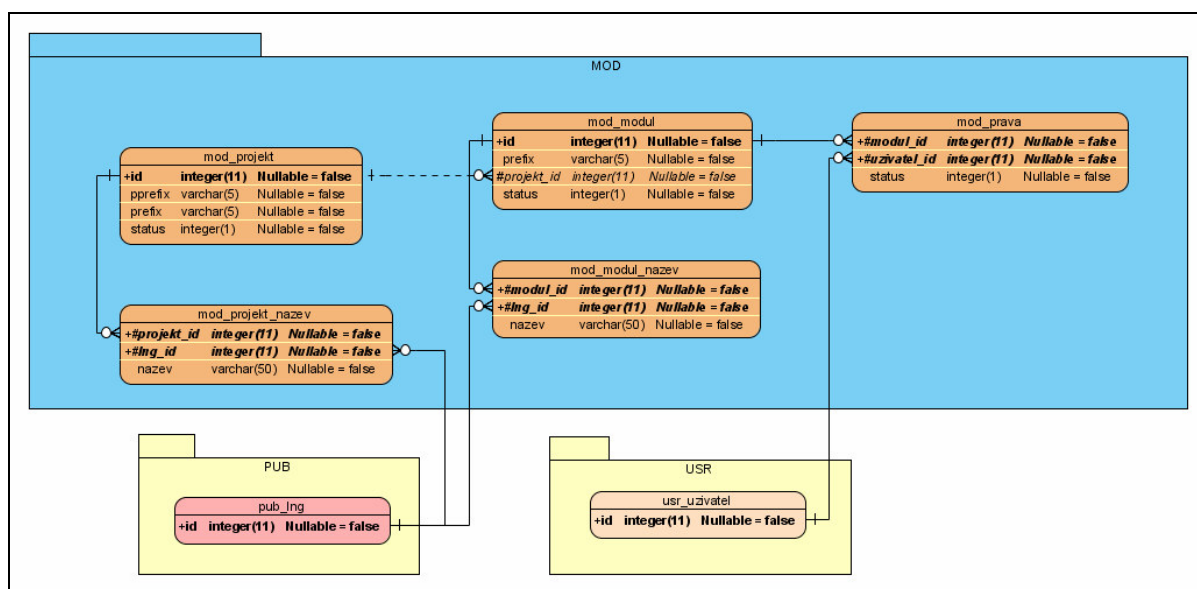
6.4 Implementace modulů

Implementace všech modulů vychází z návrhů uvedených v kapitole 5.6. V případě, že nebyly požadovány změny, byla snaha o to, aby funkcionality jednotlivých modulů byla stejná jako u původní verze systému.

6.4.1 Modul Správa informačního systému

Uživatelské rozhraní tohoto modulu je přístupné pouze správcům systému. Umožňuje jim vytvářet nové projekty a přidávat přístupové moduly k těmto projektům (samozřejmě je nutné provést implementaci těchto modulů).

Tento modul má prefix „ais“ a je k němu vytvořen jediný přístupový modul se zkratkou „ais“ – třídy s implementací přístupového modulu jsou tudíž uloženy v adresáři „/n_is/ais_ais“. Modul operuje nad upravenou databází modulu Rozhraní původního IS ACSA (viz Obrázek 25). Dále do jeho kompetence spadají společné databázové tabulky (viz Příloha 1: Obrázek 30).



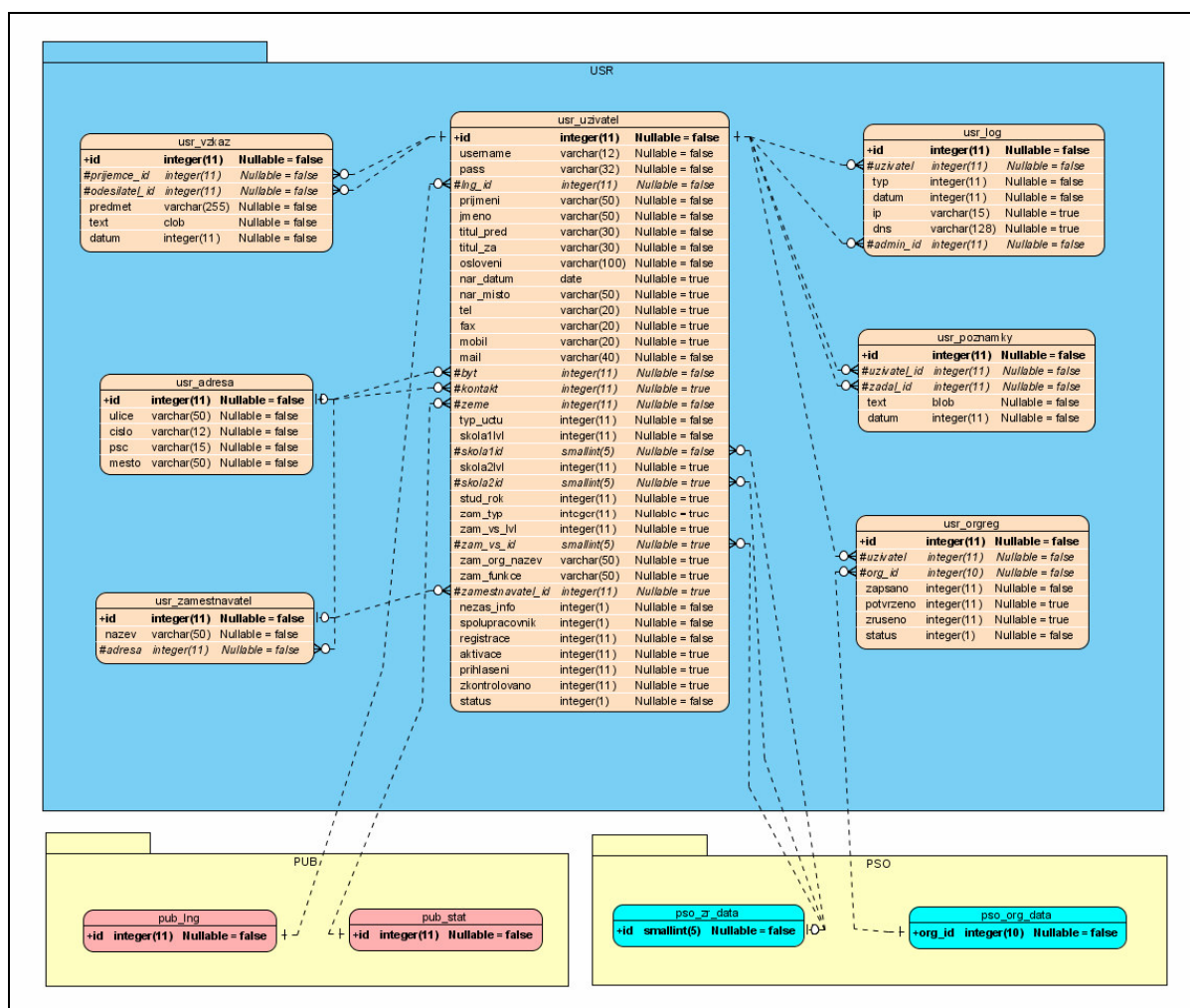
Obrázek 25: Databázové tabulky modulu Správa informačního systému

6.4.2 Modul Správa uživatelů

Správa uživatelů má dva přístupové moduly. Jeden pro správce uživatelů a druhý pro standardního uživatele. Modul umožňuje měnit informace o uživatelích. Správce může také nastavovat práva pro přístup uživatelů k dalším modulům.

Prefix modulu je „usr“ a existují k němu dva přístupové moduly. Jeden je určen pro běžného uživatele, má prefix také „usr“ a jeho implementace je v adresáři „/n_is/usr_usr“. Druhý je určen pro administrátora, prefix je „adm“ a implementace se nachází v adresáři „/n_is/usr_adm“.

Databázové tabulky Správy uživatelů jsou na Obrázku 26.



Obrázek 26: Databázové tabulky modulu Správa uživatelů

6.4.3 Modul Překlady

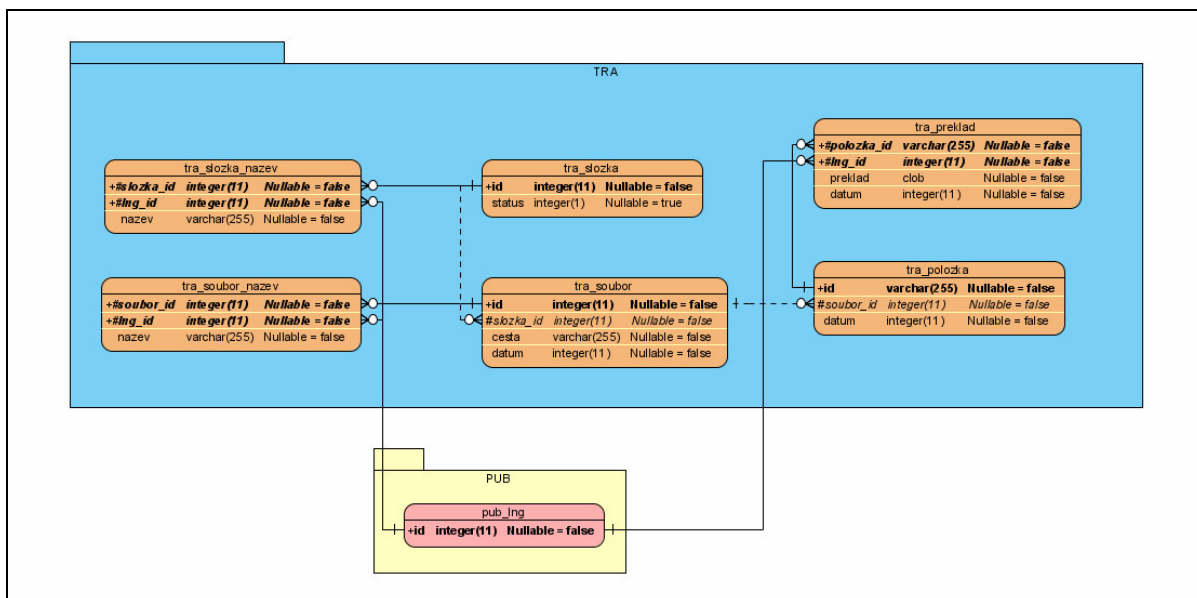
Tento modul má také dva přístupové moduly. Jedná se o přístupový modul pro překladatele, který vkládá překlady textů, a přístup pro správce, který vytváří složky a zadává cesty k umístění jednotlivých souborů se zdrojovými texty.

Prefix modulu je „tra“. Přístupový modul pro překladače má prefix „usr“ a jeho implementace se nachází v adresáři „n_is/tra_usr“. Přístupový modul pro správce má prefix „adm“ a implementace tohoto modulu se nachází v adresáři „n_is/tra_adm“.

V původní verzi modulu byl problém s rychlostí zpracování textových souborů. Výchozí i přeložené soubory obsahovaly množství konstant, kterým byl přiřazen patřičný text. Problém byl především při načítání textů z výchozích jazykových souborů, kdy bylo nutné provést načtení jejich obsahu pomocí regulárních výrazů. V případě velkých souborů byl tento proces značně časově náročný. Databázové schéma modulu Překlady je na Obrázku 27.

V nové verzi došlo k rozdělení velkých jazykových souborů na menší, které mají souvislost s konkrétní šablonou. Dále bylo přistoupeno ke změně formátu výchozích jazykových souborů. Protože v současné verzi PHP již není problém pracovat s XML, budou nové zdrojové soubory právě v tomto formátu. Soubory s překlady v jednotlivých jazycích zůstanou ve formátu původním. Dále je změněno pojmenování souborů, které se nyní vztahuje k šabloně, ke které jazykový soubor patří. Název souboru je vytvořen podle pravidla: „název šablony“_lng_“id jazyka“.php. Příklad jazykových souborů je obsahem Přílohy 2.

Důležitou podmínkou je zachování současných názvů jednotlivých položek z původních souborů, neboť právě tyto názvy představují vazby k jejich současným překladům v databázi. Dále je důležité vytvářet unikátní názvy jednotlivých položek. Z praktického pohledu programátora je také vhodné, aby název položky vyjadřoval její obsah. Při vytváření položek nových bude jejich název tvořen následovně: „prefix projekt“_“prefix_modul“_“název šablony“_“název položky“.



Obrázek 27: Databázové tabulky modulu Překlady

6.4.4 Modul Správa webové prezentace

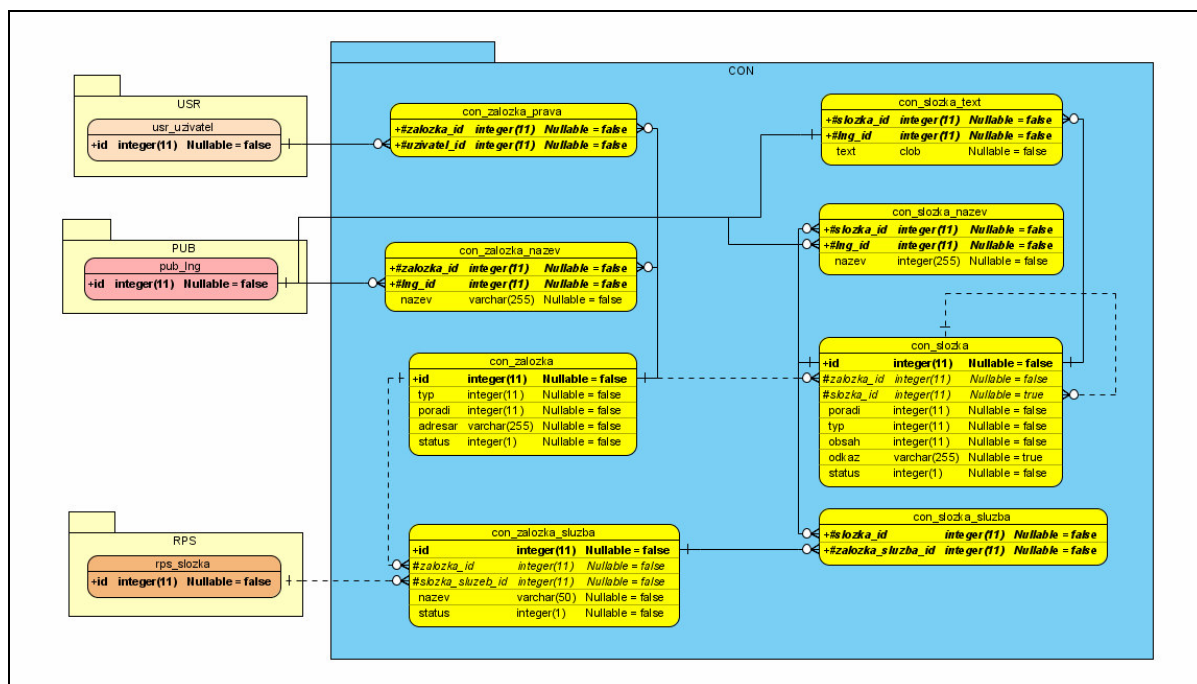
Tento modul obsahuje také dva přístupové moduly. První slouží k administraci záložek – jejich vytváření, registraci služeb a přidělování správců záložky, a druhý slouží příslušným správcům záložky k editaci záložky jim přidělené.

Prefix modulu je „con“. Přístupový modul pro administrátora má prefix „adm“ a jeho implementace se nachází v adresáři „n_is/con_adm“. Přístupový modul pro správce záložky má prefix „usr“ a implementace tohoto modulu se nachází v adresáři „n_is/con_usr“.

Na každé záložce je možné zaregistrovat si služby Redakčního systému a tyto služby v rámci této záložky využívat. Uživatelské rozhraní služeb je možné umístit do libovolné části záložky. Administrační rozhraní je součástí rozhraní pro editaci záložky.

Vytváření obsahu jednotlivých stránek je možné dvěma způsoby – jednak pomocí WYSIWIG editoru anebo ručně vytvořit šablonu a v systému zadat její umístění.

Databázové schéma je na Obrázku 28.



Obrázek 28: Databázové tabulky modulu Správa webové prezentace

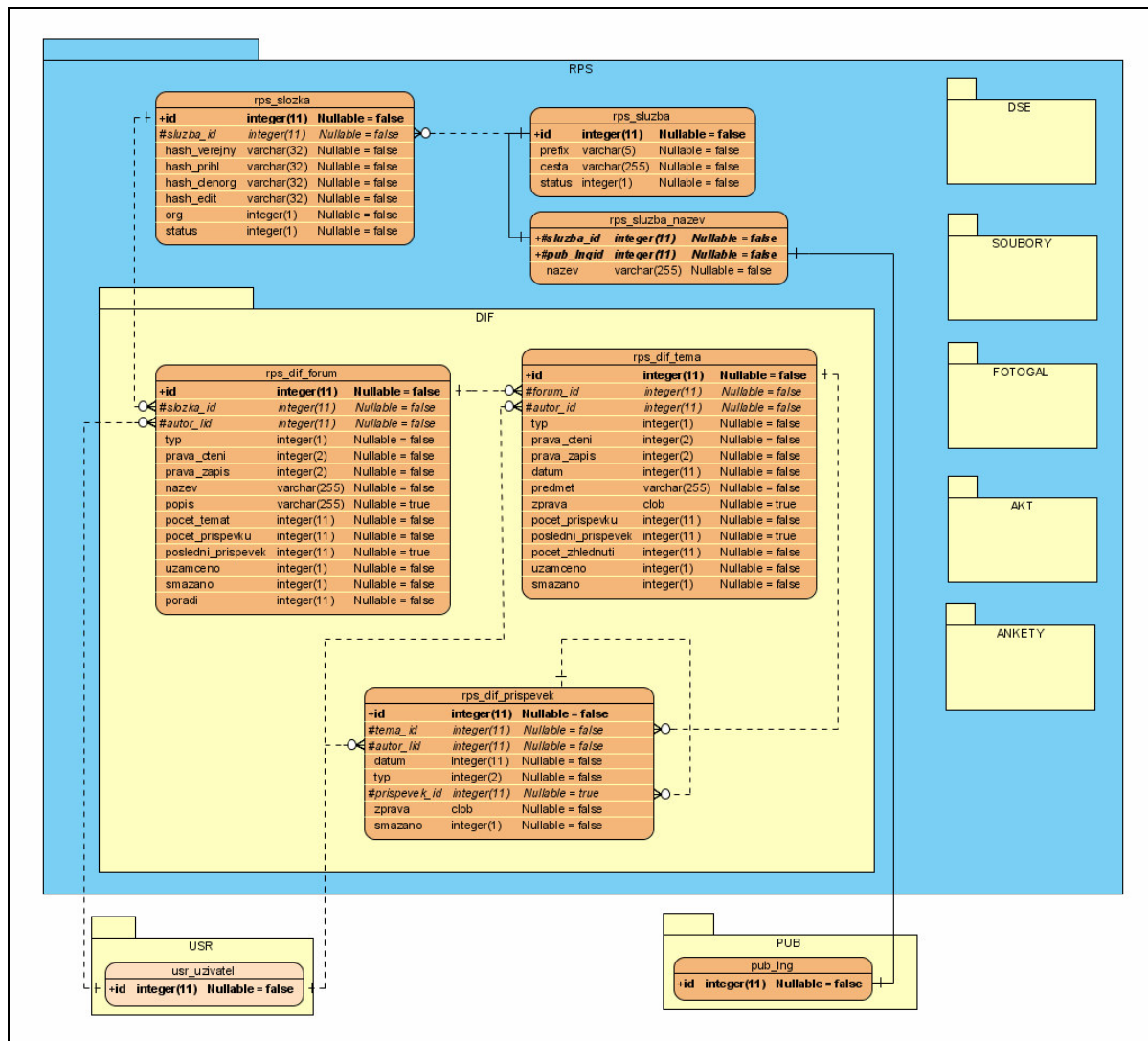
6.4.5 Modul Redakční a publikační systém

K modulu Redakčního a publikačního systému existuje pouze jeden přístupový modul určený správcům, kteří mohou na jednom místě provádět správu všech zaregistrovaných služeb.

Prefix tohoto modulu je „rps“. Implementace jednotlivých služeb je v podadresářích adresáře „rps“ – služby mají vlastní prefixy a jim odpovídající adresáře. Přístupový modul pro správce je v adresáři „n_is/rps_adm“.

Schéma databáze tohoto modulu je na Obrázku 29. Z důvodu lepšího pochopení integrace jednotlivých služeb obsahuje také databázové schéma služby Diskusní fóra.

Implementace jednotlivých služeb Redakčního a publikačního systému přesahuje rámec této diplomové práce.



Obrázek 29: Databázové tabulky modulu Redakční a publikační systém

7 Ověření systému na reálných datech

Otestování funkčnosti implementovaného řešení je provedeno na datech získaných z „ostré“ databáze původního systému IS ACSA. Tato data je nutné transformovat do podoby vhodné pro systém nový a také je vhodné odstranit osobní data uživatelů tak, aby nemohlo dojít k jejich zneužití.

7.1 Úprava dat původního systému

Protože došlo k určitým změnám v databázových tabulkách oproti původnímu systému, je nutné připravit skript, který dokáže z databáze původního systému vytvořit databázi vhodnou pro nový systém. Tento skript nesmí v žádném případě jakkoli poškodit informace obsažené v databázi. Více informací o tomto skriptu viz Příloha 4.

Další úpravou dat ostrého systému je „anonymizace“ osobních informací uživatelů tak, aby nebylo jednoduše zjistitelné, o jakého uživatele se jedná. Tento proces proběhne nad daty modulu Správa uživatelů. Jedná se o nahrazení údajů uvedených v Tabulce 1. Skript provádějící tuto transformaci je také součástí Přílohy 4.

Ačkoli i po této úpravě zůstávají některé osobní informace nezměněné, jejich spojení s konkrétní osobou je již značně znesnadněno.

Informace	Nová hodnota
Uživatelské jméno	user_1, user_2, user_3, ...
Heslo	acsatest
Příjmení	Příjmení1, Příjmení2, ...
Jméno	Jméno1, Jméno2, ...
Oslovení	Vážený uživateli
Datum narození	Původní datum změněné o pseudonáhodnou hodnotu.
Telefon, fax, mobil	+420123456789
Email	user_X@acsa.vutbr.cz
Adresa	Adresa ACSA.

Tabulka 1: Úprava osobních dat uživatelů

7.2 Testování nového systému

Jednotlivé součásti systému byly v průběhu implementace samostatně testovány. Po přidání nových částí bylo provedeno integrační testování, kterým bylo ověřeno fungování nového systému jako celku. Následovat by mělo akceptační testování zákazníkem. V tomto případě je zákazníkem ACSA

a zažitý postup při přebírání nových částí informačního systému ke skutečnému používání je takový, že konkrétní uživatelé (obvykle zaměstnanci této organizace) jsou seznámeni s novou součástí a po určitém čase testování reportují své zkušenosti a požadavky. Tyto poznatky jsou následně řešeny s vedením ACSA a v případě, že jsou schváleny, je přistoupeno k jejich implementaci do systému. Postup provádění přejímacích testů, které budou uskutečněny po dokončení této práce, bude mít následující charakter:

1. seznámení uživatelů s novým systémem,
2. stanovení doby testování systému uživateli,
3. po uplynutí doby testování budou získány jejich připomínky a požadavky,
4. konzultace připomínek a požadavků s vedením,
5. implementace schválených připomínek a požadavků.

Výše uvedený proces by měl zajistit splnění odůvodněných požadavků a také zabránit různým nevhodným změnám systému, protože ke změnám bude přistoupeno po projednání možností změn a dopadů jejich realizace.

8 Možná rozšíření

V této kapitole jsou navrženy možnosti rozšíření nového IS ACSA, které byly objeveny v průběhu zpracování této práce. Při budoucích úvahách o dalším rozšiřování nového IS ACSA je tato kapitola vhodným startovním místem.

8.1 Nové uživatelské rozhraní

Vzhledem k tomu, že návrh komponenty View jádra systému je proveden tak, aby bylo možné využívat různé způsoby zobrazení poskytovaných dat, patří mezi rozšíření i návrh implementovat nové způsoby prezentace informací.

Je možné implementovat zobrazování pomocí některého z rozšířených šablonovacích systémů, které nabízejí zcela jistě větší možnosti než systém aktuálně použitý. Jako příklad lze uvést velice populární systém Smarty¹².

Při rozšiřování uživatelského rozhraní je možné uvažovat i o použití systému v mobilních zařízeních a vytvoření odpovídajících šablon právě pro tato zařízení.

Dalším rozšířením uživatelského rozhraní může být zobrazení vhodné pro tisk, které lze velmi výhodně spojit s generováním výstupů ve formátu dokumentů PDF¹³.

8.2 Překlad dynamických textů

IS ACSA obsahuje velké množství textů vkládaných uživateli. Tyto texty jsou uloženy na různých místech v databázi. Obvykle se jedná o tabulky, které obsahují položku s textem a další položku určující jazyk tohoto textu. Překlad těchto textů je obvykle možný v patřičných modulech. Bylo by velmi vhodné, kdyby k těmto textům existoval přístup z modulu Překlady. Dále nastíním dva možné přístupy k vyřešení tohoto problému.

Jednou možností je registrovat patřičné tabulky a odpovídající sloupce v překladovém modulu a překlad provádět přímo nad danou tabulkou. Tento přístup však nezachytí možné souvislosti mezi textovými daty.

Druhou možností je vytvoření rozhraní mezi modulem Překlady a jednotlivými moduly, které mají zájem o přeložení vlastních textů. Tento způsob by měl umožnit zachycení vazeb mezi překládanými texty, jeho realizace je však složitější, než je tomu u varianty první.

Na závěr jen dodejme, že zcela jistě existuje i mnoho dalších variant řešení tohoto problému.

¹² více informací lze nalézt v [25]

¹³ PDF – Portable Document Format, více informací lze nalézt v [26]

8.3 Redakční systém založený na AJAXu¹⁴

Aby bylo možné používat služby Redakčního systému v libovolné části IS ACSA, funguje tento systém uvnitř rámců vkládaných na požadovaná místa v systému. Tato metoda je funkční, ale může v některých případech působit problémy.

Vzhledem k tomu, jak je navrženo fungování Redakčního systému, který je do velké míry nezávislý, neměl by být problém realizovat jej pomocí technologie AJAX. Vkládání rámců do stránek by bylo nahrazeno pomocí HTML prvků DIV a obsah by byl získáván pomocí asynchronní komunikace se serverem.

8.4 Automatizované komponenty

V průběhu implementace se ukázalo, že by bylo vhodné implementovat určité typy komponent s automatickým chováním – zobrazením a zpracováním jejich požadavků. Týká se to především seznamů entit a formulářů pro editaci daných entit.

V prvním případě jde o tabulku obsahující záznamy daného typu, která nabídne tyto akce:

- seřazení záznamů podle vybraného sloupce,
- přechod k editaci vybraného záznamu,
- odstranění vybraného záznamu,
- zobrazení podrobností záznamu,
- listování v záznamech v případě jejich velkého počtu.

Ve druhém případě jde o automatické vytváření a zpracování formulářů, které nabídne pokročilé kontroly vyplnění formuláře pomocí vlastních funkcí.

8.5 Personalizace systému

Některé části IS ACSA, jako např. výchozí stránka v privátní části, by mohly být uživatelsky přizpůsobitelné. Rozdělení každé takovéto stránky na určité oblasti a ponechání na uživatelově volbě, jaká data se v dané oblasti zobrazí, by mohlo práci v systému velmi zpříjemnit.

Realizace změn nutných pro personalizaci systému by neměla představovat větší problém. Důležitým rozhodnutím však je, které části systému personalizovat a jaké informace v těchto částech nabízet.

¹⁴ AJAX – Asynchronous Javascript and XML, více informací lze nalézt v [27]

9 Závěr

Cílem této diplomové práce bylo provedení návrhu a implementace základů nové verze IS ACSA s ohledem na zajištění co nejlepší udržitelnosti. Největším nedostatkem původní verze IS ACSA je právě její obtížná údržba, která je dána především neexistující dokumentací. Na základě této skutečnosti byl kladen velký důraz na zdokumentování návrhu a následné implementace nové verze IS ACSA. Hlavní přínos nové verze IS ACSA není v oblasti poskytované funkčnosti, nýbrž v oblasti dokumentace, struktury a možností další rozšiřitelnosti tohoto systému.

Práce na tomto projektu probíhala v průběhu dvou semestrů, kdy nejdříve byl zpracován semestrální projekt, na nějž navazuje tato diplomová práce. V rámci semestrálního projektu bylo provedeno studium témat týkajících se údržby software, používaných architektur a důkladná analýza původního IS ACSA. Na základě těchto poznatků byl zvolen architektonický rámec MVC, který posloužil k návrhu a implementaci nové verze IS ACSA.

V rámci diplomové práce byl nejprve vytvořen detailní návrh jádra nové verze IS ACSA. Následoval návrh modulu schopného spolupracovat s novým jádrem. V souladu s návrhem tohoto modulu byly provedeny návrhy modulů Správa informačního systému, Správa uživatelů, Správa webové prezentace, Překlady a modul Redakční a publikační systém. Po dokončení návrhu jednotlivých modulů bylo přistoupeno k implementaci jádra a již zmíněných modulů. Pro otestování použitelnosti nové verze IS ACSA byla použita transformovaná data z "ostré" databáze původní verze IS ACSA.

Domnívám se, že cíle, které byly v rámci této práce stanoveny, se mi podařilo splnit. Práce na novém IS ACSA pro mě byla velkým přínosem. Díky ní jsem získal mnoho nových zkušeností a znalostí týkajících se obecně vývoje software. Velkým přínosem pro mě byla také implementace jádra IS ACSA založená na rámci MVC, která ukázala, jak důležité je správně pochopit a navrhnout architekturu nového systému a jaké komplikace mohou způsobit chyby v těchto krocích. Nejvíce však oceňuji možnost účastnit se vývoje skutečného systému, který bude používán v praxi.

Práce na projektu budou pokračovat i po skončení diplomové práce. Ještě není rozhodnuto, zdali budou stávající moduly Portál studentských organizací a Akce integrovány v současném stavu nebo přestavěny od základů v souladu s architekturou nového IS ACSA. V každém případě pro nasazení systému v praxi bude nutné tyto moduly nějakým způsobem integrovat. Další možný rozvoj systému byl nastíněn v 8. kapitole. Praktické použití systému zcela jistě přinese mnoho poznatků o tom, jak by bylo možné a vhodné systém rozšířit.

Seznam obrázků

Obrázek 1: Schéma rámce MVC	18
Obrázek 2: Schéma rámce PCMEF	19
Obrázek 3: Rozdělení uživatelů podle přístupových práv	24
Obrázek 4: Spolupráce mezi jednotlivými moduly IS ACSA	25
Obrázek 5: Hlavní činnosti a uživatelé Rozhraní IS ACSA	26
Obrázek 6: Hlavní činnosti a uživatelé modulu Správa uživatelů	26
Obrázek 7: Hlavní činnosti a uživatelé modulu Překlady	27
Obrázek 8: Hlavní činnosti a uživatelé modulu Správa webové prezentace	28
Obrázek 9: Hlavní činnosti a uživatelé modulu Akce	29
Obrázek 10: Hlavní činnosti a uživatelé Portálu studentských organizací	30
Obrázek 11: Hlavní činnosti a uživatelé modulu Redakčního a publikačního systému	31
Obrázek 12: Nástin komunikace mezi vrstvami MVC a uživatelem IS ACSA.....	34
Obrázek 13: Diagram tříd komponenty Model.....	37
Obrázek 14: Diagram tříd komponenty View.....	38
Obrázek 15: Diagram tříd komponenty Controller.....	39
Obrázek 16: Příklad rozhraní s využitím vzoru Fasáda	40
Obrázek 17: Diagram tříd vzorového modulu nového IS ACSA	41
Obrázek 18: Konceptuální diagram tříd modulu Správa informačního systému.....	42
Obrázek 19: Konceptuální diagram tříd modulu Správa uživatelů.....	43
Obrázek 20: Konceptuální diagram tříd modulu Správa webové prezentace	44
Obrázek 21: Konceptuální diagram tříd modulu Redakční a publikační systém.....	45
Obrázek 22: Konceptuální diagram tříd modulu Překlady	46
Obrázek 23: Adresářová struktura nového IS ACSA	49
Obrázek 24: Šablony použité pro zobrazení IS ACSA.....	51
Obrázek 25: Databázové tabulky modulu Správa informačního systému	52
Obrázek 26: Databázové tabulky modulu Správa uživatelů	53
Obrázek 27: Databázové tabulky modulu Překlady	54
Obrázek 28: Databázové tabulky modulu Správa webové prezentace	55
Obrázek 29: Databázové tabulky modulu Redakční a publikační systém	56
Obrázek 30: Databázové tabulky společné pro celý IS ACSA.....	65

Literatura

- [1] **Maciaszek, L.: Practical Software Engineering: A Case Study Approach.** Pearson, 2005.
- [2] **Kagan, E. a kol.: Software Maintenance as a Part of Software Life Cycle.** Department of Computer Science, Tucte University, 2003. Dokument dostupný na URL http://hepguru.com/maintenance/Final_121603_v6.pdf (květen 2008).
- [3] **Wikipedia. Software maintenance, 2008.** Dokument dostupný na URL http://en.wikipedia.org/wiki/Software_maintenance (květen 2008).
- [4] **Anthony Aaby: Software Maintenance, 2003.** Dokument dostupný na URL <http://moonbase.wwc.edu/~aabyan/435/Maintenance.html> (prosinec 2007).
- [5] **Fowler, M.: Refaktoring: Zlepšení existujícího kódu.** Grada 2003.
- [6] **Sommerville, I.: Software Engineering.** Pearson 2004.
- [7] **Huška, M.: Informační systém Akademického centra studentských aktivit,** [Bakalářská práce]. Brno 2005.
- [8] **Žilka, R.: Redakční a publikační systém Akademického centra studentských aktivit,** [Bakalářská práce]. Brno 2006.
- [9] **Šramka, M.: PHP, MySQL: Modul pro stavbu a zpracování formulářů pro dotazníkovou šetření,** [Bakalářská práce]. Brno 2006.
- [10] **Kolda, J.: MySQL a PHP – informační portál studentských organizací,** [Diplomová práce]. Brno 2005.
- [11] **Wikipedia. Software architecture, 2008.** Dokument dostupný na URL http://en.wikipedia.org/wiki/Software_architecture (květen 2008).
- [12] **Arlow, J.: UML a unifikovaný proces vývoje aplikací.** Computer Press 2003.
- [13] **Wikipedia. Architectural patterns , 2008.** Dokument dostupný na URL [http://en.wikipedia.org/wiki/Architectural_pattern_\(computer_science\)](http://en.wikipedia.org/wiki/Architectural_pattern_(computer_science)) (květen 2008).
- [14] **Fowler, M.: Catalog of Patterns of Enterprise Application Architecture, 2003.** Dokument dostupný na URL <http://martinfowler.com/eaCatalog/> (květen 2008).
- [15] **Gamma, E.: Návrh programů pomocí vzorů.** Grada 2003.
- [16] **Šramka M.: Udržitelná verze informačního systému Akademického centra.** Brno, 2008, [Semestrální projekt], FIT VUT v Brně.
- [17] **Wikipedia. Y2K problem, 2008.** Dokument dostupný na URL <http://en.wikipedia.org/wiki/Y2K> (květen 2008).
- [18] **Návrhové vzory (design patterns), 2008.** Dokument dostupný na URL <http://objekty.vse.cz/Objekty/Vzory> (květen 2008).
- [19] **Wikipedia. WYSIWYG, 2008.** Dokument dostupný na URL <http://en.wikipedia.org/wiki/WYSIWYG> (květen 2008).

- [20] **PHP MVC Frameworks, 2008.** Dokument dostupný na URL http://www.phpwact.org/php/mvc_frameworks (květen 2008).
- [21] **The Model-View-Controller (MVC) Design Pattern for PHP, 2007.** Dokument dostupný na URL <http://www.tonymarston.net/php-mysql/model-view-controller.html> (květen 2008).
- [22] **The no framework PHP MVC framework, 2006.** Dokument dostupný na URL <http://toys.lerdorf.com/archives/38-The-no-framework-PHP-MVC-framework.html> (květen 2008).
- [23] **Template Engines, 2006.** Dokument dostupný na URL http://www.massassi.com/php/articles/template_engines/ (květen 2008).
- [24] **Understanding MVC in PHP, 2005.** Dokument dostupný na URL http://www.onlamp.com/pub/a/php/2005/09/15/mvc_intro.html (květen 2008).
- [25] **Smarty: Template Engine, 2008.** Dokument dostupný na URL <http://www.smarty.net/> (květen 2008)
- [26] **Wikipedia. Portable Document Format, 2008.** Dokument dostupný na URL http://en.wikipedia.org/wiki/Portable_Document_Format (květen 2008).
- [27] **Darie a kol.: AJAX a PHP – tvoříme interaktivní webové aplikace.** Zoner Press 2006.
- [28] **PHP Documentator, 2008.** Dokument dostupný na URL <http://www.phpdoc.org> (květen 2008).

2 Jazykové soubory

2.1 Formát původního jazykového souboru

V původním IS ACSA byl tento formát použit jak pro výchozí, tak pro přeložené jazykové soubory. V nové verzi byl zachován pouze u souborů přeložených, kde je výhodný. Tyto soubory jsou vkládány do šablon, které na místech obsahujících text používají takto definované konstanty.

```
<?
DEFINE('ais_ais_projekt_edit_nadpis_projekty','Editace projektu');
DEFINE('ais_ais_projekt_edit_nadpis_moduly','Editace modulu');
DEFINE('ais_ais_projekt_edit_nazev','Název');
DEFINE('ais_ais_projekt_edit_preklady','Překlady');
DEFINE('ais_ais_projekt_edit_jazyk','Jazyk');
DEFINE('ais_ais_projekt_edit_ulozit','Uložit');
DEFINE('ais_ais_projekt_edit_zpet','Zpět');
?>
```

2.2 Formát nového jazykového souboru

V novém IS ACSA mají výchozí jazykové soubory, které vytváří programátor, následující formát:

```
<?xml version="1.0" encoding="utf-8"?>
<lngsoubor name="nazev_souboru" lng="id_jazyka">
  <polozka id="nazev_polozky1">Text1</polozka>
  <polozka id="nazev_polozky2">Text2</polozka>
</lngsoubor>
```

3 Zdrojové kódy

Zdrojové kódy všech částí této práce jsou uloženy na přiloženém CD. Jedná se především o následující:

- zdrojový soubor tohoto textu,
- programové zdrojové kódy,
- databázové skripty,
- dokumentace.

Přesné umístění jednotlivých částí je uvedeno v souboru „ReadMe“, který je umístěn v kořenovém adresáři přiloženého CD.

4 Databázové SQL skripty

Vzhledem k nutnosti úprav databázové struktury i dat samotných byly vytvořeny skripty zajišťující tuto činnost. Dále byl vytvořen také skript sloužící pro jednoduchou „anonymizaci“ uživatelských informací.

Jednotlivé skripty jsou k dispozici na přiloženém CD v adresáři „sql“, který je v kořenovém adresáři. V tomto adresáři jsou soubory s následujícími skripty:

- newisacsa.sql – vytvoření databáze pro nový IS ACSA,
- oldDB_transfrom.sql – transformace původní databáze, pro použití s novým IS ACSA,
- anonym.sql – odstranění uživatelských informací,
- transformační skripty rozdělené pro jednotlivé moduly.

Konkrétní data sloužící pro naplnění databáze nejsou přiložena, neboť se jedná o citlivé informace, ve vlastnictví ACSA, které není vhodné a možné dále distribuovat.

5 Dokumentace IS ACSA

Dokumentace byla vytvořena programem phpDocumentator a je k dispozici na přiloženém CD v adresáři „doc“, který se nachází v kořenovém adresáři. Dokumentace je ve formátu HTML, výchozí soubor „index.html“ je také v již uvedeném adresáři.

6 Testovací verze

Testovací verze nového IS ACSA je dostupná na adrese <http://pc-acsaserver.ro.vutbr.cz/newisacsa/>.

Pro testovací účely je možné se přihlásit pod uživatelskými jmény a hesly (jméno:heslo):

- admin:test,
- uzivatel:test,
- zalozka:test,
- prekladatel:test.

Uživatel „admin“ má přístup ke všem částem nového IS ACSA. Uživatel „uzivatel“ je standardním uživatelem IS ACSA, který má přístup pouze ke svému vlastnímu účtu. Uživatel „zalozka“ je správcem určité záložky a může měnit její obsah. Uživatel „prekladatel“ může vkládat překlady textů.

Dále je k dispozici Wiki systém na adrese <http://pc-acsaserver.ro.vutbr.cz/newisacsa/wiki/>, ke kterému je možné se přihlásit pomocí uživatelského jména „usertest“ a hesla „test“. Tento systém obsahuje některé další informace týkající se IS ACSA.

Tato testovací verze IS ACSA bude dostupná až do obhajoby této práce. V případě problémů s dostupností testovací verze nebo i jiných je možné kontaktovat autora pomocí emailu xsramk00@stud.fit.vutbr.cz.