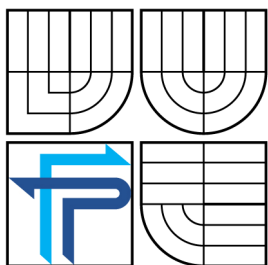


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA PODNIKATELSKÁ**

**ÚSTAV MANAGEMENTU**

FACULTY OF BUSINESS  
INSTITUTE OF MANAGEMENT

## NÁVRH INFORMAČNÍHO SYSTÉMU

INFORMATION SYSTEM PROPOSAL

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MGR. PETR VÝMOLA**

**VEDOUcí PRÁCE**

SUPERVISOR

**DOC. ING. MILOŠ KOCH, CSC.**

BRNO 2010

## **ZADÁNÍ DIPLOMOVÉ PRÁCE**

Výmola Petr, Mgr.

---

Řízení a ekonomika podniku (6208T097)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává diplomovou práci s názvem:

**Návrh informačního systému**

v anglickém jazyce:

**Information System Proposal**

Pokyny pro vypracování:

Úvod  
Cíle práce, metody a postupy zpracování  
Teoretická východiska práce  
Analýza problému  
Vlastní návrhy řešení  
Závěr  
Seznam použité literatury  
Přílohy

---

Podle § 60 zákona č. 121/2000 Sb. (autorský zákon) v platném znění, je tato práce „Školním dílem“. Využití této práce se řídí právním režimem autorského zákona. Citace povoluje Fakulta podnikatelská Vysokého učení technického v Brně. Podmínkou externího využití této práce je uzavření „Licenční smlouvy“ dle autorského zákona.

## Seznam odborné literatury

BASL, Josef, BLAŽÍČEK, Roman. Podnikové informační systémy: Podnik v informační společnosti – 2. výrazně přepracované a rozšířené vydání. 2008. vyd. Praha : Grada Publishing, a.s., 2008. 283 s. ISBN-978-80-247-2279-5.

MOLNÁR, Zdeněk. Efektivnost informačních systémů. 1. vyd. Praha : Grada Publishing, spol. s r.o., 2000. 144 s. ISBN 80-7169-410-X.

SODOMKA, Petr. Informační systémy v podnikové praxi. 1. vyd. Brno : Computer Press, a.s., 2006. 351 s. ISBN 80-251-1200-4.

DOSTÁL, Petr, RAIS, Karel, SOJKA, Zdeněk. Pokročilé metody manažerského rozhodování. 1. vyd. Praha : Grada Publishing, a.s., 2005, 168 s. ISBN 80-247-1338-1.

Vedoucí diplomové práce: doc. Ing. Miloš Koch, CSc.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2009/2010.

L.S.

---

PhDr. Martina Rašticová, Ph.D.  
Ředitel ústavu

---

doc. RNDr. Anna Putnová, Ph.D., MBA  
Děkanka

V Brně, dne 7.2.2010

## **Abstrakt**

S rostoucím počtem služeb, které Masarykova univerzita poskytuje svým studentům a s měnící se legislativou se jako nutnost ukazuje implementovat nový informační systém pro zpracování plateb za služby poskytované na univerzitě. Cílem této práce je zanalyzovat současný stav, zvážit požadavky uživatelů, vybrat vhodný informační systém a navrhnout jeho implementaci. Je nezbytně nutné, aby se náklady na zavedení nového systému udržely v přijatelných mezích a především aby systém byl provozovatelný při zachování současného počtu zaměstnanců. Je požadována univerzalita systému z pohledu rozličného charakteru poskytovaných služeb a prodávaného zboží, stejně jako jeho vysoký výkon. Systém musí být schopen zvládat nárůst jak počtu klientů, tak služeb.

## **Abstract**

Because of a growing number of services offered by the Masaryk University to its students, and because of the ever-changing legislation of the Czech Republic, a need for an implementation of a new information system, capable of processing payments for university's services, has arisen. The goal of this thesis is to analyse the current condition of payments at the university, consider users' demands, choose an appropriate information system and project its implementation. It is utterly necessary to keep the costs of implementation at acceptable levels and above all the system must be able to be operated without the need for new employees. The proposed system must be universal because of various natures of services offered and goods sold, as well as it must be efficient. It is required that the system will be able to handle the growth of both number of clients and provided services.

## **Klíčová slova**

Jednotné univerzitní konto, zpracování plateb, úhrady, pohledávka, dobropis, kauce, daň z přidané hodnoty, operativní účetní evidence, automatizace vytváření účetních dokladů

## **Keywords**

Unified university account, payments processing, settlements, claim, credit note, bail, value added tax, operative accounting evidence, automatization of the creation of accounting documents

## **Čestné prohlášení**

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně pod vedením doc. Ing. Miloše Kocha, CSc.

Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 31. května 2010

.....

## **Poděkování**

Na tomto místě bych rád poděkoval svému vedoucímu, doc. Ing. Miloši Kochovi, CSc., za poskytnuté vedení při zpracovávání diplomové práce, stejně jako své rodině, přátelům a spolupracovníkům za trpělivost a neustálou podporu při mém nedostatku času na ně v celém průběhu mého studia.

## **Bibliografická citace práce**

VÝMOLA, P. *Návrh informačního systému* . Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2010. 83 s. Vedoucí diplomové práce doc. Ing. Miloš Koch, CSc.

# Obsah

|                                                                        |    |
|------------------------------------------------------------------------|----|
| Úvod.....                                                              | 10 |
| 1 Cíle práce, metody a postupy zpracování .....                        | 12 |
| 1.1 Okolí systému .....                                                | 12 |
| 1.1.1 IS MU .....                                                      | 12 |
| 1.1.2 IS Magion .....                                                  | 13 |
| 1.1.3 Inet MU .....                                                    | 13 |
| 2 Teoretická východiska práce .....                                    | 15 |
| 2.1 Informační systém.....                                             | 15 |
| 2.2 Vývoj podnikových IS .....                                         | 16 |
| 2.3 Datový pohled na informační systémy .....                          | 19 |
| 2.4 Procesní pohled na informační systémy .....                        | 20 |
| 2.5 Efektivnost informačních systémů.....                              | 23 |
| 2.6 Přínosy informačních systémů a technologií .....                   | 25 |
| 2.6.1 Finanční ukazatele .....                                         | 25 |
| 2.6.2 Nefinanční měřitelné ukazatele .....                             | 26 |
| 2.6.3 Nekvantifikovatelné ukazatele.....                               | 26 |
| 2.7 Lidský aspekt efektivnosti informačních systémů a technologií..... | 27 |
| 2.8 Projekty zavádění informačních systémů do podniků .....            | 29 |
| 2.9 Datová, aplikační a prezentační vrstva .....                       | 32 |
| 2.10 Klient – server architektura.....                                 | 34 |
| 2.11 Data mining.....                                                  | 35 |
| 3 Analýza problému.....                                                | 38 |
| 3.1 Clearing.....                                                      | 38 |
| 3.2 Clearing 2.....                                                    | 39 |
| 3.3 Zhodnocení současného stavu .....                                  | 42 |
| 4 Vlastní návrhy řešení .....                                          | 44 |
| 4.1 Varianta 1: Dodání systému externím dodavatelem .....              | 44 |
| 4.1.1 Silné stránky .....                                              | 44 |
| 4.1.2 Slabé stránky.....                                               | 45 |
| 4.1.3 Příležitosti .....                                               | 45 |
| 4.1.4 Hrozby .....                                                     | 45 |
| 4.1.5 Zhodnocení varianty č. 1 .....                                   | 46 |
| 4.2 Varianta 2: Vytvoření systému vlastními silami.....                | 46 |
| 4.3 Návrh systému .....                                                | 47 |
| 4.3.1 Hrazené služby.....                                              | 48 |
| 4.3.2 Podrobnější pohled na služby .....                               | 50 |
| 4.3.3 Návrh jádra systému .....                                        | 51 |
| 4.3.4 Externí systémy.....                                             | 53 |
| 4.3.5 Položky .....                                                    | 53 |
| 4.3.6 Komunikace s okolními systémy.....                               | 56 |
| 4.3.7 Aplikace – součásti systému – pro uživatele .....                | 57 |
| 4.3.8 Generování inkasa a srážek ze mzdy .....                         | 59 |
| 4.3.9 Provázání systému s účetnictvím.....                             | 61 |
| 4.3.10 Podklady pro odvod daní .....                                   | 62 |
| 4.3.11 Správa klientských dluhů.....                                   | 65 |

|                                                |    |
|------------------------------------------------|----|
| 4.3.12 Bezpečnost systému .....                | 65 |
| 4.3.13 Zodpovědnost za procesy v systému ..... | 68 |
| 4.4 Porovnání s jinými systémy .....           | 69 |
| 4.5 Vznik systému.....                         | 71 |
| 5 Závěr .....                                  | 73 |
| 5.1 Přínos řešení.....                         | 73 |
| 5.2 Výhledy do budoucna .....                  | 75 |
| 5.3 Zkušenosti z provozu .....                 | 76 |
| 5.4 Závěrečné zhodnocení .....                 | 77 |
| 6 Seznam použité literatury .....              | 79 |
| 6.1 Tištěné zdroje.....                        | 79 |
| 6.2 Elektronické zdroje .....                  | 80 |
| Seznam tabulek .....                           | 81 |
| Seznam obrázků .....                           | 82 |
| Seznam příloh .....                            | 83 |

## Úvod

Pro každou společnost či organizaci s více než několika zaměstnanci je v současné době již téměř nezbytné mít informační systém, který by vhodným a efektivním způsobem podporoval hlavní činnost firmy.

V případě, že se jedná o soukromou společnost, se v praxi jako zcela dostačující ukazují různé ERP systémy, většinou dodávané jako již hotový produkt; stejný systém, případně jeho části, používá celá řada různých subjektů. Výhodou je vyzkoušenost celého řešení a s tím se pojící relativní časová a finanční nenáročnost nasazení do provozu. Nevýhodou pak je nepružnost – ať při řešení problémů v provozu, implementaci nových funkcí nebo změnách nevyhovujících částí.

Druhou možností je vytvoření celého informačního systému „na klíč“ – rozšířením již existujícího o požadovanou funkčnost nebo naprogramováním zcela nového řešení. V případě takové organizace, jakou je Masarykova univerzita, tedy organizace se zhruba čtyřmi tisíci zaměstnanci a třiceti tisíci studenty – potenciálními uživateli informačního systému, je tento způsob prakticky jedinou možností.

V této práci popíšeme vznik a nasazení informačního systému, sloužícího z pohledu studentů jako elektronická peněženka, kterou mohou platit za různé služby poskytované univerzitou; z pohledu ekonomů univerzity pak systém slouží jako operativní evidence účetnictví, což s sebou kvůli nutnosti vyhovění daňové legislativě nese mnoho omezujících požadavků.

V první části si stručně vymezíme problém, před kterým Masarykova univerzita nedávno stála, popíšeme existující okolí systému a tuto část zakončíme nastíněním možných řešení.

Dále budeme pokračovat stručným teoretickým úvodem do řešeného problému – vymezíme si pojem informační systém a další, které se úzce dotýkají problému, před kterým stojíme.

Poté budeme moci přistoupit již k samotné analýze výchozího stavu, zohledníme již existující části informačního systému stejně jako jeho předchůdce a přesně vymezíme požadavky na systém, uvedené v předchozí kapitole. Z těchto poznatků budeme moci plynule pokračovat do další části, ve které již konkrétně popíšeme zvolené řešení a vznik samotného systému. Dále pak uvedeme poznatky z nasazení a provozu, stejně jako problémy, se kterými se systém potýká.

V závěrečné části pak již jen zhodnotíme přínosy a nedostatky zvoleného řešení a porovnáme je s výsledky odhadovanými před samotným rozhodováním. Rovněž si dovolíme krátký pohled do budoucnosti na další výzvy, které nás při provozu systému očekávají.

# 1 Cíle práce, metody a postupy zpracování

Základním cílem této práce je navrhnout, implementovat a poté posoudit v ostrém provozu informační systém, který bude splňovat následující požadavky:

- bude plně integrovaný s již existujícími informačními systémy na univerzitě
- bude splňovat veškeré zákonné požadavky, zejména náležitosti zákon č. 235/2004 Sbírky, o dani z přidané hodnoty a zákona č. 563/1991 Sbírky, o účetnictví
- co nejvíce usnadní samotný proces účtování
- bude respektovat zvyklosti z dříve provozovaného systému, ze kterého do něj budou importována data
- bude snadno použitelný pro klienty
- bude zvládat narůstající počty klientů i transakcí

Systém by měl být navržen co nejobecněji tak, aby se dokázal vypořádat jak se změnou daňových zákonů (zejména operování s výšemi základní a snížené sazby DPH), tak i s vývojem ostatních systémů na univerzitě, se kterými musí komunikovat.

Vzhledem k tradici a jisté prestiži spojené s vývojem vlastních informačních systémů je možné předeslat, že bude zvoleno řešení implementace vlastního systému – dodávka hotového řešení bude také zvážena, nicméně velmi rychle zamítnuta.

## 1.1 Okolí systému

Rozeberme si nyní první požadavek z předchozího seznamu. Aby bylo možné jej splnit je nutné vědět o jaké systémy se jedná. Ve stručnosti si je proto popíšeme.

### 1.1.1 IS MU

První z informačních systémů provozovaných na univerzitě zařazujeme spíše jen pro úplnost. Jedná se o informační systém pro podporu studijních záležitostí jak pro učitele, tak pro studenty, jsou v něm tedy informace o zapsaných předmětech, složených zkouškách či rozvrhy hodin.

Jediná oblast, kterou by bylo možné zvažovat při nasazení nového systému, je platba za studium (ať již při překročení standardní doby studia či při dalším studiu na úrovni

již dosaženého titulu). Zde se ovšem v průběhu jednání objevil velký problém se splátkovým kalendářem. Protože školné je v případě Masarykovy univerzity poměrně vysoké, mnoho studentů jej nemůže nebo nechce zaplatit naráz a sjednávají si splátkové kalendáře. To samé platí pro případné zpětně vyměřené školné, kde se uvedený problém násobí.

Z hlediska daní a systému je však vhodné, aby se jednalo o jednu pohledávku za studentem, ke které se váže množina splátek – každá s vlastní splatností a (teoreticky) s odlišnou částkou od ostatních.

Sám o sobě by tento požadavek nebyl neřešitelný, bohužel se však objevil až po nasazení systému do provozu. Za jeho nezahrnutím do počáteční analýzy tak můžeme hledat jak nedbalost, tak i jisté politikaření – za vývojem studijního systému IS MU stojí uzavřená skupina, která nemá s vývojem ekonomických informačních systémů žádné personální vazby a je občas ochotna si prosazovat za každou cenu své řešení, či implementaci již jednou implementovaného do svého systému.

### **1.1.2 IS Magion**

Druhým na univerzitě provozovaným systémem je IS Magion. Jedná se o komerční řešení stejnojmenné zlínské firmy, určené především pro vysoké školy – používá jej zhruba deset českých vysokých škol, které se zároveň podílejí na vývoji tohoto systému předáváním poznatků firmě Magion a testováním implementovaných řešení.

Jedná se o systém skládající se ze dvou hlavních částí. První z nich je Personální a mzdový systém. Jak již název napovídá, tato část systému slouží pro ekonomické pracovníky pro správu zaměstnanců, výpočty jejich mezd, odvody sociálního a zdravotního pojištění a podobně.

Druhou částí je Ekonomický informační systém Magion, sloužící především pro samotné účtování. Je také využíván pro realizování veškerých toků peněz směrem z univerzity a na univerzitu, je v něm implementována příjmová i výdajová pokladna a bankovní rozhraní.

### **1.1.3 Inet MU**

Posledním relevantním na univerzitě provozovaným systémem je Intranet MU – zkracovaný Inet MU. Zde je implementováno jak rozšíření funkcí systému Magion –

personalistiky i ekonomiky o aplikace přístupné i pro systém Magion neoprávněným zaměstnancům (typickým příkladem jsou například informace uchovávané v personalistice či elektronický výplatní lístek), tak podpůrné aplikace pro obsluhu počítačových studoven provozovaných univerzitou (například online sledování obsazenosti studovny), aplikace pro různé provozní služby (správa půjčování klíčů od dveří na univerzitě, správa softwaru instalovaného na počítačích v majetku MU, rezervace a půjčování majetku MU či aplikace pro obsluhu univerzitní telefonní ústředny), informační systém pro evidenci projektů (jeho popis by mohl vydat na další samostatnou diplomovou práci) a konečně, jak bude popsáno v dalším textu, i nově implementovaný systém „elektronické peněženky a podpory účtování“.

Vymezením okolí systému tedy dostáváme poměrně jasné obrysy toho, jak bude nový informační systém vypadat. Možnost zakoupení (upraveného) komerčního řešení byla zvážena, nicméně brzy zavrhnuta. Důvodů bylo několik: od jistou prestiž přinášejícího vývoje vlastního řešení přes zkušenosti s vývojem podobného systému v předchozích letech po získané poznatky z nasazení IS Magion – splňujícího požadavky leč za cenu vysokých finančních i časových nákladů a potřebě zaměstnávání podpůrných techniků přímo na univerzitě kvůli příliš dlouhé reakční době při řešení problémů.

## 2 Teoretická východiska práce

### 2.1 Informační systém

V předchozích kapitolách bylo nadužíváno slovní spojení „informační systém“ bez toho, aby bylo upřesněno oč se jedná. Tento nedostatek nyní napravíme.

Informační systém je nejčastěji definován jako systém pro sběr, udržování, zpracování a poskytování informací a dat – tuto definici nalezneme jak v odborné literatuře [1], tak i na různých místech internetu. Jiné zdroje udávají, že se jedná o jakoukoliv kombinaci informačních technologií a lidských aktivit, které podporují rozhodování, plánování a řízení. V širším smyslu slova se tak toto slovní spojení používá pro označení všech interakcí mezi lidmi, algoritmickými procesy, daty a technologiemi [12]. Je tudíž nutné si uvědomit, že hovoříme-li o informačních systémech, nesoustředíme se pouze na informační technologie.

Proto v této práci budeme při návrhu systému uvažovat i jeho lidskou část, konkrétně interakci obsluhy s informačním systémem. Intuitivně funkci informačního systému každý chápe – v následujícím textu se podíváme na některé z aspektů, které budou využity při návrhu informačního systému pro zpracování plateb na Masarykově univerzitě.

V dalším textu se často budeme setkávat s tzv. Enterprise Resource Planning (zkracováno ERP) systémy. Ačkoliv zvažovaný informační systém nespadá přesně do této kategorie, je vhodné si tento pojem definovat. ERP systémy představují softwarové nástroje používané k řízení podnikových dat. ERP systémy pomáhají podnikům v oblasti dodavatelského řetězce, příjmu materiálu, skladového hospodářství, přijímání objednávek od zákazníků, plánování výroby, expedice zboží, účetnictví, řízení lidských zdrojů a v dalších podnikových funkcích [1]. Další zdroj definuje ERP systémy jako velmi účinný nástroj, pokrývající plánování a řízení hlavních interních podnikových procesů (tedy takových, jichž je podnik vlastníkem a má nad nimi plnou kontrolu) na strategické až operativní úrovni [8].

Obecně je možné říci, že náš informační systém bude tvořit část ERP systému Masarykovy univerzity.

## 2.2 Vývoj podnikových IS

Basl [1] uvádí, že uplynulých deset až patnáct let se v podnicích neslo ve znamení zavádění komplexních informačních systémů kategorie ERP. Jedná se o aplikace, které nejvýrazněji ovlivňují současné podniky, nejen díky počtu implementací, ale především kvůli své důležitosti. ERP systémy používá podle údajů z roku 2008 více než 90 % podniků zařazených v TOP 100 České republiky; ERP systémy tudíž ovlivňují rozhodování v podnicích s významným podílem na exportu, zaměstnanosti i na tvorbě HDP. Ekonomický i společenský význam je tedy značný [1].

Pro poznání informačního systému v podniku je třeba pochopit reálné postavení informačních a komunikačních technologií, které tvoří jeden z nejdůležitějších formálních rámců podnikových IS. Na rozdíl od ostatních technologií v podniku totiž neslouží jen přesně vymezené skupině zaměstnanců, naopak se týkají celého podniku a všech jeho oblastí [1]. Náзор, že IS je vybírán, kupován a spravován IT oddělením je v současné době už překonaný a ojedinělý.

Informační systém v podniku může být chápán v širším rámci v závislosti na nosičích informace:

- informace zapsané a zpracovávané nejčastěji prostřednictvím relační databáze, sloužící k automatizaci činností či k podpoře rozhodování
- informace uložené na dalších nosičích – formulářích, dokladech, zprávách a předpisech, či nestructurovaně v nějaké aplikaci ICT
- informace, které nejsou nikde zaznamenány – existují například pouze v hlavách zkušených zaměstnanců a jsou využívány operativně podle potřeby

Od těchto tří hlavních druhů nosičů lze odvodit tři roviny chápání informačního systému:

- informační systém podporovaný ICT
- informační systém formalizovaný
- obecně komplexní sociotechnický informační systém podniku

Přitom každá další rovina v sobě obsahuje celou rovinu předchozí. Všechny tyto roviny jsou důležité a není možné je podceňovat.

Podívejme se nyní na varianty pořízení nového informačního systému do podniku. V osmdesátých a devadesátých letech minulého století převažoval vývoj systémů vlastními silami, v současné době se začíná více rozvíjet dodání systému externí firmou. Výhody a nevýhody jednotlivých řešení přehledně shrnuje následující tabulka.

| Varianta řešení                     | Výhody                                                                                                                                                                                          | Nevýhody                                                                                                                                                                                         |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Rozvoj existujícího řešení          | <ul style="list-style-type: none"> <li>- maximální využití existujících zdrojů a investic</li> <li>- z krátkodobého hlediska laciné a rychlé</li> <li>- uspokojení okamžitých potřeb</li> </ul> | <ul style="list-style-type: none"> <li>- nemusí odpovídat budoucím požadavkům</li> <li>- celkové náklady mohou být vyšší</li> <li>- výsledným produktem může být méně kvalitní systém</li> </ul> |
| Vývoj nového systému na míru        | <ul style="list-style-type: none"> <li>- může přesně odpovídat potřebám podniku</li> <li>- řízený vývoj</li> </ul>                                                                              | <ul style="list-style-type: none"> <li>- celkově dražší řešení</li> <li>- časově náročné řešení</li> <li>- riziko negarantovaného konečného produktu a jeho dalšího vývoje</li> </ul>            |
| Nákup hotového softwarového systému | <ul style="list-style-type: none"> <li>- z dlouhodobého hlediska finančně méně náročný</li> <li>- rychlejší zavedení</li> <li>- zaručená funkčnost a další vývoj</li> </ul>                     | <ul style="list-style-type: none"> <li>- nemusí přesně splňovat všechny požadavky</li> <li>- závislost na dodavateli</li> </ul>                                                                  |

**Tabulka 1: Varianty řešení IS. Převzato z [1]**

Kromě přechodu od vývoje vlastních řešení k nákupu hotových se také měnil procesní přístup k jednotlivým ERP systémům. Jejich vývojové generace – přesněji řečeno

jednotlivá paradigmat, která byla uplatňována při vývoji ERP systémů – shrnuje následující tabulka.

|                       | 1. generace<br>1975                         | 2.<br>generace<br>1985                                       | 3.<br>generace<br>1992                                     | 4. generace<br>1996                                            | 5. generace<br>2000                                    |
|-----------------------|---------------------------------------------|--------------------------------------------------------------|------------------------------------------------------------|----------------------------------------------------------------|--------------------------------------------------------|
| Způsob zpracování     | Dávkové zpracování                          | Zpracování v dialogu                                         | Zpracování v dialogu i v dávce                             | Možnost volby zpracování                                       | Zpracování prostřednictvím internetu                   |
| Přenositelnost        | Spojení s určitým počítačem                 | Spojení s operačním systémem                                 | Přenositelnost mezi operačními systémy                     | Třívrstvé aplikace (DB, vlastní aplikace, prezentace)          | Integrace aplikací SOA – Service Oriented Architecture |
| Programové prostředky | Nižší programovací jazyky                   | Vyšší programovací jazyky                                    | Relační databáze a programovací nástroje SQL               | JAVA a objektové databáze                                      | Prostředky XML                                         |
| Uživatelské podmínky  | Neinteraktivní                              | Standardní obrazovky – textový režim                         | Konfigurovatelné uživatelské obrazovky – prostředí Windows | Multimediální aplikace, internetové prostředí a webové stránky | Přístup přes mobilní zařízení, tendence ke službám     |
| Funkčnost             | Plánování především materiálových požadavků | Materiálové a kapacitní plánování a řízení výrobních zakázek | Integrovaný informační systém řízení podniku               | Dodavatelsko odběratelské řetězce                              | e-business, CRM, BI                                    |

Tabulka 2: Základní vývojové generace ERP systémů. Převzato z [1]

Pro současné období je typický vývojový systém, který se projevuje orientací na služby, které navazují na předchozí etapu zaměřenou na klient-server architekturu. Pro jejich integraci jsou vyvíjeny integrační platformy, například v případě velmi rozšířeného SAP se jedná o produkt SAP Netweaver.

## **2.3 Datový pohled na informační systémy**

Na informační systémy také existují dva pohledy – datový a procesní. Datové sjednocení různých aplikací IS prostřednictvím společné databáze představuje jeden z principálních fenoménů úspěšnosti a rozvoje podnikových IS nasazovaných od devadesátých let minulého století. Důležitost dat pro aplikace ERP dokládá i autor knihy Necessary but not Sufficient E. Goldratt, který sílu ERP spatřuje ve schopnosti sdílet, udržovat, skladovat a znovu vyvolávat data.

Relační databáze napomohly sjednocením podnikových dat a jejich snadnější dostupností ke snížení nákladů na materiálové zásoby, zkrácení časů realizace zakázek i k přesnější a rychlejší podpoře rozhodování. Důsledky se nejprve projevily v oblasti operativních transakčních aplikací ERP, později pak na ně navázaly pokročilejší aplikace – za všechny jmenujme třeba data mining.

Využití datového pohledu při dekompozici podnikových IS patří mezi nejčastější přístupy, je na něm například založen populární modelovací jazyk UML [14]. Datový pohled je rovněž důležitou součástí architektury IS podniku. Jedná se o svým způsobem technologický přístup k chápání IS. Znázornit jej lze formou na sebe navazujících vrstev, kde základ tvoří hardware, nad ním leží operační systém, databázový systém a aplikační software.

V dřívějších dobách byl využíván velice často, mimo jiné proto, že vlastní analýzy potřeb uživatelů byly prováděny ve větší míře a na jejich základě byla navrhována ucelená řešení [1]. K podpoře takových řešení sloužila celá řada CASE (Computer Aided Software Engineering) nástrojů a metodik – například SDM (Software Development Methodology) nebo na Masarykově univerzitě vyvinutá metoda HIT [9]. S příklonem podniků k dodavatelsky realizovaným parametrizovatelným řešením se tento model chápání podnikového IS přesunul k tvůrcům aplikací, tedy mimo vlastní podnik.

Vrstvy datového IS lze využít i potřebě přenositelnosti jednotlivých řešení – mohou na sobě být do jisté míry nezávislé. Jedná se tak o obdobu třívrstvé architektury (viz kapitola 2.9), kdy datová vrstva je oddělena od vrstev aplikační a prezentační.

Basl [1] dále uvádí, že v podnikových IS se používá několik kategorií dat, které jsou potřebné pro správné fungování systému. Jedná se o následujících pět skupin:

- číselníky, používané pro identifikaci položek, pracovišť, nákladových středisek, zákazníků, dodavatelů apod.
- kmenová data, obsahující informace o výrobku, způsobu jeho realizace, výrobní základně, dodavatelských materiálu včetně adres a zákaznických včetně adres
- zakázková data s údaji o zakázce pro konkrétního zákazníka s vazbou na požadované termíny, množství, strukturu a provedení výrobku
- archivní data obsahující údaje k již realizovaným a uzavřeným zakázkám
- parametry upřesňující fungování ERP systému a jeho jednotlivých modulů v konkrétním podniku

Z hlediska provozu IS/IT je pak důležité členění databází na provozní, školicí a vývojové. V první z nich jsou ukládána skutečná podniková data, další dvě slouží, jak jejich název napovídá, k vyvíjení a testování software a školení uživatelů. Neprovozní databáze může obecně být jedna či naopak jich může být i více než dvě.

## **2.4 Procesní pohled na informační systémy**

Jedním z důvodů zavádění podnikových IS je i zlepšení podnikových procesů. Procesní orientace podniků je významný fenomén, který souvisí se všemi podnikovými aktivitami a projekty.

Snaha podniků o přeměnu vykonávaných činností v souladu s principy procesního řízení ovlivňuje tvorbu a využití příslušných softwarových aplikací i představu o modelu informačního systému v podniku.

Pro lepší pochopení procesní pohledu na podnikové IS si nejprve definujeme, co proces je. Podle definice ČSN EN ISO 9001:2001 je proces soubor vzájemně souvisejících nebo vzájemně působících činností, které přeměňují vstupy na výstupy. Proces je započat spouštěcí událostí (situace nebo okamžik v čase), která může být specifikována pro každou činnost zvlášť. Nejdůležitější je výsledný stav procesu, přinášející hodnotu

pro zákazníka. U procesu lze identifikovat hranice a přiřadit mu měřitelné parametry sledující jeho účinnost a účelnost. Důležité také je, aby proces byl standardizovaný a opakovatelný [1].

S každým procesem souvisí základní objekty, které jej ovlivňují. Jedná se o:

- cíle, kterých má pomocí procesu být dosaženo
- vstupy, které jsou procesem spotřebovávány nebo přetvářeny
- výstupy, které jsou výsledkem nebo produktem procesu
- podpůrné objekty, které jsou procesem užívány, ale nejsou spotřebovávány ani se nemění
- řídicí objekty, které řídí běh procesu

Procesy se podle významu pro podnik dělí na klíčové, které jsou určené k naplnění poslání firmy a určené pro vnějšího zákazníka, podpůrné, které jsou určené pro vnitřního zákazníka podniku a nelze je bez ohrožení firmy z podniku vyčlenit, a vedlejší, které jsou také určené pro vnitřního zákazníka a je možné je bez problému outsourcovat [1].

Každý proces samozřejmě má svoji kvalitu, nebo-li stupeň toho, jak dobře je v podniku zvládnutý a definovaný. V praxi je známá metodika CMM (Capability Maturity Model), která procesy rozděluje do následujících šesti kategorií podle stupně jejich zralosti.

- neexistující – pozorovatelný proces neexistuje, při výskytu události reaguje organizace spontánně, podnik neví, že existuje problém, který je třeba řešit
- náhodný – podnik si je vědom problému, ale události se stále řeší individuálně, „ad hoc“ přístupem
- opakovaný – v podniku existuje snaha o vytvoření standardních procesů, jejich využití je však stále pouze intuitivní
- formalizovaný – existuje standardizace a popis jednotlivých procedur, vlastní realizace však stále zůstává v rukou jednotlivců
- měřitelný – k formalizovanému procesu je přidán proces měření průběhu a jeho kontroly
- optimalizovaný – proces na základě měření a kontrol se zlepšil až do nejlepšího možného stavu, jsou definovány a používány tzv. „best practices“

V okamžiku, kdy víme, co jsou procesy, jak se dělí a v jaké kategorii zralosti se mohou nacházet, je načase se zamyslet nad tím, jak mohou být podpořeny informačními systémy. Podle toho, jak je proces automatizovatelný, může být podpořen IS – případně proveden celý automaticky s minimálními zásahy operátora. Například zpracování příchozí platby bankou může proběhnout celé automaticky po příchodu výpisu z banky, zatímco proces vytvoření reklamního letáku je informačním systémem spíše jen podpořen.

Basl [1] dále uvádí, že vazba mezi podnikovými procesy a informačními systémy je velmi těsná. S tímto tvrzením se nedá nesouhlasit – z určitého pohledu by IS měl být navrhován tak, aby sloužil především k podpoře provádění podnikových procesů. Procesní přístup je tak možné použít ve všech hlavních fázích životního cyklu IS v podniku:

- před implementací, kdy se podnikový IS navrhne ve vazbě na zanalyzované podnikové procesy, které je možné v tomto okamžiku zoptimalizovat
- v průběhu implementace, kdy popsané „best practices“ mohou implementaci zrychlit a zlevnit
- v průběhu provozu IS, kdy IS je provozován na základě optimalizovaných procesů a slouží k jejich podpoře

Definujme si ještě způsob měření procesů, který je nezbytně nutný k tomu, aby podle CMM mohl být proces na jednom ze dvou nejlepších stupňů co se zralosti týče. Existuje soustava nástrojů na podporu výkonnosti podniku a jeho měření (tzv. BPP – Business Process Performance) [1]. Sledovanými ukazateli mohou být například.

- doba průběhu jednotlivých činností či celého procesu
- včasnost zahájení či ukončení činností
- počet vstupů a výstupů za jednotkové období
- množství potřebných zdrojů
- doba čekání požadavků na zpracování procesem
- zpoždění výsledků pro zákazníky

K měření procesů však nestačí pouze existence vhodného IS naplněného správnými daty. Je nutné definování vhodných metrik výkonnosti jednotlivých procesů, stanovení jejich výchozích a limitních hodnot a vah těchto metrik [1]. Hodnoty, získané měřením

procesů, je možné zahrnout do mzdového ohodnocení jednotlivých zodpovědných pracovníků – to však záleží na organizaci a charakteru probíhajících procesů.

## **2.5 Efektivnost informačních systémů**

Zajímavou oblastí, nad kterou se musíme pozastavit, je efektivnost IS. Ačkoliv je jasné, že univerzita nový informační systém potřebuje, je třeba – zejména ve vypjaté situaci krácení veřejných rozpočtů – zvolit či navrhnout takový systém, který bude dostatečně efektivní a umožní univerzitě v nejlepším případě ušetřit na jeho provozu nebo na jeho obsluze.

Molnár [7] uvádí, že na podnik je možné se dívat ze dvou hledisek. Častější je pohled na podnik jako na racionální systém, který klade důraz na organizaci jako celek a na účel, pro který byla vytvořena. Pozornost je zaměřena především na koordinaci činností jednotlivých skupin jednotlivců a rozhodování o činnostech je důsledně odvozováno od cílů organizace a je podpořeno vymezením authority. V těchto rozhodovacích procesech hraje mimořádnou úlohu informace ve smyslu zajištění spojitosti – podporuje koherenci systému.

Druhé pojetí, tedy pojetí podniku jako přirozeného systému, chápe organizaci jako soubor individuí, která jsou ve vzájemné interakci a prostřednictvím společné činnosti naplňují své individuální zájmy a cíle, mezi které patří například veřejný úspěch a uznání, materiální výhody, možnost tvořit apod. Pozornost je pak kladena na sladění individuálních cílů jak navzájem, tak se záměry organizace. Vymezení rozhodovací pravomoci je volnější, stejně jako možnost interpretace informace, která plní novou významnou roli – přispívá ke kohezi systému, jeho vnitřní soudržnosti [7].

Je třeba si uvědomit, že zatímco v běžné komerční firmě je spíše zdůrazněn první pohled na podnik – to souvisí především s cílem dosahování maximálního zisku, postavení na trhu či vysoké produktivity, v akademickém prostředí je zdůrazněno zejména druhé pojetí.

Molnár [7] dále uvádí dva možné pohledy na IS/IT. Tyto systémy nejenže přispívají ke zvyšování efektivnosti a úspěšnosti podniku, ale současně vytvářejí podnikové jmění – především nehmotné. Data, která se v podniku sbírají, zpracovávají a uchovávají po určitou dobu, v sobě často skrývají netušené informační bohatství. Toto bohatství roste

s časem, po který se data zpracovávají a je reprezentováno znalostmi, které je možno využít dále k rozvoji podniku. Zavedením IS/IT se nejen racionalizují podnikové procesy, ale zvyšuje se know-how pracovníků a výrazně se zlepšuje podniková kultura. Dva rozdílné pohledy na informační systém pak jsou pohled výdajový a pohled majetkový – jejich srovnání je v následující tabulce.

| Pohled výdajový                | Pohled majetkový                  |
|--------------------------------|-----------------------------------|
| Taktický – krátkodobý – pohled | Strategický – dlouhodobý – pohled |
| Musíme to udělat               | Můžeme si dovolit to neudělat?    |
| Uděláme to a půjdeme od toho   | Nikdy to neskončí                 |
| Analýza nákladů                | Hodnocení investic                |
| Kde na to seženeme peníze?     | Musíme si na to plánovat peníze   |
| Řízení nákladů                 | Hledání užitku                    |
| Účtování výdajů                | Účtování majetku                  |

**Tabulka 3: Srovnání výdajového a majetkového pohledu na IS/IT. Převzato z [7]**

Z tabulky 3 je vidět, jaká stanoviska musíme při volbě informační strategie společnosti pečlivě uvážit. V případě univerzity jsou některé ukazatele důležitější, některé méně podstatné.

Další zajímavou otázkou je, co se vlastně od informačního systému očekává? Molnár [7] na to odpovídá jednoznačně – podstatou celkové efektivity IS/IT je, aby na konci každého takového projektu byl spokojený uživatel, a to na všech stupních řízení a oblastí užití IS/IT. Z tohoto konstatování vyplývá, že bychom měli věnovat velkou pozornost uživateli a jeho očekáváním. Ta mohou být samozřejmě velmi rozdílná podle toho, jaké postavení a jakou roli uživatel v podniku má a také podle toho, jak je informačně gramotný. V našem případě je jednoznačně rozdílné očekávání účetních univerzity - kromě společného základu správného zacházení s penězi požadují především vhodné sestavy a správné zacházení s DPH – od klientů – studentů – kteří by preferovali především jednoduchost použití systému a snadnou kontrolu výdajů; DPH se většiny z nich příliš netýká.

## 2.6 Přínosy informačních systémů a technologií

Systematizace ukazatelů přínosů je nutná proto, abychom hned na začátku životního cyklu IS/IT dokázali tyto ukazatele definovat pro konkrétní aplikaci a konkrétní podnik, uměli stanovit způsob jejich vyhodnocování i konkrétní odpovědnost za dosažení určité hodnoty tohoto ukazatele [7].

Ukazatele přínosů IS/IT můžeme klasifikovat z několika hledisek a to na:

- finanční (měřené v peněžních jednotkách) a nefinanční (měřené jinými fyzikálními jednotkami jako jsou počet, čas apod.)
- kvantitativní (měřené kardinální stupnicí) a kvalitativní (měřené ordinální stupnicí či logickou hodnotou)
- přímé (u kterých můžeme prokázat jednoznačný příčinný vztah k dosaženému přínosu) a nepřímé (u kterých musíme stanovit nějaké zástupné ukazatele vyjadřující změnu)
- krátkodobé (projevující se do půl roku po implementaci IS/IT) a dlouhodobé (projevující se později)
- absolutní (vyjádřené měřitelnou hodnotou) a relativní (vyjádřené bezrozměrným číslem)

Z uvedeného je vidět, že možnosti používání ukazatelů i jejich kombinací jsou skutečně bohaté a nelze tudíž definovat jeden univerzální systém použitelný pro všechny podniky a situace. Vždy se musíme řídit tím, co jsme schopni hodnotit, a zejména pak hlediskem účelu a zodpovědnosti [7].

Obecně je možné ke všem ukazatelům (nezávisle na jejich charakteru) říci, že vždy musíme sledovat hledisko účelnosti, které je obecně měřitelné mírou dosažení cílů, tedy ukazatelem  $účelnost = \frac{\text{dosažená hodnota cíle}}{\text{plánovaná hodnota cíle}}$ . To znamená, že musíme vždy při plánování jakékoliv aplikace IS/IT stanovit nebo odhadnout žádoucí nebo požadovanou hodnotu [7].

### 2.6.1 Finanční ukazatele

Finanční ukazatele se obvykle určují v etapě plánování IS/IT, kdy potřebujeme zdůvodnit ekonomickou výhodnost dané investice. Většinou aplikujeme některý ze standardních ukazatelů efektivnosti jako jsou analýza nákladů a přínosů, diskontovaný

cash flow, vnitřní míra výnosnosti, čistá současná hodnota, doba návratnosti investice nebo návratnosti kapitálu. Obecně jsou uváděné zkratkou ROI (Return Of Investment). Často používaný ukazatel je také ukazatel návratnosti kapitálu, označovaný ROA (Return of assets) a doba obratu oběžných prostředků [7].

Z pohledu univerzity však tyto ukazatele nejsou příliš podstatné – univerzita se podle nich řídit nebude, neboť se nejedná o podnik, provozovaný za účelem generování zisku.

### **2.6.2 Nefinanční měřitelné ukazatele**

Mezi nefinanční, avšak měřitelné ukazatele přínosů IS/IT, patří především produktivita, která poskytuje hodnotnou informaci o vztahu mezi vstupními náklady a výstupním užitekem. Jedná se o poměr mezi množstvím výstupů a množstvím vstupů za danou časovou jednotku [7].

Ukazatel produktivity můžeme vyhodnocovat pro nejrůznější zdroje, které používáme pro výrobu či službu, jinými slovy pro tvorbu hodnoty pro zákazníka. Důležité je, abychom stanovili vhodné jednotky jak pro čitatele, tak pro jmenovatele, a byli schopni veškeré hodnoty systematicky sledovat a vyhodnocovat. Cílem by měl být růst produktivity, tedy ideálně zároveň zvyšování výstupů a snižování vstupů.

S obecnou platností uvedeného tvrzení se jistě dá souhlasit – bohužel však v naší oblasti je těžko uplatnitelné. Problém je už v prvním kroku – je totiž těžko určitelné, co je v případě našeho systému výstupem. Dalo by se snad uvažovat o počtu zpracovaných položek na klientském účtu na pracovníka, či počtu aktivních klientů na pracovníka.

Mezi další používané ukazatele patří například zkrácení průběžné doby vývoje a výroby, snížení počtu reklamací, zvýšení podílu na trhu, rozšíření výrobního sortimentu a řada dalších podle konkrétní podnikatelské činnosti. Většina z nich je však po krátkém zamyslení však vyjádřitelná ve finančních jednotkách [7].

### **2.6.3 Nekvantifikovatelné ukazatele**

K tomu, abychom mohli hodnotit, zda dochází k žádoucím či nežádoucím změnám nekvantifikovatelných ukazatelů (často nazývaných měkké ukazatele), je obvykle třeba si najít nějaký kvantifikovatelný ukazatel, jehož změna co nejlépe reflektuje změnu původního ukazatele (tzv. zástupný ukazatel) [7].

Molnár [7] uvádí následující obvyklé měkké ukazatele

- zlepšení dobrého jména podniku
- spokojenost zákazníků
- zvýšení zákaznické věrnosti
- flexibilita podniku, kreativita v přijímání nových produktů, služeb, procesů či struktur
- reakce na nové potřeby trhu
- zlepšení pracovního prostředí
- zvýšení kvalifikace pracovníků podniku
- přidání hodnoty produktu či službě

V našem případě bychom mohli hovořit o následujících měkkých ukazatelích a využít je

- zvýšení prestiže univerzity vytvořením unikátního systému
- lepší servis pro studenty univerzity spolu s lepším přehledem o výdajích
- snadnější rozšiřování nabídky služeb hrazených z jednotného konta
- snížení zátěže kladené na účetní univerzity

Takto bychom mohli vyjmenovat celou řadu nejrozmanitějších požadavků – vždy je však dříve, než daný požadavek vytvoříme, nutno mít jasno, zda v budoucnu bude schopni říci, že došlo k nějaké změně v požadované oblasti. Pro náš případ to platí zhruba z 50 % - zatímco u druhých dvou jsme schopni rozdíl snadno zjistit (například v počtu člověkohodin nutných k přidání nové služby do nabídky), u prvních dvou to tak zcela neplatí.

## **2.7 Lidský aspekt efektivnosti informačních systémů a technologií**

Podívejme se ještě na jednu zajímavou, již na počátku textu zmíněnou, partii efektivnosti IS/IT, kterou je role člověka v jejím vnímání. Člověk vystupuje v IS/IT ve dvou rolích, jako jeho tvůrce a uživatel [7]. V první roli není možné člověka chápat jen jako analytika či programátora, ale především jako vedoucího, zodpovědného za zavádění informačního systému do podniku. Molnár [7] dále uvádí, že jen málo řídicích pracovníků si tuto roli uvědomuje do všech důsledků, tedy že člověk je nedílnou součástí IS/IT a jako takového je nutné tento „lidský zdroj“ řídit, tj. plánovat, organizovat, motivovat, kontrolovat a kultivovat. Výkonný hardware a software se stává

zbytečným v okamžiku, kdy s ním lidé nejsou spokojeni nebo jej neumějí používat. Efektivnost IS/IT tudíž závisí na lidech mnohem více, než na samotných informačních technologiích.

Člověk, jakožto uživatel IS/IT vnímá výhody a nevýhody systémů z několika hledisek:

- ekonomické hledisko
- pohodlí při práci
- uspokojení z práce
- sociální postavení
- jistota práce
- možnost a schopnost ovlivňovat změny v podniku

Každé z těchto hledisek má několik svých podpohledů, které je však nejspíše zbytečné uvádět – zájemce si je může dohledat v literatuře [7].

Další hledisko, které musí vedoucí pracovník mít na mysli, je rozdílnost lidských povah a celková kultura podniku při zavádění aplikace IS/IT na podporu skupinové nebo týmové práce. Molnár [7] toto označuje jako tzv. groupware, jehož charakteristickými znaky jsou sdílení informací skupinou pracovníků s tím, že každý uživatel se rozhoduje sám, kdy a která data ze společné databáze použije nebo do ní uloží. Hlavním cílem je podpora spolupráce a koordinace. Přínosem groupware je především zvýšení výkonu týmu, který je možno měřit zejména zkrácením času nutného pro splnění určitého úkolu (v případě našeho ekonomického softwaru se jedná například o zaúčtování souhrnných dokladů do účetnictví).

Zavedením groupware se však mohou také objevit či zvýraznit nežádoucí efekty, jako je nedostatečná zkušenost či dovednost některých členů týmu s novou technologií, jejich rozdílné individuální postoje a osobní charakteristiky, slabá vzájemná soudržnost či nedůvěra mezi členy týmu až neochota sdělovat volně své myšlenky ostatním. Groupware obvykle bourá dosavadní hierarchické bariéry a organizační normy, což někteří členové týmu mohou vnímat jako ohrožení svého postavení [7].

Z uvedeného vyplývá, že je nutné v podniku pro řízení zdrojů zvolit tzv. IT manažera – v zahraničních podnicích označovaný jako CIO – Chief Information Officer.

D. O. Aleen z Lockheed Corporation uvádí, že IT manažer není ředitel informací, protože informace nevlastní ani je nevytváří. V nejlepším případě řídí procesy

zpracování a distribuce informací [7]. Za tímto účelem musí plnit řadu rolí, které vyžadují komplexní vzdělání a zkušenosti. Jedná se především o následující úlohy:

- vytvářet vizi IS/IT a prosazovat ji
- poznávat podnik a zejména jeho trhy a zákazníky
- získávat důvěru v útvar pro IS/IT
- pečovat o informační gramotnost pracovníků
- dbát na systémovou vyspělost IS/IT v podniku
- rozvíjet informační strukturu

Vedle IT manažera je další důležitou osobou vedoucí projektu IS/IT, protože informační strategie se uskutečňuje právě formou realizace jednotlivých projektů. Vedoucí projektu musí mít důvěru jednotlivých majitelů a managementu podniku a musí být ochoten převzít odpovědnost za výsledek celého projektu. Jako vedoucí realizačního týmu musí být vybavený potřebnými pravomocemi a správně motivován [7].

Ke své ruce musí mít vedoucí projektu kvalitní tým pracovníků, schopných zvládnout svěřené oblasti IS/IT, kteří disponují nejen odbornými znalostmi, ale především by to měli být lidé nadšení pro danou věc s konstruktivním přístupem k řešení problémů [7].

Při splnění uvedených podmínek pak je možné dosáhnout úspěchu při projektu zavádění IS/IT, v opačném případě je projekt odsouzen k neúspěchu již před svým započatím.

## **2.8 Projekty zavádění informačních systémů do podniků**

Další oblastí, která by neměla být opomíjena, jsou projekty zavádění informačních systémů do podniků. Je totiž nutné si uvědomit, že veškeré změny v oblasti IS vždy probíhají formou projektu bez ohledu na to, zda se jedná o vytvoření nového IS, jeho implementaci či úpravu stávajícího systému [1].

Na rozdíl od tradičních projektů – např. ve stavebnictví, kdy výsledek má zcela konkrétní hmotnou podobu – mají projekty z oblasti IS i velmi podstatnou nehmotnou stránku. Kromě naplnění daty se zejména často mění podniková kultura v celé organizaci a její chod tak může být závažným způsobem narušen – mimo jiné i kvůli změně v podnikových procesech.

Projekty IS počítají s tím, že podnik má sestavenou informační strategii, ve které má vytyčené směry rozvoje IS ve prospěch zlepšení postavení a hodnoty podniku, zvýšení

hodnoty pro zákazníka a zefektivnění procesů v podniku. Pro tyto oblasti je nutné realizovat následující kroky:

- provedení analýzy současného stavu
- zpracování návrhu řešení
- sestavení projektového plánu realizace
- realizace změny a uvedení do provozu
- údržba a další rozvoj, včetně případné aktualizace informační strategie

Platí tedy, že projekty IS/IT se jako všechny ostatní potýkají s nejrůznějšími představitelnými problémy a balancují mezi třemi základními hledisky – náklady, termíny a kvalitou, kde platí obecná poučka, že lze mít pro podnik pouze dvě ze třech hledisek výhodná (například lze, aby projekt byl rychlý a kvalitní, ale v takovém případě nebude levný).

Z průzkumů vychází, že úspěšnost zavádění projektů je stále nízká – vysoké procento systémů není nikdy úspěšně použito (některé zdroje uvádějí až 85% neúspěšnost projektů IS [10]).

Projekty zavádění IS probíhají v několika základních krocích. Obecně platí, že s rozumnými náklady se lze vrátit o jeden krok zpět, návrat o více než jeden krok je velmi nákladný – jak časově, tak finančně – a odpovídá téměř novému započetí celého projektu [1]. Zmíněné kroky jsou následující:

- rozhodnutí o změně podnikového IS, které musí vycházet z jasného záměru v souladu s business strategií podniku. V této fázi je možné použít SWOT analýzu, která dá bližší informace o současném a možném budoucím stavu. Je nutné také uvážit přání všech zájmových skupin.
- vytvoření řešitelského týmu s jasně daným vedoucím, který je zodpovědný za úspěch projektu zavádění IS do podniku. Každý krok týmu by měl být dokumentovaný.
- výběr vhodného systému a jeho dodavatele, případně rozhodnutí o vytvoření systému vlastními silami. Je nutné provést co možná nejobjektivnější srovnání dostupných IS a jejich otestování nad zkušebními daty (reálnými podnikovými či u dodavatele). V menších podnicích mohou s výběrem pomoci i externí specialisté. Je nutné také hodnotit nejen produkt, ale i jeho

dodavatele – vynikající IS je zbytečný v okamžiku, kdy je dodán nestabilní firmou, která po jednom roce zanikne a ukončí tak jeho podporu. Rovněž je nutné hodnotit i další kritéria, jako je rychlost a cena implementace.

- uzavření smlouvy na zavedení IS – v případě zvolení externího dodavatele, většinou se jedná o rámcovou smlouvu o dílo, která upravuje obecné principy a otázky systémové integrace či projektu.
- vlastní implementace, která se může rozpadnout do několika podetap. Každá firma k ní může přistupovat jinak; příkladem může být implementace SAP, která se dělí do fází analýzy požadavků a návrhu koncepce, detailního návrhu a realizace, přípravy produktivního systému a zahájení provozu. Každá z nich se pak dělí na další dílčí úkoly a činnosti – zájemce je může nalézt v [1]. K implementaci a zavedení systému do podniku je možné přistoupit několika způsoby – buď se jedná o jednorázový přechod, kdy k určenému datu přejde celý podnik nárazově na nový IS, paralelní přechod, kdy jsou chvíli používány jak starý, tak nový IS, postupný přechod, kdy na nový IS přecházejí jednotlivé části podniku nebo paralelní přechod po částech.
- provoz a údržba vybraného systému, kdy je důležité metodické řízení provozu i vlastní podnikové informatiky. V současné době existují de facto dva standardy řízení návrhu, implementace a zejména provozu IS/IT, které jsou promítnuty i do norem ČSN. Jedná se o metodiky ITIL (IT Infrastructure Library) a COBIT (Control Objectives for Information and related Technology). Obě metodiky mají společné to, že k řízení IS/IT přistupují především z pohledu procesů. Existují dokonce převodní tabulky, které mapují jednotlivé procesy metodiky ITIL na procesy metodiky COBIT [5], nejedná se však o bijektivní zobrazení, některé procesy metodiky ITIL tudíž nemají svůj obraz v metodice COBIT a naopak. Rozdíl je také v jazyku metodik – metodika COBIT vznikla v několika profesionálních firmách, je napsána složitěji, zatímco metodika ITIL je spíše shrnutí „best practices“ používaných při návrhu informačních systémů.

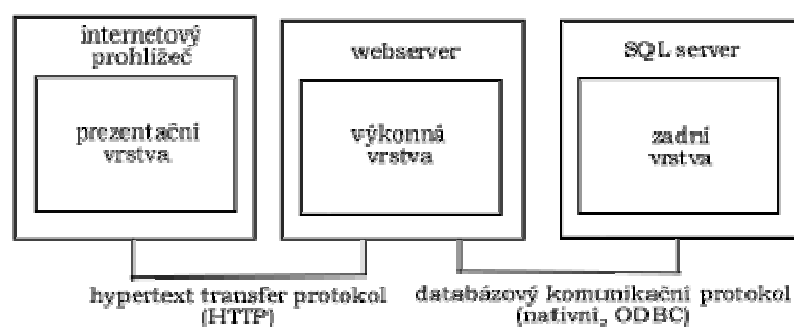
## 2.9 Datová, aplikační a prezentační vrstva

Jedním z dalších trendů v poslední době je striktní oddělování datové, aplikační a prezentační vrstvy. Protože se jedná o rozumnou součást návrhu složitějších systémů, a protože tento princip bude použit i u návrhu nově vznikajícího informačního systému, uveďme si o co se jedná.

Datová vrstva se stará o uchovávání dat, jejich zpřístupňování a zaručuje jejich konzistenci. Nejčastěji je reprezentována relační databází, může se však jednat i o souborový systém nebo webovou službu.

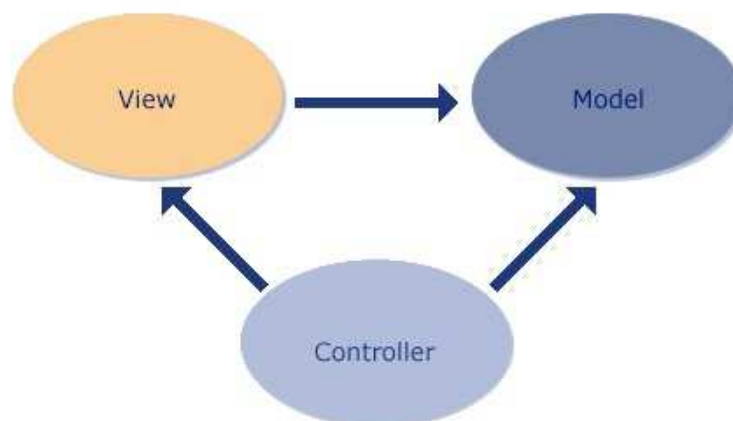
Aplikační vrstva má za úkol zprostředkovat výměnu dat mezi datovou a prezentační vrstvou a naopak a případně tato data vhodným způsobem transformovat. Tato vrstva kontroluje chování systému prováděním detailního zpracování dat. Často je napsána v nějakém objektovém programovacím jazyce jako je například Java, C# nebo .Net.

A konečně prezentační vrstva zobrazuje aplikační vrstvou připravená data klientovi a naopak přijímá vstupy od klienta. Často je tvořena stránkami ve formátu (X)HTML, které jsou snadno přístupné na téměř libovolné platformě a poskytují značnou funkcionalitu a flexibilitu.



Obrázek 1: Třívrstvá architektura. Převzato z interval.cz

Podobnou architekturou je typ MVC neboli model-view-controller, česky model-pohled-řadič. Ačkoliv se tato také skládá ze tří prvků, jsou uspořádány jinak. Rozdíly jsou vidět na obrázcích 1 a 2. Jedná se o softwarovou architekturu, která rozděluje datový model, řídicí logiku a uživatelské rozhraní do vrstev tak, aby modifikace jedné měla žádný nebo minimální vliv na ostatní.



**Obrázek 2: Model-view-controller. Převezato z ucancode.net**

Komponenty pohled a řadič jsou v předchozím dělení součástí prezentační vrstvy, model pak odpovídá aplikační vrstvě.

Ačkoliv může být koncept MVC realizován různým způsobem, obecně platí tento princip:

- Uživatel provede nějakou akci v uživatelském rozhraní (např. stiskne tlačítko).
- Řadič obdrží oznámení o této akci z objektu uživatelského rozhraní.
- Řadič přistoupí k modelu a v případě potřeby ho zaktualizuje na základě provedené uživatelské akce (např. zaktualizuje nákupní košík uživatele).
- Model je pouze jiný název pro aplikační vrstvu. Aplikační logika zpracuje změněná data (např. přepočítá celkovou cenu, daně a expediční poplatky pro položky v košíku). Některé aplikace užívají mechanismus pro persistentní uložení dat (např. databázi). To je však otázka vztahu mezi doménovou a datovou vrstvou, která není architekturou MVC pokryta.
- Komponenta pohled použije zaktualizovaný model pro zobrazení zaktualizovaných dat uživateli (např. vypíše obsah košíku). Komponenta pohled získává data přímo z modelu, zatímco model nepotřebuje žádné informace o komponentě pohled (je na ní nezávislý). Je důležité podotknout,

že řadič nepředává doménové objekty (model) komponentě pohledu, nicméně jí může poslat příkaz, aby svůj obsah podle modelu zaktualizovala.

- Uživatelské rozhraní čeká na další akci uživatele, která celý cyklus zahájí znovu [11].

Tento princip bude použit v návrhu našeho systému – respektive v návrhu jeho uživatelského rozhraní. Implementováním architektury MVC a skriptování na straně klienta (viz kapitola 2.12) je možné dosáhnout významného snížení nároků na systémové prostředky serveru. Vzhledem k předpokládanému zatížení (tisíce uživatelů) se optimalizace stává esenciální součástí návrhu systému.

## 2.10 Klient – server architektura

Jako předposlední část teoretického pozadí práce si uvedeme něco o tzv. klient – server architektuře informačního systému. Princip je skutečně jednoduchý – jedná se o rozdělení operací mezi nejméně dva stroje, z nichž jeden plní roli serveru (je hlavním počítačem) a ostatní jsou klienty (jejich činnost závisí na serveru). Na serveru většinou bývají uložena data, se kterými podnik pracuje a je zde spuštěna aplikace pro jejich obsluhu. Klientské stanice pak mají vlastní aplikace s množinou funkcí nutnou pro činnost pracovníků a mohou u sebe mít uloženu lokální kopii centrální databáze nebo její části. Mohou však také stále přistupovat přímo k datům v centrální databázi – odpadají tak starosti s udržováním konzistence mezi lokálními kopiemi a hlavním úložištěm. Volba způsobu však závisí především na technickém řešení, rozsáhlosti systému a technologických možnostech – nedá se tudíž jednoznačně říci, který způsob je obecně lepší či výhodnější.

S předchozím také souvisí skriptování na straně serveru a na straně klienta, anglicky nazývané server-side scripting a client-side scripting. Skriptování na straně serveru znamená, že klient odešle na server data s požadavkem o provedení určité akce nad těmito daty. Po provedení požadavku server vrátí odpověď – změněná data nebo výsledek operace.

Skriptování na straně klienta naopak znamená, že klient sám má všechny prostředky pro provedení požadované akce a není tak třeba komunikovat se serverem. Nutno

podotknout, že klientské pracovní stanice bývají méně výkonné než servery a proto bývá tento způsob skriptování vyhrazen spíše pro méně náročné operace.

## 2.11 Data mining

Jednou z možných technik, které mohou být použity v některých oblastech zvažovaného informačního systému, je tzv. data mining. Dostál, Rais a Sojka [2] uvádějí, že data mining, neboli dolování z dat, je oborem, který zastřešuje širokou škálu technik, používaných v řadě odvětví [2]. Tento obor se začal rozvíjet počátkem 90. let dvacátého století za účelem vylepšení řízení rizika u bank specializovaných na kreditní karty.

Data mining pomáhá cílům firmy získáváním a vyhodnocováním dat, na jejichž základě je možné získat nového zákazníka, stávajícího zákazníka udržet, odhalit rizikového zákazníka a nalézt způsob aby se správný výrobek setkal se správným zákazníkem ve správném čase. K tomuto cíli je možné dospět za pomoci metod poskytujících optimální výsledky. Protože data nejsou vždy kompletně známa, je vhodné použít některou z rozhodovacích metod založených na fuzzy logice, umělých neuronových sítích nebo na genetických algoritmech. [2]

Příkladem použití data miningu je například následující situace: zákazník, který si zažádá o převedení telefonního čísla od jednoho operátora k jinému se v případě, že byl u původního dostatečně dlouho a měl dobrou platební morálku, téměř jistě dočká telefonátu od původního operátora, kde mu obchodní zástupce nabídne výhodnější podmínky, než měl doposud. Cílem je udržení platících zákazníků – zejména v současné situaci, kdy je trh mobilních operátorů již téměř nasycen.

Pomocí data miningu se vytváří tzv. model chování zákazníka, což lze chápat jako odladěný program, využívající jednu ze zmíněných metod. Je pro něj však potřebné získat správná a kvalitní zdrojová data. Platí zásada, že dodáme-li chybná data, bude analýza vždy chybná; dodáme-li správná data, je výsledek analýzy závislý na kvalitě zpracování dat. V případě data miningu je podstatné, zda volíme model pro udržení ziskových zákazníků (jako v uvedeném příkladu s telefonními operátory), model pro získání nových zákazníků, vyhledání rizikových zákazníků nebo vyhledání ztráty

zákazníka. Je třeba vybrat taková data, která mají pro zvolený model dostatečnou vypovídací hodnotu [2].

Rais, Dostál a Sojka [2] dále uvádějí, že data mining probíhá v následujících krocích:

- 1) Získání dat. Firma může data získat buďto svépomocí, tedy jejich shromažďováním, nebo nákupem. Obě metody mají jak pozitiva, tak negativa.
- 2) Klasifikace dat. Data je třeba roztřídit podle předem daných kritérií, například na kvalitativní a kvantitativní data. Vždy je třeba si uvědomit, jak budeme s daty dále pracovat – fuzzy logika je schopná pracovat jak s kvalitativními tak i s kvantitativními daty, zatímco pro umělé neuronové sítě je nutné kvalitativní data kvantifikovat. Kvantitativní data mohou být nominální, ordinální, intervalová nebo spojitá.
- 3) Vzorkování dat. Používá se při velmi rozsáhlých datových souborech – náhodným výběrem se vybere reprezentativní vzorek.
- 4) Čištění dat. Představuje kontrolu, zda v datech nejsou chyby – například hodnoty mimo povolený rozsah či chybějící hodnoty. Chyba se nahrazuje hodnotou v akceptovatelných mezích.
- 5) Sumarizace. Používá se v případě velkého počtu dat za pomoci sčítání, odečítání a průměrování. Data se často agregují do denních, týdenních, měsíčních nebo ročních součtů a průměrů.
- 6) Redukce znamená snížení počtu dat jejich zúžením.
- 7) Segmentace rozčleňuje spojité proměnné do segmentů.
- 8) Transformace se provádí za účelem filtrace, například odstranění trendových a sezónních složek.
- 9) Tvorba modelu. Je nutné vybrat správný model. Fuzzy logika se používá při práci s kvantitativními daty, umělé neuronové sítě se používají při výpočtech binárních a dvouúrovňových vstupů (například reakce na nabídku) a genetické algoritmy se používají tam, kde potřebujeme určit shluky, provést optimalizaci nebo kde se větší počet náhodně vybraných modelů upravuje iterací a hledá ten nejlepší pro zadanou úlohu.
- 10) Ověření modelu je kritickou částí celého procesu, neboť je nutné odhalit chybná zdrojová data či nevhodné postupy. Mělo by se provádět na datech, která nebyla použita k tvorbě modelu. Dává-li model dobré výsledky i pro jiná data, lze

hovořit o robustním modelu s obecnou platností. V opačném případě je třeba provést tzv. převzorkování.

11) Implementace modelu je využití výsledků pro praktickou potřebu.

12) Údržba modelu. S postupujícím časem klesá kvalita vlastností modelu a tím i kvalita výsledků, které model poskytuje. Model je možné obnovit nebo v případě zásadních nových dat navrhnout zcela znovu.

Příklady použití data miningu jsou uvedeny ve stejné publikaci [2].

Jaké jsou možnosti aplikace této zajímavé metody v námi uvažovaném systému? Vzhledem k tomu, že v situaci univerzity neexistuje příliš možností k získávání nových či udržování stávajících zákazníků, je možné uvažovat pouze o vyhodnocování rizikového chování klientů. Univerzita jim samozřejmě nemůže už z principu odepřít přístup do systému a poskytování služeb. Je však možné navrhnout systém pro správu dlužníků. Dlužníkům pak může být nevyhověno v případě žádostí o speciální zacházení – například o prodloužení doby splatnosti pohledávky. Pro tento účel je využití data miningu naopak velice vhodné – při vyřizování žádostí není vhodné zatěžovat obsluhu systému procházením celé transakční historie daného klienta, je však možné nabídnout určitá agregovaná data, ideálně spolu s určenou klasifikací například z množiny bezproblémový klient, občasný prohřešek, notorický neplatič.

### 3 Analýza problému

Bylo by snad až naivní předpokládat, že Masarykova Univerzita dosud neprovozovala systém, který by splňoval požadavky, kladené na operativní účetní evidenci. Podívejme se nyní do historie a popišme si, jak univerzita došla až k potřebě implementace nového systému.

#### 3.1 Clearing

Původním řešením uvedeného problému byl systém nazývaný Clearing. Tato verze vznikla na konci minulého tisíciletí, tedy zhruba v letech 1998 – 2000. Funkčně byla na svou dobu velmi pokročilá, nicméně technologicky brzy přestala vyhovovat. Nyní je již několik let neprovozovaná a skončila v propadlišti dějin.

Přesto považuji za vhodné popsat alespoň ve stručnosti její principy. Systém vznikl jako nástroj pro zúčtování plateb za ubytování na kolejích. To s sebou samozřejmě neslo jak výhody, tak problémy.

Díky tomu, že byl systém napsán na míru jedné konkrétní aplikaci, mohl být relativně jednoduchý. Mezi jeho klienty patřili pouze studenti ubytovaní na kolejích, kterých v té době bylo okolo dvou tisíc. Každý z nich měl vlastní účet v systému, na který byly shromažďovány pohledávky a příchozí platby. V případě vzniku přeplatku pak bylo možné studentovi peníze vrátit.

Protože systém sloužil pouze pro platby za jeden druh služeb, stačilo, aby uměl komunikovat proprietárním protokolem s ubytovacím systémem Správy kolejí a menz Masarykovy univerzity (nazývaným ISKaM – Informační systém kolejí a menz). Veškeré příchozí platby tak měly přesně daný účel – sloužily k úhradě ubytování – a mohly být považovány za úhradu již vystavené faktury. Tím si univerzita velice zjednodušovala život, neboť odpadala celá složitá administrativa a účtování kolem přijatých záloh.

Systém také uměl uhradit pouze celou pohledávku naráz – nejlépe to snad ilustruje krátký příklad: Pokud měl student na účtu dvě pohledávky ve výši 1000 a 1000 Kč, a zaslal nejprve platbu ve výši 500 Kč, byly pohledávky stále považovány za

nezaplacené. Teprve po vkladu dalších 500 Kč bylo možné uhradit první z nich a po vkladu dalších 1000 Kč druhou.

Systém také neevidoval vazbu mezi platbou a pohledávkou. Peníze na klientském účtu neměly žádnou historii – jednalo se o homogenní hromádku, která rostla a klesala podle toho, jak platby přicházely na účet a byly používány na úhrady pohledávek či vráceny klientům.

Nebyla uchovávána žádná historie o klientských účtech – pouze zda je účet aktivní (a tedy plně funkční) či pasivní – zrušený.

Pro klienty systém nabízel pouze výpis z účtu s informacemi o pohybech, aktuálním stavu neuhrazených pohledávek a údajů nutných k realizaci bezhotovostního platebního styku. Pro správce systému pak několik sestav nutných pro úspěšné zaúčtování vystavených pohledávek a přijatých a vydaných plateb. Netřeba jistě dodávat, že veškeré účetní operace probíhaly ručně – účetní si vytiskl sestavy s celkovými částkami a následně vytvořil v IS Magion potřebné účetní doklady.

I přes popsaná omezení byl systém úspěšně několik let provozován a vaz mu srazilo až jeho technologické řešení. Celý systém byl totiž vytvořen jako množina databázových procedur v databázi Informix. V roce 2002 však bylo rozhodnuto, že veškeré informační systémy budou sjednoceny nad databázi Oracle a databáze Informix bude postupně zrušena. To byl impuls pro vytvoření nového systému, nazývaného Clearing 2.

Pro zajímavost je možné uvést, že poslední systém byl převeden do databáze Oracle v prvním čtvrtletí roku 2008 – jednalo se o systém personalistiky a zpracování mezd. Celý přechod tak Masarykově univerzitě zabral bezmála šest let, celkové náklady se ani neopovažuji odhadovat.

## **3.2 Clearing 2**

Systém, nazývaný Clearing 2, vznikl jako evoluce původního systému Clearing jeho kompletním přepsáním do prostředí univerzitního intranetu. Nejdříve si proto přiblížme jeho technické řešení.

Systém byl vytvořen jako soustava aplikací v jazyce Java, operující nad databází Oracle. To s sebou proti původnímu návrhu neslo mnoho výhod – zejména oddělení datové části od aplikační logiky. Také prezentační vrstva byla oddělena – aplikace

generovaly XML soubory, které byly následně pomocí XSL šablon transformovány do v běžném internetovém prohlížeči zobrazitelných HTML stránek. Protože v době vzniku systému, v roce 2003, byly internetové prohlížeče ve srovnání se současností na nízké úrovni, platila zásada, že veškeré operace musí probíhat na straně serveru – veškeré skriptování v klientské části systému bylo zapovězeno.

Přejdeme nyní k (jistě zajímavějšímu) popisu funkcí systému. Je nutné předeslat, že ne všemi funkcemi systém disponoval od počátku své existence – naopak během svého provozu byl relativně často rozšiřován o další funkce a aplikace, které byly nezbytné k jeho provozu. Popis tudíž bude vycházet ze stavu na konci života systému.

Jedním ze základních požadavků bylo zachování veškeré funkčnosti první verze systému. Zachovány zůstaly i veškeré principy původního systému.

Zaměříme se však na některé nové funkce, které přibýly v průběhu užívání systému. První z nich je rozšíření možností nabíjení účtu penězi. Zatímco první verze uměla pouze přijímat platby zaslané převodem, verze číslo dvě zvládala i hotovostní vklady pokladnou a automatické inkasování z bankovního účtu. Zejména druhý způsob považuji za významný a (ve své době) ne zcela běžný, proto si myslím, že si zaslouží podrobnější popis.

Podívejme se nyní stručně na vkládání hotovosti přes pokladnu. Princip byl jednoduchý: student přinese peníze na jedno z určených míst na kolejích Masarykovy univerzity. Paní pokladní následně vytvoří příjmový pokladní doklad v systému Magion a případně dá studentovi potvrzení. Tento pokladní doklad se pak zobrazí v aplikaci systému Clearing 2, odkud je možné nechat připsat částku studentovi na účet. Druhý krok však mohla provádět pouze správa systému. Tento způsob se vyznačoval značnou odolností proti zneužití a chybám, nicméně byl velmi nepružný – platby se do systému dostávaly s minimálně jednodenním zpožděním. Zejména zahraniční studenti však tuto možnost uvítali.

Inkasování peněz z bankovních účtů klientů probíhalo následujícím způsobem. Před prvním generováním inkasních příkazů musel student zajít osobně za pověřenou osobou (nejčastěji ubytovatelkou na kolejích) a odevzdat jí potvrzení o vlastnictví bankovního účtu, který mu byl následně zapsán do systému. V určené datum na začátku měsíce (zhruba mezi 5. a 10. dnem) byly následně vystaveny požadavky na inkaso tak, aby na klientských účtech byly ve stejný den – podle znalosti délky provádění byly nejdříve

generovány příkazy pro klienty s účty z bank, ze kterých převod peněz trval tři dny (takových byla většina), další den pro studenty s účty u banky, ze které převod trval dva dny a naposled pro studenty s účtem u stejné banky, jako má univerzita. Výše inkasa byla určena jako suma neuhrazených pohledávek na klientském účtu mínus aktuální zůstatek. Požadavky na inkaso byly následně vloženy do tzv. bankovního rozhraní systému Magion, ze kterého již automaticky odešly do banky. Po jejich vyřízení (stejně jako po příchodu peněz zaslaných příkazem k úhradě) byly bankovní výpisy opět vloženy do bankovního rozhraní, odkud si je systém Clearing 2 dokázal již sám vytáhnout, spárovat s požadavky a peníze připsat na klientské účty. V případě, že inkaso bylo bankou odmítnuto, další požadavek ve stejném měsíci generován nebyl, protože platil oprávněný předpoklad, že by též nebyl proveden.

Výhodou tohoto způsobu placení byla nezávislost na klientech. Stačilo zřídit v bance povolení k inkasu a zanést potvrzení ubytovatelce a veškeré další platby již probíhaly automaticky. I způsob generování příkazů, ačkoliv byl určen experimentálně, se ukázal jako vhodně zvolený – zcela vyhovoval požadavkům na něj kladeným.

Dále byl systém rozšířen o možnosti generování ekonomických sestav pro podporu účetnictví – ovšem jednalo se o jednu velmi jednoduchou aplikaci, ve které bylo možné zjistit celkovou částku přijatých a vydaných plateb nebo pohledávek za určité období. Přesto na této jednoduché sestavě bylo po několik let založeno celé účtování, týkající se systému Clearing 2.

Systém také poskytoval prostředky pro exportování svých dat do XML souborů pro jejich případné další zpracování v jiném systému, či pro jejich zálohování. Přibyl i výpis příchozích plateb, které nebylo možné automaticky spárovat se správným klientským účtem (například z důvodu zadání chybného variabilního symbolu). Podle tohoto výpisu mohl každý klient, v případě, že jím odeslaná platba nenavýšila zůstatek na jeho účtu, zjistit, zda se jeho platba nenachází v seznamu neidentifikovatelných. V kladném případě pak bylo nezbytně nutné, aby dodal příkaz k úhradě (či jeho kopii) oprávněným osobám, které mohly pomocí další aplikace platbu přiřadit na správný účet.

### 3.3 Zhodnocení současného stavu

Systémy Clearing a jeho přímý nástupce Clearing 2 vyhovovaly Masarykově univerzitě několik let. Situace se však změnila a žádá si nový přístup. Podívejme se proto nyní, v jakých ohledech současný systém univerzity nevyhovuje.

Vedení univerzity rozhodlo, že je vhodné, aby veškeré služby poskytované univerzitou byly hrazeny z jednoho konta. K ubytování na kolejích tak přibude tisk na jednotlivých fakultách, prodej kancelářských potřeb pro studijní účely, stravování v menzách a úhrady soukromé telefonie zaměstnanců. Pro univerzitu je takovéto řešení výhodné, protože odpadá nepříjemná manipulace s hotovými penězi. Dá se také očekávat vyšší obrát (a tím pádem i vyšší zisk) u některých poskytovaných služeb – lidé neradi dávají z ruky hotové peníze; v případě, že veškeré transakce budou hrazeny bezhotovostně „kartou“, je možné očekávat častější nákupy a vyšší útraty.

O možnost hradit nákupy z univerzitního konta také projevil zájem několik samostatných komerčních subjektů, majících provozovny v budovách univerzity. Konkrétně se jedná o několik knihkupectví a o firmu, provozující nápojové a jídelní automaty.

Celkově se tak jedná o tři kategorie služeb, které by měly být systémem hrazeny. První z nich jsou služby, které jsou poskytovány univerzitou, probíhají opakovaně a po delší dobu a není nezbytně nutné, aby byly uhrazeny okamžitě. Typickým příkladem je ubytování na kolejích.

Druhou skupinou jsou služby, které jsou poskytovány univerzitou a mají jednorázový charakter. Je tudíž vhodné, aby nebyly poskytovány na dluh. Jako příklad je možné uvést tisk a kopírování v univerzitních studovnách.

Poslední skupinou jsou služby, které neposkytuje univerzita. I zde je vhodné, aby je klient uhradil okamžitě (nebo, lépe řečeno, aby služba nebyla poskytnuta, pokud na klientském účtu není dostatečný zůstatek pro její úhradu). Teoreticky by bylo možné tyto služby zahrnout do druhé skupiny a nerozlišovat jejich poskytovatele; z účetních důvodů je však vhodné je alespoň pro analýzu systému rozdělit. Pokud se ukáže, že se k nim dá přistupovat stejně, bude moci být systém jednodušší.

Rozlišení služeb do třech skupin s sebou však nese nutnost zásadní změny systému. Jistě, bylo by možné do Clearingu 2 nové vlastnosti doimplementovat. Je však otázka,

zda by tím zásadně neutrpěla stabilita již několikrát rozšiřovaného systému, který nebyl připraven na odhadovanou zátěž.

Přidáme-li k tomu předpokládané požadavky na zprovoznění automatického účtování služeb a plateb, které jistě vzejdou z ekonomického odboru rektorátu a požadavek na vytvoření systému správy pohledávek – neboť se vzrůstajícím počtem studentů a poskytovaných služeb bude přímo úměrně vzrůstat i počet neuhrazených služeb, zjistíme, že ideálním postupem je napsat celý systém znovu a lépe s využitím zkušeností z tvorby a provozu předchozích systémů.

## 4 Vlastní návrhy řešení

Z předchozích kapitol vyplynulo, že je potřeba zavést zcela nový systém, který by řešil požadavky na něj kladené lépe, než to dokáže stávající řešení.

Než však přistoupíme k samotnému řešení, je třeba se zastavit a zauvažovat. Půjdeme cestou, kterou jsme již jednou šli, a vytvoříme systém vlastními silami, nebo zvolíme hotové řešení? Odpověď není jednoduchá a ovlivní provoz Masarykovy univerzity na několik budoucích let.

Pro úplnost je však nutné dodat, že výběr varianty proběhl bez toho, aby do něj autor práce mohl jakkoliv zasáhnout. Srovnání uvažovaných variant tak lze považovat spíše za teoretickou možnost výběru; výhodněji však i tak vyšla zvolená varianta.

### 4.1 Varianta 1: Dodání systému externím dodavatelem

První z variant, jak již nadpis napovídá, je dodávka systému externím dodavatelem. Srovnajme si proto silné a slabé stránky tohoto řešení, nejdříve bez ohledu na to, zda vůbec existuje společnost schopná dodat systém, který by splnil požadavky, a zároveň po několik následujících let poskytovat podporu za akceptovatelné ceny. Budeme však předpokládat, že by se jednalo o systém, který již existuje – v případě zadání vývoje nového systému externí společnosti by mnoho z pozitiv odpadlo.

Protože nepotřebujeme nijak přesné metody analýzy, použijeme známou metodu SWOT – (S)trengths, (W)eaknesses, (O)pportunities, (T)hreats, čili silné a slabé stránky, příležitosti a hrozby uvažovaného řešení.

#### 4.1.1 Silné stránky

**Rychlost nasazení** – již existující systém (s drobnými až středně rozsáhlými úpravami) bude jistě možné dodat během relativně krátké doby (v řádu měsíců).

**Cena systému** – pravděpodobně bude nižší než odpovídající mzdové náklady na vlastní vývojáře. Soukromé společnosti navíc pracují většinou efektivněji.

#### 4.1.2 Slabé stránky

**Kvalita systému** – v případě dodávky externí firmou není možné mít dokonalou kontrolu nad kvalitou systému, který může být na první pohled funkční, avšak při zvýšené zátěži či požadavcích na rozšíření může rychle začít vykazovat neuspokojující chování či rychlost.

**Rychlost reakce** – v případě potřeby úpravy systému či zásahu do databáze, kterou by nezvládli technici na univerzitě se reakční doba může prodloužit, což může mít nepříjemné následky zejména v době účetní uzávěrky.

**Nutnost udržovat podporu na univerzitě** – pro běžnou správu systému je nutné udržovat technickou podporu přímo na univerzitě. Může se jednat o zaměstnance externí firmy, který však bude pravděpodobně většinu času nevytížen. V případě, že se bude jednat o zaměstnance univerzity, je nutné, aby dokonale znal systém a byl tak schopen operativně řešit co nejvíce problémů. Po většinu pracovní doby se může věnovat jiné agendě.

**Vedení univerzity** – vedoucí nepreferují tuto variantu řešení. Pracovníci oddělení zodpovědného za vývoj a provoz informačních systému by museli přesvědčit celé vedení univerzity (tedy rektora, kvestora, prorektory a tajemníky jednotlivých fakult) o výhodnosti uvedeného postupu.

#### 4.1.3 Příležitosti

**Snížení stavu pracovníků** – v budoucích letech se dá očekávat tlak na snížení mzdových nákladů ve státních a příspěvkových organizacích. Out-sourcinglem dodávky požadovaného systému je možné snížit náklady a přesunout finance z mezd na jiný účel. Ve skutečnosti se vydá jen o něco méně prostředků, nicméně ve výhodnější formě.

#### 4.1.4 Hrozby

**Zánik firmy** – pokud dodavatelská firma zanikne během používání systému, vystavuje se univerzita velkému riziku.

**Nedodání vyhovujícího systému** – pokud univerzita stráví příliš dlouhou dobu hledáním vhodného dodavatele či pokud vybraný dodavatel nedokáže dodat vyhovující systém, ztratí univerzita značné množství finančních prostředků a času.

**Nemožnost rozšíření systému** – v případě, že nebude možné rozšířit již provozovaný systém o další požadované funkce či vlastnosti (například z důvodu legislativních změn), ocitne se univerzita opět na počátku analýzy problému a výběru vhodného systému.

#### **4.1.5 Zhodnocení varianty č. 1**

Ačkoliv uvedená varianta má určité výhody, nedokáže tyto překonat slabé stránky navrhovaného řešení. Největší překážkou je preference vedení univerzity, které má svůj vlastní názor na další směřování a koncepci rozvoje informačních systémů. To je samozřejmě zcela legitimní a v případě existující historie informačních systémů i dobře pochopitelný stav – předchozí systémy se ukázaly jako úspěšné a je tudíž zbytečné začít hledat nějaká nová „ad hoc“ řešení.

V případě, že by vedení univerzity nemělo takto jednoznačný názor, byla by varianta dodání systému externí firmou pravděpodobně též zamítnuta – i když nejspíše ne tak rychle a jednohlasně. Musela by proběhnout celá řada analýz a diskusí, ze kterých by vyplynulo nejvýhodnější řešení (případně řešení podporované nejvýznamnějšími představiteli univerzity). Pokud by však bylo i v tomto případě zvoleno vytvoření systému vlastními silami, jednalo by se jen o ztrátu času a finančních prostředků. Zejména v případě časové ztráty se univerzita vystavuje postihu ze strany finančního úřadu za neplnění daňových povinností.

## **4.2 Varianta 2: Vytvoření systému vlastními silami**

Po zavrnutí předchozí varianty zbývá již jen varianta číslo 2 – vytvoření zcela nového informačního systému s využitím zkušeností z vytváření předchozích systémů. Tento přístup bude náročnější jak finančně tak personálně, ovšem přinese plnou kontrolu nad výsledkem po celou dobu jeho používání. V případě, že bychom chtěli udělat SWOT analýzu, zjistili bychom, že platí přesný opak uvedených vlastností než u varianty číslo 1.

Vezměme proto způsob změny systému za daný a pokusme se jej zanalyzovat s ohledem na dříve uvedené požadavky, navrhnout jeho funkčnost a nakonec jej i vytvořit za použití dostupných programových prostředků.

### 4.3 Návrh systému

Z předchozího vyplývá, že jsme se rozhodli vytvořit systém vlastními silami. Před započítím jeho návrhu je však třeba učinit ještě jedno další rozhodnutí, které ovlivní celou koncepci systému. Jedná se o dilema, zda systém používat jako tzv. „elektronickou peněženku“, nebo-li používat systém jako ekvivalent bankovního účtu a platby kartou u obchodníka. Druhou volbou je postavit systém jako „operativní účetní evidenci“ a veškeré operace s penězi provádět jako účetní operace – tedy je považovat za přijaté zálohy nebo platby na úhradu faktury. Systém samotný bude sloužit jako podklad pro účtování a bude poskytovat detaily k souhrnným dokladům v účetním systému.

První způsob je implementačně mnohem jednodušší – stačí sledovat příchozí platby a výdaje, nezajímá nás, z které platby byl který výdaj uhrazen. Jádrem systému tak může být menší a rychlejší. S tímto způsobem se však pojí nepříjemná komplikace v podobě nutnosti získat bankovní licenci od České národní banky. V současné době existují dva stupně této licence – první, jehož získání je víceméně administrativní záležitostí, má bohužel omezení na uchování platebního prostředku pro klienta ve výši odpovídající nejvýše 150 euro (v závislosti na platném kurzu 4000 – 4500 Kč). To je poměrně silné omezení, které by funkci systému značně limitovalo, protože takováto částka by mohla být snadno vyčerpána i jednou platbou za měsíční ubytování na některém z luxusnějších pokojů na kolejích MU. Druhý stupeň licence umožňuje volnější nakládání s prostředky, jeho získání je však značně obtížnější a pro univerzitu by pak platila podobná pravidla jako pro banky. Mimo jiné je získání této licence podmíněno držením minimálních rezerv u ČNB a aplikováním bezpečnostních protokolů pro zaměstnance, studenty i jednotlivé transakce.

Z tohoto důvodu bylo zvoleno řešení druhé, tedy pojmout celý systém jako operativní účetní evidenci. Řešení to bude jak analyticky, tak implementačně náročnější, nicméně bude průchozí z pohledu vedení univerzity a požadavků ČNB. Důsledkem tohoto rozhodnutí pak bude vytvoření systému, který je v rámci prostředí vysokých škol v České republice zcela unikátní. Systém bude splňovat veškeré požadavky, kladené zákony o účetnictví a o dani z přidané hodnoty.

V tento okamžik se také mění přístup k práci. Zatímco výběr formy dodání systému a zadání požadavků přicházelo shora – od vedení univerzity a budoucích uživatelů, od

tohoto bodu byla práce předána řešitelskému týmu Ústavu výpočetní techniky, jehož součástí je i autor celé této diplomové práce.

#### **4.3.1 Hrazené služby**

Máme-li navrhovat systém pro úhradu služeb je nutné nejdříve vědět, které služby univerzita poskytuje a přeje si je mít hrazené z jednotného peněžního konta. Proto si je nyní vyjmenujme a popíšme pro lepší představu o požadovaných funkcích systému.

První ze služeb, která byla hrazena již pomocí předchozích systémů, je ubytování na kolejích. Pro rezervaci lůžka vyžaduje Správa kolejí a menz zaplacení předubytovací kauce – většinou v průběhu léta. Po zaplacení kauce se student musí ubytovat do předem daného data, případně svou rezervaci zrušit. V takovém případě dostane peníze zpět, nevyhoví-li těmto podmínkám, jeho částka propadá bez náhrady. Po ubytování je třeba zaplatit tzv. ubytovací kauci, která slouží jako pojistka v případě neuhrazeného posledního ubytování nebo vzniklých škod. Tato kauce je vrácena studentovi v okamžiku, kdy se z kolejí odubytuje.

Každý měsíc pak student platí za ubytování a služby s ním spojené – a to jak využívané celý měsíc (například internet na pokoji nebo zapůjčení počítače), tak jednorázové (využití pračky nebo zapůjčení vysavače). V případě pozdní úhrady je student penalizován – režim si ovšem řídí ubytovací systém sám a nepožaduje jeho implementaci do nově vytvářeného systému. V některých případech je nutné mít možnost snížit výši pohledávky za studentem o určitou částku, případně pohledávku zcela zrušit a peníze vrátit studentovi – například pokud byla studentovi naúčtována služba ve vyšší výši nebo vícekrát než by odpovídalo skutečnému stavu.

Pro správnou funkčnost musí náš systém umět přijímat od ubytovacího systému požadavky na vložení pohledávky za studentem, kauce, uvolnění nebo propadnutí kauce a snížení výše pohledávky. Ubytovacímu systému musí být schopen předávat data o úhradách pohledávek a potvrzovat provedení požadavků. Služby je možné poskytovat na dluh, tedy není nezbytně nutné, aby student měl před poskytnutím služby na svém účtu dostatečný zůstatek na její úhradu.

Druhou službou jsou tisky a kopírování v jednotlivých počítačových studovnách po celé univerzitě. V současné době jsou tisky obhospodařovány systémem SafeQ brněnské firmy YSoft v různých verzích. Pro jejich provoz stačí, když bude systém schopen přijímat požadavky na úhradu tisku. Od ubytování se však služby tisku liší ve dvou

zásadních parametrech. Za prvé není možné je poskytovat na dluh, tudíž se tiskový systém musí našeho systému nejdříve dotázat, zda má klient dostatečný zůstatek na svém účtu pro vytištění celé úlohy a v negativním případě odmítnout službu poskytnout. Za druhé, zatímco v případě ubytování na kolejích se jedná o jednotky pohledávek za studenta a měsíc (a vzájemná komunikace systémů tak nemusí být nijak intenzivní), u tisků může – zejména v případě zkouškových období – jít denně o desítky položek pro každého návštěvníka počítačové studovny. Systém proto musí být schopen komunikovat a reagovat na požadavky dostatečně rychle – tak, aby tiskový systém nebyl zdržován ve svém provozu.

Třetí službou, která má být hrazena, je prodej drobného zboží. V tomto případě může být prodejcem jak univerzita, tak i externí komerční subjekt (předpokládá se, že do začátku bude služba nabízena jen společností, které mají provozovnu na půdě univerzity). Režim se bude zřejmě lišit podle toho, kdo zboží prodává, nicméně podobně jako u tisků platí, že student musí mít na svém účtu dostatečný zůstatek na koupi požadovaného zboží.

Další službou, kterou je vhodné hradit z jednotného univerzitního konta, je stravování v menzách. Ačkoliv by se mohlo zdát, že by mělo podléhat stejnému režimu jako tisky nebo drobný prodej, není tomu tak. Zatímco v případě výpadku systému nebo síťové komunikace a neposkytnutí tiskové služby je možné od studentů očekávat pochopení, v případě nevydání pokrmu je situace značně odlišná. Hladoví studenti mají tendence k protestům a navařená jídla také není možné uschovávat po neomezenou dobu. V současné době teprve probíhá jednání se Správou kolejí a menz, jehož finální výsledek je zatím možné pouze odhadovat. Velmi pravděpodobně však budou studenti jistým způsobem úvěrováni – tedy budou jim poskytována jídla na dluh do určité výše či počtu obědů, nebo si budou z celouniverzitního systému „nabíjet“ svoje konto v menze. Je téměř jisté, že tato služba nebude spuštěna ihned, ale teprve po důkladném odzkoušení celého systému a jeho případném otestování na části zaměstnanců.

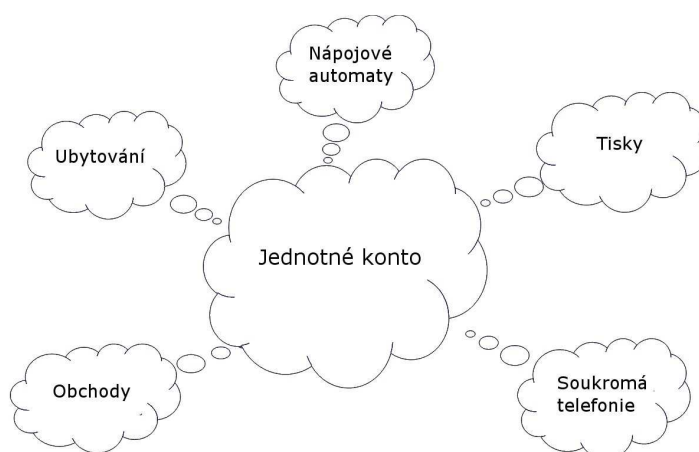
Předposlední uvažovanou službou je provoz nápojových a jídelních automatů. Podobně jako u tisků je poskytnutí služby (vydání nápoje či bagety) podmíněno dostatečným zůstatkem na účtu. Z důvodu bezpečnosti – uvažované automaty totiž nemají v sobě integrovanou klávesnici pro zadání pinu – a z důvodu vyhovění podmínkám ČNB bude

nutné implementovat uživatelem volený limit na platbu pomocí identifikační karty – podobný, jakému podléhají platební karty vydávané bankami.

A konečně, protože klienty systému mohou být i zaměstnanci univerzity, nabídne jim systém i možnost hradit soukromé hovory ze služebního telefonu – z pevné linky i z mobilního. To by samozřejmě nebylo dostatečně atraktivní bez možnosti nechat si svůj celouniverzitní účet nabíjet převodem ze mzdy. Po zpřístupnění této možnosti odpadne značně nepříjemná manipulace s drobnými částkami (v řádu korun či desetikorun) v hotovosti.

#### 4.3.2 Podrobnější pohled na služby

Z předchozí podkapitoly vyplynulo, že služby je možné rozdělit podle dvou různých kritérií. Protože náš systém žádné služby poskytovat nebude, nazýváme služby poskytující systémy „externí“. Vztah jednotného konta a externích systémů je naznačen na obrázku 3.



**Obrázek 3: Jednotné konto jako spojující uzel pro hrazené služby**

Prvním ze zmíněných kritérií je možnost poskytnutí služby na dluh nebo ne a s tím související režim předávání požadavků na úhradu služby. Pokud je externí systém ochoten poskytovat službu na dluh, měl by předávat data o provedených službách

dávkově – například jednou denně. Tyto systémy nazýváme „off-line“ systémy. Typickým reprezentantem je ubytování na kolejích.

V opačném případě musí externí systém předávat požadavky na úhradu okamžitě – je totiž možné, že student bude službu využívat opakovaně (například si bude před zkouškou tisknout různé materiály) a mohla by mu tak být poskytnuta i v případě nedostatečného zůstatku. Proto je nutné pohledávku předat ihned a zároveň ji ihned zpracovat. Typickým zástupcem této skupiny jsou již zmíněné systémy tisku a kopírování – nazýváme je „on-line“ systémy.

Druhé podstatné rozdělení je zda je poskytovatelem služby univerzita nebo nějaký jiný, komerční, subjekt. V prvním případě je možné vystavit pohledávku za studentem (nebo-li v účetní terminologii odběratelskou fakturu), kterou student musí uhradit k datu její splatnosti. Pohledávka má přiřazenu sazbu DPH podle platného zákona, poměrnou část z hodnoty pohledávky musí univerzita odvést státu. Také je nutné klientovi vystavit daňový doklad na odebranou službu.

V druhém případě, tedy pokud univerzita není poskytovatelem služby, je nutné úhradu služby pojmout jako výdej platby externímu subjektu podle podmínek ČNB. Univerzita v tomto případě nárokuje případné odvedené DPH ze zálohy zpět a o zdanění služby se stará její poskytovatel ve vlastní režii.

#### **4.3.3 Návrh jádra systému**

Pokud nyní víme, jaké služby budeme naším systémem hradit, je možné začít navrhovat jeho jádro tak, aby vyhovělo veškerým požadavkům na něj kladeným.

Jako základní stavební prvek zvolíme klientský účet. Připomeňme si, že mít účet dovolíme kterékoliv osobě s aktivním vztahem k univerzitě. Z právních důvodů však nebudeme zřizovat účty osobám automaticky, ale bude nutné vyžadovat jejich aktivaci klientem – nejprve stvrzením souhlasu v Inetu a následně potvrzením souhlasu tzv. faktickým konáním. Jako faktické konání bylo zvoleno buď složení minimální zálohy na účet (ve výši 50 Kč) nebo podepsání souhlasu s převodem ze mzdy. Je možné, že jak budou připojovány další externí systémy, rozšíří se repertoár faktického konání o další způsoby.

Z uvedeného je zřejmé, že bude nutné evidovat pro jednotlivé účty jejich stavy. Počátečním stavem účtu bude pasivní. V tomto stavu budou zakázány téměř všechny

činnosti, klient naopak bude vyzván k potvrzení svého souhlasu s podmínkami provozování systému. Stvrdí-li jej, převede účet do stavu „aktivovaný“ a bude vyzván k potvrzení aktivace faktickým konáním. Bude mu také poskytnut variabilní symbol, pod kterým může nadále zasílat platby na svůj účet po celou dobu jeho životnosti.

Po provedení faktického konání se účet dostane do plně aktivního stavu, ve kterém se bude nacházet po většinu doby svého užívání. V tomto stavu budou povoleny veškeré operace s účtem – přijímání a vydávání plateb, přijímání pohledávek, nastavování parametrů účtu apod.

V okamžiku, kdy klient přestane být na univerzitě aktivním, bude jeho účet převeden do stavu „expirovaný“, ve kterém se bude chovat stejně jako v plně aktivním stavu. V tomto stavu se bude účet nacházet nejdéle tři měsíce. Pokud do té doby klient opět získá aktivní vztah k univerzitě (například zápisem k navazujícímu studiu), bude účet automaticky převeden zpět do plně aktivního stavu. V opačném případě bude po uplynutí uvedené časové lhůty převeden do pasivního stavu.

Tímto byl popsán základní životní cyklus klientského účtu. Z technických a administrativních důvodů zavedeme dva další stavy. Prvním z nich bude stav „vyřizovací“. Pokud vznikne za klientem pohledávka v okamžiku, kdy na univerzitě již nepůsobí a má pasivní účet, bude tento na dobu nezbytně nutnou k úhradě pohledávky převeden do vyřizovacího stavu. Následně bude převeden zpět do pasivního stavu.

Druhým pomocným stavem bude stav „vypovězený“. Bohužel je možné, že klient nebude platit své pohledávky včas a situace dojde až tak daleko, že bude částka vymáhána soudně. V případě úspěchu univerzity v soudním řízení bude klientův účet převeden do vypovězeného stavu – nebude moci využívat další služby do té doby, než uhradí pohledávky a soudní výlohy a v případě zaslání platby na účet si nebude moci vrátit přebytečné peníze.

Z bezpečnostních důvodů (a z důvodů požadovaných ČNB) si klient bude moci volit také vyšší limitu pro platby třetím stranám a to, zda se jedná o limit denní nebo týdenní.

Z technického hlediska bude klientský účet reprezentován záznamem v databázi a kromě svých parametrů (jako je vlastník účtu, variabilní symbol, stav) bude mít u sebe uloženy také předpočítané aktuální zůstatky, zbývající částky k úhradě a aktuálně vyčerpanou částku z nastaveného limitu. Tím dosáhneme značného zrychlení operací - ovšem za cenu duplikace dat v databázi. Bude tudíž nutné dávat pozor na udržení

konzistence dat (sumy položek oproti uloženým zůstatkům) a čas od času provést kontrolu pro zjištění případných nesrovnalostí.

#### **4.3.4 Externí systémy**

O externích systémech a službách padla zmínka v předchozí kapitole. Podíváme-li se na ně z technického hlediska, zjistíme, že bude nutné si uchovávat informace o externím systému – jeho název, kontaktní osobu, která tento systém spravuje a místo, kde se o tomto systému bude účtovat.

Každý systém může poskytovat více služeb, z nichž každá se účtuje samostatně. Budeme si evidovat systém, ke kterému služba patří, její název, zda je v současné době používána a pro lepší orientaci kód služby v externím systému, který bude společným identifikátorem služby v našem i v cizím systému.

K uchování těchto údajů proto budeme potřebovat dvě databázové tabulky, spojené relací 1:n – každý systém může mít více služeb, každá služba patří právě do jednoho systému.

#### **4.3.5 Položky**

Pokud již máme zaveden klientský účet, je třeba podobným způsobem zavést druhý základní stavební kámen, kterým budou jednotlivé položky, přicházející na účty.

Každá položka bude reprezentovat provedení akce v systému, která se nějakým způsobem promítne do účetnictví. Vzhledem k různorodosti prováděných akcí budou položky různých typů, o položkách stejného typu se však bude účtovat vždy stejně.

Položky rozdělíme do dvou velkých skupin. První z nich bude vznik finančního plnění (ať ve prospěch univerzity nebo ve prospěch studenta), druhou pak tzv. průběžná položka – například úhrada pohledávky studentem.

Podívejme se nejprve na první skupinu. Veškeré položky budou mít určité společné vlastnosti. Nejprve je nutné znát, komu položka patří, přesněji ke kterému z klientských účtů se vztahuje. Další důležitá vlastnost je částka položky, její popis (určitý text, upřesňující položku – podobně jako poznámka na účetním dokladu), k jaké službě se položka vztahuje, kategorii DPH položky (5%, 10%, 20%, osvobozeno od daně, nepodléhá dani...), datum vystavení položky, datum jejího uskutečnění (od něj se odvíjí zdaňovací období, do kterého položka patří), stav v jakém se položka v současné době

nachází, jednoznačná identifikace položky v našem systému, jednoznačná identifikace položky v externím systému, účetní období, do kterého položka patří a identifikace účetního dokladu, kterým je položka zanesena do účetnictví. Kromě těchto společných vlastností má každý jednotlivý typ položky své specifické vlastnosti, na které se podíváme nyní.

Prvním z typů položek je pohledávka, která vzniká v okamžiku, kdy univerzita poskytuje studentovi placené služby případně požaduje po studentovi doplňkovou platbu – například penále. Pohledávka se může nacházet ve třech stavech – neuhrazená, částečně uhrazená a plně uhrazená. Podstatnou vlastností pohledávky je její splatnost, nebo-li datum, ke kterému má být uhrazena. Další specifické vlastnosti pohledávek jsou její základ a DPH, kolik zbývá uhradit celkem, jaká částka byla dobropisována a jaká byla odepsána.

Tím nám vznikají dva další typy položek – dobropis a odpis pohledávky. Oba mají podobné vlastnosti. Dobropis představuje snížení hodnoty pohledávky o danou částku (například z důvodu nekvalitního provedení služby) – je možné dobropisovat i zcela uhrazenou položku, v kterémžto případě vzniká přeuhrazená pohledávka a klientovi jsou vráceny peníze. Jednu pohledávku je možné dobropisovat i vícekrát, celková suma částek dobropisů však nesmí přesáhnout částku původní pohledávky. Naproti tomu odpis představuje přiznání finanční ztráty z neuhrazené pohledávky, jejíž hodnota je natolik nízká, že se nevyplatí ji vymáhat soudně. Odepisuje se vždy celá neuhrazená část pohledávky – z toho vyplývá, že ke každé pohledávce může existovat nejvýše jeden odpis.

K základním vlastnostem proto bude stačit dále evidovat, která položka je dobropisována nebo odepisována pro jednoznačnost vazby.

Dalším typem položek jsou kauce. Ty vyjadřují požadované složení určité částky klientem, která mu však bude po vyhovění podmínkám vrácena. Kauce se chovají podobně jako pohledávky s několika drobnými rozdíly. Za prvé, peníze, použité na úhradu kaucí stále patří klientovi, nikoliv univerzitě – jen s nimi nemůže volně nakládat. Za druhé, kauce nepodléhají dani z přidané hodnoty a neúčtuje se o nich. Položky, reprezentující kauce, si tak budou nést kromě základní množiny vlastností ještě informaci o tom, kolik zbývá z kaucí uhradit.

Ke kaucím se vztahují další dva typy položek. Prvním z nich je uvolnění kaucí, vyjadřující vrácení peněz klientovi po splnění podmínek, vázajících se ke kauci, druhou propadnutí kaucí ve prospěch univerzity v případě nesplnění podmínek. O druhém z těchto dvou případů se již účtuje běžným způsobem – peníze jdou do výnosů organizace.

Uvolnění i propadnutí kaucí má k běžné sadě atributů navíc identifikaci kaucí, ke které se vztahuje. Obě položky mají také vždy stejnou částku jako je uhrazená část kaucí v okamžiku, kdy jsou přijaty.

Předchozí typy položek se týkaly především toku peněz z klientského účtu ve prospěch univerzity. Je však nutné zaručit i tok peněz na klientské účty – laicky řečeno nabíjení účtů. Proto bude nutné zavést další typ položky, nazývaný platba přijatá. U tohoto typu se na chvíli pozastavme. Protože je náš systém koncipován především jako operativní účetní evidence, je potřeba podle toho nakládat s přijatými platbami, zejména co se jejich vztahu k DPH týče. Ačkoliv platba přijatá bude reprezentovat příchod peněz na klientův účet, bude se moci dělit na několik dalších kategorií podle současného stavu účtu a tedy podle (implicitního) určení jejího účelu. Pokud bude na klientském účtu v okamžiku příchodu peněz splatná pohledávka (nebo-li pohledávka, jejíž datum splatnosti je dnes nebo v minulosti), považují se příchozí peníze za platbu určenou na úhradu této pohledávky a kategorizují se jako „platba přijatá po uskutečnění zdanitelného plnění“. V opačném případě se peníze považují za přijatou zálohu, kterou je nutné podle zákona zdanit aktuální sazbou DPH. V případě, že je záloha v budoucnu použita na úhradu pohledávky s jinou než základní sazbou DPH, vyžádá si univerzita poměrnou část daně od státu nazpět. Konečně poslední kategorií, do které může přijatá platba spadnout, je závazek univerzity vůči klientovi, který vzniká v okamžiku, kdy univerzita vrací klientovi jeho peníze – například při dobropisování pohledávky nebo uvolnění kaucí. Tato platba se považuje za již zdaněnou a nedaní se proto podruhé.

Přijatá platba se může i dělit na části – pokud v okamžiku jejího příchodu jsou na klientově účtu splatné pohledávky v nižší výši než je částka platby, stane se z poměrné části platby záloha a z poměrné části platba přijatá po uskutečnění zdanitelného plnění.

Konečně poslední typem položky bude platba vydaná – jak již název napovídá, bude reprezentovat výdej peněz z klientova účtu jinému subjektu, kterým může být buď

klientova banka nebo komerční firma, poskytující klientovi služby. Položka se však ve všech případech bude chovat stejně a bude mít pouze základní sadu atributů.

Tím máme popsánu první skupinu položek a můžeme se podívat na druhou, kterou tvoří tzv. průběžné položky. Bude se konkrétně jednat o úhradu pohledávky, kauce nebo platby vydané (ač zde terminologie trochu pokulhává) z platby přijaté. Jak z účetních, tak z daňových důvodů je totiž nutné vědět, jaká částka pohledávky byla uhrazena ze zálohy či z platby přijaté po uskutečnění zdanitelného plnění, či jaká částka zálohy byla vydána mimo univerzitu (pro nárokování vrácení DPH od státu). Fyzicky bude tento druh položek reprezentován vazební tabulkou, ve které budou uloženy veškeré náležitosti – tedy která položka je uhrazována, z které platby, jakou částkou a kdy se tak stalo. Připomeňme si, že je možné jednu položku uhradit více platbami a naopak jedna platba může hradit více položek – vše zcela nezávisle na sobě, v různých účetních i zdaňovacích obdobích.

#### **4.3.6 Komunikace s okolními systémy**

Jak bylo řečeno dříve, náš systém si – až na výjimky – žádné položky sám vymýšlet nebude. Je proto potřebné zajistit nějaký způsob komunikace s okolím.

Nejjednodušším řešením by vzhledem k tomu, že většina externích systémů bude provozována v síti Masarykovy univerzity, bylo povolit jim přímý přístup do databáze poskytnutím databázových procedur. Tomuto se však chceme vyhnout – jak z bezpečnostních hledisek, tak z důvodu ustupování od využívání databázových procedur ostatními systémy provozovanými na univerzitě. Z toho důvodu bylo – po diskusích se zástupci externích systémů – zvoleno řešení komunikace přes XML protokol, který bude zabezpečeným spojením zasílán na server s naším systémem, kde bude následně zpracován a odesilatelci bude vrácena odpověď.

Výhody takového řešení jsou mnohé. Především je takováto komunikace relativně bezpečná - spojení je zašifrované, požadavky jsou přijímané jen z určitých IP adres a veškerá komunikace je zaznamenávána do logů. Dále je bezpečný samotný proces zpracování – všechny příchozí požadavky jsou validovány tak, aby byly přijaty jen ty, které jsou konzistentní s v systému zavedenými pravidly (tedy například je nemožné v současné době přijmout pohledávku, podléhající 18% DPH – začne-li však takováto sazba DPH někdy v budoucnu platit, pohledávka bude přijata bez problému). A konečně

– implementovat generování XML souboru podle daných pravidel a jeho odesílání na určenou adresu je velmi jednoduché, způsob komunikace proto klade minimální obtíže připojování dalších služeb od rozličných poskytovatelů.

Podívejme se nyní na komunikaci XML protokolem podrobněji. Zjistíme, že budeme potřebovat zpracovávat dva druhy požadavků – prvním z nich je vložení položky na účet, druhým dotaz na stav položky nebo účtu. Pro urychlení komunikace a snížení zátěže serverů na obou stranách povolíme více příkazů v jednom zaslaném souboru.

První druh požadavku, vložení položky na účet, je triviální – externí systém zadá druh položky, identifikaci účtu, o jakou službu se jedná, částku a ostatní potřebné náležitosti. Náš systém přijatá data zkontroluje a v případě výskytu chyby tuto oznámí externímu systému. V opačném případě položku vloží do databáze a následně ji zpracuje. Úspěch opět oznámí externímu systému. Externí systém musí být schopen adekvátně reagovat na všechny chyby – reakce na pokus o zaslání duplicitní položky (například po chybě v komunikaci, kdy k němu nedorazila odpověď o přijetí) musí být jiná než reakce na chybu, značící pokus vložit pohledávku neexistující osobě.

Druhým typem komunikace je dotaz na stav položky či účtu. V prvním případě se externí systém ptá, zda položka byla již uhrazena – to je důležité například pro ubytovací systém, který v případě pozdních úhrad penalizuje studenty a může jim i (v případě opravdu špatné platební morálky či vysokých dluhů – tato pravidla jsou však zcela v režii správy kolejí a nás tak nemusí příliš zajímat) zrušit ubytování na kolejích. V druhém případě se externí systém – typicky tiskový – ptá, zda je klientův účet v takovém stavu, který dovolí přijmout pohledávku za tisk, především zda má klient použitelný účet a na něm dostatečný zůstatek. Náš systém dotaz vždy vyhodnotí a předá externímu systému požadovanou odpověď nebo oznámení o vzniklé chybě (opět typicky například dotaz na neexistující účet).

#### **4.3.7 Aplikace – součásti systému – pro uživatele**

Systém je samozřejmě třeba nějakým způsobem ovládat. Protože bude integrován do intranetového systému Inet, budou zde vytvořeny aplikace pro jednotlivé uživatele. Každá aplikace bude reprezentovat součást systému (například výpis z účtu, sestava přijatých pohledávek, apod.) a bude mít nadefinována přístupová práva – nebo-li v jaké roli musí být uživatel aby mohl k aplikaci přistoupit či provést určitou akci.

Rozdělme si proto uživatele na čtyři skupiny. Pokud se v budoucnosti ukáže potřeba zavést další skupinu uživatelů (neboli roli), nebude to díky implementovaným mechanismům žádný problém.

Do první skupiny budou patřit běžní uživatelé – tedy takoví, kteří nejsou zařazeni v žádné roli. Druhou skupinu budou tvořit ekonomové jednotlivých externích systémů, třetí účetní z ekonomického odboru rektorátu a čtvrtou samotná správa systému (především jeho tvůrci). Toto rozdělení bude do jisté míry hierarchické – uživatel bude mít přístup k množině aplikací zahrnující aplikace dostupné předchozím skupinám a aplikace dostupné pro tuto úroveň privilegovanosti.

Podívejme se nyní na aplikace dostupné všem klientům. Sem zaprvé patří aplikace zobrazující informace – o provozu systému, odpovědi na často kladené otázky, harmonogram provádění měsíčně se opakujících činností, seznam plateb, které se nepodařilo automaticky přiřadit k žádnému účtu (především z důvodu chybného variabilního symbolu) a mapa umístění jednotlivých míst, kde se dá platit přes náš systém nebo kde si může klient vložit peníze či vyzvednout hotovostní zůstatek na či ze svého účtu. Obsah těchto aplikací je stejný pro každého klienta.

Druhou skupinu tvoří již personalizované aplikace, tedy takové, které každému klientovi zobrazí obsah závislý na stavu jeho účtu. První z nich – a také odrazový můstek k dalšímu používání celého systému – je elektronický souhlas s podmínkami provozování systému. Dále bude mít klient přístup k výpisu ze svého účtu, k nastavování jeho parametrů (například zadávání inkasního účtu) a k zadávání příkazu k vrácení zůstatku. Tyto aplikace poskytují dostatečně silný nástroj k běžnému ovládání systému.

Aplikace pro ekonomy jednotlivých externích systémů jsou již zajímavější a početnější. Zaprvé mají možnost si zobrazit výpis z účtu pro libovolného klienta – pro možnost odpovídat na dotazy klientů a řešit případné nesrovnalosti. Dále mají ekonomové přístup k sestavám přijatých pohledávek a jejich úhrad (ovšem omezených na systém, který tyto ekonomové spravují) a mohou ručně aktivovat účty klientů (zejména zahraničních). Také mohou spravovat bankovní účty klientů – zadávat do systému a rušit klientům účty s povoleným požadavkem na provedení inkasa a účty pro vrácení zůstatku. Z bezpečnostních důvodů je zadávání vraccích účtů klientem omezeno na takové účty, ze kterých již v době ne starší dvou let a ne mladší jednoho měsíce přišly

na klientův účet platby v celkové výši alespoň 500 Kč. Poslední skupinu aplikací tvoří aplikace pro účtování předpisů pohledávek – zprvu pouze zápis čísla účetního dokladu k položkám v našem systému, později již parametrizované plně automatické účtování (vytvoření účetního dokladu systému Magion a jeho provázání s našim systémem).

Hlavní účetní pak mají přístup k sestavám pohledávek, přijatých a vydaných plateb a úhrad za všechny systémy. Kromě toho mají i přístup k plné sadě nástrojů pro účtování – mohou účtovat jak pohledávky, tak přijaté i vydané platby a úhrady. Opět platí že nejprve bude vše účtováno ručně na základě sestav a následně k položkám doplněno číslo účetního dokladu; teprve později bude doprogramováno automatické účtování – jako první je v plánu zprovoznit automatické účtování úhrad a přijatých plateb, neboť se jedná o potenciálně největší množiny položek a vznikajících dokladů. Dále mají hlavní účetní přístup k sestavám, nutným pro podávání daňového přiznání. Jedná se zejména o sestavu daňové rekapitulace a o přístup k Souhrnným daňovým dokladům pro jednotlivé klienty – jak pro kontrolu tak pro jejich vystavování a potvrzování, bude-li jej klient vyžadovat.

Zbývá sada aplikací pro správce. Kromě všech předchozích mají především skupinu aplikací, sloužících k provádění hromadných i jednotlivých operací s účty. Jedná se především o poloautomatické načítání informací o přijatých platbách bankou, generování požadavků na provedení inkasa a jejich odesílání do banky, hromadné zpracování příchozích plateb a uhrazování pohledávek, změny stavů účtů a rozesílání informačních e-mailů klientům. Správci mají také k dispozici sadu nástrojů pro kontrolu konzistence dat a správnosti jejich zaúčtování – zejména (pro rychlou kontrolu) souhlasu částek na jednotlivých účtech v systému a v účetnictví.

Tím máme zhruba popsánu množinu aplikací, která bude v našem systému k dispozici. Některé z nich budou popsány podrobněji v dalším textu – zejména takové, které se podílejí na účtování a vytváření podkladů pro přiznání k DPH.

#### **4.3.8 Generování inkasa a srážek ze mzdy**

Jednou z možností zasílání peněz na klientské účty je automatické generování požadavků na inkaso nebo na převod ze mzdy. Protože se jedná o nástroj, který dokáže uživatelům ušetřit značnou část starostí spojených s udržováním svého účtu, považuji za vhodné se o něm rozepsat podrobněji.

Nejprve se zaměříme na rozdíly. Zatímco inkaso může mít nastavené libovolný klient, převod ze mzdy logicky pouze ten, který má na Masarykově univerzitě pracovní poměr a podepsaný souhlas se srážkou ze mzdy ve prospěch našeho systému. Inkaso může proběhnout v libovolný den v měsíci, zatímco srážka ze mzdy pouze na začátku měsíce, v termínu daném harmonogramem zpracování mezd. Posledním významným rozdílem je to, že inkaso je buď provedeno celé nebo celé odmítnuto, ze mzdy může na klientský účet přibýt libovolná částka až do výše původně požadované. Tím však rozdíly končí a ostatní vlastnosti mají tyto dva procesy společné.

V okamžiku spuštění procesu generování požadavků postupuje systém následovně. Nejprve vybere ze všech klientských účtů takové, pro které má smysl požadavek vystavit. Jedná se o takové, pro které platí, že současná zbývajících suma pohledávek s datem splatnosti dříve než jeden měsíc dopředu nemůže být pokryta aktuálním použitelným zůstatkem na klientském účtu. V takovém případě se vypočte požadovaná částka, kterou je nutné na účet zaslat tak, aby zůstatek přesně postačoval na úhradu daných pohledávek, a v případě převodu ze mzdy se zaokrouhlí na celé koruny nahoru. Požadavek se pak zašle do banky nebo do personálně-mzdového systému a poznačí se, že pro daný účet byl již tento měsíc generován.

V okamžiku příchodu informace o provedení položky – po zpracování mezd nebo po provedení požadavku v bance – je odpovídajícím způsobem navýšen zůstatek na klientově účtu. Pokud by druhá strana požadavek odmítla, je o tom klient upozorněn e-mailovou zprávou.

V systému je navržena ještě jedna funkce, ovlivňující chování tohoto procesu. Každý klient má možnost nastavit si pravidelně udržovaný stálý zůstatek na svém účtu. V okamžiku generování inkas či převodů ze mzdy je pak požadovaná částka navýšena tak, aby po úhradě veškerých relevantních pohledávek na klientově účtu zůstal právě nastavený zůstatek. Tato varianta je vhodná zejména pro hrazení tisků a kopírování nebo pro nákupy v nápojových automatech. Klient se tak může zcela zbavit manipulace s hotovostí. Vhodné je také nastavení pravidelně udržovaného stálého zůstatku na nenulovou výši v situaci, kdy klient hodlá jak tisknout, tak být ubytován na kolejích – díky nezbytné prodlevě mezi vystavením požadavku a jeho provedením se může stát, že klient v tomto časovém intervalu využije služeb tiskového systému, sníží tím svůj použitelný zůstatek a následně se stane dlužníkem v ubytovacím systému.

#### 4.3.9 Provázání systému s účetnictvím

Jak již bylo zmíněno, náš systém představuje ve své podstatě operativní evidenci účetnictví. Podívejme se proto na to, jak bude provázanost s účetnictvím realizována.

V EIS Magion jsou jednotlivé účetní doklady identifikovány jednoznačně čtveřicí údajů: modul, řada, pořadí a rok. Příkladem identifikace je zaúčtování příchozí platby z banky dokladem BAN/9902/1/10, kde se jedná o první doklad dané řady a modulu v roce 2010. Možné moduly jsou BAN pro bankovní operace (a fiktivní – vnitropodnikovou – banku), POH pro účtování pohledávek, POK pro pokladnu a MAN pro ostatní doklady. Z technického hlediska je podstatné, že pro každý modul existuje v systému samostatná tabulka. Kromě svého označení má každý doklad také jednoznačné id pro jednodušší zpracování.

S výjimkou příjmů pokladnou, kde položky v našem systému budou vznikat až na základě existujících pokladních dokladů, bude vždy platit, že nejdříve vznikne položka našeho systému, která je posléze (denně nebo měsíčně) účetním dokladem zanesena do systému Magion.

Pro položky, představující vznik finančního plnění, je situace jednoduchá. V principu pro jejich zaúčtování zcela postačují aplikace, které dokáží zobrazit celkovou částku pro externí systém, službu a pracoviště univerzity, na kterém mají být zaúčtovány (například u pohledávek pro správné účtování výnosů součásti univerzity, která poskytla službu) za určené období a pro kontrolu správnosti i položkové členění (rozpis jednotlivých položek, které dají dohromady výslednou – účtovanou – částku). Na základě těchto údajů je pak možné vytvořit účetní doklad systému Magion.

Pro průběžné položky – tedy pro úhrady – je situace podobná. Stačí znát stejné údaje jako v případě vzniku finančního plnění; navíc je třeba rozlišovat, zda se jedná o úhradu ze zálohy, závazku, či přímou úhradu pohledávky po splatnosti.

Druhý krok při účtování je zápis čísla účetního dokladu k položkám, na jejichž základě vznikl. Protože účetní vyžadují možnost účtovat úhrady ze záloh dvěma doklady (prvním dokladem se zaúčtuje výdej zálohy na úhradu pohledávky nebo kauce, druhým úhrada pohledávky/kauce z vydané zálohy), je nutné, aby čísla dokladů byla zapsána do vazební tabulky. Postup je však podobný – vyberou se položky, ke kterým se doklad vztahuje a správné číslo dokladu se zapíše do databáze a spáruje s položkami. Z důvodů

řízení přístupu bude tato funkčnost v samostatné aplikaci, neboť na sestavy pro účtování může mít v principu přístup více lidí – například správci externích systémů pro kontrolu svých sestav oproti položkám v našem systému.

Pro úplnost rozeberme ještě případ příjmu peněz přes pokladnu. Zde nejprve pokladní vytvoří účetní doklad, jehož číslo rovnou zadá – pomocí zvláštní aplikace – do našeho systému. Systém si pak sám vyhledá doklad v databázi EIS Magion (přístupný přes dohodnuté integrační rozhraní) a na jeho základě vytvoří položku typu „platba přijatá“ a rovnou ji se zadaným dokladem spáruje.

Situaci pro manuální účetnictví máme již popsánu. Ideálním stavem by však vzhledem k množství plánovaných služeb bylo mít veškeré účetnictví automatizované – tedy od vytvoření dokladu EIS Magion až po jeho zápis do našeho systému. Zároveň je to funkce, kterou ekonomický odbor rektorátu požaduje, ačkoliv netrvá na jejím zprovoznění od prvního spuštění systému.

Účetní, který bude spouštět proces automatického účtování, bude postupovat podobně jako při manuálním účtování. Nejprve zvolí druh vytvářeného dokladu, externí systém, období a další náležitosti. Následně zadá požadavek a systém vytvoří podle dohodnutých pravidel potřebný počet účetních dokladů, jejichž čísla zapíše k položkám v našem systému. Je důležité, aby doklady byly vytvářeny ve stejném pořadí, v jakém zobrazují položky sestavy – jediné tak bude možné kontrolovat jejich správnost.

Protože aplikace pro automatické účtování jednotlivých druhů položek – pohledávek, plateb přijatých či úhrad – budou pravděpodobně vznikat postupně, byla jako priorita zvolena aplikace pro účtování úhrad, kde se předpokládá největší objem vznikajících dokladů a tudíž po automatizaci nejvíce ušetřené práce pro lidskou obsluhu.

#### **4.3.10 Podklady pro odvod daní**

Vzhledem k tomu, že náš systém není koncipován jako elektronická peněženka, ale jako operativní evidence účetnictví, je zodpovědný za správný odvod daní.

První z odváděných daní je daň darovací. V okamžiku, kdy student ukončí své působení na univerzitě a zanechá přeplatek na svém klientském účtu, je vyzván k jeho vyzvednutí. V některých případech – zejména pro nízké částky a zahraniční studenty – by náklady na vrácení zůstatku převýšily jeho hodnotu. V takovém případě má student možnost svůj zůstatek darovat univerzitě. Požadovaná sestava je v tomto případě velmi

jednoduchá – darovací daň se odvádí dvakrát ročně, stačí tedy, aby sestava dokázala zobrazit darované částky (včetně dárce) za vybrané pololetí vybraného roku.

Situace s druhou odváděnou daní je složitější. Jedná se o již mnohokrát zmiňovanou daň z přidané hodnoty. Zákon č. 235/2004 Sb., o dani z přidané hodnoty, dává univerzitě za povinnost vystavovat odběratelům daňové doklady, na jejichž podkladě vznikne přiznání pro odvod DPH. Tyto doklady navíc musí být číslovány a v jednom roce tvořit souvislou nepřetržitou řadu. Naštěstí není nezbytně nutné vystavovat doklad na každé jednotlivé plnění, zcela postačuje jeden doklad na všechna plnění v jednom měsíci, vystavený do 15 dnů od konce daného měsíce.

Souhrnný daňový doklad musí obsahovat následující náležitosti: své číslo, zdaňovací a účetní období, datum uskutečnění zdanitelného plnění, datum vystavení, jméno a adresu dodavatele spolu s kontaktní osobou (v našem případě je dodavatel vždy Masarykova univerzita), jméno a adresu odběratele. Dále je třeba aby obsahoval závazky klienta před a po lhůtě splatnosti, disponibilní zálohy, které klient poskytl MU a ostatní závazky MU vůči klientovi – všechny tyto údaje jak na začátku účetního období, kterého se doklad týká, tak i na jeho konci. Konečně musí doklad obsahovat rozpis plnění v daném měsíci podle sazeb DPH – každé plnění musí mít uvedeno datum plnění, jednotkovou cenu, množství, měrnou jednotku a celkovou cenu; jedná-li se o úhradu pohledávky z předchozího období, pak i číslo dokladu, na kterém byla uvedena původní pohledávka.

V případě, že se zdaňovací období rovná účetnímu, je situace jasná. Může však nastat situace, kdy bude zjištěn důvod pro vystavení pohledávky za starší zdaňovací období, které je však již účetně uzavřené a DPH odvedena. V takovém případě je nutné vystavit mimořádný daňový doklad a podat opravné přiznání k DPH za uvedené zdaňovací období.

Z hlaviček souhrnných daňových dokladů pak vzniká evidence pro daňové účely, kde je celková odváděná částka DPH (rozepsaná podle kategorií) rozepsaná na jednotlivé doklady, na jejichž základě byla sečtena. Auditoři pak mají možnost zkontrolovat správnost odevzdaného daňového přiznání kontrolou jednotlivých dokladů a rozpisu plnění.

Zřejmě bude nutné udržovat jak vytvořené hlavičky dokladů v databázi, stejně jako celkové hodnoty za daný měsíc a vazby položek na příslušný daňový doklad. Samotné

doklady a sestavu pro daňovou evidenci pak budou zobrazovat dvě samostatné aplikace – i v tisknutelné podobě.

Ukázka neúplného souhrnného daňového dokladu klienta je na obrázku 4.

#### Zůstatek

|                                     | Předchozí období | Aktuální období |
|-------------------------------------|------------------|-----------------|
| Blokované prostředky                | 1.470,00 Kč      | 1.470,00 Kč     |
| Závazky klienta do lhůty splatnosti | 0,00 Kč          | 0,00 Kč         |
| Závazky klienta po lhůtě splatnosti | 0,00 Kč          | 0,00 Kč         |
| Závazky klienta celkem              | 0,00 Kč          | 0,00 Kč         |
| Disponibilní zálohy                 | 1.198,80 Kč      | 227,80 Kč       |
| Disponibilní závazky MU             | 0,00 Kč          | 0,00 Kč         |
| Disponibilní prostředky celkem      | 1.198,80 Kč      | 227,80 Kč       |

#### Rekapitulace plnění v aktuálním období

| Sazba DPH           | Cena bez DPH    | DPH           | Cena celkem     | ZD z PP        | DPH z PP       | PP celkem      | Rozdíl ZD       | Rozdíl DPH    |
|---------------------|-----------------|---------------|-----------------|----------------|----------------|----------------|-----------------|---------------|
| základní sazba 20%  | 1.535,10        | 307,10        | 1.842,20        | 0,00           | 0,00           | 0,00           | 1.535,10        | 307,10        |
| snížená sazba 10%   | 1.408,20        | 140,80        | 1.549,00        | 0,00           | 0,00           | 0,00           | 1.408,20        | 140,80        |
| základní sazba 19%  | 0,00            | 0,00          | 0,00            | -580,83        | -110,37        | -691,20        | -580,83         | -110,37       |
| snížená sazba 9%    | 0,00            | 0,00          | 0,00            | 0,00           | 0,00           | 0,00           | 0,00            | 0,00          |
| snížená sazba 5%    | 0,00            | 0,00          | 0,00            | 0,00           | 0,00           | 0,00           | 0,00            | 0,00          |
| osvobozeno 0%       | 0,00            | 0,00          | 0,00            | 0,00           | 0,00           | 0,00           | 0,00            | 0,00          |
| není předmětem daně | 0,00            | 0,00          | 0,00            | 0,00           | 0,00           | 0,00           | 0,00            | 0,00          |
| <b>celkem</b>       | <b>2.943,30</b> | <b>447,90</b> | <b>3.391,20</b> | <b>-580,83</b> | <b>-110,37</b> | <b>-691,20</b> | <b>2.362,47</b> | <b>337,53</b> |

Pozn. 1: Uskutečněná plnění osvobozená jsou plnění v tuzemsku s nárokem na odpočet.

Pozn. 2: ZD z PP/DPH z PP je základ daně/daň z přidané hodnoty z přijaté zálohy z předchozího období, použité k úhradě v aktuálním období.

#### Rozpis plnění

| Položka                              | Číslo původního dokladu | Datum plnění<br>Splatnost | Jednotková cena<br>bez DPH | Měrná<br>jednotka | Počet | Cena<br>bez DPH | Sazba | DPH            | Celkem          | DUZP       | Datum dokladu | ID      |
|--------------------------------------|-------------------------|---------------------------|----------------------------|-------------------|-------|-----------------|-------|----------------|-----------------|------------|---------------|---------|
| Počítač / Notebook (ycollege)        |                         | 01.01.2010<br>13.01.2010  | 41,66                      | ks                | 1     | 41,66           | 20    | 8,34           | 50,00           | 01.01.2010 | 2010-01-05    | 1814727 |
| Internet na pokoji (ycollege)        |                         | 01.01.2010<br>13.01.2010  | 108,33                     | ks                | 1     | 108,33          | 20    | 21,67          | 130,00          | 01.01.2010 | 2010-01-05    | 1814728 |
| bezhotovostní vklad, převodní příkaz |                         | 14.01.2010                |                            |                   |       | 1.385,11        | 20    | 277,09         | 1.662,20        | 14.01.2010 | 2010-01-15    | 577240  |
| <b>celkem za 20 %</b>                |                         |                           |                            |                   |       | <b>1.535,10</b> |       | <b>307,10</b>  | <b>1.842,20</b> |            |               |         |
| Ubytování 31.12.2009 - 31.1.2010     |                         | 05.01.2010<br>13.01.2010  | 1.408,20                   | ks                | 1     | 1.408,20        | 10    | 140,80         | 1.549,00        | 05.01.2010 | 2010-01-05    | 1814726 |
| <b>celkem za 10 %</b>                |                         |                           |                            |                   |       | <b>1.408,20</b> |       | <b>140,80</b>  | <b>1.549,00</b> |            |               |         |
| Úhrada pohledávky ze ZAL             | SUPO/B/65089/2009       | 13.01.2010                |                            |                   |       | -580,83         | 19    | -110,37        | -691,20         | 13.01.2010 | 2010-01-13    | 1441760 |
| <b>celkem za 19 %</b>                |                         |                           |                            |                   |       | <b>-580,83</b>  |       | <b>-110,37</b> | <b>-691,20</b>  |            |               |         |
| <b>celkem plnění vč. dobropisů</b>   |                         |                           |                            |                   |       | <b>2.362,47</b> |       | <b>337,53</b>  | <b>2.700,00</b> |            |               |         |

Obrázek 4: Ukázka souhrnného daňového dokladu klienta

#### **4.3.11 Správa klientských dluhů**

Není možné předpokládat, že veškeré pohledávky budou uhrazeny k datu své splatnosti. Proto je třeba vytvořit i jistý subsystém, který se bude starat o správu dlužníků. Jaké jsou na něj kladeny požadavky?

Zaprvé není vhodné trestat klienty ihned v okamžiku vzniku dluhu. Pozdní úhrada částky může být způsobena například technickou chybou či drobným opomenutím. Je však vhodné v okamžiku vzniku dluhu (případně ten samý den) klienta na tuto skutečnost upozornit. Informaci o vzniku dluhu je také možné uložit do databáze pro další zpracování – zejména pro sledování klientovy platební morálky. Je také požadována podpora oficiálního upomínání klienta o zaplacení dluhu. Zaslání písemné upomínky totiž musí mít určité náležitosti; po odeslání třetí (v určitých intervalech) je možné začít pohledávky po klientovi vymáhat soudní cestou. Požadována je však jen poloautomatická činnost v oblasti oficiálního upomínání – systém má umět připravit oficiální dopis spolu s dlužníkovou adresou a uchovávat informace o jeho odeslání – samotné odesílání však bude probíhat pouze na vyžádání zodpovědného pracovníka.

Pro zodpovědné pracovníky jsou požadovány následující výstupy: seznam aktuálních dlužníků (řazený podle různých kritérií) a pro každého dlužníka stejná informace spolu s informacemi o odeslaných upozorněních a upomínkách a platební morálkou (průměrná výše a délka dluhu).

V této oblasti se jako vhodný nástroj ukazuje použití data miningových aplikací. Vytvořením vhodného modelu nad daty o úhradách pohledávek je možné získat požadované informace, které budou následně zpracovány a vyhodnoceny. Klienti pak mohou být rozřazeni do kategorií – podle požadavků vedení univerzity.

Tato součást systému je bohužel zatím pouze ve stadiu analýzy – ekonomický odbor rektorátu Masarykovy univerzity zatím stále neví, jakým způsobem s dlužníky nakládat – a bez definovaného cíle není možné začít sestavovat model pro data mining.

#### **4.3.12 Bezpečnost systému**

Protože náš systém bude zacházet jak s penězi, tak s citlivými údaji o jednotlivých klientech, je třeba věnovat nemalou pozornost bezpečnosti systému. Začneme proto tak, že identifikujeme jednotlivá možná místa útoku na systém.

Prvním možným útokem je zcizení přihlašovacích údajů uživatele systému. Každý uživatel, jak již bylo zmíněno, má své přístupové jméno (přezdívku v systému nebo UČO – univerzitní číslo osoby) a heslo, kterým se do systému hlásí. Případný útočník se může pokusit buď o slovníkový útok – UČO jednotlivých osob je veřejně dostupné, pokusit se odpozorovat heslo při jeho zadávání, odposlechnout jej při síťové komunikaci nebo v nestřeženém okamžiku provést škodlivou akci u neukončeného sezení u Inetu. Závažnost takového útoku závisí na tom, kterému uživateli bude přístup do systému odcizen – v případě běžného klienta může útočník napáchat mnohem menší škody než v případě správce systému.

Odpozorování hesla a slovníkový útok jsou akce, proti kterým se musí bránit především samotní uživatelé. Platí, že heslo by mělo být dostatečně bezpečné ( alespoň 8 znaků dlouhé a obsahovat malá i velká písmena, číslice a jiné než alfanumerické znaky) a zároveň by mělo být alespoň jednou ročně měněno. Uživatel by také měl dávat pozor, zda jej při zadávání hesla někdo nesleduje, tisknuté klávesy zakrývat rukou a nedovolit prohlížeči, zejména na veřejném počítači, zapamatovávat si přístupové údaje v rámci usnadnění činnosti. Poslední zmíněná činnost je v některých prohlížečích omezená nastavením autocomplete=“off“ u formulářových prvků pro zadání uživatelského jména a hesla, jiné prohlížeče však tuto volbu ignorují. Ostatní činnosti jsou v kompetenci samotného uživatele, který by si měl uvědomit, že se jedná buď o jeho peníze, nebo o peníze klientů, za které je zodpovědný.

Sezení u Inetu mají nastavitelnou dobu nečinnosti, po které budou automaticky ukončena – je v zájmu uživatele, aby se pečlivě odhlašoval a v případě sdílení počítače s dalšími lidmi měl nastavené rychlé odhlášení v případě nečinnosti.

Odposlechnutí síťové komunikace může mít největší naději na úspěch, je však nejhůře technicky realizovatelné. Jedna varianta odposlechu hesla předpokládá útok na DNS servery uživatele a podvrhnutí IP adresy jiného serveru místo serveru univerzity. Nic netušící uživatel pak zadá své přístupové údaje do cizího systému, který je následně předá útočníkovi. Proto je důležité, aby uživatelé dávali pozor na výzvy prohlížeče o změně bezpečnostního certifikátu serveru, na který přistupují. Druhou možností je přímé poslouchání komunikace na uživatelské lince. Tomu je zabráněno použitím protokolu https pro komunikaci, konkrétně pro vytvoření bezpečného kanálu nad nezabezpečenou sítí. V případě použití aktuální verze SSL (a tedy některé z posledních

verzí prohlížečů) je takováto komunikace považována za bezpečnou proti odposlouchávání. V případě použití jednoduchého SSL je komunikace však napadnutelná útokem typu „man-in-the-middle“ [15]. Realizace takového útoku je však natolik technicky a logisticky složitá, že její pravděpodobnost prohlásíme za dostatečně malou a dovolíme si toto riziko vědomě podstoupit a řešit až případné následky. Vhodnou metodou by také mohlo být pojištění proti škodám způsobeným takovýmto útokem, je však otázka, zda existuje pojišťovna ochotná dané riziko pojistit. Druhým možným – a také již zmíněným – útokem je odcizení identifikační karty uživatele. Zranitelnost systému je dvojí – buď je možné fyzicky odcizit uživatelskou identifikační kartu, nebo je možné ji zkopírovat – zařízení pro bezdotykovou kopii RFID čipů používaných v kartách vydávaných univerzitou jsou dostupná již za cenu několika set dolarů. Riziko tohoto útoku může být vysoké – nemusí se jednat přímo o krádež, ale student může svoji kartu ztratit nebo někde zapomenout. Případný nálezce pak může být sveden k vyzkoušení, zda je možné ji použít na nákup v automatu či pro kopírování – nebo-li k činnostem, které nevyžadují žádnou autentizaci do systému.

Samotným ztrátám karet se zabránit pravděpodobně nedá, je však možné minimalizovat jejich dopady. Především každý klient bude mít přednastavený limit na transakce za použití karty, jehož výši si bude moci upravovat a zároveň si bude moci zadat, zda má být limit brán jako týdenní či denní. V okamžiku zjištění ztráty karty si klient může nastavit limit na nulu, čím efektivně zabráni libovolným transakcím. Zodpovědnost je tak přenesena na klienta.

V případě zkopírování ID karty je řešení diskutabilní. Pravděpodobně však bude vhodnější tento útok řešit společně s jeho závažnějšími dopady – zatímco při zkopírování studentské karty může útočník odcizit několik set korun ze studentova účtu, při použití karty jako prostředku přístupu do budov nebo laboratoří by útočník mohl napáchat škodu o několik řádů vyšší.

Konečně třetí druh útoků může mít potenciálně nejzávažnější dopady. Podvrhnutím komunikace – neboli vydáváním se za externí systém, který má služby hrazeny přes svůj systém – lze získat informace o stavu klientského účtu (pro pozdější zneužití při jiném druhu útoku), vystavit klientovi pohledávku a tím ho připravit o peníze, nebo nechat připsat libovolnou částku na libovolný účet. Útočník tak může způsobit škodu

jak klientovi, tak univerzitě, která je zodpovědná za správné účtování finančních pohybů.

Naštěstí tomuto druhu útoku je poměrně snadné předejít – nebo je možné jej alespoň velmi omezit – a zároveň je lehké jej odhalit, urychlit řešení následků a minimalizovat tak jeho celkové dopady. Ačkoliv komunikace přes zmíněný XML protokol probíhá jako zasílání XML požadavků na definovanou IP adresu a tedy v principu snadno podvrhnutelná, je lehké ji zabezpečit. Za prvé náš systém bude přijímat požadavky pouze z povolených dvojic (IP adresa, externí systém). Za druhé bude komunikace chráněna heslem – konkrétně před jejím započítím se bude požadovat, aby systémy provedly tzv. SSL handshake [13].

Pokud by se i přes tato bezpečnostní opatření podařilo útočnickovi podvrhnout systému svá data, nastoupí kontrolní procesy. Zodpovědný pracovník externího systému by měl nejméně jednou týdně zkontrolovat, zda souhlasí částky položek v jeho systému s částkami položek v systému našem. Pokud ne, měl by iniciovat křížovou kontrolu – tato kontrola by měla být implementována v systému a být automaticky spouštěna jednou měsíčně. Systémy, které vyžadují každodenní účtování budou samozřejmě kontrolovány častěji.

Nejrizikovější by tak bylo podvrhnutí komunikace s automatickým bankovníkem, kterým je možné „nabíjet“ svůj účet penězi. Proto bude kontrolován obzvlášť pečlivě. Každému výběru peněz z bankovníku budou muset být přítomni alespoň dva zaměstnanci univerzity, kteří spočítají vloženou hotovost a porovnají ji s dostupnými sestavami. Každý pohyb u bankovníku bude zároveň nahráván bezpečnostní kamerou s vysokým rozlišením tak, aby bylo vidět kdo vkládal jakou částku. V případě zjištěných nesrovnalostí tak bude konzultován záznam a dohledána příčina problému – předpokládá se však, že častěji než útok bude příčinou selhání techniky – například špatné rozpoznání bankovky. Stejným záznam bude použit v případě reklamací studentů při vložení hotovosti a jejím špatném připsání na účet.

#### **4.3.13 Zodpovědnost za procesy v systému**

V případě, že navrhujeme takto složitý systém, je třeba mít přesně definovanou zodpovědnost za procesy v tomto systému probíhající. Tyto procesy si můžeme rozdělit do několika kategorií, každé pak přiřadíme skupinu zaměstnanců, která bude za danou

kategorii zodpovědná. Každou takovou skupinu by měli tvořit alespoň dva zaměstnanci tak, aby byli v případě nemoci či dovolené navzájem zástupní.

Přirozené vytváření skupin zaměstnanců se odvíjí od dříve zmíněných uživatelských rolí – je zřejmé, že zaměstnanec musí být zodpovědný za to, co dělá; naopak nemůže být zodpovědný za proces, který nemůže žádným způsobem ovlivnit.

Správci systému budou zodpovědní za technické řešení – budou muset být schopni reagovat na případné výpadky a zajistit hladký každodenní běh systému. Kromě toho budou komunikovat s klienty a odpovídat na jejich dotazy, případně dotazy předávat osobám, které mají dostatečné znalosti k jejich zodpovězení. Lepší název skupiny než „správci systému“ by tak mohl být „centrum podpory“.

Ekonomové jednotlivých externích systémů budou zodpovědní za správnost položek, předávaných do centrálního systému a jejich následné zaúčtování.

Hlavní účetní univerzity budou zodpovědní za správné zaúčtování úhrad a přijatých plateb. Dále budou vydávat metodické pokyny ohledně způsobu účtování jednotlivých položek či dokladů. Budou také úzce spolupracovat s auditory a právníky – budou zodpovídat za to, že systém vyhovuje platné legislativě ČR.

#### **4.4 Porovnání s jinými systémy**

Ačkoliv je náš systém v určitém ohledu již ve svém zárodku v prostředí českých vysokých škol unikátní, jsou vyvíjeny či existují i další podobné systémy. Pro jejich srovnání s našim systémem jsem se zúčastnil konference EUNIS, konané v květnu 2009 ve Špindlerově mlýně. Ačkoliv téma konference se zabývalo především čipovými kartami, bylo zde představeno i několik systémů elektronické peněženky.

Celkově zde byla prezentována tři další řešení. Prvním z nich je systém elektronické peněženky na České Zemědělské Univerzitě v Praze [3]. Systém zde využívá informaci o aktuálním zůstatku uloženou na RFID čipu identifikační karty. Tento systém má své výhody – například není nezbytně nutné, aby veškeré platební terminály stále komunikovaly s centrálním serverem, stejně jako nevýhody – především z hlediska bezpečnosti při ztrátě karty, tak i z hlediska nutnosti mít sjednaný vyšší stupeň bankovní licence ČNB. V době konání konference však existovala pouze koncepce systému. Systém by však měl umožnit podobný rozsah hrazených služeb jako náš systém.

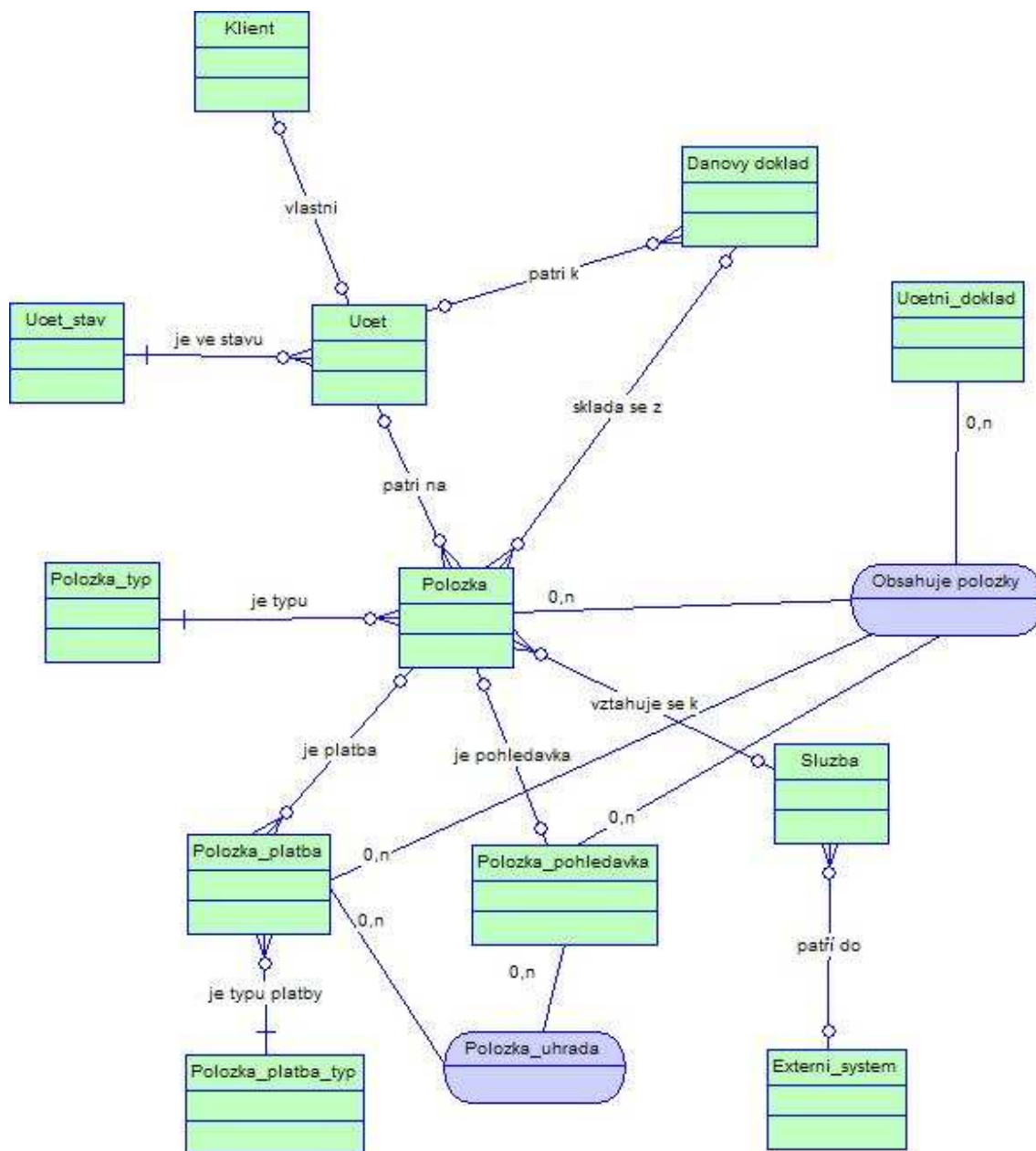
Podobný systém, nazývaný Transakční a zúčtovací systém, vyvíjí i ČVUT [4]. Na konferenci byl bohužel zmíněn pouze v krátkosti. Celková koncepce systému je již podobnější tomu našemu – čipová karta je používána jen jako prostředek pro identifikaci klienta v některých aplikacích – ale opět systém pracuje spíše jako elektronická peněženka, do něj z jedné strany přitékají peníze od klientů a z druhé strany jsou vydávány platby na úhradu služeb. Poznámka o bankovní licenci ČNB platí proto i zde.

Posledním zmíněným systémem byl systém provozovaný na VUT v Brně [6]. Tato univerzita se rozhodla jít cestou nasazení systému podobnému tomu na ČVUT, avšak dodávanému externí firmou. Systém pracuje s jednotným univerzitním kontem, jeho správa je však zařazena pod Správu kolejí a menz VUT. Také neoperuje s odváděním DPH z přijatých plateb – chová se tedy opět jako elektronická peněženka. K jeho nevýhodám patří nemožnost zapojit do systému externí subjekt.

Podrobnější popisy – především posledního systému – lze nalézt ve sborníku konference EUNIS 2009, konkrétně v příspěvcích autorů z České zemědělské univerzity [3], Českého vysokého učení technického [4] a Vysokého učení technického v Brně [6].

## 4.5 Vznik systému

Na základě uvedených specifikací byl následně vytvořen konceptuální datový model (jeho podrobný popis viz příloha 1), který byl poté převeden do fyzického datového modelu. Ten však v této práci neuvádíme, neboť není natolik podstatný a v závislosti na použité databázi se jednotlivé fyzické datové modely mohou od sebe mírně odlišovat.



**Obrázek 5: Konceptuální datový model systému**

Po vytvoření databázového schématu byl systém implementován, stejně jako jeho předchůdci, v jazyce Java.

Jednotlivé třídy byly rozděleny do třech základních skupin: třídy jádra, reprezentující jednotlivé entity z konceptuálního datového modelu, mající v sobě metody pro své vytváření, rušení a manipulaci. Druhou skupinu tvoří třídy, mající na starost opakované operace s entitami – například při příchodu peněz z banky je třeba provést celou řadu operací, navíc v jedné transakci, tedy tak, aby při případném vzniku chyby zůstala databáze v konzistentním stavu. Podobná situace platí při uhrazování položek, zpracování libovolné příchozí platby nebo generování inkasa. Rozdělení do více tříd je také vhodné z důvodu snadnější spolupráce více vývojářů – i v situaci, kdy je intenzivně využíván verzovací systém jako je CVS nebo SubVersion.

Třetí skupinu tříd tvoří jednotlivé, dříve zmíněné, aplikace. Každá z nich je tvořena třídou, která se stará o vytvoření jednotlivých prvků stránky a reakce na události. Použitý framework se stará o výstup do XML, který je následně za použití XSL šablony, kterou má každá aplikace svoji, transformován do (X)HTML. Výhodou tohoto přístupu je skutečnost, že každý vývojář se může starat jen o specifika dané aplikace a společné prvky jsou generovány automaticky.

Ukázky kódu třídy jádra a aplikace spolu s přiloženou transformační šablonou se nachází v přílohách 2, 3 a 4.

Celkově se jedná zhruba o 140 – 150 různých tříd o souhrnné velikosti 2,5 MB a dalších 100 XSL šablon o souhrnné velikosti 1,5 MB. Při běžně uvažovaném počtu znaků na řádek – tedy osmdesáti – se tak jedná o 50 000 řádků kódu. Opět se potvrzuje, že systém pro správu pohledávek, tak jak byl požadován a navrhnout, není triviální záležitost a vyžaduje velké úsilí, má-li být projekt jeho nasazení úspěšný.

## 5 Závěr

### 5.1 Přínos řešení

Jako poslední krok nám zbývá pozastavit se nad celkovým přínosem řešení. Na přínos systému se můžeme podívat z několika rozličných hledisek. Pozastavíme se především nad dvěma hledisky. Jedná se o hledisko zaměstnanců univerzity jakožto správců systému a o hledisko studentů jakožto největší skupiny odběratelů služeb poskytovaných univerzitou a hrazených naším systémem.

Jaké jsou výhody navrhnutého řešení z pohledu studentů? Zaprvé – a tento požadavek zazníval v minulosti od jednotlivých studentů opakovaně – budou veškeré jednotlivé účty v jednotlivých tiskových systémech a v ubytovacím systému spojeny do jednoho. Studenti tak budou mít mnohem lepší kontrolu nad svými výdaji – stavy účtů v jednotlivých tiskových systémech byly například nekontrolovatelné přes internet – a v případě potřeby prvního využití služeb některého z externích systémů je tak budou moci využít ihned bez zřizování nových účtů a vkládání peněz. Budou také moci svůj účet dobíjet na mnoha různých místech po celé univerzitě – například v případě potřeby uhradit dluh za odebrané služby v hotovosti po zavíracích hodinách univerzitních pokladen.

Další výhodou je přesné určení zodpovědnosti za procesy v systému. Studenti se tak mohou se žádostmi o pomoc obracet na centrum podpory, které jim vždy odpoví a zajistí řešení jejich problému. Nebudou tak muset shánět kompetentní osobu pro každý systém zvlášť – centrum podpory dotaz zodpoví samo nebo jej předá kompetentní osobě.

Z pohledu univerzity jsou přínosy také znatelné na první pohled. Asi nejdůležitější vlastností nového systému je jeho vyhovění požadavkům legislativy České republiky. V opačném případě se totiž univerzita vystavuje riziku nejrůznějších postihů za neplnění zákonem daných podmínek. Především při nedostatečných podkladech k odvodům daně z přidané hodnoty jsou sankce přísné – a nemluvíme jen o finančních sankcích, ale o problémech, které by univerzitě mohl způsobit hloubkový audit – i kdyby jen z časového hlediska.

Další velkou výhodou je, podobně jako z pohledu studentů, koncentrace veškerých plateb do jednoho systému. Znamená to především nižší náklady na jejich údržbu, které budou navíc hrazeny z centralizovaných prostředků univerzity. Pro fakulty to znamená snížení finanční zátěže, představované udržováním tiskových systémů ve vlastní režii. Obsluha jednotlivých systémů byla odhadnuta na půl úvazku na systém. Protože takovýchto tiskových systémů bylo na univerzitě provozováno 7, jedná se o možné snížení počtu pracovních míst o 3,5, což představuje měsíční úsporu mzdových prostředků ve výši zhruba 100 000 Kč.

Jednotliví pracovníci také nemusí být zaškolováni do specifik jednotlivých systémů, univerzita tak ušetří náklady na pořádání školení. To samé platí pro účetní – díky jednotné účetní metodice a automatice účtování se snižuje zátěž na účetní jednotlivých fakult a přenáší na centralizované pracoviště. O úspoře byt' jediného pracovního místa zde nemůže být řeč, spíše o zachování současného stavu – což je ovšem vzhledem k rozšiřování aktivit univerzity pozitivní výsledek. Jinými slovy – účtování zvyšujícího se počtu služeb, které univerzita nabízí svým studentům a zaměstnancům, bude bez exponenciálního nárůstu poskytovaného počtu bez problému zvládnutelné současnými silami. Jednotná účetní metodika, vytvářená ve spolupráci s externí auditorskou firmou, pak zaručuje správnost a jednotnost účetních postupů a rychlejší propagování změn v legislativě do našeho systému, díky němuž bude více času na případné technické změny – za všechny jmenujme časté změny v zákoně o DPH.

Hodnotíme-li ovšem přínosy, musíme zároveň zhodnotit i náklady na zavedení systému na Masarykovu univerzitu. Vyjádřit přesnou částku není jednoduché, spokojíme se proto s odhadem založeným na dostupných datech. Návrh a realizace samotného systému zabrala celou pracovní dobu třem zaměstnancům a část pracovní doby dalších dvou po dobu necelých šesti měsíců. Náklady na vývoj se tak dají odhadnout na zhruba milion korun. Do pravidelných, každoměsíčních, nákladů nebudeme počítat platy zaměstnanců, kteří by se starali o účetnictví či o jednotlivé tiskové systémy před zavedením nového celouniverzitního platebního systému. Veškerou agendu, spojenou s udržováním systému v chodu a komunikací s klienty zvládá několik zaměstnanců po část své pracovní doby – celkově se jedná o zhruba 0,5 úvazku, v peněžním vyjádření může jít o zhruba 20 000 Kč měsíčně.

Celková úspora nákladů tak činí zhruba 80 000 Kč měsíčně, což znamená, že počáteční investice ve výši jednoho milionu korun se zaplatí asi za 12 – 13 měsíců, nebo-li po roce provozu systému. Dají se ovšem očekávat další investice spojené s rozšiřováním funkčnosti systému a se vzrůstajícím počtem služeb – například připojení systému stravování v menzách si jistě vyžádá další analýzy a úpravy dle požadavků Správy kolejí a menz na jedné straně a účetních metodiků na straně druhé.

## 5.2 Výhledy do budoucna

Zbývá se zamyslet nad tím, co náš systém očekává v budoucnosti. Z organizačních důvodů nebyly na jednotné platební konto připojeny všechny možné systémy zároveň. Z tiskových systémů tak zbývá dořešit přechod právnické, přírodovědecké a ekonomicko-správní fakulty na centrální platební systém. Podobně úhrady soukromého telefonního hovorného zaměstnanců probíhají jen na fakultě sociálních studií a na ústavu výpočetní techniky – ostatní pracoviště zatím i přes tlak rektorátu Masarykovy univerzity váhají. Konečně se také připravuje připojení systému Aleph z knihoven MU pro placení pokut za pozdně vrácené knihy a dalších poplatků.

Dále se zvažuje připojení systému stravování v menzách, které bylo prozatím odloženo. Je totiž třeba nastavit podmínky citlivě – v další podkapitole, kde si popíšeme některé zkušenosti z provozu totiž bude uvedeno, že někteří studenti mají problém s chápáním funkce systému, provázanosti jednotlivých systémů a prioritizace úhrad. To je zanedbatelný problém v případě tisků či plateb za ubytování na kolejích, v případě stravování se však problém značně zvyšuje. Na jednu stranu mají hladoví studenti tendence se bouřit a hlasitě protestovat, ač jim oběd byl odepřen jejich vinou, na straně druhé je nutné aby Správa kolejí a menz byla schopná zaplatit za potraviny k přípravě obědů. V jednání je v současné době nastavení jistého způsobu uvěřování klientů, jeho podmínky však zatím zůstávají ve velmi hrubých obrysech.

Je také potřeba dořešit některé problémy s Českou národní bankou. Ačkoliv se při analýze, návrhu a implementaci systému zdálo, že získání základní bankovní licence by neměl být problém, v posledních dnech se ukázalo, že tomu tak není. ČNB proti myšlence plateb externím subjektům vystoupila poměrně ostře a vyžádala si doplnění informací a zároveň pozastavení jakéhokoli již učiněného jednání s externími firmami.

Systém tudíž, ač připraven na různé způsoby plateb, bude moci být až do odvolání provozován jen v rámci plateb za služby poskytované univerzitou.

Konečně zkušenosti z provozu ukazují, jak již bylo také zmíněno, že systém, ač bezchybně fungující, je z pohledu studentů velmi složitý. Bohužel zjednodušit systém jako takový nemůžeme, ale je možné mu udělit tzv. „lidskou tvář“, tedy vytvořit aplikace, které studentovi dají přehlednější údaje o jeho účtu, případně ho krok za krokem provedou nastavením parametrů, seznámí ho s jejich významem a upozorní ho na důležité termíny, které musí dodržovat.

### **5.3 Zkušenosti z provozu**

Navrhnutý systém byl po nějakou dobu již provozován, je proto na čase se podívat na zkušenosti z provozu, které byly za danou dobu nashromážděny.

První zjištění je jednoznačně pozitivní. Systém pracuje zcela bezchybně, je schopen zpracovávat velké objemy plateb i klientů. V současné době, v druhém čtvrtletí roku 2010, má téměř 15 000 aktivních klientů a měsíčně zpracovává asi 65 000 položek s celkovou částkou 24 milionů Kč. Výskyt chyb by se za celý rok 2010 dal spočítat na prstech jedné ruky a vždy se jednalo buď o chybu obsluhy (například o zadání špatného data či služby k položce) nebo o chybu hardware (například chybné rozeznání vkládané bankovky do bankovníku z důvodu špatné kalibrace jednotky). Díky transparentnosti systému a jasnému určení odpovědností jsou však tyto incidenty zjištěny téměř okamžitě a následně i ihned řešeny – většinou ručním zásahem do databáze.

Druhé zjištění se týká snadnosti připojování dalších systémů. Po počátečních, především koncepčních, nejasnostech při připojování prvního systému (zmiňme snad jen nejistotu jak řešit kladné i záporné zůstatky na účtech aktivních i již dávno neaktivních osob v samostatném tiskovém systému a tragicky vedené účetnictví, které nepracovalo se zálohami, ale přijaté peníze od studentů účtovalo rovnou jako výnosy) se další systémy daří připojovat bez problémů. Čas na organizaci připojení systému zabere zhruba dva týdny jednání a dva dny technických příprav včetně konsolidace stávajícího účetnictví.

Dalším zjištěním je přístup studentů k systému. Valná většina z nich si systém chválí a ozývají se i dotazy, kdy budou všechny služby na univerzitě hrazeny z jednotného

účetního konta. Pro poctivost je však nutné uvést, že se ozývají i opačné názory – sice v řádově menším počtu, ale i tak je není možné zanedbat. Z devadesáti procent se však jedná o nespokojenost po vlastní chybě studenta či nepochopení funkce systému. Tím se dostáváme k jediné zřetelně negativní vlastnosti systému a tou je jeho složitost. Protože systém podporuje mnoho funkcí a zároveň se jedná o peníze a o účetnictví, je pochopitelné, že některým studentům připadá složitý. Zejména se jedná o studenty prvních ročníků, kteří na začátku svého působení na univerzitě musí zvládnout pochopit několik různých systémů zároveň. Z tohoto důvodu se plánuje se portál pro snazší přístupnost systému, který pomůže jak začátečníkům, tak pokročilým uživatelům, kteří se setkají s problémem, jehož řešení neznají. Důležitou se zdá být i propagace systému tak, aby se změnil pohled některých studentů na něj z „věci, co musíme používat“ na „věc, kterou chceme používat, protože je to výhodné“.

## **5.4 Závěrečné zhodnocení**

Masarykova univerzita se potýkala s potřebou nového informačního systému pro celouniverzitní zúčtování plateb za služby poskytované svým studentům. Tato potřeba se stávala čím dál tím palčivější až nezbylo než jednat. Na základě analýz byl navrhnut informační systém pro zúčtování plateb a pohledávek jednotlivých studentů a zaměstnanců univerzity. Tento systém byl následně implementován a uveden do provozu.

Při návrhu systému bylo nutné uvážit celou řadu omezujících podmínek a požadavků – od vyhovění legislativě České republiky po přání jednotlivých fakult a vedení univerzity (je třeba mít na paměti, že akademické prostředí je odlišné od běžného podniku především ve značné nezávislosti jednotlivých zaměstnanců a nechuti k příkazům shora). Značnou část tvorby systému tak zabralo vyjednávání s nejrozličnějšími zájmovými skupinami a snaha přizpůsobit jim systém tak, aby nebyl odmítnut ještě před svým spuštěním.

Samotné naprogramování systému pak ve srovnání s analýzou bylo značně jednodušší, i tak však systém dosahuje zhruba 50 000 řádků kódu. Fyzický datový model pak obsahuje zhruba 25 tabulek, včetně pomocných pro ukládání historie změn některých entit, číselníků a vazebných tabulek.

Zkušenosti z nasazení systému pak ukazují, že pokud se přechod ze starého systému na nový provede v přesně určené datum, na které budou všichni zúčastnění nachystáni, a systémy budou dostatečně otestovány v cvičném prostředí, je přechod zvládnutelný s minimem obtíží, které jsou řešitelné operativně.

Provoz systému a jeho postupné rozšiřování na celou univerzitu pak odhalil, že mezi studenty i zaměstnanci je o využívání systému zájem. Klienti také vítají čím dál tím širší záběr poskytovaných služeb a u systémů, které jsou dosud hrazeny zvlášť, vznášejí dotazy, kdy budou napojeny na jednotné univerzitní konto. Z druhé strany naopak občas vznikne problém při ne zcela dostatečném pochopení funkčnosti systému klientem, což vede tvůrce systému k záměru vytvořit určitý portál pro jednodušší použitelnost systému i nováčky na univerzitě. Podobně se zrodil i nápad vytvořit informační leták, ve kterém budou přehlednou formou shrnovat základní znalosti nutné k používání systému. Celkově se dá říci, že projekt vytvoření nového systému pro zpracování plateb za služby poskytované univerzitou pomocí jednotného účetního konta byl úspěšný. Systém splňuje veškeré požadavky na něj kladené a zároveň je dostatečně rozšiřitelný i pro budoucí aplikace.

## 6 Seznam použité literatury

### 6.1 Tištěné zdroje

[1] BASL, Josef, BLAŽÍČEK, Roman. Podnikové informační systémy : Podnik v informační společnosti – 2. výrazně přepracované a rozšířené vydání. 2008. vyd. Praha : Grada Publishing, a.s., 2008. 283 s. ISBN 978-80-247-2279-5.

[2] DOSTÁL, Petr, RAIS, Karel, SOJKA, Zdeněk. Pokročilé metody manažerského rozhodování. 1. vyd. Praha : Grada Publishing, a.s., 2005. 168 s. ISBN 80-247-1338-1.

[3] DVOŘÁK, Petr; MACH, Jiří. Implementace systému elektronické peněženky na ČZU v Praze. In *Čipové karty a elektronický podpis na vysokých školách*. Místo : Nakladatelství, 2009. s. 11-22.

[4] HOLÝ, Radek; KALIKA, Marek. ID Karty - technologie, služby a procesy v univerzitním prostředí. In *Čipové karty a elektronický podpis na vysokých školách*. Místo : Nakladatelství, 2009. s. 33-44.

[5] KOCH,M., DOVRTĚL,J., Management informačních systémů, Brno: Vysoké učení technické Brně, Fakulta podnikatelská, 2006. ISBN: 80-214-3262-4

[6] KŘIVÁNEK, Vítězslav; KUREČKA, Radomír. Bezhotovostní platby a jednotné konto. In *Čipové karty a elektronický podpis na vysokých školách*. Místo : Nakladatelství, 2009. s. 22-33.

[7] MOLNÁR, Z. Efektivnost informačních systémů. 1. vyd. Praha: Grada, 2000. 142 s. ISBN 80-7169-410-X.75.

[8] SODOMKA, Petr. Informační systémy v podnikové praxi. 1. vyd. Brno : Computer Press, a.s., 2006. 351 s. ISBN 80-251-1200-4.

[9] STANÍČEK, Zdenko. Datové modelování s použitím metody HIT. In *DATASEM 96*. 1996. vyd. Brno : CS-COMPEX a.s., 1996. s. 35-64. ISBN 80-902250-5-5.

[10] VRANA, Ivan; RICHTA, Karel. *Zásady a postupy zavádění podnikových informačních systémů : Praktická příručka pro podnikové manažery*. První vydání. Praha : Grada Publishing, 2005. 188 s. ISBN 80-247-1103-6.

## 6.2 Elektronické zdroje

[11] BERNARD, Borek. *Zdroják.cz* [online]. 2009-05-07 [cit. 2010-05-09]. Úvod do architektury MVC. Dostupné z WWW: <<http://zdrojak.root.cz/clanky/uvod-do-architektury-mvc/>>.

[12] *Britannica Online Encyclopedia* [online]. 2010 [cit. 2010-05-09]. Information system. Dostupné z WWW: <<http://www.britannica.com/EBchecked/topic/287895/information-system>>.

[13] *IBM - United States* [online]. 2010 [cit. 2010-05-19]. The SSL handshake. Dostupné z WWW: <[http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itame2.doc\\_5.1/ss7aumst18.htm](http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itame2.doc_5.1/ss7aumst18.htm)>.

[14] *Object Management Group - UML* [online]. 2009 [cit. 2010-05-09]. OMG Modeling and Metadata Specifications. Dostupné z WWW: <[http://www.omg.org/technology/documents/modeling\\_spec\\_catalog.htm#UML](http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML)>.

[15] *Toolbox for IT* [online]. 2010 [cit. 2010-05-19]. Man-in-the-Middle Attack. Dostupné z WWW: <[http://it.toolbox.com/wiki/index.php/Man-in-the-Middle\\_Attack](http://it.toolbox.com/wiki/index.php/Man-in-the-Middle_Attack)>.

## Seznam tabulek

|                                                                                    |    |
|------------------------------------------------------------------------------------|----|
| Tabulka 1: Varianty řešení IS. Převzato z [1].....                                 | 17 |
| Tabulka 2: Základní vývojové generace ERP systémů. Převzato z [1].....             | 18 |
| Tabulka 3: Srovnání výdajového a majetkového pohledu na IS/IT. Převzato z [7]..... | 24 |

## Seznam obrázků

|                                                                       |    |
|-----------------------------------------------------------------------|----|
| Obrázek 1: Třívrstvá architektura. Převzato z interval.cz .....       | 32 |
| Obrázek 2: Model-view-controller. Převzato z ucancode.net.....        | 33 |
| Obrázek 3: Jednotné konto jako spojující uzel pro hrazené služby..... | 50 |
| Obrázek 4: Ukázka souhrnného daňového dokladu klienta.....            | 64 |
| Obrázek 5: Konceptuální datový model systému .....                    | 71 |

## Seznam příloh

|                                                                |     |
|----------------------------------------------------------------|-----|
| Příloha 1 - Popis konceptuálního datového modelu systému ..... | 84  |
| Příloha 2 - Ukázka programového kódu jádra systému .....       | 87  |
| Příloha 3 - Ukázka programového kódu aplikace .....            | 94  |
| Příloha 4 - Ukázka kódu XSL šablony .....                      | 103 |
| Příloha 5 - Ukázka aplikace systému .....                      | 105 |

## Příloha 1 - Popis konceptuálního datového modelu systému

**Klient:** entita jednoznačně identifikující osobu na univerzitě pomocí tzv. UČO – univerzitního čísla osoby, které je vazbou do společné centrální databáze osob. Udržuje mimo jiné v sobě aktivitu klienta vůči „jednotnému univerzitnímu kontu“, tedy zda má klient dovoleno či z nějakých důvodů zakázáno mít účet.

**Účet:** entita reprezentující klientský účet, obsahující souhrnné údaje o tomto účtu. Obsahuje variabilní symbol účtu, napočítané zůstatky a zbývající částku k úhradě, stav účtu, datum, od kterého je účet v daném stavu, datum, kdy byl účet naposledy změněn, nastavený pravidelně udržovaný stálý zůstatek a informace o limitech pro transakce pomocí identifikační karty. K této entitě existuje i historická tabulka, ve které jsou uloženy jednotlivé změny stavů účtů.

**Účet\_stav:** číselník stavů, ve kterém se může klientský účet nacházet spolu s podrobnějšími popisy jednotlivých stavů.

**Položka\_typ:** číselník jednotlivých typů položek, zpracovávaných naším systémem, včetně podrobnějších popisů.

**Položka:** entita, reprezentující jednu účetní operaci na klientském účtu, která měla za následek změnu rozvahy. Nese v sobě informace o svém stavu, typu, popisu, datu vystavení, uskutečnění, splatnosti, účetním období, do kterého spadá, jednoznačnou identifikaci v externím systému, daňovém dokladu, na kterém je zahrnuta, kategorii DPH a klientském účtu, ke kterému se vztahuje.

**Položka\_platba:** entita, která svou existencí závisí na existenci entity položka, která je typu platba (přijatá, z dobropisu nebo z kauce). Obsahuje rozšiřující údaje o položce typu platba – základ, daň z přidané hodnoty a zbývající částky, které jsou využitelné pro úhrady pohledávek.

**Položka\_platba\_typ:** číselník typů plateb včetně podrobnějšího popisu.

**Položka\_pohledávka:** entita, která svou existencí závisí na existenci entity položka, která je typu pohledávka, vydaná platba nebo kauce. Nese v sobě doplňující informace o základu a dani z přidaného hodnoty pohledávky, zbývajících částce k úhradě a do aktuálního okamžiku již uhrazených, odepsaných a dobropisovaných částkách pohledávky.

**Položka\_úhrada:** asociační entita, která reprezentuje proběhlou úhradu pohledávky z určité platby. Nese v sobě informace o okamžiku úhrady, částce, základu a dani z přidané hodnoty proběhlé úhrady, o účetním období, do kterého proběhlá úhrada spadá a identifikaci daňového dokladu, na kterém je uvedena.

**Daňový\_doklad:** entita, reprezentující hlavičku souhrnného daňového dokladu klienta. Obsahuje informace o účtu, ke kterému doklad patří, pořadové číslo dokladu, zůstatky klientského účtu na začátku a na konci měsíce, ke kterému doklad patří spolu s určením účetního a zdaňovacího období, které doklad reprezentuje.

**Účetní\_doklad:** entita, která reprezentuje obraz dokladu systému Magion v našem systému. Nese v sobě informace o typu dokladu, jeho jednoznačnou identifikaci, popis v našem systému, stav (volba z možností zaúčtovaný a stornovaný), celkovou souhrnnou částku, datum vystavení dokladu a uskutečnění (zdanitelného) plnění a identifikaci primárního dokladu, pokud se jedná o doklad, který závisí na nějakém jiném.

**Účetní\_doklad\_obsahuje\_položky:** asociační entita mezi účetním dokladem, položkami našeho systému, platbami v našem systému a úhradami v našem systému.

**Externí\_systém:** číselník jednotlivých externích systémů, spolupracujících s naším systémem. Obsahuje identifikaci systému jeho zkratkou, plný název a popis, kontaktní e-mail na správce systému a zodpovědnou osobu za provozování tohoto systému.

**Služba:** číselník služeb, poskytovaných jednotlivými externími systémy. Obsahuje identifikaci služby jak v našem, tak v externím systému, její název a popis, stav a informaci, zda se jedná o službu, která je hrazena identifikační kartou.

**Bankovní\_účet:** entita, reprezentující bankovní účty jednotlivých klientů, obsahující informace o typu účtu (pro provádění inkasa nebo vracení zůstatku) a dále číslo účtu, kód banky, specifický symbol, nastavený inkasní limit a rozsah časového intervalu, ve kterém je bankovní účet v našem systému platný.

## Příloha 2 - Ukázka programového kódu jádra systému

V této příloze následuje ukázka programového kódu zodpovědného vygenerování inkasních příkazů.

```
package jfx.app.supo;

import java.math.BigDecimal;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Timestamp;
import jfx.app.banka.BU;
import jfx.app.supo.banka.BUO;
import jfx.app.supo.banka.Banka;
import jfx.app.supo.kernel.CertificatedConnection;
import jfx.app.supo.kernel.DBConnection;
import jfx.app.supo.kernel.SUPOException;
import jfx.app.supo.kernel.Služba;
import jfx.app.supo.kernel.Ucet;
import jfx.db.SqlUtils;
import jfx.exception.JfxException;
import jfx.security.LogRing;
import jfx.security.OscisUCOKonverze;
import jfx.security.ServerEnv;
import jfx.utils.EMail;
import jfx.utils.SimpleTask;
import jfx.utils.TaskEnvironment;
import jfx.utils.TimeU;
**
*
* @author vymola
*/
public class GenerovaniInkasa implements SimpleTask{

    public void main(TaskEnvironment env) throws Exception {
        env.setVerboseFile("app.supo.generovani-inkasa.log");
        CertificatedConnection
con=DBConnection.certificateConnection(env.getConnection());
        int pocet[]=generujInkasa(con,env);
        env.verbose(1,"zkoumáno SUPO účtů:"+pocet[0]+", odloženo:"+pocet[1]+
```

```

        ", vygenerováno:"+pocet[2]+", bez iBU0="+pocet[4]+", se zakázaným žádáním o
inkaso="+pocet[6]);

```

```

    }

```

```

    /** Generuje požadavky na inkaso. Generuje se těm, kteří tento kalendářní měsíc
    ještě neměly vygenerované
    * žádné inkaso, mají neuhrazené pohledávky (i po uhrazení všeho co lze) a/nebo
    požadují zálohu a mají
    * zadaný inkasní účet. Musí být povolena inkasní služba. Commit se dělá po každém
    SUPO účtu.
    * Inkasovat se dají jen účty, které mají natvrdo povoleno inkaso, případně povoleno
    inkaso pro systém BAN.
    * @return počet vygenerovaných inkas v indexu 2, počet odloženého generování v
    indexu 1, celkový počet zkoumaných účtů v indexu 0.
    * index 3 počet účtů, u kterých byly uhrazeny nějaké pohledávky
    * index 4 počet účtů, ke kterým nebylo nalezeno inkasní BU0
    * index 5 obsahuje počet POH, které byly uhrazeny.
    * index 6 obsahuje počet SUK, které jsou ve stavu se zakázaným generováním inkasa
    **/

```

```

    public static final int[] generujInkasa(CertificatedConnection con, TaskEnvironment
    env) throws SUPOException, SQLException, JfxException{
        int pocet[]=new int[7];
        pocet[0]=0;pocet[1]=0;pocet[2]=0;pocet[3]=0;pocet[4]=0;pocet[5]=0;pocet[6]=0;
        PreparedStatement ps=null;
        PreparedStatement psCekaSeNaInkaso = null;
        ResultSet rs=null;
        ResultSet rsCekaSe = null;
        Ucet.Operation ucOp = new Ucet.Operation();
        ucOp.setTaskEnvironment(env);
        Banka.Operation bankaOp = new Banka.Operation();
        bankaOp.setTaskEnvironment(env);
        BU0.Operation buoOp = new BU0.Operation();
        BU.Operation buOp = new BU.Operation();
        Date aktualniMesic=TimeU.dateUtilToDateSql(new java.util.Date());
        int rok=TimeU.getRok(aktualniMesic);
        int mesic=TimeU.getMesic(aktualniMesic);
        Date prvniMoznyPrichodPenezVMesici=TimeU.datum(10,mesic,rok);
        Date mezSplatnosti = TimeU.prictiMesice(prvniMoznyPrichodPenezVMesici,1);
        int pocetPodezrelychZaznamu;
        Date today = null;
        try {
            ucOp.connect(con);
            bankaOp.connect(con);
            buoOp.connect(con);
            buOp.connect(con);
            Sluzba inkasniSluzba = bankaOp.getInkasniSluzba();

```

```

if (!inkasniSluzba.jeAktivni()) {
    System.out.println("Služba je pozastavena, končím.");
    return pocet;
}

ps=con.prepareStatement(
    "select count(*) " +
    "from supo.UCET " +
    "where uhrada_z + zaloha_inkasa > 0"
    );

rs=ps.executeQuery();
rs.next();
pocetPodezrelychZaznamu=rs.getInt(1);

SqlUtils.sqlCleanup(ps,rs);
ps=con.prepareStatement(
    "select VS " +
    "from supo.UCET " +
    "where uhrada_z + zaloha_inkasa > 0"// and stav='A'
    );
rs=ps.executeQuery();
psCekaSeNaInkaso = con.prepareStatement("SELECT polozka_id FROM supo.polozka
where ucet_id = ? and stav='C' and sluzba_id in (select sluzba_id from supo.sluzba,
supo.ext_system where kod='1' " +
    " and sluzba.ext_system_id = ext_system.ext_system_id and
ext_system.zkratka = 'PaM')");
while (rs.next()) {
    pocet[0]++;
    Ucet ucet = ucOp.najdiDleVS(rs.getString("VS"));
    try {
        ucOp.lock(ucet);
        if (today==null) {today = TimeU.pulnoc(con.getLockDate(ucet));}

        char povolitInkaso = ucOp.povolitInkaso(ucet);
        if (povolitInkaso=='A') {
            BUO buo = buoOp.najdi(ucet.getOscis(),"I");
            if (buo==null) {//osoba nema inkasni ucet
                pocet[4]++;
            }
            else //existuje inkasni ucet osoby
            {
                Timestamp posledniGenerovaniInkasa =
ucOp.getCasVlozeniPosledniPolozky(ucet,inkasniSluzba,"PPR");
                boolean uzByloTentoMesicInkaso;
                if (posledniGenerovaniInkasa==null) {
                    uzByloTentoMesicInkaso = false;
                }
            }
        }
    }
}

```

```

else if (rok == TimeU.getRok(posledniGenerovaniInkasa) &&
mesic == TimeU.getMesic(posledniGenerovaniInkasa))
{
    uzByloTentoMesicInkaso = true;
}
else
{
    uzByloTentoMesicInkaso = false;
}
if (!uzByloTentoMesicInkaso) { //je treba vygenerovat

    BigDecimal vysePotrebnehoInkasa =
ucOp.getInkasniCastka(ucet,mezSplatnosti);

    psCekaSeNaInkaso.setInt(1,ucet.getID());
    rsCekaSe = psCekaSeNaInkaso.executeQuery();
    if (rsCekaSe.next()) {
        env.verbose(0,"CEKA SE NA SRAZKU ZE MZDY "+ucet);
    }
    else
    {

        if (vysePotrebnehoInkasa.signum(>0) {
            if
(vysePotrebnehoInkasa.compareTo(BigDecimal.valueOf(100))<0)
            {
                env.verbose(2,"VYPOCTENA VYSE INKASA
"+vysePotrebnehoInkasa+"KC PRO UCET "+ucet+" JE MENSI NEZ 100KC");
            }
            else {
                Date prvniMoznyDenPrichoduPenez =
ucOp.splatnostPrvniNeuhrazenePohledavky(ucet);
                if (prvniMoznyDenPrichoduPenez == null)
                {
                    prvniMoznyDenPrichoduPenez = today;
                }

                if
(prvniMoznyDenPrichoduPenez.before(prvniMoznyPrichodPenezVMesici)) {

prvniMoznyDenPrichoduPenez=prvniMoznyPrichodPenezVMesici;

                }

                if
((vysePotrebnehoInkasa.compareTo(ucOp.getZalohuInkasa(ucet).subtract(ucOp.getDisponibiln
iZustatek(ucet)))==0)
&&(prvniMoznyDenPrichoduPenez.compareTo(prvniMoznyPrichodPenezVMesici)!=0)) {

```

```

env.verbose(2, "UCET "+ucet+" BY ZADAL JEN O
ZALOHU NA BUDOUCI UHRADY, NEGENERUJI INKASO");
} else {
BigDecimal limit = buoOp.getLimit(buo);

Date generovatInkaso =
TimeU.prictiPracovniDny(buoOp.datumGenerovaniInkasa(buo.getBU(), null, prvniMoznyDenPrichod
uPenez), TimeU.TOK_CASU_VZAD, 2);

//env.verbose(2, "Generovat inkaso:
"+generovatInkaso.toString());

Date dnes=TimeU.pulnoc(con.getLockDate(ucet));
if (!dnes.before(generovatInkaso)) {
    if (limit.signum()>0 &&
limit.compareTo(vysePotrebnehoInkasa)<0)
    {
        if (!ServerEnv.isDevelopmentServer() &&
!env.isRollbackMode()) {
            Email.send("supo-
inet@ics.muni.cz", OscisUCOKonverze.getUCO(ucet.getOscis())+
"@mail.muni.cz", "SUPO - inkaso
překročilo limit",

"Dobrý den, systém SUPO Vám vypočetl
nutnou výši inkasa na "+vysePotrebnehoInkasa.toString()+"Kč, což je více, než Vámi
nastavený limit "

+limit.toString()+"Kč.\nRozdíl ve výši
"+vysePotrebnehoInkasa.subtract(limit).toString()+"Kč doplatte jiným způsobem (viz
https://inet.muni.cz/app/supo/info?zalozky=platby), " +
"nebo Vám hrozí penalizace za opožděné
úhrady.\nS pozdravem správa systému SUPO\n\n"+
"Hello, to pay the outstanding items on
your SUPO account the system would need to collect CZK
"+vysePotrebnehoInkasa.toString()+"", which is more " +
"than the limit (CZK
"+limit.toString()+"") you had entered into the system.\n" +
"You will need to pay the difference of
CZK "+vysePotrebnehoInkasa.subtract(limit).toString()+" using different means (see
https://inet.muni.cz/app/supo/info?zalozky=platby), " +
"or you may be penalized for belated
settlements.\nSincerely the SUPO system administration\n");}
vysePotrebnehoInkasa = limit;
}
env.verbose(2, "ZADAM O INKASO: "+ucet+" "+buo+"
"+vysePotrebnehoInkasa);

bankaOp.inkasujPenize(ucet, buo, vysePotrebnehoInkasa);
env.verbose(4, "ZADAM O INKASO - end");
pocet[2]++;
} else {

```



```

        SqlUtils.rollback(con);
        throw sqle;
    }
    catch (SUP0Exception supoe)
    {
        SqlUtils.rollback(con);
        throw supoe;
    }
    finally
    {
        ucOp.disconnect();
        bankaOp.disconnect();
        buOp.disconnect();
        buoOp.disconnect();
        SqlUtils.sqlCleanup(ps,rs);
        SqlUtils.sqlCleanup(psCekaSeNaInkaso,rsCekaSe);
    }

}

```

Z ukázky byly odstraněny z důvodu rozsahu komentáře k jednotlivým částem kódu.

## Příloha 3 - Ukázka programového kódu aplikace

V této příloze si uvedeme ukázkou programového kódu aplikace pro spuštění procesu automatického účtování přijatých plateb.

```
package jfx.app.supo;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Date;
import java.util.Enumeration;
import java.util.Iterator;
import java.util.List;
import javax.naming.NamingException;
import jfx.app.supo.kernel.DBConnection;
import jfx.app.supo.kernel.DokladTypInfo;
import jfx.app.supo.kernel.SubsystemInfo;
import jfx.app.supo.kernel.TypUctu;
import jfx.app.supo.kernel.UcPripadInfo;
import jfx.db.Roles;
import jfx.db.SqlUtils;
import jfx.security.FSecurity;
import jfx.sys.Lang;
import jfx.sys.batch.BatchSimpleTask;
import jfx.sys.batch.BatchTask;
import jfx.utils.TimeU;
import jfx.wui.App;
import jfx.wui.AppDescription;
import jfx.wui.Persistent;
import jfx.wui.StateType;
import jfx.wui.advance.WBatchQueue;
import jfx.wui.advance.WTabbedPane;
import jfx.wui.ann.Parent;
import jfx.wui.ann.PartialRequest;
import jfx.wui.component.WCheckBox;
import jfx.wui.component.WComboBox;
import jfx.wui.component.WDateField;
import jfx.wui.component.WList;
import jfx.wui.component.WPanel;
import jfx.wui.component.WSubmitButton;
import jfx.wui.component.WTextField;
import jfx.wui.event.OnAction;
import jfx.wui.event.OnSelection;
```

```

import jfx.wui.model.DataModel;
import jfx.wui.model.DataModelFactory;
import jfx.wui.model.DataModelFactory.SimpleListItem;
import jfx.wui.validator.DateValid;
import jfx.wui.validator.LengthValid;
import jfx.wui.validator.MustSelectOneValid;
import jfx.wui.validator.ValidGroup;
import jfx.wui.validator.ValidationAndGroup;

/**
 *
 * @author vymola
 */
@AppDescription(stateType=StateType.SESSION,
xslFile="app/supo/AutomatickeUctovaniPlateb.xsl")
public final class AutomatickeUctovaniPlatebWApp extends App {

    private static final String APP_ID = "ekon.supo.uctovani.automatickeuctovaniplateb";

    @Persistent
    private final WTabbedPane tabbedPane = new WTabbedPane("tabbedPane");

    @Persistent
    @Parent("tabbedPane")
    private final WPanel vyberParametru = new WPanel("vyberParametru");

    @Persistent
    @Parent("tabbedPane")
    private final WPanel vysledky = new WPanel("vysledky");

    @Persistent
    @PartialRequest
    @Parent("vyberParametru")
    @MustSelectOneValid
    @ValidGroup("vyberOK")
    private final WComboBox<TypUctu> typUctu = new WComboBox<TypUctu>("typUctu");

    //ucetni obdobi - rok
    @Persistent
    @PartialRequest
    @Parent("vyberParametru")
    @MustSelectOneValid
    @ValidGroup("vyberOK")
    private final WComboBox<SimpleListItem> roky = new WComboBox<SimpleListItem>("roky");

    //ucetni obdobi - mesic
    @Persistent

```

```

    @PartialRequest
    @Parent("vyberParametru")
    @MustSelectOneValid
    @ValidGroup("vyberOK")
    private final WComboBox<SimpleListItem> mesice = new
WComboBox<SimpleListItem>("mesice");

    @Persistent
    @PartialRequest
    @Parent("vyberParametru")
    @DateValid(shortFormat=false)
    @ValidGroup("vyberOK")
    private final WDateField zdanovaciObdobi = new WDateField("zdanovaciObdobi");

    @Persistent
    @PartialRequest
    @Parent("vyberParametru")
    @ValidGroup("vyberOK")
    @MustSelectOneValid
    private final WComboBox<SubsystemInfo> subSystemy = new
WComboBox<SubsystemInfo>("subSystemy");

    @Persistent
    @PartialRequest
    @Parent("vyberParametru")
    @MustSelectOneValid
    @ValidGroup("vyberOK")
    private final WComboBox<DokladTypInfo> typDokladu = new
WComboBox<DokladTypInfo>("typDokladu");

    @Persistent
    @PartialRequest
    @Parent("vyberParametru")
    @ValidGroup("vyberOK")
    //private final WComboBox<UcPripadInfo> ucetniPripad = new
WComboBox<UcPripadInfo>("ucetniPripad");
    private final WList<UcPripadInfo> ucetniPripad = new
WList<UcPripadInfo>("ucetniPripad");

    @Persistent
    @PartialRequest
    @LengthValid(min=0,max=15)
    @ValidGroup("vyberOK")
    @Parent("vyberParametru")
    private final WTextField textVDokladu = new WTextField("textVDokladu");

    @Persistent
    @Parent("vyberParametru")

```

```

private final WSubmitButton odeslat = new WSubmitButton("odeslat");

@Persistent
@PartialRequest
@Parent("vyberParametru")
private final WCheckBox commit = new WCheckBox("commit", false);

@Persistent
@Parent("vysledky")
private final WBatchQueue davky = new WBatchQueue("davky");

@Persistent
private final ValidationAndGroup vyberOk = new ValidationAndGroup();

private DataModel<DokladTypInfo> naplnTypyDokladu(Connection con) throws
SQLException {
    PreparedStatement psTypyDokladu = null;
    ResultSet rsTypyDokladu = null;
    List<DokladTypInfo> v = new ArrayList<DokladTypInfo>();
    try {
        psTypyDokladu = con.prepareStatement(" select doklad_typ, nazev from
magion_ext.uc_pripad_doklad_typ " +
            " where doklad_typ = 'PPR'");

        rsTypyDokladu = psTypyDokladu.executeQuery();
        while (rsTypyDokladu.next()) {
            v.add(new DokladTypInfo(rsTypyDokladu.getString(1),
rsTypyDokladu.getString(2)));
        }
        return DataModelFactory.create(v, "[id(dokladTyp), request] popis");
    } finally {
        SqlUtils.sqlCleanup(psTypyDokladu, rsTypyDokladu);
    }
}

private DataModel<SubsystemInfo> naplnSubsystemy(Connection con, Date datum, String
ucetTyp) throws SQLException {
    PreparedStatement psSubsystemy = null;
    ResultSet rsSubsystemy = null;
    List<SubsystemInfo> v = new ArrayList();

    try {
        psSubsystemy = con.prepareStatement(" select
subsystem, supextsys.nazev, supextsys.ext_system_id from
magion_ext.UC_PRIPAD, magion_ext.ext_system mextsys, supo.ext_system supextsys" +
            " where uc_pripad.ext_system_id = mextsys.ext_system_id " +
            " and mextsys.zkratka = 'SUPO' " +

```

```

        " and doklad_typ in ('PPR') and klasifikace = ? and "+
        " uc_pripad.platno_od <= ? and uc_pripad.platno_do >= ? " +
        " and subsystem = supextsys.zkratka " +
        " and uc_modul = 'MAN' "+
        " group by subsystem,supextsys.nazev,supextsys.ext_system_id " +
        " order by subsystem,supextsys.nazev");
psSubsystemy.setString(1, ucetTyp);
psSubsystemy.setDate(2, TimeU.dateUtilToDateSql(datum));
psSubsystemy.setDate(3, TimeU.dateUtilToDateSql(datum));
rsSubsystemy = psSubsystemy.executeQuery();
while (rsSubsystemy.next()) {
    v.add(new SubsystemInfo(rsSubsystemy.getString(1),
rsSubsystemy.getString(2),rsSubsystemy.getInt(3)));
}
//list.setData(StructU.toRTreeEnumeration(v));
return DataModelFactory.create(v, "[id(extSystemId), request]
subsystemNazev");
} finally {
    SqlUtils.sqlCleanup(psSubsystemy, rsSubsystemy);
}
}

private DataModel<UcPripadInfo> naplnUcPripady(Connection con, Date datum,
SubsystemInfo subSystem, DokladTypInfo dokladTyp, String ucetTyp) throws SQLException {

    PreparedStatement psUcPripady = null;
    ResultSet rsUcPripady = null;
    List<UcPripadInfo> v = new ArrayList<UcPripadInfo>();
    try {
        psUcPripady = con.prepareStatement("select uc_pripad_id,popis from
magion_ext.uc_pripad where klasifikace = ? " +
        " and doklad_typ = ? and subsystem = ? and uc_modul = 'MAN' "+
        " and platno_od <= ? "+
        " and platno_do >= ? ");

        psUcPripady.setString(1, ucetTyp);
        psUcPripady.setString(2, dokladTyp.getDokladTyp());
        psUcPripady.setString(3,subSystem.getSubsystemZkratka());
        psUcPripady.setDate(4, TimeU.dateUtilToDateSql(datum));
        psUcPripady.setDate(5, TimeU.dateUtilToDateSql(datum));

        rsUcPripady = psUcPripady.executeQuery();

        while (rsUcPripady.next()) {
            v.add(new UcPripadInfo(rsUcPripady.getString("uc_pripad_id"),
rsUcPripady.getString("popis")));
        }

        return DataModelFactory.create(v, "[id(ucPripadId), request] popis");
    }
}

```

```

    } finally {
        SqlUtils.sqlCleanup(psUcPripady, rsUcPripady);
    }
}

private DataModel<SimpleListItem> naplnRoky(Date datum) {
    List<SimpleListItem> list = new ArrayList();
    for (int i = 2002; i <= TimeU.getRok(datum); i++) {
        list.add(new SimpleListItem(""+i, ""+i));
    }
    return DataModelFactory.create(list, "[id(id), request] value");
}

private DataModel<TypUctu> naplnTypyUctu(Connection con) throws SQLException{
    TypUctu.Operation typOp = new TypUctu.Operation();
    try {
        typOp.connect(DBConnection.certificateConnection(con));
        Enumeration enumerace = typOp.nactiCiselnik(true);
        List<TypUctu> list = new ArrayList<TypUctu>();
        while (enumerace.hasMoreElements()) {
            TypUctu tu = (TypUctu)enumerace.nextElement();
            list.add(tu);
        }
        return DataModelFactory.create(list, "[id(typ_uctu_id), request] nazev");
    } finally {
        typOp.disconnect();
    }
}

private DataModel<SimpleListItem> naplnMesice() {
    List<SimpleListItem> list = new ArrayList();
    list.add(new SimpleListItem("1", Lang.alv("Leden", "January")));
    list.add(new SimpleListItem("2", Lang.alv("Únor", "February")));
    list.add(new SimpleListItem("3", Lang.alv("Březen", "March")));
    list.add(new SimpleListItem("4", Lang.alv("Duben", "April")));
    list.add(new SimpleListItem("5", Lang.alv("Květen", "May")));
    list.add(new SimpleListItem("6", Lang.alv("Červen", "June")));
    list.add(new SimpleListItem("7", Lang.alv("Červenec", "July")));
    list.add(new SimpleListItem("8", Lang.alv("Srpen", "August")));
    list.add(new SimpleListItem("9", Lang.alv("Září", "September")));
    list.add(new SimpleListItem("10", Lang.alv("Říjen", "October")));
    list.add(new SimpleListItem("11", Lang.alv("Listopad", "November")));
    list.add(new SimpleListItem("12", Lang.alv("Prosinec", "December")));
    return DataModelFactory.create(list, "[id(id), request] value");
}

```

```

public AutomatickeUctovaniPlatebWApp() throws SQLException, NamingException {
    Connection con = null;
    try {
        con = jfx.security.DBConnection.getOraConnection();
        typUctu.setDataModel(naplnTypyUctu(con));
        roky.setDataModel(naplnRoky(new Date()));
        mesice.setDataModel(naplnMesice());
        typDokladu.setDataModel(naplnTypyDokladu(con));
        ucetniPripad.setEnabled(false);
        // subSystemy.setDataModel(naplnSubsystemy(getContext().getConnection(), new
Date(), ((TypUctu)typUctu.getSelectedValue()).getTyp()));

        subSystemy.setEnabled(false);
        odeslat.setEnabled(false);
        textVDokladu.setText("");

    } finally {
        SqlUtils.sqlCleanup(con,null, null);
    }

}

@OnSelection(component="typUctu")
private void povolSubsystemy() throws SQLException {
    subSystemy.setDataModel(naplnSubsystemy(getContext().getConnection(), new
Date(), ((TypUctu)typUctu.getSelectedValue()).getTyp()));
    subSystemy.setEnabled(true);
}

@OnSelection(component="typDokladu")
private void povolitUcPripady() throws SQLException {

    Date datum = TimeU.datum("1", mesice.getSelectedValue().getId(),
roky.getSelectedValue().getId());
    SubsystemInfo subSys = subSystemy.getSelectedValue();
    DokladTypInfo dokladT = typDokladu.getSelectedValue();
    TypUctu ucetT = typUctu.getSelectedValue();

    if (subSys != null && dokladT != null && ucetT != null) {
        ucetniPripad.setDataModel(naplnUcPripady(getConnection(), datum, subSys,
dokladT, ucetT.getTyp()));
    }
    ucetniPripad.setEnabled(true);
    getContext().addRequestedComponent(ucetniPripad);
}

```

```

@OnSelection(component="ucetniPripad")
private void povolZpracovani() throws SQLException {
    //textVDokladu.setText(ucetniPripad.getSelectedValue().getPopis());
    //textVDokladu.setText(ucetniPripad.getSelectedValues().)
    /*UcPripadInfo ucPripad = ucetniPripad.getSelectedValues().iterator().next();

    textVDokladu.setText(ucPripad.getPopis());*/

    odeslat.setEnabled(true);
}

@OnAction(component="odeslat")
private void stisknutoZpracovat() {
    System.out.println("ZPRACOVAT");
    Iterator<UcPripadInfo> iter = ucetniPripad.getSelectedValues().iterator();
    while (iter.hasNext()) {
        //System.out.println(iter.next().getPopis());
        UcPripadInfo ucP = iter.next();
        System.out.println(ucP.getUcPripadId() + ucP.getPopis());
    }
    System.out.println("KONEC");

    if (vyberOk.isValid()) {
        System.out.println("Vsechno dobry");

        AutomatickeUctovaniPlatebTask uctoTask = new
AutomatickeUctovaniPlatebTask(ucetniPripad.getSelectedValues(), TimeU.datum("1",
mesice.getSelectedValue().getId(), roky.getSelectedValue().getId()),
        zdanovaciObdobi.getDate(), subSystemy.getSelectedValue(),
        textVDokladu.getText(),
        typUctu.getSelectedValue(), FSecurity.vratOsCis());

        BatchTask task = new BatchSimpleTask(uctoTask, APP_ID, "SUPO
"+subSystemy.getSelectedValue().getSubsystemNazev()+
"+(commit.isSelected()?"(commit)": "(test)",
        Roles.SUPO_UCT_H, !commit.isSelected(), 3);

        WBatchQueue.addTask(task);

        tabbedPane.setSelectedTab(vysledky);

```

```

        }

    }

    @OnSelection(component="subSystemy")
    private void zmenenSubsystem() throws SQLException {
        if (typDokladu.getSelectedValue()!=null) {
            povolitUcPripady();
        }
    }

    @Override
    protected void run() throws Exception {
        // TODO: write code here
    }
}

```

## Příloha 4 - Ukázka kódu XSL šablony

V této příloze si uvedeme ukázkou kódu XSL šablony aplikace z přílohy 5.

```
<?xml version="1.0" encoding="iso-8859-2"?>

<!DOCTYPE test [
  <!ENTITY nbsp "&#160;">
]>

<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0"
  xmlns:jfx="https://inet.muni.cz/jfx"
  xmlns:jff="java:jfx.xml.XSLFunctions">
  <xsl:import href="/output/jfx_html_template-std.xsl"/>
  <xsl:import href="/output/jfx_output_utils.xsl"/>
  <xsl:import href="/output/jfx_wui_basic.xsl"/>
  <xsl:import href="/output/jfx_wui_advance.xsl"/>

  <xsl:output method="xhtml"/>

  <xsl:template match="/" mode="pageBody">

    <xsl:call-template name="servlet-help">
      <xsl:with-param name="text">
        Tato aplikace slouží k zadání parametrů a spuštění automatického účtování
        přijatých plateb.
      </xsl:with-param>
    </xsl:call-template>

    <form method="post" action="automatickeuctovaniplateb">

      <xsl:apply-templates select="//jfx:wui[@id='tabbedPane']"/>
    </form>

  </xsl:template>

  <xsl:template match="jfx:tab[@id='vyberParametru']">
    Výběr parametrů
  </xsl:template>
  <xsl:template match="jfx:tab[@id='vysledky']">
    Výsledky
  </xsl:template>
  <xsl:template match="jfx:wui[@id='odeslat']" mode="text">
    Zpracovat
  </xsl:template>
  <xsl:template match="jfx:wui[@id='textVDokladu']" mode="length">24</xsl:template>
```

```

<xsl:template match="jfx:wui[@id='vyberParametru']" mode="layout" >
  <table>
    <tr><td>Typ účtu:</td><td><xsl:apply-templates
select="jfx:wui[@id='typUctu']"/></td></tr>
    <tr><td>Účetní období:</td><td><xsl:apply-templates
select="jfx:wui[@id='mesice']"/><xsl:apply-templates
select="jfx:wui[@id='roky']"/></td></tr>
    <tr><td>Zdaňovací období:</td><td><xsl:apply-templates
select="jfx:wui[@id='zdanovaciObdobi']"/></td></tr>
    <tr><td>Systém:</td><td><xsl:apply-templates
select="jfx:wui[@id='subSystemy']"/></td></tr>
    <tr><td>Typ dokladu:</td><td><xsl:apply-templates
select="jfx:wui[@id='typDokladu']"/></td></tr>
    <tr><td>Účetní případ:</td><td><xsl:apply-templates
select="jfx:wui[@id='ucetniPripad']"/></td></tr>

    <tr><td>Text v dokladu:</td><td><xsl:apply-templates
select="jfx:wui[@id='textVDokladu']"/></td></tr>
    <tr><td colspan="2"><xsl:apply-templates
select="jfx:wui[@id='commit']"/><small>Zadané úkoly budou provedeny pouze v případě
zatržení</small></td></tr>
    <tr><td><xsl:apply-templates select="jfx:wui[@id='odeslat']"/></td></tr>

  </table>
</xsl:template>

</xsl:stylesheet>

```

## Příloha 5 - Ukázka aplikace systému

V této příloze uvedeme ukázkou aplikace v našem systému, kterou tvoří kód příloh 5 a 6.

The screenshot shows the 'Automatické účtování plateb' (Automatic payment accounting) application interface. The header includes the 'inet.muni.cz' logo and navigation tabs: 'Novinky', 'Personalistika', 'Ekonomika' (selected), 'Studovny', and 'Provoz'. Below these are links for 'Věda a výzkum', 'InfoZdroje', 'O Inetu', and 'Rejstřík'. The user is logged in as 'Mgr. Petr Výmola (60663)'. The main content area has a title 'Automatické účtování plateb' and a description: 'Tato aplikace slouží k zadání parametrů a spuštění automatického účtování plateb.' There are two tabs: 'Výběr parametrů' (selected) and 'Výsledky'. The 'Výběr parametrů' tab contains several dropdown menus and a text field: 'Typ účtu:' (primární účet), 'Účetní období:' (Květen 2010), 'Zdaňovací období:' (11.05.2010), 'Systém:' (Bankovníky), 'Typ dokladu:' (Přijatá platba), and 'Účetní případ:' (a list including 'PPR Bankovník 1-CPS', 'PPR Bankovník 2-UKB', 'PPR Bankovník 7-SKM', and 'PPR Bankovník 6-SKM'). There is also a 'Text v dokladu:' field and a checkbox for 'Zadané úkoly budou provedeny pouze v případě zatřžení'. A 'Zpracovat' button is at the bottom. The footer shows the date '11.05.2010' and the email 'inet@ics.muni.cz'.

inet.muni.cz

Novinky Personalistika **Ekonomika** Studovny Provoz

Věda a výzkum InfoZdroje O Inetu Rejstřík

Přihlášený uživatel -- Mgr. Petr Výmola (60663)

\*\*\*E přidat k mým oblíbeným | nastavení | česky -- english

### Automatické účtování plateb

Tato aplikace slouží k zadání parametrů a spuštění automatického účtování plateb.

**Výběr parametrů** Výsledky

Typ účtu: primární účet

Účetní období: Květen 2010

Zdaňovací období: 11.05.2010

Systém: Bankovníky

Typ dokladu: Přijatá platba

Účetní případ: --- Vyberte položku (celkem 10) ---  
PPR Bankovník 1-CPS  
PPR Bankovník 2-UKB  
PPR Bankovník 7-SKM  
PPR Bankovník 6-SKM

Text v dokladu:

☐ Zadané úkoly budou provedeny pouze v případě zatřžení

Zpracovat

Kontaktní e-mail: [supo-inet@ics.muni.cz](mailto:supo-inet@ics.muni.cz)  
Informace o serveru: [inet.muni.cz](http://inet.muni.cz)  
Přístupy z: 147.251.18.186, 84.244.86.160, ...

11.05.2010 inet@ics.muni.cz