

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## ZPRACOVÁNÍ EVIDENČNÍCH LISTŮ DŮCHODOVÉHO POJIŠTĚNÍ V INFORMAČNÍM SYSTÉMU SAP R/3

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. ONDŘEJ LEŠTINSKÝ

BRNO 2013



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# **ZPRACOVÁNÍ EVIDENČNÍCH LISTŮ DŮCHODOVÉHO POJIŠTĚNÍ V INFORMAČNÍM SYSTÉMU SAP R/3**

PROCESSING OF PENSION INSURANCE INFORMATION IN IS SAP R/3

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. ONDŘEJ LEŠTINSKÝ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Doc. Dr. Ing. DUŠAN KOLÁŘ**

BRNO 2013

## **Abstrakt**

Tato práce se zabývá zpracováním evidenčních listů důchodového pojištění v informačním systému SAP. Úvodní kapitoly práce popisují informační systém SAP, jeho vývojové prostředí a formulářové rozhraní pro tisk. Dále se práce zaměřuje na generování evidenčních listů, jejich zobrazení v SAP a jejich odeslání na Českou správu sociálního zabezpečení.

## **Abstract**

This thesis discusses processing of pension insurance information in information system SAP. Introductory chapters of this thesis describe IS SAP, development tools and interface for printing forms. Afterwards it is focused on generation insurance information, including displaying and transferring them to the Czech Social Security Administration.

## **Klíčová slova**

SAP, ABAP, ABAP Workbench, ČSSZ, ELDP, VREP, formuláře Adobe

## **Keywords**

SAP, ABAP, ABAP Workbench, CSSA, ELDP, Public interface for e-service, Adobe Forms

## **Citace**

Ondřej Leštinský: Zpracování Evidenčních listů důchodového pojištění v informačním systému SAP R/3, diplomová práce, Brno, FIT VUT v Brně, 2013

# Zpracování Evidenčních listů důchodového pojištění v informačním systému SAP R/3

## Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením pana Doc. Dr. Ing. Dušana Koláře

.....  
Ondřej Leštinský  
19. května 2013

## Poděkování

Děkuji vedoucímu práce, panu Doc. Dr. Ing. Dušanovi Kolářovi, za odborné vedení a cenné rady. Rovněž děkuji svému konzultantovi, panu Ing. Zděnkovi Zitovi, za ochotu a cenné rady.

© Ondřej Leštinský, 2013.

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

# Obsah

<b>1 Úvod</b>	<b>3</b>
<b>2 Informační systém SAP</b>	<b>4</b>
2.1 Společnost SAP AG	4
2.2 Informační systém SAP	4
2.3 Architektura SAP	5
2.4 Modul HR	6
2.4.1 Modul HR na VUT	7
2.5 ABAP Workbench	8
2.5.1 ABAP	8
2.5.2 Nástroje pro tvorbu aplikací	9
2.5.3 ABAP Dictionary	9
2.6 Vývojové prostředí	10
<b>3 Formulářové rozhraní</b>	<b>11</b>
3.1 Přímý tisk z ABAP	11
3.2 SAP Script	11
3.3 SAP Smart Forms	12
3.4 Microsoft Excel	12
3.5 SAP Adobe Forms	13
3.6 Výběr vhodného rozhraní	13
<b>4 Komunikační kanál ČSSZ</b>	<b>15</b>
4.1 Česká správa sociálního zabezpečení	15
4.2 Veřejné rozhraní pro e-Podání	15
4.3 e-Podání	15
4.3.1 Aplikační komunikační protokol	16
4.3.2 Komunikační kanály	16
4.3.3 Veřejné rozhraní pro e-Podání	16
4.3.4 Formáty zpráv	18
4.4 Návrh přímého podání tiskopisů	19
4.4.1 Odeslání podání	19
4.4.2 Vyzvednutí odpovědi	20
<b>5 Evidenční listy důchodového pojištění</b>	<b>21</b>
5.1 Vedení a předkládání ELDP	21
5.2 Způsoby předkládání ELDP	21
5.3 Způsob vyplňování ELDP	22

5.4	Návrh aplikace . . . . .	22
5.4.1	Zdrojová data . . . . .	23
5.4.2	Diagram případů použití . . . . .	24
5.4.3	Popis případů použití . . . . .	24
5.4.4	Hlavní aplikace . . . . .	25
5.4.5	Generování údajů . . . . .	26
5.4.6	Zobrazení údajů . . . . .	27
5.4.7	Tisk formulářů . . . . .	27
<b>6</b>	<b>Implementace</b>	<b>28</b>
6.1	Použité konvence . . . . .	28
6.2	Hlavní obrazovka . . . . .	28
6.3	Zpracování evidenčních listů . . . . .	31
6.3.1	Definice interních tabulek . . . . .	32
6.3.2	Inicializace údajů . . . . .	33
6.3.3	Získ osobních čísel . . . . .	34
6.3.4	Generování dat . . . . .	34
6.3.5	Uložení vygenerovaných dat . . . . .	40
6.3.6	Zobrazení vygenerovaných dat . . . . .	42
6.4	Tabulkové zobrazení . . . . .	42
6.4.1	ALV tabulka . . . . .	43
6.4.2	Uložení změn . . . . .	44
6.4.3	Opravný formulář . . . . .	44
6.4.4	PDF formulář . . . . .	44
6.5	Vytváření XML souborů . . . . .	45
6.6	Odesílání dat ČSSZ . . . . .	46
6.6.1	Zobrazení podání . . . . .	47
6.6.2	Odesílací menu . . . . .	47
6.6.3	Šifrování a digitální podpis . . . . .	48
6.6.4	Příprava GovTalk obálky pro odeslání podání . . . . .	49
6.6.5	Odeslání podání . . . . .	49
6.6.6	Náhradní řešení . . . . .	50
6.6.7	Zpracování odpovědi . . . . .	51
6.6.8	Dotaz na výsledek zpracování . . . . .	52
6.6.9	Zobrazení chyb . . . . .	52
6.7	Hromadný tisk . . . . .	53
6.8	Testování . . . . .	54
6.9	Programová a uživatelská dokumentace . . . . .	54
<b>7</b>	<b>Závěr</b>	<b>56</b>
7.1	Zhodnocení práce . . . . .	56
7.2	Další rozšíření . . . . .	56

# Kapitola 1

## Úvod

Téměř všechny střední a větší společnosti zpracovávají a ukládají obrovské množství datových záznamů. Tento proces, při kterém často spolupracuje několik uživatelů nad stejnou databází, musí vyhovovat nejprísnějším požadavkům na zabezpečení dat a architekturu systému. Software, který splňuje tyto požadavky, je možné označit jako podnikový informační systém. Jednou ze společností, která takové systémy nabízí, je společnost SAP AG. Skupina produktů SAP představuje kompletní řešení především pro všechna interní oddělení podniku a jejich produkty patří k nejvíce rozšířeným v dané sféře. Řešení pod souhrnným názvem systém mySAP je používáno i na vysokých školách, výjimkou není ani Vysoké učení technické v Brně, kde se komponenta SAP R/3<sup>1</sup> používá od roku 2004 a zajišťuje zde podporu pro personalistiku, účetnictví, evidenci majetku a mnoho dalších důležitých věcí.

Tato práce se zaměřuje na rozšíření funkcionality informačního systému SAP na VUT o zpracování evidenčních listů důchodového pojištění (dále jen ELDP), jejichž vedení je jednou z povinností zaměstnavatelů, uloženou zákonem o organizaci a provádění sociálního zabezpečení č. 582/1991 Sb. Cílem práce je nahradit stávající způsob zpracování ELDP na VUT tak, aby celý proces byl prováděn v prostředí SAP, odpovídal aktuálním zákonům a generoval správná data. Výsledná implementace umožní jak ruční, tak i hromadné vyplnění evidenčních listů na základě údajů obsažených v SAP. Hlavním důvodem reimplementace je skutečnost, že řešení nabízené SAP negeneruje data na VUT vždy správným způsobem. Částečně je to způsobeno tím, že VUT má specificky upravený způsob vedení zaměstnanců. Stávající generování ELDP proto vedlo k velkému množství chyb, které bylo potřeba ručně opravovat. Mezi další důvody patří to, že stávající řešení neumožňuje vyplněné evidenční listy v elektronické podobě přímo ze SAP odeslat České správě sociálního zabezpečení (dále jen ČSSZ) a také nemožnost vytisknout vygenerovaná podání do originálního formuláře.

Práce je členěna následujícím způsobem. Druhá kapitola seznámí čtenáře se základními pojmy týkající se IS SAP, popisuje architekturu systému a to, jakým způsobem zde probíhá vývoj aplikací. Text třetí kapitoly shrnuje tisková rozhraní a blíže se zaměřuje na Adobe Forms. Následující kapitola charakterizuje komunikační kanál ČSSZ a obsahuje návrh aplikace, která skrz něj umožní univerzální komunikaci. Pátá kapitola seznámí čtenáře se samotným generováním ELDP včetně návrhu aplikace. Popis implementace navržené aplikace je obsažen v šesté, předposlední, kapitole. Závěr diplomové práce hodnotí dosažené výsledky a navrhuje další rozšíření.

---

<sup>1</sup>R znamená real-time, 3 je označení verze

## Kapitola 2

# Informační systém SAP

Kapitola seznámí čtenáře se základními pojmy týkající se SAP a stručně charakterizuje architekturu systému. Blíže popisuje modul HR na VUT v Brně, ze kterého budou získávána data pro evidenční listy. Závěrečné části kapitoly jsou zaměřeny na programovací jazyk ABAP a jeho vývojové prostředí. Informace obsažené v této kapitole jsou čerpané ze zdrojů [2] [4] [5], pokud není explicitně uvedeno jinak.

### 2.1 Společnost SAP AG

SAP AG je Německá společnost zabývající se vývojem podnikových informačních systémů, též známých jako systémy ERP. Poskytuje počítačové řešení integrace a automatizace podnikových procesů, jako je finančníctví, logistika či lidské zdroje. Společnost byla založena bývalými pracovníky IBM v roce 1972. O rok později byl dokončen vývoj jejich první aplikace pro oblast finančního účetnictví, která se stala základem systému SAP R/1. Firma se do povědomí dostala v osmdesátých letech minulého století uvedením řešení SAP R/2. Absolutní špičkou ve vývoji ERP se stala v roce 1992. Od té doby se stále drží na vrcholu. SAP R/3 je založený na architektuře klient-server a využití relační databáze.

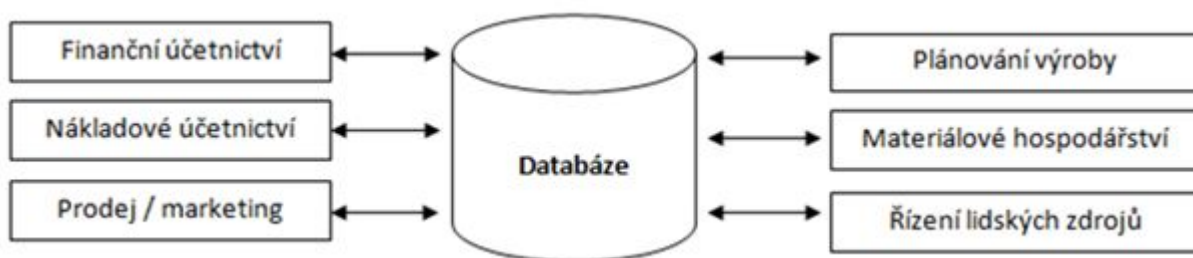
K dalšímu posunu došlo uvedením SAP R/3 Enterprise v roce 2002. Původní systém báze byl vyměněn za nový Web Application Server. Z hlediska funkčnosti se nejednalo o velký zásah, došlo k přeuspořádání modulů za účelem vývoje nových rozšíření. Od roku 2004 jsou nově uspořádané komponenty dodávány na trh, jejich centrálním produktem se stal *mySAP Business Suite*. Technologické komponenty byly úplně odděleny od aplikačních komponent a nadále jsou souhrnně označovány *SAP NetWeaver*.

Firma je úspěšná mimo jiné tím, že nabízí kompletní řešení pro všechna oddělení podniku a pokrývá i procesy. Postupem času se z tradiční společnosti čistě zaměřené na ERP software stala firma nabízející celou řadu firemních řešení a služeb podporujících integraci a spolupráci.

### 2.2 Informační systém SAP

Informační systém SAP byl navržen tak, aby umožnil kompletní řízení potřeb společnosti. Umožňuje to vícevrstvá architektura klient-server, stejné uživatelské rozhraní aplikací, integrace veškerých aplikací (obr. 2.1) a rozsáhlá podpora služeb včetně školení, konzultací, kontroly kvality a mnoho dalších.





Obrázek 2.1: Integrace dat

Všechny produktové komponenty závisí na klientovi a jsou připraveny tak, aby bylo možné v jedné instanci vytvořit více právně a ekonomicky samostatných podniků. Komponenty podporují více jazyků a jsou připraveny pro nasazení v mezinárodním prostředí.

Každému uživateli systému je přiděleno jedinečné uživatelské jméno, se kterým se do systému přihlašuje, aby byl schopen pracovat. Principiálně je systém založen na pozitivním uživatelském konceptu, tj. je-li v systému vytvořen nový uživatel, nemá z počátku žádná práva. Až v závislosti na pozici v podniku, kterou uživatel zastává, mu jsou povoleny potřebné transakce.

Celkově se systém může zdát zbytečně složitý, ale má to dobrý důvod. Zajišťuje, že veškeré změny objektů jsou transparentní a vystopovatelné. Výhody SAP:

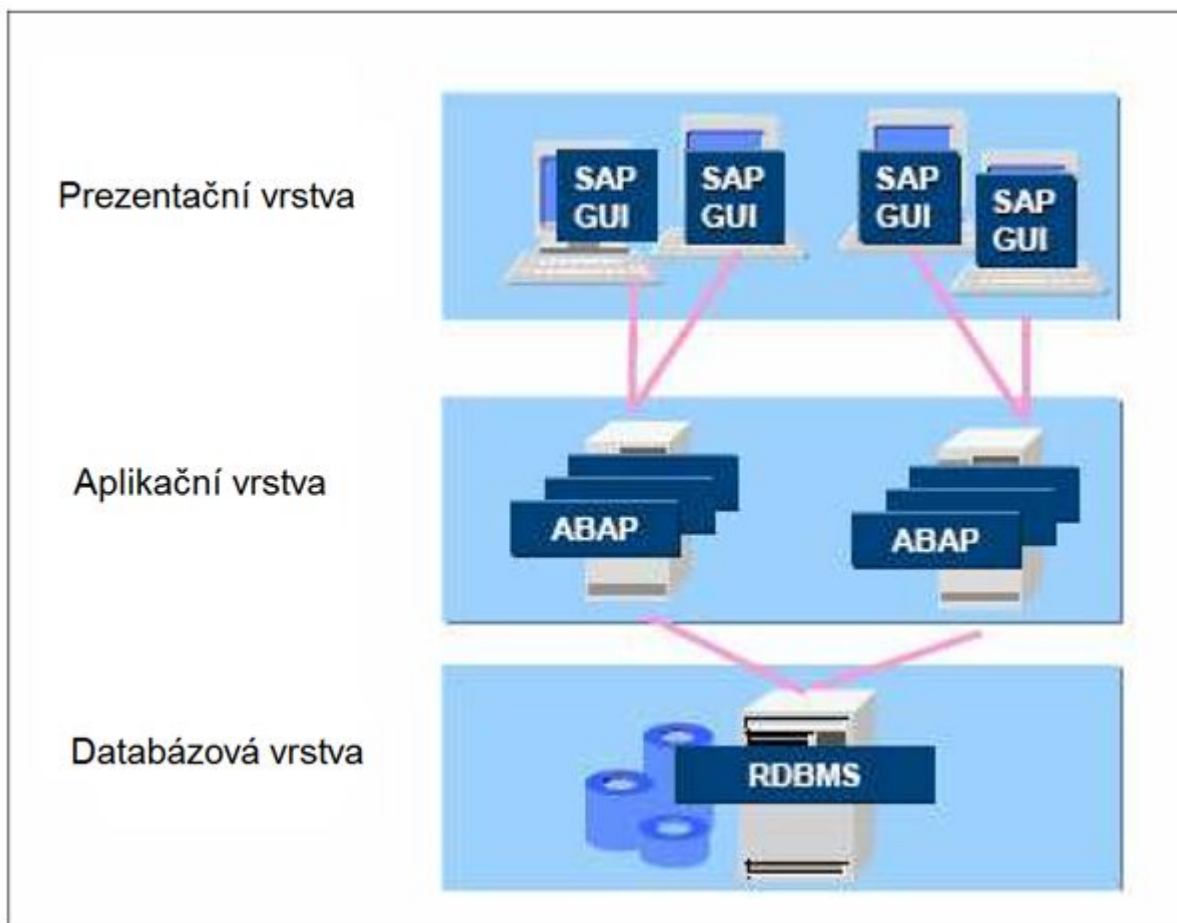
- Lepší plánování, řízení a kontrola podnikových obchodních procesů.
- Jednotná a konzistentní databáze.
- Zvýšení flexibility s ohledem na možnost přizpůsobení IS a obchodního procesu měnícím se požadavkům.
- Zkrácení doby potřebné k provedení a dokončení jednotlivých obchodních procesů.
- Kvalitativní zlepšení podnikových obchodních procesů.

## 2.3 Architektura SAP

Provoz ERP aplikací vyžaduje uživatelskou nezávislost na technickém řešení implementace. Základním principem systému SAP je třívrstvá architektura, kterou lze provozovat na řadě platform. Systém SAP je složen z následujících vrstev:

- **Prezentační vrstva** se stará o vstup a výstup dat a komunikaci uživatele s počítačem pomocí dialogových obrazovek, která se uskutečňuje na klientské části – prezentačním serveru. Prezentační servery tvoří prezentační vrstvu systému SAP. Speciální programový balík pro grafické uživatelské rozhraní, SAP GUI, zajišťuje komunikaci mezi prezentačním serverem a aplikačním serverem, bez něj není možná komunikace se systémem SAP.
- **Aplikační vrstva** slouží k centrálnímu zpracování dat – aplikační server. Komunikuje s klientem přes rozhraní SAP GUI a s databází přes systém pro správu databází. Tento systém pracuje v pozadí různými způsoby v závislosti na fyzické databázi. Aplikační server se může skládat z více fyzických serverů.

- **Databázová vrstva** zajišťuje uložení dat na svůj databázový server. V závislosti na množství dat a dalších faktorech mohou být aplikační a databázové servery umístěny na jednom nebo více počítačích.



Obrázek 2.2: Architektura systému [5]

Veškerá data (programy, nastavení systému či data aplikací) jsou uložena na databázovém serveru a podle požadavků aplikací (finanční účetnictví, logistika a jiné), které jsou zpřístupněny na aplikačních serverech, načítají data a předávají je prezentačnímu serveru.

## 2.4 Modul HR

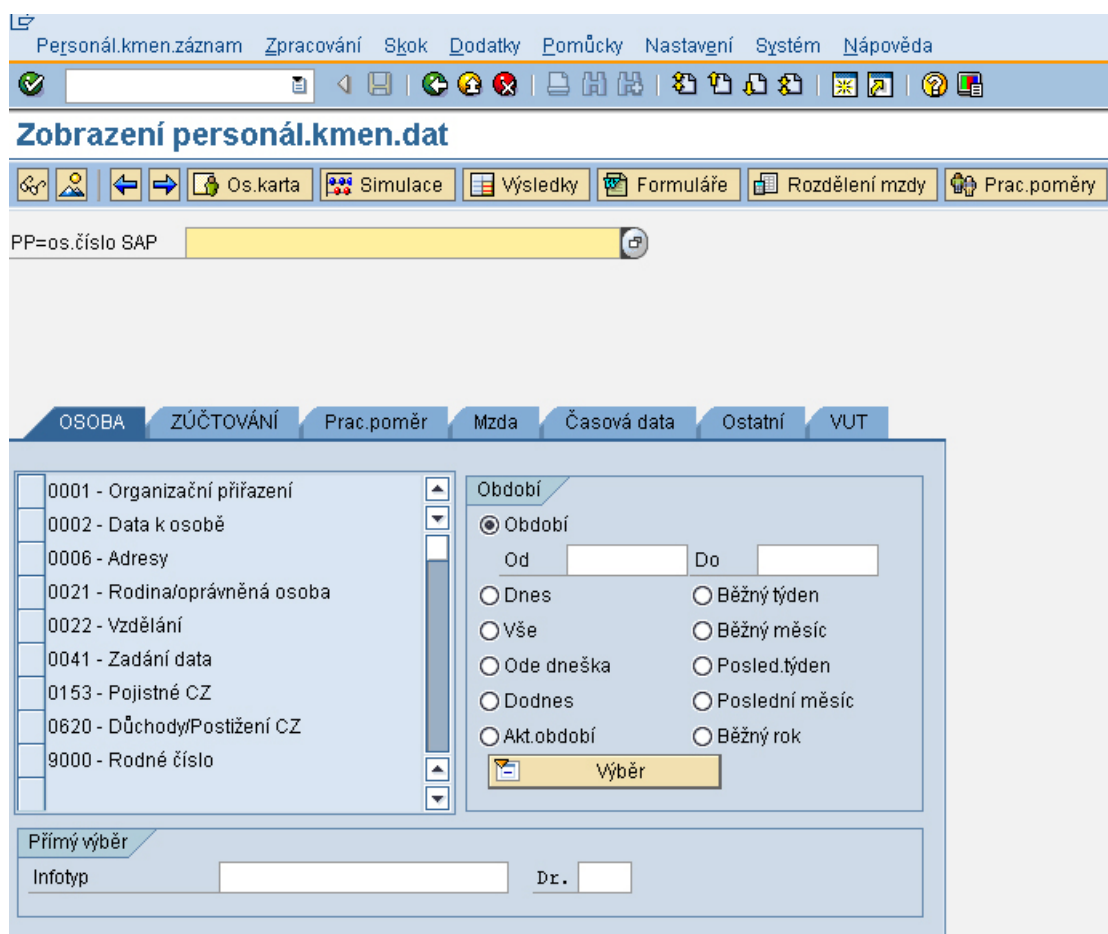
Klasické R/3 aplikace jsou obvykle rozděleny do 3 kategorií, lidské zdroje (HR), finance (FI) a logistika (SD). Každá z těchto kategorií obsahuje stovky pracovních procesů uspokojující požadavky moderních společností. Z pohledu této práce je nejdůležitější kategorie týkající se lidských zdrojů.

Aplikace, které jsou součástí modulu personalistiky nebo též řízení lidských zdrojů, umožňující provádění úkonů souvisejících s tímto oborem:

- nábor personálu,

- správa uchazečů o zaměstnání,
- zúčtování mezd,
- řízení pracovní doby,
- zúčtování pracovních výloh.

Kmenová data zaměstnanců jsou zadávána a udržována prostřednictvím tzv. **infotypů**. Ty seskupují jednotlivá datová pole z hlediska jejich obsahu. Každý infotyp je identifikován jedinečným čtyřmístným klíčem. Z hlediska databáze představují vzájemně související množinu datových záznamů. Pro každý infotyp je určena jedna obrazovka (obr. 2.3), která slouží k zadávání a údržbě dat.



Obrázek 2.3: Zobrazení personálních kmenových dat

#### 2.4.1 Modul HR na VUT

Na VUT tvoří modul HR jednu z nejdůležitějších součástí. Spravuje údaje o všech zaměstnancích VUT a zabezpečuje měsíční výpočet jejich mezd. VUT je veřejná vysoká škola, která spadá pod veřejnou správu, která má hlavně v oblasti HR svá specifika - plat zaměstnance se skládá z tarifní mzdy, osobního ohodnocení a odměny.

Každému pracovně právnímu vztahu je přiřazeno osobní číslo, jehož počáteční číslo zároveň identifikuje skupinu pracovníků:

- 1 značí pracovní poměry,
- 2 značí navazující vztahy (od používání se ustupuje),
- 7 značí dohodu o pracovní činnosti (DPČ) a
- 8 značí dohodu o provedení práce (DPP).

Druh pracovně-právního vztahu je definován v infotypu 0016 (například hlavní na dobu určitou). Zaměstnanec může mít přiřazeno několik pracovních poměrů a tedy několik osobních čísel. V infotypech 0031 a 0121 lze související osobní čísla logicky propojit. Propojení mimo jiné slouží ke správnému stanovení daně a pojistného. V IT 0121 je zvoleno hlavní osobní číslo a IT 0031 slouží pro zadání propojení. To je jeden z hlavních rozdílů oproti klasickým firmám, kde je zaměstnanec veden pod jediným osobním číslem. Velkým rozdílem je i vytvoření vlastního IT 9008, který byl zaveden pro slučování dohod o provedení práce u jednoho zaměstnance.

Také je používáno tzv. více-zdrojové financování, to znamená, že pro výpočet mezd jsou použity údaje z infotypu 0008 (základní příjmy) a substituční tabulka. Z tohoto důvodu musel být při nasazení pozměněn standardní algoritmus pro výpočet mezd. Celkově HR modul obsahuje velké množství zákaznických úprav bez dostupné dokumentace, proto je někdy velmi komplikované zjistit, co vše případná úprava ovlivní.

## 2.5 ABAP Workbench

Pro vývoj aplikací v prostředí SAP slouží vestavěná sada nástrojů ABAP Workbench, která obsahuje všechny nepostradatelné nástroje pro vývoj a návrh programů, tvorbu obrazovek, definici menu a spoustu dalších. Nezbytnou součástí je i nástroj pro ladění programů. Vývoj aplikací probíhá na systému pro testování a vývoj, uživatel musí být registrován v systému jako programátor, k čemuž potřebuje speciální vývojový klíč, o který správce systému musí požádat přímo společnost SAP.

### 2.5.1 ABAP

ABAP<sup>1</sup> je programovací jazyk, vyvinutý společností SAP, určený zejména pro databázové aplikace orientované na dialogy. Jazyk podporuje objektově orientované programování. ABAP programy nejsou uloženy v externích souborech, jako je tomu u jiných programovacích jazyků, jako například C++ či Java, ale jsou uloženy v SAP databázi. Tam je uložen zdrojový kód a generovaný kód. Generovaný kód může být přirovnán k „bytecode“ z prostředí programovacího jazyka Java. ABAP programy jsou prováděny pod kontrolou běhového systému, který je součástí SAP jádra. Je tedy zřejmá podobnost s virtuálním strojem Java VM. „Runtime“ systém je totiž zodpovědný za zpracování příkazů, logickou posloupnost obrazovek a reakce na události (klepnutí na tlačítko). Klíčovou komponentou „runtime“ systému je databázové rozhraní převádějící databázově nezávislé ABAP příkazy Open SQL do nativního SQL, kterému rozumí SRBD. Databázové rozhraní řídí veškerou komunikaci ABAP programů s relační databází [16].

---

<sup>1</sup>Advanced Business Application Programming

ABAP je imperativní jazyk, který se svou syntaxí dá přirovnat k programovacímu jazyku C. Jednotlivé ABAP příkazy a kompletní syntax jazyka může čtenář nalézt v oficiálním manuálu vydaném společností. [8]

### 2.5.2 Nástroje pro tvorbu aplikací

Pravděpodobně nejdůležitější nástroj používaný pro tvorbu reportů je **Editor ABAP**. Slouží pro tvorbu nových či editování stávajících programů. Pro ovládání editoru jsou připraveny dvě nástrojové lišty umístěné nad částí určenou pro psaní kódu. Lišty mimo jiné obsahují ovládací tlačítka pro přepínání režimů (změna a zobrazení zdrojového kódu), pro změnu stavu programu z aktivního na neaktivní – v praxi může existovat několik verzí objektu, ale jen jedna z nich je v danou chvíli aktivní. Také zahrnují tlačítka pro syntaktickou kontrolu zdrojového kódu a aktivaci programu. Pokud program není aktivní, systém touto funkcí vytvoří aktivní verzi a verzi pro spuštění. Aktivace zahrnuje i kontrolu syntaxe.

Pro návrh ovládacích prvků obrazovky typu nabídkové lišty, tlačítka či ikony, lze použít nástroj **Menu Painter**. V tomto nástroji se definuje, které funkce mají být vyvolány a jakým způsobem mají být volány po výběru položky z nabídky nebo stisknutím klávesy. Dalším nástrojem je **Screen Painter**, který je určený k návrhu elementů pro vstup a výstup dat. Umožňuje definici pozice textů, vstupních polí, výstupních polí, navrhuje jednotlivé karty dialogů, tabulky atd. **Function Builder** je nástroj umožňující vývoj a správu funkčních modulů (dále jen FM). Funkční moduly si lze představit jako standardní podprogramy pro přesně vymezené úlohy, které lze zavolat z libovolného programu. FM mají definovaná rozhraní pro vstup a výstup dat a měly by být schopny zachytávat chyby. Příkladem takového modulu může být převod měny. Poslední, programátory hojně využívaný, je nástroj **Object Navigator**, který zastřešuje všechny související objekty projektu. Pro složitější projekty je nepostradatelný.

Všechny důležité aplikace v prostředí ABAP lze spouštět pomocí stromové struktury nabídky systému SAP nebo prostřednictvím **kódu transakce**, který může být rovněž zobrazen i ve stromové struktuře nabídky. Například kód transakce pro spuštění nástroje na editaci zdrojových kódů je „SE38“. Nástroj umožňuje i spuštění aplikace. Toho se využívá, jestliže program z nějakého důvodu nemá přiřazený kód transakce.

### 2.5.3 ABAP Dictionary

Pro čtení a úpravy dat v databázových tabulkách slouží nástroj **ABAP Dictionary**, který umožňuje zobrazovat, měnit a vytvářet objekty slovníku, jako jsou databázové tabulky, datové typy, skupiny typů, domény, nápovědy vyhledávání nebo objekty blokování. V systému SAP existují tři typy databázových tabulek:

- **Transparentní tabulky** přesně odpovídají fyzické reprezentaci v databázi.
- **„Pool“ tabulky** obsahují zvláštní typy dat (parametry programů či jejich dokumentace) a neslouží pro ukládání dat podnikových aplikací. „Pool“ tabulek tvoří jediná tabulka ve fyzické databázi. Záznamy jsou identifikovány názvem „pool“ tabulky a klíčovými poli.
- **„Cluster“ tabulky** kombinují datové záznamy se stejným klíčem z více tabulek do jednoho fyzického datového záznamu, tím dochází ke snížení počtu datových záznamů.

Při vytváření tabulky je nutné dodržet určitá pravidla. Název tabulky může mít maximálně 16 znaků a systém nerozlišuje mezi malými a velkými písmeny. Podle jmenné konvence začínají názvy všech uživatelských objektů písmenem Y nebo písmenem Z.

## 2.6 Vývojové prostředí

V běžné praxi se setkáváme se dvěma až čtyřmi úrovněmi, přes které musí nově vytvořený objekt projít, aby se mohl začít používat. Minimálně se jedná o testovací a produkční systém. Podle technického vybavení serveru mohou oba systémy běžet na jednom fyzickém počítači, technicky jsou však rozděleny na samostatné instance s vlastní databází. V prostředí se čtyřmi systémy jsou mezi vývojové a produkční systémy umístěny ještě dva systémy pro zajištění kvality. Jednotlivé systémy:

- **Systém pro testování a vývoj** se používá pro programování a testování. Nejprve jsou uložena nastavení SAP pro každého zákazníka dle jeho potřeb (tzv. „customizing“). Systémy pro testování a vývoj obvykle slouží k simulaci a k testování podnikových procesů. Tyto systémy mají často velký počet uživatelů a nelze zaručit ochranu dat na stejné úrovni, jakou vyžaduje produkční systém.
- **Konsolidační (ověřovací) systém** ověřuje kvalitu vývoje. Objekty z výše popsaného systému jsou transportovány do konsolidačního systému, který ověřuje kvalitu vývoje. Zpravidla obsahuje aktuálnější data v databázi a i proto přístup k tomuto systému podléhá přísnějším požadavkům. Systém bývá často používán pro akceptační testy. Pokud výsledek vývoje funguje bez problémů, je přesunut do produkčního systému.
- **Produkční systém** představuje nejdůležitější systém a pro společnost je největším rizikem. Tudiž má nejlepší ochranu. Nelze přímo měnit parametry systému, nelze vyvíjet a měnit systémová nastavení. Otestované objekty jsou do produkčního systému importovány přes transporty, jejichž průběh je protokolován.



Obrázek 2.4: Organizace vývojového prostředí v systémech SAP [5]

## Kapitola 3

# Formulářové rozhraní

Ve třetí kapitole jsou popsány některé z možností pro tvorbu tiskových formulářů v IS SAP. Nejdříve bude čtenář seznámen s nástroji, které poskytuje přímo SAP, to je jednoduchý textový výpis, SAP Script, respektive SAP Smart Forms. Popsány jsou i produkty jiných firem, konkrétně produkty společností Microsoft a Adobe. Na závěr kapitoly je diskutován výběr vhodného nástroje pro tvorbu formulářového rozhraní. Informace uvedené v této kapitole jsou čerpány z oficiálních dokumentací vydaných společnostmi SAP [9] [10] a www stránek [1] [7].

### 3.1 Přímý tisk z ABAP

Nejjednodušší realizací výstupu ze SAP je tisk údajů v programu na obrazovku pomocí příkazu WRITE jazyka ABAP. Text je vypisován na obrazovku uživatele, který ji pomocí standardní volby vytiskne. Jedná se o poměrně častou variantu využívanou hlavně u nenáročných výstupů. Na VUT se například daná realizace používá při tisku výplatních pásek a doposud se používá i pro evidenční listy důchodového pojištění. Výhody:

- jednoduchost,
- rychlost tisku.

Nevýhody:

- pouze textový výstup,
- pro změnu formátu je nutný zásah programátora,
- kvalita tisku.

### 3.2 SAP Script

Vzhledem k tomu, že ABAP samotný neumožňuje zpracování a formátování textu s využitím tabulek, obrázků či jiných objektů, nabízí SAP nástroje, které takové zpracování textu dovolují. Jedním z těchto nástrojů je SAP Script. SAP Script se skládá z 5 komponent:

- editor pro zadávání a úpravu textů,
- styly a formuláře pro tisk,

- výstupní modul,
- programové rozhraní a
- databázové tabulky pro ukládání textů, stylů a formulářů.

Práce s formuláři probíhá tak, že do vytvořeného formuláře jsou přes připravené programové rozhraní doplněny požadované údaje. Rozhraní je řízeno z prostředí programu v jazyce ABAP za pomoci funkčních modulů. Po doplnění údajů je formulář předán výstupnímu modulu k dalšímu zpracování, typicky k tisku. Nevýhodou je, že tento nástroj nepodporuje interakci s uživatelem a jeho úprava vyžaduje zásah programátora. [9]

### 3.3 SAP Smart Forms

SAP Smart Forms je další nástroj od společnosti SAP pro tvorbu a údržbu formulářů určených pro výstup dat (tiskárna, fax, email či XML soubor). SAP Smart Forms je náhradou za Smart Script. Jeho výhodou je v tom, že tvorba a údržba formulářů zabere o dost méně času a samotná úprava formuláře je možná i bez znalosti programování přímo přes grafické uživatelské rozhraní. Dalším rozšířením je podpora exportování dat pomocí XML souborů. Při aktivaci systém automaticky generuje funkční modul a běhové prostředí. Vkládat lze statické i dynamické tabulky. Zajímavou funkcí je potlačení grafického pozadí při tisku. Tvorba Smart Forms zahrnuje:

- rozložení formuláře,
- definice zobrazovaných elementů,
- logiku formuláře a
- rozhraní pro přenos dat do formuláře.

Při aktivaci Smart Forms systém generuje funkční modul, který zaobaluje potřebné atributy. Jakmile aplikace zavolá tento funkční modul, jsou skrz jeho rozhraní přenesena dříve vybraná tabulková data do formuláře přesně podle jeho definice. Stále není ovšem umožněna editace dat<sup>1</sup> v zobrazeném formuláři. [10]

### 3.4 Microsoft Excel

Mezi nejvíce rozšířené kancelářské balíky patří bezesporu Microsoft Office. SAP podporuje hned několik způsobů pro práci se soubory z tohoto balíku. Z hlediska zaměření této práce se jako nejvhodnější formát jeví formát tabulkového procesoru Excel.

Vytvoření nového souboru je možné pomocí volání funkčního modulu, který je založený na rozhraní Spreadsheet ze SAP Office Integration. Výhodou je jednoduché rozhraní (předává se interní tabulka a jméno souboru) vytvářející nativní binární soubor. Nevýhodou je, že při vytvoření souboru nemusí některé datové typy jazyka ABAP být zkonvertovány na správný tvar. Mezi další nevýhody patří to, že nepodporuje práci s více listy, neumožňuje změny rozložení a formátování a také absence zpracování na pozadí. Stejně vlastnosti má i funkční modul, který obstarává čtení ze souboru.

<sup>1</sup>Podporována je jednoduchá interakce v internetovém prohlížeči [3], která je pro účely této práce nepoužitelná.



SAP Office Integration obsahuje rozhraní I\_OI\_SPREADSHEET dovolující tvorbu více sofistikovaných tabulkových souborů, podporuje barevné zvýrazňování, individuální formátování, práci s více listy a spoustu dalších věcí. I přes nepřeberné množství možností formátování je použití snadné. Neumožňuje však zpracování na pozadí. Podporuje jak tvorbu souborů, tak jejich čtení.

Další možností je využít pouze Microsoft OLE (předchozí metody částečně již OLE používají), které ve srovnání s předešlými metodami nabízí více možností pro tvorbu souborů – rozložení pro tisk či definice hlavičky, respektive patičky souboru. Důsledkem toho je nižší výkon a nutné specifické znalosti OLE příkazů v prostředí Excel. Vadou také je, že případná chyba se neprojevuje přímo v prostředí SAP. Opět není umožněno zpracování na pozadí.

Existuje také možnost využít externí nástroje. Většinou se jedná o nástroje, které konvertují formát csv nebo wk1 do nativního formátu Excel. Program může být spuštěn přímo z ABAP reportu. [7]

### 3.5 SAP Adobe Forms

Vzájemná spolupráce firem SAP a Adobe vedla ke vzniku SAP Adobe Forms, umožňující tvorbu interaktivních formulářů – uživatelé mohou nejen zobrazit požadované údaje, ale také mohou doplnit další údaje, které systém zpracuje, a v ideálním případě uloží. Adobe Forms mohou být použity libovolnou aplikací založenou na SAP NetWeaver [13]. Jeho součástí je služba pro generování PDF formulářů. U interaktivního formuláře služba extrahuje data z formuláře a formou XML souboru je zasílá do systému. Hlavní rysy:

- tvorba šablon a dokumentů po jejich sloučení,
- editace formulářů online i offline,
- odeslání pomocí zabezpečených metod,
- kontrola konzistence,
- digitální podpis a
- rozšířené funkce, jako například komentáře ve formuláři.

SAP Adobe Forms nabízí velké vylepšení oproti Smart Forms a interaktivní formuláře by měly technologii v budoucnu plně nahradit.

Pro vytváření a modifikaci SAP Adobe Forms slouží nástroj Form builder v prostředí SAP a Adobe Livecycle Designer, respektive Adobe Reader na počítači. Interaktivní formuláře v SAP lze tvořit i pomocí technologie Java v prostředí Netwear Development Studio nebo pomocí jazyka ABAP v prostředí ABAP Workbench. [1]

### 3.6 Výběr vhodného rozhraní

První popsaná varianta je ideální pro nenáročný hromadný tisk, vzhledem k technickému vybavení a nízkým nákladům budou hromadné textové výstupy realizovány právě tímto způsobem. Další dvě výše popsané varianty sice umožňují grafický výstup, ale jsou již mírně zastaralé a od jejich používání se ustupuje. Postupně se nahrazují novějšími Adobe Forms.

Editační formulář lze realizovat jako klasickou obrazovku v nástroji Screen Painter. SAP nabízí celou řadu grafických prvků, které lze do obrazovky vložit. Od nejjednodušších textových popisků až po kontejnery, ve kterých lze zobrazit celou tabulku. Editovatelný formulář lze realizovat přímo jako obrazovku programu s využitím běžných vstupních polí.

**Evidenční list důchodového pojištění**

Nastavení psacího stroje    Technický kód    za rok    Typ ELDP    Oprava ELDP ze dne

Kód OSSZ    Místně příslušná OSSZ (UP PSSZ)

   -   

---

**1. Identifikace pojištěnce**

Příjmení (poslední)	Jméno	Titul	Datum narození	Rodné číslo pojištěnce
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Ulice	Číslo domu		Rodné příjmení	
<input type="text"/>	<input type="text"/>		<input type="text"/>	
Obec	Pošta	PSC (Post Code)	Stát	Místo narození
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

---

**2. Průběh pojištění v daném roce**

Kód	MR	Od	Do	Dny	1	2	3	4	5	6	7	8	9	10	11	12	1-12	Vylouč. doby	Výměrovací základ	Doby odečt.
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
																		Celkem	Celkem	Celkem
																		<input type="text"/>	<input type="text"/>	<input type="text"/>

---

**3. Identifikační údaje zaměstnavatele a podpisy**

Název zaměstnavatele	IC	Variabilní symbol
<input type="text"/>	<input type="text"/>	<input type="text"/>
Vydělečná činnost od		
<input type="text"/>		
Datum vyhotovení ELDP	Datum a podpis pojištěnce	
<input type="text"/>	<input type="text"/>	

3 5 4 5 6 0 5 4 6 4    Podpis a razítko zaměstnavatele    Pro potřeby OSSZ

Obrázek 3.1: Vzor formuláře

## Kapitola 4

# Komunikační kanál ČSSZ

Tato kapitola seznámí čtenáře se základní činností České správy sociálního zabezpečení (dále jen ČSSZ) a popíše způsob, jakým s ní lze elektronicky komunikovat. Konkrétně přes komunikační kanál nazvaný Veřejné rozhraní pro e-Podání. Součástí této kapitoly je i návrh aplikace pro přímé podávání tiskopisů z IS SAP. Veškeré informace o ČSSZ uvedené v následujícím textu jsou přebrány z jejich [www stránek \[20\]](#). Informace o komunikačním kanále jsou čerpány také z jejich příručky [\[18\]](#) určené přímo pro vývojáře aplikací.

### 4.1 Česká správa sociálního zabezpečení

ČSSZ je největší a v rámci státní správy ČR zcela výjimečnou finančně správní institucí spravující agendu asi 8,5 milionu klientů (z toho více než 2,9 milionu důchodců). Mimo jiné rozhoduje o dávkách důchodového pojištění a odpovídá za vedení klasických a automatizovaných evidencí dat potřebných pro rozhodování ve věcech důchodového pojištění, nemocenského pojištění a lékařské posudkové služby. [\[20\]](#)

### 4.2 Veřejné rozhraní pro e-Podání

Vzhledem k opakující se agendě nabízí ČSSZ službu zvanou „**e-Podání**“, která je určena pro opakované zasílání vybraných formulářů. Díky této službě dochází k urychlení vzájemné komunikace subjektů s ČSSZ. Díky 24hodinové dostupnosti, automatické odezvě a možnosti okamžité opravy opětovným zasláním, je používání této služby výhodné. Služba byla navržena s důrazem na základní bezpečnostní cíle elektronické komunikace, používá šifrovací certifikáty pro zajištění důvěrnosti, otisk souboru pro ověření integrity a podpisové certifikáty pro zajištění nepopíratelnosti. ČSSZ vyžaduje registraci v databázi elektronicky podávajících klientů.

### 4.3 e-Podání

Elektronické podání lze chápat jako elektronickou datovou strukturu, která může obsahovat 1 až 1500 tzv. formulářů. V jednom podání nelze kombinovat formuláře různých typů. Za formulář je považována ta část elektronického podání, která nese informace odpovídající vyplněnému jednotlivému papírovému formuláři (tiskopisu). Druhy e-Podání:

- evidenční listy (ELDP, RELDP),

- oznámení o nástupu do zaměstnání (ONZ, PRIHL),
- hlášení pracovní neschopnosti (HPN),
- příloha k žádosti o dávku (NEM PRI) a
- přehled OSVČ (OSVC PRE).

#### 4.3.1 Aplikační komunikační protokol

Aplikačním komunikačním protokolem (dále AKP) rozumíme sadu komunikačních pravidel na aplikační úrovni. Samostatný protokol je definován dvojicí **podání–odpověď**, kde podání je zpráva obsahující data elektronických formulářů či data požadavků. Na každé podání je právě jedna odpověď, ve které se nachází buď potvrzení o přijetí podání do zpracování, seznam chyb, které je nutné opravit, anebo data vyžádaná požadavkem.

Kvůli komplexnímu zpracování podání nelze garantovat synchronní odpovědi v rámci jediného HTTP spojení, odpovědi jsou dostupné v řádu jednotek minut. Aplikace implementující APK musí počítat s výpadkem spojení a musí proto umět opětovné spuštění. Odpověď na podání obsahuje informace o přijetí či zamítnutí. Některé typy podání (ONZ, ELDP, POSTP, PVPOJ) jsou zamítnuta jako celek tehdy, když se v kterémkoliv formuláři vyskytla chyba. Takové podání je nutné zaslat opravené celé znovu.

Pro ostatní (NEM PRI, HPN) je podání zamítnuto jako celek pouze v případě, že neobsahuje ani jeden zpracovatelný formulář. V opačném případě je podání označeno jako přijaté, ovšem zpracovány jsou jen formuláře nevykazující chyby (jedná se o tzv. „částečné přijetí podání“), pak je nutné vyhodnotit informace v odpovědi na úrovni jednotlivých formulářů a umožnit případnou opravu a opětovné zaslání.

#### 4.3.2 Komunikační kanály

Pro doručení e-Podání jsou vyhrazeny dva elektronické kanály:

- Veřejné rozhraní pro e-Podání (VREP).
- Informační systém datových schránek (ISDS).

Pro každý kanál jsou zvlášť popsány požadavky (registrace, potřebné identifikátory) pro autentizaci v rámci komunikačního kanálu, komunikační vzor a rozhraní. V rámci této práce bude detailně popsán jen kanál VREP.

#### 4.3.3 Veřejné rozhraní pro e-Podání

VREP je komunikační kanál s přímým rozhraním ČSSZ dostupný na základě registraci v systému ČSSZ. Zaslání podání přes VREP vyžaduje kvalifikovaný certifikát vydaný akreditovanou certifikační autoritou. K ověření oprávnění je nutné požádat o registraci na příslušné správě sociálního zabezpečení (předložení/zaslání vyplněného formuláře „*Oznámení o pověření k zajištění všech úkonů souvisejících s e-Podáním*“, respektive „*Sdělení doplňujících údajů*“ a kvalifikovaného certifikátu). Registrace na ČSSZ je jednorázová záležitost, před vypršením platnosti registrovaného certifikátu je možné registraci obnovit elektronicky, jinak je nutná opakovaná návštěva územního pracoviště. Autentizace je spolu s autorizací prováděna na úrovni jednotlivých zpráv. Na úrovni komunikačního kanálu není autentizace prováděna. VREP přímo podporuje aplikační komunikační protokol podání–odpověď. Protokol je implementován sadou zpráv uvedenou v tabulce 4.1.

Zasílá podávající	Zasílá VREP
Podání (submission request)	Doručenka (submission acknowledgement)
Dotaz (submission poll)	Odpověď (submission response)
Ukončení transakce (delete request)	Potvrzení (delete response)

Tabulka 4.1: Protokol podání–odpověď (chybné případy)

Podávající software zasílá podání ve zprávě „**submission request**“, na kterou VREP (pokud je podání správné) odpovídá (synchronně, v rámci HTTP spojení) doručenkou „**submission acknowledgement**“. Doručenka obsahuje **unikátní identifikátor podání** (VREP Correlation ID) a časovou značku (datum a čas převzetí opatřený podpisem technologické komponenty). Časová značka slouží primárně jako „podací lístek“ k prokázání zaslání podání v daném čase a sekundárně umožňuje ověřit identitu příjemce (VREP). Poté se musí podávající dotázat na stav vstupního zpracování, doporučený interval mezi podáním a tímto dotazem je hodnota navrátnějšího elementu „**PollInterval**“. V ostatních případech je doporučeným intervalem 5 minut. Dotaz na zpracování povinně obsahuje VREP unikátní identifikátor podání. VREP na dotaz odpovídá synchronně, jednou z následujících možností:

- **Odpovědí** (submission response), pokud už vstupní zpracování proběhlo a alespoň jeden formulář podání je v pořádku.
- **Chybou** (submission error), pokud už vstupní zpracování proběhlo a žádný formulář v podání nelze přijmout.
- **Doručenkou** (submission acknowledgement), v případě, že vstupní zpracování ještě nebylo dokončeno. V takovém případě je nutné dotaz opakovat.

Pokud VREP odpoví informací o přijetí (submission response) či zamítnutí podání (submission error), je vyžadováno, aby klientská aplikace zaslala požadavek na uzavření transakce (delete request), na který VREP odpoví potvrzením o uzavření transakce (delete response). Podávající aplikace musí korektně uzavírat transakce, v opačném případě bude považována za aplikaci porušující pravidla pro zasílání e-Podání. Pokud VREP na uzavření transakce odpoví doručenkou, je třeba opakovat požadavek po uplynutí určitého časového intervalu. V případě výskytu problému je vrácena chyba a sekvence zpráv běží jednou z cest uvedenou v tabulce 4.2.

Zasílá podávající	Zasílá VREP
Podání (submission request)	Chyba (submission error)
Podání (submission request) Dotaz (submission poll)	Doručenka (submission acknowledgement) Chyba (submission error)
Podání (submission request) Dotaz (submission poll) Ukončení transakce (submission error)	Doručenka (submission acknowledgement) Odpověď (submission response) Chyba (submission error)

Tabulka 4.2: Protokol podání–odpověď (ideální případ)

Chyba jako reakce na dotaz zpracování může být:

- Logická chyba AKP (dotaz na neexistující podání).
- Syntaktická chyba AKP (chybný formát, zpráva bez CorrelationID).
- Chyba na úrovni podávaných dat.

Chyby jako reakce na podání či jako reakce na požadavek na uzavření transakce:

- Chyba AKP (nesprávný formát zprávy, nenaplnění povinných polí).

Přístup k VREP je možný přes **rozhraní POX**<sup>1</sup> nebo přes **rozhraní webových služeb**. Obě rozhraní jsou na úrovni HTTP protokolu bezstavová. První zmíněné, rozhraní POX, které je přímo postavené nad základním protokolem HTTP, přímo podporuje komunikační vzor APK podání-odpověď. Podávající software posílá sestavenou zprávu v těle standardního HTTP požadavku (metodou POST). Taková zpráva obsahuje potřebné struktury podání či požadavek na vyzvednutí odpovědi. Reakce na zprávy jsou vráceny opět ve formátu XML synchronně v těle HTTP odpovědi v rámci jednoho HTTP spojení. Další komunikace probíhá dle uvedeného vzoru v rámci dalšího spojení.

Rozhraní webových služeb VREP rovněž přímo podporuje komunikační vzor implementující AKP podání-odpověď. Služba je nasazena na jednom koncovém bodě s anonymní autentizací, implementuje metody pro podání (**Submit**), dotaz na výsledek zpracování a požadavek (**Poll**) na uzavření transakce (**Dispose**). Jednotlivé zprávy komunikačního vzoru jsou předávány jako samostatná volání metod s daty a parametry předávanými v polích struktur definovaných v rámci kontraktu služby.

#### 4.3.4 Formáty zpráv

Jak bylo zmíněno již výše, podání je datová struktura obsahující data formulářů, jež odpovídají vyplněným papírovým tiskopisům. Struktura XML formátu se liší pro jednotlivé typy podání. Pro zabezpečený způsob přenášení těchto specifických dat je XML formát zprávy definován dle schématu **GovTalk**, tělo zprávy tvoří tzv. „**CSSZ Message**“. Pro všechna data jsou podporována kódování UTF-8, Windows 1250, ISO Latin 2, doporučeným je UTF-8. Úkoly zprávy „GovTalk Message“:

- Obsahuje informace o řízení AKP nezávisle na komunikačním kanálu.
- Umožňuje asynchronní volání odeslání podání.
- Používá se pro dotaz na výsledek zpracování nezávisle na komunikačním kanálu.
- Obsahuje elementy pro přenos vlastních dat podání („obálka“).
- Může nést specifické informace pro daný komunikační kanál (autentizací údaje pro PVS, požadovaná verze časové značky v doručence pro VREP aj.).

**CSSZ Message** („obálka ČSSZ“) je určena pro **podání** a **odpověď**. „CSSZ Message“ typu podání musí nést data podání a podpis. Data podání jsou definovaná pro každý typ podání (jednotlivé typy jsou definovány v [21]). Samotná data jsou zašifrována a výsledek musí být zakódován pro přenos v textové podobě. Před zpřístupněním dat neoprávněným uživatelům jsou data chráněna pomocí šifrování, které je založeno na standardu PKCS/CMS a využívá asymetrických klíčů X.509 certifikátů. ČSSZ na svých internetových stránkách vystavuje

<sup>1</sup>POX = „plain old XML“, XML over HTTP

šifrovací certifikát, pomocí kterého je třeba data zašifrovat, aby je bylo možné v systému pro zpracování e-Podání rozšifrovat. Výsledkem šifrování jsou binární data, která je nutno pro přenos v textové podobě zakódovat použitím algoritmu **Base64**.

Autentizace je založena na identifikaci na základě certifikátu, použitého k podpisu podání. Identifikace je prováděna vůči registrační databázi, stejně jako následná autorizace. Data podpisu jsou zakódována pro přenos v textové podobě. Podepisují se původní data podání před komprimací, šifrováním a kódováním pro přenos.

Při dotazu na výsledek není „CSSZ Message“ ve zprávě zahrnuta, jako identifikační parametr slouží „Correlation ID“. Informace o výsledku nese zpráva typu odpověď, která v části „CSSZ Message“ obsahuje strukturu s informacemi o výsledku zpracování. Data odpovědi (XML struktura) mohou být vložena přímo do „CSSZ Message“. Jsou dostupné i informace pro ověření integrity odpovědi. [18]

## 4.4 Návrh přímého podání tiskopisů

Návrh aplikace částečně vychází z [18]. Tento dokument obsahuje příklady metod určených pro komunikaci přes WS VREP v jazyce C#. Aplikace zajišťující přímé podávání z IS SAP nebyla minimálně na VUT dosud implementována, pro odeslání potřebných dat se používá externí aplikace.

Pro správné chování APK si aplikace musí uchovávat některé údaje. Hlavně kvůli potřebnému znovu navázání komunikace musí být někde ukládány získané unikátní identifikátory podání (v této práci označované jako ID korelace případně Correlation ID). Za tímto účelem je navržena tabulka 4.3.

Název pole	Popis pole
<b>ID podání «PK»</b>	Jedinečná identifikace podání.
<b>Typ podání «PK»</b>	Identifikace typu podání.
ID korelace	Identifikátor transakce, doplněn po zpracování synchronní zprávy (Correlation ID)
Status	Nabývá hodnot: odesláno, zpracovává se, potvrzeno, zamítnuto, chyba.
Datum odeslání	Datum poslední změny.
Čas odeslání	Čas poslední změny.
Odeslal	Systém automaticky doplní login přihlášeného uživatele.

Tabulka 4.3: Pomocná tabulka pro uložení odeslaných podání

### 4.4.1 Odeslání podání

Uživatel zvolí variantu pro odeslání podání a vybere XML soubor, který chce odeslat ČSSZ. Při odeslání se provedou následující kroky:

1. **Výběr odesílaného podání.** Systém zobrazí nabídku vygenerovaných XML souborů, ze kterých si uživatel zvolí jedno podání (XML soubor), které chce odeslat.

2. **Kontrola vstupního souboru.** Ověření cesty k vybranému XML souboru. Pokud soubor nelze otevřít, je informován o této skutečnosti uživatel a zpracování dál nepokračuje.
3. **Vytvoření digitálního podpisu.** Vybraný XML soubor je soukromým klíčem uživatele podepsán, výsledek je načten do SAP a převeden na Base64 kódování.
4. **Vytvoření zašifrované zprávy.** Vybraný XML soubor je veřejným klíčem ČSSZ zašifrován, výsledek je opět načten do SAP a převeden na Base64 kódování.
5. **Příprava na odeslání podání.** Systém zkontroluje, zda podání již nebylo odesláno. Pokud ne, je z dostupných údajů vytvořena GovTalk obálka (submission request) zapouzdřující „CSSZ Message“.
6. **Odeslání podání.** Odesílací aplikace naváže spojení s ČSSZ přes VREP. Odeslání probíhá na základě připravených dat protokolem HTTPS, metodou POST.
7. **Zpracování doručky (synchronní zpráva).** Z údajů uvedených v synchronní odpovědi HTTPS spojení je vytvořen záznam v databázi (4.4) – nastaví se stav na „Zpracovává se“ a doplní se ID korelace, přihlašovací jméno uživatele, datum a čas odeslání. Uživatel je o odeslání dat vhodným způsobem informován.

Název pole	Hodnota
ID podání	Název XML souboru.
Typ podání	Identifikace typu podání (př. ELDP).
ID korelace	Získané ID korelace.
Status	„Zpracovává se“
Datum odeslání	Aktuální datum.
Čas odeslání	Aktuální čas.
Odeslal	Přihlašovací jméno uživatele.

Tabulka 4.4: Záznam pomocné tabulky

#### 4.4.2 Vyzvednutí odpovědi

Uživatel vybere z nevyzvednutých podání to, které požaduje přijmout. Aplikace vytvoří odpovídající zprávu a přes VREP ji odešle. Postup vyzvednutí odpovědi:

1. **Výběr zpracovávaného podání.** Uživatel vybere jeden z XML souborů ve stavu „zpracovává se“.
2. **Příprava dotazu.** Aplikace vytvoří GovTalk dotaz (submission poll).
3. **Odeslání dotazu na odpověď (asynchronní zpráva).** Odesílací aplikace naváže spojení s ČSSZ přes VREP a odešle připravená data protokolem HTTPS, metodou POST.
4. **Zpracování odpovědi (synchronní zpráva).** V rámci HTTPS spojení systém zpracuje obdrženu odpověď a podle ní aktualizuje stav podání v databázi. V případě zamítnutí je příčina chyby uchována v databázi. Aplikace informuje uživatele o stavu zpracování.



## Kapitola 5

# Evidenční listy důchodového pojištění

Hlavním cílem této kapitoly je popsat návrh aplikace pro generování evidenčních listů důchodového pojištění (dále jen ELDP), proto je na jejím začátku čtenář seznámen se způsobem jejich vedení a předkládání. Jsou zmíněny základní údaje potřebné pro správné vyplnění ELDP. Popis návrhu obsahuje tabulku nutných infotypů, ze kterých jsou čerpány potřebné údaje. Samotný návrh se skládá z diagramu případů užití a popisu aplikací, které vykonávají jednotlivé případy. Jako podklady pro tuto kapitolu autor použil [19] [22].

### 5.1 Vedení a předkládání ELDP

Pro každou osobu účastnou na důchodovém pojištění vede zaměstnavatel ELDP vždy za jednotlivý kalendářní rok, respektive jeho část, pokud došlo k zahájení výdělečné činnosti nebo jejímu ukončení v průběhu kalendářního roku. Jestliže má zaměstnanec více souběžně výdělečných činností u téhož zaměstnavatele v průběhu kalendářního roku, vyhotovuje zaměstnavatel pro každou výdělečnou činnost ELDP samostatně. Pokud výdělečná činnost zaměstnance trvala k 31. prosinci, ELDP se vyplňuje po účetní závěrce, nejpozději však do **30. dubna**. Skončí-li účast před 31. prosincem, vyplňuje se do 1 měsíce po konečném vyúčtování příjmů. ELDP se vyhotovuje ve dvou stejnopisech, jeden si s podpisem zaměstnance nechává zaměstnavatel, druhý je určený pro zaměstnance.

### 5.2 Způsoby předkládání ELDP

Pro ELDP je zákonem předepsaný tiskopis, který vydává ČSSZ a je ho možné předkládat buď písemně do předepsaného tiskopisu, nebo v elektronické podobě. Předepsaný tiskopis má formu:

- **Tiskopis třídílný** je připraven pro vyplňování na psacím stroji.
- **Tiskopis jednodílný** slouží pro elektronické vyplňování pomocí volně dostupné aplikace, která je k dispozici na internetových stránkách ČSSZ. Aplikace zaměstnavateli umožňuje přímé vyplnění údajů do šablony a provedení tisku údajů do originálního tiskopisu ELDP. Kompatibilní tiskopis by měl být také výstupem formulářového rozhraní popsaného ve třetí kapitole této práce.

Elektronické podání:

- **Ve formě datové zprávy** (viz. předchozí kapitola)

### 5.3 Způsob vyplňování ELDP

Údaje v ELDP jsou rozděleny do následujících částí:

#### 1. Záhloví

Obsahuje **technický kód**, čtyřčíslí daného **kalendářního roku** a **typ ELDP** označující charakter údajů vykazovaných na ELDP.

#### 2. Identifikace pojištěnce

Pro identifikaci pojištěnce slouží platné **příjmení** v době provádění zápisu do ELDP, **křestní jméno**, **titul**, **datum narození**, devítimístné, eventuálně desetimístné **rodné číslo**, **adresa** (ulice, číslo domu, obec, pošta, PSČ, stát), **rodné příjmení** a **místo narození**.

#### 3. Průběh pojištění

Údaje o průběhu výdělečné činnosti v kalendářním roce se ve většině případů vyplňují do jednoho řádku. Existují však i situace, kde je nutné vyplnit více řádků, například u zaměstnanců, kteří dovršili důchodový věk a jejich pojištění trvá i nadále. Oddíl průběh pojištění v daném roce vyžaduje vyplnění třímístného kódu, pole „**MR**“ (malý rozsah) a byla-li výdělečná činnost zahájena před rokem zpracovávaným rokem (uváděny údaje pro rok 2012), je do pole „**Od**“ zadána hodnota 1. 1. 2012, obdobná situace je i u pole „**Do**“, pokud pracovní poměr pokračuje i v roce 2013, je uvedeno datum 31. 12. 2012. V opačných případech je použito právě ono datum zahájení, respektive ukončení pracovního poměru v rámci kalendářního roku. V kombinaci s třímístným kódem je možná i situace, kdy pole „**Od**“ – „**Do**“ není vyplněno. Do pole „**Dny**“ se uvádí počet kalendářních dnů, kdy trvá pojištění v daném kalendářním roce. Do celkového součtu dnů se nezahrnují měsíce, kdy pojištěnec nebyl účasten pojištění. Pole s označením **1** až **12** jsou považovány za kalendářní měsíce a umožňují znakem „**X**“ označit ty měsíce, které se nepovažují za dobu důchodového pojištění. Dále je nutné vyplnit pole „**Výlučné doby**“, „**Vyměřovací základ**“ a „**Doby odečt**“. Součet těchto hodnot je vypočítán do pole „**Celkem**“. Bližší informace o vyplňování, včetně příkladů, jsou uvedeny v metodické příručce [19].

#### 4. Identifikační údaje zaměstnavatele a podpisy

Poslední oddíl ELDP vyžaduje vyplnit **název zaměstnavatele**, pod kterým je zapsán v obchodním, živnostenském nebo jiném rejstříku, a **obec sídla**. Dále je vyžadováno **identifikační číslo** přidělené zaměstnavateli pro potřebu ČSÚ, **variabilní symbol**, údaj „**výdělečná činnost od**“, **datum vyhotovení ELDP**, **datum a podpis pojištěnce** a samozřejmě je **podpis a razítko zaměstnavatele**.

### 5.4 Návrh aplikace

Při návrhu aplikace je třeba vycházet z již existujícího řešení, které poskytuje přímo IS SAP a na upraveném řešení registru pojištěnců. Odesílání registru pojištěnců, neboli hlášení o nástupu do zaměstnání/skončení, ČSSZ je další zákonnou povinností zaměstnavatele. ČSSZ

páruje pracovní poměry zaměstnanců na základě kódů přidělených tímto registrem (rodné číslo – zaměstnavatel – kód). Standardní řešení přidělování kódů v SAP však v minulosti nefungovalo na VUT úplně správně. Bylo proto nahrazeno vlastním řešením, které s původní sadou programů pro ELDP nepracuje úplně správně.

Samotný proces zpracování ELDP by vzhledem k zažitému postupu a uložení dat nebylo dobré výrazně měnit. Návrh musí brát v potaz to, že výsledná aplikace bude nasazená u běžných koncových uživatelů a musí být proto dostatečně intuitivní a robustní. Proto i samotný proces zpracování musí být navržen tak, aby byl co nejjednodušší, a aby ho zvládl vykonávat uživatel jen po přečtení uživatelského manuálu. V návrhu musí být zohledněna i možnost úpravy zpracovaných dat, a to z jakéhokoliv důvodu. Samozřejmostí je kontrola veškerých vstupních polí, které uživatel zadává, a kontrola potřebných oprávnění pro spuštění reportu.

#### 5.4.1 Zdrojová data

Jak bylo zmíněno již ve druhé kapitole, data o zaměstnancích se ukládají do infotypů, které musí být řádně vyplněné před samotným zpracováním. Konkrétně se jedná o infotypy uvedené v tabulce 5.1.

Označení	Název
0000	Opatření
0002	Data k osobě
0006	Adresy
0015	Doplňková platba
0016	Součásti smlouvy
0021	Rodina
0041	Zadání data
0153	Pojistné CZ
2001	Nepřítomnosti
9008	Dohody o provedení práce VVŠ (zákaznický infotyp)

Tabulka 5.1: Zdrojové infotypy pro ELDP

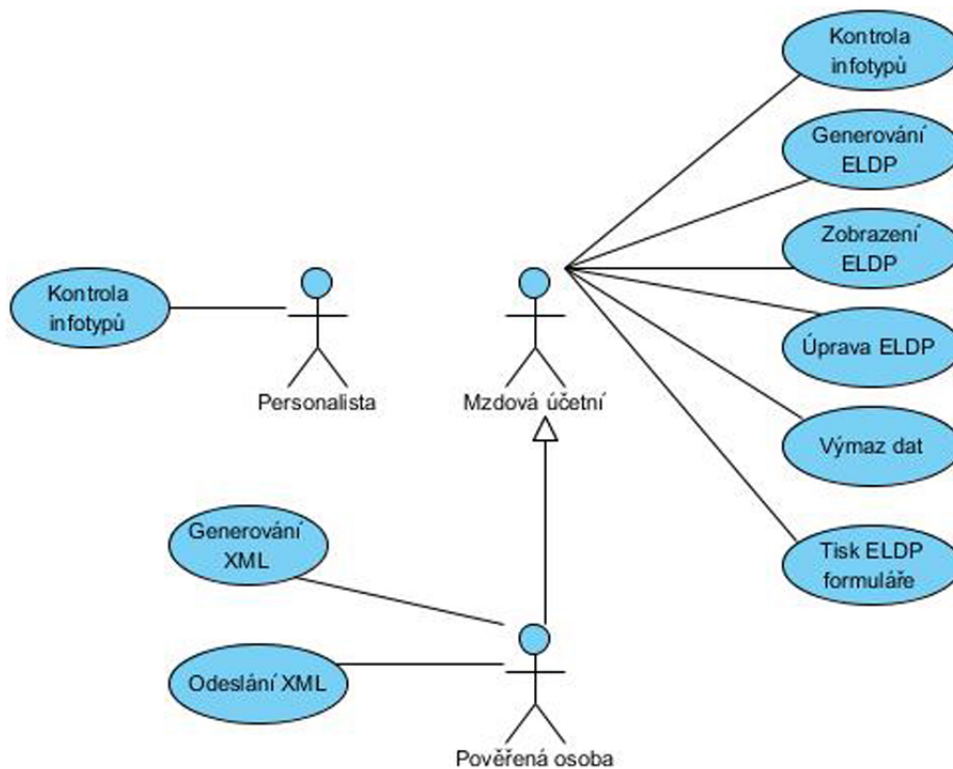
Ne všechny potřebné informace jsou ukládány v infotypech. Pro zadání údajů jsou využívány standardní transakce, které uchovávají data v těchto tabulkách:

Označení	Název
T5T51	Číselník pošt
T5T57	Číselník okresů ČSSZ
T536A	Adresy
T5TELDP	Adresář pro „cluster“ tabulku
HRPY_RGDIR	Adresář pro výsledky zúčtování

Tabulka 5.2: Zdrojové tabulky pro ELDP

### 5.4.2 Diagram případů použití

Na obrázku 5.1 je znázorněn diagram případů užití, jeho bližší popis je v následující podkapitole.



Obrázek 5.1: Diagram případů užití

### 5.4.3 Popis případů použití

1. **Kontrola zadaných údajů v odpovídajících infotypech.** Příslušný personalista, respektive mzdová účetní, provede kontrolu údajů v odpovídajících infotypech zaměstnanců, ke kterým zpracovává agendu.
2. **Zpracování údajů a jejich uložení.** Mzdová účetní s neomezeným oprávněním spustí generování ELDP pro všechna osobní čísla. Zaměstnanec může mít více pracovních poměrů spadajících pod několik mzdových účetní. Report postupně čte všechny pracovní poměry z logické databáze a zpracovává je na základě pravidel uvedených v [19]. Výsledkem zpracování jsou údaje uložené v databázi a textový výpis o průběhu zpracování. Textový výpis je vytisknut a slouží pro ruční kontrolu.
3. **Kontrolní tabulkový výpis ELDP.** V dalším kroku mzdová účetní ověří správnost zpracování spuštěním dalšího reportu, který ji formou tabulkového výpisu zobrazí vygenerované ELDP. Přehledově se zobrazí osobní číslo zaměstnance, rok a další informace o vygenerovaném ELDP. Současné řešení rovněž obsahuje funkcionalitu, která dovoluje odstranit chybně vygenerované ELDP. V případě, že nastane situace, kdy je nutná oprava již zpracovaných dat, je možné provést jejich změnu. Změna se

promítne pouze do vygenerovaných výsledků. Je vhodné umožnit i ruční vytvoření zcela nového (nevygenerovaného) ELDP. Uživatelské rozhraní bude mít podobu jako originální formulář ČSSZ.

4. **Tvorba XML souborů s daty ELDP.** Pověřená osoba vygeneruje XML soubory obsahující ELDP určené pro přenos skrze komunikační kanál. XML soubor bude uložen na lokální disk přihlášeného uživatele. Formát výstupní struktury je uveden na [www stránkách ČSSZ \[20\]](#). XML soubor bude podepsán soukromým klíčem uživatele a zašifrován veřejným klíčem ČSSZ.
5. **Odeslání XML souborů podle pravidel ČSSZ a zpracování odpovědi.** Pomocí aplikace navržené v předchozí kapitole bude vygenerovaný XML soubor odeslán ČSSZ a obdržená odpověď bude vhodně interpretována uživateli.
6. **Odeslání upravených dat.** V případě zamítnutí podání, o čemž program uživatele informuje, oznámí tuto skutečnost pověřená osoba mzdové účetní, nebo sama provede potřebné kroky (úprava infotypu, vymazání daného ELDP a jeho nové vygenerování, případně jeho oprava) tak, aby bylo možné podání znovu odeslat, to znamená nové vygenerování XML souboru.
7. **Tisk ELDP.** Zaměstnavatel je povinen vydat pojištěnci stejnopis nejpozději v den, kdy předkládá ELDP orgánu sociálního pojištění. To ale vzhledem k počtu zaměstnanců na VUT není možné, proto se podání zaměstnancům vždy předává až po odeslání. Pokud to technické vybavení povolí, bude program obsluhující tisk realizovat výstup pomocí Adobe Forms. Výsledný tiskopis bude shodný s originálem vydávaným ČSSZ. V případě, že technické prostředky daný tisk neumožní, bude nutné zachovat stávající textový formát, popřípadě nastavit výstup tak, aby šel vytisknout do předpřipravených originálních formulářů. Nejpravděpodobnější varianta je však využití stávajícího řešení, tedy textový výpis na prázdný papír. Předpokládá se použití rychlých jehličkových tiskáren.

#### 5.4.4 Hlavní aplikace

Navrhovaná aplikace nabídne ve svém uživatelském rozhraní možnost výběru jedné z následujících operací:

- generování údajů,
- zobrazení údajů,
- generování XML souborů,
- odeslání souboru a
- hromadný tisk.

Ve všech případech je nejprve nutné zajistit, aby zpracování prováděla jen osoba k tomu určená – zkontrolují se oprávnění. Doporučeným postupem je využít logickou databázi (dále jen LD) PNP [14]. LD automaticky kontroluje oprávnění uživatele pro přístup k datům, je proto výhodné nejdříve projít v cyklu všechna osobní čísla a získat požadovaná data zaměstnanců pro následné zpracování.

### 5.4.5 Generování údajů

Pro správné párování pracovních poměrů s ČSSZ je třeba zajistit, aby aplikace pro generování údajů využívala stejné značení pracovních poměrů (první znak třímístného kódu v průběhu pojištění) jako registr pojištěnců, který je rovněž pro ČSSZ generován. Pro získání správných označení byly pro VUT dodělány funkční moduly ZHR\_VUT\_MAIN\_PERNR a ZHR\_DRVCI\_VUT. Jejich postupným použitím se získají všechna osobní čísla zaměstnance, jednotlivé pracovní poměry, včetně označení, data platnosti a informace o tom, zda se jedná o práci malého rozsahu. Pokud se jedná o dohodu o provedení práce, obsahuje struktura i číslo smlouvy. Další zpracování bude brát v úvahu jen ty pracovními poměry, které jsou platné v požadovaném období a bylo z nich odváděno sociální pojištění. Získané záznamy jsou určené pro část formuláře „průběh pojištění“. Ve většině případů bude možné doplnit přímo do výsledného formuláře první znak pole **kód**, určit **práci malého rozsahu** a data „**Od**“ a „**Do**“.

Poté je možné začít vyplňovat **kolonky s číselným označením měsíce**, znak „X“ je v poli obsažen, pokud zaměstnanec nesplnil následující podmínky:

- Nedosáhl započítatelného vyměřovacího základu – výsledky zúčtování
- Vykonávání činnosti zakládající účast na pojištění alespoň v jednom dni – kontrola nepřítomností z výsledků zúčtování

**Vyměřovací základ** se vypočítá jako součet všech příjmů v daném období. Informace je dostupná ve výsledcích zúčtování. V případě, že se jedná o práci malého rozsahu, jsou do vyměřovacího základu započítány jen hodnoty splňující podmínku započítatelného příjmu, tj. měsíční mzda vyšší než 2500 Kč. Pole „vyměřovací základ“ v takovém případě bude navíc obsahovat až počáteční datum měsíce, v němž byla daná podmínka poprvé splněna. Práce malého rozsahu má dopad i na kolonky s označením měsíce – pokud není v daném měsíci splněna výše zmíněná podmínka, je měsíc označen znakem „X“. Do „**Vyloučené doby**“ se počítají všechny dny dočasné pracovní neschopnosti zadané v IT 2001.

Dovršil-li zaměstnanec v daném roce důchodový věk (údaj dostupný v IT 0041) budou stávající záznamy rozšířeny způsobem uvedeným v tabulce 5.3 (znak „X“ reprezentuje první znak trojmístného kódu).

Kód	Od	Do
X	Od	Datum důchodový věk - 1
XD	Datum důchodový věk	Do

Tabulka 5.3: Úprava záznamů pro důchodový věk

U každého poměru zaměstnance, který dovršil důchodový věk v předchozích letech, bude znak „D“ doplněn také. Pole „**Doby odečt.**“ (dále DO) bude vyplněno následovně:

$$DO = Do - DDV + 1$$

V případě, že nebyl dovršen důchodový věk, zůstane DO nevyplněné (pro výpočet rovnou nule). Na základě výše uvedených informací lze vypočítat počet dní (PDNM udává počet dní v nepřítomných měsících):

$$Dny = (Do - Od + 1) - PDNM - DO$$

Zbylé znaky třímístného kódu budou zatím pro jednoduchost obsahovat „+“. Možné situace budou diskutovány až v implementační části na základě zpětné vazby budoucích uživatelů.

K jednotlivým záznamům o průběhu pojištění získaným předchozím zpracováním budou přiřazeny osobní údaje (tabulka 5.4), doplňované ve formuláři do části identifikace pojištěnce.

Jméno pole	Infotyp
jméno, příjmení, rodné příjmení, místo, stát, datum narození, tituly, rodné číslo	IT 0002
místo, místní část, ulice, PSČ, pošta	IT 0006

Tabulka 5.4: Údaje potřebné pro identifikaci pojištěnce

Poslední část formuláře bude doplněna na základě dat uvedených v uživatelském rozhraní. Jedinou výjimku tvoří pole „výdělečná činnost od“, do kterého bude vloženo datum prvního pracovního dne zaměstnance zadané v IT 0041.

Z takto získaných dat je nyní možné naplnit vestavěnou strukturu T5TELDP2004, která představuje konečnou podobu ELDP formuláře. Trvá-li výdělečná činnost i po 31. 12. je typ formuláře ELDP nastaven jako „01“. Pro všechny získané záznamy, u nichž výdělečná činnost končí ve zpracovávaném roce, je zvolen typ „02“. Další typy podání nejsou zatím uvažovány. Z důvodu omezeného počtu polí (jsou k dispozici 3 řádky) pro zápis průběhů pojištění musí aplikace patřičným způsobem zajistit rozdělení záznamů a dopočítat pole „Celkem“.

Zpracovaná data se ukládají do „cluster“ tabulky. Práce s touto speciální tabulkou probíhá skrze předpřipravená makra nad strukturou složenou z T5TELDP2004, třídícího klíče a dalších informací o generování. Třídícím klíčem je myšlena kombinace 40 znaků sestavených z osobního čísla zaměstnance, roku, data od, data do, stavu a strany podání (1 strana = 3 průběhy). O výsledku zpracování je uživatel generován formou textového výpisu.

#### 5.4.6 Zobrazení údajů

Vygenerované údaje bude možné zkontrolovat v grafické tabulce, která umožní i případnou editaci. V prostředí SAP se pro tento účel využívá ALV tabulka (popsána například v [3]). Díky ní může uživatel mimo jiné načtená data libovolně řadit a filtrovat, což je velice výhodné pro kontrolu vyplnění veškerých požadovaných polí. Důležitou vlastností je také to, že ALV tabulka dovoluje uživateli vybrat požadovaný řádek. S ním lze pak provádět další operace, jako je tisk do PDF souboru či zobrazení obrazovky s formulářem pro snadnější doplnění dat. Jedná se o jeden z hlavních požadavků v navrhované aplikaci.

#### 5.4.7 Tisk formulářů

Ve výsledné aplikaci budou realizovatelné 3 typy výstupu pro tisk:

- **Informativní**, dostupný po vygenerování a uložení dat. Standardní formulář rozšířený o nákladové středisko a osobní číslo zaměstnance.
- **PDF**, doplnění údajů do originálního formuláře.
- **Hromadný tisk**, po odeslání a potvrzení výsledků. Dle zákonné povinnosti se předává všem zaměstnancům.

## Kapitola 6

# Implementace

Šestá kapitola se věnuje samotné realizaci dříve navržených aplikací. Výsledkem řešení je jediný program spojující obě dvě předešlé aplikace. Kapitola postupně popisuje implementaci celého procesu zpracování ELDP. Po krátkém úvodu, který seznámí čtenáře s konvencemi používanými ve zdrojovém kódu, následuje část popisující uživatelské obrazovky. Text práce se dále věnuje samotnému generování evidenčních listů, jejich kontrolnímu zobrazení a uložení formou XML souboru. Následující část popisuje implementaci, která zajišťuje odesílání dat ČSSZ, zpracování odpovědi a její následnou reprezentaci v SAP. V závěru kapitoly je nastíněn princip hromadného tisku a popis testování.

### 6.1 Použité konvence

Kritéria pro volbu programovacího jazyka jsou již předem daná. Programy v SAP jsou z převážné většiny implementovány v programovacím jazyce ABAP. Pro přehlednost zdrojového kódu jsou stanoveny konvence (například pojmenování proměnných, tvorba tabulek či příprava dat před zpracováním), které byly striktně dodržovány v rámci celé aplikace.

Samotný program využívá velkého počtu interních tabulek, které v ABAP nelze ideálně předávat v parametrech funkce, proto jsou interní tabulky nastaveny jako globální (tabulky lze v parametrech předávat, ale bohužel nelze efektivně filtrovat data v cyklu LOOP).

Pro uchování stavu, typicky pravda/nepravda, je používána jednoznaková proměnná typu CHAR. Obsahuje-li hodnota libovolný znak, je brána jako logická pravda, je-li hodnota prázdná, jedná se o logickou nepravdu. Stejným způsobem se chová i standardní grafický výběrový prvek, jehož hodnota je v případě zaškrtnutí nastavena na „X“. Pro určení zvolené hodnoty je využíváno dotazu, zda je hodnota prázdná.

Proměnné jsou pojmenovávány českými názvy a taktéž komentáře jsou v češtině. Celkově je zdrojový kód hojně komentován. Funkce vždy obsahuje hlavičku, která popisuje význam funkce a vstupní, respektive výstupní parametry. V rámci těla se vyskytují komentáře u složitějších příkazů.

### 6.2 Hlavní obrazovka

Na hlavní obrazovce vynikají standardní výběrová pole PNP databáze (obrázek 6.1). Pro úplnost jsou v tabulce 6.1 uvedeny názvy v práci používaných PNP polí, přes které je možné získat, respektive ovlivnit data zpracovávaná logickou databází. V každém programu využívající PNP mají globální dostupnost.



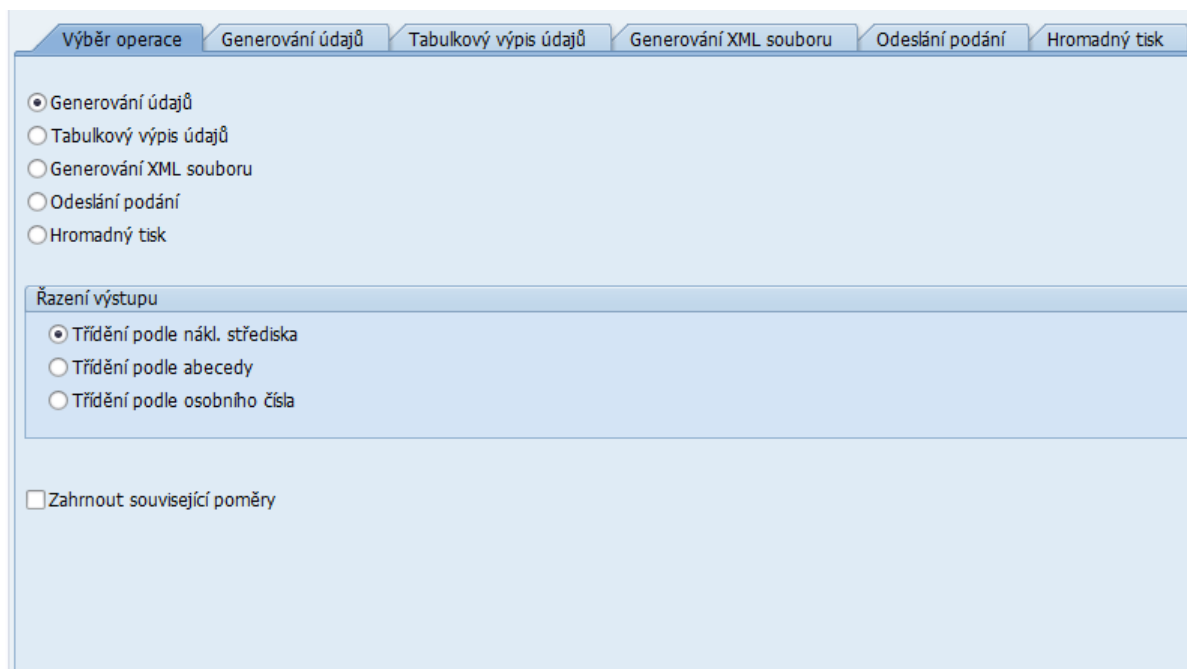
Obrázek 6.1: Grafické rozhraní logické databáze PNP

Název pole	Interní tabulka	Význam
Období výběru dat od	PNPBEGDA	Počáteční datum, hodnota „od“ pro proces zpracování.
Období výběru dat do	PNPENDDA	Koncové datum, hodnotu „do“ pro proces zpracování.
Období výběru osob od	PNPBEGPS	Počáteční datum, pro výběr osobní čísla.
Období výběru osob do	PNPENDPS	Koncové datum, pro výběr osobní čísla.
Osobní číslo	PNPPERNR	Interní tabulka sjednocující osobní čísla.
Status zaměstnání	PNPPSTAT2	Stav osobního čísla (aktivní, vystoupil, mimo evidenční stav).
Skupina pracovníků	PNPPERSG	Pracovní právní vztahy (1 pracovní poměr, 4 mimo evidenční stav, 7 DPČ, 8 DPP).

Tabulka 6.1: Proměnné logické databáze PNP

Obrazovka je dále rozšířena o možnosti výběru požadované operace, setřídění výstupu a o menu doplňující vstupní parametry jednotlivých operací (obrázek 6.2). V SAP bývá zvykem umožnit uživateli testovací běh programu (data jsou vygenerována, ale neukládají se). Tato možnost je zde rovněž zohledněna. Definice úvodní obrazovky včetně menu a záložek je popsána v modulu ZHR\_DP\_ELDP\_GENEROVANI\_SCR (viz. dále tabulka 6.3). Celkem je vytvořeno 6 záložek. V první záložce jsou definovány nabízené operace, zbylé záložky obsahují parametry, které dovolují blíže ovlivnit vybranou operaci. Výhradně jsou tvořeny

kombinací vstupních a zaškrtačacích polí. Vstupní textové pole je v některých případech rozšířeno o možnost výběru dat z přednastavené nabídky. To je vhodné například při výběru místní pobočky ČSSZ. Pro VUT by stačilo pole předdefinovat na hodnotu „**772 – Brno město**“. Minimálně však před vytvořením varianty s před vyplněnými poli je vhodné, aby funkcionality byla implementována. Podobná je situace i při výběru složky pro ukládání vygenerovaných XML souborů. I zde je nastavena reakce na vstupní pole. Pro otevření dialogového okna s možností výběru požadované složky se volá vestavěná statická metoda. Výsledkem volání je otevření dialogového okna, které dovolí vybrat cestu k požadované složce.



Obrázek 6.2: Grafické menu aplikace

Součástí definice úvodní obrazovky jsou i kontroly zadaných dat. Kontroluje se platnost období výběru dat a období výběru osob v PNP obrazovce. Za platné údaje jsou považovány situace splňující následně uvedené body:

- Všechny hodnoty jsou zadané.
- Rok u obou zadaných dat je shodný.
- Zpracovávaný rok je stejný nebo větší než rok 2012.

Kromě výše uvedených kontrol je ověřován maximální počet formulářů v XML souboru, který dle pravidel ČSSZ nesmí přesáhnout 1500 podání.

Po výběru požadované operace program připraví data a přesune uživatele do nové obrazovky (tabulka 6.2). Každá obrazovka je jednoznačně identifikovaná čtyřmístným číslem.

Číslo obrazovky	Popis
1000	Hlavní obrazovka (automaticky generovaná).
0100	Tabulkové zobrazení vygenerovaných evidenčních listů.
0200	Tabulkové zobrazení dat připravených k odeslání.
0201	Odesílací nabídka.
0202	Zobrazení chyb u zamítnutého podání.
2009	ELDP formulář (převzato z původního řešení).

Tabulka 6.2: Označení obrazovek

### 6.3 Zpracování evidenčních listů

Hlavní program, pojmenovaný ZHR\_DP\_ELDP\_GENEROVANI, dovoluje uživateli vybrat jednu z několika nabízených operací. Jednotlivé operace jsou pro snadnou orientaci rozloženy do několika modulů, pojmenovaných dle běžných konvencí SAP (tabulka 6.3).

Název modulu	Význam modulu
ZHR_DP_ELDP_GENEROVANI_TOP	Definice typů, globálních proměnných a konstant.
ZHR_DP_ELDP_GENEROVANI_SCR	Definice úvodní obrazovky.
ZHR_DP_ELDP_GENEROVANI_F01	Podprogramy pro generování dat.
ZHR_DP_ELDP_GENEROVANI_F02	Definice pomocných tříd.
ZHR_DP_ELDP_GENEROVANI_F03	Podprogramy pro tabulkové zobrazení.
ZHR_DP_ELDP_GENEROVANI_F04	Podprogramy pro pomocný formulář.
ZHR_DP_ELDP_GENEROVANI_F05	Pomocné podprogramy pro práci s „cluster“ tabulkou.
ZHR_DP_ELDP_GENEROVANI_F06	Podprogramy pro generování XML souborů.
ZHR_DP_ELDP_GENEROVANI_F07	Podprogramy pro odesílání dat.
ZHR_DP_ELDP_GENEROVANI_F09	
ZHR_DP_ELDP_GENEROVANI_F10	
ZHR_DP_ELDP_GENEROVANI_F08	Podprogramy pro hromadný tisk.

Tabulka 6.3: Popis modulů

Program je nastaven tak, aby používal logickou databázi PNP – personální kmenová data. Tato volba se provádí ve vlastnostech programu. Díky tomu lze pohodlně příkazem GET\_PERNR procházet zadaná osobní čísla a programátor se tak nemusí starat o ověřování autorizace uživatele. Běžně se před cyklus, který jednotlivá osobní čísla načítá, doplňuje sekce volána v programu jen jednou. Typicky slouží pro inicializaci proměnných. Pro úplnost je dobré uvést, že GET\_PERNR je standardní volání, které je využíváno u drtivé většiny aplikací v modulu HR.

Potřebná oprávnění pro přístup k zadanému osobnímu číslu, respektive k jeho údajům, jsou kontrolována automaticky. Nemá-li uživatel na osobní číslo oprávnění, nedostane se osobní číslo vůbec do zpracování. Další výhodou je snadný přístup k těmto datům. V programu se definují požadované infotypy, ze kterých mají být čerpány informace. V případě této práce vypadá příkaz následovně:

INFOTYPES : 0000, 0001, 0002, 0006, 0016, 0021, 0041, 0153, 2001, 9008.

To znamená, že jsou načítány informace o organizačním přiřazení, osobní informace

zaměstnanec, adresa zaměstnance, příjmy, rodina, zadaná data, pojištění, nepřítomnosti a případné dohody o provedení práce. V podstatě jde o údaje z tabulky 5.1. V praxi to vypadá následovně: všechny informace o daném infotypu jsou dostupné v interní tabulce začínající písmenem „P“ (např. P0000). V hlavičce získané tabulky je vždy poslední záznam (dle zadaného intervalu na úvodní výběrové obrazovce). Pro získání hodnot platných pro dřívější období lze použít příkaz LOOP.

Většina implementovaných operací je rozdělena na 2 části: získání potřebných dat a jejich zpracování. Většinou proto program, nejdříve pro každé osobní číslo, volá funkce typu „získat data“ a poté, už se všemi shromážděnými informacemi, „zpracovat data“. Pro většinu prováděných operací umožňuje takový způsob řešení využít společnou interní tabulku, která je vhodným způsobem naplněna (liší se podle prováděné operace). Kromě operace generování evidenčních listů jsou vždy shromážděny všechny údaje a až poté jsou zpracovány jako celek. Při generování se celá operace vždy provádí jako celek pro jedno osobní číslo. Až po uložení do databáze jsou data za účelem seřazení shromažďována v interní tabulce a poté v požadovaném pořadí vypsána na obrazovku.

### 6.3.1 Definice interních tabulek

Interní tabulky jsou obdobou klasických databázových tabulek. Rozdílem je, že vznikají až v rámci programu a po celý jeho běh jsou uloženy v hlavní paměti. Po ukončení programu je obsah interní tabulky vymazán (stejně jako běžná proměnná). Interní tabulku lze vytvořit na základě struktury definované v ABAP Dictionary nebo přímo ve zdrojovém kódu. Případně ji programátor může definovat jako kombinaci obou dvou přístupů. V programu jsou využívány všechny tři způsoby. Interní tabulky společné pro všechny moduly jsou definovány v modulu ZHR\_DP\_ELDP\_GENEROVANI\_TOP. Každý modul zvlášť obsahuje definici vlastních struktur na svém začátku.

Následuje výčet nejdůležitějších struktur. První struktura, určená pro všechny moduly, se skládá z ELDP formuláře a nákladového střediska. Je reprezentována tabulkou GT\_ELDP. Nebýt požadavku řazení podle nákladového střediska, mohl být použit přímo typ popisující formulář. Pro účely kontrolního tisku vygenerovaných podání je ještě k dispozici tabulka stejného typu GT\_ELDP\_TISK. Dále je zde definována struktura, která udržuje potřebné informace o zaměstnavateli (název, adresa, variabilní symbol apod.).

Na začátku modulu ZHR\_DP\_ELDP\_GENEROVANI\_F01 je kompletně ve zdrojovém kódu popsán typ určený pro načítání sekvenčních čísel nutných pro přístup k výsledkům zúčtování. Dále je definován typ s informacemi o průběhu pojištění reprezentovaný tabulkou GT\_PRUBEH. Obsahuje tedy třímístný kód pracovního poměru, datum od, datum do, počet dní, třináctiznakový řetězec popisující dobu nepřítomnosti, dny vyloučené doby, vyměřovací základ, dny doby odečtené a další pomocné proměnné. Výsledky zúčtování jsou po měsících ukládány do pomocné tabulky GT\_MESIC.

Formát zobrazovaných informací v grafické tabulce se liší oproti „cluster“ tabulce. Pro tento účel byla v ABAP Dictionary založena struktura kopírující původní formulář. U některých prvků jsou ale kvůli zobrazení změněny datové typy (nulová hodnota typu číslo se v tabulce nezobrazovala a to by mohlo mást uživatele). Pro ovlivnění vzhledu tabulky musí být struktura rozšířena o další prvky (barva buněk, styl zobrazení). Interní tabulka určená pro zobrazení podání je pojmenovaná GT\_ALV\_ELDP. Podobným způsobem jsou definovány všechny tabulkové výstupy. Za zmínku ještě stojí definice výstupního XML formátu (ABAP Dictionary, struktura ZHR\_DP\_ELDP\_XML), která obsahuje větší počet polí než klasický formulář. Tvoří tabulku GT\_ELDP\_XML.

### 6.3.2 Inicializace údajů

Ještě před načítáním osobních čísel jsou inicializovány základní potřebné údaje. Nejdříve jsou načteny informace o zaměstnavateli. V SAP existuje funkční modul, jehož provedení přímo vrací požadované údaje o zaměstnavateli. Proměnné jsou uloženy do globální struktury. ČSSZ přiděluje každé organizaci jednoznačně identifikující variabilní symbol, který určuje organizaci. Tento důležitý údaj je uložen v tabulce „podklady pro převodní příkazy za zaměstnavatele“, pod klíčem „SP“. Získaná hodnota je umístěna opět do uvedené struktury. Mezi další inicializované tabulky patří interní tabulka obsahující informace z číselníku pošt. Údaje z ní jsou použity pro doplnění povinného pole pošta, které se v do IT 0006 nezadává (zadává se pouze PSC na jehož základě se údaje párují).

Mezi inicializovanými údaji je i kalendářní rok, za který se formulář podává, byla vyhrazena globální proměnná, do které je přiřazen rok zadaný v PNPENDDA. Přímou v kódu jsou definovány používané konstanty uchovávané mzdivé druhy. Jako mzdivý druh si čtenář může představit jednoznačný klíč identifikující druh platby. Pro potřeby ELDP jsou významné dva:

- /311 – vyměřovací základ sociálního pojištění v roce 2012 do částky 1 206 576 Kč,
- /31S – vyměřovací základ sociálního pojištění v roce 2012 nad 1 206 576 Kč.

Stejným způsobem jsou vymezeny i konstanty pro zisk nepřítomností zaměstnance. Nepřítomnosti použité pro výpočet vyloučené doby a doby odečtené jsou uvedeny v tabulce 6.4. Nepřítomnosti využívané pro určení měsíců vyloučených z pojištění jsou v tabulce 6.5.

Druh nepřítomnosti	Popis
0200	Nemoc placená nová.
0201	Nemoc placená pokračující.
0205	Nemoc neplacená.
0210	Pracovní úraz nový.
0211	Pracovní úraz pokračující.
0220	Nepracovní úraz placený.
0221	Nepracovní úraz pokračující.
0230	Ošetření člena rodiny (OČR) – neosamělý nové.
0231	OČR – neosamělý pokračující.
0240	OČR – osamělý nové.
0241	OČR – osamělý pokračující.

Tabulka 6.4: Nepřítomnosti – vyloučené doby

Druh nepřítomnosti	Popis
0306	Neplacené volno.
0530	Rodičovský příspěvek.

Tabulka 6.5: Nepřítomnosti – vyloučené měsíce

### 6.3.3 Získ osobních čísel

Jak bylo zmíněno dříve, osobní čísla zadaná na úvodní obrazovce jsou postupně načítána z logické databáze. V jednom programu bohužel nelze vícekrát volání GET PERNR použít. V zásadě by to nevadilo, kdyby neměli na mzdové účtárně požadavek na možnost zpracování všech souvisejících osobních čísel zaměstnance (ovlivnitelné parametrem na vstupní obrazovce). Za tímto účelem byl vytvořen nový program, který čte zadaná osobní čísla a dohledává čísla související. Na VUT je k dispozici funkční modul vracející požadovaná související osobní čísla. Získaná osobní čísla jsou vypisována na obrazovku nebo jsou postupně ukládána do interní paměti, která je na konci programu uložena do sdílené paměti.

SAP podporuje, a jeho programy často takovou možnost i využívají, spuštění jiného programu s vhodnými parametry za běhu. Příkazem SUBMIT se provede zadaný program a na jeho konci dojde k předání výsledků zpět (uložením do paměti). Parametrem je myšleno jak vstupní pole úvodní obrazovky, tak i skryté pole. Dokonce lze vyvolat celou variantu. Program bývá často rozšířen o skrytý parametr, který uživatel při běžném spuštění, například přes kód transakce, není schopen ovlivnit. Nevýhodou je, že datovou strukturu ukládanou do paměti nelze definovat za běhu programu, musí být známá už při implementaci.

Pro předání načtených osobních čísel pomocného reportu byla zvolena struktura shodná s PNPPERNR (viz. tabulka 6.1). Uložení do sdílené paměti obstarává standardní příkaz jazyka ABAP, rovněž tak čtení ze sdílené paměti. Pro identifikaci paměťového prostoru jsou oba příkazy povinně rozšířeny o identifikátor. Je zřejmé, že obě varianty musí dále obsahovat i lokálně definovanou strukturu pro uložení, respektive čtení předávaných dat. V hlavním programu jde přímo o strukturu PNPPERNR.

### 6.3.4 Generování dat

Na základě pravidel ČSSZ bylo implementováno řešení vyhovující potřebám VUT. Nezhledňuje proto všechny možné situace, které mohou dle ČSSZ nastat. Generování probíhá zvlášť pro každé osobní číslo. Proces je rozdělen na 4 části:

- Načtení pracovních poměrů z registru pojištěnců a jejich úprava pro potřeby ELDP.
- Vyplnění ELDP formuláře dle pravidel ČSSZ.
- Uložení formuláře do „cluster“ tabulky.
- Tisk vygenerovaných dat pro ruční kontrolu údajů.

Ještě před spuštěním generování je uživatel dotázán, zda chce opravdu generovat data, pokud ano, je z hlavního programu spuštěn podprogram pro generování ELDP. Jak bylo již v návrhu zmíněno, pro získání údajů o pracovních poměrech zaměstnance a údajů z registru pojištěnců, jsou k dispozici funkční moduly ZHR\_VUT\_MAIN\_PERNR a ZHR\_DRVCI\_VUT. První z nich vrací údaje o všech osobních číslech zaměstnance tak, že projde infotypy, ve kterých je možné zadat pracovní poměr zaměstnance (infotypy 0000, 0008, 0016 a 9008). Výstupem je struktura s informacemi o pracovních poměrech. Obsahuje důležité informace, jako například hlavní osobní číslo a osobní číslo poměru, první den začátku pracovního poměru, poslední den pracovního poměru, první den pracovního poměru v rámci celého VUT (může být, a často je, údaj vztahující se k poměru vedeném na jiném osobním čísle) a číslo smlouvy.

Získaná struktura je povinným vstupem pro funkční modul ZHR\_DRVCI\_VUT, který každému z předaných poměrů doplní detaily. Hlavně ale správně dosadí kód vyžadovaný ČSSZ.

Funkční modul je poměrně složitý a jeho popis přesahuje rozsah této práce. Důležité je, že vrací interní tabulku, která kromě jednoznačového označení pracovního poměru obsahuje číslo smlouvy u DPP a informaci, zda je to práce malého rozsahu.

Jednotlivé řádky nově získané tabulky již lze chápat jako řádky ve výsledném formuláři, proto jsou vybírány jen ty, jež se váží ke zpracovávanému období. Postup zpracování získaných řádků, ve kterém se postupně plní připravená struktura je následující:

1. Pokud je na úvodní obrazovce vybrán parametr „Zahrnout jen ukončené“, zkontroluje se, zda v IT 0000 nebyl ve zpracovávaném období pracovní poměr zaměstnance ukončen. Je-li pracovní poměr v daném období aktivní, je v tomto případě zpracováváný průběh vyřazen.
2. Získání data nástupu do zaměstnání z infotypu 0041 (druh data „Nástup do zaměstnání“). Bohužel byl tento infotyp v minulosti omylem smazán a jen částečně obnoven. Při testování se proto vyskytoval problém, že u navazujících pracovních poměrů je zadán pouze nový datum nástupu do zaměstnání a ten původní (vyžadovaný ČSSZ) chybí. S tímto ale nelze z programátorského hlediska nic dělat a při kontrole mzdovou účetní musí být údaj dodatečně ručně opraven.
3. Nastavení příznaku o odeslání ELDP v minulých letech. Po domluvě na mzdové účtárně algoritmus nastaví příznak v případě, že datum nástupu do zaměstnání je dříve než začátek zpracovávaného intervalu. Naneštěstí může však nastat situace, že údaj o nástupu do zaměstnání není možné z výše uvedeného důvodu zjistit. V takovém případě se program podívá do uložených ELDP z minulých let a pokud nalezne pro osobní číslo uložené podání, je příznak nastaven. Program nevyhledává ihned přímo v databázi, protože se mohlo podávat papírové podání, které v SAP není vedené. Nastavení příznaku má vliv na výpočet práce malého rozsahu a DPP.
4. Upravení počátečního a koncového data na zpracováváný interval. Datum, nacházející se ve zpracovávaném intervalu, se neupravuje.
5. Případné nastavení příznaku o ukončení pracovního poměru ve zpracovávaném období.
6. Získání dnů, ve kterých se zaměstnanci narodily děti z IT 0021. Informace je nutná pro výpočet nepřítomností u mateřské dovolené.
7. Nastavení příznaku o narození dítěte v rámci mateřské dovolené (v rámci zpracovávaného intervalu, ale i před ním).
8. Případné nastavení příznaku o úmrtí zaměstnance. Informace uložena v IT 0000. Má vliv na typ podávaného formuláře.
9. Vložení naplněné struktury s informacemi o průběhu pojištění do interní tabulky GT\_PRUBEH.
10. Získání informací nutných pro doplnění hlavičky formuláře. Programová struktura reprezentující ELDP byla vzhledem ke kompatibilitě s „cluster“ tabulkou převzata z původního řešení SAP. Popis v této fázi načítaných údajů je uveden v tabulce 6.6.
11. Doplnění nákladového střediska a uložení získané struktury do interní tabulky GT\_ELDP.

Název v ELDP formuláři	Zdroj informace
Osobní číslo zaměstnance	Aktuálně zpracovávané os. číslo.
Za rok	Rok ze zpracovávaného intervalu.
Datum vyhotovení ELDP	Aktuální datum.
Křestní jméno	Interní tabulka P0002.
Titul	Interní tabulka P0002.
Příjmení (poslední)	Interní tabulka P0002.
Rodné datum narození	Interní tabulka P0002, pokud není zadáno, bere se příjmení.
Místo narození	Interní tabulka P0002.
Stát narození	Interní tabulka P0002.
Rodné číslo	Interní tabulka P0002.
Datum narození	Interní tabulka P0002.
Datum důchodového věku	Interní tabulka P0041, druh data „Důchodový věk“.
Obec	Interní tabulka P0006.
Ulice	Interní tabulka P0006.
Číslo domu	Interní tabulka P0006.
PSČ (post code)	Interní tabulka P0006.
Klíč státu	Interní tabulka P0006.
Pošta	Interní tabulka s poštami. (prvních 5 písmen bez mezer a číslic).
Název zaměstnavatele	Připravená struktura.
Obec zaměstnavatele	Připravená struktura.
IČ	Připravená struktura.
Variabilní symbol	Připravená struktura.
Kód OSSZ	Parametr na vstupní obrazovce.
Místně příslušná OSSZ	Parametr na vstupní obrazovce.
Výdělečná činnost od	Zpracovávaný průběh pojištění.

Tabulka 6.6: Identifikace pojištěnce

Pokud je interní tabulka GT\_ELDP neprázdná, je spuštěn další podprogram, který na základě vyplněných tabulek GT\_ELDP a GT\_PRUBEHY vytvoří finální formulář a uloží ho do „cluster“ tabulky. Postup zpracování:

1. Prochází se interní tabulka GT\_ELDP (obsahuje jediný záznam, který je společný pro několik získaných průběhů pojištění).
2. Načtení adresáře pro výsledky zúčtování. Výsledky vypočítaných mezd se pro každý měsíc z důvodů komprese ukládají do speciální „cluster“ tabulky. Pro přístup k těmto datům je vyžadováno sekvenční číslo (identifikátor určující záznam), které je souběžně při výpočtu mezd ukládáno do adresáře výsledků. Specifikem je používání tzv. zpětného přepočtu mezd. To znamená, že spolu s výpočtem mezd za aktuální kalendářní měsíc jsou vypočítány i mzdy pro všechny předchozí měsíce v daném roce. Tak dochází k mnohonásobnému výskytu výsledků pro jednotlivé měsíce. Zaručeně správné výsledky jsou v nejnovějších výsledcích.



K výsledkům se přistupuje několika způsoby. Nejjednodušším, v této práci použitým, přístupem je získání nejnovějšího sekvenčního čísla pro jednotlivé měsíce z tabulky `HRPY_RGDIR` a následné volání funkčního modulu `PYXX_READ_PAYROLL_RESULT` se sekvenčním číslem zadaným v parametrech. Všechna získaná sekvenční čísla jsou spolu s číslem měsíce a osobním číslem uložena do pomocné interní tabulky. Vyhledávání sekvenčních čísel probíhá ve 2 fázích: v první je pomocí SQL dotazu zjištěn nejaktuálnější měsíc a poté, ve druhé fázi, je dohledáno samotné sekvenční číslo.

3. Kontrola, zda nebylo pro zpracováváný interval ELDP už jednou vygenerováno. Jestliže bylo podání řádně potvrzeno, vrací funkce příznak pro vyřazení aktuálního průběhu ze zpracování. Ten může být v dalším kroku v závislosti na vstupních parametrech interpretován i tak, že se má generovat opravný formulář. Pokud bylo podání zasláno a potvrzeno jen pro část intervalu, je tento interval ze zpracování odstraněn. Zamítnuté ELDP se bere jako by neexistovalo. Bylo-li podání pouze vygenerováno a neodesláno, je v závislosti na parametru „přepis opravených“ zadávaného na vstupní obrazovce vyřazeno nebo přepsáno.
4. Vrací-li funkce z předchozího kroku příznak pro přepsání, je původní podání odstraněno z databáze. Je-li vybrán parametr „znovu generování opravených“ a zároveň má být aktuální ELDP vyřazeno, je podání dále zpracováváno jako opravné. To znamená, že typ formuláře začíná číslem „5“ a rovněž je doplněno do pole „Datum opravy“ aktuální datum.
5. Procházení interní tabulky `GT_PRUBEHY`.
6. Načtení výsledků zúčtování. Určená funkce, ve které se na základě dříve načteného katalogu zúčtování prochází jednotlivé měsíce zpracováváného intervalu a postupně plní interní tabulku `GT_MESIC`. Z pohledu zpracování jde o velice důležité údaje. Pro každý měsíc jsou doplněny informace o počtu kalendářních dní, vyměřovacím základu, počtu nepřítomných dní a počtu dní, které lze započítat do vyloučené doby. Pro usnadnění výpočtů jsou již zde kalkulace rozděleny na před a po dni dosažení důchodového věku. Příznakem jsou navíc označeny ty měsíce, které již nespádají do období pracovního poměru (například byl-li zaměstnanec vyplacen dodatečný příjem po skončení pracovní smlouvy).

Za zmínku určitě stojí již jednou dotčený funkční modul `PYXX_READ_PAYROLL_RESULT`. Ten na základě sekvenčního čísla a osobního čísla vrací výsledek zúčtování mezd pro odpovídající měsíc. Mezi výsledky jsou kromě mzdových záležitostí i informace o nepřítomnosti zaměstnance v daném měsíci. Pro přehlednost je vhodné zdůraznit, že se rozlišují dva odlišné typy nepřítomností, započítatelné buď do vyloučené doby, nebo jako měsíce bez pojištění (nepřítomné měsíce). Získávané údaje jsou samozřejmě stejné údaje jako ty, co se zadávají do infotypu 2001. Protože se do IT 2001 zadávají i dovolené a jiné nepřítomnosti, z pohledu ELDP však nepodstatné, jsou vybírány jen ty, které byly inicializovány na začátku programu v tabulce 6.4, respektive v 6.5. Poněkud komplikovanější je vyhodnocování nepřítomností u zaměstnanců na mateřské dovolené. Podle pravidel ČSSZ se do výpočtu musí zahrnout den narození potomka, který je již obsažen v informacích o průběhu pojištění. Počet dní od začátku pracovní neschopnosti se bere jako doba vyloučená. Dny od narození do konce mateřské dovolené se započítávají do měsíců bez pojištění.

Mírnou nepříjemnost představuje nepřítomnost typu ošetření člena rodiny (dále jen OČR). Maximální souvislá doba trvání započítatelná do ELDP je dlouhodobě stanovena na 9 kalendářních dní pro osoby neosamělé a na 16 kalendářních dní pro osoby osamělé. Údaje může uživatel ovlivnit na úvodní obrazovce. Protože jsou nepřítomnosti získané funkčním modulem rozděleny, nelze vždy správně určit celkovou dobu trvání v jednom celku (zaměstnanec může v jednom měsíci OČR čerpat vícekrát), proto je OČR započítáváno spolu s ostatními nepřítomnostmi bez omezení maximální délky. Spolu s tím je ale nastaven příznak o využívání OČR. V další části jsou data načtena přímo z tabulky PA2001. Protože už jsou všechny OČR nepřítomnosti ve vyloučených dobách zahrnuty, existuje korekční proměnná uchovávací počet dní nad limit.

7. Zjištění dodatečného příjmu. Byly-li zaměstnanci vyplaceny peníze po ukončení pracovního poměru, bere se to z pohledu ČSSZ jako dodatečný příjem. Jeho zařazení závisí na druhu pracovního poměru. Pro práci malého rozsahu a dohodu o provedení práce je dodatečný průběh započítán do posledního zúčtovaného měsíce. Pro běžný pracovní poměr musí být dodatečný příjem veden jako samostatný průběh pojištění označený kódem „P++“.
8. Výpočet celkového vyměřovacího základu pro zpracovávané období. Počítá se jako součet vyměřovacích základů (vyšších než minimální stanovená částka pro práci malého rozsahu a dohody o provedení práce) za jednotlivé měsíce.
9. Rozdělení průběhu na dva (při dosažení důchodového věku během zpracovávaného intervalu). Původní záznam je časově omezen datem důchodového věku, navíc je jeho vyměřovací základ vynulován. Kromě úpravy intervalu je nově vzniklý průběh interně označen příznakem „důchod“.
10. Určení zpracovávaného průběhu. Pro každý druh pracovního poměru platí jiná pravidla výpočtu pro průběh pojištění.
11. Pokud se jedná o **běžný pracovní poměr**, přidá se průběh s dodatečným zúčtováním mezi ostatní a spustí se odpovídající podprogram. V něm jsou procházeny jednotlivé měsíce načtené v tabulce GT\_MESIC. Vybrány jsou ale jen ty, které zasahují do období aktuálního průběhu. Měl-li zaměstnanec nulový příjem a zároveň žádné vyloučené doby, je vybraný měsíc označen jako nepojištěný (ve formuláři znakem „X“). Počet dní v takovém měsíci je připočten do proměnné LV\_NEPRIT. Může však nastat situace, že zaměstnanci byla zúčtovaná mzda za měsíc, ve kterém nebyl ani jeden den v práci (odměna). V takovém případě je počet dní připočítáván do korekční proměnné LV\_KOR, která je na závěr odečtena od vyloučených dob. Samotné vyloučené doby se za jednotlivé měsíce postupně ukládají do pomocné proměnné LV\_VD.

U zaměstnanců v důchodovém věku je výpočet rozšířen o doby odečtené. Do zpracování se promítají až ve chvíli, kdy má zaměstnanec zadanou nepřítomnost. Pokud je v předchozím výpočtu celý měsíc označen jako nepřítomný, započítává se to pouze do dob odečtených, v opačném případě je nepřítomnost započítána do vyloučených dob i do dob odečtených. Ukázka dopočítání výsledných hodnot (pseudo algoritmus):

```

dny = ( konec průběhu - začátek průběhu ) + 1.
vyloučené doby = lv_vd - korekce OČR.
IF důchod.
    doby odečtené = lv_neprit + lv_vd - korekce OČR.
    dny = dny - doby odečtené.
ELSE.
    dny = dny - lv_neprit.
ENDIF.
vyloučené doby = vyloučené doby - lv_kor.

```

V případě, že jsou vyloučeny všechny měsíce, je místo dvanácti „X“ použit jediný symbol, umístěný na pozici třináctého měsíce.

12. Pokud se jedná o **práci malého rozsahu (PMR)**, je opět spuštěn odpovídající podprogram. Hlavním rozdílem oproti běžnému poměru je skutečnost, že výpočet je ovlivněn minimálním vyměřovacím základem zadávaným na hlavní obrazovce. Pro představu čtenáře: minimální vyměřovací základ pro práce malého rozsahu pro rok 2012 činí 2500 Kč za měsíc.

Jde-li o pracovní poměr započatý během zpracovávaného intervalu, tak program nejdříve prochází všechny načtené měsíce a hledá první měsíc, ve kterém je částka vyšší než minimální vyměřovací základ. Není-li žádný takový měsíc nalezen, je zpracovávaný poměr ze zpracování vyřazen a ELDP se neodesílá. Počáteční datum průběhu je nastaveno na první den v měsíci, ve kterém byl zúčtován příjem splňující uvedenou podmínku. Následuje průchod zbylých měsíců (pokud pracovní poměr započal před zpracovávaným intervalem, začíná se prvním měsícem). Algoritmus je od tohoto momentu téměř shodný jako u běžného pracovního poměru, jediným rozdílem je kontrola, zda je splněn minimální vyměřovací základ. Není-li, je měsíc označen za nepřítomný.

13. Pokud jde o **dohodu o provedení práce (DPP)**, není výpočet příliš odlišný od PMR. Opět je stanoven minimální vyměřovací základ (pro rok 2012 je tato částka 10 000 Kč). Hlavním rozdílem je povinnost určit vyměřovací základ za celou fyzickou osobu. Přesáhl-li celkový součet minimální hranici, jsou podle čísla smlouvy dohledány částky náležící k jednotlivým poměrům. Přes číslo smlouvy v aktuálně zpracovávaném průběhu pojištění je v IT 0015 dohledána vyplacená částka. Výhodou je, že se u prací malého rozsahu na VUT nezadávají nepřítomnosti. Dosud žádný takový případ nenastal, podle metodické příručky je však i taková situace možná, proto je pro výpočet vyloučených dob převzat algoritmus z PMR.

14. Získání třímístného kódu. Pro potřeby VUT je nutné rozlišovat jen 3 typy kódů:

- běžný pracovní poměr (například 1++),
- pracovní poměr v důchodu (například 1D+) a
- dodatečně zúčtovaný pracovní příjem (například P++).

15. Sloučení všech průběhů do samostatných formulářů. Čte se tabulka **GT\_ELDP**, ze které jsou vybírány formuláře, ve kterých není doplněn kód prvního průběhu (vždy 1 záznam). V procházených formulářích je hlavička formuláře již vyplněna, a proto se

přidávají pouze řádky do sekce průběhy pojištění v daném roce. Před jejich doplněním jsou seskládány podle kódu a zároveň tak, aby šly časově za sebou. Až poté se prochází tabulka s průběhy pojištění GT\_PRUBEHY. Vzhledem k tomu, že ELDP formulář má pouze 3 pole pro zadání průběhů pojištění, jsou získávané řádky tabulky interně číslovány. Na každý z procházených řádků je aplikována funkce, která podle interního číslování zařadí průběh správně do formuláře, spolu s tím je doplněn typ formuláře a aktuální součtové údaje (celkem vyloučené doby, celkem vyměřovací základ a celkem doby odečtené). Po přidání třetího řádku je vyplněný formulář přidán do tabulky GT\_ELDP (nebude znovu načten, protože nespĺňuje uvedenou podmínku – má vyplněn kód). Tímto vznikl kompletní ELDP formulář. Protože tabulka GT\_PRUBEHY může obsahovat další řádky, je sekce „průběh pojištění v daném roce“ vymazána a pokračuje se v doplňování zbylých průběhů pojištění. Na závěr jsou všechny získané ELDP ještě jednou seřazeny podle začátku prvního průběhu.

16. Vygenerované formuláře jsou uloženy do „cluster“ tabulky.

### 6.3.5 Uložení vygenerovaných dat

Jak již bylo několikrát zmíněno, pro ukládání ELDP je k dispozici „cluster“ tabulka. Popisovaná aplikace využívá původní tabulku určenou pro originální řešení dodávané společností SAP. Díky tomu lze využít originální makra, která po naplnění vhodných struktur načtou požadovaný evidenční list nebo ho uloží. Princip práce je založený na vytvoření tzv. třídícího klíče. Tedy textového řetězce složeného ze 40 znaků. Skládá se z osobního čísla, roku, intervalu a aktuálního stavu. Třídící klíč páruje záznam v tabulce T5TELDP (význam tabulky viz. 5.2) a podání uložené v „cluster“ tabulce (jde o téměř shodný způsob jako u čtení výsledků zúčtování mezd, jen místo funkčního modulu se zde využívá makro, a místo sekvenčního čísla se používá třídící klíč). Uložení dat vyžaduje naplněnou strukturu odpovídající tabulce T5TELDP, výše zmíněný třídící klíč a vyplněná data evidenčního listu, která jsou povinně předávána jako tabulka s jedním vloženým řádkem. Po spuštění makra je kromě uložení formuláře přidán i nový záznam do databázové tabulky T5TELDP. Pro načítání uložených informací se nejprve vybere požadovaný záznam v tabulce T5TELDP (například přes osobní číslo či rok). Ze získaného záznamu je přečten třídící klíč. Spuštěním odpovídajícího makra dojde k naplnění tabulky typu T5TELDP2004, její jediný řádek obsahuje požadované údaje.

V tabulce T5TELDP se dle pole RELID (v SAP označováno jako „Id. relace“) rozlišují dva stavy ELDP:

- **EL** označuje podání, jehož data lze měnit.
- **SE** označuje podání do jehož dat již zasahovat nelze.

Dále jsou pro rozlišení stavu podání k dispozici 3 jednoznačková datová pole:

- **H\_TYPE** určuje, zda je vygenerovaná položka nová či opravená.
- **H\_TRANSPORT** určuje, zda položka je odeslaná nebo neodeslaná.
- **H\_STATUS** určuje, zda je položka potvrzená nebo zamítnutá.

Každý formulář postupně prochází stavy uvedenými v tabulce 6.7.

Modul ZHR\_DP\_ELDP\_GENEROVANI\_F05 obsahuje funkce výhradně určené pro práci s „cluster“ tabulkou. Pro zpřístupnění maker pro práci s „cluster“ tabulkou musí být do programu vložen modul RPCELDT2I. K dispozici jsou makra uváděná v tabulce 6.8.

Stav	Popis stavu	Typ	1	2	3
1.	Formulář je nově vygenerován.	EL	N		
2.	Formulář je ručně opraven.	EL	R		
3.	Formulář je zahrnut v XML souboru.	SE			
4.	Formulář je odeslán ČSSZ.	SE	R	D	
5a.	Formulář je ČSSZ potvrzen, připraven pro hromadný tisk.	SE	R	D	Y
5b.	Formulář je ČSSZ zamítnut, vrací se do stavu 2.	SE	R	D	N

Tabulka 6.7: Stav podání (1–H.TYPE, 2–H.TRANSPORT, 3–H.STATUS)

Název makra	Název třídícího klíče	Tabulka s daty	Popis
RP-EXP-EL	EL-KEY.	eldp	uložení
RP-EXP-SE	SE-KEY.	eldp	uložení
RP-IMP-EL	EL-KEY.	eldp	načtení
RP-IMP-SE	SE-KEY.	eldp	načtení

Tabulka 6.8: Makra pro práci s „cluster“ tabulkou

Z předchozího výkladu je zřejmé, že ZHR\_DP\_ELDP\_GENEROVANI\_F05 obsahuje implementace, které se musí používat napříč celou aplikací. Jsou k dispozici funkce pro uložení nově vygenerovaného podání. V jejich parametrech je rozlišeno, zda je evidenční list nově vygenerovaný nebo ručně vytvořený (opravený). Samotné uložení do „cluster“ tabulky uskutečňuje vyhrazená funkce. Při procesu generování je až na úrovni této funkce rozlišen testovací běh programu. Vybral-li uživatel na úvodní obrazovce parametr testovací běh, jsou všechny následně uvedené kroky přeskočeny. Z předaných dat se nejdříve vytvoří třídící klíč, poté je v další funkci připravena struktura T5TELDP. Dále je vložen záznam s daty do tabulky ELDP a je spuštěno makro RP-EXP-EL.

Stejně pracuje i další funkce uplatňovaná při změně typu podání z „EL“ na „SE“, tedy při generování XML souboru. Odeslání formuláře do ČSSZ je také podmínkou ke změně stavu. Mění se však pouze sloupec H\_TRANSPORT. Potvrzení o přijetí, respektive zamítnutí ELDP, je realizováno samostatnými funkcemi, liší se od sebe jen stavem, který se k vybranému podání přiřazuje.

V aktuálně popisovaném modulu jsou k dispozici rovněž podprogramy pro čtení uložených dat. První z nich vrací pro zpracovávané období nejnovější uložený formulář, respektive formulář, který obsahuje poslední průběh pojištění v zadaném období. Funkce je součástí funkcionality, která dokáže navázat na poslední odeslaný průběh.

Jako zdroj dat pro tabulkové zobrazení a generování XML souboru je vyhrazena další funkce. Podle osobního čísla a roku zpracování jsou načítány evidenční listy typu „EL“. U všech takto získaných formulářů se kontroluje stav odeslání. To znamená že, pokud ELDP nebylo vůbec odesláno nebo již bylo zamítnuto a čeká na opravu, je zařazeno do tabulky mezi výsledky. Opakem je funkce shromažďující odeslaná podání, využívá se pro účely hromadného tisku.

### 6.3.6 Zobrazení vygenerovaných dat

Výstup generování je textový výpis. Jelikož nelze ovlivnit to, v jakém pořadí logická databáze zpracovává osobní čísla, musí být výsledné seřazení realizováno ve zdrojovém kódu. Po uložení formuláře do databáze je vygenerované podání zařazeno i do tabulky GT\_ELDAP\_TISK. Až jsou všechna čísla zpracována, spustí se podprogram, ve kterém jsou nejdříve záznamy seřazeny dle požadavku uživatele, poté jsou postupně procházeny a tisknuty na obrazovku. Výstup výsledných informativních údajů je realizován standardním výpisem na obrazovku WRITE. Výsledné rozložení prvků je s mírnými úpravami převzato z původního řešení (obrázek 6.3).

```
Program ZHR_DP_ELDAP_GENEROVANI 1
Nákladové středisko: 90370
Osobní číslo: 01007326 Za rok: 2012 Typ: 01 Oprava:
Kód OSSZ: 772 Místně příslušná OSSZ: Brno - město
1. Identifikace pojištěnce
Příjmení Jméno Titul Datum narození Rodné číslo
Leštinský Ondřej Bc. 02.06.1988
Ulice Číslo domu Rodné příjmení
Čsl. armády 24 Leštinský
Obec Pošta PSČ Stát Místo narození Stát nar.
Šumperk Šumpe 787 01 CZ Šumperk CZ
2. Průběh pojištění v daném roce
Kód MR Od Do Dny 1 2 3 4 5 6 7 8 9 10 11 12 VyD Vyměř.základ D0d
1+- N : 01.06.2012 - 31.12.2012 214 . . . . . 000 000
3. Identifikace údajů zaměstnavatele a podpisy
Název zaměstnavatele IČ Variabilní symbol
VUT v Brně 00216305 7720103444
Výdělečná činnost od Datum ELDAP
20.04.2013
```

Obrázek 6.3: Výstup vygenerovaného podání

## 6.4 Tabulkové zobrazení

Mezi standardní možnosti zobrazení v SAP patří tabulkové zobrazení, programátory označované jako ALV. V původním řešení zpracování ELDAP od SAP nebylo zahrnuto. Funkcionalita byla implementována na základě přání mzdové účtárny. Cílem je přehledně zobrazit vygenerovaná data a usnadnit tak kontrolu, zda jsou vyplněna všechna požadovaná pole. Tabulkové zobrazení lze realizovat několika způsoby, v této práci jsou všechny tabulky realizovány následujícím způsobem:

1. Vytvoření obrazovky v programu Screen Painter.
2. Umístění kontejneru do vytvořené obrazovky.
3. Definice PROCESS BEFORE OUTPUT. Určuje moduly, které se automaticky provedou před spuštěním obrazovky. V rámci modulu je vybráno menu obrazovky, vytvořena instance objektu reprezentující prvek kontejner definovaný v předchozím bodě. Zobrazovaná tabulka je instancí speciální třídy reprezentující tabulku, při jejím vytváření se v konstruktoru vyžaduje výše vytvořený objekt kontejneru.

Dále je načtena struktura zobrazovaných dat, ze které je funkčním modulem vytvořen tzv. „field-katalog“ reprezentovaný tabulkou, jejíž řádky představují jednotlivé sloupce zobrazované tabulky. Jejich úpravou je programátor schopný určit nové názvy sloupců uváděné v záhlaví, zobrazovanou velikost polí a spoustu dalších užitečných věcí. Poté je nastaven tzv. „layout“. Všechna připravená data (název struktury, „field-katalog“, „layout“ a samotná data) jsou nakonec předána odpovídající metodě vestavěné třídy, která zajišťuje zobrazení tabulky.

4. Definice `PROCESS AFTER INPUT`. Určuje modul spouštějící se jako odezva na činnost uživatele (například klepnutí na položku menu). Modul je implementován jako rozhodovací strom volající vhodnou funkci v závislosti na předaném argumentu.

### 6.4.1 ALV tabulka

Pro každé osobní číslo čtené logickou databází je volána funkce pro přípravu dat. V ní jsou načtena neodeslaná podání z „cluster“ tabulky. Získaná podání jsou převedena do formátu zobrazovaného v tabulce (postupně se doplňují záznamy v `GT_ALV_ELDP`). Při převodu je zohledněna požadovaná editovatelnost polí, avšak jen u neklíčových buněk (nelze upravovat osobní číslo a rok). Navíc jsou barevně odlišeny již zkontrolované (opravené) řádky. Součástí struktury tvořící tabulku `GT_ALV_ELDP` je speciální tabulka sloužící pro ovlivnění barev v jednotlivých buňkách. Uvede se pouze název sloupce a číselný kód barvy. Paleta barev v SAP je poměrně omezená, ale i tak vysoce dostačující pro běžné zvýrazňování.

Před samotným zobrazením obrazovky je zdrojová tabulka seřazena dle výběru uživatele. Při spuštění nové obrazovky se automaticky provede modul typu `PROCESS BEFORE OUTPUT`. Menu přiřazené této obrazovce se skládá z těchto položek:

- opuštění obrazovky,
- uložení změn,
- spuštění PDF formuláře,
- opravný formulář,
- generování XML souboru a
- automatické doplnění údajů.

Třída reprezentující tabulku obsahuje nespočet metod pro snadné ovládání. Nejčastěji se volá metoda vracející všechny označené řádky tabulky a metoda, která podle grafické tabulky aktualizuje změněné záznamy ve zdrojové tabulce. Pro většinu popsanych operací je zásadní výběr právě jednoho řádku. Samotná data jsou dohledána ve zdrojové tabulce přes číslo řádku.

Pro načtení vybraného řádku je využit tzv. `FIELD SYMBOL` [11]. Změna jeho obsahu se automaticky promítá do zdrojové tabulky bez nutnosti použít další příkazy. Toto se hlavně využije u opravného formuláře a při automatickém doplnění údajů. Pro grafické znázornění změn je nezbytné zavolat metodu, která ALV tabulku zaktualizuje.

Uživatel může ručně přidat nové záznamy přes nabídku tabulky. Do nově vzniklého prázdného řádku musí dopsat veškeré povinné údaje případně použít funkci pro automatické doplnění údajů. Vyhledání potřebných údajů je závislé na existenci záznamu se zadaným osobním číslem v infotypu 0002. Operace „Automatické doplnění údajů“ doplní hlavičku formuláře a tím uživateli hodně ušetří práci. Doplněvané údaje hlavičky jsou shodné s těmi, které již byly popsány v části o generování ELDP.

### 6.4.2 Uložení změn

Nově přidané záznamy musí být spolu s upravenými záznamy uloženy do „cluster“ tabulky. Aby nedocházelo k opakovanému přepisování dat, jsou vždy vybrány jen ovlivněné řádky. Celá zdrojová tabulka je řádek po řádku porovnávána s pomocnou tabulkou, která obsahuje původní nezměněná data. Nebyl-li záznam nijak změněn, je odstraněn z původní tabulky, naopak liší-li se záznam, provede se kontrola veškerých povinných polí, pokud není některé z povinných polí zadáno, je uživatel informován, na kterém řádku jaký údaj chybí a zpracování končí bez uložení, v opačném případě je u záznamu v původní tabulce nastaven příznak změny. Kontroluje se, zda pro osobní číslo existuje záznam v infotypu 0002, a zda má uživatel dostatečná oprávnění pro jeho zadání. U nově vložených záznamů je kromě kontroly sestaven třídící klíč pro uložení a nastaven příznak nového záznamu. Uložení, respektive výmaz dat, je možné až po porovnání všech záznamů. Původní tabulka nyní obsahuje veškeré uživatelem ovlivněné záznamy. Rozlišuje 3 typy záznamů:

- změněný záznam (příznak změny),
- nový záznam (příznak nového záznamu) a
- smazaný záznam (bez příznaku).

Změněný záznam je nejdříve odstraněn z „cluster“ tabulky a poté znovu vložen. Mohly se totiž změnit údaje zadávané do třídícího klíče, který nelze změnit. Nový záznam je upraven pro zobrazení v ALV tabulce a současně je ve standardní formě uložen do „cluster“ tabulky. Nemá-li záznam příznak, znamená to, že se nachází jen v původní tabulce a tím pádem byl v zobrazované tabulce smazán. Proto je odstraněn i z databáze.

### 6.4.3 Opravný formulář

Pro přehledové zobrazení je tabulka dostačující, pro pohodlnější práci uživatele je však vhodné zobrazit data ve formuláři, který má shodné rozložení prvků jako originální papírový formulář. Opravný formulář je realizován pomocí samostatné obrazovky. Původním záměrem bylo využít pro tento účel interaktivní PDF formuláře, které se však vzhledem k rychlosti načítání nedaly použít. Obrazovka zobrazující formulář byla převzata z originálního řešení. Opravný formulář je spuštěn vlastním podprogramem, který zjistí vybraný řádek a zobrazí novou obrazovku s formulářem, po opuštění obrazovky promítne případně změny do zobrazené tabulky. Samotná logika zpracování je obsažena v modulu ZHR\_DP\_ELDAP\_GENEROVANI\_F04. Grafické prvky obrazovky s formulářem jsou navázány na strukturu tabulky. Provedené změny může uživatel uložit přímo na této obrazovce. Před samotným uložením jsou údaje opět zkontrolovány.

### 6.4.4 PDF formulář

Jak již bylo popsáno v kapitole 3.6 – výběr vhodného rozhraní, uživatel si může zobrazit podání v originálním formuláři. PDF zobrazí samostatná funkce. A stejně jako u ostatních funkcí v menu nabídce, je nejdříve načten řádek, který uživatel vybral. Data z řádku jsou transformována do požadované podoby, úprava se týká zobrazení dat v podání. Například je z dat v průběžích pojištění odstraněn rok či je eliminováno zobrazení nul. Výsledkem je připravená struktura. Zobrazování PDF souborů umožňuje speciální tiskárna určená výhradně pro tisk PDF. Uživatelé ale mají nastavenou výchozí fyzickou tiskárnu, proto je potřeba programově ošetřit přiřazení PDF tiskárny. Toho lze dosáhnout naplněním určené



struktury a jejím předáním odpovídajícímu funkčnímu modulu. Nakonec se spustí další funkční modul s parametry dokumentu a hlavně se strukturou nesoucí zobrazovaný obsah.

Formulář byl realizován v nástroji Adobe LiveCycle Designer, který je po instalaci na lokální stanici dostupný i v SAP. Formulář se skládá ze dvou stran formátu A5. První strana vypadá jako originální formulář. ČSSZ ale vyžaduje tisk do originálního formuláře, proto se na druhé straně tiskne jen text do prázdné stránky. Formulář byl vytvořen tak, že do pozadí PDF šablony byl vložen obrázek originálního formuláře získaný v [20] a podle rozvržení jeho prvků byly dodatečně nastaveny pozice výstupních textových polí.

## 6.5 Vytváření XML souborů

Zkontrolovaná podání jsou odesílána na ČSSZ. Požadovaným formátem je XML struktura definovaná v [25] (XSD je přiloženo na DVD). Vytváření XML souborů se opět skládá ze dvou částí: příprava dat a jejich zpracování. Pracuje se s tabulkami `GT_ELDP` a `GT_ELDP_XML`. Údaje požadované v XML souboru se opět mírně liší oproti běžnému formuláři, proto jsou při zpracování upravovány do požadované podoby. Formát polí je definován v [23]. Úpravy spočívají v odstranění desetinných míst u zadávaných částek a smazání úvodních nul u zadávaných dní. Pro potřeby e-Podání je formulář rozšířen o další elementy, které jsou použity pro očíslování jednotlivých formulářů, respektive průběhů pojištění.

Při přípravě dat jsou z „cluster“ tabulky vybírána jen neodeslaná podání pro osobní číslo a zpracováván rok. Získávané údaje jsou přidávány do tabulky `GT_ELDP`. Při generování XML souboru dochází ke změně stavu údajů v „cluster“ tabulce. Formulář, který je uložen do XML, už může být upraven jen v případě, že bylo podání ČSSZ zamítnuto. Generování výstupního souboru je omezeno počtem uvedených podání. ČSSZ stanovilo limit na 1500 formulářů v rámci jednoho podání. Uživatel si v úvodní obrazovce může požadovaný počet upravit dle svého uvážení. Dosud se jedno podání skládalo z 300 formulářů (zachováno ve výchozí variantě). V programu je uvedený problém řešen tak, že se zdrojová tabulka podle požadavku uživatele nejdříve seřadí a hned poté je vygenerován první název výstupního XML souboru, který je do tabulky `T5TELDP` přiřazen všem ovlivněným záznamům. Název souboru tvoří:

- identifikátor systému,
- jméno přihlášeného uživatele,
- datum,
- čas a
- pořadové číslo.

Tabulka `GT_ELDP` je procházena v cyklu řádek po řádku. V každé iteraci se přidává záznam do pomocné tabulky `GT_ELDP_XML`. Jakmile je počet záznamů v pomocné tabulce roven zadanému limitu, je vytvořen XML soubor.

Samotné vytvoření XML souboru je v ABAP celkem snadná záležitost. Prostředí nabízí několik variant k realizaci. Řešení použité v této práci se zakládá na XSLT transformaci. Na podobném principu je založeno i například SAP NetWeaver PI [6]. V literatuře [17] jsou popisovány 4 druhy XSLT transformací v ABAP:

- XML do XML,

- XML do ABAP,
- ABAP do XML a
- ABAP do ABAP.

V implementaci řešení jsou aplikovány dvě varianty: **XML do ABAP** a **ABAP do XML**. První zmíněná varianta slouží pro načtení odpovědi při odesílání zpráv ČSSZ. V této části práce je podstatná druhá zmíněná varianta, tedy vytvoření XML souboru z interní tabulky. Principiálně jde vždy o stejný postup:

1. Vytvoření XSLT transformace v ABAP editoru.
2. Vytvoření a naplnění zdrojové struktury daty.
3. Deklarace tabulky pro uchování výsledků.
4. Spuštění příkazu CALL TRANSFORMATION.

Výhradně pro převod interní tabulky GT\_ELDAP\_XML do XML souboru s ELDAP je implementována transformace ZXSL\_ELDAP\_ABAP\_TO\_XML. Získaný výsledek je uložen k uživateli do počítače. Pro práci se soubory ABAP nabízí speciální třídu. Po uložení je pomocná struktura vyprázdněna, vygeneruje se nový název souboru a pokračuje se v průchodu tabulkou GT\_ELDAP. Pro představu čtenáře je uveden příklad ELDAP podání (dostupný na [24]):

```
<?xml version="1.0"encoding="UTF-8"?>
<RELDAP version="2009.1" xmlns="http://schemas.cssz.cz/ELDP09">
<eldp09 sqnr="1" yer="2012" typ="01" dep="115" dre="" tco="" nam="Praha 5">
<client bro="500706050">
<name sur="Kolářek" fir="František" tit=""></name>
<adr cit="Praha 6 - Dejvice" str="Svatovítská" num="1480/17a" pne="16000"
cnt="CZ" pos="Praha"></adr>
<birth dat="1950-07-06" nam="Červěňák" cit="Spišská Nová Ves"></birth>
</client>
<items coun="1" sdex="" sinc="10000" sdar="">
<t1 cod="1++" sre="A" fro="2012-01-01" to="2012-01-19" din="19" inc="10000"
row="1" m1="" m2="" m3="" m4="" m5="" m6="" m7="" m8="" m9="" m10="" m11=""
m12="" m13=""></t1>
</items>
<comp nam="CSSZ - testovací a.s." id="62156489" vs="1159876547"
cre="2012-01-19" fro=""></comp>
</eldp09>
</RELDAP>
```

## 6.6 Odesílání dat ČSSZ

Hlavním cílem této části aplikace je předat vygenerovaná data ČSSZ. Při návrhu bylo odeslání dat uvažováno jako samostatná aplikace. Ve výsledném řešení je však i tato část zařazena do hlavní aplikace. Při jejím spuštění z hlavní obrazovky se uživateli zobrazí tabulka s názvy všech XML souborů, ve kterých figurují osobní čísla odpovídající zadání na úvodní obrazovce. Uživatel může dle stavu zpracování buď odesílat podání, přijímat

odpověď, nebo manuálně potvrdit správnost dat, respektive zamítnout chybné. Odesílaná data jsou v XML formátu v GovTalk obálce. Původní XML soubor je ve zprávě zastoupen v zašifrované podobě a formou digitálního podpisu.

Systém uchovává bližší informace o stavu všech odesílaných podání. Informace jsou umístovány do tabulky ZHR\_CSSZ, která vychází z návrhu v kapitole 4. Tato tabulka rozšiřuje údaje o podání, které nelze zařadit do T5TELDP. Aplikace odesílající elektronická podání je z části tvořena následujícími třídami:

- ZCL\_DP\_CSSZ\_PODANI\_ABSTR je abstraktní třída, která reprezentuje e-Podání jako celek. Obsahuje metody pro práci s tabulkou ZHR\_CSSZ a metody pro odeslání podání, respektive jeho zpracování.
- ZCL\_DP\_ELDP tvoří evidenční list, své vlastnosti třída dědí z výše uvedené abstraktní třídy. Obsahuje potřebné parametry pro definici ELDP.
- ZCL\_DP\_CSSZ\_GOV\_TALK reprezentuje XML zprávu – GovTalk obálku. Skládá se z metod určených pro manipulaci s obálkou (nastavení a zobrazení vybraných parametrů, příprava XSLT transformace, serializace).

Za účelem znovupoužitelnosti je do řešení zahrnuta abstraktní metoda. ELDP tvoří pouze zlomek agendy, kterou ČSSZ vyžaduje. Na VUT se také elektronicky odesílají informace o nástupu do zaměstnání/skončení. Implementované řešení bere v potaz možnost snadného rozšíření, například v budoucnu o třídu ZLC\_DP\_ONZ.

Informace o stavu podání, uložené v tabulce T5TELDP (6.7), doplňují údaje v tabulce ZHR\_CSSZ.

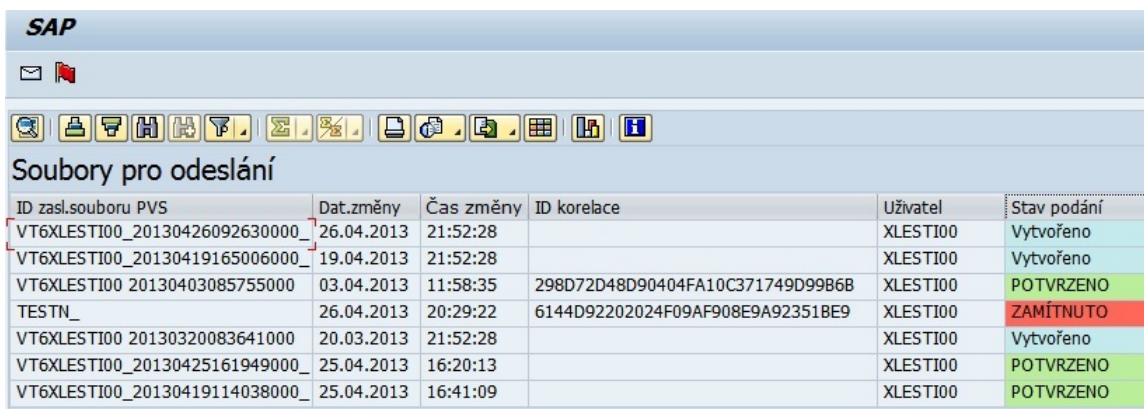
### 6.6.1 Zobrazení podání

Podobně jako u ostatních tabulkových zobrazení jsou nejdříve shromážděny zdrojové informace. V tomto případě to konkrétně znamená, že se podle aktuálně zpracovávaného osobního čísla prohledá tabulka T5TELDP, ze které se zjistí stav odeslání podání. Bylo-li podání již odesláno, je název souboru zařazen do interní tabulky typu TY\_IDSOUB. Tato interní tabulka se skládá z identifikátoru XML souboru, data poslední změny, času poslední změny, přihlašovacího jména uživatele, který změnu provedl, id korelace a stavu podání. Při programovém zpracování je zajištěna eliminace výskytu duplicitních názvů – každý název XML souboru se vyskytuje nanejvýš jednou. Před zařazením záznamu do interní tabulky jsou doplněny zbylé chybějící údaje. K jejich získání slouží statické metody třídy ZCL\_DP\_CSSZ\_PODANI\_ABSTR. Pokud ještě podání nebylo odesláno, je použito aktuální datum a čas. Rovněž jsou barevně rozlišeny jednotlivé stavy podání. Výsledná tabulka je umístěna v obrazovce 0200.

Na další obrazovku (určenou pro odesílání podání) se uživatel dostane buď výběrem řádku a klepnutím na obálku nebo tak, že dvakrát poklepe na vybraný řádek. Pokud bylo podání ČSSZ zamítnuto, může si chybu, která zamítnutí způsobila, zobrazit v další obrazovce (výběr řádku a zvolení červené vlajky v menu).

### 6.6.2 Odesílací menu

Přechod z předchozí stránky opět zajišťuje odpovídající podprogram. K dalšímu zpracování předává celou strukturu odpovídající vybranému řádku. Pro snadnější aktualizaci polí se



ID zasl.souboru PVS	Dat.změny	Čas změny	ID korelace	Uživatel	Stav podání
VT6XLESTI00_20130426092630000_	26.04.2013	21:52:28		XLESTI00	Vytvořeno
VT6XLESTI00_20130419165006000_	19.04.2013	21:52:28		XLESTI00	Vytvořeno
VT6XLESTI00_20130403085755000_	03.04.2013	11:58:35	298D72D48D90404FA10C371749D99B6B	XLESTI00	POTVRZENO
TESTN_	26.04.2013	20:29:22	6144D92202024F09AF908E9A92351BE9	XLESTI00	ZAMÍTNUTO
VT6XLESTI00_20130320083641000_	20.03.2013	21:52:28		XLESTI00	Vytvořeno
VT6XLESTI00_20130425161949000_	25.04.2013	16:20:13		XLESTI00	POTVRZENO
VT6XLESTI00_20130419114038000_	25.04.2013	16:41:09		XLESTI00	POTVRZENO

Obrázek 6.4: Obrazovka pro odeslání

předává struktura jako „field-symbol“. Jednoznačným identifikátorem podání je název souboru, na jehož základě se díky informacím z T5TELDP, opět formou tabulky, zobrazí osobní a rodná čísla všech zaměstnanců, kteří jsou v podání zahrnuti. Uvedená osobní čísla se ukládají do interní tabulky. Po jejich načtení je spuštěna obrazovka 0201.

Při inicializaci obrazovky se nejdříve vytvoří instance třídy ZCL\_DP\_ELDP. Dále je kromě vytvoření ALV tabulky do obrazovky doplněn i email uživatele. Email se používá jako jeden z autentizačních údajů při odesílání podání. Emailová adresa se spolu s ostatními informacemi o uživateli zadává v transakci „SU01“. Zadávané údaje jsou dostupné přes určený funkční modul. Podle výchozí cesty uvedené na vstupní obrazovce a názvu XML souboru je vytvořena cesta k souboru s podpisem a k souboru se zašifrovanými daty (více viz. další podkapitola).

### 6.6.3 Šifrování a digitální podpis

Původním záměrem byla realizace digitálního podpisu a šifrování přímo v SAP. Bohužel instalovaná verze SAP tento záměr na VUT neumožnila realizovat. Hlavní problém je absence SHA-2, které ČSSZ vyžaduje. Proto se k daným účelům využívá stávající způsob řešení, který spočívá ve vytvoření šifrovaného souboru a odděleného podpisu v zakoupené aplikaci **TrustPort eSign** [15] přímo na počítači uživatele. Pro maximální uživatelskou přívětivost je implementována funkce, která přímo ze SAP spustí průvodce vytvořením digitálního podpisu nebo šifrovaného souboru. Tímto procesem vznikne soubor s příponou **.esg** (digitální podpis) a soubor s příponou **.enc** (zašifrovaná podoba).

Uživatel je podle grafických ukazatelů informován o existenci digitálního podpisu, respektive zašifrovaného souboru. Správné nastavení ukazatelů zabezpečuje určená funkce. Vzhledem k binární podobně vytvořených souborů nekontroluje obsah, ale jen existenci požadovaných souborů. K vytvoření uvedených grafických prvků lze opět využít jeden z funkčních modulů. Vstupem je název požadované vestavěné ikony a výstupem je přímo grafická reprezentace ikony. Proto se jako výstupní proměnná používá přímo grafický prvek vložený do obrazovky v nástroji Screen Painter.

#### 6.6.4 Příprava GovTalk obálky pro odeslání podání

Stiskem tlačítka s ikonou obálky na ovládacím panelu se spustí podprogram pro odeslání. V jeho těle se připravuje objekt odpovídající ELDP podání. Uložené údaje jsou následně odeslány. Postup přípravy dat je následující:

1. Kontrola stavu podání, musí být ve stavu „Vytvořeno“.
2. Nastavení cesty k souboru s digitálním podpisem včetně kontroly jeho existence.
3. Nastavení cesty k souboru se zašifrovanou částí včetně kontroly jeho existence.
4. Nastavení emailu podle zadání na obrazovce.
5. Nastavení potřebného hesla.
6. Vytvoření podání. Postupně jsou načteny kryptograficky upravené soubory. Pro jejich načtení je implementována metoda, která binární soubor načte do pomocné tabulky. Další funkční modul převede vstupní tabulku na binární řetězec, který je poté konečně uložen do formátu kódování Base64.
7. Naplnění GovTalk obálky s požadavkem pro odesílání dat. Pro jistotu je hlavička obálky vyplněna stejným způsobem jako v doposud používané aplikaci. Hlavně jde o identifikaci odesílatele pomocí variabilního symbolu organizace, identifikátoru odesílatele, emailové adresy a „tajného“ hesla. Všechny uvedené údaje jsou v čistě textové podobě. Hlavní autentizace je kryptograficky zabezpečena pomocí digitálního podpisu. Odesílaná data jsou rovněž zašifrována veřejným klíčem (RSA 2048 bitů). Uvedené čistě textové údaje slouží pouze pro přístup k dalšímu zpracování.
8. Transformace vytvořené obálky do podoby XML souboru, XSLT transformace typu ABAP do XML. Výsledný XML soubor je uložen do vnitřní proměnné jako textový řetězec.
9. Nastavení adres serverů pro odeslání. ČSSZ poskytuje pro podání dvě adresy:
  - <https://vrep1.cssz.cz/VREP/submission>
  - <https://vrep2.cssz.cz/VREP/submission>

Obě adresy může uživatel změnit na hlavní obrazovce aplikace.

#### 6.6.5 Odeslání podání

Jakmile jsou data v objektu připravena, zavolá se metoda pro odeslání. V SAP na VUT není bohužel korektně nastaven certifikát pro vytváření HTTPS spojení a veškeré pokusy o jeho nastavení končily neúspěchem. Vytvoření HTTPS spojení vždy přeruší chybová hláška: „**ICM\_HTTP\_CONNECTION\_FAILED**“. Proto se původní záměr, odesílat přímo ze SAP, nepodařil úplně realizovat. Odpovídající metoda obsahuje implementaci, která by po korektní instalaci potřebného certifikátu dle [12] měla fungovat. Z výše uvedených důvodů se však prakticky metodu nepodařilo ověřit.

V implementované metodě je vytvořen standardní objekt typu HTTP klient, který zapouzdřuje dotaz i následnou odpověď. Parametry spojení jsou nastaveny dle tabulky 6.9. Po odeslání připraveného dotazu je přečtena odpověď. Získaná odpověď je v binární podobě, proto je ještě před předáním převedena na běžný textový řetězec.

Typ parametru	Hodnota
HTTP metoda	Post
HTTP verze	1.1
Časový limit	60 sekund
Typ zprávy	Text/xml
Číslo portu	443

Tabulka 6.9: Nastavení HTTPS spojení

### 6.6.6 Náhradní řešení

Vzhledem k tomu, že je odesílání formulářů nedílnou součástí aplikace, bylo implementováno náhradní řešení. Základní myšlenkou je neodesílat data ze SAP, ale z pomocné aplikace, která je bez vědomí uživatele spuštěná na pozadí, přímo v jeho operačním systému. Jejím účelem je přečíst soubor uložený na disku, odeslat ho na zadanou adresu a uložit na disk získanou odpověď. Aby k aplikaci všichni potřební uživatelé měli přístup, lze ji uložit do společného síťového adresáře. Všichni uživatelé, kterých se odesílání dat týká, používají operační systém Windows, proto byl zvolen programovací jazyk z rodiny .NET, konkrétně jazyk C#. Požadované HTTPS spojení je poměrně snadno realizovatelné a celkově je vývoj aplikací v jazyce C# rychlý.

Na straně SAP je implementována metoda pro odesílání podání ze systému Windows. Metoda před spuštěním externí aplikace zkontroluje existenci aplikace dle cesty zadané na úvodní obrazovce a připraví vše potřebné pro odeslání podání. Samotná výměna dat mezi SAP a externí aplikací probíhá pomocí souborů ukládaných na pevný disk uživatele a skrze vstupní parametry aplikace (tabulka 6.10).

Parametr	Význam
-a1	Primární adresa serveru.
-a2	Záložní adresa serveru.
-d	Cesta k požadavku.
-o	Cesta pro uložení odpovědi.
-s	Cesta pro uložení pomocného souboru.
-l	Spuštění v ladícím režimu.

Tabulka 6.10: Vstupní parametry náhradní aplikace

Z předchozího výkladu je zřejmé, že se nejdříve musí sestavit vstupní parametry a až potom lze do počítače uživatele uložit soubor s požadavkem obsahujícím připravenou GovTalk zprávu. Poté již nic nebrání spuštění pomocné aplikace v C# provádějící následující kroky:

1. Zpracování parametrů programu. Podle argumentů uvedených ve tabulce 6.10.
2. Uložení pomocného souboru dle zadaných parametrů. Hlavním účelem je synchronizace s ABAP programem (při testování vznikala prodleva mezi spuštěním v SAP a samotným spuštěním programu).
3. Načtení odesílaného požadavku dle zadaných parametrů.

4. Navázání HTTPS spojení, odeslání načtené zprávy a čtení odpovědi. Parametry spojení, doplňované do instance vestavěné třídy `HttpRequest`, jsou stejné jako při odesílání ze SAP.
5. Uložení odpovědi do souboru.
6. Ukončení aplikace.

Zatímco se externí aplikace pokouší odeslat vytvořený požadavek, program v SAP čeká v aktivní smyčce. Nejprve čeká na vytvoření pomocného souboru jako znamení, že se program spustil, a poté určitý čas čeká na vytvoření odpovědi. Aktivní čekání je realizováno jako cyklus s určeným počtem iterací. V každé iteraci se na nějaký čas (3 sekundy) vykonávání programu pozastaví. Po opětovném spuštění se zkontroluje, zda nevznikl požadovaný soubor s odpovědí. Pokud soubor vznikl, je jako textový řetězec načten pro další zpracování do SAP. Aktivní čekání je ukončeno a veškeré pomocné soubory, vytvořené na disku uživatele, jsou voláním ze SAP odstraněny. Odpověď, opět jako textový řetězec, je předána jako návratová hodnota metody.

Případný výpadek, ať už napájení počítače uživatele či počítačové sítě nezpůsobí díky emailovému potvrzení ze strany ČSSZ a možnosti manuálního potvrzení/zamítnutí výrazné potíže. Výpadek počítačové sítě má za následek to, že externí program neuloží požadovanou odpověď na disk. Z tohoto důvodu je výše uváděné aktivní čekání omezeno počtem iterací. Nenačetl-li se soubor s odpovědí, je jako návratová hodnota vrácen prázdný řetězec.

### 6.6.7 Zpracování odpovědi

Získaná odpověď je zpracována ve třídě `ZCL_DP_CSSZ_GOVTALK`. Aplikuje se na ní XSLT transformace typu XML do ABAP. Před spuštěním transformace jsou z textové proměnné, která obsahuje odpověď, odstraněny atributy identifikující jmenné prostory. Pokud byly zahrnuty ve zdrojovém souboru, nedařilo se v SAP transformaci realizovat. Transformace je navržena tak, aby sloužila ke zpracování všech odpovědí ČSSZ, výsledkem jsou dvě interní tabulky. První je pomocná tabulka obsahující veškeré elementy z hlavičky GovTalk obálky. Jediný řádek v pomocné tabulce je pro další zpracování překopírován do vnitřní struktury objektu. Druhá tabulka slouží k načtení případných chyb při odpovědi na dotaz o stavu podání. Na základě kombinace aktuálního stavu podání a přijaté zprávy je určen nový stav podání (viz. tabulka 6.11).

Aktuální stav	Přijatá zpráva	Nový stav
Vytvořen	acknowledgement, submit	Zpracovává se
Zpracovává se	response, submit	Potvrzeno
Zpracovává se	error, submit	Zamítnuto
		Chyba

Tabulka 6.11: Aktualizace stavu podání

Podání ve stavu „zpracovává se“ je uloženo do databáze. Pokud nastala při odesílání chyba (je ve stavu „chyba“), je o tom uživatel chybovým hlášením informován, data může odeslat znovu. Do databáze se tato informace nepromítá. Typicky tato situace může vzniknout po zpracovávání prázdné odpovědi.

Vybral-li uživatel před odesláním parametr „automatické zpracování odpovědi“ je v tomto okamžiku nastaven ve funkci časovač na dobu uvedenou v obdržené zprávě (pro jistotu se

přidává 5 sekund navíc). Po uplynutí této doby je spuštěno nové vlákno, které se na pozadí dotáže na výsledek zpracování a podle něj aktualizuje stav. Pro tento účel se využívá vestavěných tříd, které SAP nabízí. Po uplynutí zadané doby se vyvolá výjimka, která spustí definovanou funkci pro automatické vyzvednutí odpovědi (viz. dále). O aktualizaci stavu se uživatel dozví grafickým znázorněním v tabulce, a také standardní hláškou ve spodní liště. Prakticky se tato možnost ale nevyužívá, protože email o přijetí/zamítnutí uživatel zpravidla obdrží dříve než za dobu uvedenou v atributu „PollInterval“. Uživatel v takovém případě využívá možnost manuálního potvrzení, respektive zamítnutí, které je k dispozici v menu obrazovky.

### 6.6.8 Dotaz na výsledek zpracování

Po odeslání podání obdrží pověřený uživatel email od ČSSZ se stavem podání, a kromě toho komunikační protokol dovoluje odeslat dotaz na stav zpracování přímo přes VREP. Dotaz se odesílá stisknutím tlačítka v menu, nebo při automatickém zpracování. Kroky při vytvoření dotazu a jeho odeslání:

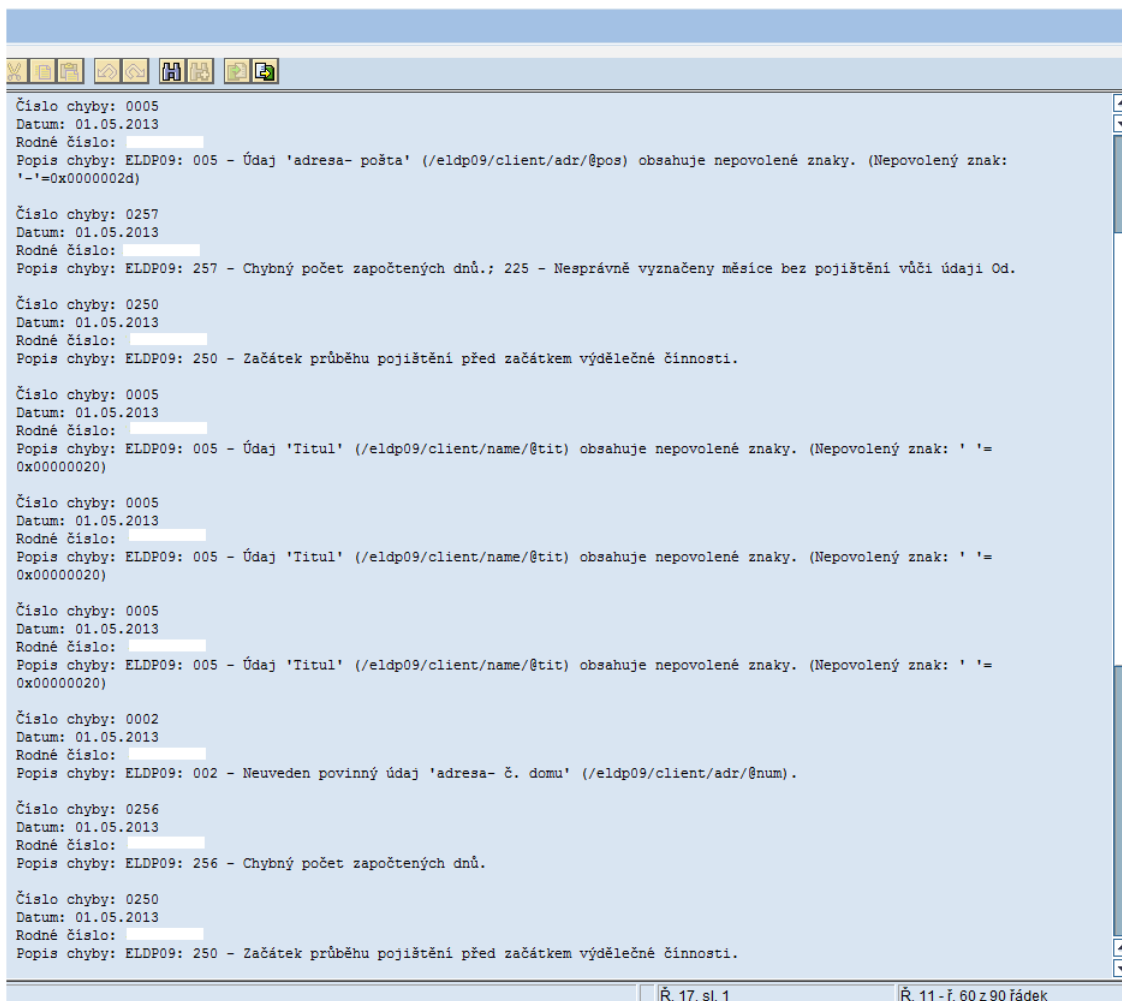
1. Kontrola, zda je podání ve stavu „zpracovává se“.
2. Vytvoření dotazu na výsledek zpracování. Dochází k naplnění parametrů GovTalk obálky a XSLT transformaci ABAP do XML.
3. Nastavení adres serverů pro dotaz. ČSSZ poskytuje pro dotaz na výsledek dvě adresy:
  - <https://vrep1.cssz.cz/VREP/poll>
  - <https://vrep2.cssz.cz/VREP/poll>
4. Odeslání podání.
5. Aktualizace stavu. Pokud bylo podání přijato/zamítnuto promítne se změna do databázových tabulek ZHR\_CSSZ a T5TELDP. Všechny záznamy zahrnuté v odeslaném podání jsou označeny jako potvrzené/zamítnuté. Nastala-li chyba v přenosu či zpracování odpovědi, je o vzniklé situaci uživatel informován, do databáze se nic neukládá.

### 6.6.9 Zobrazení chyb

Při zpracování odpovědi od ČSSZ jsou pomocí XSLT transformace případné chyby zpracovány a fyzicky uloženy do tabulky ZHR\_CSSZ\_CHYBY. Uživatel si tak může u zamítnutých podání zobrazit příčinu zamítnutí. Při ukládání chybové zprávy bylo zjištěno omezení na maximální počet znaků v jednom sloupci. SAP umožňuje uchovat nanejvýš 255 znaků. Podobné limity má i sloupec v ALV tabulce, ten dokonce zobrazí jen 128 znaků. Na vzniklou situaci bylo reagováno zvýšením počtu sloupců uchovávajících chybové zprávy a nahrazením ALV tabulky vhodně formátovaným textovým polem. Pro uložení chybové hlášky jsou nyní k dispozici 3 pole o 255 znacích.

Pro zobrazení chyb je vyhrazena samostatná obrazovka 0202. Při jejím spuštění je k dispozici již vytvořená instance třídy reprezentující e-Podání, která obsahuje přímo metodu pro načtení chyb. Jejím zavoláním se naplní interní tabulka chybami vázajícími se k zamítnutému podání. Ze získané tabulky se postupným procházením záznamů sestaví vhodně formátovaný textový řetězec, který je uživateli znázorněn v textovém poli vytvořeném vestavěnou třídou (obrázek 6.5).





Obrázek 6.5: Zobrazení chyb u zamítnutého podání

## 6.7 Hromadný tisk

Na závěr zpracování jsou všechna potvrzená podání dle zákonné povinnosti v papírové podobě předána zaměstnancům. Program ve fázi přípravy dat prochází zadaná osobní čísla a zjišťuje v tabulce T5TELDAP stav odeslání. V modulu určeném pro práci s „cluster“ tabulkou je připravená funkce, která do tisku zahrne veškerá osobní čísla, která již přijala ČSSZ (RELID je roven „SE“ a sloupec H.STATUS má hodnotu „Y“ viz. tabulka 6.7). Zároveň je poznamenán třídící klíč, na jehož základě je spuštěno makro pro načtení požadovaného formuláře. Získaný formulář je zařazen mezi ostatní do interní tabulky GT\_ELDP.

Tak jako u všech hlavních operací je před samotným zpracováním získaná tabulka seřazena dle zadání uživatele. Dle požadavku odpovídá výstupní formát stávajícímu řešení a je tak přizpůsoben rychlé jehličkové tiskárně. Záznamy jsou tisknuty na obrazovku příkazem WRITE. Výsledný výstup je po dokončení generování ručně odeslán na příslušnou tiskárnu. Celý program má přednastavené parametry definující grafický výstup. Šířka obrazovky je nastavena na 123 znaků a současně je vypnuto standardní zobrazení hlavičky stránky. Uvedené nastavení ovlivňuje všechny textové výpisy v programu.

Zvolená šířka obrazovky odpovídá šířce papíru, na který se každoročně ELDP tiskne. Celkově lze ELDP formulář přehledně zobrazit na 29 až 32 řádcích (v závislosti na počtu průběhů pojištění), proto se pro tisk se používá papír, který je oproti běžně používanému papíru ve své polovině vodorovně oddělitelný. V programu lze jednotlivé evidenční listy jednoduše oddělit příkazem `NEW-PAGE`. Při volbě tiskárny v uživatelské nabídce SAP si uživatel jen nastaví počet řádků dle použitého papíru.

Podprogram prochází v cyklu interní tabulku `GT_ELDP` a postupně tiskne hlavičku formuláře, průběh pojištění v roce a informace o organizaci. Evidenční list je rozšířen o prostor pro podpis zaměstnance a zaměstnavatele. Za každým formulářem je proveden přechod na další stránku.

## 6.8 Testování

Veškeré testování probíhalo na vývojovém systému VUT označeném VT6. K dispozici byla kopie produkční databáze z listopadu roku 2012. Zpracování ELDP je každoročně se opakující agenda, a proto jsou SAP k dispozici všechny generované ELDP z minulých let. Uložené evidenční listy by měly být správné, protože jsou potvrzené přímo ČSSZ. Přístup k uloženým datům je díky zachování původní „cluster“ tabulky stejný jako v implementované aplikaci. Testování správnosti vygenerovaných dat bylo prováděno nad zvolenými množinami osobních čísel, které byly vždy voleny podle potřeby tak, aby co nejlépe odpovídaly aktuálně testovanému problému. Podle rozsahu dat zadaných v „období výběru dat“ vygenerovala aplikace evidenční list, který poté program automaticky porovnal s uloženým potvrzeným formulářem. Případná nesrovnalost byla vypsána na obrazovku. Chybný údaj byl poté analyzován a případně konzultován se mzdovou účetní. Výsledkem byla větší oprava zdrojového kódu a následné znovu otestování. U některých případech se stávalo, že program data vygeneroval správně dle pravidel ČSSZ, ale vzhledem ke špatnému zadání v některém z infotypů docházelo k chybovému hlášení.

Pro testování komunikace přes VREP byl u ČSSZ zřízen testovací účet s vlastním variabilním symbolem. Vývojářům aplikací ČSSZ zajišťuje komunitní prostředí pro testování e-Podání. POX má stejně jako produkční systém 2 adresy pro každý typ zprávy. Díky tomu bylo možné otestovat komunikaci popsanou v [18]. Základem bylo navázání HTTPS spojení přes VREP, odeslání fiktivních dat a zpracování odpovědi. Po úspěšném získání ID korelace mohl být v další fázi odeslán dotaz na výsledek zpracování. Protože u testovacího účtu nebyl zaevidován soukromý klíč, bylo podání zamítnuto a odůvodněno uvedeným chybovým hlášením: *„Pověření k dané e-sluzbě ('CSSZ\_RELDP') není zaznamenáno v registru podávajících na OSSZ nebo certifikát, kterým je e-Podání podepsáno, není zaznamenán v registru podávajících na OSSZ. Kontaktujte pracovníka OSSZ.“* I přes zamítnutí podání může být (a měl by být) komunikační protokol řádně ukončen. Pro otestování správného ukončení se stačilo znovu dotázat na stav podání se shodným ID korelace. Vrátila-li se odpověď typu „podání s uvedeným ID korelace neexistuje“, bylo podání správně ukončeno. Závěrečné testování již probíhalo na ostrých datech. Testovalo se vždy pro malou skupinu vybraných osobních čísel tak, aby je bylo možné v nejhorším případě odevzdat ručně.

## 6.9 Programová a uživatelská dokumentace

Dle požadavku zadavatele diplomové práce je součástí řešení i uživatelská dokumentace, ve které jsou názorně popsány jednotlivé kroky u všech implementovaných operací. Doku-

mentace je rozdělena na kapitoly zabývající se každou z možných operací. Každá z těchto kapitol je slovně popsána a ilustrována obrázky.

Spolu s uživatelskou dokumentací byla vytvořena i programová dokumentace. Její základ vychází z kapitoly popisující implementaci řešení a doplňují ji automaticky generované dokumentace zdrojového kódu. Obě dokumentace se nachází na přiloženém DVD.

# Kapitola 7

## Závěr

V závěrečné kapitole této práce je zhodnoceno navržené a implementované řešení, jsou diskutovány jeho přínosy a nedostatky.

### 7.1 Zhodnocení práce

Výsledné řešení je již nasazeno v praxi a díky němu se úspěšně podařilo odeslat ELDP za rok 2012 pro všechny zaměstnance VUT. Všechny nedostatky hlášené v rámci ostrého provozu byly obratem vyřešeny. Často se jednalo o chyby pramenící ze špatného zadání informací do infotypu. Pro některé případy se proto připravuje přesný metodický pokyn s postupem zadávání. Typicky se jednalo o chyby v zadání nástupu do zaměstnání a čísla popisného u adresy zaměstnance.

Aplikace bude plně použitelná až pro ELDP za rok 2014, dosud se totiž evidenční listy pro ukončené pracovní poměry nezadávaly do SAP, ale vyplňovaly se ručně. Od chvíle uvedení této aplikace se plánuje čtvrtletní elektronické odesílání všech ukončených pracovních poměrů.

Originální HR programy mohou být každý měsíc aktualizovány podle současné legislativy vydáváním tzv. AOP. Protože se však jedná o zákaznický vytvořený program, případné změny se promítnou pouze do původního řešení SAP. Hlavní nevýhodou je tedy nutnost sledování vydaných změn ČSSZ a jejich následné ruční zanesení do aplikace. Nejde naštěstí o nic dramatického, za poslední roky se měnil jen minimální vyměřovací základ u prací malého rozsahu z 2000 Kč na 2500 Kč. Uvedenou částku lze proto v aplikaci ovlivnit přímo na hlavní obrazovce, respektive ve zvolné variantě, tedy bez zásahu do programu.

Práci se podařilo realizovat téměř podle záměrů autora. Na zjištěné problémy v konfiguraci SAP bylo vždy vhodně zareagováno (šifrování přes TrustPort a odesílání skrze externí aplikaci) a problémy byly vyřešeny. Celkově jde o užitečnou aplikaci, která je prakticky používaná a ulehčuje práci velkému počtu lidí. Pro zajímavost uvádám, že v loňském roce bylo stejné množství dat úspěšně zpracováno a odesláno až začátkem měsíce června, letos se to povedlo již 3. května.

### 7.2 Další rozšíření

Výsledné řešení je přizpůsobeno potřebám VUT a nezahrnuje proto veškeré situace, které dle ČSSZ mohou nastat. Zejména se jedná o přidělování vybraných kódů průběhu pojištění a počítání vyloučených dob při plnění vojenských povinností. Pro nasazení aplikace například

na jiných vysokých školách by bylo dobré zjistit, zda současné řešení vyhovuje i jejich potřebám a případné možnosti do řešení zahrnout.

Po přechodu VUT na novější verzi SAP (ta současná je z roku 2009) by bylo určitě dobré implementovat požadované kryptografické funkce přímo v SAP a nevyužívat tak dále externí aplikaci pro vytvoření digitálního podpisu a šifrovaných dat. Spolu s tím vytvořit vhodné úložiště pro uchování uživatelských certifikátů.

Při úvahách nad dalším rozšířením aplikace nelze nezpomenout na její odesílací část, která kvůli špatně nastavenému certifikátu neumožňuje realizovat HTTPS spojení přímo ze SAP. Proto by bylo vhodné certifikát správně nainstalovat. Nebo ještě lépe, využít pro přenos SAP NetWeaver PI a odeslání dat realizovat přes WS.

# Literatura

- [1] ABAP–CODE: SAP Adobe Forms. [online], 2007, [cit. 2012-23-12].  
URL [http://www.abap-code.com/adobe\\_forms.html](http://www.abap-code.com/adobe_forms.html)
- [2] A.Hernández, J.; Keogh, J.; Martinez, F. F.: *SAP R/3 Handbook*. Emeryville, California, USA: The McGraw-Hill, 2006, ISBN 0-07-225716-4.
- [3] Heilman, R.; Jung, T.: *Next Generation ABAP Development*. Boston, USA: SAP Press, 2007, ISBN 978-1-59229-139-7.
- [4] Kühnhauser, K.-H.: *ABAP Výukový kurz*. Holandská 8, Brno: Computer press a.s., 2009, ISBN 978-80-251-2117-7.
- [5] Maassen, A.; Schoenen, M.; Frick, D.; aj.: *SAP R/3 Kompletní průvodce*. Holandská 8, Brno: Computer press a.s., 2007, ISBN 978-80-251-1750-7.
- [6] Mandy Krimmel, J. O.: *SAP NetWeaver Process Integration*. Boston, USA: Galileo Press, 2010, ISBN 978-1-59229-344-5.
- [7] Markus Sehr: Excel with SAP - An overview. [online], 2012, on May 31 2012, [cit. 2012-23-12].  
URL <http://wiki.sdn.sap.com/wiki/display/ABAP/Excel+with+SAP+-+An+overview>
- [8] SAP AG: BC ABAP User's guide. [dokument ve formátu PDF], 1998 [cit. 2012-23-12].  
URL <http://help.sap.com/printdocu/core/print40b/EN/pdf/bcaba.pdf>
- [9] SAP AG: BC SAPscript: Printing with Forms. [dokument ve formátu PDF], 2001 [cit. 2012-23-12].  
URL <http://help.sap.com/printdocu/core/print46c/en/data/pdf/BCSRVSCRPROG/BCSRVSCRPROG.pdf>
- [10] SAP AG: SAP Smart Forms. [dokument ve formátu PDF], 2001 [cit. 2012-23-12].  
URL <http://help.sap.com/printdocu/core/print46c/en/data/pdf/bcsrvscrsf/bcsrvscrsf.pdf>
- [11] SAP AG: Field Symbols. [online], 2004, [cit. 2013-03-05].  
URL [http://help.sap.com/saphelp\\_nw04/helpdata/en/fc/eb3860358411d1829f0000e829fbfe/content.htm](http://help.sap.com/saphelp_nw04/helpdata/en/fc/eb3860358411d1829f0000e829fbfe/content.htm)
- [12] SAP AG: Example Program: Executing an HTTP Request. [cit. 2013-10-04].  
URL [http://help.sap.com/saphelp\\_nw70ehp2/helpdata/en/1f/93163f9959a808e1000000a114084/content.htm](http://help.sap.com/saphelp_nw70ehp2/helpdata/en/1f/93163f9959a808e1000000a114084/content.htm)

- [13] SAP Comunity Network: SAP NetWeaver Technology Platform. [online], 2012, [cit. 2012-30-12].  
URL <http://scn.sap.com/community/netweaver>
- [14] SAP Library: HR Logical Databases. [online], 2012, [cit. 2012-30-12].  
URL [http://help.sap.com/saphelp\\_46c/helpdata/en/c6/8a15381b80436ce1000009b38f8cf/content.htm](http://help.sap.com/saphelp_46c/helpdata/en/c6/8a15381b80436ce1000009b38f8cf/content.htm)
- [15] TrustPort: TrustPort eSign PRO. [online], 2012, [cit. 2013-03-05].  
URL <http://www.trustport.com/cz/produkty/office/trustport-esign-pro>
- [16] Wikipedia: ABAP. [online], 2012, last modified on 19 December 2012 [cit. 2012-23-12].  
URL <http://en.wikipedia.org/wiki/ABAP>
- [17] Wood, J.: *ABAP Cookbook: Programming Recipes for Everyday Solutions*. Boston, USA: Galileo Press, 2010, ISBN 978-1-59229-326-1.
- [18] ČSSZ: Podávací a dotazovací protokol pro e-podání ČSSZ, verze 1.3. [dokument ve formátu PDF], 2010, datum aktualizace 31.12.2011 [cit. 2012-23-12].  
URL <http://www.cssz.cz/NR/rdonlyres/0ACDBB35-F78A-4B1A-8258-7249B55C788E/0/csszpodavacidotazovaciprotokol20110608.pdf>
- [19] ČSSZ: Metodická pomůcka pro vyplňování ELDP. [dokument ve formátu DOC], 2012, [cit. 2012-23-12].  
URL [http://www.cssz.eu/NR/rdonlyres/64DCFF20-7D74-4A74-9E7B-B59823210AE6/0/metodicka\\_pomucka\\_od\\_112012\\_opr.doc](http://www.cssz.eu/NR/rdonlyres/64DCFF20-7D74-4A74-9E7B-B59823210AE6/0/metodicka_pomucka_od_112012_opr.doc)
- [20] ČSSZ: Profil organizace. [online], 2012, [cit. 2012-23-12].  
URL <http://www.cssz.cz/cz/o-cssz/profil-organizace/>
- [21] ČSSZ: Struktura datové věty. [online], 2012, [cit. 2012-23-12].  
URL <http://www.cssz.cz/cz/e-podani/pro-vyvojare/struktura-datove-vety/>
- [22] ČSSZ: Všeobecné zásady pro vyplňování ELDP. [dokument ve formátu DOC], 2012, [cit. 2012-23-12].  
URL [http://www.cssz.eu/NR/rdonlyres/338D46FE-0A87-4E77-BEC2-74F25389BC6C/0/vseobecnezasady\\_od\\_1\\_1\\_2012\\_1.doc](http://www.cssz.eu/NR/rdonlyres/338D46FE-0A87-4E77-BEC2-74F25389BC6C/0/vseobecnezasady_od_1_1_2012_1.doc)
- [23] ČSSZ: Datová věta ELDP09. [online], 2013, [cit. 2013-03-03].  
URL <http://www.cssz.cz/cz/e-podani/zakladni-informace/e-podani-evidencnich-listu-duchodoveho-pojisteni/definice-e-podani-eldp09-eldp12/datova-veta-eldp09.htm>
- [24] ČSSZ: Datová věta ELDP12 - příklad. [online], 2013, [cit. 2013-03-03].  
URL [http://www.cssz.cz/NR/rdonlyres/97DEE6BD-72BD-4837-93B8-5A8E2078B6EC/0/datova\\_veta12\\_priklad.png](http://www.cssz.cz/NR/rdonlyres/97DEE6BD-72BD-4837-93B8-5A8E2078B6EC/0/datova_veta12_priklad.png)
- [25] ČSSZ: ELDP – xsd schéma datové věty eldp09. [online], 2013, [cit. 2013-03-03].  
URL [http://www.cssz.cz/NR/rdonlyres/F8538962-A01D-4024-B79B-7F3A7A852ECE/0/web\\_ELDP\\_xsd\\_2.htm](http://www.cssz.cz/NR/rdonlyres/F8538962-A01D-4024-B79B-7F3A7A852ECE/0/web_ELDP_xsd_2.htm)