

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

FORMULÁŘE ADOBE V SYSTÉMU SAP

DIPLOMOVÁ PRÁCE

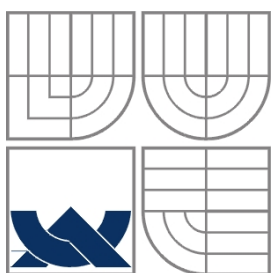
MASTER'S THESIS

AUTOR PRÁCE

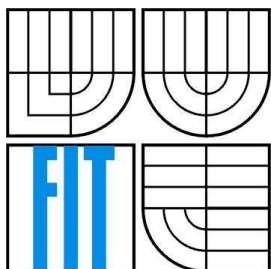
AUTHOR

Bc. MARTIN HÁS

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

FORMULÁŘE ADOBE V SYSTÉMU SAP

ADOBE FORMS IN SAP

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MARTIN HÁS

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAROMÍR MARUŠINEC, Ph.D.

BRNO 2009

Abstrakt

Tato diplomová práce byla zaměřena na studium možností vývoje aplikací v programovacím jazyce ABAP v ekonomickém informačním systému mySAP. Dále byly studovány možnosti integrace SAP s Adobe PDF formuláři a možnosti integrace Microsoft Excel a porovnány jejich výhod. V teoretické části jsou také shrnuty základní informace o technologii systémů SAP na platformě NetWeaver a charakteristika jednoho z hlavních produktů mySAP.

V praktické části je práce věnována konkrétnímu business procesu zpracování nákupní objednávky, kam spadá také likvidační list faktury. Výsledkem práce je analýza, návrh a implementace konkrétního řešení pro generování formuláře “likvidační list faktury” v systému SAP. Pro ověření vybraného řešení bylo použito vývojové prostředí SAP VUT.

Abstract

This master thesis was oriented to study of the development application possibilities in programming language ABAP in the information system mySAP. There were studied integration possibilities of SAP system with Adobe PDF forms and application MS Excel. Advantages and disadvantages of these two technologies were compared. The theoretical part of diploma thesis describes also the technology of SAP systems based on NetWeaver platform and the main product mySAP.

The practical part describes a concrete business scenario of purchase order process where is also invoice verification list included. The result of the work is analysis, design and implementation of concrete solution for “invoice verification list” generating in SAP system. A development SAP VUT application was used for implementation and testing.

Klíčová slova

SAP, mySAP, NetWeaver, ABAP, ABAP Workbench, XLS, PDF, formuláře Adobe, Adobe liveCycle Designer, Form Builder, transakce SAP.

Keywords

SAP, mySAP, NetWeaver, ABAP, ABAP Workbench, XLS, PDF, Adobe forms, Adobe liveCycle Designer, Form Builder, SAP transaction.

Citace

Hás Martin: Formuláře Adobe v systému SAP, Brno, FIT VUT v Brně, 2009

Formuláře Adobe v systému SAP

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Jaromíra Marušince, Ph.D.

Další informace mi poskytl Bc. Milan Smejkal.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jméno Příjmení
26. května 2009

Poděkování

Na tomto místě bych rád poděkoval vedoucímu diplomové práce panu Ing. Jaromíru Marušincovi, Ph.D. za odborné vedení a trpělivost v průběhu řešení této diplomové práce a Bc. Milanovi Smejkalovi za ochotu a technickou výpomoc. Především bych chtěl poděkovat všem členům své rodiny, kteří mi během celého studia vyjadřovali svoji podporu.

© Martin Hás, 2009.

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

Obsah

Obsah	1
Úvod	3
1.1 Struktura diplomové práce	3
2 Podnikové informační systémy	5
2.1 Ekonomické informační systémy	5
2.1.1 Podnikové procesy	5
3 Produkty mySAP	6
3.1 Historie společnosti SAP	6
3.2 Skupina produktů SAP	6
3.3 Ekonomický informační systém mySAP	8
3.3.1 Řešení mySAP	8
3.3.2 SAP Enterprise Portal	9
3.4 Technologické komponenty	9
4 Vývoj aplikací v jazyce ABAP	11
4.1 Historie jazyka ABAP	11
4.2 Architektura SAP systémů	11
4.2.1 Programy v jazyce ABAP	12
4.2.2 Jádro a báze služby	13
4.3 Programování v jazyce ABAP	13
4.3.1 ABAP Workbench	13
5 Formuláře Adobe	15
5.1 Adobe LiveCycle Designer	15
5.2 Kategorie Adobe formulářů.....	16
5.2.1 Interaktivní formuláře	16
5.2.2 Neinteraktivní formuláře.....	17
5.2.3 Formuláře typu “Vytiskni a vyplň“	18
6 Integrace SAP s externími formáty	19
6.1 Integrace SAP s MS Excel	19
6.2 Integrace SAP s formuláři Adobe	20
6.2.1 Integrace.....	21
6.2.2 Vývoj formulářů.....	21
6.2.3 Form Builder	22
7 Likvidační list faktury	23
7.1 Obchodní scénář	23

7.1.1	Definice nákupní objednávky	23
7.1.2	Proces zpracování nákupní objednávky	24
7.2	Analýza likvidačního listu.....	25
7.2.1	Analýza předlohy formuláře	25
7.2.2	Identifikace polí formuláře v systému SAP	28
7.2.3	Analýza zdrojových dat v back-end systému.....	29
7.2.4	Praktické použití nástroje ABAP Debugger	32
7.2.5	Sledování systému	34
7.2.6	Zdroje dat pro likvidační list faktury	36
8	Implementace Adobe formuláře.....	38
8.1	Implementace rozhraní.....	38
8.1.1	Rozhraní formuláře	39
8.1.2	Globální definice.....	40
8.1.3	Inicializace	41
8.2	Implementace formuláře	42
8.2.1	Práce s vestavěnou aplikací Adobe LiveCycle Designer	43
8.2.2	Zpracování Likvidačního listu faktury v Adobe LiveCycle Designer	46
8.2.3	Obslužný program SAP pro vyvolání formuláře	48
9	Závěr	49
	Literatura	51
	Seznam příloh	52
	Příloha 1. – Zdrojové kódy rozhraní formuláře	53
	Příloha 2. – Obslužná rutina pro vyvolání PDF formuláře z prostředí SAP	59
	Příloha 3. – Prostředí Adobe LiveCycle Designer.....	61

Úvod

Jedním z trendů v rychle se rozvíjejícím odvětví informačních technologií se v současnosti stala integrace softwarových produktů. Díky spolupráci různého software od různých výrobců se práce uživatelů výpočetní techniky stává snadnější a efektivnější. Rovněž na poli informačních systémů se do popředí dostávají produkty, které nejsou izolované od okolního software, ale dokáží spolupracovat s nejrozšířenějšími programy, které běžný uživatel využívá ke své každodenní práci na PC.

V diplomové práci jsem se zaměřil na možnost integrace ekonomického informačního systému mySAP od společnosti SAP AG s externími formáty souborů. Konkrétním cílem bylo najít vhodné řešení pro reprezentaci formulářů mimo prostředí systému mySAP, které má z velké části nahradit složité zpracovávání tištěných dokumentů. Ve výsledku bude uživatel schopen pomocí formulářů exportovaných do vhodného formátu komunikovat se systémem SAP prostřednictvím emailu nebo webu, aniž by se musel do systému hlásit.

Pro integraci se systémem mySAP jsou vhodné formáty, které jsou rozšířené mezi uživateli a dostupné na většině počítačových stanic. Mezi tyto formáty patří formáty XLS aplikace MS Excel a PDF společnosti Adobe.

1.1 Struktura diplomové práce

Práce je členěna do několika kapitol rozdělených dle tématických okruhů.

Ve druhé kapitole je uvedena základní charakteristika informačních systémů, konkrétně systémů zaměřených na ekonomiku podniku (tzv. ERP).

Třetí kapitola se věnuje produktům SAP, jejich členěním a technologií. V jedné z podkapitol je stručně uvedena historie společnosti SAP.

Čtvrtá kapitola se zabývá vývojovým jazykem ABAP a jeho možnostmi pro vývoj aplikací v systémech SAP.

V páté kapitole je popsána technologie Adobe formulářů a vývojového prostředí LiveCycle Designer.

Šestá kapitola je věnována integraci systému SAP s externími formáty, konkrétně formát XLS produktu MS Excel a PDF společnosti Adobe.

Sedmá kapitola popisuje praktickou část diplomové práce. Zaobírá se konkrétním business procesem zpracování nákupní objednávky a analyzuje likvidační list faktury ve front-end i back-end systému SAP.

Osmá kapitola se zabývá samotným vývojem interaktivního formuláře. Zahrnuje vývoj rozhraní v SAP, design formuláře v prostředí Adobe LiveCycle a vytvoření obslužné rutiny pro zobrazení formuláře.

V závěru jsou shrnuty výsledky práce a navrhnut její další vývoj.

Na konci úvodu diplomové práce se ještě zmíním, jaké kapitoly jsem převzal ze semestrálního projektu. Jedná se o teoretickou část diplomové práce, kapitoly 1 až 6.

2 Podnikové informační systémy

Standardní podnikový informační systém je tvořen aplikacemi několika typů. Aplikace jsou složeny buď ze softwarových balíčků, nebo z integrovaných, úplných balíčků. Typy aplikací [6]:

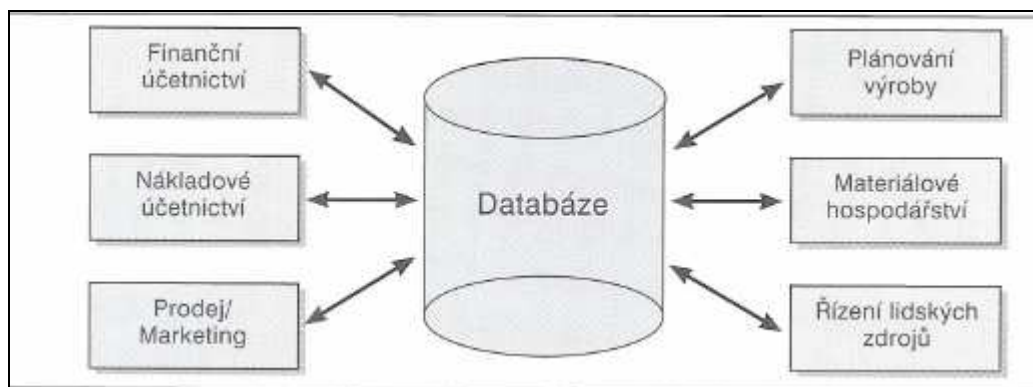
- **Kancelářské aplikace** – slouží k provedení běžných základních funkcí nezávislých na pracovním místě uživatele (např. aplikace pro zpracování textu).
- **Obchodní aplikace** – jejich funkce podporují specifická pracovní místa (např. aplikace “Prodej“, využívaná referenty v oddělení prodeje).
- **Komunikační aplikace** – zpřístupňují základní komunikační funkce (např. elektronickou poštu) a v rámci podniku jsou využívány všemi zaměstnanci bez ohledu na pracovní místo.
- **Odvětvové aplikace** – určeny k podpoře procesů specifických odvětví (např. telekomunikací či pojišťovnictví).

2.1 Ekonomické informační systémy

Ekonomický informační systém, obecně známý pod zkratkou ERP – Enterprise resource planning (volně přeloženo “Systémy pro plánování podnikových zdrojů”), je systém navržený ke koordinaci všech zdrojů, informací a aktivit, které jsou denně využívány v podnikových procesech [2].

ERP systémy podporují většinu podnikových systémů, kde pracují nad databází spravující data potřebných k provozu oblastí podniku jako je výroba, finance, lidské zdroje či vedení projektů [2].

Diagram integrace dat v ERP systému je uveden na Obr. 2.1.



Obr. 2.1: Integrace dat v ERP systému [6]

2.1.1 Podnikové procesy

Součástí podnikového procesu jsou všechny činnosti, které vedou k realizaci daného výstupu užitečného pro zákazníka. Proces má svého vlastníka, přiřazené zdroje a disponibilní čas k seberealizaci [1].

3 Produkty mySAP

3.1 Historie společnosti SAP

Společnost SAP byla založena v roce 1972 skupinou pěti zakladatelů, kteří byli původně zaměstnanci IBM, jako System Applications and Products (Systémové aplikace a produkty). Jejich cílem bylo vyvinout standardní software pro řízení podnikové ekonomiky [6].

O rok později byl dokončen vývoj prvního standardního softwaru pro oblast finančního účetnictví. Tento produkt vytvořil také základ systému SAP R/1, kde písmeno R je zkratkou ze slov Real Time-Datenverarbeitung (Zpracování dat v reálném čase) [6].

Navazujícím systémem se stal systém SAP R/2, který lze označit za první systém ERP. SAP R/2 se značně rozšířil, nicméně jeho provoz stále vyžadoval použití sálových počítačů. V roce 1988 byly akcie firmy SAP uvedeny na burzu [6].

Od roku 1992 je na trh dodávána verze SAP R/3. Ve srovnání se staršími verzemi jde o zcela přepracovaný produkt, založený na architektuře klient-server s využitím relačních databází. Systém byl upraven tak, aby bylo možné jej provozovat na hardware od různých výrobců a pod různými operačními systémy. Díky tomuto produktu dosáhla firma SAP celosvětově vedoucího postavení na trhu se standardními softwary pro řízení podnikové ekonomiky [6].

V roce 2002 byl na trh uveden SAP R/3 Enterprise. Stávající základní systém byl nahrazen produktem SAP Web Application Server (SAP WebAS). Od roku 2004 se stal centrálním produktem balík mySAP Business Suite. Technologické komponenty byly zcela odděleny od aplikačních komponent a jsou dále souhrnně označovány SAP NetWeaver [6].

Sídlo společnosti se nachází městě Waldorf v Německu a její název vznikl ze zkratky “Systeme, Anwendungen, Produkte in der Datenverarbeitung” (anglicky “Systems-Applications-Products in data processing” [2].

V současnosti je společnost SAP největším světovým dodavatelem software pro informační systémy podniků a organizací všech velikostí. Software SAP dnes používá více než 76 000 zákazníků ve 120 zemích světa [3].

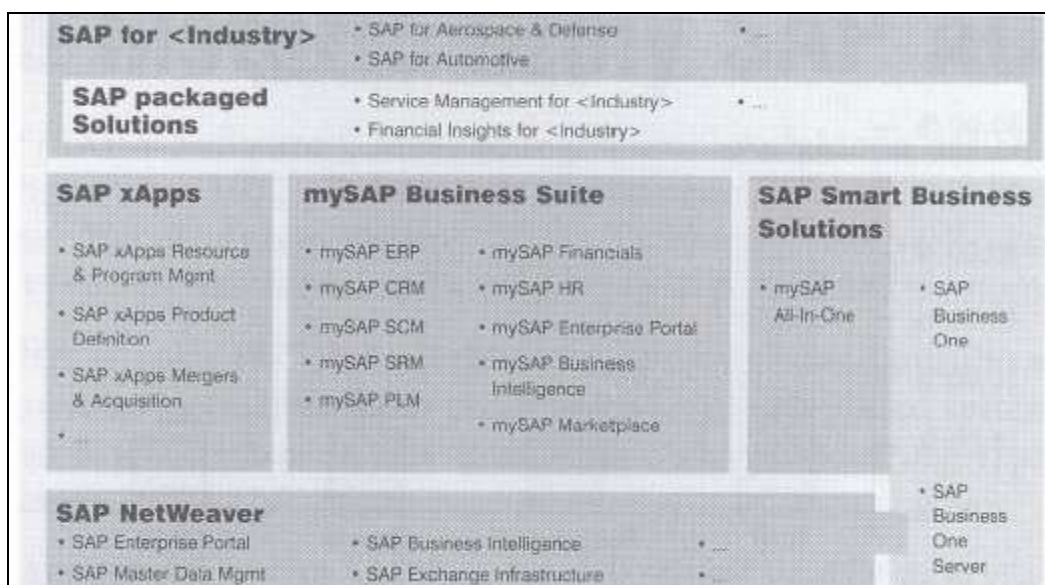
3.2 Skupina produktů SAP

Skupina produktů SAP představuje kompletní řešení především pro všechna interní oddělení podniku. Současně tato skupina pokrývá i všechny procesy, které přesahují rámec podniku. Z tohoto důvodu některé komponenty částečně přesahují rámec klasických systémů ERP [6].

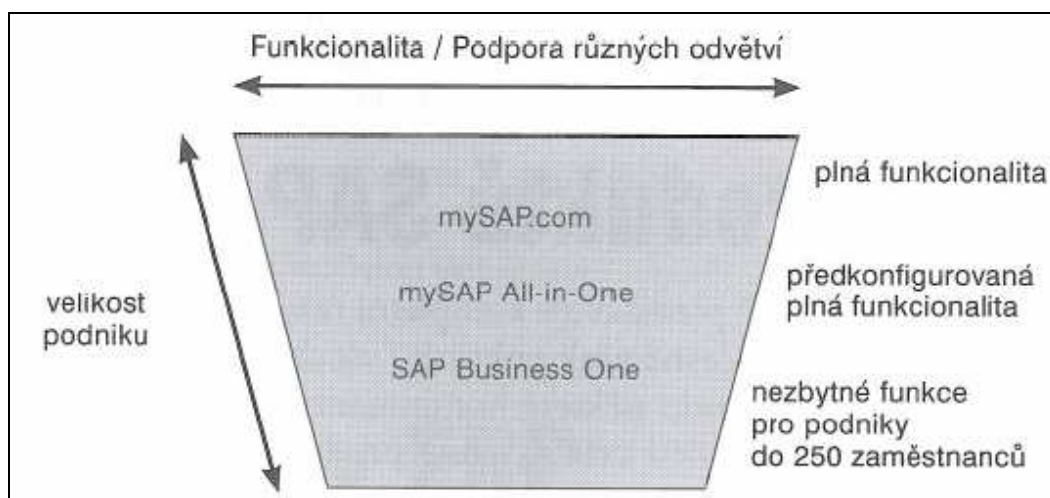
Celá skupina SAP produktů je vertikálně rozdělena do tří oblastí, viz *Obr.3.1*.

Na nejnižší úrovni se nacházejí technologické komponenty, souhrnně označované jako **SAP NetWeaver**.

V prostřední úrovni se nacházejí ty komponenty, které se vztahují k řízení podnikové ekonomiky. Tato úroveň je dále rozdělena do tří oblastí, a to v horizontálním směru. Souhrnné označení **SAP xApps** (Extended Applications) je použito pro řešení řízení podnikové ekonomiky, která přesahují rámce jednotlivých komponent a propojují ostatní komponenty pomocí xApps. Tímto způsobem lze vytvořit procesy, které probíhají několika komponentami, ale aplikace xApps vyžadují nasazení portálu *SAP Enterprise Portal*. Uprostřed se nachází skupina produktů **mySAP Business Suite**, obsahující všechny komponenty pro řízení podnikové ekonomiky. Tato úroveň je doplněna řešeními **SAP Smart Business Solutions**. Mezi nimi se nacházejí taková řešení, která jsou určena pro středně velké či menší podniky. Kromě produktu mySAP All-in-One (řešení pro středně velké podniky) sem patří i samostatný produkt SAP Business One (řešení pro malé podniky; viz Obr. 3.2). V nejvyšší úrovni se nacházejí průmyslová řešení, a to speciální dodatková řešení využívaná pouze v určitém odvětví [6].



Obr. 3.1: Skupina produktů SAP [6]



Obr. 3.2: Řešení pro malé podniky a mySAP [6]

3.3 Ekonomický informační systém mySAP

Ekonomický informační systém mySAP je významným produktem společnosti SAP na platformě NetWeaver. Kompletní název řešení je “mySAP Business Suite/Solutions“. Jde o soubor adaptivních řešení pro společnosti k optimalizaci obchodních procesů. Produkty mySAP business Suite lze aplikovat ve společnostech všech velikostí a jsou orientovány na zapojení zaměstnanců a partnerů do procesů, které přinášejí zákazníkovi potřebnou hodnotu [1].

3.3.1 Řešení mySAP

Jednotlivá řešení mySAP lze rozdělit do tří základních oblastí [1]:

1. Horizontální řešení
2. Odvětvová řešení
3. Infrastruktura a služby

3.3.1.1 Horizontální řešení

Horizontální řešení představují v mySAP komplexní soubor business řešení, která jsou klíčová pro všechna odvětví, např. řízení vztahu se zákazníky (CRM – Customer Relationship Management), řízení vztahu s dodavateli (SRM – Supplier Relationship Management), řízení dodavatelského řetězce (SCM – Supply Chain Management), podporu manažerského rozhodování (BI – Business Intelligence), finanční řízení (FI – Financials), lidské zdroje (HR – Human Resources) a mobilní business [1].

Hlavní principy přístupu aplikovaného v mySAP spočívají v důležitosti integrace e-business řešení, tj. ve spojení systémů front-end a back-office vytvářející tak nové možnosti pro sdílení informací i za hranicemi podniku a umožňující plně využívat potenciál obchodování po internetu. Tento přístup je základem popisovaného integrovaného řešení mySAP Business Suite [1].

3.3.1.2 Odvětvová řešení

Odvětvová řešení mySAP Business Suite zahrnují 23 průmyslových řešení zaměřených na jednotlivá aplikační odvětví. Patří sem široká škála oborů, např. bankovníctví, chemický průmysl, zdravotnictví, média či telekomunikace [1].

Potřeba odvětvových řešení vychází ze skutečnosti, že se jednotlivá průmyslová odvětví z hlediska podpory podnikových procesů v informačních systémech vzájemně odlišují a není možné aplikovat ve všech případech pouze jedno jednotné řešení. Výhodnější je možnost přizpůsobení řešení pro specifický průmysl [1].

3.3.1.3 Infrastruktura a služby

Tato řešení podporují a podtrhují horizontální a odvětvová řešení technologiemi a službami tak, aby byla zajištěna jejich rychlá implementace [1].

3.3.2 SAP Enterprise Portal

Jedno z hlavních rozšíření v mySAP je ve směru využití internetových technologií pro komunikaci uživatele se systémem (Enterprise portále) a komunikaci s okolím podniku, tj. zákazníky, dodavateli a partnery [1].

SAP Enterprise Portal neboli uživatelský portál umožňuje uživateli rychlý a jednotící přístup ke všem zdrojům, službám, aplikacím a informacím (strukturovaným i nestrukturovaným) z jednoho místa z browseru (prohlížeče). Z hlediska koncových zařízení k tomu lze využít buď osobní počítač (popř. notebook) a nebo některé z mobilních zařízení [1].

Jedná se o uživatelskou integraci, která zajišťuje, aby běžný uživatel po vstupu do portálu již nemusel rozpoznávat, na kterém systému (z produktů mySAP Business Suite či jiných dodavatelů) daná aplikace, transakce či služba běží [1].

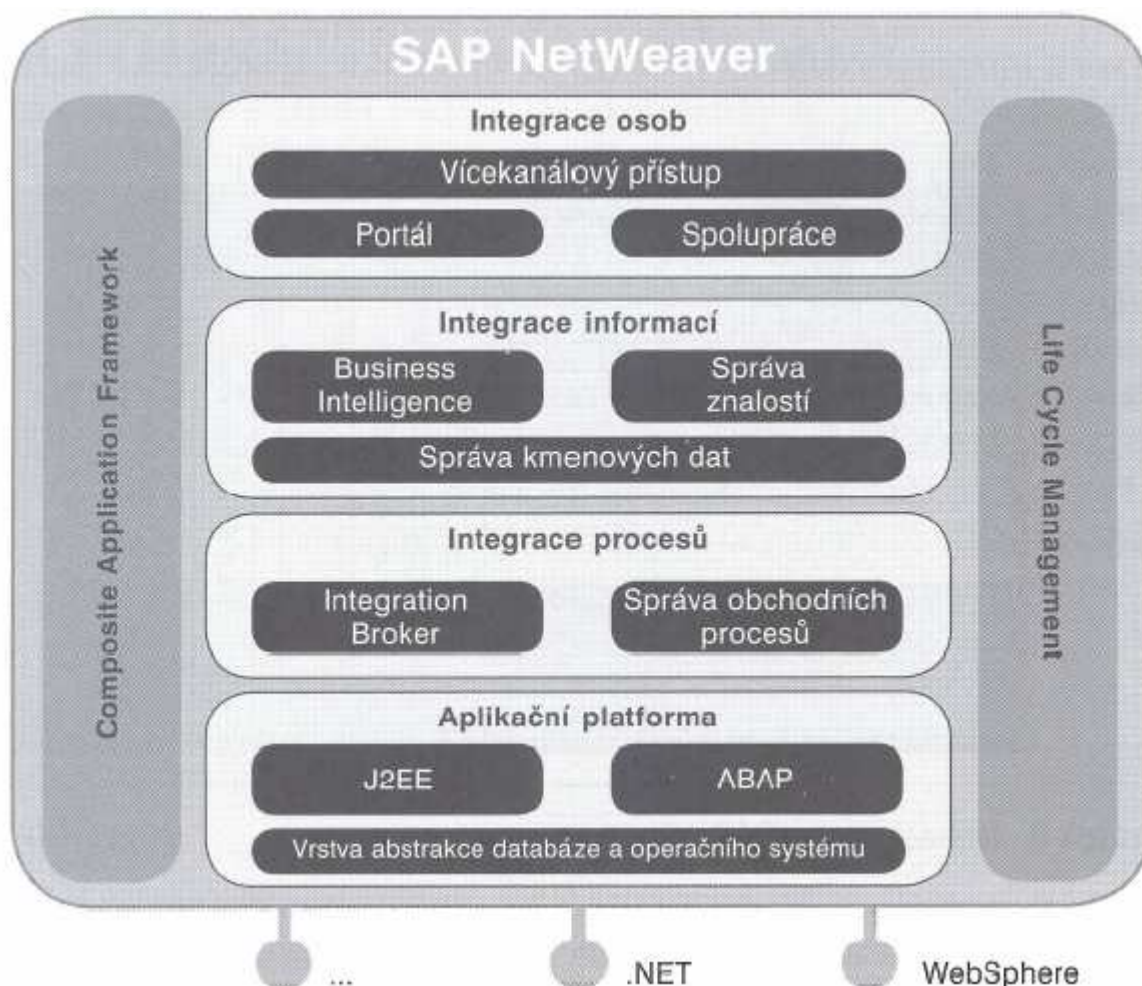
3.4 Technologické komponenty

Přechod od systému SAP R/3 ke skupině produktů mySAP zahrnuje i změnu technologických komponent, jejichž souhrn firma SAP označila pojmem SAP NetWeaver, viz Obr. 3.3, [6].

SAP NetWeaver je charakteristický především svými čtyřmi integračními úrovněmi. Do těchto úrovní byly nově uspořádány všechny níže uvedené technologické komponenty [6]:

- **Integrace osob:** Zde se nachází všechny technologické komponenty, které podporují spolupráci osob či skupin osob. SAP Enterprise Portal vytváří webovou plochu pro integraci různých požadavků. Pomocí něj je možné vytvořit jednotné přístupy pro zaměstnance, partnery, dodavatele či zákazníky. Podobně je podporována spolupráce jednotlivých skupin, a to elektronickými jednacími místnostmi či společným úložištěm dokumentů v síti.

- **Integrace informací:** Součástí této úrovně jsou komponenty podporující integraci informací. Zde lze zmínit komponentu pro správu znalostí (Knowledge Warehouse) či komponentu pro harmonizaci kmenových dat (Master Data Management).
- **Integrace procesů:** Jedná se o úroveň zajišťující integraci procesů. Její součástí jsou dvě technologické komponenty: Integration Broker a Business Process Management (Správa obchodního procesu). Patří sem rovněž nástroj pro integraci podnikových aplikací nazvaný SAP Exchange Server.
- **Aplikační platforma:** Tato úroveň nahrazuje původní komponenty báze (Basis Components) zcela novým serverem SAP WebAS (SAP Web Application Server). Současně byly technologie této úrovně zásadně inovovány. SAP WebAS nabízí otevřená webová rozhraní a lze jej nasadit i samostatně. Kromě podpory klasického programovacího jazyka ABAP/4 SAP WebAS umožňuje i programování v jazyce Java. Proto je součástí této úrovně i aplikační server odpovídající standardu J2EE.



Obr. 3.3: SAP NetWeaver [6]

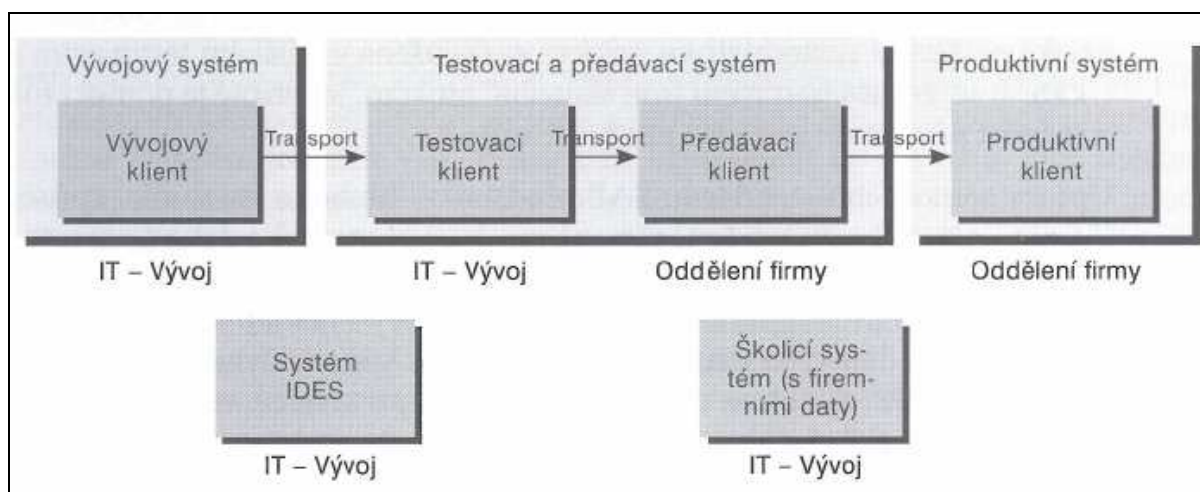
4 Vývoj aplikací v jazyce ABAP

ABAP je programovací jazyk používaný pro vývoj aplikací mySAP od firmy SAP [2].

4.1 Historie jazyka ABAP

Počátky jazyka ABAP sahají do 70. let 20. století, kdy vznikl makro assembler pro generování reportů v systémech R/2. Název ABAP vznikl z “Allgemeiner Berichts-Aufbereitungs-Prozessor”. V 80. letech se ABAP vyvinul do podoby interpretačního jazyka pro aplikace v R/2 a umožňoval programovat tzv. dialogově řízené transakce. Začátkem devadesátých let byl ABAP uveden jako “programovací jazyk 4. generace” pod názvem ABAP/4, “Advanced Business Application Programming”. Od této doby jsou aplikace pro produkty mySAP psány v ABAP/4 a pouze systémové jádro je psáno v programovacím jazyku C. Nová etapa jazyka ABAP začala kolem roku 2000, kdy bylo provedeno objektové rozšíření jazyka. Od roku 2003 se pro vývoj aplikací mySAP na vývojové platformě NetWeaver používá i jazyk Java [2].

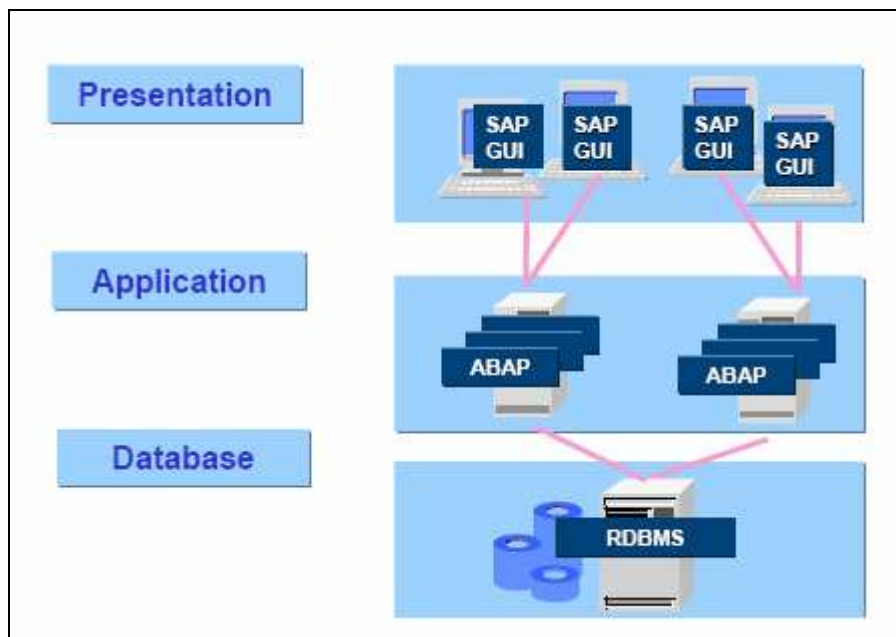
Typická organizace vývojového prostředí v systémech SAP je uvedena na Obr. 4.1.



Obr. 4.1: Organizace vývojového prostředí v systémech SAP [6]

4.2 Architektura SAP systémů

Programy vytvořené v jazyce ABAP běží na třívrstvé architektuře SAP aplikace (viz Obr. 4.2), která se skládá z prezentační, aplikační a databázové vrstvy. Prezentační vrstvu tvoří grafické uživatelské rozhraní (SAPgui) a ABAP programy běží v aplikační vrstvě. Tyto programy zpracovávají uživatelské požadavky a komunikují s nativní databází přes jednotku DBMS – Database Management System [7].



Obr. 4.2: Třívrstvá architektura v systémech SAP [7]

Výhodou vícevrstevných architektur je rovnoměrné rozložení zátěže, což vede ke zlepšení výkonu celého systému [7].

Jelikož databáze obsahují veškerá data SAP systému, je vhodné oddělit aplikační a databázový server. Třívrstvé SAP systémy se instalují každý na jiný stroj a komunikují spolu přes síťové rozhraní [7].

Vhodné je rovněž oddělit zpracovávání uživatelského vstupu a formátování datového výstupu od aplikační vrstvy. K tomuto účelu slouží prezentační vrstva. SAPgui a aplikační server jsou navrženy tak, aby se minimalizovalo množství transportovaných dat mezi těmito vrstvami. To umožňuje běh prezentační vrstvy i na počítačích s pomalým připojením k serveru [7].

4.2.1 Programy v jazyce ABAP

Specifikou jazyka ABAP je způsob uložení ABAP programů. Programy jsou v mySAP aplikacích uloženy v databázi, nikoliv v separátních externích souborech jak je tomu u programů vytvořených v jazycích Java či C++. V databázích existuje veškerý ABAP kód ve dvou formách: *zdrojový kód*, který může být zobrazen a modifikován pomocí vývojového nástroje “ABAP Workbench“ a *generovaný kód*, který je binární reprezentací zdrojového souboru. ABAP programy jsou spouštěny a řízeny runtime systémem, který je částí SAP jádra. Runtime systém je zodpovědný za zpracování ABAP příkazů. Rovněž řídí logiku toku návazných obrazovek (screens) a odpovídá na příchozí události typu stisknutí tlačítka myši či aktivace tlačítka z obrazovky programu. [2]

Klíčovou komponentou ABAP runtime systému je databázové rozhraní, které převádí databázově-nezávislé příkazy z “Open SQL“ do jazyka databáze nad kterou SAP aplikace běží (“Native SQL“). Databázové rozhraní řídí veškerou komunikaci ABAP programu s relační databází.

Rozhraní také obsahuje funkce jako je načítání často používaných dat do lokální paměti aplikačního serveru [2].

4.2.2 Jádru a базové služby

Jádru a базové služby tvoří runtime prostředí pro SAP aplikace. Runtime prostředí je z větší části napsáno v C a C++, ale obsahuje i části napsané v jazyce ABAP. Služby jádra jsou následující [7]:

- **Řízení běhu aplikací:** Všechny aplikace běží na softwarovém procesoru (virtual machine).
- **Administrace uživatelů a procesů:** Jelikož jde o multi-uživatelské prostředí, každý uživatel může spustit několik nezávislých aplikací. Komponenty jádra přebírají odpovědnost za úkoly, které má obvykle na starosti operační systém. Uživatelé se tedy přihlašují do SAP systému a spouští aplikace uvnitř systému samotného, aniž by přišli do styku s operačním systémem hostujícího počítače. Jediným uživatelem hostujícího počítače je SAP systém.
- **Řízení přístupu k databázi:** Každý SAP systém je spojen s databázovým systémem, který se skládá z báze řízení dat a databáze samotné. Aplikace nekomunikují přímo s databází, ale využívají k tomuto účelu služby jádra.
- **Komunikace:** Aplikace v SAP systému mohou komunikovat s ostatními aplikacemi přes služby jádra.
- **Monitorování systému a administrace:** Komponenty jádra rovněž obsahují programy, které umožňují monitorování SAP systému za běhu a modifikaci runtime parametrů.

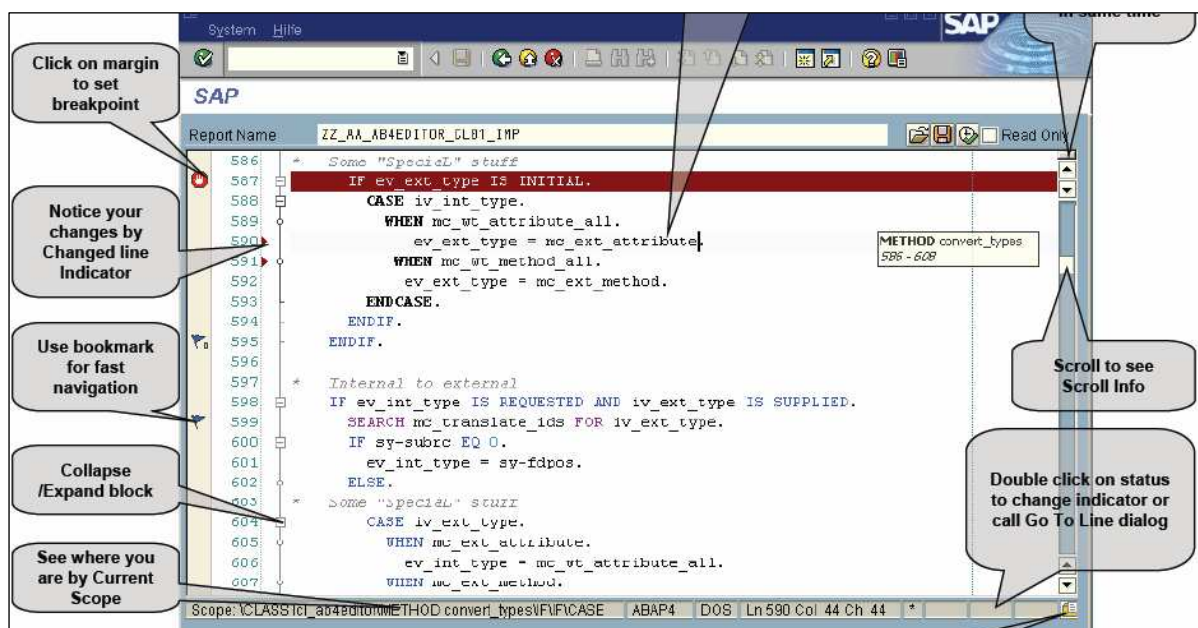
4.3 Programování v jazyce ABAP

K programování v jazyce ABAP se využívá integrované prostředí ABAP workbench.

4.3.1 ABAP Workbench

ABAP Workbench je plnohodnotné vývojové prostředí pro aplikace psané v jazyce ABAP. V prostředí je možné vytvářet, editovat a testovat aplikace a organizovat jejich vývoj. Jde o plně integrované prostředí v SAP systémech, které je stejně jako ostatní aplikace napsané v programovacím jazyce ABAP [7].

ABAP workbench obsahuje různé nástroje pro editaci objektů SAP systému. S využitím těchto nástrojů lze projít celým vývojem softwarového produktu. Jednotlivé nástroje jsou integrovány, což znamená, že např. ABAP editor (viz Obr. 4.3) rozpozná objekty vytvořené ostatními nástroji a programátor může double-klikem spustit potřebný nástroj [8].



Obr. 4.3: Prostředí ABAP editoru [8]

5 Formuláře Adobe

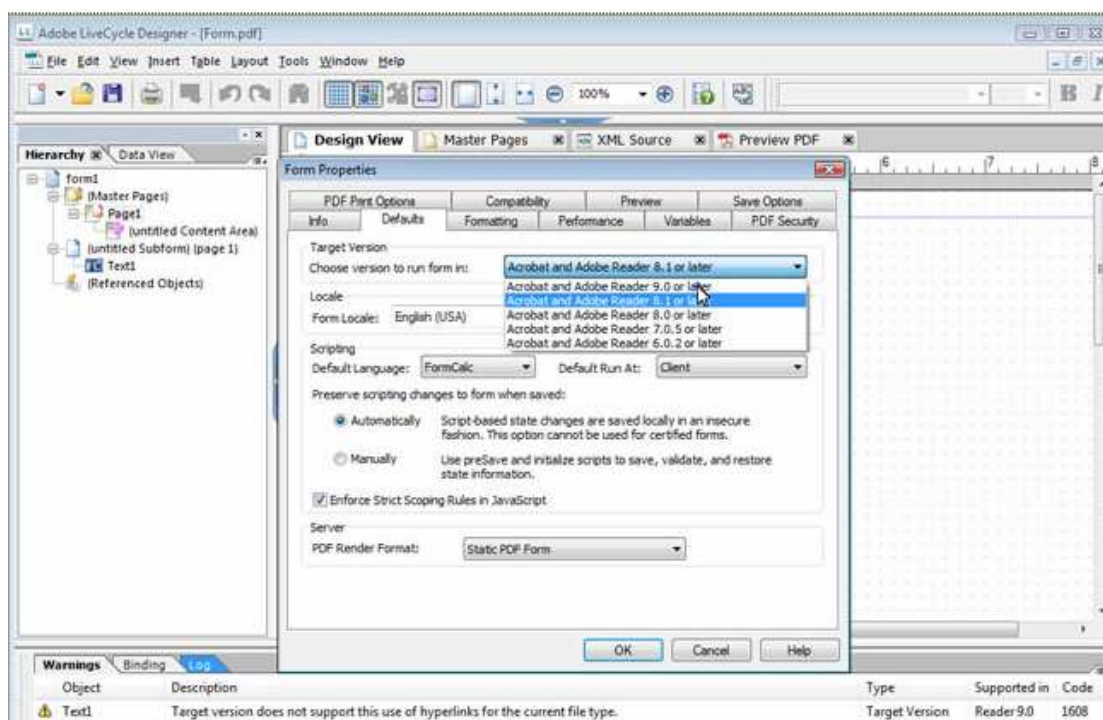
Integrace interaktivních Adobe formulářů v prostředí SAP produktů může do budoucna nahradit klasické papírové formuláře. Společnosti všech velikostí jsou závislé na formálních dokumentech, které obsahují významné obchodní informace. V této oblasti společnost Adobe spolupracuje s SAP na vytvoření ustálené automatické komunikace pomocí Adobe formulářů. Cílem je podpora zákazníků, kteří budou moci vytvářet znovupoužitelné interaktivní formuláře pro jejich business procesy [5].

Adobe Formulář je obecně dokument, který může uživatel prohlížet nebo modifikovat. Vzhled formuláře vychází z designu, který byl navržen v prostředí LiveCycle Designer [4].

Formulář typicky sbírá a prezentuje strukturovaná data a tvoří front-end rozhraní business procesu. LiveCycle Designer vytváří formuláře a dokumenty, které mohou být spojeny s obchodními daty. Výstupem jsou různé datové formáty zahrnující PDF, HTML nebo Postskript [4].

5.1 Adobe LiveCycle Designer

Adobe LiveCycle Designer je grafické vývojové prostředí pro tvorbu formulářů. Uživatelé formulářů mohou vyplnit Adobe formuláře online, potvrdit data a výsledek vytisknout, nebo prázdný vytištěný formulář vyplnit ručně. Vývojové prostředí umožňuje vytvořit libovolný design formuláře, definovat příslušnou logiku či upravit stávající formulář, aby odpovídal papírovému vzoru nebo striktním legislativním požadavkům [4]. Screenshot aplikace Adobe LiveCycle Designer je uveden na *Obr. 5.1*.



Obr. 5.1: Screenshot aplikace Adobe LiveCycle Designer

5.2 Kategorie Adobe formulářů

Adobe formuláře lze rozdělit do tří kategorií:

5.2.1 Interaktivní formuláře

Interaktivní formuláře jsou vytvořeny pro získávání informací přímo od koncových uživatelů. Uživatel vyplní formulář online a vrátí formulář s daty systému dle specificky nastaveného procesu. Uživatel může vyplnit formulář v programech Acrobat Professional, Acrobat Standard nebo Adobe Reader. Další možností je vyplnění formuláře v prostředí LiveCycle Forms, které lze integrovat ve webovém prohlížeči [4]. Pro vlastní programovací část diplomové práce byl použit interaktivní formulář zobrazený na Obr. 5.2.

Výhody interaktivních formulářů oproti papírovým jsou následující [4]:

- Interaktivní formuláře eliminují těžkopádné a časově náročné zpracování papírových formulářů.
- Interaktivní formulář může být doručen přes internet, intranet nebo email. Proces výměny dokumentů lze zautomatizovat, lze uložit formuláře v důvěryhodném formátu a ochránit obsah a integritu dokumentů.
- Interaktivní formuláře mohou být přímo zpracovány v cílovém systému (například mySAP) a získaná data integrována automaticky do databáze nebo využita programy na serveru.

- Za použití interaktivních formulářů je možné vytvořit online tok procesů, kdy budou formuláře za použití vestavěné logiky automaticky přeposílány mezi jednotlivými uživateli.

Part No.	Description	Quantity	Unit Price	Amount

Obr. 5.2: Příklad interaktivního formuláře Adobe

V základním scénáři potřebuje uživatel pouze program Adobe Reader, aby mohl elektronicky vyplnit formulář a poslat dokument jeho vlastníku nebo vytisknout formulář a dále zpracovat papírovou kopii. Součástími formuláře jsou typické interaktivní prvky jako nabídky “selection list“, “drop-down list“ a “check-boxes“, automatické výpočty, kontrolní zprávy, digitální podpisy a potvrzovací tlačítka [4].

Autor formuláře může pro rozšíření funkcionality interaktivních polí použít zabudované funkce jazyka FormCalc nebo skriptovací jazyk JavaScript. Interaktivní formuláře mohou obsahovat příkazová tlačítka, která umožní uživateli exportovat data z formuláře do souboru či databáze nebo odeslat automaticky data emailem na specifickou adresu. Autor může do formuláře začlenit i validaci dat, aby byly před zpracováním ověřeny uživatelské vstupy. Formulář může s uživatelem komunikovat např. prostřednictvím potvrzovací zprávy, aby byla potvrzena určitá specifická data [4].

5.2.2 Neinteraktivní formuláře

Adobe formuláře mohou být navrženy, aby prezentovaly informace koncovému uživateli bez možnosti modifikace. Data mohou pocházet z různých zdrojů – databází, webových služeb či různých systémů. Koncovému uživateli se zobrazí již předvyplněná data. Tyto formuláře jsou známy jako neinteraktivní. Klasickým příkladem neinteraktivního formuláře je výpis z bankovního účtu nebo telefonní účet. Možné je rovněž zkombinovat interaktivní i neinteraktivní přístup, kdy část dat je

určena pouze k prezentaci, zatímco část formuláře umožňuje uživateli zadat doplňující informace a odeslat formulář k dalšímu zpracování [4].

5.2.3 Formuláře typu “Vytiskni a vyplň”

Dalším typem jsou formuláře, které slouží jako šablona pro ruční vyplnění. Uživatel přijme typicky PDF soubor, který si vytiskne a vyplní jej manuálně. Formulář je poté odeslán faxem či klasickou poštou [4].

6 Integrace SAP s externími formáty

Možnost exportu a importu dat mezi SAP systémem a externím programem výrazně zvyšuje efektivitu práce uživatele. Uživatel SAP si vždy nevystačí pouze s aplikacemi uvnitř systému a tiskovými sestavami. Mnohdy je užitečnější a snadnější využít funkce aplikací jako je balík MS Excel či Adobe reader. Ačkoliv SAP obsahuje nástroje pro podporu jiných aplikací, ne vždy jsou tyto funkce přímo dostupné koncovému uživateli, ale vyžadují práci programátora a jistou úpravu stávajících SAP aplikací.

6.1 Integrace SAP s MS Excel

Existující řešení v SAP obsahují několik vylepšení, které umožňují v ABAP kódu manipulovat s objekty aplikace MS Excel. V jazyce ABAP lze například vytvořit aplikaci, která bude plnit data předdefinovanou šablonu formátu MS Excel XLS (viz *Obr. 6.1*).

V tomto případě bude ideální vytvořit program, který nebude přímo závislý na designu XLS souboru. Jednou z možností je editace prázdného listu šablony, kam se v jednoduchém formátu vyexportují data do jednotlivých buněk předem určených k tomuto účelu. Buňky z ostatních listů budou poté na tyto hodnoty odkazovat. Výhodou tohoto řešení je možnost změny designu Excel souboru, aniž by se modifikoval ABAP program. Naopak problém nastane, pokud uživatel potřebuje exportovat dokument s proměnným počtem řádků nebo pokud bude chtít hodnotu v souboru modifikovat – bude nutné měnit data přímo v listu, na který se z formuláře odkazuje.

Druhou možností je vytvořit ABAP program, který vytvoří celý Excel soubor včetně designu, případně bude modifikovat data v Excel souboru přímo ve formuláři. Nevýhodou je nutnost změny ABAP programu při požadavku na změnu Excel šablony, což se neobejde bez zásahu programátora.

	A	B	C	D	E	F	G	H	I	J	
1	VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ										
2											
3											
4											
5											
6											
7	LIKVIDAČNÍ LIST										
8	faktury přijaté od dodavatele a ostatních plateb hrazených v Kč										
9											
10	Číslo účetního dokladu *			Dodavatel							
11				číslo objednávky	číslo faktury	interní číslo	název				
12											
13											
14											
15	Dodavatel je plátcem DPH:				ano		Bude uplatňován nárok na odpočet DPH?				ne
16											
17	rekapitulace daňového dokladu										
18	základ daně		daň (dle daňové sazby) v Kč			přepočtený koeficient daně		zůstatek k likvidaci dle rozpisu vč. DPH			
19			0% 9% 19%								
20			X X					0,00			
21	X		X					0,00			
22	X		X					0,00			
23	halešové vyznání										
24											
25											

vzor - LL CZ karta nespotřebovaného nákupu

Obr. 6.1: Příklad Excel formuláře "Likvidační list" sloužícího jako šablona pro data z SAP systému

Transport dat z prostředí SAP do souboru MS Excel umožňuje například funkční modul MS_EXCEL_OLE_STANDARD_DAT.

S pomocí tohoto funkčního modulu lze vyplnit existující Excel list daty z interní ABAP tabulky. Příklad reportu exportujícího data do excelovského souboru se nachází v příloze A. [9]

Další možností je použití knihovny OLE2[10]. Zvolení vhodného přístupu k řešení je součástí implementační části.

6.2 Integrace SAP s formuláři Adobe

Produktem strategického partnerství společností SAP a Adobe jsou integrované SAP Adobe formuláře. Jde o poslední řešení týkající se formulářů v systémech SAP, uvolněné s platformou SAP Netweaver 2004 [11].

Interaktivní PDF formuláře ukládají uživatelem vyplněná data v XML formátu. Jakmile SAP systém PDF formulář přijme, je schopen XML data extrahovat a dále zpracovat. V opačném směru SAP systém naopak generuje PDF dokument, který sloučí s XML daty [12].

Hlavní funkce SAP interaktivních formulářů jsou následující:

- Vytváření šablon a tvorba dokumentů vycházejících z těchto šablon.
- Editace formulářů – online i offline.
- Zasílání formulářů se zabezpečením.

- Kontrola konzistence.
- Digitální podpisy pro zvýšení bezpečnosti.
- A další pokročilé funkce jako komentáře ve formulářích apod.

SAP Adobe formuláře doznaly znatelných vylepšení v porovnání s původní technologií Smartforms, kterou by měli do budoucna plně nahradit. Nový koncept interaktivních formulářů umožňuje využití, která byla v předchozí technologii nemožná [11].

6.2.1 Integrace

SAP interaktivní Adobe formuláře jsou součástí SAP Webového Aplikačního Serveru a mohou být použity libovolnou aplikací, která je založena na technologii SAP NetWeaver. [12]

Součástí řešení je software Adobe LiveCycle Designer integrovaný v SAP NetWeaver Developer studio a Adobe Dokument Services, což je webová služba pro generování PDF formulářů v runtime prostředí. V interaktivních scénářích služba extrahuje XML data z formuláře a zasílá je do systému [12].

Adobe Dokument Services běží na platformě SAP J2EE. Tzn., že musí být v systému nainstalována podpora pro Javu na SAP webovém aplikačním serveru [12].

6.2.2 Vývoj formulářů

Interaktivní formuláře v SAP mohou být vytvořeny dvěma způsoby:

- Interaktivní formuláře ve Web Dynpro (Dynamic program) pro technologii JAVA v prostředí SAP Netweaver Development Studio
- Interaktivní formuláře ve Web Dynpro (Dynamic program) pro ABAP v prostředí ABAP Workbench

Nezbytnými součástmi pro vývoj formulářů jsou v obou případech instalace programů Adobe LiveCycle Designer a Adobe Reader [11].

K vývoji PDF formulářů se používá nástroj Form Builder integrovaný v ABAP Workbench, který umožňuje vytvoření kompletního formuláře bez větších znalostí programování.

V nástroji Form Builder jsou umožněny následující činnosti:

- **Vytvoření rozvržení formuláře:** Za použití Adobe LiveCycle Designer je možné navrhnout stránky formuláře a vytvořit design výsledného vzhledu.
- **Specifikace dat použitých ve formuláři:** Zde se specifikuje, která data, tabulky, texty a grafika budou propojeny s aplikacemi SAP.

- **Migrace Smart formulářů:** Form Builder umožňuje import formulářů původní technologie SmartForms a jejich převedení do PDF formulářů. Migrační nástroj je k dispozici v transakci SMARTFORMS.
- **Import existujících PDF souborů a formulářů:** Slouží k importu formulářů vytvořených v samostatné instalaci LiveCycle Designer.
- **Programování funkcí:** Za použití JavaScript nebo FormCalc je možné naprogramovat určité funkce, jako jsou výpočty či validace dat. Funkčnost je stejná jako v klasické instalaci Adobe LiveCycle Designer.
- **Tisk a archivace PDF formulářů:** Je možné nakonfigurovat parametry výstupu tak, aby se při tisku v jednotlivých modulech systému použily Adobe PDF formuláře na místo standardního výstupu.

6.2.3 Form Builder

Nástroj Form Builder integrovaný v prostředí ABAP workbench umožňuje kompletní tvorbu Adobe PDF formulářů. Pro vývoj s nástrojem Form Builder uživatel potřebuje autorizaci pro objekt S_DEVELOP a instalaci aplikace Adobe LiveCycle Designer na počítači, z kterého přistupuje do systému SAP.

Proces vytvoření formuláře lze shrnout následovně:

1. Vytvoření objektu typu formulář. K tomuto účelu slouží transakce SFP.
2. Přiřazení rozhraní (Interface), které definuje původ dat, které se zobrazí ve formuláři.
3. Aktivace přiřazeného rozhraní.
4. Přiřazení parametrů z rozhraní, které budeme využívat ve formuláři.
5. Vytvoření obsahu formuláře v integrovaném prostředí Adobe LiveCycle Designer.
6. Aktivace formuláře.

7 Likvidační list faktury

Cílem diplomové práce bylo vytvořit Adobe formuláře, jehož datovým zdrojem jsou data ze systému SAP, které definují likvidační list faktury. V současnosti se tento dokument vyplňuje ručně. Pro zpracování likvidačního listu do elektronické podoby v podobě interaktivního Adobe formuláře je třeba provést následující kroky, které jsou popsány v následujících kapitolách:

- 1) Seznámení se s business procesem zpracování nákupní objednávky.
- 2) Analýza existujícího dokumentu likvidační list faktury.
- 3) Identifikace údajů k likvidačnímu listu faktury v SAP front-end aplikaci.
- 4) Analýza datových zdrojů v back-end prostředí SAP systému.
- 5) Implementace rozhraní Adobe formuláře pro komunikaci se systémem SAP.
- 6) Implementace Adobe formuláře likvidačního listu faktury v prostředí Adobe LiveCycle Designer.
- 7) Implementace obslužné rutiny v systému SAP řízení Adobe formuláře.
- 8) Otestování funkčnosti interaktivního formuláře, porovnání s předlohou.

7.1 Obchodní scénář

Likvidační list faktury je součástí procesu zpracování nákupní objednávky. Konkrétní implementace likvidačního listu je úzce spjata s pracovním procesem ekonomického oddělení VUT. Následující kapitoly popisují tento proces a definují objekty z hlediska aplikace v SAP, které se vyskytují v kontextu s tímto procesem. V aplikaci SAP je ústředním prvkem transakce ME23N – Zobrazení nákupní objednávky. Většina níže zmiňovaných údajů z ekonomické terminologie se zde nachází. Ostatní lze dohledat v přidružených transakcích (kmenový záznam dodavatele, nákladové středisko apod.).

7.1.1 Definice nákupní objednávky

Nákupní objednávka je formální výzvou dodavateli k dodání určitého materiálu nebo služby za udaných podmínek a zahrnuje tyto údaje:

- dodavatel,
- skupina nákupu (nákupčího, ten kdo objednává),
- objednávané zboží, materiál nebo službu,
- množství,
- cenu,
- termín dodávky a podmínky dodávky,
- platební podmínky.

Objednávka stanoví, zda bude objednávaný materiál převzat na sklad nebo půjde do spotřeby. Slouží jako podklad pro zpracování, příjem zboží a likvidaci faktury (tzv. likvidace došlé faktury s referencí k objednávce) [13].

7.1.1.1 Typy nákupů

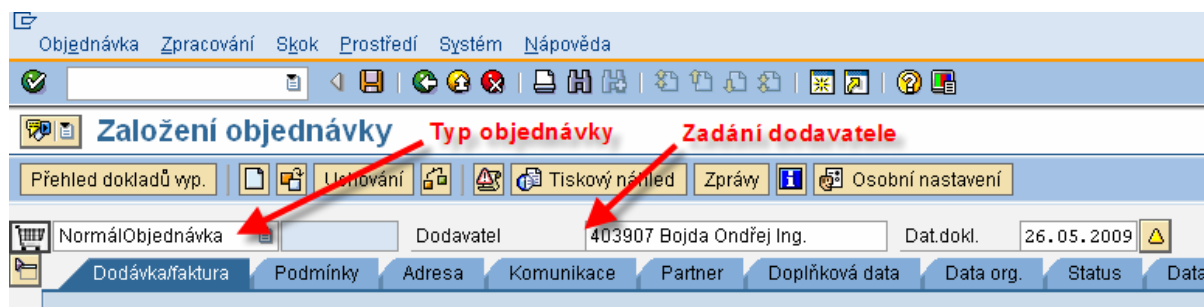
- 1 Nákup do spotřeby na nákladové středisko nebo na projekt (kód přiřazení účtu K). Povinné objekty controllingu – uvádí se vždy zdroj financování. Účtuje se na nákladové středisko nebo prvek SPP (projekt). Materiál nebo služba může nebo nemusí být vázána na kmenový záznam materiálu.
- 2 Nákup dlouhodobého majetku – investice (kód přiřazení účtu A). Povinný objekt controllingu je IM – karta nedokončené investice, která má v sobě definici zdroje financování a prvku SPP.
- 3 Nákup materiálu na sklad (kód přiřazení – prázdné pole).
Objednávka nákupu materiálu na sklad neobsahuje přiřazení nákladových prvků [13].

7.1.2 Proces zpracování nákupní objednávky

7.1.2.1 Založení nákupní objednávky

Nákupní objednávka se zakládá v transakci ME21N. Objednávka může být založena ručně uživatelem nebo jako podklad může sloužit požadavek na objednávku, který se zakládá v transakci ME51N. V případě existujícího požadavku na objednávku se tento dokument přetáhne do formuláře pro založení v objednávky a některá pole se z něj zkopírují do nové objednávky.

Standardně se ponechává normální objednávka (interní pro dodavatele v rámci VUT) a vyplní se číslo dodavatele.



Obr. 7.1: Založení nákupní objednávky v systému SAP (transakce ME21N)

Uživatel ve formuláři postupně vyplní hlavičku objednávky, která obsahuje obecné údaje, přehled položek o základních údajích o objednávaných produktech/službách (číslo položky, počet, cena, datum dodání, ...), a detail položek, kde jsou ostatní údaje o položce.

Po správném vyplnění všech údajů se objednávka uloží. Tisk objednávky se standardně provádí při uložení objednávky. Pozdější vyvolání tisku je možné z transakce ME9F.

7.1.2.2 Vývoj nákupní objednávky

K nově vytvořené objednávce je dodavatelem vytvořena faktura k nakupovanému zboží. Ta je mimo systém SAP dodána na ekonomické oddělení VUT. Zde je vystaven likvidační list faktury, který slouží jako podklad pro likvidaci faktury v transakci MIRO – Založení došlé faktury. Aby mohlo dojít k likvidaci faktury, je nejprve třeba schválit likvidační list faktury na příslušných místech. Likvidační list svými podpisy opatřují odpovědný řešitel zakázky, příkazce, správce rozpočtu a následně účetní při zaúčtování došlé faktury.

7.2 Analýza likvidačního listu

Stěžejní částí práce byla analýza likvidačního listu zahrnující dvě části. V první části bylo třeba zanalyzovat existující předlohu pro likvidační list faktury, seznámit se se vztahy mezi daty v rámci formuláře a vytvořit návrh pro zpracování dokumentu v Adobe LiveCycle Designeru. Druhá část zahrnovala správnou identifikaci požadovaných údajů pro budoucí tisk ve formuláři, jejich dohledání a interpretaci v back-end systému SAP.

Jako podklad pro analýzu byla použita předloha ve formátu XLS aplikace MS Excel. Předloha obsahuje názvy polí, požadované rozložení a definované výpočetní vzorce, které probíhají interaktivně nad daty formuláře.

Analýza “papírové” předlohy byla provedena následujícími kroky:

- Analýza polí v předloze formuláře
- Identifikace vstupních polí formuláře ve front-end aplikaci SAP.
- Analýza front-end aplikace SAP a vyhledání zdrojů dat v back-end systému.

7.2.1 Analýza předlohy formuláře

Součástí zadání úkolu byl formulář ve formátu XLS, který se skládal z ručně editovaných polí a polí, které se dopočítávaly dle předem určených vzorců. Ručně editovaná pole se dále dělila na pole, jejichž hodnota se získala z aplikace SAP a pole s hodnotou z jiných zdrojů (např. faktura dodavatele). Obsahem této kapitoly je souhrn všech polí formuláře a určení jejich účelu a zdroje pro hodnotu. Pole čistě textového charakteru jsou v tomto souhrnu pominuta.

7.2.1.1 Analýza hlavičky dokumentu

Název pole hlavičky	Zdroj dat	Možnost editace	Popis
Číslo pracoviště	SAP	Ano	
Název pracoviště	SAP	Ano	
Číslo účetního dokladu	Ručně	Ano	Vyplňuje ručně ekonomické oddělení
Číslo objednávky	SAP	Ne	
Číslo faktury	Ručně	Ano	Číslo faktury dodavatele
Interní číslo	SAP	Ne	
Název	SAP	Ne	
Plátce DPH	SAP	Ano	
Odpočet DPH	Ručně	Ano	

Výstupem analýzy hlavičky předlohy formuláře je výše uvedená tabulka. Pole v hlavičce jsou jednoznačná. Při konzultaci byl vznesen požadavek, aby v prvotní části užívání formuláře byla modifikovatelná i většina polí dotažených z aplikace SAP. Výjimku tvoří číslo objednávky a dodavatel, což jsou údaje, které jednoznačně určují, ke které objednávce likvidační list náleží.

7.2.1.2 Analýza rekapitulace daňového dokladu

Tabulka rekapitulace daňového dokladu počítá součty pro různé daňové skupiny (obr. 6.2). Obsahuje hodnoty pro tři nejčastěji využívané daňové sazby. Sumy počítá z rozpisu položek. Pole se vypočítávají podle následujících vzorců:

- Základ daně – počítá sumu základů daně, a to následovně.
 - o V prvním řádku je suma výše likvidace pro položky bez daně.
 - o V druhém řádku se počítá suma základů daně pro všechny položky, kde daňová sazba činí 9 procent.
 - o V třetím řádku se pak počítá suma základů daně pro položky, kde DPH činí 19 procent.
- Daň (dle daňové sazby v Kč) – pole počítá sumu všech daní pro určitou daňovou sazbu.
 - o Pole 0% standardně neobsahuje žádnou hodnotu, protože daň pro nulovou sazbu by neměla existovat.
 - o Pole 9% počítá sumu všech daní u položek, kde je daň stanovena na 9%.
 - o Pole 19% počítá sumu všech daní u položek, kde je daň stanovena na 19%.
- Přepočtený koeficient daně – vypočítá se na řádku jako daň / základ daně.

- Zůstatek k likvidaci dle rozpisu vč. DPH – počítá se na řádku tabulky jako součet základu daně + daň – suma požadované výše likvidace vč. DPH na položce pro odpovídající daňové procento (0%, 9%, 19%).

Krom vypočítaných polí se do této tabulky ručně vkládá haléřové vyrovnání ke korekci chyb vzniklých zaokrouhlováním.

základ daně	daň (dle daňové sazby) v Kč			přepočtený koeficient daně	zůstatek k likvidaci dle rozpisu vč. DPH
	0%	9%	19%		
		X	X		0,00
	X		X		0,00
	X	X			0,00
haléřové vyrovnání					

Obrázek 7.2 Tabulka rekapitulace daňového dokladu

7.2.1.3 Rozpis úhrad

Rozpis úhrad se skládá z polí dotažených z SAP aplikace a z polí z nich vypočítaných. Ručně se zde vyplňuje případně pouze pole poznámka.

Pole dotažená ze SAP aplikace:

- Zdroj
- Nákladové středisko
- SPP prvek
- Položka
- Daňová sazba položky
- Požadovaná výše likvidace vč. DPH (dopočítává se v programu SAP)

Vypočítaná pole včetně vzorců:

- Základ daně – vypočítá se z podílu: požadovaná výše likvidace vč. DPH / přepočtený koeficient daně z tabulky rekapitulace daňového dokladu povýšený o 1. Odpovídající koeficient se určí podle daňové sazby na řádku položky.
- Daň – vypočítá se násobku: vypočtený základ daně * příslušný přepočtený koeficient.
- Skutečně účtovaný náklad tř. 5xx xxx – Vyplňuje se pouze pokud je zvoleno “ano” v poli nárok na odpočet DPH na hlavičce formuláře. Vypočítá se ze sumy vypočteného základu daně a daně.
- Položka (2.) – Pole položky je rozděleno na dvě části. Ty se obvykle rovnají. Výjimku tvoří kombinace, kdy je dodavatel plátcem DPH a zároveň nebude uplatňován nárok na odpočet DPH. Pak bude druhé číslo položky automaticky nastaveno na účet „343 xxx“,

pokud je daňová sazba 0 procentní nebo na účet „343 105“ pokud je daňová sazba 5ti procentní.

7.2.1.4 Rekapitulace rozúčtování

V rekapitulaci jsou vypočítány mezisoučty pro sumy základů daní a daní zvlášť. Částky jsou opět rozděleny podle daňových sazeb. Konečnou částku pak zobrazují pole „k likvidaci celkem“ a „k úhradě celkem“, kde je zohledněna případná dřívější platba zálohy.

7.2.1.5 Zápatí formuláře

Zápatí formuláře definuje místo pro podpisy jednotlivých osob, které zodpovídají za zpracování a schválení likvidačního listu.

7.2.2 Identifikace polí formuláře v systému SAP

Zdrojem dat pro “Likvidační list faktury” je SAP transakce ME23N – Zobrazení nákupní objednávky. Pro testovací účely lze také využít transakce ME21N (vytvoření nákupní objednávky) a ME22N (editace nákupní objednávky).

The screenshot displays the SAP ME23N transaction interface. At the top, the title bar reads 'NormálObjednávka 351200002 založil Mgr. Anna Kolouchová'. Below this, a toolbar contains icons for document overview, print, and personal settings. The main header area shows the document type 'NormálObjednávka', document number '351200002', supplier '100432 BCS, S.R.O.', and document date '19.01.2004'. A table titled 'Hlavička' (Header) lists the items, with the first item being 'Bulletin Grantové agentury' (Material: 10, Unit: K, Quantity: 1, Date: 02.02.2004). Below the table, there are icons for document actions and a 'Dodat dispozice' button. The 'Položka' (Item) section shows the selected item '[10] Bulletin Grantové agentury'. At the bottom, the 'Data materiálu' (Material Data) tab is active, displaying a table with columns for material, quantity, percentage, unit price, and other details. The table shows a single row for material '1' with a quantity of '1,000' and a unit price of '12250'.

S...	Pol.	P	Materiál	Kr.text	Obj.množství	JOC	na	ZMJ	T	Dat.dodáv.	Ce
	10	K		Bulletin Grantové agentury	1	KS	1	KS	D	02.02.2004	

S...	A...	Množství	Procent	Nákl.stř.	Účet HK	PrÚs	Zákazka	Prvek SPP	Místo vykládky
	1	1,000	0,0	12250		12	1100		

Obr.7.3: Nákupní objednávka v systému SAP

Identifikaci polí provedeno za spolupráce zkušeného uživatele, či aplikačního specialisty. Názvy polí v “papírovém” formuláři nejsou v mnoha případech relevantní a popisy polí se mohou dle

implementace či verze aplikace lišit. Některá pole se rovněž nenachází přímo v transakci nákupní objednávky.

Přiřazení polí z formuláře k polím aplikace je uvedeno v tabulkách *Tab 7.1* a *Tab. 7.2*. Sloupce tvoří po řadě název pole ve formuláři, název SAP transakce (slouží pro lokalizaci pole v aplikaci), cesta v okně aplikace (konkrétní záložka) a přesný název odpovídajícího pole v SAP aplikaci. Ve sloupci transakce je uvedena na prvním místě transakce, kde se pole nachází ve vztahu k nákupní objednávce, na druhém místě pak hlavní transakce objektu obsahující podrobnější informace o daném atributu.

Tab. 7.1: Hlavička formuláře

Název pole	Transakce SAP	Záložka	Název v aplikaci SAP
Číslo pracoviště	ME23N/KS13	Položka/Přiřazení účtu	Nákladové středisko
Název pracoviště	KS13	Základní obrazovka	Popis nákladového střediska
Číslo objednávky	ME23N	Hlavička objednávky	Číslo nákupního dokladu
Interní číslo	ME23N/MK03	Hlavička objednávky	Dodavatel
Název	ME23N/MK03	Hlavička objednávky	Dodavatel
Dodavatel je plátcem DPH	MK03	Řízení	DPH

Tab. 7.2: Řádek formuláře

Název pole	Transakce SAP	Záložka	Název v aplikaci SAP
Zdroj	ME23N/CO03	Položka/Přiřazení účtu	Zakázka
Nákladové středisko	ME23N/KS13	Položka/Přiřazení účtu	Nákladové středisko
SPP prvek	ME23N/CJ13	Položka/Přiřazení účtu	Prvek SPP
Položka	ME23N/FBL3N	Položka/Přiřazení účtu	Účet HK
Požadovaná výše likvidace vč. DPH	ME23N	Položka	Cena netto (+DPH)
Daňová sazba položky	ME23N	Položka/Faktura	Znak daně

Tabulky *Tab 7.1* a *Tab. 7.2* definují viditelná pole, která vstupují do formuláře z pohledu uživatele SAP aplikace.

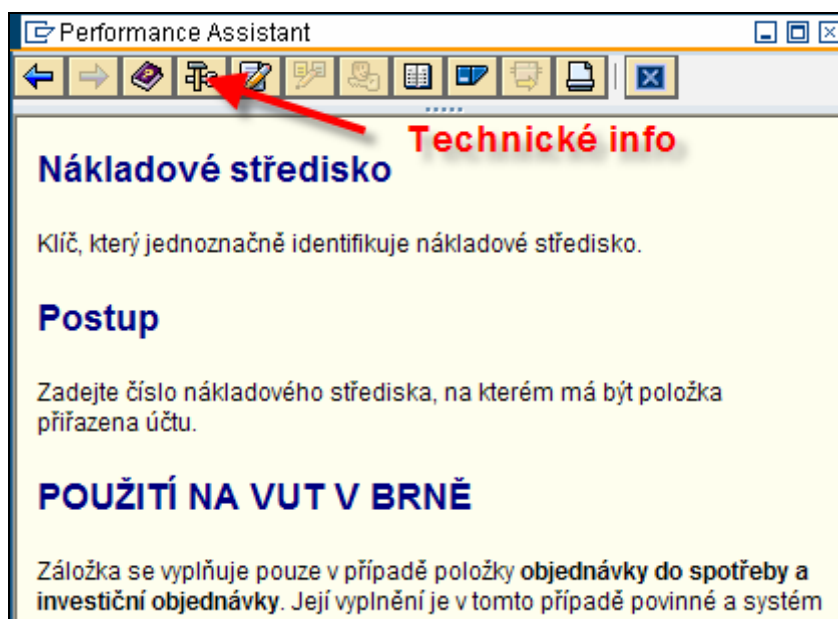
7.2.3 Analýza zdrojových dat v back-end systému

V dalším kroku byla provedena analýza struktury dat v back-end systému.

Uživatelé jsou data v SAP transakci prezentována ve formě propojených obrazovek, což ovšem nijak nevypovídá o struktuře databázové vrstvy. Pro vzhledání tabulek a datových polí, které poslouží jako datový zdroj pro Adobe formulář, nabízí SAP aplikace níže uvedené nástroje:

- Technické informace.
- Přehled použití datových polí.
- ABAP Debugger.
- Sledování systému (transakce ST01).

Technické informace lze zobrazit přímo z okna transakce, a to vyvoláním přes tlačítko F1 (help), které zobrazí informace o aktuálním poli s aktivním kurzorem, viz Obr. 7.4. Tyto informace usnadňují uživateli orientaci v aplikaci. Jednou z funkcí je tlačítko „Technické info“, viz Obr. 7.5. Technické informace jsou rozděleny do několika částí odkazujících se na řídicí program transakce, či číslo aktivní obrazovky. Z hlediska analýzy aplikace pro potřeby Adobe formulářů je významná oblast „Data pole“. V ideálním případě se v kolonce „Název tabulky“ nachází jméno tabulky logické databáze, ovšem ve většině případů (včetně Obr. 7.5) SAP aplikace ukládá data při inicializaci transakce do struktur, které více vyhovují potřebám programu. Ze struktur název zdrojových tabulek přímo vyčíst nelze.



Obr.7.4: Informace o poli aplikace SAP. Nápopěda nad polem je k dispozici přes tlačítko F1

Data dynpra	
Název programu	SAPLMEACCTVI
Číslo obrazovky	1000

Data GUI	
Název programu	SAPLMEGUI
Status	NORMAL

Data pole	
Název tabulky	MEACCT1000
Název pole	KOSTL
Náp.vyhled.	KOST
Datový prvek	KOSTL
Dod.dat.prvku	5
ID parametru	KOS

Označení pole pro batch-input	
Pole dynp.	MEACCT1000-KOSTL
Název programu	SAPLMEGUI
Č.obrazovky	0014

☒ Navigace
 ☐

Obr. 7.5: Technické informace nad polem “Nákladové středisko”

Mezi důležité údaje funkce “Technické info” patří “Název pole“, resp. “Datový prvek“. Datový prvek je definicí pole použitého ve struktuře programu a stejně tak určuje i definici pole v databázové tabulce. Dvoj-klikem na jméno datového prvku, či přes tlačítko navigace lze datový prvek zobrazit, viz Obr. 7.6.

Dictionary: Zobraz.datového prvku Přehled použití

Datový prvek: KOSTL Aktiv.

Krátký popis: Nákladové středisko

Vlastnosti **Datový typ** **Doplň.vlast**

☒ Základní typ
☒ Doména

Datový typ: KOSTL

Délka:

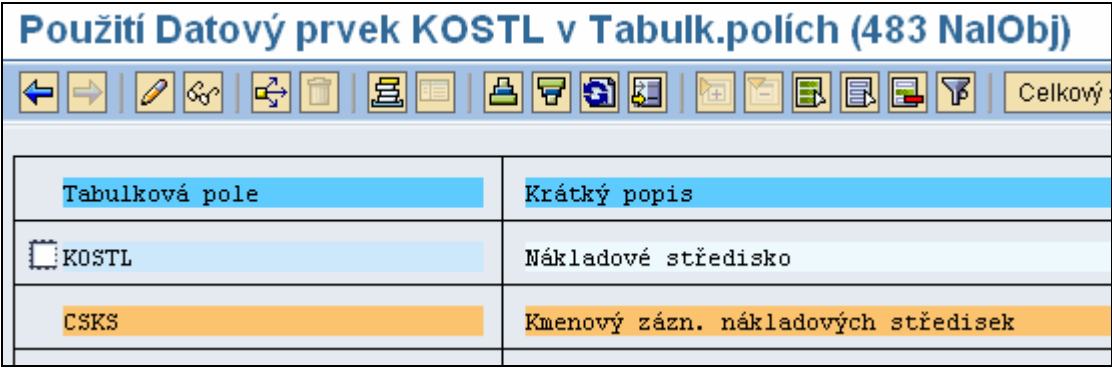
Použití v

☒ Tabulk.polích
☐ Pole struktury
☐ Pole view
☐ Typy tabulek
☐ Náp.vhl

Obr. 7.6: Zobrazení datového prvku aplikace. Přes tlačítko „Přehled použití“ se vyhledají mimo jiné existující tabulky s tímto prvkem.

SAP aplikace také umožňuje vyhledávání nad datovým prvkem. K tomu účelu slouží tlačítko “Přehled použití“, viz Obr. 7.6. Přehled použití je všestranná funkce, která dokáže nalézt výskyty požadovaného prvku napříč aplikací SAP. Časová náročnost funkce je závislá na velikosti prohledávaného prostoru. V případě hledání datových zdrojů postačí zvolit přehled použití v tabulkových polích.

Výstupem funkce “Přehled použití“ je seznam všech tabulek, v kterých se hledané pole vyskytuje. Nicméně u klíčových komponent, jako je datový prvek nákladového střediska (Obr. 7.7.), nebude výstup sestavy laickému uživateli užitečný, neboť počet použití v databázových tabulkách se může pohybovat v rámci několika set. Pro upřesnění výběru lze použít vyhledávání dle krátkého popisu tabulky, kde lze intuitivně zadat část řetězce “Nákladová střediska“. Další možností je využití silného SAP nástroje “ABAP Debugger“.



Tabulková pole	Krátký popis
☐ KOSTL	Nákladové středisko
☒ CSKS	Kmenový záz. nákladových středisek

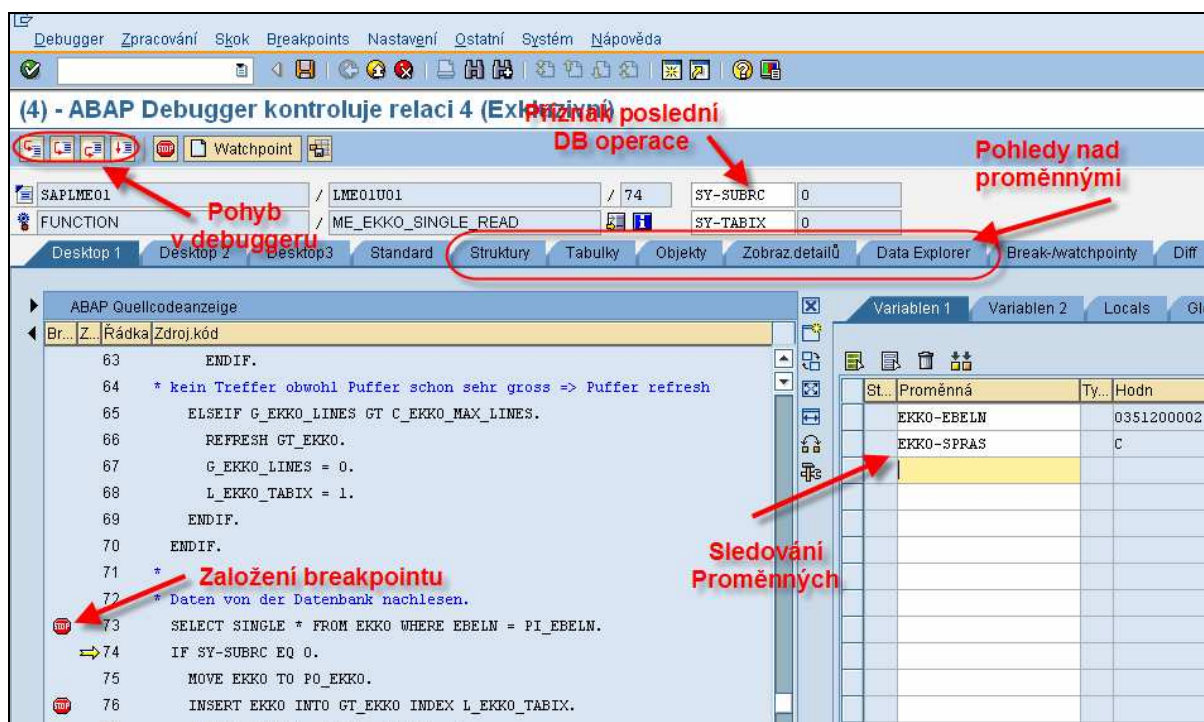
Obr. 7.7: Výstup sestavy „Přehled použití“ při vyhledávání v databázových tabulkách.

7.2.4 Praktické použití nástroje ABAP Debugger

Nástroj ABAP debugger slouží jako klasický debugger ke krokování SAP programů a získání údajů o procesech na pozadí front-end aplikace. Jde o velice silný nástroj, který umožňuje nahlédnout hluboko do struktury SAP aplikací, což na druhou stranu s sebou nese poměrně složitou orientaci v programech a obtížné vyhledávání potřebných údajů v nepřehledném množství kódu. V následujících odstavcích přiblížím využití ABAP debuggeru při vyhledávání relevantních databázových tabulek.

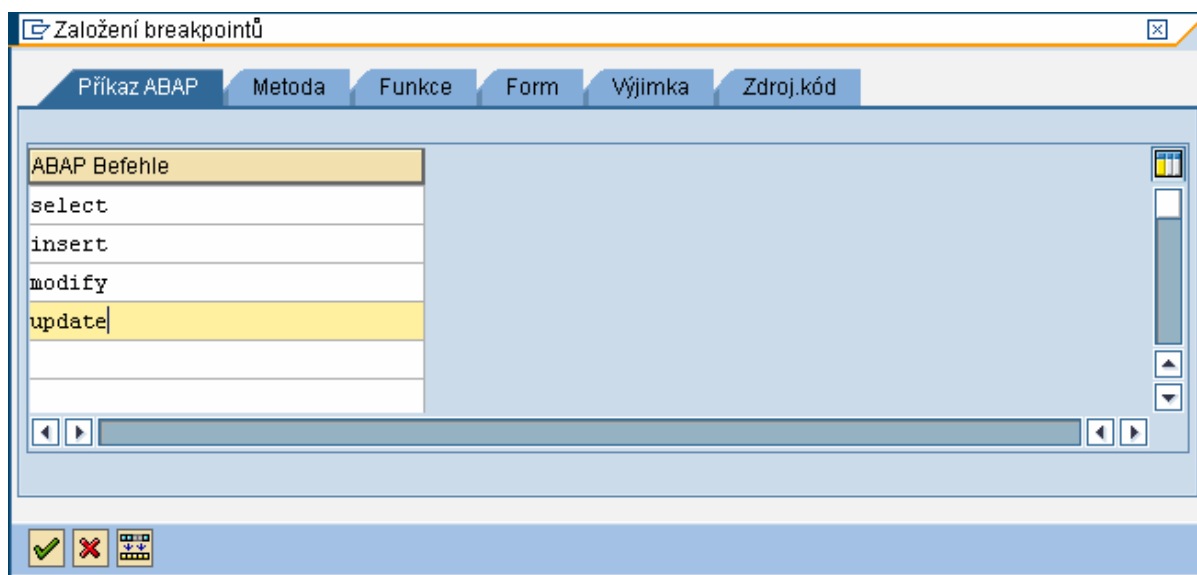
ABAP debugger nad požadovaným programem lze spustit dvěma způsoby. Prvním je nastavení bodu přerušení (breakpoint) přímo v kódu programu. Kód programu lze zobrazit například v transakci SE38 – “ABAP editor“ nebo přímo ve vývojovém nástroji “ABAP Development Workbench“ přes transakci SE80 – “Object Navigator“. Tato činnost nicméně vyžaduje znalost programu, který s analyzovanou transakcí pracuje a rovněž znalost vhodného místa pro nastavení bodu přerušení.

Druhou možností je aktivace debuggeru přímo z uživatelské transakce v aplikaci SAP. K tomuto účelu slouží příkaz “/h”, který se zadá do pole pro spuštění transakce. Příkaz aktivuje debugger a SAP aplikace se při libovolném uživatelském vstupu v okně transakce přepne do debugger módu. Prakticky lze tohoto postupu využít například pro zjištění vstupů do databáze při volání transakce ME23N – Zobrazení nákupní objednávky. Příklad použití nástroje ABAP debugger je uveden na Obr. 7.8.



Obr. 7.8: Příklad použití nástroje ABAP debugger

Důležitou funkcí pro identifikaci databázových tabulek je možnost hromadného nastavení breakpointu nad všemi operacemi určitého typu, viz Obr. 7.9. Funkce se aktivuje v menu “Breakpoints/Breakpoint u/Příkaz“. Stejným způsobem lze definovat body přerušení u volání podprogramů nebo výjimek. Pokud tedy definujeme breakpoint u všech volání funkce select, získáme přehled o všech přístupech do databázových tabulek v transakci ME23N, aniž by bylo třeba podrobně krokovat celou rutinu na pozadí zvolené transakce.

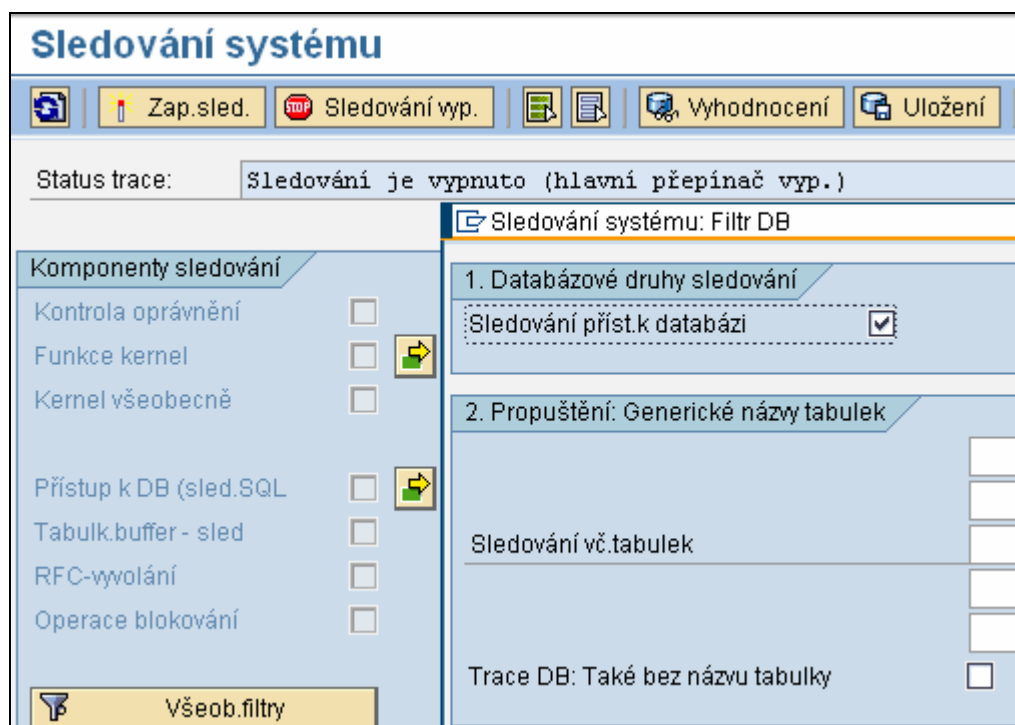


Obr. 7.9: Definice breakpointů u databázových operací. Program se zastaví u všech specifikovaných přístupů do logické databáze.

Výstupem filtrování příkazů “select” v ABAP debuggeru je opět poměrně vysoký počet tabulek. Nicméně v kombinaci s přehledem použití datových polí je možné již většinu hledaných databázových tabulek identifikovat.

7.2.5 Sledování systému

Posledním nástrojem pro analýzu SAP aplikací je robustní nástroj pro sledování systému – transakce ST01. Nástroj je vhodný například ke kontrole a analýze oprávnění uživatelů k přístupu ke komponentám systému. Rovněž lze s jeho pomocí analyzovat i databázové operace, byť jeho výstup ve formě textového výpisu je poměrně nepřehledný a zpravidla vyžaduje další zpracování za pomoci nástrojů pro zpracování textů.



Obr. 7.10: Sledování systému – transakce ST01. Nastavení pro sledování přístupu k databázi.

Sledování systému se nastaví zatržením požadovaných oblastí a stisknutím tlačítka “Zapnutí sledování”. Pro sledování databázových přístupů se zatrhne volba “Přístup k DB (sledování SQL)“, kterou lze v submenu ještě konkrétněji specifikovat, viz Obr. 7.10. Po zapnutí sledování začne ihned probíhat záznam všech databázových operací, proto je vhodné mít již připravené jiné okno aplikace s testovanou transakcí a po aktivaci transakce sledování opět zastavit. Databázové sledování poměrně viditelně snižuje výkon aplikace a po více úkonech již není možné výsledky správně vyfiltrovat.

hh:mm:ss:ms	Typ	Doba(us)	Objekt	Text
02:22:48:816	SQL	350	EKPO	Prog: SAPLMEPO Řádka: 30891 NávrKód: 0
02:22:48:816	SQL	4	EKPO	Prog: SAPLMEPO Řádka: 30891 NávrKód: 0
02:22:48:816	SQL	1200	EKPO	Prog: SAPLMEPO Řádka: 30891 NávrKód: 1403
02:22:48:818	SQL	222	LFA1	Prog: SAPLWY01 Řádka: 1173 NávrKód: 0
02:22:48:818	SQL	3	LFA1	Prog: SAPLWY01 Řádka: 1173 NávrKód: 0
02:22:48:818	SQL	678	LFA1	Prog: SAPLWY01 Řádka: 1173 NávrKód: 0

Obr. 7.11: Část výstupu transakce ST01 při filtrování přístupů k databázi. Zobrazení pro transakci ME23N, tabulky hlavička nákupní objednávky – EKPO a dodavatel LFA1.

Výstupem sledování systému je textový výpis, viz Obr. 7.11. Název databázové tabulky se nachází ve sloupci “Objekt”. Pro zobrazení konkrétního SQL dotazu je třeba otevřít řádek dvojklikem. Celý výpis lze také exportovat do jednoho z předdefinovaných formátů (integrace s MS Excel, Word, txt).

7.2.6 Zdroje dat pro likvidační list faktury

Výsledkem analýzy transakce ME23N pro zobrazení nákupní objednávky za pomoci výše uvedených nástrojů je seznam tabulek logické databáze aplikace SAP a jejich polí, které vstupují do likvidačního listu faktury, viz Tab. 7.3.

Tab. 7.3: Výčet identifikovaných tabulek je následující

Tabulka SAP	Popis tabulky	Seznam polí z front-end aplikace
EKKO	Hlavička nákupního dokladu	Číslo nákupního dokladu, Číslo účtu dodavatele
LFA1	Kmenový soubor dodavatelů	Jméno, Povinnost DPH
EKPO	Položka nákupního dokladu	Číslo položky, Hodnota objednávky netto, Znak DPH
EKKN	Přiřazení v nákupním dokladu	Zakázka, Nákladové středisko, Účet hlavní knihy, SPP prvek
CSKS	Kmen. záznam nákl. středisek	Nákladové středisko, Profit centrum
CSKT	Texty nákl. středisek	Popis nákl. střediska (název pracoviště)
KONP	Podmínky (položka)	Hodnota DPH (dle znaku DPH)
A003	DPH	(hodnoty pro identifikaci DPH)
PRPS	Prvek SPP	SPP prvek, Profit centrum

Tabulky EKKO, EKPO, EKKN tvoří kostru nákupní objednávky. Tabulka EKKO nese informace o hlavičce, EKPO definuje záznam řádku objednávky. Jedna položka, nakupovaný materiál nebo služba obecně odpovídá jednomu řádku v této tabulce. Tabulka EKKN pak slouží k zaznamenání účetních údajů o položce, tedy „kam a na co“ se bude daný nákup účtovat. K objednávce nedílně patří tabulka LFA1 se záznamy o dodavatelích. Mnoho inicializačních nastavení při založení objednávky se kopíruje právě ze záznamu dodavatele.

Tabulky CSKS, CSKT a PRPS jsou ve zdrojových datech zahrnuty čistě kvůli údajům „Název pracoviště“ a „Číslo pracoviště“. „Číslo pracoviště“ odpovídá nákladovému středisku, které je již definované v tabulce přiřazení v nákupním dokladu, ale „Název pracoviště“ lze vyhledat pouze přes tuto tabulku. Navíc v některých případech je účtování definováno na SPP prvek (tabulka PRPS), za těchto podmínek se „Číslo pracoviště“ odvozuje od pole „Profit centrum“ v tabulce SPP prvku. „Název pracoviště“ se i v tomto případě vyhledává přes tabulku nákladového střediska (propojením je právě pole „Profit centrum“).

Poslední skupinu tvoří tabulky A003 pro identifikaci DPH a KONP pro zjištění konkrétní sazby DPH. Oproti ostatním záznamům, které lze z databáze získat prostým příkazem SELECT, je vyhledání a výpočet DPH v SAP aplikaci složitou záležitostí a v implementaci zabírá podstatnou část

kódu. Funkce pro výpočet DPH zahrnuje ještě několik dalších propojovacích tabulek, které zde není důležité uvádět.

Pro prohlížení struktury tabulek a záznamů v databázi slouží transakce SE11 – ABAP Dictionary (zde lze zobrazit a upravovat i ostatní datové typy) a SE16 – Data Browser. S využitím těchto transakcí lze pohodlně pracovat s tabulkovými poli, či vyhledat vhodné testovací záznamy. Transakce SE11 – ABAP Dictionary pro tabulku EKPO je uvedena na Obr. 7.12.

Dictionary: Zobrazení tabulky

Transp.tabulka: **EKPO** Aktiv.

Krátký popis: Položka nákup.dokladu

Vlastnosti Expedice a údržba **Pole** Nápověda/kontrola zadávání Pole měny

Náp.vyhl. Instalovaný typ

Pole	Key	Inic...	Datový prvek	Datový t...	Délka	Deset...	Krátký popis
MANDT	☒	☒	MANDT	CLNT	3	0	Klient
EBELN	☒	☒	EBELN	CHAR	10	0	Číslo nákupní
EBELP	☒	☒	EBELP	NUMC	5	0	Číslo položky
.INCLUDE	☐	☐	EKPODATA	STRU	0	0	Položka nákup
LOEKZ	☐	☐	ELOEK	CHAR	1	0	Znak výmazu v
STATU	☐	☐	ASTAT	CHAR	1	0	Status poptávk
AEDAT	☐	☐	PAEDT	DATS	8	0	Datum změny
TXZ01	☐	☐	TXZ01	CHAR	40	0	Kr.text

Obr. 7.12: Transakce SE11 – ABAP Dictionary pro tabulku EKPO – Hlavička nákupní objednávky

8 Implementace Adobe formuláře

Všechny podklady pro zahájení implementace Adobe formuláře byly prostudovány a možnosti integrací uvedeny v kapitole 7. Vstupem pro implementaci bylo také studium potřebných datových zdrojů, na kterých byl Adobe formulář vystavěn. Implementace se skládá z těchto celků:

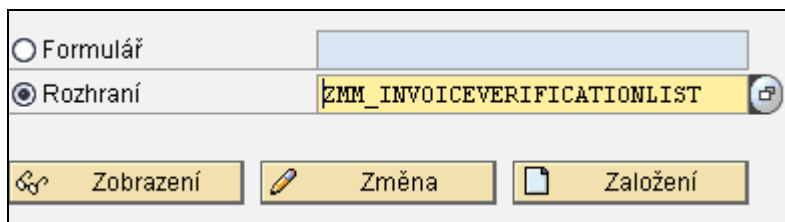
- Rozhraní formuláře (interface)
- Adobe formulář.

Rozhraní definuje prostředí mezi SAP aplikací a Adobe formulářem a pravidla pro komunikaci. Součástí rozhraní jsou vstupy do formuláře v případě exportu ze SAP aplikace a výstupy do SAP aplikace v případě importu dat. Rozhraní dále zajišťuje inicializaci vstupních parametrů formuláře, a to v případě, když nejsou definovány na vstupu obslužným SAP programem.

8.1 Implementace rozhraní

Rozhraní bylo vytvořeno v SAP transakci SFP – Form builder. Rozhraní pro “likvidační list faktury” bylo pojmenováno podle anglického překladu “ZMM_INVOICEVERIFICATIONLIST”. Počáteční znak “Z (nebo Y)” identifikuje všechny objekty, které nejsou součástí standardní SAP implementace a byl uváděn u všech uživatelsky vytvořených objektů. Zkratka „MM“ určuje zařazení úpravy do modulu Materiálového hospodářství (Material Management).

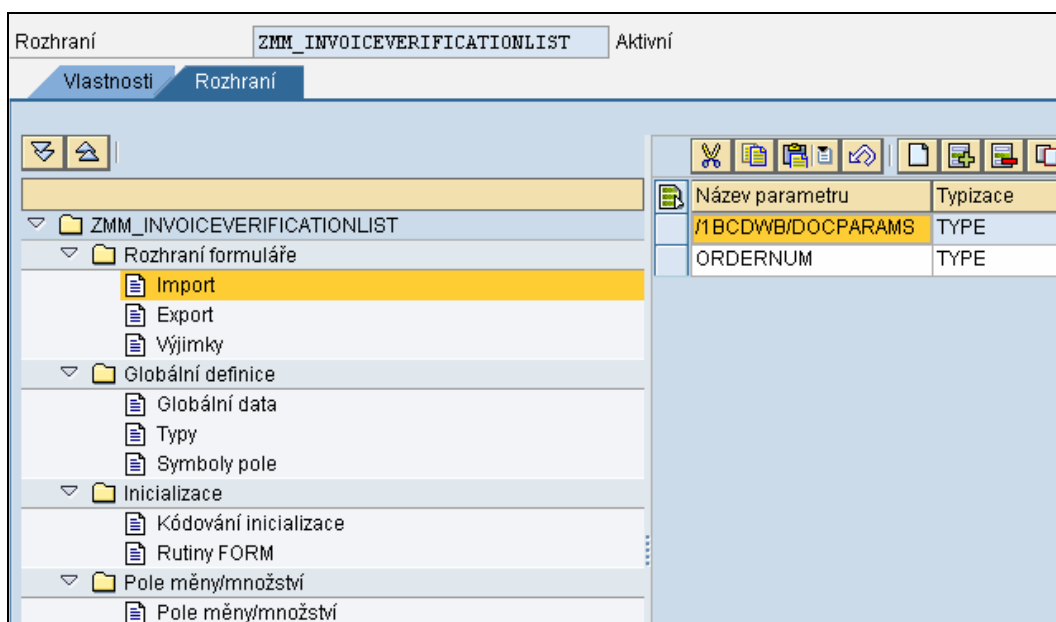
Pro vytvoření rozhraní byl zvolen název formuláře a stisknuto tlačítko založení, viz *Obr. 8.1*. V dalším kroku jsem byl aplikací vyzván ke zvolení typu rozhraní. Rozhraní typu XML se zakládá zejména pro webové dynamické programy (WebDynPro). V případě “likvidačního listu faktury” bylo zvoleno rozhraní založené na ABAP dictionary.



Obr. 8.1: Založení rozhraní formuláře “Likvidačního listu faktury” v transakci SFP

Prostředí nástroje Form builder (*Obr. 8.2*) obsahuje celkem čtyři složky:

- Rozhraní formuláře
- Globální definice
- Inicializace
- Pole měny/množství.



Obr. 8.2: Editace rozhraní Adobe formuláře v transakci SFP

8.1.1 Rozhraní formuláře

Ve složce rozhraní formuláře se nachází definice importních polí, exportních polí a výjimek. Importní pole definují vstupy ze SAP aplikace do formuláře. Standardním parametrem je struktura /1BCDWB/DOCPARAMS typu SFPDOCPARAMS, která definuje několik základních parametrů pro inicializaci formuláře. Patří sem zejména jazykový klíč, stát a příznak pro umožnění editace PDF dokumentů. Definice vstupního parametru (Obr. 8.3) se skládá ze jména parametru, typu a označení typu. Dále lze vybrat, zda je parametr volitelný a popř. navrhnout standardní inicializační hodnotu. Označení typu musí odkazovat na existující strukturu, tabulku či prvek v ABAP dictionary. Nevýhodou je, že nelze na vstupu definovat interní tabulku, nicméně vlastní lokální struktury lze vytvořit v inicializační části rozhraní.

Název parametru	Typizace	Označení typu	Nepovinné	Před.hodn.	Navrhovaná hodnota
/1BCDWB/DOCPARAMS	TYPE	SFPDOCPARAMS	☒	☒	
ORDERNUM	TYPE	EKKO-EBELN	☐	☐	

Obr. 8.3: Definice vstupních parametrů v rozhraní formuláře (transakce SFP)

Stejným způsobem lze definovat exportní parametry, které vracejí data z formuláře do SAP aplikace. Předdefinovaná struktura se v tomto případě jmenuje FPFORMOUTPUT a obsahuje v sobě datové typy nesoucí PDF dokument, XML strukturu, počet stránek a jazykovou verzi formuláře.

Třetí skupinou parametrů jsou definované výjimky. Ty lze použít pro ošetření chyb, které mohou vzniknout například v inicializaci rozhraní. **Obsloužit** je lze v **obslužném** programu, který má

na starosti vyvolání a zobrazení formuláře. Standardní výjimky jsou USAGE_ERROR, SYSTÉM_ERROR a INTERNAL_ERROR.

V rozhraní formuláře “likvidačního listu faktury” byl implementován pouze jeden vstupní parametr, a to číslo objednávky. Všechny ostatní data jsou od něj odvozené a jejich načtení probíhá ve fázi inicializace rozhraní. Výstupní parametry či nestandardní výjimky současná implementace formuláře nevyužívá.

8.1.2 Globální definice

Globální definice obsahuje definici globálních dat, globálních typů a symbolů pole. Všechny zde definované prvky lze použít v kterékoliv části rozhraní.

Definice globálních dat je totožná s definicí vstupních parametrů rozhraní. Značným omezením je nutnost definice proměnné pouze s klíčovým slovem “TYPE”. Nicméně pro definici interní tabulky bylo nutné vytvořit proměnnou typu “TYPE TABLE OF”, neboť klíčové slovo “TYPE” definuje pouze jeden řádek tabulky. Byly využity konstrukce jazyka ABAP v záložce “Typy” a nadefinovány datové typy pro tabulky. V globálních datech jsou tedy nadefinovány všechny zdrojové tabulky, jejichž názvy byly získány v kapitole Analýza likvidačního listu (Obr. 8.4).

Nově vytvořeným datovým typem je struktura “PO_LINE_STRUCT”. Tato struktura definuje řádek tabulky likvidačního listu, jak je zobrazen v Adobe formuláři. Do polí struktury vstupují záznamy z více tabulek; pole “požadovaná výše likvidace vč. DPH” se navíc dopočítává. Z uvedeného důvodu byla vytvořena nová struktura. Skládání řádků z různých zdrojů přímo ve skriptech Adobe formuláře by bylo neefektivní. Pokud to povaha dat umožňuje, je vhodné provést všechny operace nad daty přímo v rozhraní formuláře.

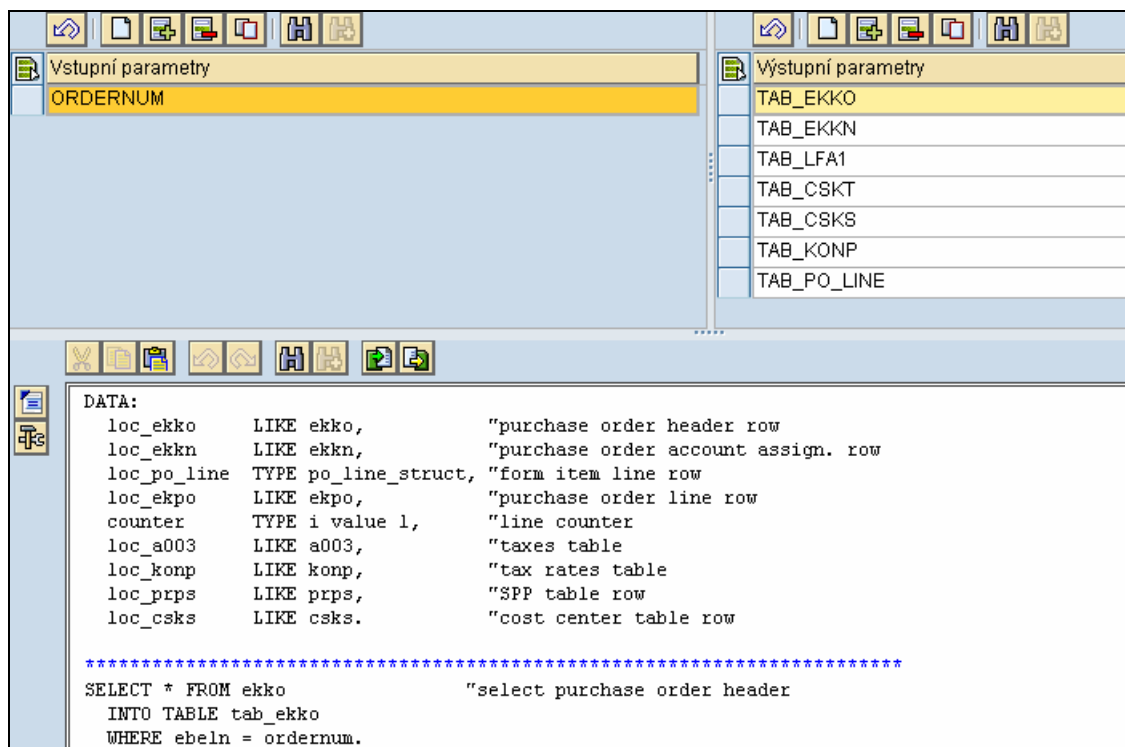
Nad strukturou “PO_LINE_STRUCT” byl vytvořen typ “TP_PO_LINE”, který definuje interní tabulku.

	Název proměnné	Typizace	Označení typu	Navrhovaná hodnota	Konstanta
	TAB_EKKO	TYPE	 TP_EKKO		☐
	TAB_EKKN	TYPE	 TP_EKKN		☐
	TAB_LFA1	TYPE	 TP_LFA1		☐
	TAB_CSKS	TYPE	 TP_CSKS		☐
	TAB_CSKT	TYPE	 TP_CSKT		☐
	TAB_PO_LINE	TYPE	 TP_PO_LINE		☐
	TAB_KONP	TYPE	 TP_KONP		☐

Obr. 8.4: Definice globálních tabulek v rozhraní likvidačního listu

8.1.3 Inicializace

Ve složce Inicializace se nacházejí zdrojové soubory “Kódování inicializace” a “Rutiny FORM”. Kódování inicializace obsahuje rutinu pro načtení dat z databázových tabulek do interních struktur, které budou předány Adobe formuláři (Obr. 8.5).



Obr. 8.5: Obrazovka kódování inicializace v rozhraní formuláře

V inicializačním souboru se definují vstupní parametry. Jde o stejné parametry, které jsou definovány v části import. Výstupem jsou pak definované globální parametry. Do Adobe formuláře vstupují pole, která jsou definovaná jako výstupní proměnné v souboru inicializace.

Soubor “Rutiny FORM” obsahuje zdrojové kódy funkcí, které se v jazyce ABAP definují klíčovým slovem “FORM”. V inicializaci tyto funkce pak lze vyvolat příkazem “PERFORM název_rutiny”. V rozhraní likvidačního listu obsahuje soubor rutin funkci pro výpočet DPH, která je tvořena obsáhlým kódem. Vyvoláním rutiny v inicializaci ze vzdáleného místa se zdrojový kód inicializace značně zpřehlední.

Poslední záložka “pole měny/množství” slouží k vytvoření vazeb mezi polem s hodnotou a polem s jednotkou nad definovanými globálními daty.

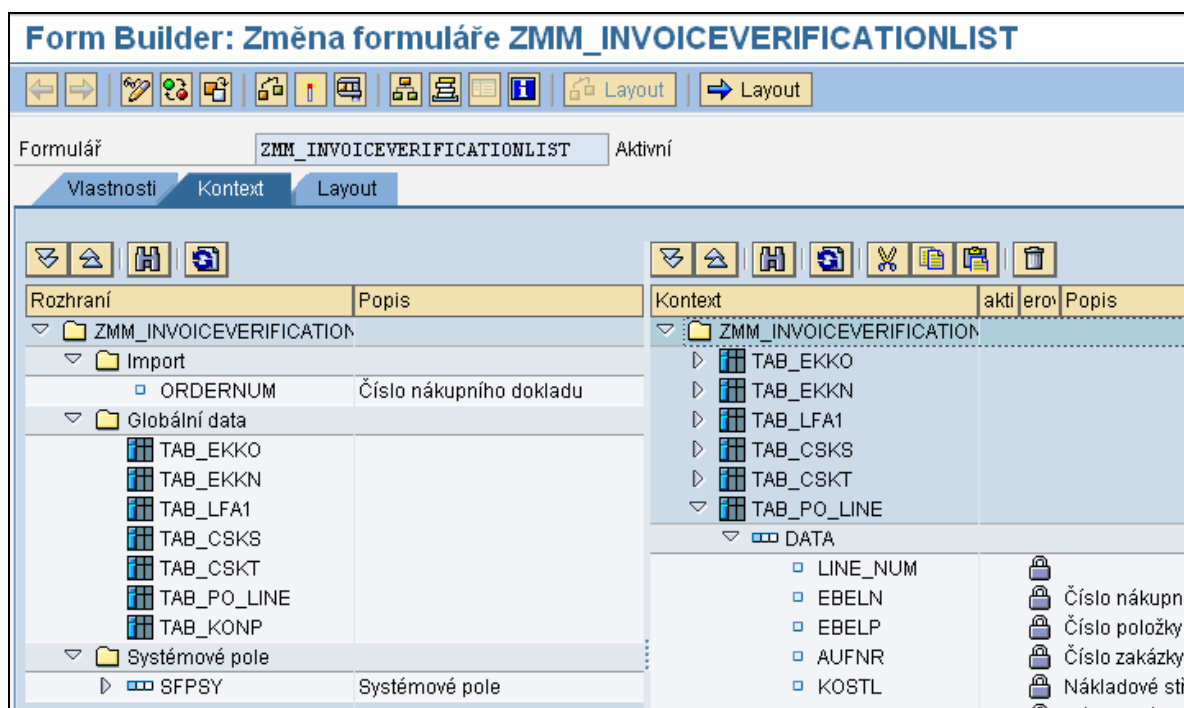
8.2 Implementace formuláře

Adobe formulář byl vytvořen stejně jako rozhraní v transakci SFP. Při zakládání bylo zadáno pouze jméno a bylo přiřazeno rozhraní specifikované v předchozí kapitole 8.1. Jméno Adobe formuláře bylo zvoleno stejné jako název příslušného rozhraní – ZMM_INVOICEVERIFICATIONLIST.

Obrazovka pro změnu formuláře obsahuje záložky: vlastnosti, kontext a layout.

Záložka “Vlastnosti” obsahuje systémové informace o formuláři (datum založení, autor, ...), popis a definované rozhraní.

Na záložce “Kontext” se specifikují data, která vstupují z rozhraní do formuláře (Obr. 8.6). Obrazovka je rozdělena do dvou polovin. V levé části se zobrazují data rozhraní, v pravé pak Kontext Adobe formuláře. Data z rozhraní se přiřazují k formuláři prostým přetažením.



Obr. 8.6: Editace Adobe formuláře “Likvidační list faktury” v transakci SFP

Tabulky rozhraní jako EKKO, LFA1 obsahují poměrně vysoký počet polí (např. tabulka EKKO – 138 záznamů). Formulář přitom využívá z každé tabulky pouze jedno či dvě pole. Načtením obsahu všech polí by se práce s formulářem stala neúměrně pomalá a pro uživatele nepřijatelná. Z tohoto důvodu byla ve formuláři likvidačního listu deaktivována všechna nepoužívaná pole. Deaktivace byla provedena v kontextové nápovědě příslušného pole (pravé tlačítko myši). V kontextovém menu se zobrazí volby pro deaktivaci pole a pro aktivní nastavení. Tímto způsobem lze jednou deaktivované pole opět lehce vložit do zdrojových dat formuláře.

8.2.1 Práce s vestavěnou aplikací Adobe LiveCycle Designer

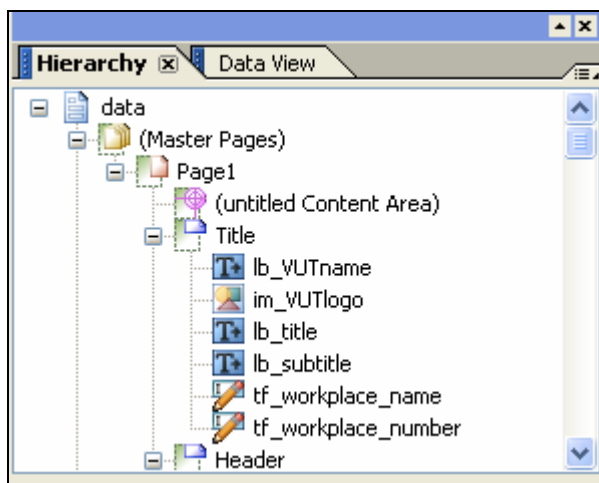
Vzhled a funkce formuláře byly vytvořeny ve vestavěné vývojové aplikaci Adobe LiveCycle Designer (Obr. 8.12). Okno Adobe designeru je rozděleno do několika částí. Jeho základní rozdělení je podobné například aplikacím MS Office.

V centrální části se nachází okno reprezentující grafické rozvržení navrženého formuláře. Formulář obsahuje záložky Design view, Master pages a Preview PDF. První záložka tvoří tělo dokumentu. Záložka Master pages slouží k definování komponent, které se opakují na každé stránce. Většinou se jedná o záhlaví a zápatí stránky. Kliknutím na třetí záložku Preview PDF se spouští aplikace Adobe Reader, která zobrazí náhled nad vytvořeným PDF dokumentem.

Po stranách se nacházejí panely s jednotlivými funkcemi pro editaci formuláře. Funkce jsou seříděny do tématických skupin. Tyto skupiny lze ve vývojovém prostředí přeskupovat.

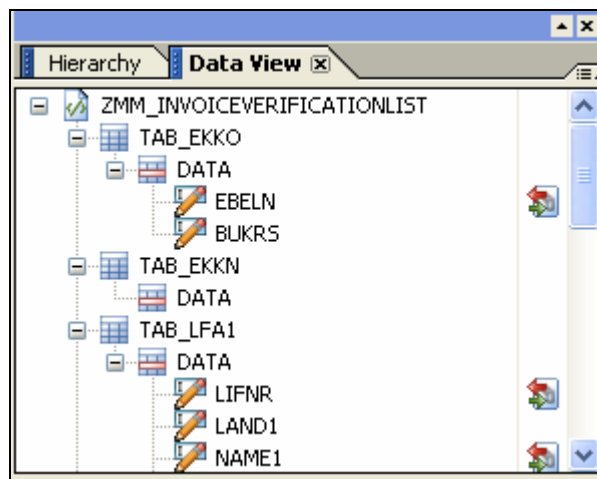
Hlavní skupiny funkcí jsou následující:

- **Hierarchy** – Skupina zobrazuje hierarchickou strukturu prvků formuláře. Formulář definuje kořenový uzel “data”. Na nejvyšší úrovni jsou jeho potomky jednotlivé stránky formuláře, které se dále dělí na stránky hlavičky (opakované na všech stranách dokumentu) a stránky obsahu. Struktura se dále větví až ke koncovým prvkům stránky, kterými jsou většinou pole a texty formuláře.



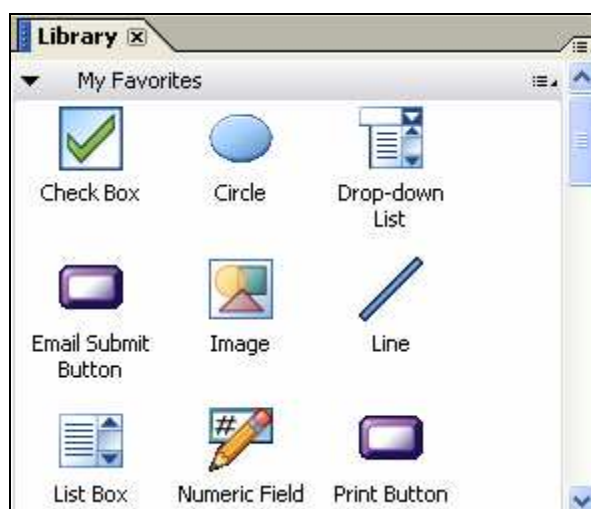
Obr. 8.7: Příklad hierarchické struktury PDF dokumentu

- **Data view** – Záložka Data view obsahuje datovou strukturu PDF dokumentu vycházející z rozhraní formuláře. Kořenovým uzlem stromu je název rozhraní. Strom neobsahuje všechna pole rozhraní, ale pouze ta, která byla přenesena do kontextu formuláře. Ikona v pravém sloupci ukazuje, zda je datové pole již připojeno k prvku formuláře. Propojení lze provést přetažením pole do grafického rozvržení nebo nastavením atributu “binding” ve vlastnostech prvku formuláře.



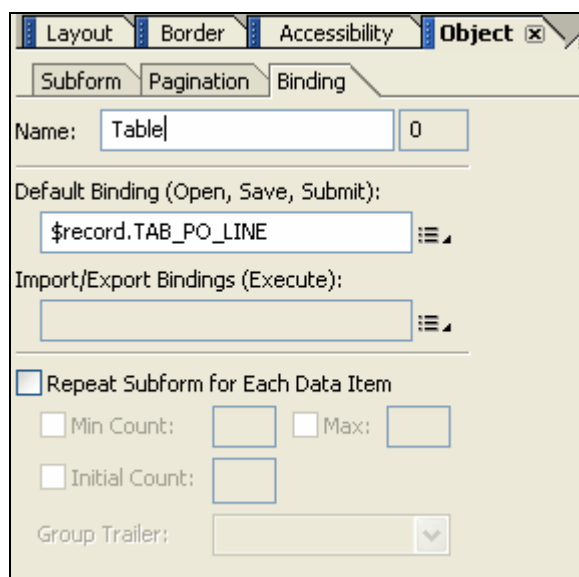
Obr. 8.8: Příklad datové struktury v PDF dokumentu

- **Library** – Tato záložka obsahuje předdefinované prvky formuláře. Do grafického layoutu se zavádějí přetažením. Prvků je široká škála, rozdělená do několika logických skupin.



Obr. 8.9: Ukázka knihovny předdefinovaných komponent formuláře

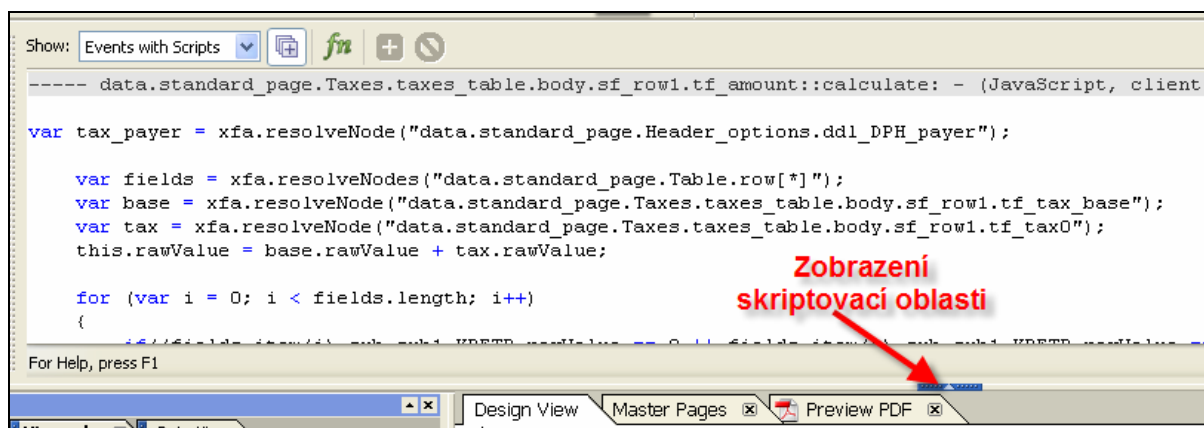
- **Properties** – Okno “vlastnosti” umožňuje měnit vzhled, chování či funkčnost prvku. Jednotlivé atributy se mění podle typu komponenty. Pro určení pozice a zobrazení komponenty ve formuláři slouží záložka Layout. Záložka Border definuje grafické ohraničení prvku. Accessibility určuje roli komponenty (např. řádek tabulky, tabulka) a definuje např. kontextovou nápovědu nad prvkem. Poslední záložka Object obsahuje vlastnosti určující chování prvku. Záložky se mění podle typu objektu (Field u pole apod.). Definují typ komponenty, popis, počet znaků, možnost editace, apod. Pro většinu typů komponent je společná pouze záložka Binding, která definuje propojení na datový prvek (Obr. 8.10).



Obr. 8.10: Vlastnosti komponenty formuláře

Další komponentou vývojového prostředí je lišta menu s panelem nástrojů. Většina funkcí je duplicitních k postraním panelům. Ostatní položky menu obsahují funkce typické pro běžný textový editor (lupa, editace, vložení, zobrazení, ...). V porovnání se standardní aplikací Adobe LiveCycle Designer chybí ve vestavěné verzi v prostředí SAP nabídka “Soubor“, která obvykle obsahuje řídicí funkce aplikace (uložení, otevření, konec, ...). Tato funkčnost je nahrazena řídicím programem aplikace SAP.

Posledním významným prvkem Adobe designeru je vestavěný skriptovací editor s podporou programovacích jazyků FormCalc a JavaScript. Jazyk FormCalc je jednoduchým skriptovacím jazykem, který využívá svých zabudovaných funkcí. Jeho použitelnost je podobná např. vzorcům v aplikaci MS Excel. FormCalc je pro svou jednoduchost vhodný zejména pro výpočty nad tabulkami. Pro složitější programování ve formuláři lze využít jazyk JavaScript (Obr. 8.11).



Obr. 8.11: Ukázka skriptu nad polem v jazyce JavaScript

Zdrojový kód naprogramovaných funkcí se vkládá do předdefinovaných metod objektů formuláře. Téměř všechny komponenty ve formuláři vlastní metody, které lze ve skriptovacím editoru přetěžovat. Výjimku tvoří textová pole a grafické elementy.

Jednotlivé metody se volají v různé fázi zpracování formuláře a dělí se na metody:

- **Interakce komponenty** (initialize, calculate, validate)
- **Uživatelský vstup** (enter, exit, mouseEnter, mouseExit, change, full, mouseUp, mouseDown, click)
- **Operace nad formulářem** (preSave, postSave, preprint, postPrint, preSubmit, docReady, docClose a form:ready, layout:ready).

8.2.2 Zpracování Likvidačního listu faktury v Adobe LiveCycle Designer

V předchozí kapitole byla obecně popsána práce s Adobe designerem. Tato kapitola popisuje část konkrétní implementace Adobe formuláře v Adobe designeru pro Likvidační list faktury. Implementace rozhraní formuláře a formuláře v back-end systému SAP jsou rovněž popsány v předchozích částech.

8.2.2.1 Design formuláře

První částí práce bylo vytvoření designu formuláře v požadovaném rozvržení. Samostatně byla navržena hlavička formuláře, která se tiskne opakovaně na všech stránkách dokumentu. Do hlavičky spadají název dokumentu, název a číslo pracoviště, číslo účetního dokladu a informace o dodavateli včetně čísla objednávky.

V těle formuláře pak byl navržen zbytek polí v designu předlohy. Nestandardně bylo provedeno pouze řešení tabulky položek. Jelikož jsou pole ve sloupcích položka a základ daně/daň rozděleny do dvou menších polí (obr. 8.13), nebylo možné použít standardní prvek "tabulka". Místo toho byla celá tabulka navržena za pomoci komponenty subform (obr. 8.14).

SPP prvek	Položka	Požadovaná výše likvidace vč. DPH	Daňová sazba položky (%)	Základ daně / daň
	123	119	19	100,00
	123			19,00

Obr. 8.13: Problém zdvojených polí v tabulce

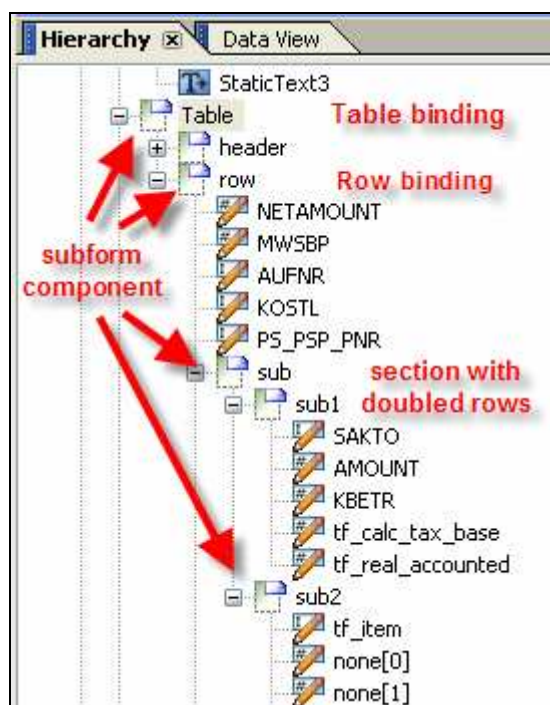
Prvek subform obvykle slouží pouze ke svázání určité skupiny prvků, nicméně v tomto případě při správném propojení na data alternativní tabulka z komponent subform se chová stejně jako klasická tabulka a navíc umožňuje tvorbu volnějšího designu.

Navázání tabulky po datové stránce je následující:

Subform Table.binding = \$record.TAB_PO_LINE

Subform Table.row.binding = DATA[*]

Dále je pak již klasické provázání položek 1:1.



Obr. 8.14: Řešení tabulky pomocí komponent subform

Požadovaného chování formuláře se dosahuje kombinací atributů jednotlivých komponent. Např. vlastnost “data pattern” definuje formát zobrazení číselného pole apod.

8.2.2.2 Skripty formuláře

Veškeré funkce ve formuláři byly vytvořeny v jazyce JavaScript. Téměř všechny implementované funkce se zabývají výpočtem hodnot z ostatních polí, nebo zobrazením pole v závislosti na určité podmínce. V případě výpočtu hodnot byly využity metody “calculate”, řízení zobrazení polí pak řídí metoda “initalize”. Příkladem typické funkce nad polem je např. výpočet hodnoty u pole “základ daně s daňovou sazbou 0%”:

```
var fields = xfa.resolveNodes("data.standard_page.Table.row[*]");
this.rawValue = 0;
for (var i = 0; i < fields.length; i++)
{
    if((fields.item(i).sub.sub1.KBETR.rawValue == 0 ||
fields.item(i).sub.sub1.KBETR.rawValue == null))
        this.rawValue +=
fields.item(i).sub.sub1.tf_calc_tax_base.rawValue;
}
```

JavaScript se odkazuje na určitý uzel hierarchické struktury přes funkci *resolveNode(„node“)*. V případě vrácení uzlu typu pole, jako jsou např. řádky tabulky, používá se *resolveNodes(„node[*]“)*, kde symbol „*“ určuje vrácení všech řádků. Na konkrétní řádek se pak odkazuje přes *.item(i)*, kde *i* číslo řádku, číslováno od 0.

8.2.3 Obslužný program SAP pro vyvolání formuláře

Pro testovací účely byl vytvořen SAP report ZMM_XHASMA00_IVL v testovacím systému VUT. Report lze spustit v transakci se38. Vstupním parametrem je číslo objednávky, poté se vygeneruje Adobe formulář likvidační list faktury. Formulář se zobrazí ve vestavěném okně SAP v aplikaci Adobe Reader. Formulář lze uložit do formátu PDF nebo vytisknout. Data vyplněná v aplikaci SAP se přenášejí spolu s formulářem.

9 Závěr

V první části diplomové práce byla prezentována technologie SAP založená na platformě NetWeaver, její stěžejní produkt mySAP a možnosti vývoje v jazyce ABAP v prostředí ABAP workbench.

V další části byly prozkoumány možnosti integrace produktu společnosti SAP – mySAP s externími formáty XLS programu MS Excel a PDF společnosti Adobe. Z práce vyplývá, že platforma NetWeaver podporuje obě možnosti, přičemž obě přinášejí určité výhody a nevýhody. Vytvoření aplikací exportujících data do formátu MS Excel se jeví vhodné pro lokální řešení, jelikož řešení nezasahuje do administrativy systému a lze pro něj vytvořit poměrně jednoduchý skript. Nevýhodou ovšem bude těžkopádnost při modifikaci excelovské šablony, kdy bude zřejmě nutné pozměnit i příslušný program.

Univerzálnějším řešením je využití již zabudované podpory v SAP systémech pro interaktivní Adobe PDF formuláře. PDF formuláře nabízejí příjemné drag-and-drop prostředí pro jejich tvorbu v software Adobe LiveCycle Designer. Rovněž je zde k dispozici široká škála funkcí, od designu, přes výpočty, validace, až po možnosti zabezpečení v podobě digitálního podpisu. Dosažené výsledky s PDF formuláři mnohostranně přesahují možnosti integrace s MS Excel.

V praktické části diplomové práce jsem zanalyzoval proces zpracování nákupní objednávky v systému SAP a důsledně jsem se zaměřil na část procesu týkající se likvidačního listu faktury. Fakturu jsem zanalyzoval z pohledu SAP aplikace a pro vývoj interaktivního formuláře zvolil možnost integrace SAP s PDF formuláři Adobe.

V poslední části práce jsem zdokumentoval vývoj rozhraní pro komunikaci SAP s PDF formulářem a vývoj samotného formuláře v prostředí SAP a vestavěné aplikaci Adobe LiveCycle Designer.

Výsledkem práce je funkční formulář, který splňuje podmínky zadání. Formulář lze vyvolat v aplikaci SAP. Na vstupu se zadá číslo nákupní objednávky a formulář při svém generování převezme příslušná data z aplikace. Vyvolání formuláře je řízeno samostatným SAP programem a nevolá se přímo z transakce nákupní objednávky. Důvodem je udržení konzistence standardních SAP objektů. Pokud bude rozhodnuto, že vytvořený Adobe formulář bude využit v reálném procesu, příslušné tlačítko v transakcích nákupní objednávky bude doplněno.

V dalších krocích se nabízí rozšíření funkčnosti formuláře o zpětný import do prostředí SAP, kde by schválený likvidační list faktury mohl sloužit jako podklad pro transakci MIRO – Založení faktury. Rovněž by bylo možné rozšířit funkce v Adobe formuláři dle konkrétních požadavků a zkušeností z praxe uživatelů. Vítanou funkcí by byla možnost přidání prázdných řádků v likvidačním listu a převedení formuláře na rozhraní s XML bází, což by umožnilo práci s formulářem ve webových dynamických programech SAP. Současné rozhraní formuláře na ABAP Dictionary nabízí

méně možností. Například funkci pro dynamické přidávání řádků v PDF dokumentu se mi se stávajícím rozhraním formuláře nepodařilo implementovat.

Celkově shledávám interaktivní formuláře velmi užitečným nástrojem nejen pro SAP aplikace, kde zřejmě postupně nahradí starší technologie tisku reportů SAPscript a SmartForms.

Literatura

- [1] Basl, J., Benda, L.: *Podpora podnikových procesů produkty SAP*. Praha, Oeconomica 2003
- [2] Wikipedia: *Wikipedia, The Free Encyclopedia*. 2008, [Online; navštíveno 30.12.2008].
URL <http://en.wikipedia.org/wiki/Abap>
- [3] SAP: *SAP Solutions*. 2008, [Online; navštíveno 28.12.2008].
URL <http://www.sap.com>
- [4] Adobe: *Help Ressource Center*. 2008, [Online; navštíveno 27.12.2008].
URL <http://livedocs.adobe.com>
- [5] SAP Comm. Network: *SAP Interactive Forms by Adobe*. 2008, [Online; navštíveno 1.1.2009].
URL <https://www.sdn.sap.com/irj/sdn/adobe>
- [6] Maassen, A., Schoenen, M., Frick, D., Gadatsch, A.: *SAP R/3 kompletní průvodce*. Brno, Computer Press 2007
- [7] SAP AG: *ABAP programming*. SAP AG 2003
- [8] SAP Comm. Network: *Integrated Development with ABAP workbench*. 2008, [Online; navštíveno 2.1.2009]. URL <https://www.sdn.sap.com/irj/sdn/abap-workbench>
- [9] Let us ABAP: *SAP and ABAP facts*. 2008, [Online; navštíveno 1.1.2009].
URL <http://abaplovers.blogspot.com/2008/05/abap-internal-table-to-excel-sheet.html>
- [10] SAP Basic: *SAP ABAP programming*. 2008, [Online; navštíveno 3.1.2009].
URL <http://www.sap-img.com/ab018.htm>
- [11] ABAP Code: *SAP Adobe Forms*. 2008, [Online; navštíveno 4.1.2009].
URL http://www.abap-code.com/adobe_forms.html
- [12] SAP AG: *SAP Help Portal*. 2008, [Online; navštíveno 4.1.2009].
URL <http://help.sap.com>
- [13] Suchomel, T.: *Zpracování nákupní objednávky*. Brno, VUT 2006

Seznam příloh

1. Zdrojové kódy rozhraní formuláře
2. Obslužná rutina pro vyvolání pdf formuláře z prostředí SAP
3. Prostředí Adobe LiveCycle Designer
4. Návrh formuláře Likvidační list faktury .XLS
5. Likvidační list faktury .PDF

Příloha 1. – Zdrojové kódy rozhraní formuláře

TYPES:

```
tp_ekko TYPE TABLE OF ekko,          "purchase order header
tp_ekkn TYPE TABLE OF ekkn,          "purch. order line account assign.
tp_lfa1 TYPE TABLE OF lfa1,          "vendor
tp_cskks TYPE TABLE OF cskks,        "cost center
tp_cskt TYPE TABLE OF cskt,          "cost center texts
tp_konp TYPE TABLE OF konp.          "tax rates
```

TYPES:

```
BEGIN OF po_line_struct,              "item line
  line_num TYPE i,                     "sequence number
  ebeln     LIKE ekkn-ebeln,            "reference to purch. order header
  ebelp     LIKE ekkn-ebelp,            "reference to purch. order line
  aufnr     LIKE ekkn-aufnr,            "order
  kostl     LIKE ekkn-kostl,            "cost center
  kokrs     LIKE ekkn-kokrs,            "cost center group
  ps_psp_pnr LIKE ekkn-ps_psp_pnr,      "SPP
  sakto     LIKE ekkn-sakto,            "item
  amount    LIKE ekpo-netwr,            "line amount with taxes
  netamount LIKE ekpo-netwr,            "net amount (without taxes)
  mwsbp     LIKE komp-mwsbp,            "tax amount
  kbetr     LIKE konp-kbetr,            "tax rate
END OF po_line_struct.
```

TYPES tp_po_line type TABLE OF po_line_struct.

Název proměnné	Typizace	Označení typu
TAB_EKKO	TYPE	TP_EKKO
TAB_EKKN	TYPE	TP_EKKN
TAB_LFA1	TYPE	TP_LFA1
TAB_CSKS	TYPE	TP_CSKS
TAB_CSKT	TYPE	TP_CSKT
TAB_PO_LINE	TYPE	TP_PO_LINE
TAB_KONP	TYPE	TP_KONP

DATA:

```
loc_ekko     LIKE ekko,                "purchase order header row
loc_ekkn     LIKE ekkn,                "purchase order account assign. row
loc_po_line  TYPE po_line_struct,      "form item line row
loc_ekpo     LIKE ekpo,                "purchase order line row
counter      TYPE i value 1,           "line counter
loc_a003     LIKE a003,                "taxes table
loc_konp     LIKE konp,                "tax rates table
loc_prps     LIKE prps,                "SPP table row
loc_cskks    LIKE cskks.               "cost center table row
```

```
SELECT * FROM ekko                      "select purchase order header
  INTO TABLE tab_ekko
 WHERE ebeln = ordernum.
```

```

*****
LOOP AT tab_ekko INTO loc_ekko.    "select data relevant to purchase order

    SELECT * FROM lfa1              "select vendor
        INTO TABLE tab_lfa1
        WHERE lifnr = loc_ekko-lifnr.

    SELECT * FROM ekkn              "select lines account assignment
        INTO TABLE tab_ekkn
        WHERE ebeln = loc_ekko-ebeln.

*   SELECT * FROM ekkn              "init form item lines from ekkn table
*   INTO CORRESPONDING FIELDS OF TABLE tab_po_line
*   WHERE ebeln = loc_ekko-ebeln.

ENDLOOP.

*****
SELECT SINGLE * FROM ekko          "select purchase order to local struct
    INTO loc_ekko
    WHERE ebeln = ordernum.

LOOP AT tab_ekkn INTO loc_ekkn.    "fill form item lines

    loc_po_line-line_num    = counter.
    loc_po_line-aufnr       = loc_ekkn-aufnr.
    loc_po_line-kostl       = loc_ekkn-kostl.
    loc_po_line-kokrs       = loc_ekkn-kokrs.
    loc_po_line-ps_psp_pnr  = loc_ekkn-ps_psp_pnr.
    loc_po_line-sakto       = loc_ekkn-sakto.
    loc_po_line-ebeln       = loc_ekkn-ebeln.
    loc_po_line-ebelp       = loc_ekkn-ebelp.

    SELECT SINGLE * from ekpo      "select relevant order line
        INTO loc_ekpo
        WHERE ebeln = loc_ekkn-ebeln
        AND ebelp = loc_ekkn-ebelp.

    SELECT SINGLE * from a003      "select tax record
        INTO loc_a003
        WHERE kappl = 'TX'        "identify tax record
        AND aland = loc_ekko-lands
        AND mwszk = loc_ekpo-mwszk.

    SELECT SINGLE * from konp      "select tax rate record
        INTO loc_konp
        WHERE knumh = loc_a003-knumh.

    loc_po_line-kbetr = loc_konp-kbetr / 10.
    PERFORM calculate_tax          "calc tax
    USING loc_ekkn-ebeln loc_ekkn-ebelp
    CHANGING loc_po_line-mwsbp.
    loc_po_line-netamount = loc_ekpo-netwr.
    loc_po_line-amount = loc_ekpo-netwr + loc_po_line-mwsbp.

    APPEND loc_po_line TO tab_po_line.
    loop at tab_po_line into loc_po_line.
        loc_po_line-line_num = 30.
        MODIFY tab_po_line INDEX sy-tabix FROM loc_po_line
            TRANSPORTING line_num.
    endloop.
    clear loc_po_line.
    counter = counter + 1.
ENDLOOP.

*****

```

```

*LOOP AT tab_po_line INTO loc_po_line "select workplace info
LOOP AT tab_ekkn INTO loc_ekkn.

  IF NOT loc_ekkn-kostl IS INITIAL.

    SELECT * FROM csks          "select cost center
      INTO TABLE tab_csks
      WHERE kostl = loc_ekkn-kostl
        AND kokrs = loc_ekkn-kokrs
        AND datab <= loc_ekko-bedat "cost center validity in voucher date
        AND datbi >= loc_ekko-bedat.

  ELSEIF NOT loc_ekkn-ps_psp_pnr IS INITIAL.

    SELECT SINGLE * FROM prps    "select cost center from SPP
      INTO loc_prps
      WHERE pspnr = loc_ekkn-ps_psp_pnr.

    IF sy-subrc = 0.
      SELECT * FROM csks
        INTO TABLE tab_csks
        WHERE prctr = loc_prps-prctr.
    ENDIF.

  ENDIF.

LOOP AT tab_csks INTO loc_csks.
  SELECT * FROM cskt          "select cost center texts
    INTO TABLE tab_cskt
    WHERE kokrs = loc_csks-kokrs
      AND kostl = loc_csks-kostl
      AND datbi = loc_csks-datbi
      AND spras = loc_ekko-spras.
ENDLOOP.
EXIT.
ENDLOOP.

*-----*
*      FORM calculate_tax      *
*-----*
FORM calculate_tax USING      p_ebeln TYPE ekpo-ebeln
                             p_ebelp TYPE ekpo-ebelp
                             CHANGING p_mwsbp TYPE komp-mwsbp .

TABLES : ekko , ekpo , t001 , komk , komp .

CONSTANTS: bstyp_info VALUE 'I',
            bstyp_ordr VALUE 'W',
            bstyp_banf VALUE 'B',
            bstyp_best VALUE 'F',
            bstyp_anfr VALUE 'A',
            bstyp_kont VALUE 'K',
            bstyp_lfpl VALUE 'L',
            bstyp_lerf VALUE 'Q'.

DATA : taxcom TYPE taxcom ,
      t_konv TYPE TABLE OF komv WITH HEADER LINE .

DATA: BEGIN OF tkomv OCCURS 50.
      INCLUDE STRUCTURE komv.
DATA: END OF tkomv.

DATA: BEGIN OF tkomvd OCCURS 50. "Belegkonditionen
      INCLUDE STRUCTURE komvd.
DATA: END OF tkomvd.

```

```

DATA : BEGIN OF tkomvh OCCURS 50.
      INCLUDE STRUCTURE komv.
DATA : vtext LIKE t685t-vtext.
DATA : END OF tkomvh.

SELECT SINGLE *
      INTO ekko
      FROM ekko
      WHERE ebeln = p_ebeln .

SELECT SINGLE *
      INTO ekpo
      FROM ekpo
      WHERE ebeln = p_ebeln
      AND ebelp = p_ebelp .

SELECT SINGLE *
      INTO t001
      FROM t001
      WHERE bukr = ekko-bukrs .

taxcom-bukrs = ekpo-bukrs.
taxcom-budat = ekko-bedat.
taxcom-waers = ekko-waers.
taxcom-kposn = ekpo-ebelp.
taxcom-mwskz = ekpo-mwskz.
taxcom-txjcd = ekpo-txjcd.
taxcom-shkzg = 'H'.
taxcom-xmwst = 'X'.
IF ekko-bstyp EQ bstyp_best.
    taxcom-wrbtr = ekpo-netwr.
ELSE.
    taxcom-wrbtr = ekpo-zwert.
ENDIF.

taxcom-lifnr = ekko-lifnr.
taxcom-land1 = ekko-lands.
taxcom-ekorg = ekko-ekorg.
taxcom-hwaer = t001-waers.
taxcom-llief = ekko-llief.
taxcom-bldat = ekko-bedat.
taxcom-matnr = ekpo-ematn.
taxcom-werks = ekpo-werks.
taxcom-bwtar = ekpo-bwtar.
taxcom-matkl = ekpo-matkl.
taxcom-meins = ekpo-meins.

IF ekko-bstyp EQ bstyp_best.
    taxcom-mglme = ekpo-menge.
ELSE.
    IF ekko-bstyp EQ bstyp_kont AND ekpo-abmng GT 0.
        taxcom-mglme = ekpo-abmng.
    ELSE.
        taxcom-mglme = ekpo-ktmng.
    ENDIF.
ENDIF.
IF taxcom-mglme EQ 0.
    taxcom-mglme = 1000.
ENDIF.
taxcom-mtart = ekpo-mtart.

IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

CALL FUNCTION 'J_1BSA_COMPONENT_ACTIVE'
      EXPORTING

```

```

        bukrp                = ekko-bukrs
        component              = 'BR'
    EXCEPTIONS
        component_not_active = 1
        OTHERS                = 2.

IF sy-subrc IS INITIAL.

    komk-mandt = ekko-mandt.
    komk-kalsm = ekko-kalsm.
    IF ekko-kalsm = ''.
        komk-kalsm = 'RM0000'.
    ENDIF.
    komk-kappl = 'M'.
    komk-waerk = ekko-waers.
    komk-knumv = ekko-knumv.
    komk-lifnr = ekko-lifnr.
    komp-kposn = ekpo-ebelp.
    komp-matnr = ekpo-matnr.
    komp-werks = ekpo-werks.
    komp-matkl = ekpo-matkl.
    komp-infnr = ekpo-infnr.
    komp-evrtn = ekpo-konnr.
    komp-evrtp = ekpo-ktprn.

    CALL FUNCTION 'RV_PRICE_PRINT_ITEM'
        EXPORTING
            comm_head_i = komk
            comm_item_i = komp
            language     = 'E'
        TABLES
            tkomv        = tkomv
            tkomvd       = tkomvd.

    CALL FUNCTION 'J_1B_NF_PO_DISCOUNTS'
        EXPORTING
            i_kalsm = ekko-kalsm
            i_ekpo  = ekpo
        IMPORTING
            e_ekpo  = ekpo
        TABLES
            i_konv  = t_konv.

    IF NOT ekko-llief IS INITIAL.
        taxcom-lifnr = ekko-llief.
    ENDIF.

ENDIF.

CALL FUNCTION 'FIND_TAX_SPREADSHEET'
    EXPORTING
        buchungskreis = t001-bukrs
    EXCEPTIONS
        not_found      = 1
        OTHERS         = 2.

CALL FUNCTION 'CALCULATE_TAX_ITEM'
    EXPORTING
        i_taxcom        = taxcom
    IMPORTING
        e_taxcom        = taxcom
    EXCEPTIONS
        mwskz_not_defined = 1
        mwskz_not_found   = 2
        mwskz_not_valid   = 3
        steuerbetrag_falsch = 4

```

```

country_not_found = 5
OTHERS            = 6.

IF sy-subrc <> 0.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

p_mwsbp = taxcom-wmwst .

ENDFORM.          " calculate_tax

```

Příloha 2. – Obslužná rutina pro vyvolání PDF formuláře z prostředí SAP

```
*&-----*
*& Report  ZMM_XHASMA00_IVL
*&
*&-----*
*&
*&-----*

REPORT  ZMM_XHASMA00_IVL.

PARAMETERS : po_num TYPE ekpo-ebeln.

data : FM_NAME          TYPE RS38L_FNAM,
       FP_OUTPUTPARAMS  TYPE SFPOUTPUTPARAMS,
       FP_DOCPARAMS     TYPE SFPDOCPARAMS.

* open jo for printing
CALL FUNCTION 'FP_JOB_OPEN'
  CHANGING
    IE_OUTPUTPARAMS      = FP_OUTPUTPARAMS
  EXCEPTIONS
    CANCEL                = 1
    USAGE_ERROR           = 2
    SYSTEM_ERROR          = 3
    INTERNAL_ERROR        = 4
    OTHERS                = 5.
IF SY-SUBRC <> 0.

ENDIF.

* open form
call function 'FP_FUNCTION_MODULE_NAME'
  exporting
    i_name = 'ZMM_INVOICEVERIFICATIONLIST'
  importing
    e_funcname = fm_name.

if sy-subrc <> 0.
  message w001(ZXHASMA00) with 'ZMM_INVOICEVERIFICATIONLIST'.
endif.

* set form parameters
fp_docparams-langu   = 'C'.
fp_docparams-country = 'CZ'.
fp_DOCPARAMS-FILLABLE = 'X'.

* Call the generated function module

CALL FUNCTION FM_NAME
  EXPORTING
    /1BCDWB/DOCPARAMS      = FP_DOCPARAMS
    ordernum               = po_num
  * IMPORTING
  *   /1BCDWB/FORMOUTPUT    =
  EXCEPTIONS
    USAGE_ERROR            = 1
    SYSTEM_ERROR           = 2
    INTERNAL_ERROR         = 3.
IF SY-SUBRC <> 0.
```

ENDIF.

* Close the spool job

CALL FUNCTION 'FP_JOB_CLOSE'

* IMPORTING

* E_RESULT =

EXCEPTIONS

USAGE_ERROR = 1

SYSTEM_ERROR = 2

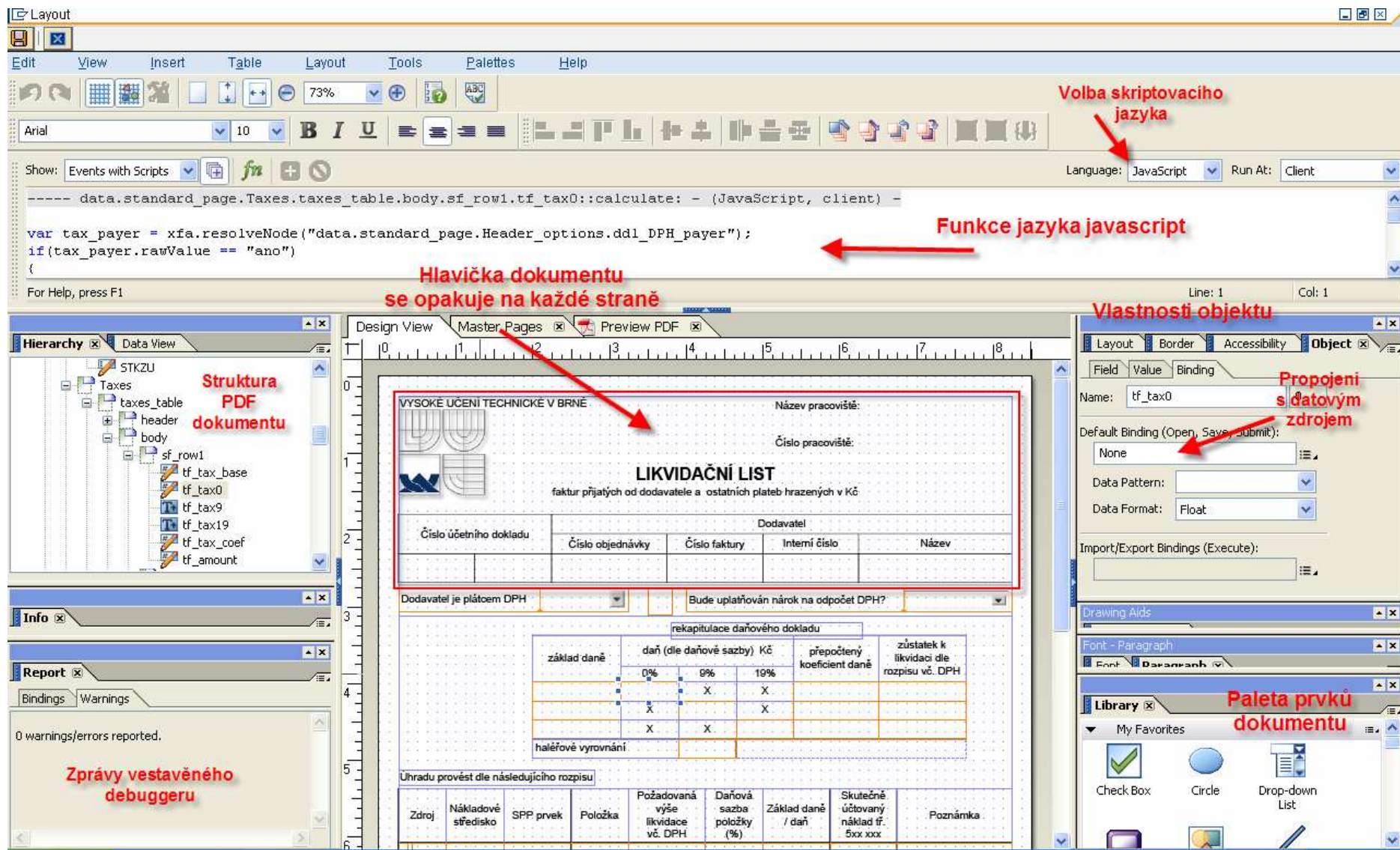
INTERNAL_ERROR = 3

OTHERS = 4.

IF SY-SUBRC <> 0.

ENDIF.

Příloha 3. – Prostředí Adobe LiveCycle Designer



Obr. 8.12: Vývojové prostředí Adobe LiveCycle Designer vestavěné v aplikaci SAP