

VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Katedra elektrotechniky a informatiky

Obor Počítačové systémy

**Zákaznická transakce ZSQS v SAP/R3 pro
objednávání vzorků nových materiálů**

bakalářská práce

Autor: Petr Uher

Vedoucí práce: ing. Pavel Došek

Jihlava 2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce:

Petr Uher

Studijní program:

Elektrotechnika a informatika

Obor:

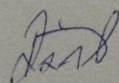
Počítačové systémy

Název práce:

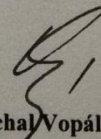
Zákaznická transakce pro informační systém SAP R/3

Cíl práce:

Vytvořit zákaznickou transakci pro informační systém SAP R/3 pro objednávání vzorků nových materiálů od dodavatelů pro potřebu vstupní kontroly. Transakce ZSQS pojme celý proces, tzn. od automatického založení (kopií ze vzoru) kmenových dat materiálu a bezplatné objednávky, přes zpracování seznamu požadovaných dokumentů (certifikáty, testy) a dalších upřesňujících požadavků, až po tisk vzniklého dokumentu požadavku na vzorky a jeho odeslání dodavateli e-mailem přímo z informačního systému.



Ing. Pavel Došek
vedoucí bakalářské práce



Ing. Bc. Michal Vopálenský, Ph.D.
vedoucí katedry
Katedra elektrotechniky a informatiky

Anotace

Práce se zabývá tvorbou zákaznické transakce ZSQS, která slouží na objednávání vzorků materiálů od dodavatelů v podnikovém informačním systému SAP R/3. Transakce byla vytvořena s cílem doplnit chybějící části procesu do informačního systému tak, aby nebylo nutné používat žádné další aplikace. V teoretické rovině jsou zde přiblíženy SAP R/3 objekty a metody použité v práci.

Klíčová slova

SAP, ABAP, ZSQS, objednávání vzorků, zákaznická transakce

Abstract

The thesis pursues the development of ZSQS user transaction, which is to secure the purchasing process of material samples from suppliers within the enterprise information system SAP R/3. The transaction has been developed with the objective of completion of the process missing parts in the information system so that no further implementation of software applications would be needed. From the theoretic viewpoint, an emphasis has been put to disclose and describe SAP R/3 objects and methods used within this thesis.

Key words

SAP, ABAP, ZSQS, samples ordering, customer transaction

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „AZ“).

Souhlasím s umístěním bakalářské práce v knihovně VŠPJ a s jejím užitím k výuce nebo k vlastní vnitřní potřebě VŠPJ.

Byl/a jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje AZ, zejména § 60 (školní dílo).

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom/a toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutí licence.

V Jihlavě dne 16. 5. 2013

.....

Podpis

Poděkování

Na tomto místě bych rád poděkoval svému vedoucímu práce ing. Pavlu Doškovi za konzultace při tvorbě práce a možnost vytvářet ho pod jeho vedením. Dále bych chtěl poděkovat své rodině za maximální podporu během časově náročného studia.

Obsah

1	Úvod	8
2	SAP R/3 jako informační systém v ALCZ.....	10
2.1	Historie SAP R/3.....	10
2.2	Členění SAP R/3	10
2.3	Vývojové prostředí SAP R/3.....	11
2.3.1	ABAP Runtime prostředí.....	12
2.3.2	Transakce a obrazovky	12
2.3.3	Typy ABAP programů.....	14
2.3.4	ABAP Workbench	14
2.3.5	ABAP editor – transakce SE38.....	15
2.3.6	ABAP Dictionary.....	15
2.3.7	Vývojář v ABAP.....	17
3	Transakce ZSQS	19
3.1	Popis procesu a jeho analýza.....	19
3.2	Report ZCZ_PUR_SAMPLES.....	22
3.2.1	Deklarace tabulek	23
3.2.2	Deklarace vstupních parametrů	25
3.2.3	Deklarace proměnných	26
3.2.4	BAPI funkce	26
3.2.5	Vyhledání objednávky a založení kmenových dat materiálu	28
3.3	Report ZCZ_SAMPORD1	29
3.3.1	Výběrová obrazovka	30
3.3.2	Interní tabulky.....	32
3.3.3	Zákaznické tabulky	34
3.3.4	SAP ALV (Advanced List Viewer)	36
3.3.5	Export do Excelu	38
3.3.6	Výběr dat.....	38
3.3.7	SY-SUBRC	40
3.3.8	START-OF-SELECTION	40
3.3.9	Ikony Další data a Změna Objednávky.....	42
3.3.10	Podprogram CALL_DETAIL_SCREEN	43
3.3.11	Screen Painter	44
3.3.12	Smartforms.....	48
3.3.13	Další podprogramy v reportu ZCZ_SAMPORD1	50
3.4	SE63 Překlad v SAP R/3.....	50

4	Závěr	53
5	Seznam použité literatury	54
6	Seznam obrázků	55
7	Seznam použitých zkratek	56
8	Přílohy	57
8.1	Obsah přiloženého CD	57

1 Úvod

Jako téma své bakalářské práce jsem si vybral popis vývoje uživatelské transakce pro podnikový informační systém SAP R/3. Volba to byla jednoznačná, protože s informačním systémem SAP R/3 pracuji již od konce roku 2001, kdy jsem nastoupil do svého minulého zaměstnání, do firmy Bosch Diesel, s.r.o. S tímto okamžikem je rovněž spojeno moje první setkání se s tímto informačním systémem, který tenkrát ještě byl v plné režii jihlavského závodu. Počínaje fyzickým umístěním serverů, přes jejich údržbu, až po podporu uživatelů pracujících se SAP R/3. Celý systém je stavěný modulárně, jednak kvůli tomu, že v něm firma nemusí nutně využívat všechny moduly nebo dokonce je nepotřebuje, ale i z hlediska podpory ze strany profesionálních poradců – systém je tak obsáhlý, že jeden člověk je schopen pojmout detailně, do hloubky maximálně dva moduly. V této době jsem podporoval uživatele pracující s příbuznými moduly FI (Finance) a CO (Controlling). Pomalu jsem pronikal do tajů tvorby queries, v podstatě jednoduchých SQL dotazů, spojujících data z různých tabulek.

K samotnému programování jsem se však dostal až po změně zaměstnání, na přelomu let 2007 a 2008. To jsem nastoupil do svého stávajícího zaměstnání, do firmy Automotive Lighting, s.r.o. Zde se sice primárně starám o uživatele a modul SD (Sales and Distribution), podporu prodeje a částečně i FI a CO, ale svoji působnost jsem rozšířil o programování v jazyce ABAP/4, ve kterém jsou dnes programovány aplikace pro produkty SAP. Proto, když se objevil ve firmě úkol analyzovat a naprogramovat řešení problému, který je popsán v zadání bakalářské práce, rozhodování o vhodném tématu bylo jasné.

K vytvoření transakce ZSQS, objednávání vzorků materiálů od dodavatelů, se přistoupilo vzhledem k neúnosné situaci na oddělení vstupní kontroly ve firmě Automotive Lighting (ALCZ/SQA). Současná evidence objednaných vzorků spočívala v tabulce vytvořené v Microsoft Excelu a umístěné na souborovém serveru oddělení, do které se zapisují jednotlivé požadavky. Pro vlastní objednání vzorků se používá objednávka v informačním systému SAP/R3. Předloha formuláře, který se odesílá dodavateli, je vytvořena v aplikaci Microsoft Word. Vyplněný formulář je potom přikládán jako příloha k e-mailu, který je elektronickou poštou zaslán na adresu dodavatele.

Spuštěním aplikace v produktivním systému se tak odbourá veškerá pomocná administrativa vedená mimo celopodnikový informační systém a celý proces se bude výhradně odehrávat přímo v něm. To s sebou přinese spoustu výhod – počínaje snížením počtu možných lidských chyb, odbouráním různých souborů v různých složkách, přes ruční posílání e-mailů až po naplnění dlouhodobé firemní politiky – co možná největší integrace do informačního systému.

Pokud jde o strukturu práce samotné – nelze zde přesně oddělit teoretickou část od té praktické. Celý koncept jsem pojal tak, že v okamžiku, kdy se objeví nějaký nový důležitý stavební prvek reportu, je zde představen i v teoretické rovině, aby se případný čtenář nemusel vracet.

2 SAP R/3 jako informační systém v ALCZ

2.1 Historie SAP R/3

SAP R/3 je softwarovým produktem společnosti SAP, který slouží pro řízení podniku (Enterprise resources planning – ERP). ERP je informační systém, který integruje a automatizuje velké množství procesů souvisejících s produkčními činnostmi podniku. Typicky se jedná o výrobu, logistiku, distribuci, správu majetku, prodej, fakturaci, a účetnictví. První verze SAPu bylo finanční účetnictví pojmenované R/1. Tato verze byla nahrazena verzí R/2 na konci 70. tých let. R/2 zahrnovala už celou řadu obchodních aplikací a byla na konci 80. a začátku 90 let velmi úspěšná. Úspěšná byla hlavně díky poptávce velkých nadnárodních společností, které potřebovaly informační systém s podporou různých jazyků a různých platebních měn. S rozvojem informačních technologií vydal SAP AG klient – server verzi SAP R/3 („R“ znamená „Real Time Data Processing“ – zpracování dat v reálném čase a „3“ three-tier“ , což značí právě architekturu klient – server). Tato nová architektura je kompatibilní s různými platformami a operačními systémy, jako je Microsoft Windows nebo Unix. Tyto vlastnosti otevřely této verzi dveře pro další zákazníky. Verze byla oficiálně vypuštěna 6. července 1992. Verze prošla svým vývojem, od 1.0A až po 6.0.

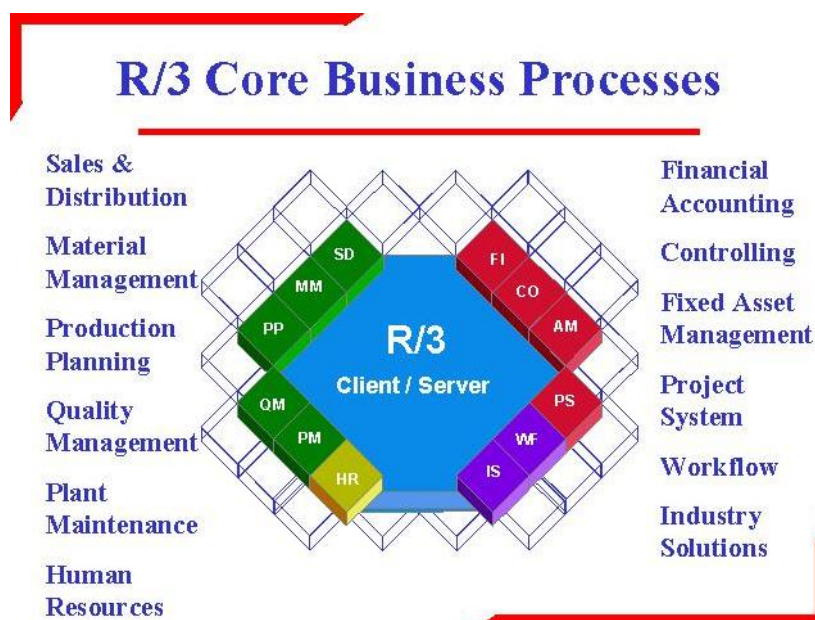
2.2 Členění SAP R/3

SAP R/3 je vnitřně rozčleněn do jednotlivých celků, takzvaných modulů, které se kryjí s logickým členěním typické firmy.

SAP R/3 se skládá z následujících modulů:

- **FI (Financial Accounting)** Finanční účetnictví
- **CO (Controlling)** Kontroloing
- **AM (Asset Management)** Evidence majetku
- **PS (Project system)** Plánování dlouhodobých projektů
- **WF (Workflow)** Řízení oběhu dokumentů
- **IS (Industry Solutions)** Specifická řešení různých odvětví
- **HR (Human Resources)** Řízení lidských zdrojů
- **PM (Plant Maintenance)** Údržba

- **MM (Materials Management)** Skladové hospodářství a logistika
- **QM (Quality Management)** Management kvality
- **PP (Production Planning)** Plánování výroby
- **SD (Sales and Distribution)** Podpora prodeje



Obrázek 1: Moduly SAP R/3

2.3 Vývojové prostředí SAP R/3

Vyvíjet uživatelské programy pro SAP je v současných verzích možné buď v původním programovacím jazyku ABAP nebo od roku 2003 nově i v Javě. Původní verze ABAPu (a jehož jádro je naprogramováno v jazyku C) byla vytvořena jako nástroj pro generování reportů v R/2, aby se později vyvinul do podoby interpretačního jazyka pro vývoj aplikací v R/2. V průběhu 90. let byl ABAP v SAP R/3 uveden jako programovací jazyk 4. generace pod názvem ABAP/4 „Advanced Business Application Programming“. Od této doby jsou aplikace pro produkty SAP psány výhradně v ABAP/4. V roce 1999 SAP uvolnil rozšíření ABAP Objects společně s verzí 4.6C.

2.3.1 ABAP Runtime prostředí

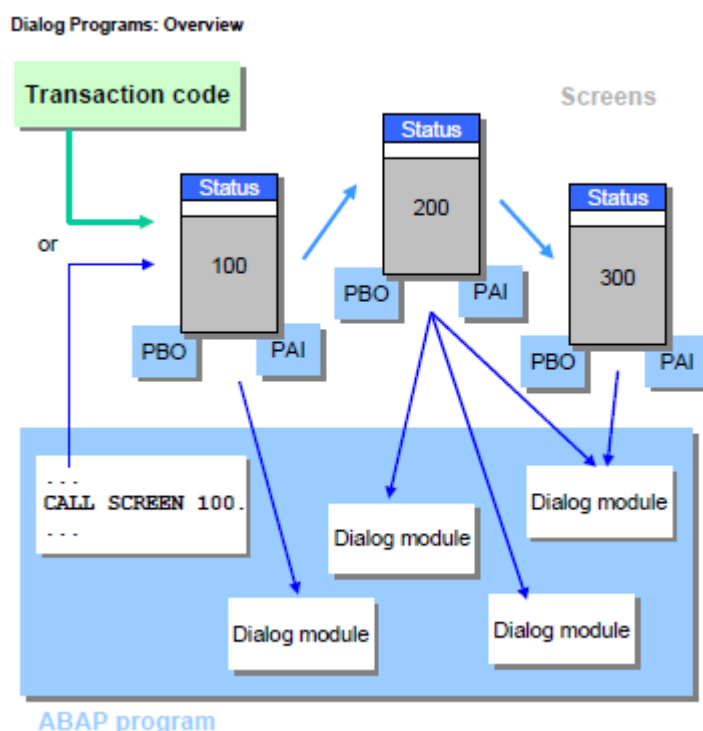
Všechny ABAP programy nejsou, tak jako například Java nebo C++ programy uloženy v externích souborech, ale jsou uvnitř databáze SAP. V databázi je veškerý ABAP kód uložen ve dvou formách: jako zdrojový kód, který může být prohlížen a editován pomocí ABAP Workbench Tools a binární kód reprezentující cosi srovnatelného s Java bytecode. Spuštění ABAP programu je pod kontrolou runtime systému, který je součástí SAP kernelu. Runtime systém kontroluje vykonávání příkazů, sleduje logický tok obrazovek a odezvy na události, jako je klikání uživatelů na tlačítka v programu. Klíčovou komponentou ABAP runtime systému je databázový interface, který překládá databázově nezávislé ABAP příkazy („Open SQL“) na příkazy srozumitelné pro „nativní SQL“. Databázový interface obhospodařuje veškerou komunikaci relační databáze s ABAP programem. To znamená včetně bufferování tabulek a často používaných dat do lokální paměti aplikačního serveru. Kontrolu syntaxe, generování kódu je součástí vrstvy SAP báze. Ve své práci bych chtěl obsáhnout především objekty používané v transakci ZSQS.

2.3.2 Transakce a obrazovky

V terminologii SAPu se rovná spuštění transakce spuštění programu. Termín transakce má v IT světě několik různých významů. V případě SAPu vychází tento pojem ze zkratky OLTP (Online Transaction Processing), kde několik uživatelů pracuje v jednom systému v dialogovém modu. Zde potom „transakce“ představuje požadavek uživatele. Běžný způsob spuštění ABAP kódu probíhá zadáním kódu transakce, např. transakce „MM03“ spustí program pro zobrazení kmenových dat materiálu. Transakce může být, v případě, že uživatel systému ji zná nazpaměť, vepsána přímo do dialogového okna nebo může být dohledána a spuštěna ze systémového nebo uživatelského menu. Další způsob spuštění reportu je vyvoláním jeho jména v transakci SE38. Platí pravidlo, že report nemusí mít transakci na spuštění. Právě v tomto případě se použije již zmiňovaná transakce SE38. V praxi, vzhledem k bezpečnosti systému a systému oprávnění, mívají transakce používané běžnými uživateli definované transakce. Kód transakce se definuje v Repository Browser v ABAP Workbench nebo užitím transakce SE93. V této transakci se definuje vazba na název spouštěného reportu a číslo první obrazovky. I v oblasti názvosloví transakcí je zaveden systém umožňující rychlou orientaci a odhad, do kterého modulu konkrétní transakce patří. Zde uváděné transakce patří mezi tzv.

systémové, zkratka SE znamená (System Environment), následuje pořadové číslo. Zde je potřeba také zdůraznit, že pro zákaznické transakce, a to platí obecně i pro další objekty, jsou vyhrazeny SAPem názvy začínající na „Y“ nebo „Z“. Jako příklad uvedme transakci ZSQS, která je předmětem této práce.

Na transakce velmi těsně navazují obrazovky, hlavně pokud se jedná o reporty řízené dialogem. Běh programu je kontrolován sérií dialogu s uživatelem. Po spuštění transakčním kódem se nastartuje úvodní obrazovka, která umožní uživateli zadat nebo požadovat údaje. Logika toku obrazovek poté reaguje na uživatelské požadavky voláním různých ABAP modulů. Tam se mohou provádět příkazy pro zobrazení dat nebo updatování databáze.



Obrázek 2: Struktura dialogového programu

Druhy transakcí:

- dialogová transakce – v ní je právě běh programu řízen sledem obrazovek
- transakce reportu – provádění programu má zpravidla 3 kroky, zadání vstupních dat (výběrová obrazovka), zpracování dat a výstup dat (sestava)

- varianty transakcí – umožňují přednastavit hodnoty polí použitých v transakci, nastavit atributy polí nebo i skrýt celé obrazovky
- parametrická transakce – umožní přednastavit hodnoty polí na úvodní obrazovce. Tato se dá v transakci potlačit, pokud jsou definovány hodnoty pro všechny její pole.

2.3.3 Typy ABAP programů

ABAP rozlišuje 2 druhy spustitelných programů:

- reporty
- module pool

Reporty

A ty nespustitelné

- moduly INCLUDE
- podprogramy
- funkční skupiny
- objektové třídy
- Interfejsy

2.3.4 ABAP Workbench

Tento nás bude zajímat kvůli tvorbě reportu především. Obsahuje totiž různé nástroje pro tvorbu a editování programů. Nejdůležitější jsou (kódy transakcí jsou uvedeny v závorkách):

- **ABAP Editor** pro psaní a editaci reportů (**SE38**)
- **ABAP Dictionary** pro editaci a zakládání tabulkových struktur (**SE11**)

- **Menu painter** pro vytváření uživatelského menu (**SE41**)
- **Screen painter** pro definování uživatelských obrazovek a jejich logického toku (**SE51**)
- **Function builder** pro vytváření a prohlížení funkčních modulů (**SE37**)
- **Object navigator** jako nástroj zastřešující všechny výše uvedené transakce v jednom (**SE80**)

2.3.5 ABAP editor – transakce SE38

Editor se používá pro vytváření a editaci zdrojových textů reportů a dalších spjatých komponent. Po vyvolání editoru tak máme možnost pracovat s těmito objekty:

- zdrojový kód – vyvolá ABAP editor
- varianty – spustí nástroj pro údržbu variant
- atributy – umožní změnit atributy programu – např. druh programu, který určuje, jak bude proveden
- dokumentace – zde je možné pořídit dokumentaci k reportu. Tato je pak při běhu reportu dostupná v systémovém menu přes volbu Goto -> Documentation
- textové elementy – umožní editovat textové elementy programu. Patří sem všechny texty, které se objeví na výběrové nebo na výstupní obrazovce

2.3.6 ABAP Dictionary

Dictionary obsahuje veškerá metadata, která se vyskytují v SAP systému. Vývojáři používají ABAP Dictionary transakce k zobrazení nebo údržbě těchto metadat. ABAP Dictionary centrálně popisuje a spravuje veškerá data v systému. Je kompletně integrováno v ABAP Development Workbench. Všechny další komponenty Workbench mohou aktivně přistupovat k definicím uloženým v ABAP Dictionary.

Stručný popis nejdůležitějších druhů objektů Dictionary:

- **Tabulky** – jsou to datové kontejnery umístěné v relační databázi, ve většině případů ve vztahu 1:1 mezi definicí tabulky v ABAP Dictionary a definicí té samé tabulky v databázi, tzn. stejné technické názvy, stejná pole). Takovéto tabulky nazývám transparentní. Kromě těchto, obvyklých, existují ještě tabulky netransparentní. Mezi netransparentní patří poolové a clusterové tabulky se naproti transparentním v databázi nezakládají. Data těchto tabulek se ukládají v resp. v clusteru tabulek. Pro poolové a cluster zakládání indexů a technických nastavení nutné.

Dictionary: Zobrazení tabulky

Transp.tabulka: EKKO Aktiv.
Krátký popis: Hlavička nákup.dokladu

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

Náp.vyhl. Instalovaný typ 1 / 143

Pole	K.	I..	Datový prvek	Da...	D...	D...	Krátký popis
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT		3	0Klient
EBELN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EBELN	CHAR		10	0Číslo nákupního dokladu
.INCLUDE	<input type="checkbox"/>	<input type="checkbox"/>	EKKODATA	STRU		0	0Hlavička nákup. dokladu - datová část
BUKRS	<input type="checkbox"/>	<input type="checkbox"/>	BUKRS	CHAR		4	0Účetní okruh
BSSTYP	<input type="checkbox"/>	<input type="checkbox"/>	BSSTYP	CHAR		1	0Typ nákupního dokladu
BSART	<input type="checkbox"/>	<input type="checkbox"/>	BSART	CHAR		4	0Druh nákup.dokladu
BSAKZ	<input type="checkbox"/>	<input type="checkbox"/>	BSAKZ	CHAR		1	0Řídící znak k druhu nákupního dokladu
LOEKZ	<input type="checkbox"/>	<input type="checkbox"/>	ELOEK	CHAR		1	0Znak výmazu v nákupním dokladu
STATU	<input type="checkbox"/>	<input type="checkbox"/>	ESTAK	CHAR		1	0Status nákupního dokladu
AEDAT	<input type="checkbox"/>	<input type="checkbox"/>	ERDAT	DATS		8	0Datum založení záznamu
ERNAM	<input type="checkbox"/>	<input type="checkbox"/>	ERNAM	CHAR		12	0Jméno referenta, který objekt založil
PINCR	<input type="checkbox"/>	<input type="checkbox"/>	PINCR	NUMC		5	0Interval položky
LPONR	<input type="checkbox"/>	<input type="checkbox"/>	LPONR	NUMC		5	0Číslo poslední položky
LIFMR	<input type="checkbox"/>	<input type="checkbox"/>	ELIFM	CHAR		10	0Číslo účtu dodavatele
SPRAS	<input type="checkbox"/>	<input type="checkbox"/>	SPRAS	LANG		1	0Jazykový klíč
ZTERM	<input type="checkbox"/>	<input type="checkbox"/>	DZTERM	CHAR		4	0Klíč platebních podmínek

Obrázek 3: Zobrazení definice tabulky (SE11)

- **Indexy** – zajišťují zrychlený přístup k tabulkovým datům. Jsou definovány podle klíče, který se definuje z polí tabulek. Definice klíče odpovídá nejčastějším dotazům. Klíčů může být k tabulce definováno několik, systém pak sám, podle zadaného dotazu, použije ten nejvýhodnější.
- **Views** – umožňují definovat virtuální tabulky, které neobsahují fyzická data. Obsahují např. omezený výběr polí z jedné nebo několika navzájem pospojovaných tabulek. Views zabírají malé místo v databázi, protože obsahují pouze definice dat.

- **Struktury** – představují komplexní data, podobné struct v C++. Příkladem může být struktura obsahující data z modulu FI.
- **Datový prvek** – těmi jsem měl možná začít, představují ten elementární stavební prvek, ze kterých se skládají tabulky, či struktury. Datový prvek je elementární typ. Popisuje vlastnosti typu (datový typ, délka pole a případně počet desetinných míst) a informace relevantní pro dynpra (vysvětlující texty resp. nápověda k polím) nestrukturovaných datových objektů (polí tabulek resp. struktur nebo proměnných). Pole tabulek resp. struktur se stejným obsahovým významem by měla odkazovat na stejný datový prvek. Tím je zajištěno, že tato pole zůstanou z hlediska jejich vlastností vždy konzistentní. Datový prvek může být v programech ABAP odkazován pomocí TYPE. Tak mohou být v programech ABAP definovány proměnné, které přebírají vlastnosti typu datového prvku.

2.3.7 Vývojář v ABAP

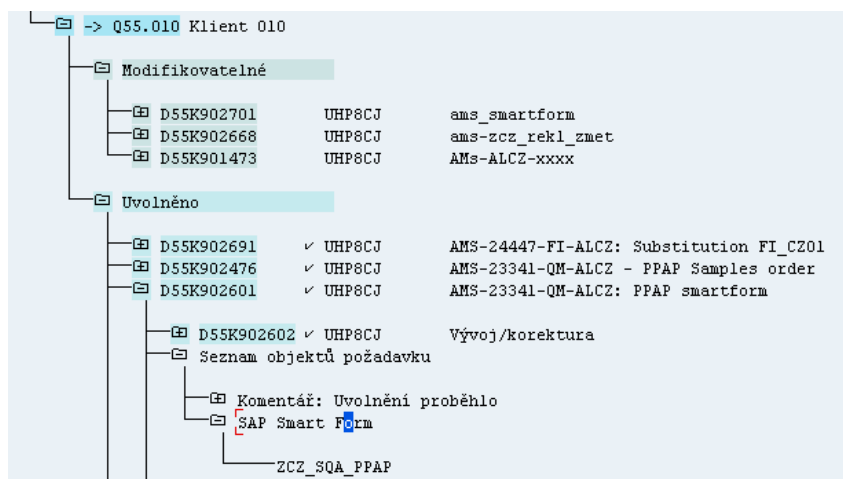
Každý vývojář SAPu se musí, dříve než může začít něco vytvářet, zaregistrovat jako vývojář na portále firmy SAP AG.

SSCR - Registered Users						
Developers Registered by Me = 6						
This list contains all developers you have registered. You may maintain only those developers in installations for which you are authorized.						
Use the first line for entering a search/filter term (* for fuzzy search).						
	Registered user	Customer	Installation	Installation name	Registration date	Registration key
<input type="checkbox"/>	UHP8CJ	517199	0020096424	SAP R/3 Installation	08.03.2011	13701306263801490355
<input type="checkbox"/>	RST8CJ	517199	0020096424	SAP R/3 Installation	25.05.2007	35731413601093639920
<input type="checkbox"/>	NEM8AC	517199	0020096424	SAP R/3 Installation	16.02.2007	07072427941694520678
<input type="checkbox"/>	CN88 II	517199	0020096424	SAP R/3 Installation	24.11.2006	27174467007270308801

Obrázek 4 : Přehled vývojářů na SAP portále

SAP eviduje každou instalaci SAPu všechny vývojáře, kterým na základě registrace přidělí registrační klíč. Ten musí vývojář použít při prvním pokusu o změnu v systému, kdy po něm systém požaduje klíč vývojáře. Pro každou instalaci musí mít vývojář jiný klíč vývojáře. Instalací se rozumí každá instalace SAPu, jejíž hierarchie je standardně třístupňová – skládá se z vývojového, testovacího a produktivního systému. Tyto systémy jsou mezi sebou propojeny tzv. transportními cestami, ve kterých je určeno,

jakým směrem, odkud kam, budou směřovat jednotlivé transporty mezi systémy. Obvykle je to samozřejmě z vývojového do testovacího a z testovacího do produktivního systému. Proces vývoje probíhá tak, že veškeré nové programy, tabulky, obecně objekty se vytvoří ve vývojovém systému. Každý pokus o vytvoření libovolného nového objektu vyžaduje zadání vývojové třídy. Vývojová třída seskupuje ABAP objekty jako jsou transakce, reporty, funkční moduly a další do těchto tříd. To umožňuje strukturovanost R/3 Repository, a také to, že objekty mohou být přiřazeny různým R/3 komponentám. Přiřazení objektu do vývojové třídy je zaznamenáno v Object Directory (TADIR). Toto přiřazení rovněž umožňuje sledování změn a transport objektů mezi systémy. Při vývoji veškeré související objekty a změny se sdružují to tzv. transportního požadavku, aby se jednoduše daly všechny objekty, které spolu logicky souvisí, jednoduše přenášet jako balíček. Potom nehrozí, že se na něco zapomene.



Obrázek 5 : Detail transportního požadavku

Ve vývojovém systému proběhne rovněž první test funkčnosti. Pokud je vše v pořádku, transportuje se transportní požadavek do testovacího systému, kde proběhne testování už za aktivní účasti uživatelů, kteří testovaný vývoj požadovali. Pokud vše dopadne podle očekávání, může se přistoupit k transportu požadavku do produktivního systému. Výhodou této hierarchie je bezesporu to, že systémy jsou konzistentní, stejně nastavené. Jediné, v čem se samozřejmě liší, jsou data. Čas od času se provede kopie produktivního systému do testovacího, čímž se zaktualizují data. Toto se děje obvykle po ukončení roku nebo po/před nějakým updatem systému. Častější aktualizace nepřipadá v úvahu, protože samotná kopie se provádí 1:1 a bývá časově velmi náročná. Záleží samozřejmě

na velikosti databáze. V případě ALCZ, která je v SAPu ještě se třemi dalšími závody, trvá kopie víc než dva dny.

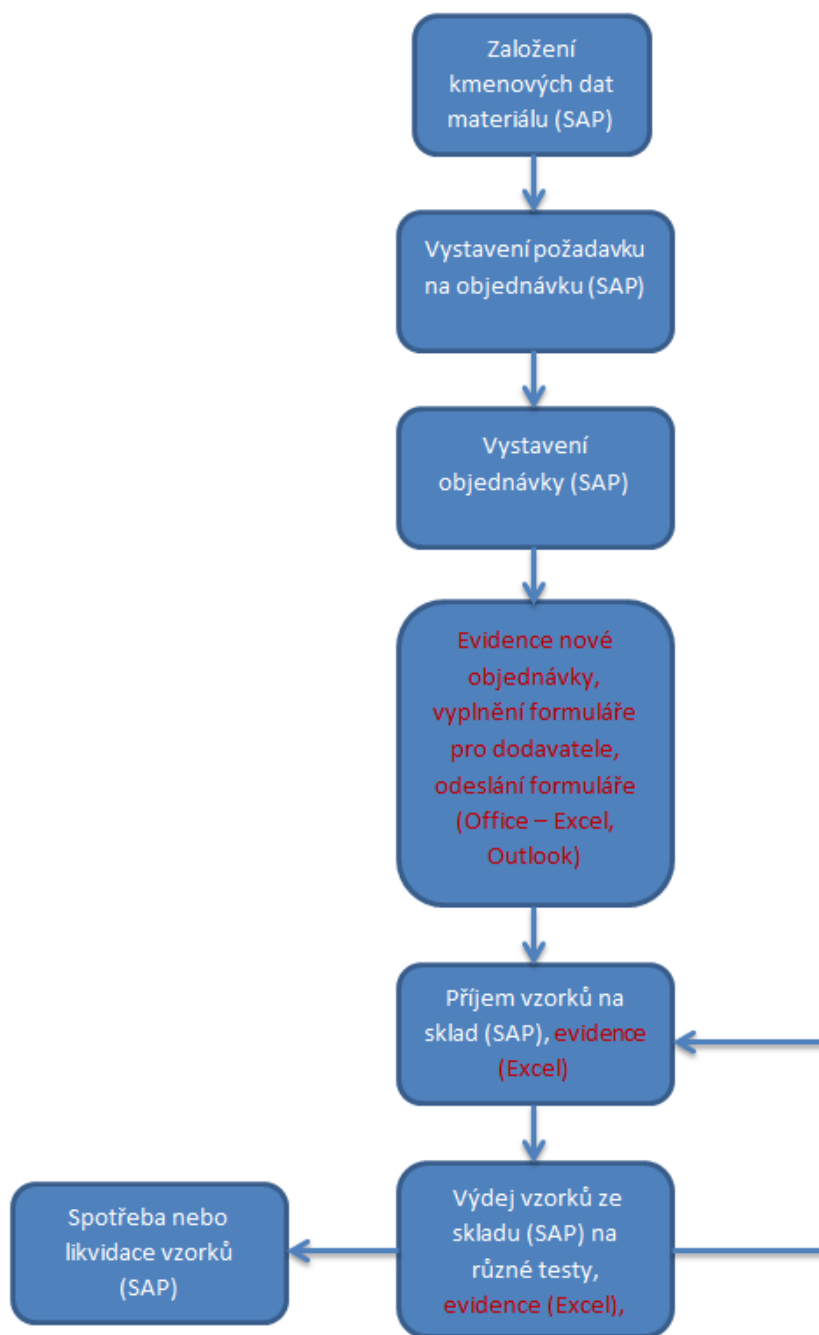
3 Transakce ZSQS

3.1 Popis procesu a jeho analýza

Jak už jsem nastínil v úvodu této práce, ve firmě ALCZ vznikla potřeba integrovat proces objednávání vzorků materiálů od dodavatelů do celopodnikového systému, protože část procesu stejně probíhá v informačním systému – ať už je to vytvoření objednávky, či příjem, uskladnění, popřípadě výdej materiálu.

Oddělení kvality dnes zaujímá ve firmách stále důležitější pozici. Proto se interně člení na několik částí, minimálně na kvalitu vstupní a výstupní. Důvody jsou pochopitelné. Na dnešním trhu panuje tvrdá konkurence, jakékoliv pochybení v oblasti kvality výrobků by mohlo mít katastrofální dopad na ekonomickou prosperitu firmy, ztráta zákazníků by mohla zapříčinit i krach společnosti. Zadání na tuto transakci vzešlo z oddělení, které má na starost kvalitu vstupních, nakupovaných komponent. Oddělení úzce spolupracuje s oddělením nákupu, které potom, na základě výsledků předepsaných testů, nasmlouvá s dodavatelem kontrakt na dodávku komponent.

Na začátku musela proběhnout podrobná analýza celého procesu. V podstatě se dotýká čtyř oddělení – nákupu, kvality, EAP10 a logistiky. Oddělení nákupu zajišťuje založení požadavku na objednávku, objednávku samotnou a kontakt s dodavatelem, zatímco kvalita ve spolupráci s logistikou příjem materiálu, jeho uskladnění, event. přeskladnění, tak jak je zde vidět. Oddělení EAP10 patří organizačně pod centrální vývoj, ale má na starost kromě výkresové dokumentace i zakládání kmenových dat materiálu v SAPu. Níže uvádím jednotlivé kroky celého procesu, červeně jsou zvýrazněné kroky mimo podnikový systém, které budou integrovány do SAPu:



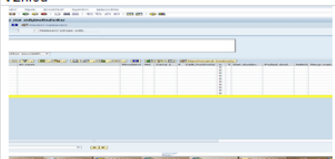
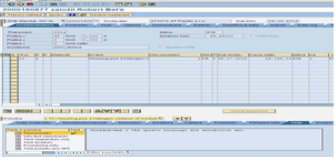

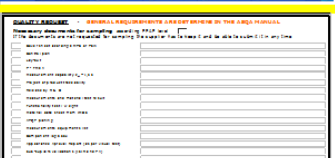
Proces objednávání vzorků v Automotive Lighting, s.r.o.

1. **Založení kmenových dat materiálu, zajišťuje oddělení EAP10.**
2. **Vystavení požadavku na objednávku v SAP R/3 – zajišťuje kvalita.**

3. Vystavení samotné objednávky na vzorky, odeslání objednávky, komunikace s dodavatelem – zajišťuje oddělení nákupu.

4. Příjem materiálu, jeho uskladnění a případné přeskladnění na různé sklady kvůli různým testům a zkouškám – zajišťuje oddělení kvality ve spolupráci s logistikou

Po ukončení fáze analýzy procesu, která zahrnovala schůzky se zástupci jednotlivých oddělení, se s posbíraných poznatků určily otevřené body, které bylo třeba dořešit.

KROK	Vzhled	Popis	Otevřené body	Zodpovědnost
Krok č.1		Vystavení požadavku na objednání vzorků		Z: PUR nebo MSS
Krok č.2		Vystavení objednávky na dodávku dílů pro MSS nebo linku	V některých případech se neobjednávají vzorky pro MSS. Pokud nebude tento krok lze přímo od požadavku provést krok č. 3 (přeskočení kroku č.2)? ICT: tento krok nelze vynechat, protože k němu se bude vázat provedení kopie pro krok č.3. Lze však pro kopii využít libovolnou již dříve vystavenou objednávku !	Z: PUR
Krok č.3		Kopie objednávky na dodávku dílů pro vzorkování Objednávka pro SQA Změna TTNR : z : x xxx xxx xxx xxMxx na: x xxx xxx xxx xxQxx	Doplnit : a) Standardní počet kusů - 10 b) V případě objednávky vzorků - cena 0Kč c) automaticky přidělení skladu vzorků d) Možnost doplnit nákupní skupinu SQA ICT: lze nastavit	Z: PUR
Krok č.4		Specifikace požadavků SQA k objednávce vzorků	Doplnit : a) Možnost změny počtu kusů b) Možnost změny nákupní skupiny ICT: není problém nastavit	Z: SQA
Krok č.5		Odeslání objednávky na vzorky na dodavatele Odeslání - outlock - Email		Z: SQA
Krok č.6		Příjem v ALCZ Sklad vzorků SQA (nutno zřídit v SAP)		Z: LOG
Krok č.7		Vyjádření SQA k danému vzorkování Odeslání zprávy na dodavatele - Outlock - Email	a) V případě zamítnutých vzorků možnost vygenerování kroku č. 3 a 4 b) možnost uzavření objednávky v případě nižšího počtu kusů ICT: lze nastavit jak ukončení objednávky pokud by počet kusů nebyl shodný, tak vrácení do kroku č.3	Z: SQA

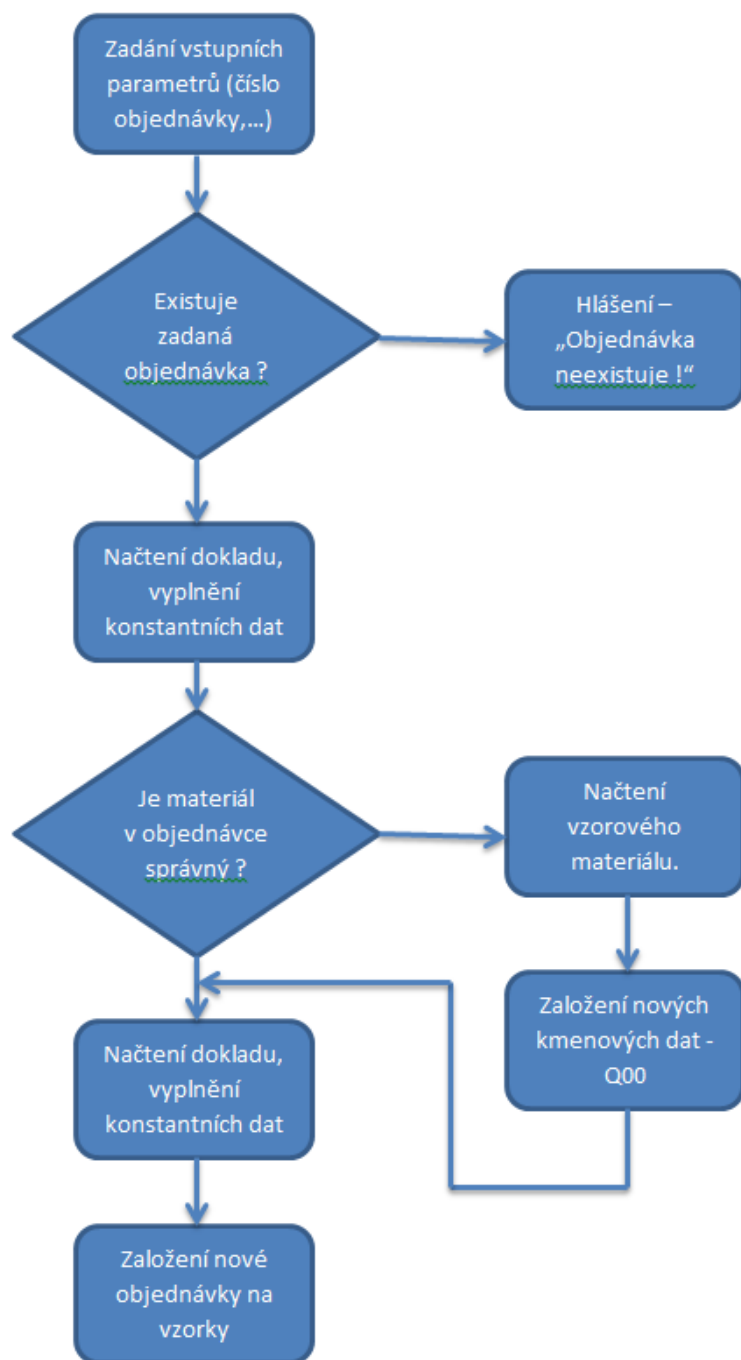
Obrázek 6 : Přehled otevřených bodů

V prvním kroku celého procesu je potřeba založit pořadavek na objednávku, což je v souladu se standardním procesem v SAPu. K tomuto kroku nebyly žádné připomínky. Vystavení objednávky – v tomto kroku v některém případě nebylo v minulosti nutno

vystavovat objednávku. Ve většině případů šlo o specifikaci vlastností, které se měly zakomponovat a zohlednit při vlastním programování reportu. Jak je z obrázku patrné, šlo o nastavení standardního počtu kusů objednávaných vzorků (na 10 ks). Přednastavená hodnota se dá libovolně přepisovat. Dále pak o to, že se bude jednat o objednávku zdarma, neoceněnou. Tato informace byla důležitá vzhledem k výběru BAPI funkce, která se použila pro založení objednávky, podrobněji dále v textu. Automatické přidělení skladu pro přijímané vzorky, to byl další požadavek. Tato informace se udržuje v kmenových datech materiálu a ne jejím základě pak při vlastním procesu příjmu materiálu se informace použije pro samotné uskladnění právě do tohoto skladu. Možnost doplnit (spíš bylo míněno změnit nákupní skupinu v nové objednávce) byl důležitý vzhledem ke skutečnosti, aby jako vzor nové objednávky mohla být použita skutečně jakákoliv vzorová objednávka od různých skupin nákupu, přičemž nákupní skupina patřící kvalitě je jednoznačně daná a musí být správně uvedena na všech objednávkách vzorků.

3.2 Report ZCZ_PUR_SAMPLES

Je prvním ze dvou reportů, které logicky patří do transakce ZSQS. Je volán z reportu, který je svázán přímo s transakcí, reportem ZCZ_SAMPORD1. Zde bych se rád ještě krátce zdržel u zavedeného názvosloví používaného ve firmě Automotive Lighting. Zde veškeré zákaznické reporty vytvořené v jihlavském závodě začínají na „ZCZ“. Je to kvůli snadné identifikaci původu, protože v systému se nacházejí i zákaznické programy, které buď byly migrovány z předchozího systému nebo byly vytvořeny na IT oddělení v Německu (AL Lighting), či v Itálii (Magnetit Marelli).



Vývojový diagram reportu ZCZ_PUR_SAMPLES

3.2.1 Deklarace tabulek

Report začíná, tak jak je zvykem u snad všech programovacích jazyků, deklarací použitých tabulek, proměnných a vstupních parametrů, které se zadávají na úvodní

obrazovce. Report bude pracovat se standardními tabulkami, ve kterých jsou uložena **kmenová data materiálu** – a to konkrétně :

MARA – kmenová data materiálu

MARC – data materiálu závodově závislá

MAKT – popis materiálu

Z oblasti **nákupu** se budou používat tabulky :

EKKO – hlavička nákupního dokladu

EKPO – položky nákupního dokladu

EKET – termíny dodání pro objednávku

LFA1 – kmenová data dodavatelů

T161 – druhy nákupních dokumentů

T024E – tabulka nákupních organizací

A **všeobecné** tabulky:

T001W – seznam závodů

T006A – jazykově závislé texty k měrným jednotkám

DD07T – jazykově závislé texty k doménám

Report v ABAPu vyžaduje deklarovat všechny tabulky, které se ve zdrojovém textu vyskytnou. V tomto případě jsou všechny tabulky standardní, tzn. definované SAPem. Názvy tabulek, pravděpodobně z historických důvodů, vycházejí z německého jazyka. Např. tabulky kmenových dat materiálu začínají na M, nákupní pak na E. Tabulky začínající na T jsou vyhrazeny pro customizing – uživatelské nastavení SAPu.

3.2.2 Deklarace vstupních parametrů

je uvedena klíčovým slovem PARAMETERS

PARAMETERS:

P_WERKS LIKE marc-werks MEMORY ID wrk OBLIGATORY, „povinný parametr závod, zde CP01

P_BUKRS LIKE bkpf-bukrs MEMORY ID buk OBLIGATORY, „povinný parametr účetní okruh, zde C001

P_EBELN like ekko-ebeln, „ číslo objednávky, vzoru

P_EKGRP like ekko-ekgrp. „ nákupní skupina pro novou objednávku

P_LGNUM LIKE ltak-lgnum MEMORY ID lgn OBLIGATORY. „ povinný parametr číslo skladu

SELECT-OPTIONS: p_ebelp FOR ekpo-ebelp. „ nepovinný parametr číslo položky objednávky. Je zde použita klauzule SELECT-OPTIONS kvůli možnosti zadat rozsah položek, např. od – do, nebo jejich výčet.

Pro každý deklarovaný parametr se objeví na obrazovce jedno vstupní pole. Vlevo od vstupního pole se objeví popis pole jako text, který je možné modifikovat jako „výběrový text“. Tyto texty je možné zakládat v různých jazycích.

Obrázek 7: Úvodní obrazovka reportu

3.2.3 Deklarace proměnných

V této části se deklarují globální proměnné, které jsou přístupné v celém programu. Z úsporných důvodů neuvádím celou deklaraci, ale pouze její část. Komplet zdrojový kód bude uveden v příloze na přiloženém nosiči.

*&----> Masterdata creation

```
*&----> Masterdata creation  
DATA: gs_bapi_head LIKE bapimathead,  
      gs_bapi_mara LIKE bapi_mara,  
      gs_bapi_marc LIKE bapi_marc,  
      gs_bapi_mlgn LIKE bapi_mlgn,  
      gs_BAPI_MBEW like BAPI_MBEW.
```

Klauzule DATA uvádí blok, ve kterém jsou definovány proměnné a struktury používané v reportu. V našem případě je potřeba definovat struktury, které používá BAPI funkční moduly použité v reportu:

BAPI_MATERIAL_GET_ALL

BAPI_MATERIAL_SAVEDATA

BAPI_PO_CREATE1

3.2.4 BAPI funkce

Na tomto místě by bylo myslím vhodné uvést několik faktů o BAPI funkčních modulech. BAPI je zkratka a znamená „Business Application Programming Interface“. Je to v podstatě interface, který umožňuje přístup k procesům a datům v systémech jako je např. SAP R/3. BAPI lze použít zejména pro:

- připojení nových R/3 komponent jako je APO a BW
- nesapové systémy
- připojení R/3 systému na Internet
- **uživatelské rozšíření systému**

Posledně uvedené nás zajímá rozhodně nejvíc. BAPI funkce je programovaná tak, že po vyplnění potřebných datových struktur je jejím výstupem například nová faktura, objednávka,... V reportu ZCZ_PUR_SAMPLES jsou použity tyto funkce hned tři, jejich výčet je uveden o něco výše v textu. BAPI funkce obecně poskytuje maximální programátorský komfort, protože kromě definovaných datových struktur obsahuje i ošetření případných výjimek v programu. Vytváří tak ucelený blok, který velice urychlí práci a odstranění případných chyb v programu a jejímž výsledkem je kompletní objekt.

Struktura

BAPIRET2

Aktiv.

Krátký popis













Návratový parametr

Vlastnosti

Komponenty

Nápověda/kontrola zadávání

Pole měny/množství

            Instalovaný typ

1 / 14

Komponenta	R...	Typ komponenty	Datový typ	Délka	Des...	Krátký popis
TYPE	<input type="checkbox"/>	BAPI_MTYPE	CHAR	1	0	Typ hlášení: S Success, E Error, W Warning, I Info, A Abort
ID	<input type="checkbox"/>	SYMSGID	CHAR	20	0	Třída zpráv
NUMBER	<input type="checkbox"/>	SYMSGNO	NUMC	3	0	Číslo zprávy
MESSAGE	<input type="checkbox"/>	BAPI_MSG	CHAR	220	0	Text hlášení
LOG_NO	<input type="checkbox"/>	BALOGNR	CHAR	20	0	Aplikační protokol: Číslo protokolu
LOG_MSG_NO	<input type="checkbox"/>	BALMNR	NUMC	6	0	Aplikační protokol: Interní pořadové číslo zprávy
MESSAGE_V1	<input type="checkbox"/>	SYMSGV	CHAR	50	0	Proměnná zprávy
MESSAGE_V2	<input type="checkbox"/>	SYMSGV	CHAR	50	0	Proměnná zprávy
MESSAGE_V3	<input type="checkbox"/>	SYMSGV	CHAR	50	0	Proměnná zprávy
MESSAGE_V4	<input type="checkbox"/>	SYMSGV	CHAR	50	0	Proměnná zprávy
PARAMETER	<input type="checkbox"/>	BAPI_PARAM	CHAR	32	0	Název parametru
ROW	<input type="checkbox"/>	BAPI_LINE	INT4	10	0	Řádka v parametru
FIELD	<input type="checkbox"/>	BAPI_FLD	CHAR	30	0	Pole v parametru
SYSTEM	<input type="checkbox"/>	BAPILOGSYS	CHAR	10	0	Systém (logický systém) pochází ze zprávy

Obrázek 8: Parametr BAPI funkce

BAPI_MATERIAL_GET_ALL – funkce přečte zadaný materiál, resp. pohledy, které jsou zadány.

BAPI_MATERIAL_SAVEDATA – tato funkce umí založit kmenová data materiálu, opět záleží na zadaném rozsahu.

BAPI_PO_CREATE1 – založí objednávku. Existuje i velice podobná funkce - BAPI_PO_CREATE, která ovšem pro účel reportu má jeden nedostatek – neumí založit objednávku, která je zdarma. Což byl jediný důvod, proč se použila funkce novější.

3.2.5 Vyhledání objednávky a založení kmenových dat materiálu

Tolik k BAPI funkcím a teď zpět k reportu. Po zdárné deklaraci tabulek, vstupních parametrů a proměnných, běh reportu pokračuje hledáním zadané objednávky. Tato část vyhledává podle zadaného čísla objednávky v tabulce hlaviček nákupních dokladů – EKKO. Pokud ji nenajde, není co dělat, report vypíše chybovou hlášku a report skončí. V případě úspěšného hledání se načtou odpovídající položky objednávky. Každá objednávka může mít samozřejmě několik položek, ve výběrových parametrech je lze omezit intervalově nebo vyjmenovat jednotlivě. Ty jsou potom v cyklu procházeny a testuje se, zda je materiál v objednávce určený pro vzorky. To se pozná podle posledních tří znaků čísla materiálu. Pro vzorky bylo dohodnuto, že formát čísla materiálu bude následující:

X – XXX – XXX- XXX – XX**Q00** např. 0132.008.309.00Q00

Pokud tomu tak není, volá se podprogram, který nejdřív vyhledá materiál, který formátem nevyhovuje a zavolá se BAPI funkce BAPI_MATERIAL_GET_ALL. Funkce načte všechny informace z kmenového záznamu materiálu. Nový materiál už se správným číslem materiálu bude založen jako kopie právě tohoto materiálu. Jediné údaje, které se změní, jsou číslo skladu (zadáva se jako vstupní parametr) a znak uskladnění, pole v kmenových datech materiálu, pohledu Řízení skladu 1. Tento znak

Změna materiálu 1301.398.376.00Q00 (AL-Surovina pro vzorky)

Doplňková data Org.úrovně Kontrola dat obrazovky

Text objedn.nákupu **Řízení skladu 1** Řízení skladu 2

Materiál: 1301.398.376.00Q00 FÜHRUNGSTÜCK MBC C117 HAL LE

Závod: CP01 Automotive Lighting Jihlava

Č.skladu: C11 CZ Pávov/Střítež

Všeobecná data

Zákl.měrná jednotka	KS	Č.nebezpečné látky	
Měrná jednotka WM		Hmotnost brutto	
Výdej.měrná jednotka		Objem	
Návrh MJ z mat.		Spotřeba kapacit	/
TypSklProPlPřiprK0db		<input type="checkbox"/> Schv.Protšarže nutný	
<input type="checkbox"/> Povin.vedení šarží			

Strategie skladov.

ZnTypuSkl:Vyskl.		Zn.: Uskladnění WQ1	<input type="checkbox"/>
Znak zóny skladu		Znak sektor.skl.	

Obrázek 9 : Znak uskladnění WQ1 v kmenových datech materiálu

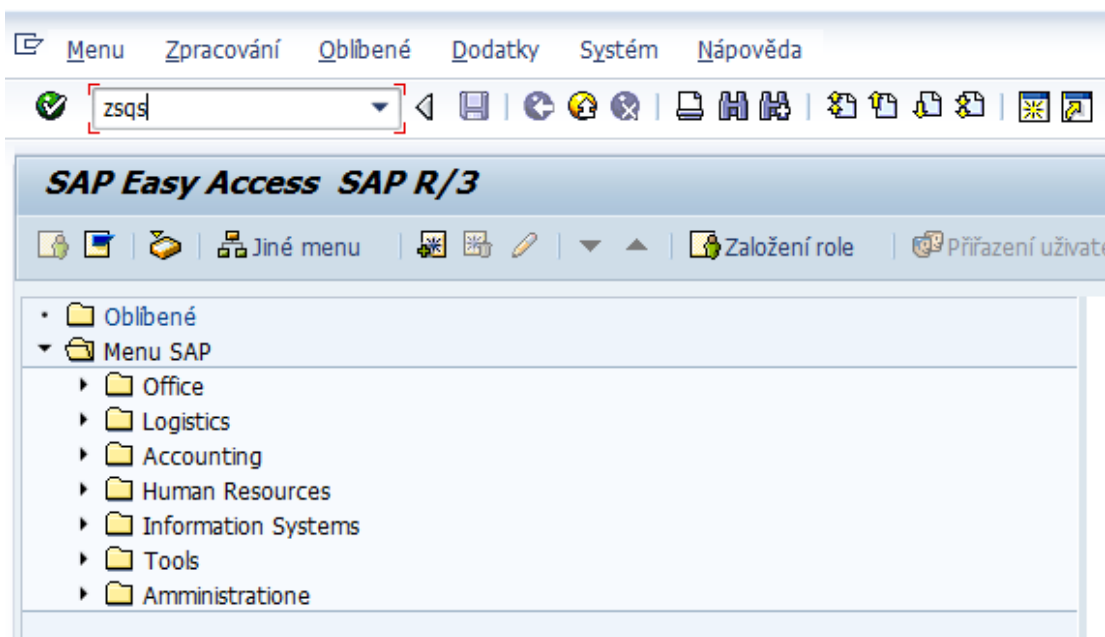
řídí, aby systém při uskladnění preferoval určité typy skladů.

Samotný materiál se vzápětí, v běhu programu, založí další BAPI funkcí - BAPI_MATERIAL_SAVEDATA. Pokud vše proběhne v pořádku, existuje v tuto chvíli nový materiál, který má číslo materiálu podle domluvené konvence a další požadované vlastnosti – správné číslo skladu a znak uskladnění. Může se tedy v běhu programu přistoupit k poslední události, kterou je založení objednávky. Tato objednávka je zdarma (zabezpečeno znakem, tudíž se neočekává faktura), má definované dodací podmínky (Incoterms), množství na položce je rovno pěti. Samotná operace je zabezpečena BAPI funkcí BAPI_PO_CREATE1. Pokud vše proběhne jak má, všechny kontroly proběhly v pořádku, výsledkem je nová objednávka na vzorky. A to je vlastně požadovaný výstup celého reportu.

Jak už jsem napsal na samotném začátku, report ZCZ_PUR_SAMPLE představuje pouze jediné tlačítko v budoucí transakci ZSQS (aktuálně leden 2013 se transakce testuje v systému Q55) .

3.3 Report ZCZ_SAMPORD1

Dal by se s trochou nadsázky označit jako hlavní program. Zabezpečuje všechny ostatní funkce popsané v procesu. Začneme od začátku, první krok je spuštění transakce. Transakci vepíšeme do příkazového řádku a stiskneme klávesu Enter:



Obrázek 10 : Systémové menu SAP R/3 a příkazový řádek

3.3.1 Výběrová obrazovka

Transakce ZSQS (nezáleží na velikosti písmen) je spojena v definici transakce s reportem ZCZ_SAMPORD1 a číslem startovací obrazovky. Po jejím zavolání proběhne kontrola oprávnění podle uživatelského profilu a v případě, že je vše v pořádku, objeví se uživateli na obrazovce tato obrazovka:

Program Zpracování Skok Systém Nápověda

PPAP - Samples order

Výběr dat

Závod	CP01		
Nákupní skupina	c*	Do	
Dodavatel		Do	
Materiál		Do	
Datum založení objednávky		Do	
Objednávka	2000169715	Do	

Další výběrová kritéria

Číslo změny		Do	
Status		Do	
Konečné rozhodnutí		Do	
Datum dodání		Do	
Kód rozhodnutí		Do	
Datum založení		Do	
Založil(uživatel)		Do	

Varianta zobrazení

Layout	
--------	--

Funkce

Kopírování objednávky

Obrázek 11 : Úvodní obrazovka transakce ZSQS

Pro tuto tzv. selection screen není nutno použít Screen Painter, na rozdíl od těch dalších. V každém reportu má automaticky přiděleno číslo 1000. Screen Painter zde nahrazují příkazy ABAP v logickém sledu. Systém potom na základě těchto příkazů sám vygeneruje obrazovku. Celá obrazovka je rozdělena na čtyři části – výběr dat, další

výběrová kritéria, varianta zobrazení a funkce. Ve zdrojovém kódu dělení na části vypadá takto:

```
SELECTION-SCREEN BEGIN OF BLOCK blok1 WITH FRAME TITLE text-r01.
parameters:
    P_werks LIKE marc-werks MEMORY ID wrk OBLIGATORY,
    P_BUKRS LIKE bkp-bukrs MEMORY ID buk OBLIGATORY,
    P_EBELN like ekko-ebeln,
    P_EKGRP like ekko-ekgrp.

SELECT-OPTIONS: p_ebelp FOR ekpo-ebelp.
parameters: P_LGNUM LIKE ltak-lgnum MEMORY ID lgn OBLIGATORY.

*      P_BUKRS like ekko-bukrs, bude naplněno programem
DATA: P_LIFNR like ekko-lifnr,
      P_BSART like ekko-bsart,
      P_BEDAT like ekko-bedat,
      P_EKORG like ekko-ekorg,
      newmatnr like marc-matnr,
      shiftmtnr like marc-matnr,
      pracmatnr like marc-matnr,
      delka(4) TYPE N.
SELECTION-SCREEN END OF BLOCK blok1.
```

Každý blok je uveden příkazem SELECTION-SCREEN BEGIN OF BLOCK, za ním jsou uvedeny vlastnosti bloku (WITH FRAME) a jeho titulek. Titulek je uložený v textových elementech, patřících k reportu. Textové elementy pak mohou existovat v několika jazycích, přímo z prostředí ABAP editoru mohou být překládány z originálního jazyka do různých dalších jazykových mutací. V bloku 3 je definováno tlačítko typu PUSHBUTTON s textem “Kopírování objednávky” a uložení hodnoty “ZCOMM1” do USER-COMMAND. Proměnná se uloží do struktury SSCRFIELDS.

Následuje definice výběrových parametrů. Zde rozlišujeme mezi PARAMETERS a SELECT-OPTIONS. První z nich neumožňuje zadat vstupní hodnoty pomocí rozmezí, pouze lze udělat výčet hodnot. U polí, která podléhají customizingu mohou po stisknutí tlačítka F4 nebo kliknutím na pravý okraj pole dostat na výběr všechny možné hodnoty k výběru. To uživateli usnadní práci a zároveň je vstupní hodnota kontrolována oproti hodnotám definovaným. Při chybném zadání systém neumožní pokračování zadávání a ve spodní části obrazovky se zobrazí zpráva typu E. To uživatele donutí opravit svá zadání. SELECT-OPTIONS oproti tomu umožní zadávat hodnoty v učitém interval nebo naopak nějaký interval zakázat i pořídit výčet hodnot. Další klauzule u příkazu PARAMETERS je LIKE. Klauzule, podobně jako TYPE se používá v různých ABAP příkazech pro definici druhu dat. Nový datový objekt převezme technické atributy existujícího objektu. V tomto případě parametr p_werks zdědí

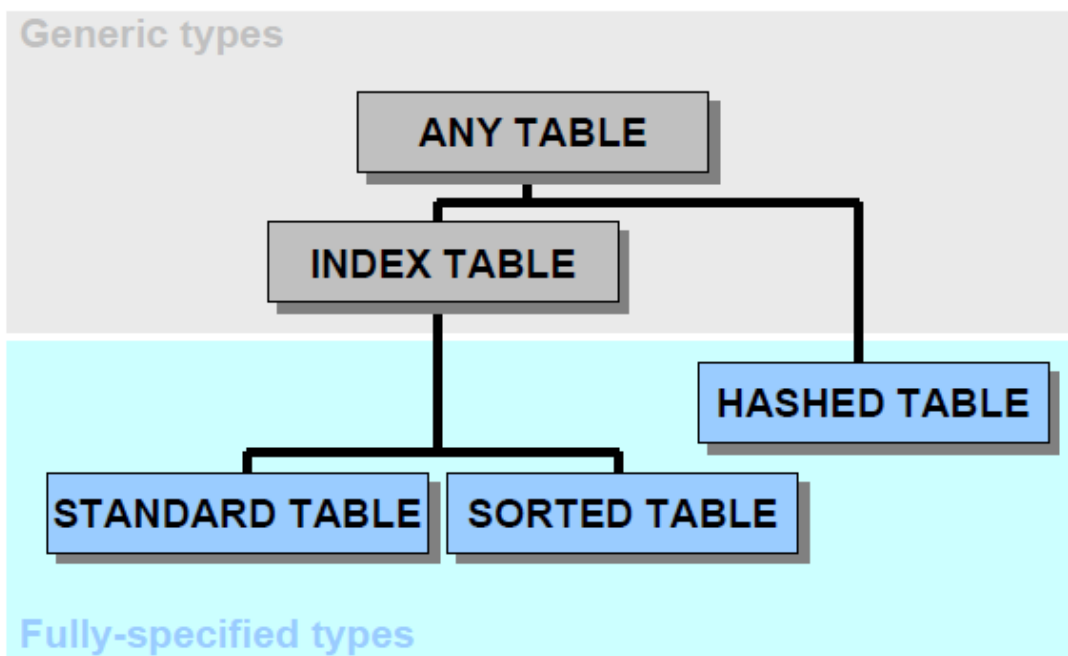
vlastnosti datového prvku werks z tabulky MARC. Jdeme dál, MEMORY ID wrk uloží do paměťové proměnné wrk hodnotu závodu, OBLIGATORY znamená povinné pole, které musí být zadáno. Příkaz SELECTION-SCREEN SKIP vytvoří volný řádek na obrazovce. Posouváme se dál v reportu, následuje deklarace dat. Jako první tu jsou interní tabulky.

3.3.2 Interní tabulky

Interní tabulka v SAPu je dynamická, sekvenční datová sada, ve které všechny záznamy mají tu samou strukturu a klíč. Data v ní jsou načtena z tabulek a umístěna v operační paměti. Znamená to, že existují po dobu běhu reportu. Interní tabulky mohou být trojího druhu :

- standardní (standard tables)
- seřazené (sorted tables)
- hešovací (hashed tables)

Hierarchy of Table Types



Obrázek 12 : Druhy interních tabulek v SAPu

Nejčastěji používané jsou standardní tabulky, kde se k datům přistupuje za použití indexu. Přístupová doba k jednotlivým záznamům interní tabulky se lineárně zvyšuje v závislosti na počtu záznamů tabulky. Obvykle se v praxi tabulka naplní daty a nakonec seřadí. V reportu používám několik interních tabulek, nejzásadnější je tabulka `gt_result`, která obsahuje v podstatě veškeré údaje, se kterými se pracuje v reportu.

Deklarace samotná je uvedena příkazem `BEGIN OF`, následuje název interní tabulky, klauzule `OCCURS` má ten samý význam jako `INITIAL SIZE`. V případě použití parametru "0" je inicializační velikost tabulky řízena systémem.

```
*---> data declaration
DATA: BEGIN OF gt_result OCCURS 0,
        icon1(4),
        icon2(4),
        werks LIKE ekpo-werks,      "závod
        ebeln LIKE ekko-ebeln,      "Objednávka
        ebelp LIKE ekpo-ebelp,      "Položka
        matnr LIKE ekpo-matnr,      "Materiál
        maktx LIKE makt-maktx,      "Text mater.
        menge LIKE ekpo-menge,      "množství
        meins LIKE ekpo-meins,      "jednotka
        ekgrp LIKE ekko-ekgrp,      "Nákupní skupina
        lifnr LIKE ekko-lifnr,      "Dodavatel
        namel LIKE lfal-namel,      "Jméno dodavatele
        unsez LIKE ekko-unsez,      "Naše značka - projekt
        eindt LIKE eket-eindt,      "datum dodávky
        budat LIKE ekbe-budat,      "datum dodávky
        belnr LIKE ekbe-belnr,      "číslo příjemky
        aennr LIKE zcz_smprd-aennr, "číslo změny
        aedtm LIKE zcz_smprd-aedtm, "datum změny
        descr LIKE zcz_smprd-descr, "popis důvodu
        sttus LIKE zcz_smprd-sttus, "status
        dlvdt LIKE zcz_smprd-dlvdt, "Datum dodání
        findc LIKE zcz_smprd-findc, "Konečné rozhodnutí
        dccod LIKE zcz_smprd-dccod, "Kód rozhodnutí
        frdat LIKE zcz_smprd-frdat, "Konec uvolnění materiálu
        mengs LIKE zcz_smprd-mengs, "Množství vzorků
        poznm LIKE zcz_smprd-poznm, "poznámka
        poznl LIKE zcz_smprd-poznl, "poznámka 2
        crdat LIKE zcz_smprd-crdat, "Datum založení
        crtim LIKE zcz_smprd-crtim, "čas založení
        crusr LIKE zcz_smprd-crusr, "Uživatel, který objekt založil
        sedat LIKE zcz_smprd-sedat, "Datum odeslání
        setim LIKE zcz_smprd-setim, "Čas odeslání
        nams1 LIKE zcz_smprd-nams1, "Jméno 1
        mail1 LIKE zcz_smprd-mail1, "E-mailová adresa1
        nams2 LIKE zcz_smprd-nams2, "Jméno 1
        mail2 LIKE zcz_smprd-mail2, "E-mailová adresa2
        reqnr LIKE zcz_smprd-reqnr,
    END OF gt_result.
```

3.3.3 Zákaznické tabulky

Vedle standardních tabulek si může uživatel systému SAP definovat i svoje tabulky, které se nazývají zákaznické. V reportu jich používám hned několik - ZCZ_SMPRD, ZCZ_SMPRD_D1, ZCZ_SMPRD_C1, ZCZ_SMPRD_C1T a ZCZ_T9DOM. Zákaznické tabulky se definují v transakci SE11 – ABAP Dictionary. Po zadání názvu, krátkého popisu, vývojové třídy se může programátor pustit do definice struktury nového objektu:

Dictionary: Zobrazení tabulky

Transp.tabulka: ZCZ_SMPRD iv.
 Krátký popis: SQA - objednávka vzorků

Vlastnosti | Expedice a údržba | Pole | Nápověda/kontrola zadávání | Pole měny/množství

Náp.vyh. | Instalovaný typ | 1 / 25

Pole	K..	I...	Datový prvek	Dato...	Délka	Des...	Krátký popis
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Klient
WERKS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	WERKS_D	CHAR	4	0	Závod
EBELN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EBELN	CHAR	10	0	Číslo nákupního dokladu
EBELP	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EBELP	NUMC	5	0	Číslo položky nákupního dokladu
REQNR	<input type="checkbox"/>	<input type="checkbox"/>	ZCZ_REQNR	CHAR	15	0	Číslo objektu
AENNR	<input type="checkbox"/>	<input type="checkbox"/>	AENNR	CHAR	12	0	Číslo změny
AEDTM	<input type="checkbox"/>	<input type="checkbox"/>	AEDTM	DATS	8	0	Datum změny
DESCR	<input type="checkbox"/>	<input type="checkbox"/>	SCMG_REASON_DESC	CHAR	80	0	Popis důvodu
SITUS	<input type="checkbox"/>	<input type="checkbox"/>	ZCZ_SITUS	CHAR	1	0	Status
DLVDI	<input type="checkbox"/>	<input type="checkbox"/>	RL_DELIVDATE	DATS	8	0	Datum dodání
FINDC	<input type="checkbox"/>	<input type="checkbox"/>	ZCZ_FINDC	CHAR	1	0	Konečné rozhodnutí
DCCOD	<input type="checkbox"/>	<input type="checkbox"/>	ZCZ_DCCOD	NUMC	4	0	Kód rozhodnutí
FRDAT	<input type="checkbox"/>	<input type="checkbox"/>	MEABMDE	DATS	8	0	Konec uvolnění materiálu
MENG3	<input type="checkbox"/>	<input type="checkbox"/>	MENGE_D	QUAN	13	3	Množství
POZNM	<input type="checkbox"/>	<input type="checkbox"/>	AKB_NOTE	CHAR	80	0	Poznámka
POZN1	<input type="checkbox"/>	<input type="checkbox"/>	AKB_NOTE	CHAR	80	0	Poznámka
SEDAI	<input type="checkbox"/>	<input type="checkbox"/>	CVDESEDAI	DATS	8	0	Datum odeslání
SETIM	<input type="checkbox"/>	<input type="checkbox"/>	CVDESETIM	TIMS	6	0	Čas odeslání
CRDAI	<input type="checkbox"/>	<input type="checkbox"/>	FRMLE_CRDAI	DATS	8	0	Datum založení
CRTIM	<input type="checkbox"/>	<input type="checkbox"/>	/SAPPO/DTE_DIAL...	TIMS	6	0	Technický čas založení
CRUSR	<input type="checkbox"/>	<input type="checkbox"/>	BU_CRUSR	CHAR	12	0	Uživatel, který objekt založil
NAMS1	<input type="checkbox"/>	<input type="checkbox"/>	NAMEV	CHAR	50	0	Celé jméno (potvrzování)
MAIL1	<input type="checkbox"/>	<input type="checkbox"/>	AD_SMTPADR	CHAR	241	0	E-mailová adresa
NAMS2	<input type="checkbox"/>	<input type="checkbox"/>	NAMEV	CHAR	50	0	Celé jméno (potvrzování)
MAIL2	<input type="checkbox"/>	<input type="checkbox"/>	AD_SMTPADR	CHAR	241	0	E-mailová adresa

Obrázek 13 : Zobrazení struktury zákaznické tabulky

V tabulce se u jednotlivých polí musí definovat následující:

- název tabulkového pole (název pole smí obsahovat jen písmena, číslice a podtržítka. Přitom nesmí být číslice na první pozici a podtržítka na druhé a třetí pozici názvu)
- klíčové pole tabulky (klíčová pole tabulky musí být souvisle na začátku tabulky, tzn. nesmí být žádné neklíčové pole mezi dvěma klíčovými poli)
- iniciální hodnota pro pole (znak, že pro toto pole je vynucen znak NOT NULL)
- datový prvek – pokud je zadán datový typ a počet míst (event. počet desetinných míst), zůstane tento sloupec prázdný. Pokud je poli přiřazen datový prvek, jsou jeho atributy převzaty z něj. Přebírají se také sémantické vlastnosti datového prvku (nápopěda k polím, zobrazení pole ve vstupní masce pomocí textů klíčových slov, nadpisy sloupců ve výstupních sestavách obsahu tabulky a formátování výstupu pomocí identifikací parametrů).
- datový typ - datový typ popisuje datový formát v uživatelském rozhraní. Použije-li se v programech ABAP pole tabulky resp. struktury nebo datový prvek, pak se datový typ zkonvertuje do formátu používaného procesorem ABAP. Při založení tabulky v databázi se datový typ tabulkového pole zkonvertuje do odpovídajícího datového formátu použitého databázového systému
- délka – počet znaků, počet platných pozic pole bez editačních znaků (např. čárky nebo body)
- počet desetinných míst - počet přípustných míst za desetinnou čárkou hodnoty
- krátký popis - Krátký popis se používá jako vysvětlující text při vytvoření seznamů a v dokumentaci (nápopěda F1)

Podíváme-li se na definici interní tabulky, je zde vidět, že velká část datových polí je ukládána právě do tabulky ZCZ_SMPRD. V tabulce ZCZ_SMPRD_C1 se ukládají doplňková data, resp. je to číselník požadavků dokumentů na dodavatele. Dokumenty jsou rozdělené na externí (pole RQTYP) a interní, je tam jejich popis (pole TRORDERTXT) a ukládají se zde i varianty výběrů dokumentů. Je možné definovat až pět kombinací nejčastěji používaných požadavků na dodavatele. A tabulka ZCZ_SMPRD_D1 je v podstatě vazba mezi tabulkami zcz_smprd a zcz_smprd_c1. Pro

každý požadavek na vzorky (REQNR) je v tabulce seznam externích a interních dokumentů, které jsou požadovány. ZCZ_SMPRD_C1T je další z tabulek. Tabulka má popis „Objednávka vzorků-požadavky na dokumenty dodavatel – texty“, jsou v ní uloženy jazykově závislé texty k požadavkům na dodavatele. Jazykovou nezávislost zabezpečuje pole SPRAS. Pole je speciálního datového typu LANG – jazykový klíč. Vlastní formát pole pro zvláštní funkce. Jazykový klíč se v uživatelském rozhraní zobrazuje jako dvoumístný, ale v databázi je uložen pouze jako jednomístný. Konverze mezi zobrazením v uživatelském rozhraní a v databázi se provádí pomocí exitu konverze ISOLA. Tento exit konverze se při aktivaci automaticky přiřadí doméně s datovým typem LANG. Konečně poslední zákaznickou tabulkou, používanou v reportu je tabulka ZCZ_T9DOM. Ta se používá pro zvláštní účely tz. ALV reportu, o kterém ještě bude řeč v následujícím textu.

3.3.4 SAP ALV (Advanced List Viewer)

Další část deklarace dat je věnována definici proměnných a datových struktur, které jsou potřeba pro fungování Advanced List Vieweru. ALV je nejpoužívanější zobrazení výsledků, sestavy kde je potřeba výstup ve formě tabulky nebo hierarchického uspořádání. Má bohaté možnosti, lze jednoduše data v tabulce řadit, filtrovat či různě seskupovat. V prvních verzích byl výstup realizován pomocí funkčních modulů, od verze SAP R/3 4.6 C je tu koncept objektově orientované třídy. Následující funkční moduly jsou nepostradatelné pro fungování ALV :

- REUSE_ALV_FIELDCATALOG_MERGE – funkční modul zabezpečuje seznam (katalog) polí, metadata z definované struktury interní tabulky
- REUSE_ALV_GRID_DISPLAY – tento funkční modul vytvoří ALV tabulkový výstup
- REUSE_ALV_COMMENTARY_WRITE – funkční modul zobrazí hlavičku ALV sestavy

Jak jsem již uvedl, později SAP pro ALV výstup uvolnil objektově orientované třídy **CL_GUI_ALV_GRID**, **CL_GUI_TREE**. Tyto třídy však nebyly úplně kompletní, proto byly nahrazeny třídami **CL_SALV_TABLE**, **CL_SALV_HIERSEQ_TABLE**.

PPAP - Samples order

Aktualizace

Obrázek 14 : ALV sestava v transakci ZSQS

Na obrázku vidíme detail hlavičky sestavy ALV používané v transakci ZSQS. Pod titulkem vidíme ikony nabídek. Vedle doprogramovaného tlačítka Aktualizace následují standardní tlačítka:

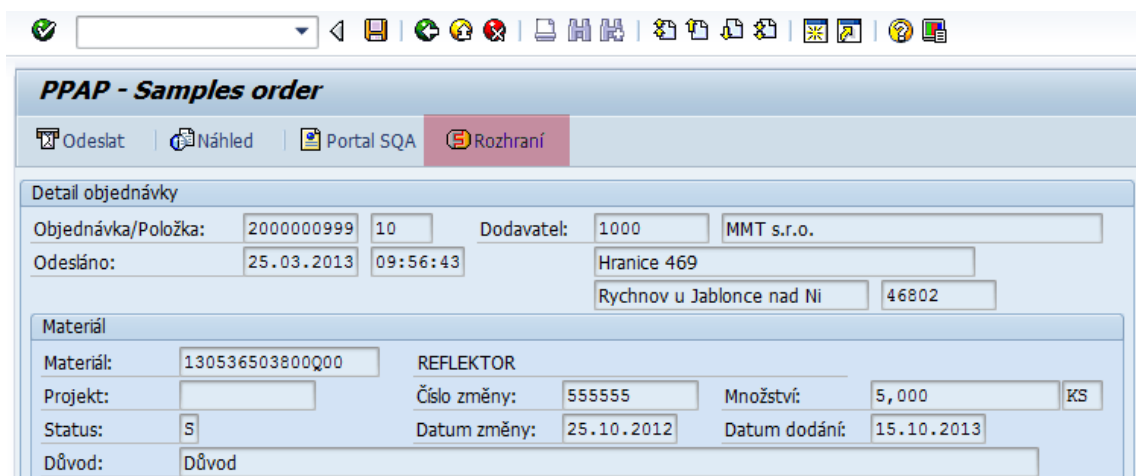
- zobrazení detailu – po označení řádku a kliknutí na ikonu se zobrazí v novém okně detail položky
- označení sloupce – zobrazí ve výstupu pouze vybrané sloupce
- označení všech/ zrušení označení záznamů
- třídění – sestupné nebo vzestupné
- filtr – umožňuje definovat uživatelský filtr i podle složitých podmínek
- sumace – vytváří součty nebo mezisoučty
- spreadsheet – funkce předává data přes XXL interface. Sestava může být uložena jako dokument SAPoffice, další možnost je uložit ji jako soubor na lokální disk počítače nebo konvertovat data pro otevření v Microsoft Excel
- ABC analýza – zobrazí klasickou analýzu. Funkce najde uplatnění především v controllingových sestavách
- Layouty – umožní měnit pohled na zobrazovaná data definicí jejich layoutů – výběrem sloupců, použitím filtrů, sumace, třídění. Každý layout lze potom uložit buď jako přístupný pro jakéhokoliv uživatele nebo jen přístupné pro právě přihlášeného uživatele

3.3.5 Export do Excelu

Další část definuje include, proměnné a tabulku pro export dat do programu Microsoft Excel.

```
*&----> data export to excel
INCLUDE ole2incl.
* handles for OLE objects
DATA: h_excel TYPE ole2_object,           " Excel object
      h_map1 TYPE ole2_object,           " list of workbooks
      h_map TYPE ole2_object,           " workbook
      h_zl TYPE ole2_object,           " cell
      h_f TYPE ole2_object.           " font
TABLES: spfli.
DATA h TYPE i.
```

V transakci je naprogramovaná funkcionalita zavolání Excelu, otevření nového prázdného sešitu a vyexportování dat. Exportují se pole číslo objednávky, číslo položky, číslo materiálu, počet kusů a měrná jednotka.



Obrázek 15 : Ikona Rozhraní – spuštění Excelu a export dat

3.3.6 Výběr dat

Po fázi deklarace se dostáváme na začátek řízení reportu. Událost AT SELECTION-SCREEN zachytává události, které mohou nastat během interakce s výběrovou obrazovkou (tzv. události PAI – Process After Input) .

```
*-----
-*
```

```

*               A T   S E L E C T I O N - S C R E E N
*-----
- *

AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_vari.
  g_r_vari-report = sy-repid.
  CALL FUNCTION 'REUSE_ALV_VARIANT_F4'
    EXPORTING
      is_variant      = g_r_vari
    IMPORTING
      es_variant      = g_r_vari
    EXCEPTIONS
      not_found       = 1
      program_error   = 2
      OTHERS          = 3.

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

*&-----
- *
*&      SELECTION-SCREEN
*&-----
- *
*&---> button
AT SELECTION-SCREEN.
  IF sscrfields-ucomm = 'ZCOMM1'.
    SUBMIT zcz_pur_samples VIA SELECTION-SCREEN AND RETURN.
  ENDIF.

*&-----
& *
*&----> start of selection
START-OF-SELECTION.

  PERFORM select_data.
  PERFORM run_alv.

END-OF-SELECTION.

*&-----

```

Funkce REUSE_ALV_VARIANT_F4 je volána, pokud uživatel chce změnit variantu zobrazení Advanced List Vieweru. Pokud nastane nějaká chyba, tzn. SY-SUBRC je různé od nuly, je generována chybová hláška s ID, typem, číslem a textovým popisem chybové hlášky. V dalším bloku se hlídá stisk definovaného tlačítka „Kopírování objednávky“. Pokud je na tlačítko kliknuto, zavolá se report ZCZ_PUR_SAMPLES, který se provede a vrátí řízení běhu programu zpět na výběrovou obrazovku.

3.3.7 SY-SUBRC

Patří mezi systémová pole ABAP. Tato pole jsou aktivní ve všech reportech. Jsou plněna automaticky runtimeovým prostředím a jako taková mohou být čtena během běhu programu. Jména a datové typy systémových polí jsou obsažena ve struktuře SYST. K jejich adresaci v reportech se používá syntaxe SY-<název pole>. Systémových polí existuje cca 150, já zde uvádím pouze ty, se kterými pracuji v reportu ZCZ_SAMPORD1. Mezi ně patří:

- REPID – obsahuje jméno aktuálně běžícího hlavního program
- LANGU – jazyk přihlášeného uživatele
- DATUM – aktuální datum (datum na aplikačním server)
- UZEIT – aktuální čas na aplikačním server
- UNAME – uživatelské jméno přihlášeného uživatele
- TABIX – index aktuálního řádku v tabulce
- MSGID – třída chybových zpráv
- MSGTY – druh chybové zprávy
- MSGNO – číslo chybové zprávy
- MSGV1-MSGV4 – proměnné zpráv
- UCOMM – hodnota je závislá na události PAI. Existují unikátní funkční kódy, které jsou triggerovány s událostmi PAI
- SUBRC – návratový kód nastavený ABAP příkazem. Pokud je rovno NULE, příkaz byl vykonán úspěšně. Asi nejčastěji používané systémové pole vůbec.

3.3.8 START-OF-SELECTION

Tato událost následuje bezprostředně po výběrové obrazovce. Ukončen je příkazem END-OF-SELECTION. Obsahem sekce jsou dva podprogramy, SELECT_DATA a RUN_ALV. Podprogram v ABAP je volán příkazem PERFORM. Podprogram

SELECT_DATA, jak je patrné ze samotného názvu vybere data na základě zadaných parametrů na vstupní obrazovce.

```
*&-----
-*
*&      Form  SELECT_DATA
*&-----
-*
*      text
*-----
-*
*  -->  p1      text
*  <--  p2      text
*-----
-*
FORM select_data .

DATA: lv_tabix LIKE sy-tabix.
DATA: gt_result_sel LIKE gt_result OCCURS 0 WITH HEADER LINE.

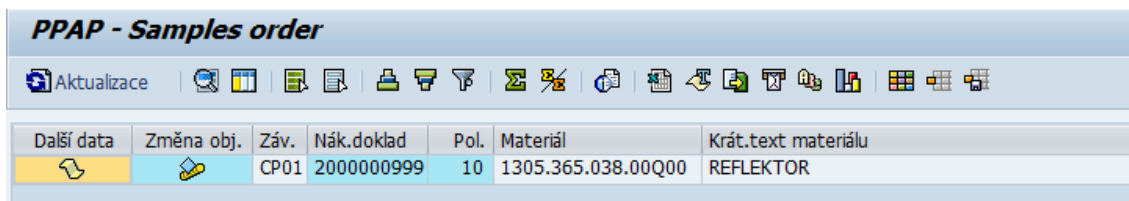
DATA: lt_ekbe LIKE ekbe OCCURS 0 WITH HEADER LINE.
DATA: lt_eket LIKE eket OCCURS 0 WITH HEADER LINE.

SELECT * FROM ekko
        JOIN ekpo
        ON ekpo~ebeln = ekko~ebeln
        INTO CORRESPONDING FIELDS OF TABLE gt_result_sel
        WHERE ekpo~werks = p_werks
              AND ekko~lifnr IN s_lifnr
              AND ekko~ekgrp IN s_ekgrp
              AND ekpo~matnr IN s_matnr
              AND ekko~aedat IN s_aedat
              AND ekko~ebeln IN s_ebeln
              AND ekpo~knttp = space.

...
```

Vybírají se data z joinu tabulek EKKO (hlavičky nákupních dokladů) a EKPO (položky nákupních dokladů). Tabulky jsou spojeny přes pole EBELN, tedy číslo nákupního dokladu, které je součástí klíče v obou tabulkách. Pokud je hledání úspěšné, naplní se interní tabulka GT_RESULT_SEL. Tato interní tabulka je definovaná jako kopie GT_RESULT v části deklarace podprogramu. Příkazem LOOP AT se celá tabulka prochází a dohledávají se další data, tentokrát podle klíče čísla objednávky + položky, v zákaznické tabulce ZCZ_SMPRD. Po celé této proceduře se zkopírují data zpět do tabulky GT_RESULT. V poslední fázi výběru dat se interní tabulka doplní ještě o další data. Začne se doplněním ikon, poté se spojením polí EBELN a EBELP vytvoří číslo objektu, REQNR, v tabulce dodavatelů LFA1 se doplní název dodavatele, z tabulky MAKT se doplní krátký text materiálu v příslušném jazyce. Dál se dohledají z tabulek EKBE a EKET termíny dodávky. Tím je interní tabulka GT_RESULT kompletně naplněna všemi potřebnými daty, která se v dalším podprogramu zobrazí.

RUN_ALV je podprogram, který má za úkol vybraná data zobrazit. Samotný podprogram je vzápětí rozdělen na dvě části, BUILD_FIELDS a LIST VIEVER. V části BUILD_FIELDS se volá funkce REUSE_ALV_FIELDATALOG_MERGE, která vytvoří seznam polí pro výstupní sestavu z interní tabulky GT_RESULT. Jsou zde ošetřeny případy názvů jednotlivých sloupců, ale pouze v některých případech, hlavně když nejsou na první pohled vypovídající, např. ICON1 nebo ICON2. U nich se název v katalogu přepíše.



Další data	Změna obj.	Záv.	Nák.doklad	Pol.	Materiál	Krát.text materiálu
		CP01	2000000999	10	1305.365.038.00Q00	REFLEKTOR

Obrázek 16 : Přepsané názvy sloupců v podpogramu BUILD_FIELDS

LIST_VIEVER je podprogram, který volá funkci REUSE_ALV_GRID_DISPLAY pro vytvoření tabulkového výstupu sestavy. Více o fungování ALV vieweru bylo popsáno v předchozí části práce.

3.3.9 Ikony Další data a Změna Objednávky

V tomto okamžiku jsou vybraná data, která uživatel zadal na výběrové obrazovce, zobrazena. V další části reportu je potřeba doplnit ke každé objednávce a její položce další data, tím se myslí především zadat data pro formulář a definovat seznam požadovaných dokumentů od dodavatele. Obsluha icon je napsána v podprogramu USER_COMMAND.

```

CASE ucomm.

WHEN '&IC1'.

CASE selfield-sel_tab_field.

WHEN 'GT_RESULT-ICON1'.
    PERFORM call_detail_screen.

WHEN 'GT_RESULT-ICON2'.
    SET PARAMETER ID 'BES' FIELD gt_result-ebeln.
    CALL TRANSACTION 'ME22N' AND SKIP FIRST SCREEN.

ENDCASE.
```

Pokud je zachycen double click na ikoně Další data, je volán podprogram CALL_DETAIL_SCREEN. V případě ikony Změna objednávky se volá standardní transakce pro změnu objednávky, ME22N. Poslední věc, která je v podprogramu ošetřena je údržba záznamů v číselnících, tabulkách ZCZ_SMPRD_Cx. Tato údržba je realizována v podprogramu CALL_TABLEVIEW, který je volán s parametrem název příslušné tabulky. V samotném podprogramu se využívá funkce VIEW_MAINTENANCE_CALL, které se předá parametr název tabulky a která umožní pracovat s jednotlivými záznamy tabulky. Dovolí editaci, přidávání, mazání záznamů.

3.3.10 Podprogram CALL_DETAIL_SCREEN

Jak již jsem uvedl, je vyvolán poklikáním na ikoně další data. Nejdřív jsou načtena data z tabulky ZCZ_SMPRD s klíčem REQNR. V dalších krocích i data z tabulek ZCZ_SMPRD_C1 a ZCZ_SMPRD_D1. V obou případech se rozlišuje, jestli jsou data již v tabulkách uložena nebo jestli probíhá fáze pořízení. Na závěr jsou v případě, že e-mailové adresy nebyly nalezeny v aktuálním záznamu v tabulce ZCZ_SMPRD, načteny emailové adresy z tabulky (číselníku) ZCZ_SMPRD_C3. Všechny údaje jsou potom zobrazeny v obrazovce 9000.

PPAP - Samples order

Odeslat | Náhled | Portal SQA | Rozhraní

Detail objednávky

Objednávka/Položka: 2000169707 10 Dodavatel: 207439 Clamason Industries Limited
Odesláno: 28.02.2013 13:54:08 Gibbons Industrial Park, Dudley Roa
Kingswinford, West Midlan DY68TG

Materiál

Materiál: 131100003300Q00 WAERMESCHUTZBLECH-W2
Projekt: Číslo změny: JANLD201510 Množství: 5,000 KS
Status: P Datum změny: 12.02.2012 Datum dodání: 25.03.2013
Důvod: Vzorkování po změně výkresu

Předvýběr:

Požadavky kvality - externí

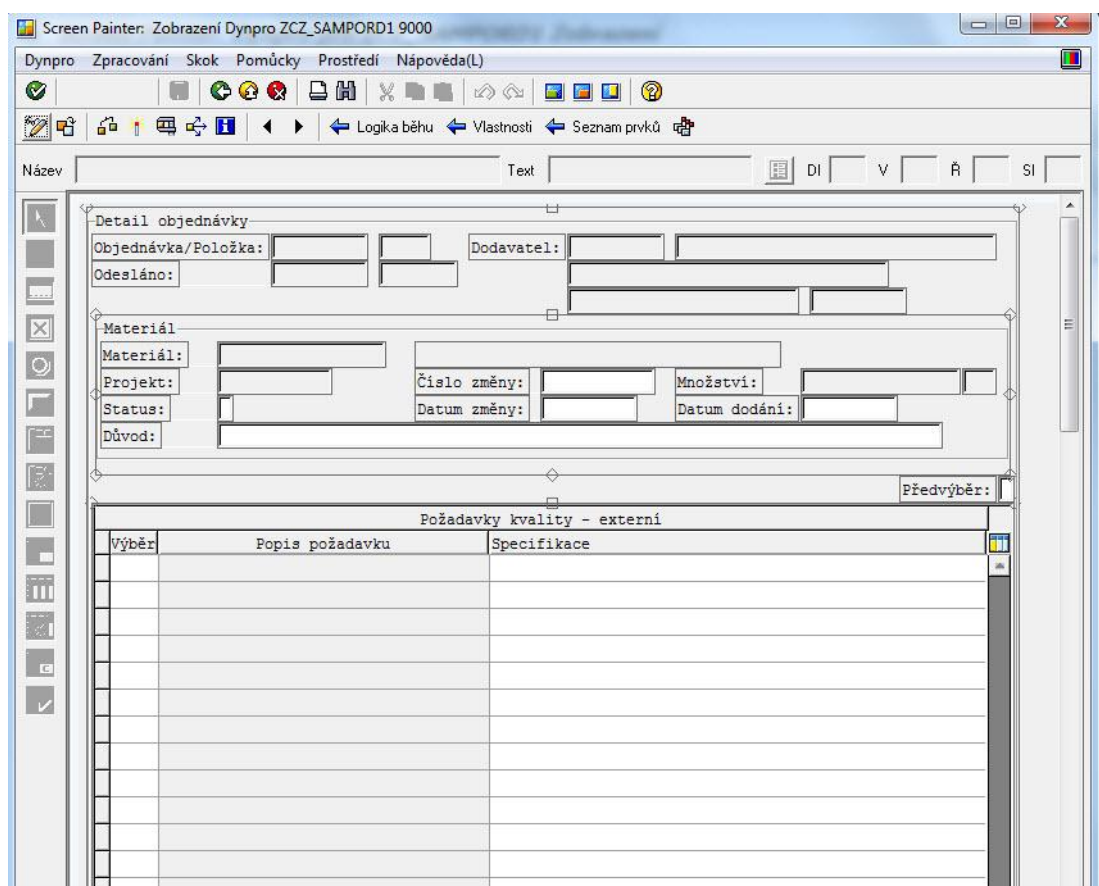
Vý...	Popis požadavku	Specifikace
X	1 - Signed requirements file	
X	2 - PSW cover sheet	
X	3 - Measuring report	
X	4 - Attest	
	5 - Material test result	
	6 - Bill of material	
	7 - Release of components	
	8 - Sub-supplier risk evaluation list	
	9 - IMDS	

Obrázek 17 : Obrazovka 9000

3.3.11 Screen Painter

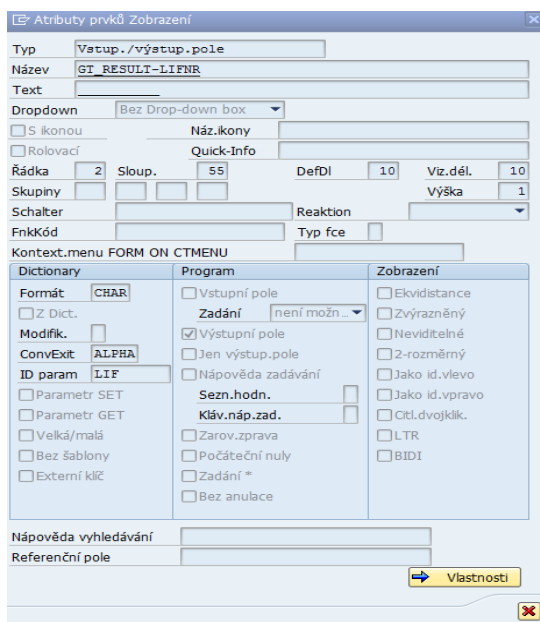
Obrazovky jsou nejčastějším uživatelským dialogem v ABAP reportech. Nedefinují se přímo v programu, ale za použití Screen Painteru. Obrazovka se skládá z vstupně/výstupních polí a logiky běhu. Obrazovka se vyvolá příkazem CALL SCREEN. Logika běhu je rozdělená mezi události PBO (Process Before Output), které jsou provedeny před tím, než je obrazovka zobrazena a události PAI (Process After Input), které se provádějí po uživatelské akci na obrazovce. Obrazovka může obsahovat různé stavební prvky neboli elementy. Ty mají za úkol buď zobrazit obsah polí nebo dovolit uživateli interakci s programem. Mezi nejdůležitější patří:

- textová pole, která nemohou být měněna ani uživatelem ani programem
- vstupní/výstupní pole, umožňují zobrazit data z reportu nebo naopak umožní zadání dat na obrazovce
- rozbalovací seznamy, ty umožní vybrat uživateli jeden záznam z množiny definovaných hodnot
- checkboxy, speciální zaškrťovací pole
- tlačítka/pushbuttons, která po kliknutí spustí PAI událost
- table controls, tabulka obsahující vstupní a výstupní pole



Obrázek 18 : Obrazovka 9000 ve Screen Painteru

Po definici logiky běhu se nadefinuje layout celé obrazovky, každému elementu se definuje celá spousta atributů, jak je vidět na následujícím obrázku:



Obrázek 19 : Atributy prvku obrazovky, pole LIFNR

Běh reportu jsme opustili v okamžiku, kdy byla zobrazena data pomocí obrazovky 9000. Pod titulkem obrazovky jsou opět funkční ikony, které zajišťují interakci s reportem.

PPAP - Samples order			
Detail objednávky			
Objednávka/Položka:	2000000999 10	Dodavatel:	1000 MMT s.r.o.
Odesláno:	25.03.2013 09:56:43	Hranice 469	
		Rychnov u Jablonce nad Ni	46802

Obrázek 20 : Ikony na obrazovce 9000

Odeslat, tato ikona zajistí odeslání vyplněného formuláře z prostředí SAP R/3 na definované e-mailové adresy. Celou proceduru zajišťuje podprogram SEND_MAIL_PDF. V něm se nejprve naplní tisková struktura z interní tabulky GT_RESULT a potom se pomocí funkce SSF_FUNCTION_MODULE_NAME vygeneruje výstupní formulář ZCZ_SQA_PPAP. Formulář se v tomto případě netiskne, ale je poslán ve formátu PDF elektronickou poštou. Ve formuláři se vyskytují jak e-mailové adresy jak kontaktních osob na straně dodavatele, tak i tyto informace o tvůrcích formuláře. Konkrétně se jedná o pracovníka nákupu, který založil objednávku na vzorky a pracovníka kvality, který zpracoval samotný formulář. Adresy jsou přečteny z uživatelského profilu v SAP funkcí EFG_GEN_GET_USER_EMAIL. Jako vstupní parametr se funkci předá uživatelské jméno, funkce vrátí e-mailovou adresu. Funkce SO_NEW_DOCUMENT_ATT_SEND_API1 odešle formulář elektronickou poštou. Této funkci se předávají parametry ve formě tabulek. První z nich je GT_OBJPACK, která obsahuje dokument SapOffice, další GT_OBJBIN, která předává řádku formuláře do SapOffice a poslední GT_RECLIST obsahující seznam příjemců. Funkce, tak jako všechny ostatní umí vrátit i výjimky.

Ikona Náhled spustí proces tisku. V systému SAP R/3 je celý proces výstupu organizován podobně jako třeba v programech Microsoft Office. Tím myslím, že po spuštění celé akce je možné vidět náhled celého výstupu, popřípadě je možné dokument rovnou vytisknout. Vše nejlépe ilustruje obrázek 21, kde je vidět okno před tiskem. V jeho rámci se vybírá tiskárna. Pokud je zvoleno výstupní zařízení LOCL, systém vybere výchozí tiskárnu z operačního systému. Dál je možné měnit název tiskové sestavy, zadat počet kopií, nastavit krycí list a v něm příjemce a oddělení příjemce.

Obrázek 21: Obrazovka před tiskem

V další sekci je možno řídit spool. Spoolový požadavek je dokument, pro který byla vybrána funkce tisku. Jako takové jsou obvykle ukládány na definovanou dobu, aniž by byly vytisknuty. Checkbox Okamžitý výstup zařídí okamžitý tisk na výstupní zařízení. V dolní části obrazovky jsou ikony pro výstup na tiskárnu nebo pro zobrazení na obrazovku. V mém reportu je celý proces výstupu naprogramován v podprogramu PREVIEW_FORM. V něm se nejdřív vybírá tiskárna, poté se volá funkce pro generování smart form formuláře - SSF_FUNCTION_MODULE_NAME. Po úspěšném výstupu se ještě zavolá funkce NAST_PROTOCOL_UPDATE, která updatuje tabulku NAST, ve které systém shromažďuje protokoly zpráv.

Ikona Portal SQA je spíš specialita, než aby měla nějaký hlubší význam. Za ní je ukryt program CALL_URL_SQA, v něm metoda třídy CL_GUI_FRONTEND_SERVICES, EXECUTE. Metodě jsou předány dva parametry, a to L_F_PARM, ve kterém je intranetová stránka oddělení SQA a L_F_APPL, kde se volá internetový prohlížeč, v konkrétním případě Internet Explorer, jako součást standardní instalace v ALCZ.

ID	Typ	Název	Zodpovědná osoba	Číslo dílce	Název	Projekt	Název firmy	Důvod vzorkování	Datum dodání	Datum dodání
24		20120313113540	Klika Jan	130101625000	BLENDE	MB C117	Zlín Precision s.r.o.		2012-03-16	2012-03-30
25		20120313114724	Klika Jan	130101625100	BLENDE	MB C117	Zlín Precision s.r.o.		2012-03-16	2012-03-30
26		20120313140803	Klika Jan	130109218900	ZIERBLENDE	MB C117	Megatech Industries Hlinsko, s.r.o.		2012-03-16	2012-03-30
27		20120313141238	Klika Jan	130109219000	ZIERBLENDE	MB C117	Megatech		2012-03-16	2012-03-30

Obrázek 22 : Intranetová stránka oddělení SQA

Zbývá popsat poslední ikonu, kterou je Rozhraní. Ošetřena je podprogramem INTERFACE_WRITE.

FORM interface_write .

```
* start Excel
CREATE OBJECT h_excel 'EXCEL.APPLICATION'.
SET PROPERTY OF h_excel 'Visible' = 1.

* get list of workbooks, initially empty
CALL METHOD OF h_excel 'Workbooks' = h_mapl.

* add a new workbook
CALL METHOD OF h_mapl 'Add' = h_map.

LOOP AT gt_result.
  h = sy-tabix + 1.
  PERFORM fill_cell USING h 1 0 gt_result-ebeln.
  PERFORM fill_cell USING h 2 0 gt_result-ebelp.
  PERFORM fill_cell USING h 3 0 gt_result-matnr.
  PERFORM fill_cell USING h 4 0 gt_result-menge.
  PERFORM fill_cell USING h 5 0 gt_result-meins.
ENDLOOP.

FREE OBJECT h_excel.

ENDFORM.                " INTERFACE_WRITE
```

V něm se vytvoří objekt H_EXCEL a jsou volány metody H_MAPL a H_MAP. Výsledkem je zavolání aplikace Microsoft Excel, otevření prázdného sešitu a export polí EBELN, EBELP, MATNR, MENGE a MEINS pomocí podprogramu FILL_CELL.

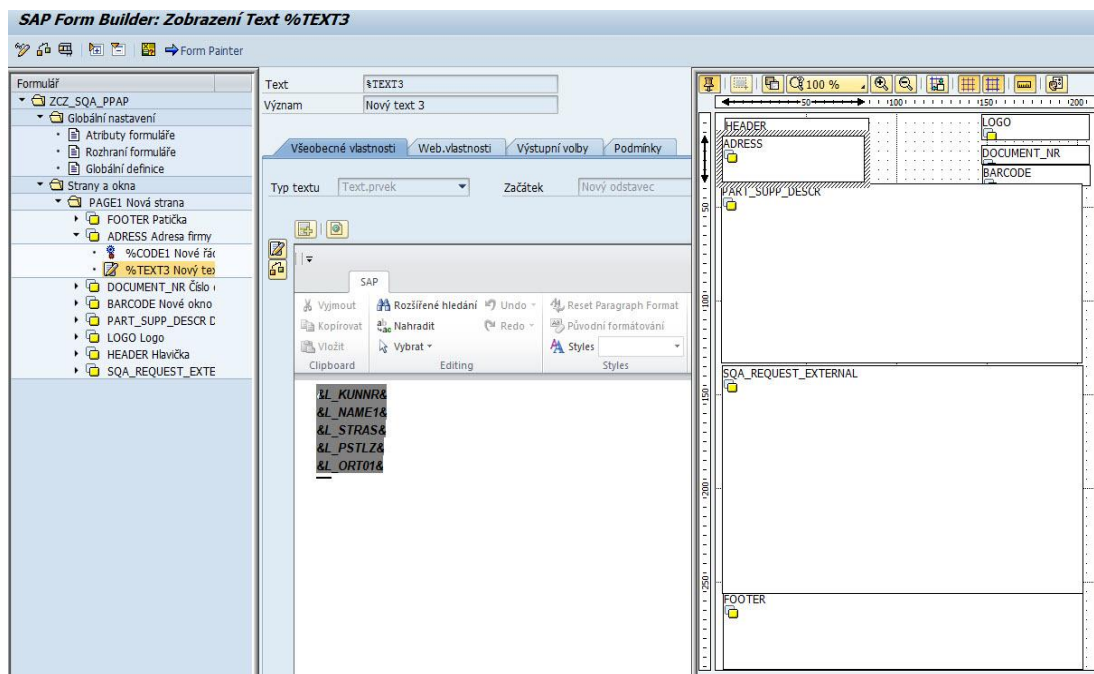
3.3.12 Smartforms

SAP R/3 rozeznává dva typy výstupních formulářů. V prvních verzích byl k dispozici FORM PAINTER, který se spouští transakcí SE71. Od verze 4.6C byl implementován typ nový, Smart Form. Vývojové prostředí se spouští transakcí SMARTFORMS. Mezi hlavní přínosy nového nástroje patří velké rozšíření grafických nástrojů pro tvorbu layoutu a logiky běhu, tak že tvůrce nemusí být znalý programování (uvádí se 90%

všech úprav). Další novinkou je možnost používat naskenovanou grafiku na pozadí, barevně rozlišovat texty. Další vylepšení:

- user-friendly integrovaný Form Painter pro grafickou úpravu formuláře
- použití Smart Styles (obsahují formáty odstavců, definici použitých fontů, znakové formáty). Každý Smart Form má v globálním nastavení, atributech formuláře přiřazený styl
- datový interface pro XML formát
- překlad formuláře je dostupný přes standardní SAP R/3 nástroj
- HTML výstup formuláře
- interaktivní webové formuláře se vstupními poli, push buttons, radio buttons a dalšími

Smart Form integruje do jedné části formulář a tiskový program. Existuje pouze tiskový program, který pomocí funkčních modulů zde již uváděných generuje další funkční modul pro definovaný Smart Form.



Obrázek 23 : Vývojové prostředí Smart Form

Na obrázku je patrné, jak je celé prostředí organizované. V levé části je stromová struktura formuláře, jsou zde vidět jednotlivé objekty formuláře. Detail právě vybraného objektu je v prostřední části obrazovky, kde jsou vidět jednotlivé pole okna. A konečně v pravé části je vidět celkové uspořádání formuláře, tedy jakési preview.

3.3.13 Další podprogramy v reportu ZCZ_SAMPORD1

Zamykání záznamů v tabulkách je velice důležitá věc, protože se pohybujeme v multiuživatelském prostředí, kde s informačním systémem pracuje průměrně dvě stě uživatelů. Z tohoto důvodu je potřeba ošetřit i zamykání záznamů, se kterými se právě pracuje. V reportu je zamykání ošetřeno v podprogramech LOCK_DOCUMENT a UNLOCK_DOCUMENT, v něm je následné odemčení. Zamykají se záznamy v tabulkách nákupních dokladů, to znamená jeden v tabulce EKKO a příslušný počet záznamů v tabulce EKPO. Používá se standardní funkce ENQUEUE_EMEKKOE, které se jako parametry předají číslo objednávky z interní tabulky GT_RESULT a módy práce s tabulkami, zde E (error). Při pokusu měnit zamčený záznam jiným uživatelem se vyvolá chybová hláška a uživatel může inkriminovaný záznam pouze prohlížet.

Podprogram SAVE_DATA zabezpečuje uložení všech pořízených informací do tabulek. Rozlišuje se, jestli se jedná o nově pořízená data, potom se ukládá pomocí příkazu UPDATE nebo jestli se jedná o modifikaci dat už načtených z tabulek. V tomto případě se použije příkaz UPDATE. V další fázi se voláním podprogramu SAVE_DETAILS ukládají další data, konkrétně seznamy interních a externích dokumentů.

Předvolby se do reportu zabudovaly až v době testování. Zjistilo se, že se opakují čtyři nejběžnější kombinace požadovaných dokumentů, které se požadují po dodavatelích. Proto se doprogramovala funkcionální výběr těchto nejčastějších variant výběru. Jako základ zde funguje tabulka ZCZ_SMPRD_C1, ve které představují jednotlivé záznamy seznam všech dokumentů, interních a externích. Ve struktuře tabulky je pět polí, které představují možnost nadefinovat pět různých variant výběru dokumentů.

3.4 SE63 Překlad v SAP R/3

V závěru mé práce bych chtěl napsat ještě několik řádek o překladech do jiných, než originálních jazyků, což má v mezinárodní firmě své opodstatnění. Centrální vývoj

produkuje své reporty v angličtině, v lepším případě je přeloží do italštiny nebo němčiny. Překládat lze veškeré objekty, které mají pole jazykový klíč. V případě customizingu (přizpůsobení, nastavení systému) je to tabulka, které název končí obvykle na písmeno T a která je přes klíč svázána s tabulkou rodičovskou.

Data Browser: Tabulka T161 1 nal.objektů

Tabulka: T161
Zobrazená pole: 33 Od 43 Stálé vedoucí sloupce: Šifra sezn. 0250

	MANDT	BSTYP	BSART	BSAKZ	PINCR	NUMKI	NUMKE	BREFN	REFBA	ABVOR	STAFO	UPINC	STAKO	PARGR	NUMKA	HITYP	LPHIS	GSFRG
<input type="checkbox"/>	010	B	ZLPA	R	00010	01		ZNBB				00000						X

Data Browser: Tabulka T161T 8 nal.objektů

Tabulka: T161T
Zobrazená pole: 5 Od 5 Stálé vedoucí sloupce:

	MANDT	SPRAS	BSART	BSTYP	BATXT
<input type="checkbox"/>	010	C	ZLPA	B	ZLPA Plán.dod. AL
<input type="checkbox"/>	010	D	ZLPA	B	Bestellanford. ZLPA
<input type="checkbox"/>	010	E	ZLPA	B	SA req AL-ZLPA
<input type="checkbox"/>	010	F	ZLPA	B	DA pur com. ouvert
<input type="checkbox"/>	010	H	ZLPA	B	SA req AL-ZLPA
<input type="checkbox"/>	010	I	ZLPA	B	AL RdA ordine aperto
<input type="checkbox"/>	010	P	ZLPA	B	Req.contrato básico
<input type="checkbox"/>	010	S	ZLPA	B	Solicitud de pedido

Obrázek 24 : Tabulka T161, druhy nákupních dokladů a jazykový záznam pro 8 jazyků v tabulce T161T

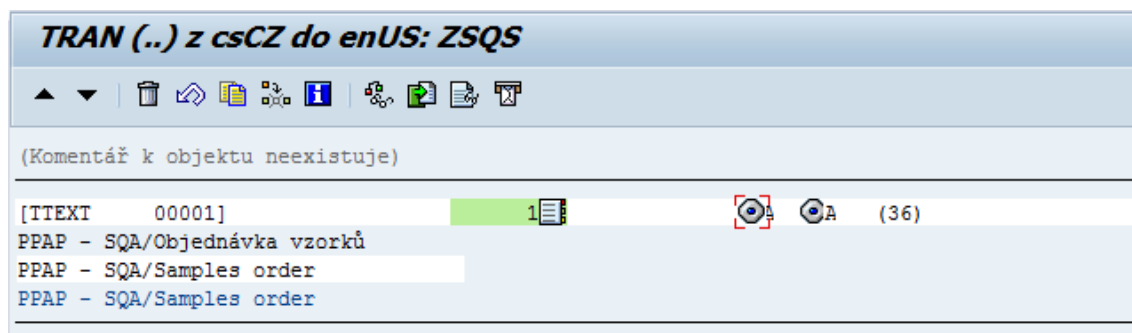
Samotný překlad se potom uskuteční v transakci SE63. Povídání o této transakci, resp. o překladatelském prostředí by bylo opět na samostatnou bakalářskou práci. Umožňuje různé režimy práce. Základní je ten, kde se přímo vyberou objekty k překladu a interaktivně se přeloží. Pokud na systému pracují vyhrazení překladatelé, dá se zvolit režim, při kterém každý překladatel má přiřazené objekty ve svém worklistu. Objekty se obvykle rozdělují podle modulů nebo podle vývojových tříd. Objekty samotné se dělí na :

- **objekty ABAP**

- **krátké texty** – jsou textové fragmenty, které se objevují v uživatelském rozhraní

- **dlouhé texty** – jsou dokumentace pořizená v transakci SE61 (dokumentace k reportům a zprávám)
- **další dlouhé texty** – sem patří SAPscript formuláře a Smartformy
- **speciální texty XSS**
- **transportní objekty**
- **objekty NON ABAP**
 - **texty rozhraní (fragmenty)** – jsou krátké, nezávislé texty obsahující pár slov, např. tlačítka nebo texty sloupců
 - **texty rozhraní (záznamy)** – tyto věty náleží k formulářům, např. krátká sada instrukcí
 - **objekty s vlastním formátováním** – sem patří souvislé texty, kde je ke každé řádce informace o jejím formátu

Překladatel musí vědět, jaký objekt patří do které kategorie. Musí být přihlášen v originálním jazyce, to znamená v jazyce přihlášení vývojáře. Překlad do cizích jazyků musí probíhat postupně, tím myslím, že nelze vybrat víc jazyků, do kterých se bude překládat, najednou. Potom se jednotlivé záznamy zobrazí na dvou řádcích. Na prvním je text v originálním jazyce, na druhém je prostor pro samotný překlad. Jako příklad zde mohu uvést překlad názvu transakce ZSQS do angličtiny. Překlad má v této chvíli status



Obrázek 25 : Překlad názvu transakce ZSQS do EN se statusem A určeno k distribuci

Modifikováno/Rozhodnutí nutné. Posledním krokem před uložením je nastavení úrovně kvality, od X, které znamená, že se překlad nedoporučuje k distribuci, přes úrovně B

(založil nový překladatel), S (standardní úroveň překladu) až po A (označení pro distribuci), takže nejkvalitnější překlad. Pro samotnou distribuci mezi systémy se používá i speciální typ transportních požadavků.

4 Závěr

Cílem mé bakalářské práce bylo naprogramovat zákaznickou transakci pro firemní informační systém. Fáze vývoje se zde prolínala s fází testování, kdy upravená verze je vždy transportována do testovacího systému k ověřovacím testům. Úpravy se víceméně týkaly designu, ať už obrazovky nebo formuláře. Doprogramované byly některé možnosti, jako je možnost vyvolat uloženou variantu seznamu dokumentů. V současné době se již celá transakce používá v produktivním systému SAP R/3 P55. Lidé, kteří se účastnili aktivně testování, předali své poznatky dál na svá oddělení a proškolili personál v používání transakce, i když nejsou potřeba nějaké zvláštní znalosti.

Během vývoje jsem nenarazil na nějaké větší problémy. Snad ta největší brzda byla časová vytíženost některých pracovníků. Občas bylo potřeba počkat na odezvu delší čas.

Největší přínos mé transakce vidím v integraci všech částí celého procesu do podnikového informačního systému. Je podle mě pohodlné, přihlásit se do jednoho systému a být schopen veškeré informace z kterékoli fáze procesu okamžitě vidět. Odpadá tak nutnost pouštět několik dalších aplikací. Tím se omezují případné chyby obsluhy a šetří se i drahocenný čas. Věřím, že časem se nějaké připomínky určitě objeví, že se najdou další nápady, jak celý report vylepšit.

Díky znalostem získaných během studia na Vysoké škole polytechnické jsem si začal uvědomovat, že i v ABAP je trendem jít cestou k objektovému programování.

5 Seznam použité literatury

- [1] MAASSEN, A., GADATSCH, A., FRICK, D., SCHONEN, M.: *SAP R/3*. Computer Press. Praha 2007. ISBN: 80-251-1750-2
- [2] KÜHNHAUSER, K.,H.: *ABAP*. Computer Press. Praha 2009. ISBN: 978-80-251-2117-7
- [3] THÜMMEL, H., HAGEN, W.: *Official ABAP Programming Guidelines*. Galileo.Martin 2009. ISBN 978-15-922-9290-5
- [4] KELLER, H., KRUGER, S.: *ABAP Objects: Introduction to Programming SAP Application*. Galileo Press. Bonn 2002. ISBN 978-02-017-5080-5
- [5] KÜHNHAUSER, K.,H., FRANZ, T.: *Discover ABAP*. Galileo Press. Bonn 2011. ISBN 978-15-922-9402-2
- [6] SAP Help Portal [online]. [cit. 2013-04-20]. *BC – ABAP Dictionary*.

Dostupné z: <http://help.sap.com/printdocu/core/Print46c/en/data/pdf/BCDWBDIC/BCDWBDIC.pdf>

- [7] SAP Help Portal [online]. [cit. 2013-04-20]. *BC – ABAP Programming*.

Dostupné z: <http://help.sap.com/printdocu/core/Print46c/en/data/pdf/BCABA/BCABA.pdf>

- [8] Wikipedia [online]. [cit. 2013-04-12]. *History of SAP R/3*.

Dostupné z: http://en.wikipedia.org/wiki/SAP_R/3#History_of_SAP_R.2F3

6 Seznam obrázků

Obrázek 1: Moduly SAP R/3	11
Obrázek 2: Struktura dialogového programu.....	13
Obrázek 3: Zobrazení definice tabulky (SE11)	16
Obrázek 4 : Přehled vývojářů na SAP portále	17
Obrázek 5 : Detail transportního požadavku	18
Obrázek 6 : Přehled otevřených bodů.....	21
Obrázek 7: Úvodní obrazovka reportu.....	25
Obrázek 8: Parametr BAPI funkce	27
Obrázek 9 : Znak uskladnění WQ1 v kmenových datech materiálu	28
Obrázek 10 : Systémové menu SAP R/3 a příkazový řádek.....	29
Obrázek 11 : Úvodní obrazovka transakce ZSQS	30
Obrázek 12 : Druhy interních tabulek v SAPu	32
Obrázek 13 : Zobrazení struktury zákaznické tabulky	34
Obrázek 14 : ALV sestava v transakci ZSQS.....	37
Obrázek 15 : Ikona Rozhraní – spuštění Excelu a export dat.....	38
Obrázek 16 : Přepsané názvy sloupců v podpogramu BUILD_FIELDS	42
Obrázek 17 : Obrazovka 9000	43
Obrázek 18 : Obrazovka 9000 ve Screen Painteru	45
Obrázek 19 : Atributy prvku obrazovky, pole LIFNR.....	45
Obrázek 20 : Ikony na obrazovce 9000	46
Obrázek 21: Obrazovka před tiskem	47
Obrázek 22 : Intranetová stránka oddělení SQA	48
Obrázek 23 : Vývojové prostředí Smart Form.....	49
Obrázek 24 : Tabulka T161, druhy nákupních dokladů a jazykový záznam pro 8 jazyků v tabulce T161T	51
Obrázek 25 : Překlad názvu transakce ZSQS do EN se statusem A určeno k distribuci	52

7 Seznam použitých zkratk

SAP R/3 – Systém, Applications and Products, Real Time Data Processing, three - tier

ABAP – Advanced Business Application Programming

FI - Financial

CO - Controlling

ERP – Enterprise Resource Planning

8 Přílohy

8.1 Obsah přiloženého CD

Na přiloženém CD se v kořenovém adresáři nachází tato bakalářská práce ve formátu *bakalarska_prace.pdf* a zdrojové texty obou reportů v textovém souboru.