

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Aplikace pro report seznamu změn v kmenových datech

Bakalářská práce

Vedoucí práce:
Ing. Jan Turčínek, Ph.D.

Martin Pavelka

Brno 2016

Chtěl bych poděkovat vedoucímu práce Ing. Janu Turčínkovi, Ph.D. za ochotu a vlídný přístup, rodině za podporu a Ing. Alanu Mičánkovi za umožnění zpracování práce a věnovaný čas.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Aplikace pro report seznamu změn v kmenových datech** vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 19. května 2016

.....

Abstract

Pavelka, Applications to export a list of changes in the master data

The work is focused on creating an application that allows creating a list of changes in the master data, an overview of the architecture of information system and SAP in particular, in which environment the application was created. Creating an application is described by UML diagrams that describe how it works, then the work is dedicated its very creation, application deployment to a production system and describe how the employees can work with the application.

Key words: SAP, Information systems, ABAP, Master data

Abstrakt

Pavelka, Aplikace pro report seznamu změn v kmenových datech

Práce je zaměřena na tvorbu aplikace umožňující tvorbu seznamu změn v kmenových datech, přehledem architektury informačního systému a zejména SAPu, pro jehož prostředí byla aplikace vytvořena. Tvorba aplikace je popsána UML diagramy, které popisují jak funguje, poté se práce věnuje její samotné tvorbě, nasazení aplikace do produkčního systému a popsání jak s ní zaměstnanci mohou pracovat.

Klíčová slova: SAP, Informační systémy, ABAP, Kmenová data

Obsah

1	Úvod	11
2	Cíl práce	12
3	Aktuální stav řešené problematiky	13
3.1	informace	13
3.2	Informační systémy	13
	Interní a veřejné systémy	13
	Okolí systému	13
	Uživatel a jeho vztah k informačnímu systému	14
	Veřejné informační systémy	14
	Interní informační systémy	15
	Architektura interního informačního systému	15
3.3	ERP systémy	18
	Obecně	18
	Vývoj	18
3.4	Architektura SAPu	19
	Části SAP ERP	19
	SAP ERP a SAP Business Suite	21
	Technická architektura SAP ERP	22
	Třístupňová architektura klient server	24
3.5	ABAP	25
	Druhy programů	25
	Tvorba programů	26
3.6	Kmenová data	26
	CDHDR	26
	CDPOS	28
3.7	UML	30
	Diagramy případů užití	30
	Stavové diagramy	31
	Diagramy sekvencí	31
	Diagramy spolupráce	31
	Diagramy činností	31
3.8	ERD	31
3.9	Tří systémová architektura	32
	Vývojový systém	32
	Testovací systém	32
	Produktivní systém	32

4	Metodika vypracování	34
4.1	Navrhování aplikace	34
4.2	Implementace kódu	34
4.3	Zavádění aplikace	34
5	Tvorba aplikace	36
5.1	Požadavky na aplikaci	36
5.2	Návrh aplikace	37
	Diagram užití	37
	Stavový diagram	38
	Sekvenční diagram	38
	Diagram spolupráce	38
	Diagram činnosti	39
	ERD aplikace	39
5.3	Vstupní okno aplikace	40
5.4	Tělo aplikace	42
5.5	Tvorba výstupu	44
5.6	Nasazení vytvořené aplikace	46
6	Diskuse	47
7	Závěr	48
8	Reference	49
9	Manuál pro práci s aplikací A	51

1 Úvod

V roce 2009 došlo k fúzi pivovaru Starobrnno a Královského pivovaru Krušovice, které oba patřili pod třetí největší pivovarní společnost Heineken. Nicméně značka pivovaru Starobrnno stále přetrvává a pivovar pokračuje ve své produkci. K řízení této pobočky je použita jedna z nejnovějších verzí SAP, SAP ECC 6 EnhP 5 běžící na SAP Netweaver 7.01(ABAP Stack).(Heineken Česká republika reports, 2015)

Informační systémy SAP jsou používány pro malé, střední i velké firmy a to díky mnoha verzím, které jsou uzpůsobeny, aby vyhovovaly různým velikostem podniků.(Heineken Česká republika reports, 2015)

Informační systém je plně nastavitelný, takže trvá značnou dobu jeho zavedení do firmy a také je třeba jej upravovat podle momentálních požadavků firmy. Z toho důvodu, je velice důležitá schopnost uchovávat změny, které byly na systému provedeny.(Heineken Česká republika reports, 2015)

Například změny prováděné na kmenových datech jsou zaznamenávány do tabulek CDHDR(hlavičky položek změn) a CDPOD(informace o změně). V systému, ale automaticky není aplikace, která by umožnila rychlý přehled těchto změn, kde by byly uvedeny informace, které jsou nejčastěji vyhledávány a používány.(Heineken Česká republika reports, 2015)

Tato bakalářská práce se bude zabývat vytvořením co nejefektivnější aplikace schopné tyto data jednoduše zpřístupnit a zobrazit skutečně hlavně to, co se ve firmě reálně vyhledává. Tato potřeba vznikla zejména kvůli požadavku na snižování počtu pracovníků na oddělení kmenových dat, které připadlo pod správu oddělení IT. To vede k potřebě detailně sledovat realizované změny, jak, proč a jak často byly provedeny.

Aplikace byla tvořena na IT oddělení podniku, kde byl dán k dispozici potřebný hardware a umožněny případné konzultace. Tvorba aplikace probíhala ve vývojové části systému, kde aplikace pracuje s aktualizovanými daty, ale do chodu produktivního systému nijak nezasahuje. Zde probíhal celý vývoj aplikace i s případnými úpravami. Až po konečných úpravách ve vývojovém systému je možno přesunout aplikaci do testovacího systému, kde je uživateli aplikace testována a až poté umístěna v produktivním systému pro reálné používání.

2 Cíl práce

Cílem práce je zanalyzovat prostředí pro tvorbu aplikace, kompatibilitu systému s různými programovacími jazyky, které by mohly být pro tvorbu aplikace použity, a také architekturu informačního systému.

Dále vybrat nejvhodnější způsob návrhu aplikace, který důkladně popíše její chod, možnosti jejího využití, s jakými daty aplikace pracuje a odkud je získává.

Dalším krokem bude tvorba návrhu a výběr nástrojů, které budou použity. Následovat bude samotná implementace kódu, která bude dodržovat standartní postupy firmy při implementování aplikací. Aplikace bude splňovat požadavky dané IT vedením podniku. Po vytvoření aplikace proběhne její kontrola a zavádění do systému. V příloze práce bude také manuál k používání.

3 Aktuální stav řešené problematiky

3.1 informace

Vlastnosti reálného světa jsou vloženy do počítače ve tvaru znaků jako „data“. Dosazením těchto dat do konceptu informačního systému nebo aplikace způsobem, aby jim uživatelé byli schopni rozumět, vznikají „informace“. (Rábová, 2008)

3.2 Informační systémy

Jednodušeji řečeno je systém určitá definovaná množina prvků a vazeb mezi těmito prvky. Jako systém je vymezená určitá část reality nebo naše představa o této realitě. Jeho rozsah se vymezuje na část fikce a reality, pro kterou je rozhodnuto, že se bude zkoumat nebo řešit. Zkoumat je možné jakoukoliv část systému, i jednotlivé prvky, který by byly považovány za subsystémy. Nahlížení na jednotlivé části systému a jejich bližší zkoumání se nazývá úroveň pohledu. (Doucek, 2005)

„Informační systém je takový systém, který je realizován na okolí pomocí vazeb, jenž jsou realizovány informacemi.“ (Kolektiv autorů, 2005)

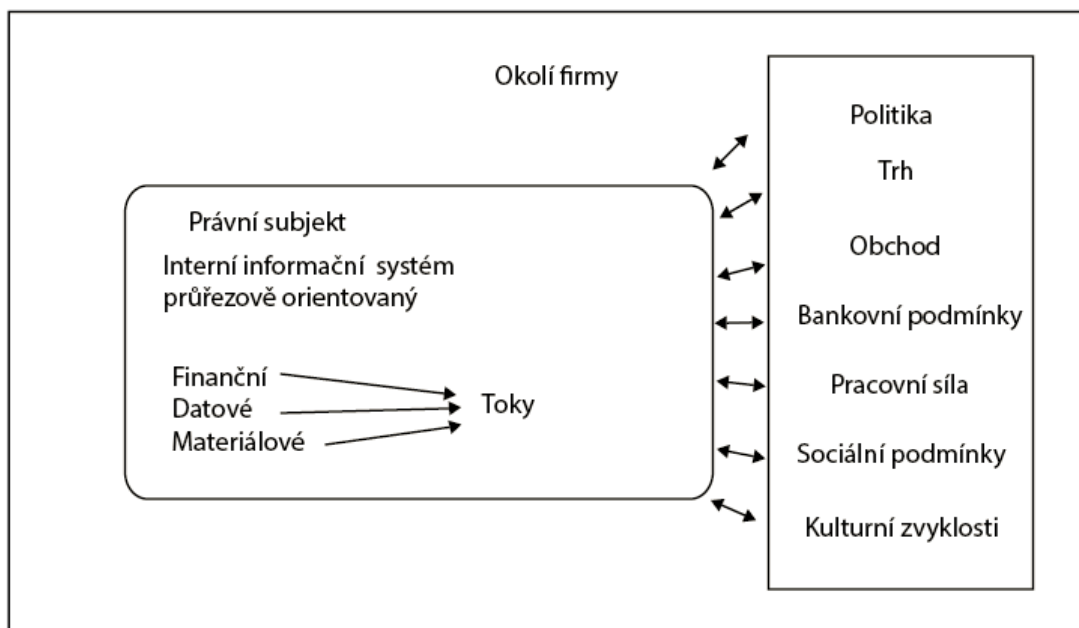
Interní a veřejné systémy

Interní systémy jsou obvykle systémy ve firmách, na úřadech, nebo institucích. Jejich realizace je založena na intranetu. Jejich typickým znakem je tvorba informací pro vlastní potřebu a pro splnění povinností dané zákonem. V okolí těchto interních systémů se nachází zdroje informací. Jedním z těchto zdrojů jsou právě veřejné informační systémy. Veřejné informační systémy fungují na bázi informačních technologií. Jejich typickým znakem je tvorba informací pro potřebu jiných subjektů. (Doucek, 2005)

Okolí systému

Po vymezení systému je zároveň určeno i jeho okolí, které má každý systém. Okolí systému jsou vazby a prvky se vztahy nebo vazbami ke zkoumanému systému. Toto okolí prostě existuje, ale není do něj z hlediska zkoumaného systému zasahováno. Z tohoto okolí jsou však získávány informace, ale také informace posílány. Okolí interního systému je často tvořeno například jinými informačními systémy v podniku, veřejnými informačními systémy, nebo jiné systémy a prvky. Okolí veřejného informačního systému jsou jiné veřejné informační systémy a uživatelé těchto systémů. Informační systémy se svým okolím široce komunikují a jejich činnost je okolím zásadně ovlivňována. Důležité pro systém je dobrá znalost vazeb k tomuto okolí a jejich přesné vymezení. (Doucek, 2005)

Na obrázku 1 lze vidět interní informační systém a jeho okolí představující politiku státu ve kterém se firma nachází, trh, na kterém firma je, obchod, bankovní podmínky, pracovní sílu, sociální podmínky a kulturní zvyklosti. (Doucek, 2005)



Obrázek 1: Schéma okolí interního informačního systému. (Doucek, 2005)

Uživatel a jeho vztah k informačnímu systému

Uživatel využívá služeb poskytovaných systémem. Uživatel také může vystupovat jako jeden z prvků systému nebo jako část jeho okolí. Pokud je jedním z jeho prvků, důraz je kladen na zkoumání jeho vazeb s ostatními uživateli. Těmto uživatelům musejí být stanoveny postupy, vydány příkazy a omezení, která se musejí dodržovat a toto dodržování hlídat. V případě uživatele, který je součástí okolí systému, není důležité řešit jeho vazby s ostatními uživateli. Řeší se jen případný zákaz vstupu uživatele k systému a tvorba pravidel pro úkony, které uživatel může využívat. Za uživatele je možné také považovat i jiný systém, pro který platí stejná pravidla jako pro prvek. Ve většině případů je uživatel prvkem v interních systémech a okolím u veřejných systémů. (Doucek, 2005)

Veřejné informační systémy

Cílem veřejných informačních systémů je obvykle podpora pro práci uživatelů, kteří mají zájem o data z určité oblasti. Tyto systémy mají typické charakteristiky, mezi které patří definování práv uživatelů k užitím dat v informačním systému, která patří jiným uživatelům. Dále je možno rozdělovat veřejné informační systémy placené a volné. Přístup může být také povolen jen některým skupinám uživatelů splňující určitá kritéria jako členství v určitých spolcích, registrace, nebo příslušnost k určitým resortům. Data v těchto systémech bývají problémově orientována. Správa veřejného informačního systému, jeho provoz a další rozvoj jsou zajišťovány firmou nebo organizací, která systém vlastní, ale je také možné tyto služby nechat na jiných firmách formou outsourcingu. Za bezpečnost dat

systému odpovídá jeho provozovatel. Veřejné informační systémy lze také rozdělit na privátní, které jsou vlastněny soukromou firmou a systémy provozované státní správou nebo jinými společenskými institucemi.(Doucek, 2005)

Interní informační systémy

Obecně jsou tyto systémy vytvořeny pro podporu činnosti nějaké právnické osoby. Slouží také pro podporu menších a malých firem právnických subjektů působících v ekonomice jako jsou malé firmy nebo fyzické osoby. Tyto systémy jsou souborem činností pro sběr, přenos, uchovávání, zpracovávání a distribuci a prezentaci dat v organizaci, pro potřeby rozhodování tak, aby řídicí pracovníci a manažeři dosáhli vysoké účinnosti při vykonávání svých řídicích funkcí. Mezi typické rysy interních informačních systémů patří fakt, že uživatel zde vystupuje na rozdíl od veřejného informačního systému jako prvek interního informačního systému. Architektura informačního systému je tvořena podle potřeb firmy a její strategií. Data jsou průřezově orientována a je z nich možné získat přehled o firmě jako celku. Data jsou také v těchto systémech uložena a obhospodařována a jejich vznik je jedním z produktů podniku. Uživatelé jsou rozděleni do skupin, kterým jsou přidělena práva odpovídající potřebám uživatele k výkonu jeho práce. Bezpečnost dat je odpovědností vlastní firmy společně se správou, provozem a dalším rozvojem informačního systému.(Doucek, 2005)

Architektura interního informačního systému

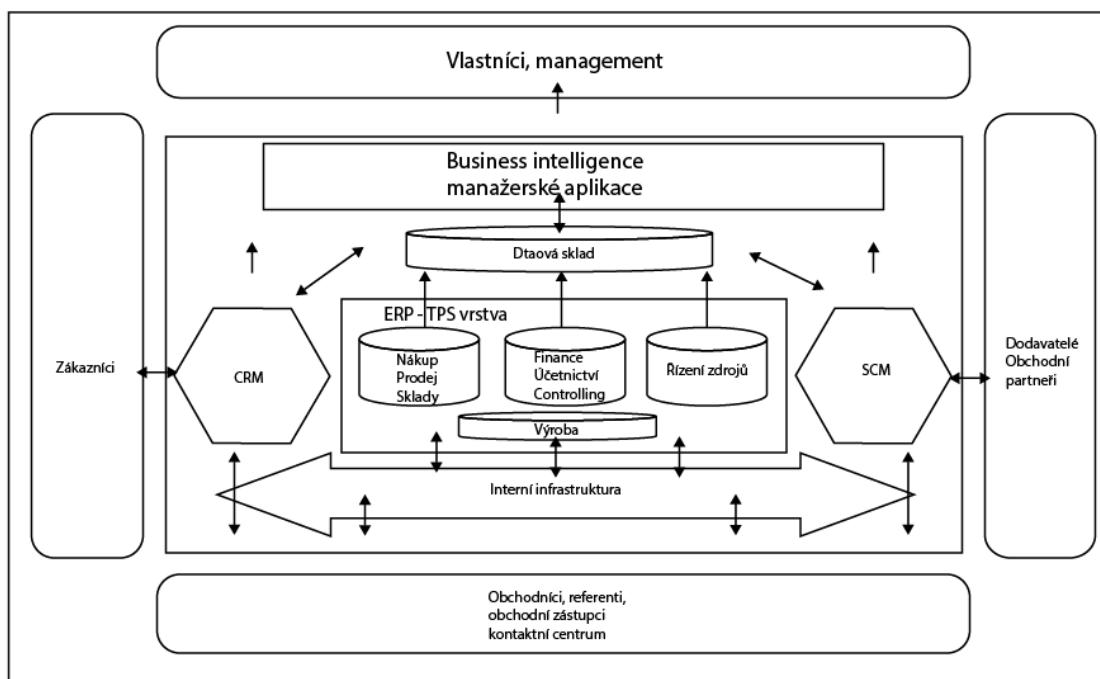
Základem pro realizaci podnikatelské strategie je přesná představa co chce firma pomocí informačního systému dosáhnout. Pro získání této představy je důležité důkladně zpracovat celkovou architekturu systému. Celková architektura může být definována bloky nebo moduly architektury, vazbami mezi nimi a vazbami na okolí, díky kterým je systém na toto okolí napojený. Pro systém je možné rozlišovat více různých bloků, z nichž každý je určen několika dimenzemi, které jsou níže uvedeny:(Doucek, 2005)

- Datovou – popis datových prvků a jejich vazeb
- Funkční – popis funkcí a složek
- Organizační – zabezpečení, které služby a funkce jsou komu zpřístupněny
- Technickou – specifikace technických prostředků a informačních technologií použitých při tvorbě informačního systému
- Programovou – základní programové vybavení, aplikační programové vybavení
- Personální – personální nároky
- Metodickou – ekonomickou - předpokládané přínosy a náklady z použití a tvorby informačního systému

Po zpracování této architektury je možné její využití pro formu rozpracování informační strategie v návaznosti na strategii podnikatelskou. Architektura může sloužit k zachycení aktuálního stavu informačních technologií v podniku nebo je možné ji využít jako znázornění vize žádoucích změn v podniku. Základními komponentami interního informačního systému jsou:(Doucek, 2005)

- Programové vybavení základní a aplikační
- Technické vybavení – počítače, servery, aktivní a pasivní prvky sítí atd.
- Uživatelé
- Organizační opatření a směrnice, kterými se provoz informačního systému řídí

Na architekturu lze nahlížet více způsoby podle toho, kterou část chceme sledovat. Architektura úloh popisuje funkčnost informačního systému, jeho programové vybavení a typy úloh probíhající ve firmě. Architektura datová popisuje datovou základnu v organizaci, s kterými data aplikace pracují, jejich vzájemné vazby a kdo nese za která data odpovědnost. Architektura technologická popisuje technické prostředky, kde jsou umístěny, kdo za ně nese odpovědnost a jejich technické parametry.(Doucek, 2005)



Obrázek 2: Znázornění podnikové architektury úloh.(Doucek, 2005)

Na obrázku 2 je znázorněna podniková architektura úloh. Uprostřed stojí ERP – TPS vrstva obsahující části architektury řešící nákup, prodej, sklady, finance účetnictví, controlling, řízení zdrojů a výroba. K této části přistupují CRM (řízení vztahů se zákazníky), SCM (Řízení dodavatelských řetězců), část interní infrastruktury a data tvořená touto částí

jsou zasílána do datového skladu, odkud k datům přistupují manažerské aplikace používané vlastním managementem organizace. Po stranách k architektuře přistupují zákazníci skrz CRM a dodavatelé a obchodní partneři přes SCM. V neposlední části jsou na architekturu napojeni obchodní referenti, obchodní zástupci a kontaktní centrum komunikující s interní infrastrukturou a s SCM a CRM.(Doucek, 2005)

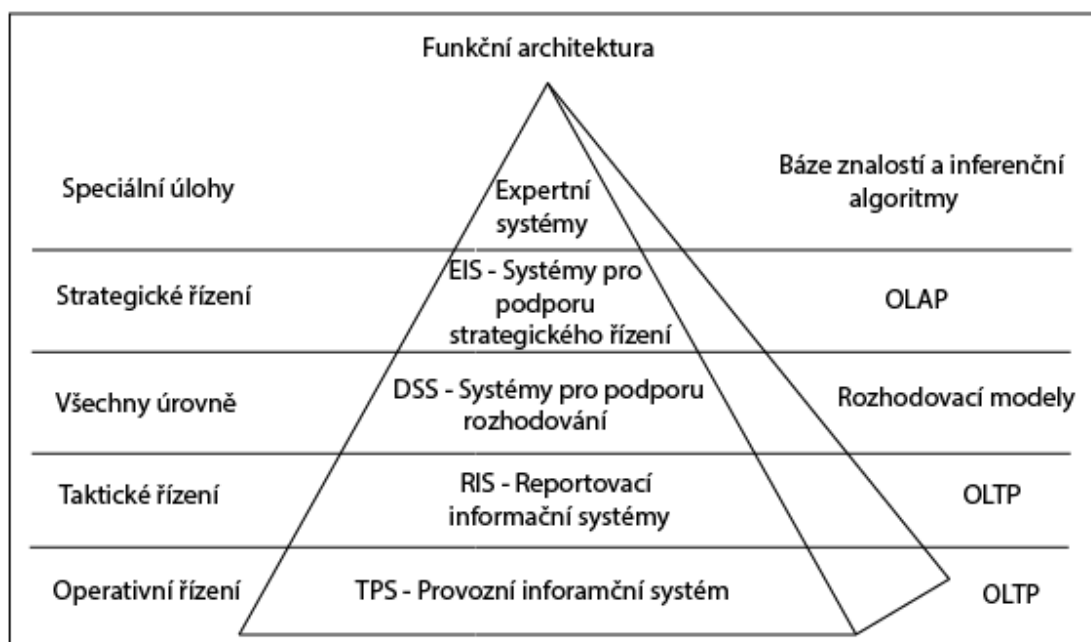
Informační systémy manažerské práce patří do části informačního systému firmy, který pracuje s vybranými nebo upravenými daty, která jsou nositeli komplexních informací charakteristickým pro danou firmu. Manažerské aplikace jsou obvykle chápány jako nástavba interpretující značné objemy transformovaných primárních dat, generovaných základními procesy. Tyto transformace je nutno vykonávat ve specializované části interního informačního systému, která je datově řízena provozním systémem (TPS). Charakter potřeb manažerských aktivit vyžaduje zcela odlišné mechanismy práce s daty, než jakých je možné dosáhnout v transakčních systémech. Základ architektury transakčních systémů je obvykle relační databáze vybudovaná na technologii OLTP (On-Line Technological Processing). Systémy schopné propojit datový potenciál provozního transakčního systému s řídicími mechanismy firmy a analytickými potřebami jednotlivých složek řízení jsou založeny na technologii OLAP (On-Line Analytical Processing). Tyto procesy jsou schopné zajistit značně efektivní a intuitivní mechanismus vícekritériální analýzy dat. Pro prezentaci dat manažerům se využívá prezentační vrstva zajišťující logiku a přehlednost výstupů pro management firem.(Doucek, 2005)

Transakční provozní systém zajišťuje základní fyziologické funkce firmy. Jeho hlavní využití je v interaktivním nebo dávkovém shromažďování základních dat firmy. Zajišťuje přenos a zpracování všech informací ve firmě. Do struktury provozního informačního systému patří především:(Doucek, 2005)

- Základní evidence účetních operací
- Evidence materiálu
- Evidence zboží
- Personální evidence
- Evidence a data o prodejkách a nákupech zboží
- Evidence zákazníků záznamy objednávek služeb a zboží
- Data pro sestavení a výpočet mezd
- Evidence obchodních případů
- Kancelářské systémy pro elektronickou výměnu dat

Na obrázku 3 je možné vidět návaznosti předávání dat mezi technologiemi OLTP a OLAP. OLTP se nacházejí na spodní části pyramidy a jsou využívány k taktickému a operativnímu řízení. OLAP je využívám pro systémy podporující strategické řízení.

Cílem transakčního informačního systému je podpora práce operativního řízení firmy na nejnižší úrovni managementu, pořizování dat, jejich správa a zajištění jejich dostupnos-



Obrázek 3: Znázornění funkční architektury.(Doucek, 2005)

ti oprávněným uživatelům. Funkce jsou pořizování dat, ukládání, jejich archivace, zajišťují provoz interního informačního systému firmy, podporu rutinních operací spojených s výkonem hlavní činnosti firmy.(Doucek, 2005)

3.3 ERP systémy

Obecně

ERP (Enterprise Resource Planing) je v posledních 15 letech velice oblíbený druh aplikací, významně ovlivňující současný podnikový byznys, díky vysokému počtu těchto aplikací a jejich důležitosti. Důkazem toho je obrovská oblíbenost ERP u předních firem působících v Česku a u firem zaměřených na export. Důležitá odlišnost od jiných technologií v podniku spočívá v tom, že se týká celého podniku a ne jen IT oddělení, protože jejich souvislost není pouze ke vztahu k informačním technologiím, ale k celé firmě.(Basl, 2012)

Vývoj

V osmdesátých letech bylo pro firmy charakteristické programování vlastních úloh, přímo na míru daného podniku. Toto řešení se v následujících deseti letech postupně transformovalo do tvorby softwarových aplikací, které byly zaměřeny na myšlenku integrace do jednoho společného řešení na bázi jednotné databázové platformy. Přechod na ERP ale nebyl jednoznačný. Každá firma se mohla rozhodnout pro jednu ze tří možností. (Basl, 2012)

Jednou z možností dalšího vývoje je rozvoj existujících softwarových řešení, který plně využil dosud investované zdroje. Nevýhodou tohoto přístupu se stává nejistota schopnosti tohoto řešení fungovat při budoucích požadavcích firmy.(Basl, 2012)

Další možnost představuje vývoj vlastního informačního systému, vytvořeného přesně podle stávajících požadavků firmy. K udržení tohoto systému je zapotřebí specializované IT oddělení, které musí být udržováno po celou dobu existence informačního systému na vysoké úrovni. Tvorba vlastního informačního systému je navíc velice finančně a časově náročná a obsahuje riziko nízké účinnosti. Třetí možností se stal nákup hotového softwaru, který vyhovoval potřebám podniku. Nevýhodou tohoto řešení je vyšší počáteční investice, ale na druhou stranu je jeho zavedení daleko rychlejší. Velkou výhodou je zejména garantovaná funkčnost a následný další rozvoj. Tato možnost také přináší velkou závislost na externí organizaci, od níž je software zakoupen.(Basl, 2012)

Ve většině podniků začala postupně převládat třetí varianta, a to buďto její plošné zavedení nebo jen částečné, kdy se v jistých částech podniku stále držely vlastní řešení. Z těchto zavedení se postupně vytvarovaly ERP systémy, které v užším slova smyslu zahrnují integraci vnitropodnikových oblastí jako je výroba, logistika finance a lidské zdroje. Postupné rozšiřování ERP zahrnovalo další aplikace jako BI (Business Intelligence), využívané pro manažerské činnosti nebo aplikace pro podporu vazeb mezi ostatními podniky, zejména dodavateli nebo zákazníky. Další důležitou součástí se stávají komponenty pro realizaci elektronického obchodu mezi podniky označované zkratkou B2B (Business to Business) a prodejem zákazníkovi B2C (Business to Customer).(Basl, 2012)

V dnešní době se setkáváme se dvěma typy podnikových systémů, rozdělené podle zaměření na specializaci podniku nebo komplexnosti systému. Tyto jsou označovány All-in-one a Best-of-Breed. All-in-one je zaměřená na komplexní pokrytí všech klíčových procesů podniků s vysokou úrovní integrace, ale nevýhodou v možných nedostacích více specializovaných funkcí. Opakem je Best-of-Breed nabízející aplikace specializované pro dané podniky, ale horší schopnost integrace.(Basl, 2012)

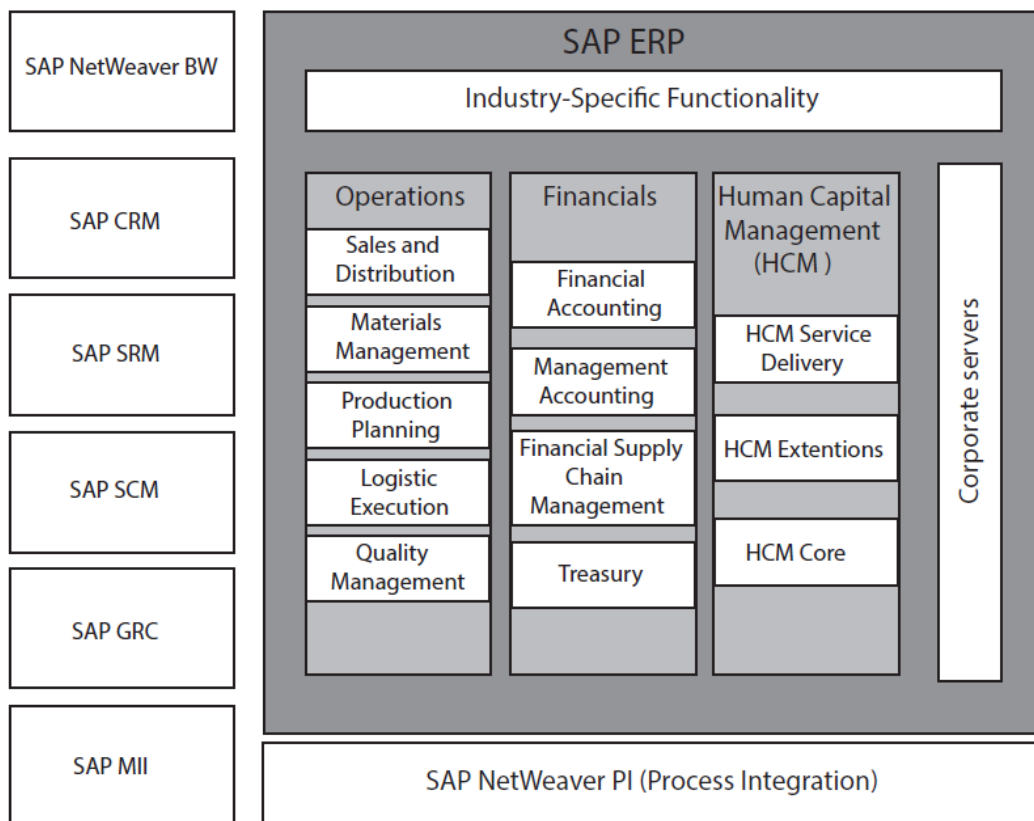
3.4 Architektura SAPu

SAP ERP, původně nazývaný SAP R/3 je software pro plánování podnikových zdrojů. Podporuje všechny obchodní procesy, které jsou potřebné v dnešních společnostech. Zahrnuje SAP ERP operace, SAP ERP Finance, a SAP ERP management pro lidské zdroje.(Boedr, 2014)

Na obrázku 4 jsou vidět všechny tři části ERP systému - Operace, Finance a Řízení lidského kapitálu, které obsahují další komponenty, které budou popsány dále. Vedle ERP části jsou zachyceny také podnikové aplikace pro podporu konkrétních případů.(Boedr, 2014)

Části SAP ERP

SAP ERP Operace je jádrem pro nákup, prodej, skladování a přepravu materiálu. Obsahuje následující funkce:(Boedr, 2014)



Obrázek 4: Přehled architektury SAP ERP.(Boedr, 2014)

- Prodej a distribuce, obsahující tvorbu prodejních příkazů, kontrolu dostupnosti požadovaných produktů, kalkulaci cen, finální doručení a fakturaci.
- Management materiálu, obsahující tvorbu objednávek a servis pro externí obchodníky, potvrzování příchozí fakturace a správu skladů.
- Plánování výroby, obsahující plánování výroby materiálu a její provádění.
- Logistika, obsahující kontrolu a organizování přesunu materiálu.
- Management jakosti, obsahující plánování jakosti produktů, inspekci, kontrolu jakosti během prodeje, produkci a zadávání veřejných zakázek.

Tyhle funkce jsou integrovány v databázi, což zaručuje spolehlivost podnikových procesů napříč různými SAP ERP operacemi založených na konzistentních datech. Na stejném principu SAP ERP Operace komunikují s SAP ERP Financemi. Všechny komponenty sdílejí stejná kmenová data. Rozsah funkcí SAP ERP Operací může být rozšířen za použití integrování dalších SAP Business Suit aplikací jako je SAP Customer Relationship Management(management vztahu se zákazníky), SAP Supply Chain Management(management

dodavatelského řetězce) a SAP Supplier Relationship Management (management vztahů s dodavateli). (Boedr, 2014)

SAP ERP Finance (FI) jsou jádrem pro finanční účetnictví, controlling, řízení finančního dodavatelského řetězce a funkce pokladen. Skládá se z následujících komponent: (Boedr, 2014)

- Finanční účetnictví, které klasifikuje a sčítá finanční transakce společnosti. K tomuto účelu finanční účetnictví poskytuje účetní knihu, účetní závazky, pohledávky a účetní aktiva. Finanční účetnictví zaznamenává obchodní transakce, jako jsou zákaznické faktury, faktury prodejců nebo platby. Založeno na transakcích balancujících částky v těchto dokumentech, finanční účetnictví periodicky generuje rozvahu stejně jako výkazy zisků a ztrát, které jsou reportovány vládě, investorům a ostatním zainteresovaným stranám.
- Management účetnictví, který je komponentou používanou pro interní účetnictví uvnitř společnosti, sloužící k přípravě dat, jako jsou produkční náklady na prodané zboží, režijní náklady a kalkulace zásob. Tato data jsou používána pro obchodní analýzu společnosti a pro podporu manažerského rozhodování.
- Management řetězce dodávek je komponenta zpracovávající účetní pohledávky a závazky, aby pomohla zajistit hladký průběh finančních prostředků. Také obsahuje online platby pro klienty a obchodníky.
- Pokladna je aplikací ze SAP pomáhající zajistit správný cash flow a likviditu společnosti pro každodenní operace. Fondy, deriváty, cenné papíry, finanční akcie, tržní rizika, půjčky jsou v této aplikaci také spravovány.

SAP ERP Management lidského kapitálu (HCM) je řešení SAPu pro řízení lidí pracujících pro společnost. SAP ERP HCM je částí SAP ERP, ale může být nasazen samostatně, aby byla ochráněna citlivá data. SAP ERP HCM je rozdělen do tří vrstev. Základ tvoří SAP ERP HCM core komponenty, jako organizační management, personální administrativní, mzdy, osobní rozvoj a management času. Poskytovaná data a funkce, která jsou znovu použita SAP ERP HCM nastavbovými komponentami, které podporují přídavné procesy jako podnikové vzdělávání. Další částí je SAP ERP HCM zásilková služba, umožňující uživatelům přistupovat k SAP ERP HCM komponentám skrz různé kanály a média. (Boedr, 2014)

SAP ERP HCM datový model je rozdělen na „infotypy“. Jeden infotyp definuje strukturu dat pro významově podobná data, která jsou uložena dohromady v databázi a jsou také dohromady zobrazována v uživatelském grafickém rozhraní, které je pro HCM procesy velice důležité. HCM procesy jsou také specifické podle zemí, kde se podnik nachází, protože musí dodržovat legálnost, dle zákonů dané země. (Boedr, 2014)

SAP ERP a SAP Business Suite

Během doby byl SAP ERP podporovaný kolekcí podnikových aplikací specifickými k různým případům. Přední členové SAP Business Suite jsou: (Boedr, 2014)

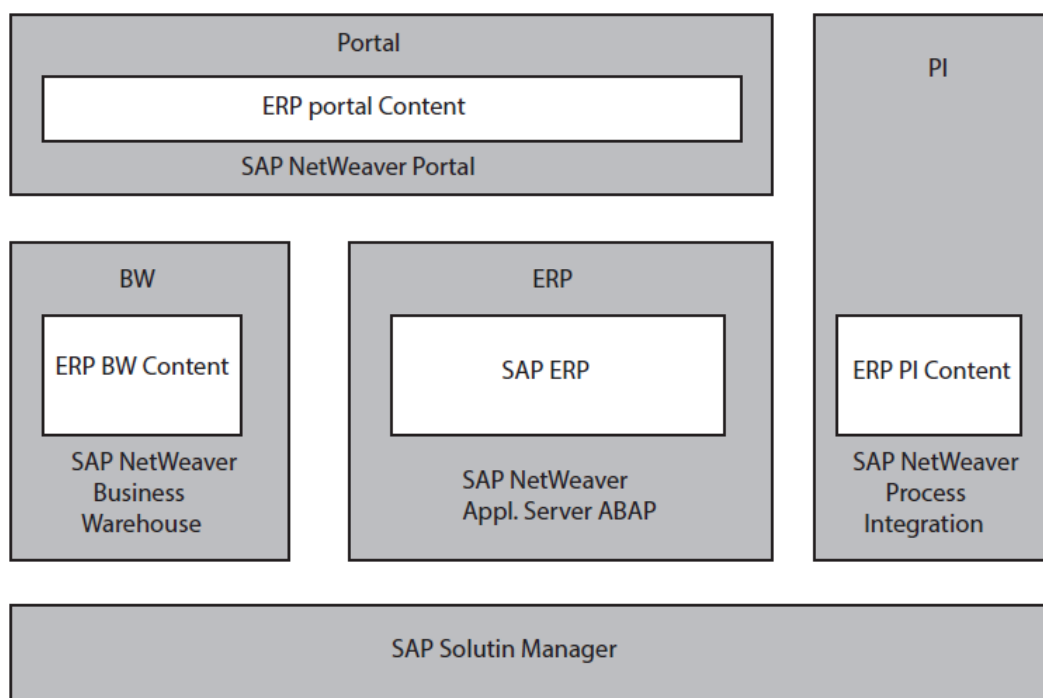
- SAP CRM (Řízení vztahu se zákazníky), který podporuje všechny procesy obsahující přímý kontakt se zákazníkem skrz celý životní cyklus vztahu. SAP CRM obsahuje obchodní scénáře jako prodej v terénu a servis, internetový prodej a centrum interakce se zákazníkem.
- SAP SRM (Řízení vztahu s dodavateli) je obsáhlý přístup k řízení toku informací mezi podnikem a jeho dodavateli. Nákupní data jsou konsolidována a řazena k vytvoření účinných veřejných zakázek, které vedou k účinnému vyjednávání s dodavateli a snižování cen.
- SAP SCM (Řízení dodavatelských řetězců) poskytuje plánování, plnění, regulaci a sledování aktivit dodavatelského řetězce za účelem přidané hodnoty, budování konkurenceschopnosti, silné logistiky, shody nabídky s poptávkou a měření výkonosti.
- SAP BW (Obchodní datový sklad) poskytuje funkce ke skladování dat, platformu pro obchodní znalosti a soupravu pro nástroje manipulace s obchodními znalostmi. Tyhle nástroje umožňují integraci, transformaci a upevnění relevantních informací v SAP BW. SAP BW ulehčuje reportování, analýzu a distribuci informací. Na základě analýzy mohou podniky vytvářet dobře založená rozhodnutí a rozdělit cíleně zaměřené aktivity. Se srozumitelnými před definovanými informačními modely dodanými pro rozdílné role v obchodní činnosti SAP BW zvyšuje použitelnost analytických funkcí a usnadňuje rychlé a nákladově výhodné implementace. SAP BW je klíčová komponenta k SAP NetWeaver.
- SAP GRC (Vedení a řízení krizí) Nabízí organizaci řešení, které je adresováno krizovému managementu, korporáčnímu vedení a dodržuje právní předpisy.
- SAP MII (Manufakturní integrace a znalosti) poskytuje přímé spojení mezi systémem v obchodech a podnikovými operacemi. Zajišťuje, že všechna data mající vliv na manufakturu jsou stále viditelná a aktualizovaná, včetně informací o objednávkách, materiálu, statusu vybavení, nákladů a kvality produktu.

Technická architektura SAP ERP

Runtime environment a designové prostředí SAP ERP je technologická platforma SAP NetWeaver. Všechny části SAP ERP jsou vytvořeny za použití SAP NetWeaver aplikačního serveru ABAP, který poskytuje specifické designové prostředí pro vývoj aplikací a robustního runtime environmentu pro jejich provoz. Následující SAP NetWeaver komponenty infrastruktury jsou používány dohromady se SAP ERP.(Boedr, 2014)

- SAP NetWeaver integrace procesů je middleware pro integraci SAP ERP procesů s procesy běžícími na ostatních SAP aplikacích nebo starších softwarech.
- SAP NetWeaver portál poskytuje na rolích založený pohled na funkcionalitu a informace poskytované více aplikacemi najednou. Umožňuje implementaci podnikového intranetového portálu, který integruje funkcionalitu ze SAP ERP, jako zaměstnanecká samoobsluha pro žádosti o dovolenou nebo pro nákup.

- SAP NetWeaver podnikový datový sklad poskytuje schopnost uchovávat data extrahována ze SAP ERP.
- SAP manažer řešení je nástroj pro zakládání, řízení a monitorování systému. Mnoho administrativních operativních úkolů může být provedeno tímto nástrojem. SAP manažer procesů nabízí následující služby:
 - Dokumentace technického prostředí a podnikových procesů.
 - Implementaci podpor stahující a instalující updaty SAP softwaru.
 - Globální šablony a řízení na základě pracovních postupů
 - Monitorování dostupnosti systému a klíčových procesů.
 - Management incidentů centrálních aplikací.
 - Nabídka zabezpečeného vzdáleného připojení pro SAP konzultanty.



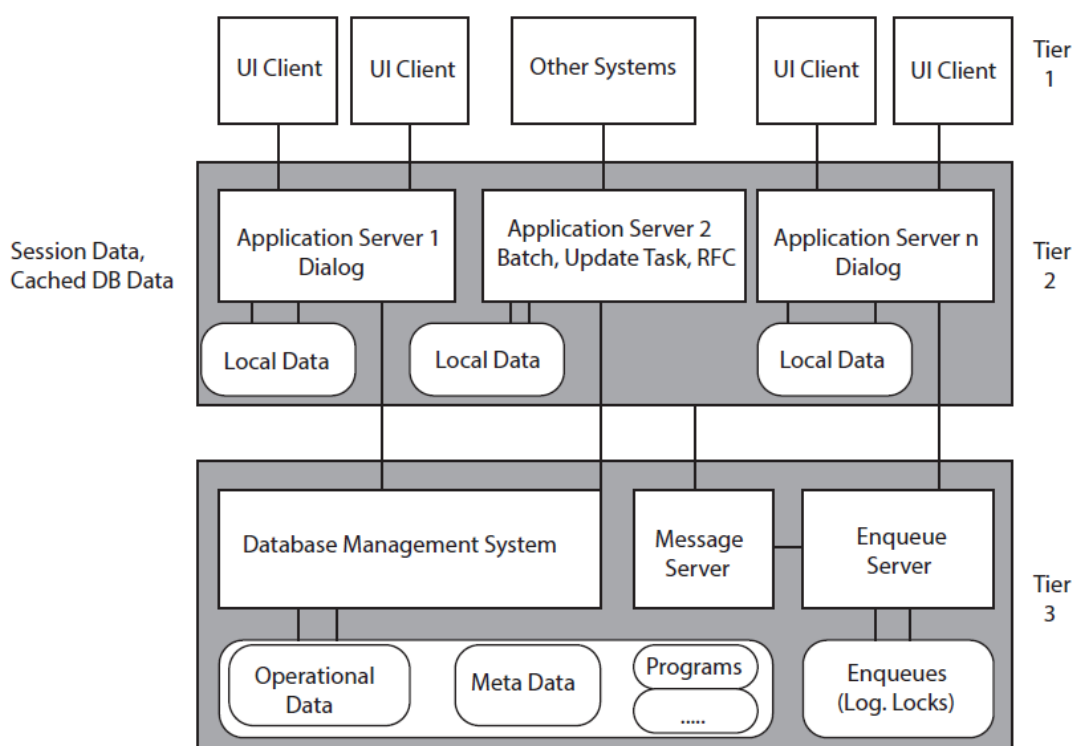
Obrázek 5: SAP ERP běžící na infrastruktuře SAP NetWeaver.(Boedr, 2014)

Během vývoje softwaru je hlavní účel softwarové architektury, aby garantovala, že výsledný software naplní dané kvalitativní požadavky. Mezi kvality, které zapříčinily obrovský úspěch SAP ERP, patří škálovatelnost, robustnost, přenosnost a adaptabilita. Zakládají se na velice rozšířené základně SAP technologií.(Boedr, 2014)

- Škálovatelnost a robustnost jsou dány používáním tří stupňové architektury klient/server. Data jsou uložena v centrálním databázovém systému, zatímco zpracování je prováděno neomezeným množstvím aplikačních serverů, které komunikují s UI klienty. Tato architektura umožňuje škálovatelnost vysoké úrovně.
- Přenosnost je zajišťována abstraktním SAP NetWeaver aplikačním serverem z operačního systému a databáze. Aspekty jako transakce jsou prováděny na aplikačních serverech ne na databázové úrovni.
- Účinný vývoj je podporován přímo k tomu vytvořeným programovacím jazykem ABAT, který umožňuje zaměřením se na podnikovou logiku. Vestavěný přenosový mechanismus umožňuje přenos softwarových změn z jednoho systému na druhý, jako z vývojového systému přes testovací systém do produktivního.

Se SAP ERP operacemi, SAP ERP Financemi a SAP ERP Řízením lidských zdrojů podniková funkcionální byla rozdělena do komponent, které přesto sdílí stejnou databázi. Díky tomu všechny podnikové procesy jsou integrovány jeden v druhém, což umožňuje například procesu prodeje zpustit proces financí a naopak. (Boedr, 2014)

Třístupňová architektura klient server



Obrázek 6: Schéma tří stupňové architektury klient/server. (Boedr, 2014)

Obrázek ukazuje celý přehled architektury. V klientské vrstvě spolu vzájemně komunikují uživatelské prostředí jako SAP GUI nebo webový prohlížeč s uživatelem. Vlastní podniková data jsou zvláště zpracována v aplikační vrstvě. Aplikační servery udržují databázová data v mezi paměti, ale také mohou být použity pro úkoly administrativy. Interaktivní relace jsou nazývány dialogovými relacemi, ale mohou vykonávat i ostatní role relací jsou v aplikačních serverech umožněny. Databázová vrstva implementuje společnou databázi sdílenou napříč všemi aplikačními servery. Tato databáze také obsahuje meta data a zdrojové kódy všech programů. V databázové vrstvě nedochází k žádnému zpracování dat kromě vyhledávání.(Boedr, 2014)

Hlavní výhodou této architektury je škálovatelnost. Pokud je potřeba vyšší výpočetní kapacity nebo najednou zvýší počet uživatelů, administrátor musí pouze přidat další aplikační server do systému. Protože se programy nacházejí v databázi, stačí pouze základní instalační procedury nezbytné ke zpuštění nového stroje. Po registraci stroje začne být server částí systému.(Boedr, 2014)

Jako následek tří systémové architektury, mnoho relací na různých aplikačních serverech potřebují přístup ke sdíleným datům společné databáze. Řešení tohoto problému je na databázové úrovni konceptem transakcí a zámků. Vzhledem k výkonnostním důvodům je logické rozmístit jednu databázovou transakci do více dialogových kroků v jedné relaci. Řešením je zvednutí transakcí a mechanismu zámků z databázové úrovně do SAP NetWeaver aplikačních serverů.(Boedr, 2014)

3.5 ABAP

Programovací jazyk ABAP (Advanced Business Application Programing) je vysoko úroveňový programovací jazyk vytvořený německou firmou SAP, patří také mezi programovací jazyky čtvrté úrovně a byl jedním z prvních jazyků zahrnující Logické databáze. V současné době je vedle Javy jazykem, pro programování SAP Application serveru, který je částí platformy NetWeaver pro tvorbu podnikových aplikací. Pro verzi SAP R/3 je také rozšíření ABAPu pro objektově orientované programování.(Basl, 2012)

Jazyk byl původně používán vývojáři pro tvorbu platformy SAP R/3, ale také aby umožnil zákazníkům tvorbu jejich vlastních aplikací. Ti mohou tvořit vlastní reporty a rozhraní pro svoje aplikace. ABAP tvoří spojení mezi podnikovými aplikacemi a operačním systémem a databází. To zaručuje, že aplikace není závislá na platformě a je snadno přenositelná z jedné platformy na druhou. SAP Netweaver v dnešní době podporuje operační systémy UNIX, Microsoft Windows, i5/OS, IBM Systém i a z/OS. Podporované databáze jsou IBM DB2, Informix, MaxDB, Oracle, a Microsoft SQL Server.(Basl, 2012)

Druhy programů

Programy vytvořené v jazyce ABAP mohou být spustitelné jednotky nebo knihovny poskytující znovupoužitelný kód pro ostatní programy, které nemusejí být nutně samy o sobě spustitelné. ABAP rozlišuje dva typy spustitelných programů. Prvním jsou reporty, což jsou relativně jednoduché programy, kde uživatel zadá jednoduchá vstupní data, kte-

rých program využije k tvorbě reportu v podobě interaktivního listu. Druhým typem jsou Module pools. Tyto programy jsou daleko komplexnější a využívají mnohem složitější vstupy, tvořené více zdroji dat.(Sap, 2012A)

Tvorba programů

Pro tvorbu programů v jazyce ABAP jsou tři různé způsoby. Jejich použitelnost záleží na verzi používaného systému. Prvním způsobem je ABAP Workbench, k tvorbě programů má 6 různých nástrojů.(Dolinskaja, 2012)

- ABAP Editor - tvorba a úpravy reportů a Module pools (transakce SE38)
- ABAP Dictionary - pro práci s databází (transakce SE11)
- Menu Painter - pro tvorbu uživatelského rozhraní (transakce SE41)
- Screen painter - pro designování obrazovky na výstupu datových toků (transakce SE51)
- Function builder - pro tvorbu modulů funkcí
- Class Builder – pro objekty tříd a jejich rozhraní

Druhým způsobem je ABAP Development tools. Je to množina pluginů pro platformu Eclipse. Developer v tomto případě instaluje potřebné nástroje a pracuje lokálně na jeho počítači. Třetím způsobem je SAP Web IDE. Jedná se o prohlížečově založené vývojové prostředí. Developer používá prohlížeč k přístupu k Web IDE.

3.6 Kmenová data

Kmenová data jsou data o produktech, subjektech, objektech, zákaznících, dodavatelích a dalších obchodních partnerech, o majetku, lidských zdrojích, financích, výrobních kapacitách, místech nebo věcech a popisují jejich vlastnosti a parametry. Jsou dlouhodobá a je u nich důležitá kvalita. Jsou důležitým zdrojem informací pro fungování každé organizace. Prochází také napříč různými podnikovými procesy a mohou být ukládána ve více aplikacích, většinou v ERP systémech, CRM a dalších. Pokud jsou data ve špatné kvalitě, mohou zapříčinit chyby a špatná rozhodnutí a tak i zvýšení procesních nákladů. Podniky by měli jejich kvalitě a řízení přikládat velký důraz.(Management mania, 2016)

CDHDR

CDHDR(Change document header) je tabulka, používaná k ukládání hlaviček provedených změn. Tabulka je dostupná ve většině verzí SAP R/3. Zobrazení tabulky je dostupné transakcemi SE11 nebo SE80, alternativní zobrazení dat lze také dosáhnout transakcí SE16. Tabulka obsahuje 14 polí, z toho jsou 4 klíčové.(Sap, 2012B)

Field	Data Element	Data Type	length (Decimals)	Checktable	Description
MANDANT	MANDT	CLNT	3	T000	Client
OBJECTCLAS	CDOBJECTCL	CHAR	15		Object class
OBJECTID	CDOBJECTV	CHAR	90		Object value
CHANGENR	CDCHANGENR	CHAR	10		Document change number
USERNAME	CDUSERNAME	CHAR	12		User name of the person responsible in change document
UDATE	CDDATUM	DATS	8		Creation date of the change document
UTIME	CDUZEIT	TIMS	6		Time changed
TCODE	CDTCODE	CHAR	20	TSTC	Transaction in which a change was made
PLANCHNGNR	PLANCHNGNR	CHAR	12		Planned change number
ACT_CHNGNO	CD_CHNGNO	CHAR	10		Change number of the document created by this change
WAS_PLANND	CD_PLANNED	CHAR	1		Flag that changes were generated from planned changes
CHANGE_IND	CDCHNGINDH	CHAR	1		Application object change type (U, I, E, D)
LANGU	LANGU	LANG	1	*	Language Key
VERSION	CHAR3	CHAR	3		3-Byte field

Obrázek 7: Schéma tabulky CDHDR.(Sap, 2012B)

- První pole také první klíčová hodnota MANDANT, která ukládá informace o klientovi. Hodnota v této položce se odkazuje na objekt tabulky T000, což je tabulka popisující informace o klientovi.
- Další klíčovou hodnou je pole OBJECTCLASS uchovávající informace o změněném objektu.
- Třetí hodnotou je OBJECTID. Položka uchovává informaci o hodnotě objektu.
- Čtvrtou a poslední klíčovou hodnotou je CHANGENR. Položka je číslem změny dokumentu.
- Pátá položka je USERNAME. Položka uchovává uživatele zodpovědného za změnu dokumentu.
- Šestá položka je UDATE. Položka uchovává datum provedené změny ve formátu RRRRMMDD. Tato položka je velice důležitá k vyhledávání informací, jelikož databáze provedených změn má obrovské množství dat a datum provedené transakce pomáhá vyčlenit poměrně přehledné hledaný výsek informací.

- Sedmou položkou je UTIME. Položka uchovává čas provedené změny dokumentu. Je uložena ve formátu HHMMSS. Jelikož během jednoho dne není provedeno příliš mnoho změn, nemá tato položka velký význam při vyhledávání, ale spíše pro informativní využití.
- Osmou položkou je TCODE. Je to kód transakce, kterou byla změna vytvořena. Hodnotou v této položce je objekt tabulky TSTC, který uchovává informace o transakci použité k vytvoření změny dokumentu.
- Devátá položka je PLANCHNGNR. Ukládá číslo měněné informace a identifikuje hlavní záznam změn. Tento záznam kontroluje správnost objektu, který má být měněn a loguje provedené změny.
- Desátá položka je ACT CHNGNO. Udává číslo dokumentu tvořené touto změnou.
- Jedenáctá položka je WAS PLANND. Ukládá informaci z plánované změny.
- Dvanáctá položka je CHANGE IND. Zaznamenává typ provedené změny. Typy změn mohou být: U(update) - změna, I(Insert) - vložení, D(delete) - smazání.
- Třináctá položka je LANGU. Tato položka udává alfanumerický kód udávající jazykový klíč.
- Poslední čtrnáctou položkou je VERSION. Zde je uložena 3-Bytová informace.

CDPOS

CDPOS(Change document items) je tabulka, určená k ukládání položek dokumentů změn. Je stejně jako tabulka CDHDR dostupná od většiny verzí SAP R/3. Také ji je možné zpřístupnit stejnými transakcemi (SE11, SE80, SE16). Každý jeden objekt této tabulky je navázán na svoji hlavičku v tabulce CDHDR v poměru 1:M, takže jeden objekt tabulky CDHDR může být hlavičkou k více objektů tabulky CDPOS. Tabulka se skládá z patnácti polí, kde osm z nich jsou klíčové hodnoty.(Sap, 2012C)

- První čtyři z těchto klíčových hodnot jsou totožné s tabulkou CDHDR (MANDANT, OBJECTCLAS, OBJECTID, CHANGENR). Položka CHANGENR zde odkazuje na tabulku CDHDR.
- Pátá klíčová hodnota je TABLENAME. Jednoduchý řetězec znaků, který udává jméno tabulky nebo struktury.
- Šestá klíčová položka je TABKEY. Informace o klíči tabulky.
- Sedmá klíčová položka je FNAME. Uchovává název pole, obsahující přechodný záznam dokumentu změny.
- Poslední osmou klíčovou hodnotou je CHNGIND. Jedná se o indikátor typu provedené změny (U, I, S, D).

Field	Data Element	Data Type	length (Decimals)	Checktable	Description
MANDANT	MANDT	CLNT	3	T000	Client
OBJECTCLAS	CDOBJECTCL	CHAR	15		Object class
OBJECTID	CDOBJECTV	CHAR	90		Object value
CHANGENR	CDCHANGENR	CHAR	10	CDHDR	Document change number
TABNAME	TABNAME	CHAR	30	*	Table Name
TABKEY	CDTABKEY	CHAR	70		Changed table record key
FNAME	FIELDNAME	CHAR	30	*	Field Name
CHNGIND	CDCHNGIND	CHAR	1		Change Type (U, I, S, D)
TEXT_CASE	CDXFELD	CHAR	1		Flag: X=Text change
UNIT_OLD	CDUNIT	UNIT	3		Change documents, unit referenced
UNIT_NEW	CDUNIT	UNIT	3		Change documents, unit referenced
CUKY_OLD	CDCUKY	CUKY	5	TCURC	Change documents, referenced currency
CUKY_NEW	CDCUKY	CUKY	5	TCURC	Change documents, referenced currency
VALUE_NEW	CDFLDVALN	CHAR	254		New contents of changed field
VALUE_OLD	CDFLDVALO	CHAR	254		Old contents of changed field

Obrázek 8: Schéma tabulky CDPOS.(Sap, 2012C)

- Devátou položkou, která již není klíčovou, je TEXT CASE. Položka uchovává informaci typu textu.
- Desátou položkou je UNIT OLD. Tato položka uchovává, informaci o položce před provedenou změnou. Vzhledem ke třem možným druhům změn, bude položka nabývat neprázdné hodnoty v případě smazání položky nebo její změny a prázdné hodnoty v případě vložení položky.
- Jedenáctá položka je UNIT NEW. Uchovává informaci o položce po provedené změně a bude stejně jako předchozí položka záviset na typu provedené změny, kdy neprázdnou hodnotu nabyde při vložení nebo změně položky a prázdnou hodnotu při jejím smazání. V této položce se nachází odkaz na objekt tabulky TCURC.
- Dvanáctá položka je CUKY OLD. Položka uchovává druh hodnoty před změnou. Prázdnost této položky se bude řídit stejným pravidlem jako ostatní položky popisující měněný dokument před změnou. V této položce se také nachází odkaz na objekt tabulky TCURC.
- Třináctá položka je CUKY NEW. Položka uchovává druh hodnoty po změně položky. Prázdnost hodnoty této položky se bude odvíjet stejně jako ostatní položky popisující měněný dokument po změně. Obě položky CUKY OLD a CUKY NEW mají vel-

ký význam, pokud se měněný dokument týká materiálů ve výrobě, který se udává v nejrůznějších jednotkách jako kilogramy či litry nebo pokud se týká peněz, kdy je důležité uchovávat informaci o měnách.

- Čtrnáctá položka je VALUE NEW. Položka udává hodnotu po změně dokumentu. Prázdnota položky se znovu řídí stejnými pravidly jako ostatní položky po změnách.
- Poslední patnáctou hodnotou je VALUE OLD. Tato položka udává hodnotu po provedené změně dokumentu a její prázdnota se řídí také stejnými pravidly jako je tomu u ostatních položek popisující hodnoty před změnami dokumentů.

3.7 UML

Sofwarové inženýrství klade důraz na souhru mezi jednotlivými úseky. Je důležité, aby analytici a vývojáři vyvíjeli pevné základy, programátoři, naproti tomu sdílí s nimi své myšlenky, upravují své návrhy a vylepšují kód. Tenhle postup je velice důležitý pro vzájemné pochopení problematiky. Celý vývojový proces musí být spojen s týmovým přístupem, kvůli specifickým znalostem, které řeší specializované osoby. Také je velice důležité mít tým analytiků, kteří řeší komunikaci s klientem. Jazyk UML je skvělý způsob jak zachytit požadavky klienta vhodnou formou, která je srozumitelná pro celý vývojový tým.(Schmuller, 2001)

UML je souhrnem především grafických notací používaných k vyjádření analytických a návrhových modelů. Umožňuje modelování složitých i jednoduchých aplikací za použití stejné formální syntaxe. Díky tomu je umožněno výsledky práce sdílet s každým, kdo je schopen s UML diagramy pracovat.(Kanisová, 2006)

UML je jeden z nejperspektivnějších nástrojů k vývoji systému a to díky schopnosti vytvářet tzv. modrotisky. Plán návrhu systému vytvořeného v UML vychází z důkladné analýzy potřeb klienta a vývoj je možné rozčlenit do krátkých časových úseků. Jazyk UML se skládá z mnoha grafických prvků, které se mezi sebou dají kombinovat a vytváří při jejich spojení diagramy. Mezi takové diagramy patří například diagram případu užití, stavový diagram, diagram sekvencí, diagram spolupráce, diagram činností, diagram komponent, nebo diagram nasazení.(Schmuller, 2001)

Diagramy případů užití

Tyto diagramy představují užitečný koncept, díky kterým analytik snáze pochopí klientovi požadavky a jak by se měl systém chovat. Na základě diagramu případu užití lze stanovit požadavky, které budou na systém kladeny uživateli. Díky vizualizaci je možné navržené případy prezentovat klientům nebo uživatelům a získat tak od nich velice užitečnou zpětnou vazbu. Diagram případu užití je tak jeden z hlavních cílů systémové analýzy.(Schmuller, 2001)

Stavové diagramy

Stavové diagramy slouží k popsání změn systému v čase. Změny v systému jsou přímým důsledkem toho, že jednotlivé objekty systému mění v čase svůj stav. Tyto změny stavu nastávají reakcí na jiné události nebo jen na prostý fakt ubíhání času. Stavový diagram je schopný všechny tyto změny popsat spolu s podmínkami, za kterých změny nastanou. (Schmuller, 2001)

Diagramy sekvencí

Diagram sekvencí popisuje komunikaci objektů systému v čase. Na rozdíl od diagramů stavových zachycují i změny, kterými objekty během běhu systému prochází. Další schopností diagramů sekvencí je možnost popsat jak objekty spolupracují mezi sebou. Diagram se skládá z objektů zakreslených běžným způsobem a přerušovaných čar, znázorňující běh času a plných šipek, které slouží jako reprezentace zpráv mezi objekty. (Schmuller, 2001)

Diagramy spolupráce

Diagram spolupráce zobrazuje interakci mezi objekty systému, je velice podobný a významově ekvivalentní diagramu sekvencí, ale na rozdíl od něj zdůrazňuje kontext a uspořádání spolupracujících objektů. Jinak řečeno, zachycuje co se děje v prostoru. Diagram spolupráce je rozšířením diagramu objektů, mimo zobrazení samotných objektů, zobrazuje také zprávy, které si objekty mezi sebou předávají. Zprávy jsou zakreslovány šipkami zakreslenými vedle asociačních čar spojující objekty. (Schmuller, 2001)

Diagramy činností

Diagram činností popisuje jednotlivé kroky procesů. Podobá se běžnému vývojovému diagramu, který je používán programátory. Kromě zobrazování jednotlivých kroků, zachycuje také body, v kterých dochází k rozhodování a větvení programu, což velice jednoduše lze použít pro firemní procesy a systémové analýzy. Diagram činností je prostředek ke zjednodušení přehledu jednotlivých kroků operací nebo procesů. Vzniká rozšířením stavového diagramu, od kterého se liší zaměřením na zobrazení činností uvedených ve stavovém diagramu. (Schmuller, 2001)

3.8 ERD

Entitně-Relační diagram je tvořen aplikací konstrukčních prvků. Je to jedna z prvních metod konceptuálního datového modelování. Základními prvky jsou entity a vztahy. (Kaluža, 2012)

Entity reprezentují typ objektů reálného světa a zakreslují se pomocí obdélníku, kde ve vrchní části je uveden název entity a ve spodní části jména atributů entit. Atribut reprezentuje elementární vlastnost entity, jako je například jméno nebo příjmení osoby. (Kaluža, 2012)

Vztahy reprezentují asociace mezi entitami. Každý vztah je možné charakterizovat třemi charakteristikami: Stupněm, který charakterizuje počet asociovaných entit, kardinalitou, která vyjadřuje počet výskytů entit a volitelností, která vyjadřuje, jestli je vztah povinný nebo ne.(Kaluža, 2012)

Mezi další prvky ERD patří doména a klíč. Doména charakterizuje množinu příslušných hodnot vztahující se k jednomu nebo více atributům a klíč je jeden nebo více atributů identifikující výskyt dané entity.(Kaluža, 2012)

3.9 Tří systémová architektura

Podniková architektura systému je rozřazena do tří systémů. Každá provedená změna je provedena nejprve ve vývojovém systému, poté je nahrána na testovací server, kde je testována a až nakonec nahrána do produkčního systému. Informace o těchto systémech byly získány z konzultací s pracovníky IT oddělení podniku.

Vývojový systém

Tento systém je jako jediný odemčen pro změny a tvorby programů, přístup k němu má pouze správa IT oddělení. Systém má aktualizovanou databázi dat, totožnou s produktivním systémem, ale cokoliv se zde změní, vytvoří, nebo se z něj odešle, nemá žádný vliv na reálné prostředí, včetně spuštěných transakcí k objednávkám zboží či služeb. Vývojový systém je tudíž primárně určen k tvorbě programů a aplikací. Tyto Programy a aplikace jsou tvořeny z velké části konzultanty. Zbývá část převážně jednodušších programů je vytvářena IT oddělením, jde převážně o podpůrné programy sloužící k usnadnění práce s daty.

Testovací systém

Testovací systém má také aktualizovanou databázi dat, ale nejsou v něm umožněny žádné změny. Systém slouží pouze k testování aplikací a transakcí, vytvořených ve vývojovém systému. Oprávnění ke spuštění transakcí je možné přidělit více uživatelům, zejména kvůli zjednodušení procesu testování, díky čemuž se může obejít zdoluhavý systém potvrzování objednávek nebo jiných transakcí, kde je jinak nutné mít kontrolu nad činností uživatelů.

Produktivní systém

Tento systém pracuje s reálnými daty, je tudíž potřeba dbát velké pozornosti nad udělováními právy pro spuštění transakcí různých uživatelů, kteří musí být autentizováni pouze svými vlastními účty, na který musejí být autorizováni pouze k výkonu těch transakcí, které přímo souvisejí s jejich pracovní činností, a také by neměli být schopni provádět ty aplikace, které přímo nepotřebují. Mnoho transakcí podléhá schvalovacímu systému, kdy se po spuštění transakce čeká na její schválení dalším uživatelem zodpovědným za kontrolu. Tento postup se uplatňuje zejména u tvorby objednávek, kde pracovník objednávající

materiál či služby není sám schopen objednávku vyřídit, ale pouze založit. Poté může být objednávka schválena dalším uživatelem k tomu zodpovědným.

4 Metodika vypracování

Tato kapitola bude obsahovat popis postupu tvorby této bakalářské práce.

Samotná bakalářská práce je tvořena kapitolami: Úvod, Cíl práce, Aktuální stav řešené problematiky, Tvorba aplikace, Nasazení, Závěr a Přílohy. V kapitole Aktuální stav řešené problematiky jsou uvedeny jak základní informace o informačních systémech, tak jsou zde i popsány základy architektury SAPu. V této kapitole jsou i uvedeny informace o kmenových datech a teoreticky jsou zmíněny i diagramy použité při návrhu aplikace.

4.1 Navrhování aplikace

Prvním krokem byla analýza požadavků pro aplikaci, podle kterých byla aplikace tvořena. Dalším krokem bylo navrhnutí aplikace. Návrh aplikace je tvořen čtyřmi UML diagramy a jedním ERD diagramem. UML diagramy použité pro návrh byly:

- Diagram případu užití
- Diagram stavový
- Diagram sekvencí
- Diagram spolupráce
- Diagram činností

Jazyk UML byl vybrán pro návrh aplikace vzhledem k mnoha diagramům, které je možné použít k popisu aplikace z více hledisek. Při použití ERD byla vybrána Martinova notace, která jasně rozlišuje kardinalitu.

4.2 Implementace kódu

Po vytvoření návrhu začala implementace kódu aplikace. Jako první bylo vytvořeno vstupní okno, které slouží v zadávání vstupních kritérií pro chod aplikace. Poté byla naimplementována část kódu pracující s databázovou vrstvou SAP architektury, pro kterou byl použit jazyk SQL. Další části pracující s aplikační vrstvou byly implementovány pomocí programovacího jazyka ABAP, který byl použit i pro vytvoření vstupního okna a pro tvorbu rozhraní pro zobrazení výstupu aplikace. Oboje je částí uživatelské vrstvy SAP architektury. Programovací jazyk ABAP byl vybrán kvůli skvělé kompatibilitě se systémem a také proto, že je aktivně využíván v podniku pro tvorbu aplikací.

4.3 Zavádění aplikace

Po dokončení tvorby aplikací se provádí první testy funkčnosti již ve vývojovém systému. Během tvorby aplikací se také provádí zkoušky funkčnosti jednotlivých částí aplikací. Proto je nutné každou část aplikace zvlášť otestovat stejně jako celou aplikaci po jejím

vytvoření. Po těchto testech se provádějí první změny, než se aplikace převede do testovacího systému.

Dalším krokem je provedení ostrých testů v testovacím systému, kde jsou odhalovány všechny potenciální nedostatky. Tyto nedostatky je nutné odstranit ve vývojovém systému.

Poté co nejsou v testovacím systému nalezeny žádné nedostatky, jsou aplikace vloženy do produktivního systému.

5 Tvorba aplikace

5.1 Požadavky na aplikaci

Požadovaná aplikace má za úkol zobrazit změny na kmenových datech podle některých z několika kritérií. Aplikace bude vytvořena podle potřeb IT oddělení Heineken Česká republika. Důvodem tvorby aplikace je přidělení správy kmenových dat pod IT oddělení. Tím pádem vzniká potřeba zpřístupnit pouze požadované informace rychle a v přehledné formě. V systému již existují transakce schopné zobrazit tabulky, které obsahují kompletní informace o změnách na kmenových datech. Ty jsou rozmístěny do více tabulek, takže rychlé zjištění důležitých informací napříč více tabulkami není jednoduché. Kritéria vyhledávání vybraná pro vyhledávací aplikaci jsou určena dle zadaných požadavků podniku:(Mičánek, 2015)

- Change document object (Měněný dokument objektu)
- Object value (Hodnota objektu)
- Document number (Číslo dokumentu)
- User (Uživatel, který provedl změnu)
- Date (Datum, kdy byla změna provedena)
- Time (Čas provedené změny)
- Transaction (Transakce, kterou byla změna provedena)

Forma výstupu je dána nástroji, kterými programovací jazyk ABAP disponuje a jsou určeny k tvorbám reportů. Výstup bude obsahovat proměnné tabulek CDHDR a CDPOS a to konkrétně:(Mičánek, 2015)

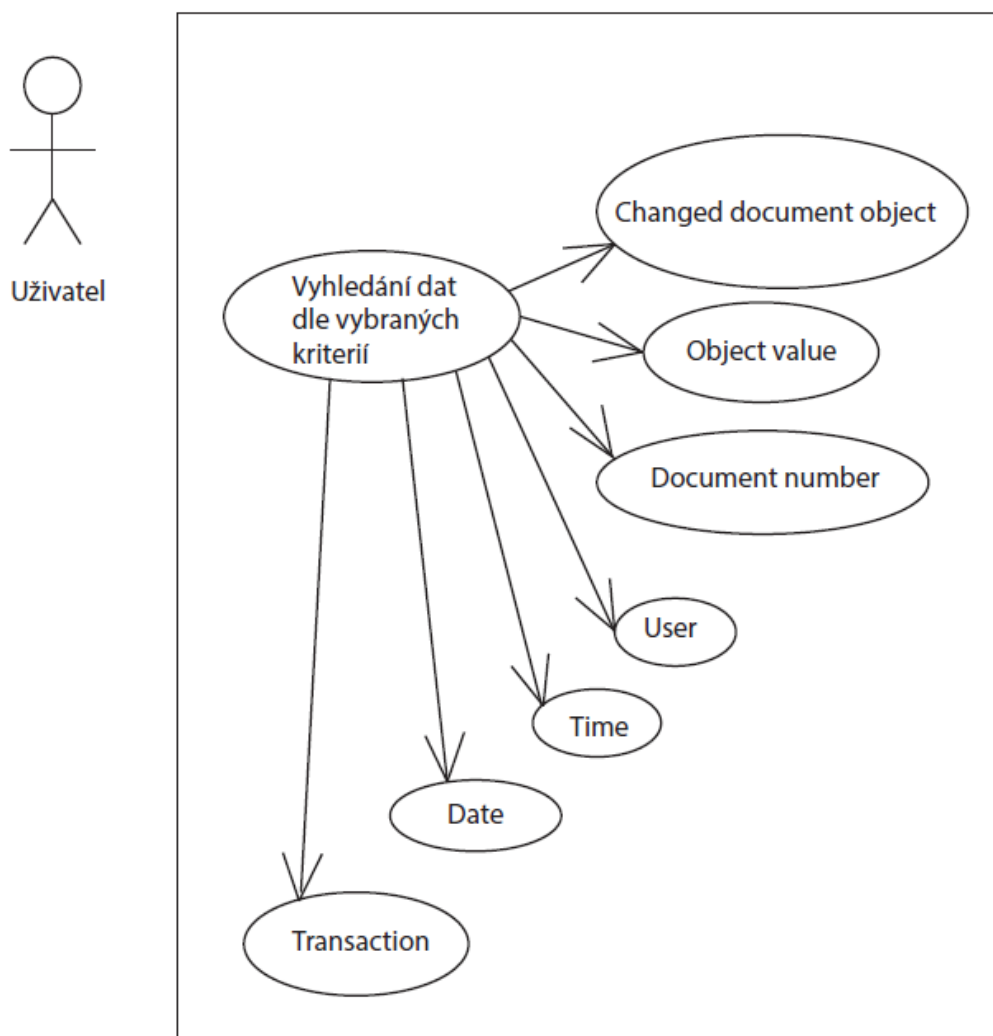
- Change document object (Měněný dokument objektu)
- Object value (Hodnota objektu)
- Document number (Číslo dokumentu)
- User (Uživatel, který provedl změnu)
- Date (Datum, kdy byla změna provedena)
- Time (Čas provedené změny)
- Transaction (Transakce, kterou byla změna provedena)
- Old Value (Původní změněná hodnota)
- New Value (Nová hodnota, na kterou byla položka změněna)

Jako další bude aplikace poskytovat přehled, ve kterém bude uveden celkový počet změn a pro každého uživatele, který provedl nějakou změnu v hledané množině položek, uveden přesný počet těchto změn. (Mičánek, 2015)

5.2 Návrh aplikace

Pro tvorbu programu bylo vytvořeno 5 UML diagramů, které popisují aplikaci z 5 různých hledisek: Diagram užití, Stavový diagram, Sekvenční diagram, Diagram spolupráce a Diagram činností. Jako další byl vytvořen Entity relationship diagram (ERD).

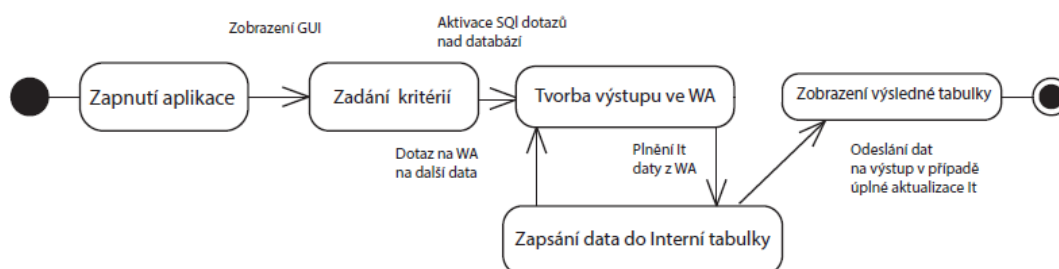
Diagram užití



Obrázek 9: Diagram užití aplikace

Aplikace se zabývá jednou hlavní funkcí a to je vyhledávání. Vyhledávat je možné podle několika hledisek a to podle Měněného dokumentu objektu, hodnoty objektu, čísla dokumentu, uživatele, který změnu provedl, času a data, v který byla změna provedena transakcí, která provedla změnu. Aplikace je spouštěna v základní verzi pouze uživatelem.

Stavový diagram



Obrázek 10: Stavový diagram aplikace

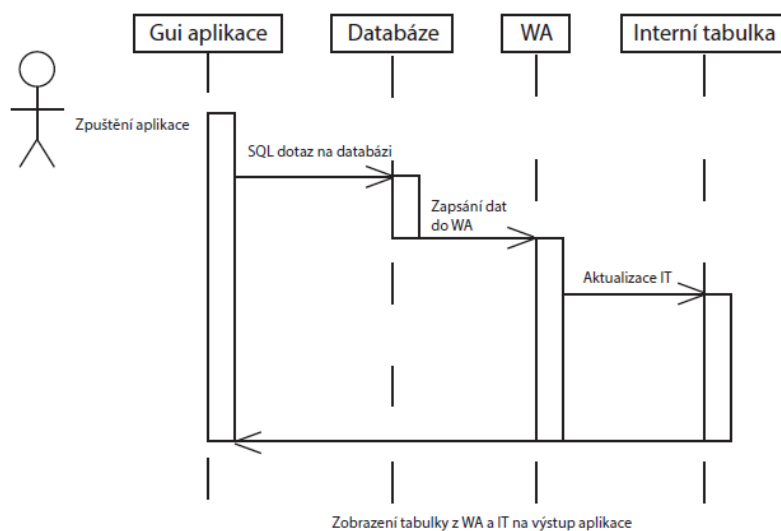
Aplikace se může nacházet ve stavu zapínání, po kterém se dostane zobrazením grafického uživatelského rozhraní do stavu, kde uživatel zadá kritéria, podle kterých započne vyhledávání. V aplikaci je vyhledávání řešeno jazykem SQL. Vyhledané informace jsou poté zapisovány do working area, kde se vykresluje tabulka s informacemi o výstupu. Během zápisu dat do working area se vytváří interní tabulka, kde jsou prováděny výpočty dodatečných informací potřebných pro celkový výpis. Ve chvíli, kdy se working area a interní tabulka zcela zaplní, se zobrazí výstup výsledné tabulky spolu s dodatečnými daty.

Sekvenční diagram

V tomto diagramu je zobrazená časová osa chodu aplikace. Uživatel zadá v grafickém rozhraní kritéria pro vyhledávání a pomocí SQL dotazů jsou z databáze vyhledána data, která jsou uložena do working area a interní tabulky. Průběh ukládání dat do working area a následně do interní tabulky probíhá celou dobu, protože každá část dat je uložena do working area a poté zpracována a zapsána do interní tabulky. Tento proces se opakuje, dokud nejsou uložena a zpracována všechna vyhledaná data.

Diagram spolupráce

V tomto diagramu jsou zaznamenány vztahy mezi jednotlivými objekty. Uživatel přistupuje k aplikaci pomocí grafického rozhraní aplikace, kde zadá požadovaná kritéria do



Obrázek 11: Sekvenční diagram aplikace

vyhledávání a na základě těchto kritérií aplikace vyhledá z databáze hledané informace. Tato data jsou ukládána do working area, odkud jsou použita pro tvorbu interní tabulky a po skončení vyhledávání spolu s interní tabulkou zobrazeny na výstupu.

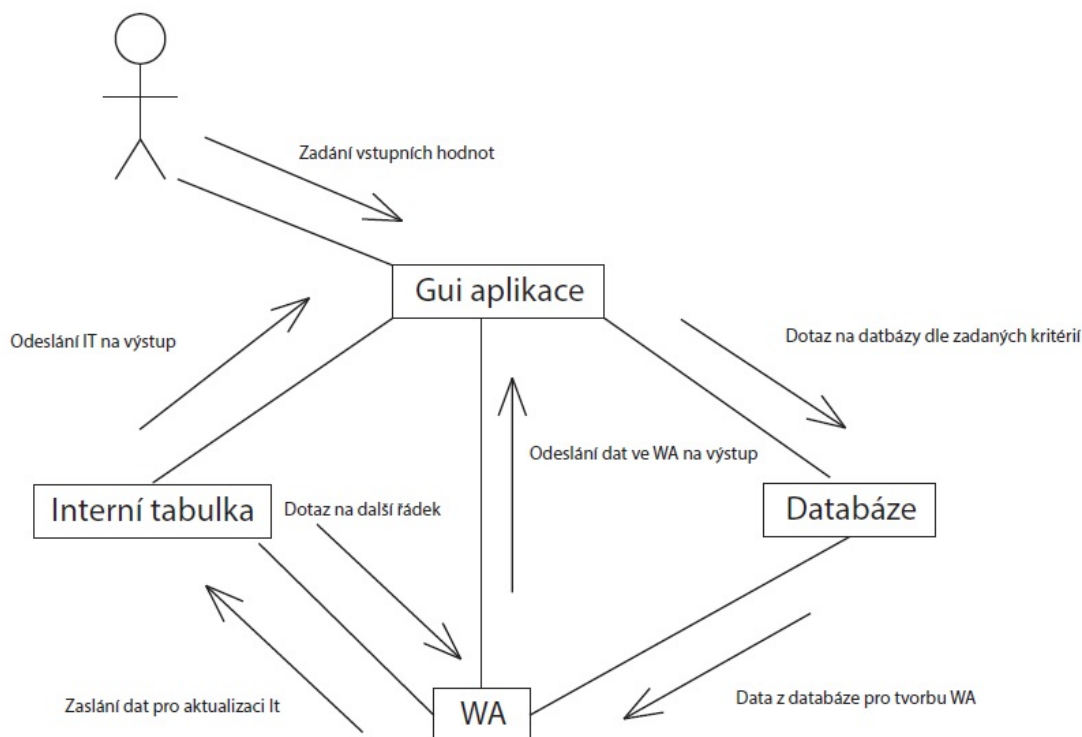
Diagram činnosti

Diagram činností na obr. 13 zobrazuje činnosti prováděné během chodu aplikace. Po zadání vstupních dat do aplikace se začnou tyto data vyhledávat a zapisovat do working area. Během zápisu do working area se vytvoří a po každém zápise aktualizuje také interní tabulka s dodatečnými informacemi. Tyto informace sdělují, kolik změn každý uživatel provedl. Při každé aktualizaci interní tabulky se vyskytují dvě možnosti. První možnost je, že uživatel již figuruje v interní tabulce, takže je pouze změněn počet změn, které provedl. Při druhém případě uživatel v interní tabulce ještě zapsaný není, a tak je tabulka rozšířena právě o tohoto uživatele. Data z WA a Interní tabulky jsou zobrazena až v případě, že byla všechna vyhledaná data uložena do WA a následně zpracována v Interní tabulce.

ERD aplikace

Na obrázku 14 jsou znázorněny 4 tabulky. Tabulka Interní tabulka a Working area jsou tvořené aplikací a CDHDR a CDPOS jsou tabulky v databázi.

V tabulce working area je uložen report z vyhledaných dat z CDPOS a CDHDR. V interní tabulce je uložen seznam uživatelů a počet změn a celkový počet změn, které provedli v množině vyhledaných dat.



Obrázek 12: Diagram spolupráce aplikace

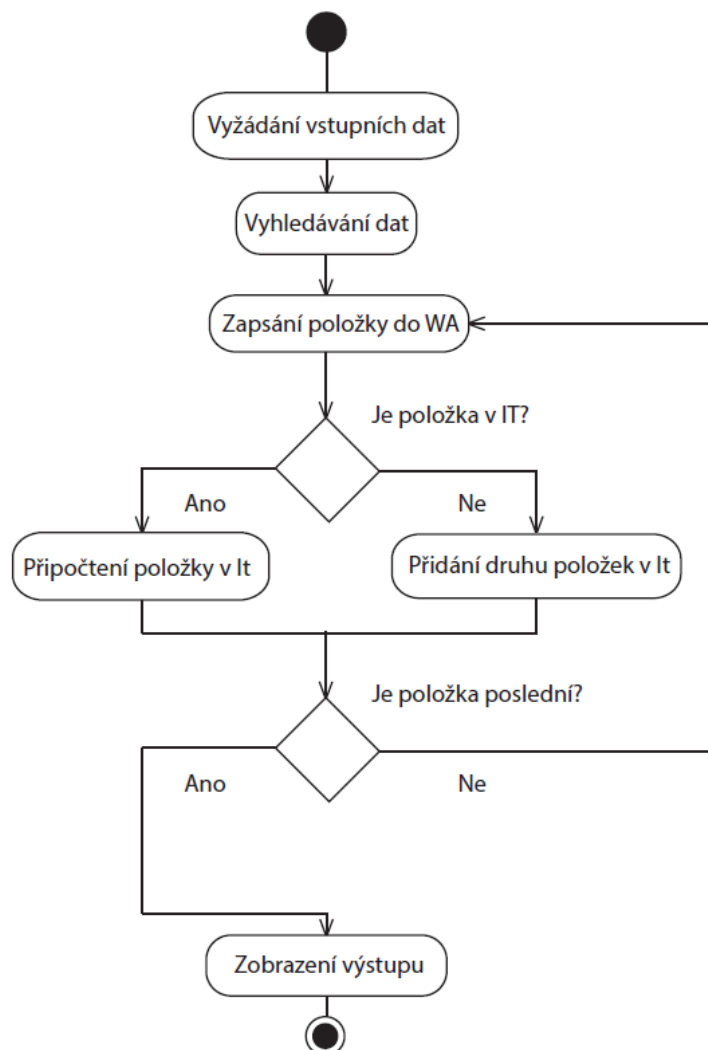
5.3 Vstupní okno aplikace

Na obrázku 15 je vidět vstupní okno aplikace, kam jsou do příslušných polí kritéria zadávána. Pro zadávání hodnot slouží dva sloupce. Díky tomu je možné kritéria zadávat v určitém rozsahu, kde levý sloupec je pro hodnoty, od kterých aplikace začne vyhledávat a pravý sloupec pro hodnoty, po které má být vyhledávání uskutečněno. U Kritéria datum je vidět checkbar, který znázorňuje povinné pole. Důvodem k tomu je obrovské množství dat, která mohou být vyhledávána a zadání data nejvíce zúží množinu hodnot, které mohou být vyhledány. Díky možnosti zadat do vyhledávání interval hodnot není povinné pole data nějak omezující.

Tvorba oken pro zadávání informací a kritérií se v programu ABAP vytvoří příkazem „select-options“, před kterým je nutné uvést zdroj informací. (SAP documentation, 1996a)

Vstupní okno aplikace se již prolíná s uživatelskou vrstvou systému a je tedy potřeba dodržovat jednotné postupy při tvorbě grafického rozhraní aplikace.

Tato struktura byla použita, protože je v podniku klasicky využívána u většiny ostatních aplikací. Bary, kde se vybírá z dat, jako jsou čas a datum, mají předdefinovaný formát a místo vypisování hodnot, se hodnoty vybírají z nabídky pomocí kalendáře a klasického časovače.



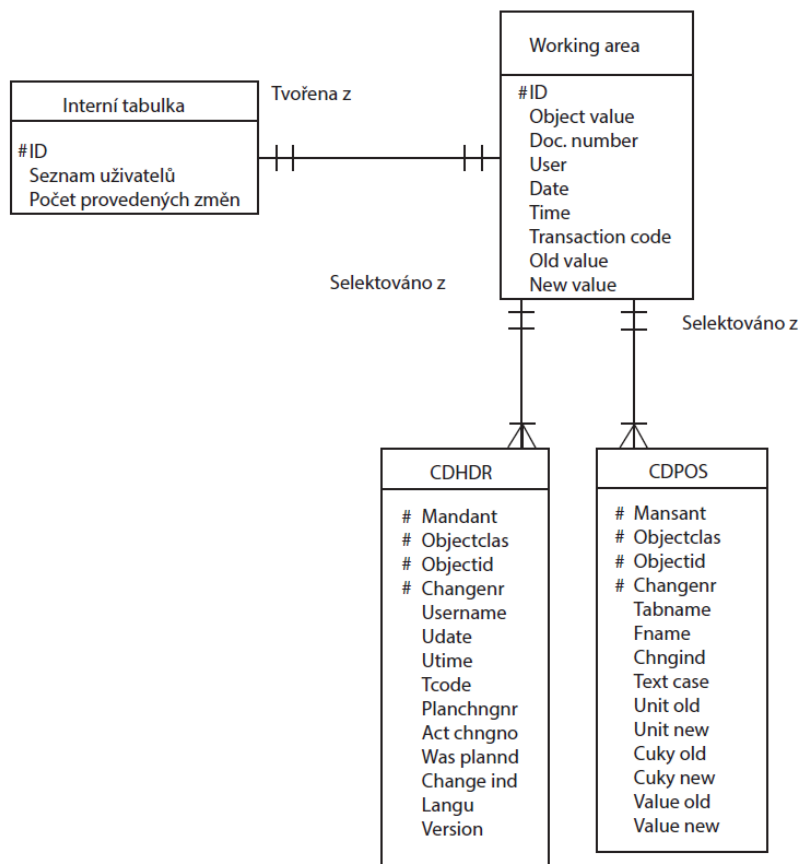
Obrázek 13: Diagram činností aplikace

Pro aplikaci byla použita sekvence příkazů:

```

tables: cdhdr, cdpos.
select-options:
  Ob_class for cdhdr-OBJECTCLAS,
  Ob_value for cdhdr-OBJECTID,
  Doc_numb for cdhdr-CHANGENR,
  User for cdhdr-username,
  Date for cdhdr-udate obligatory,
  Time for cdhdr-utime,
  Trans_cd for cdhdr-tcode.
  
```

Kde příkaz „tables: cdhdr, cdpos“ definují tabulky CDHDR a CDPOS jako zdroje dat. Příkaz „select-options“ zadává grafické rozhraní okna a zbytek příkazů pojmenovává



Obrázek 14: Znázornění ERD

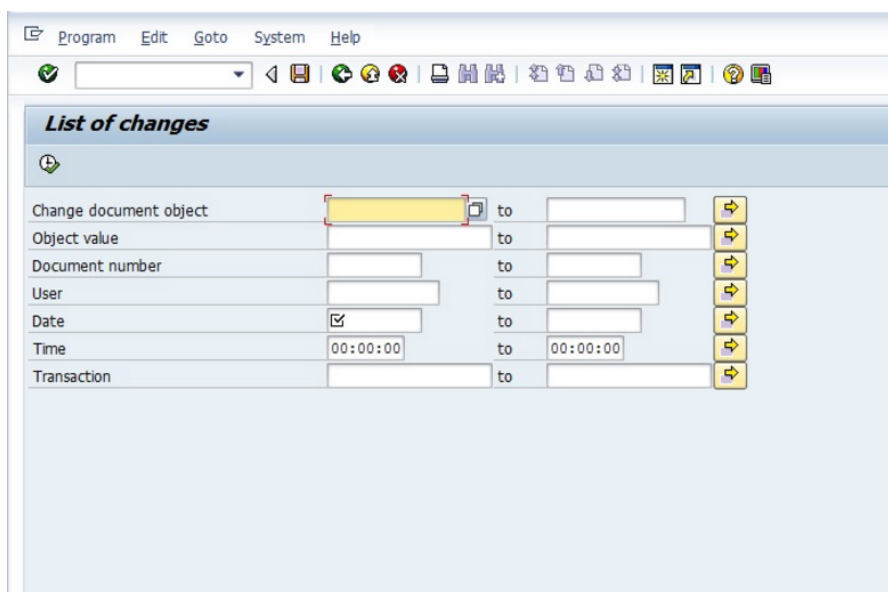
pole, do kterých jsou zadávána kritéria a proměnné data tabulek, podle kterých se bude vyhledávat.

5.4 Tělo aplikace

Samotné vyhledávání dat je prováděno SQL dotazy, hledající data z obou tabulek. Vyhledávací část jako jediná pracuje s databázovou vrstvou, což byl jedním z požadavků na strukturu aplikace. Vyhledaná data jsou poté ukládána do working area, s kterým se pracuje v aplikační vrstvě.

```
SELECT * FROM cdhdr INTO WA where udate in
Date
```

```
and username in User
and utime in Time
and tcode in Trans_cd
and OBJECTCLAS in Ob_class
and OBJECTID in Ob_value
and CHANGENR in Doc_num
ORDER BY udate utime.
```



Obrázek 15: Screen vstupního okna aplikace

Tento SQL dotaz vyhledává data z tabulky CDHDR.

```
SELECT * FROM cdpos where changenr = wa-changenr
AND OBJECTCLAS = WA-OBJECTCLAS
AND OBJECTID = WA-OBJECTID.
```

Druhý SQL dotaz přidá k datům tabulky CDHDR data tabulky CDPOS.

Jelikož ve výsledné tabulce je nutné zobrazit všechna vyhledaná data a jeden dokument o změně v tabulce CDHDR na sobě může mít navázáno více dokumentů tabulky CDPOS, tak se některá data na výstupu liší pouze částí staré a nové hodnoty.

Aby mohlo být s daty jednoduše pracováno, ukládají se výsledky hledání do WA (working area). Díky tomu je možné s daty jednoduše pracovat, což se uplatní při tvorbě dodatečných informací o počtu změn provedených jednotlivými uživateli, které jsou vyžadovány dle zadání.

Dalším důvodem pro rozdělení práce aplikace je možnost přidávat výpočetní kapacitu k aplikační vrstvě a schopnost určovat prioritu při jejím využívání různými uživateli. Díky tomu je možné aplikace používat i k jednorázovému zpracování nestandardního množství dat.

Dodatečné informace je nutné řešit dynamicky a data brát přímo z working area, aby se umožnilo zapsání všech uživatelů bez ohledu na to, kdy byli do systému přidáni. K tomuto účelu je vhodné použít interní tabulku. Interní tabulka je stejně jako working area uchovávána a zpracovávána v aplikační vrstvě.

Interní tabulku je možné v průběhu aktualizovat nebo do ní přidávat další řádky. Vložení řádku do tabulky je řešeno příkazem „append“ a aktualizaci řádku příkazem „modify“.(SAP documentation, 1996a)

Implementace tvorby interní tabulky byla provedena těmito příkazy:

```

LOOP AT it INTO gwa_user.
IF gwa_user-username = WA-username.
gwa_user-username = WA-username.
gwa_user-changes = gwa_user-changes + 1.
MODIFY it FROM gwa_user.
bool = 1.
ELSEIF gwa_user-username = ' '.
gwa_user-username = WA-username.
gwa_user-changes = 1.
MODIFY it FROM gwa_user.
bool = 1.
ENDIF.
ENDLOOP.
IF bool = 0.
gwa_user-username = WA-username.
gwa_user-changes = 1.
APPEND gwa_user TO it.
bool = 0.
ENDIF.

```

Při každém zápisu řádku do WA se provádí aktualizace interní tabulky. Prakticky mohou nastat pouze dvě možnosti. První možností je, že uživatel již v tabulce je, takže se pouze upraví hodnota proměnné `gwa_user-changes` zvýšením o jedna. Druhou možností je absence uživatele v interní tabulce, takže je do tabulky přidán. Jméno uživatele je uloženo v proměnné `WA-username`. Kód však obsahuje i třetí možnost, která nastane pouze na začátku běhu programu. Důvodem vytvoření interní tabulky ještě před začátkem vyhledávání příkazy:

```

gwa_user-username = ' '.
gwa_user-changes = 0.
APPEND gwa_user TO it.

```

Do proměnné `gwa_user-username` je uložen prázdný řetězec, který je podmínkou, aby nastala třetí možnost, kdy se přidá první uživatel do tabulky.

V kódu figuruje proměnná `bool`, která slouží jako jednoduchý indikátor, jestli v daném kroku jedna z možností již byla uplatněna. V interní tabulce také figuruje proměnná `Number`, ve které je uložen celkový počet změn. Implementace tohoto výpočtu je dána jednořádkovým příkazem uvnitř SQL dotazu:

```

Number = Number + 1.

```

5.5 Tvorba výstupu

Výsledek vyhledávání je zobrazen na výstupu v podobě tabulky s daty uložené ve WA a dodatečnými informacemi o celkovém počtu změn a změn za každého uživatele.

Stejně jako vstupní část. spadá tvorba výstupu také do uživatelské vrstvy. Použitím správných nástrojů k tvorbě grafického rozhraní výstupu se umožní jednodušší export dat

pro případné zálohování nebo další zpracování. Výstupy aplikací v SAPu jsou standardně schopny exportovat data do programů jako je například excel.

Pro tvorbu tabulek jsou v programu ABAP interaktivní příkazy vykreslující horizontální linku SY-VLINE a vertikální linku ULINE. Pro umístění linek na požadované místo se uvádí u horizontální linky číselná hodnota, která určuje pozici linky na řádku. Linka se vykresluje pouze na délku jednoho řádku. Pro horizontální linku se uvádí její délka. Hlavička tabulky je tedy tvořena příkazy:

```
ULINE AT /(230)
```

Pro vytvoření vrchní vertikální linky:

```
WRITE: / SY-VLINE, 'Change doc. obj.',
30 SY-VLINE, 31 'Object value',
80 SY-VLINE, 81 'Doc. number',
94 SY-VLINE, 95 'User',
106 SY-VLINE,107 'Date',
120 SY-VLINE,121 'Time',
135 SY-VLINE, 136 'Transaction code',
160 SY-VLINE, 161 'Old value',
195 SY-VLINE, 196 'New value',
230 SY-VLINE.
```

Poté následuje tělo tabulky vypsání příkazy:

```
WRITE: / SY-VLINE, WA-OBJECTCLAS,
30 SY-VLINE, WA-OBJECTID,
80 SY-VLINE, WA-CHANGENR,
94 SY-VLINE, WA-username,
106 SY-VLINE, WA-udate,
120 SY-VLINE, WA-utime,
135 SY-VLINE, WA-tcode,
160 SY-VLINE, 161 cdpos-value_old,
195 SY-VLINE, 196 cdpos-value_new,
230 SY-VLINE.
```

Jako poslední jsou uvedeny dodatečné informace provedených změn:

```
WRITE: 'Number of founded lines is:', Number.
WRITE:/ 'Numbers of changes per user: '.
LOOP AT it INTO gwa_user.
WRITE:/ gwa_user-username, gwa_user-changes.
ENDLOOP.
```

Pro vypsání celkového počtu provedených změn, je jednoduše vypsána již zmiňovaná proměnná number, kterou následuje cyklus vypisující celou interní tabulku.

Na obr. 16 je vidět rozmístění dat do tabulky, do které je zapisován výstup vyhledávání aplikace uložený ve working area. Pod tabulkou jsou vypisovány dodatečné informace, které jsou zapisovány v interní tabulce.

Change doc. object	Object value	Document number	User	Date	Time	Transaction	Old value	New value

Number of founded lines is: X

Numbers of changes per user:

Uživatel1: X
 Uživatel2: X
 UživatelN: X

Obrázek 16: Model zobrazení výstupu aplikace

5.6 Nasazení vytvořené aplikace

Během prvních testů aplikace ve vývojovém systému byl upravován pouze návrh zobrazení dat pro větší přehlednost a správné zobrazení dat.

Po vytvoření aplikace došlo k jejímu přeřazení na testovací server, kde došlo k jejímu testování několika uživateli. Protože je aplikace určena zejména pro potřeby IT oddělení, nebylo zapotřebí testování velkým počtem uživatelů. Aplikace při práci na menším počtu dat pracovala dostatečně rychle a vzhledem k určení jejího využití by se neměla, při správně zadaných kritériích, dostat do situace přílišného vytěžování systému.

Po nasazení do produktivního systému, aplikace zvládá pracovat bez chyb a s dostatečnou rychlostí zobrazuje výsledná data.

6 Diskuse

Před zavedením aplikace se informace ohledně změn v kmenových datech vyhledávali pomocí transakcí, které zpřístupňují tabulky v databázi, nicméně jejich propojování už neslo složitější úpravy, nebylo ošetřováno proti dotazům, které mohly příliš zatěžovat systém a jakákoliv další práce s daty vyžadovala export dat a následné zpracování nebo vytvoření složitějšího SQL dotazu, který ovšem byl vykonáván nad databázovou vrstvou. Vytvořením aplikace vznikl rychlý nástroj pro vyhledání požadovaných dat a také základ, na který se dají navazovat další zpracování dat dle potřeb, které se vyskytnou.

Jiné návrhy aplikace, které by sice generovali stejný výstup, ale neměli požadované rozčlenění práce s daty nad různými systémy, nebyly přijaty. Rozsáhlejší aplikace nabízející i větší portfolio možností, které přesně nesplňují požadovaný způsob tvorby, nejsou pro firmu přijatelné.

7 Závěr

Na základě analýzy prostředí systému byl pro implementaci aplikace vybrán programovací jazyk ABAP, který podporuje SQL dotazy použité při získávání dat z databáze. Aplikace je možné vytvářet i pomocí programovacího jazyka Java, ale použití jazyka ABAP vyšlo na základě analýzy a konzultací s IT pracovníky podniku jako nejvhodnější řešení.

Pro návrh aplikace byly vybrány diagramy UML a ERD, které dostatečně detailně popsali funkcionalitu aplikace a možnosti jejího využití.

Implementace kódu musela splňovat standartní postupy práce s databází, které jsou v podniku používány. Důvodem je umožnění jednodušších modifikací aplikace a rozdělení kódu do částí, které pracují s databázovou, aplikační a uživatelskou vrstvou systému.

Po vytvoření aplikace bylo provedeno první testování již ve vývojovém systému. Testování proběhlo úspěšně, aplikace splňovala požadované parametry a fungovala dostatečně rychle vzhledem k množství zpracovávaných dat, takže nebyly nutné žádné optimalizace. Poté byla aplikace přesunuta do testovacího systému, kde se aplikace testují klíčovými uživateli a po úspěšných testech je možné aplikace přesunout do produktivního systému, kde se naplno využívají.

Pro aplikaci byl vytvořen manuál pro práci, který podrobně popisuje, jak aplikaci využívat. Význam aplikace je zejména ve dvou bodech. Prvním je zkoumání změn a to při zjišťování proč byly změny provedeny a jestli bylo nutné je provádět.

Druhý bod se týká zaměstnanců, kteří změny provádějí. Je totiž nutné monitorovat, jestli změny provedli skutečně zaměstnanci, kteří jsou určeni změny provádět, což zjednodušuje nalezení nevhodného přidělení práv pro výkon takové změny. Také je možné zjistit čas, kdy jsou změny prováděny a kontrolu, jestli je dostatečný počet zaměstnanců, kteří by se měli o provádění změn starat.

8 Reference

- ABAP for Newbies [online]. [cit. 2016-03-21]. Dostupné z: <http://scn.sap.com/docs/DOC-7940>.
- BASL, Josef a Roman BLAŽÍČEK. Podnikové informační systémy: podnik v informační společnosti. 3., aktualiz. a dopl. vyd. Praha: Grada, 2012. Management v informační společnosti. ISBN 978-80-247-4307-3.
- BOEDER, Jochen a Bernhard GROENE. The Architecture of SAP ERP: Understand how successful software works. 2014. ISBN 3849576620.
- CDHDR SAP Table - Change document header. SAP Tables [online]. [cit. 2016-03-22]. Dostupné z: <http://www.se80.co.uk/saptables/c/cdpo/cdpos.htm>.
- CDPOS SAP Table - Change document header. SAP Tables [online]. [cit. 2016-03-22]. Dostupné z: <http://www.se80.co.uk/saptables/c/cdhd/cdhdr.htm>.
- DOLINSKAJA, Olga. Get Started with the ABAP Development Tools for SAP NetWeaver [online]. In: . [cit. 2016-03-22]. Dostupné z: <http://scn.sap.com/docs/DOC-29297>.
- DOUCEK, Petr. Informační systémy pro podporu manažerské práce. 1. vyd. Překlad Michaela Škultéty. Praha: Professional Publishing, 2005. ISBN 80-864-1979-7.
- HEINEKEN. Heineken Ceska republika 2015 - SAP.pptx. 2015.
- HOEK, Willem. Change SAP documents -- tables CDHDR and CDPOS [online]. In: . [cit. 2016-03-22]. Dostupné z: <http://www.sap123.com/a/47/>.
- KALUŽA, Jindřich a Ludmila KALUŽOVÁ. Modelování dat v informačních systémech. 1. vyd. Praha: Ekopress, 2012. ISBN 978-80-86929-81-1.
- KANISOVÁ, H. -- MÜLLER, M. UML srozumitelně. 2. vyd. Brno: Computer Press, 2006. 176 s. ISBN 80-251-1083-4.
- KOLEKTIV AUTORŮ. 101 klíčových pojmů z oblasti informačních technologií. 2., aktualiz. vyd. Překlad Michaela Škultéty. Praha: SAP ČR, 2005.
- MANAGEMENT MANIA. Kmenová data (Master Data) [online]. [cit. 2016-04-02]. Dostupné z: <https://managementmania.com/cs/kmenova-data-master-data.pdf>.
- MIČÁNEK, Alan. Zadání aplikace. [cit. 2016-04-02].
- RÁBOVÁ, I. Podnikové informační systémy a technologie jejich vývoje. 1. vyd. Brno: Tribun EU, 2008. 139 s. ISBN 978-80-7399-599-7.
- SAP DOCUMENTATION. ABAP/4 User's Guide Basics -- System R/3. 1996. ISBN B000JP8PV2.

SAP DOCUMENTATION. ABAP/4 User's Guide, Reports Transactions -- System R/3. 1996. ISBN B000G7GMEU.

SCHMULLER, Joseph. Myslíme v jazyku UML. 1. vyd. Praha: Grada, 2001. Knihovna programátora (Grada). ISBN 80-247-0029-8.

9 Manuál pro práci s aplikací A

Aplikace slouží pro tvorbu reportu seznamu změn v kmenových datech. Data je možné vyhledávat podle polí username, utime, tdate, objectclass, objectid, changenr a tcode, obsažených v tabulce CDHDR.

Hodnoty lze zadávat do levého sloupce pro hledání jen pod určitou hodnotou nebo do obou sloupců. Potom hledání odpovídá jakákoliv hodnota spadající do intervalu mezi levým a pravým sloupcem. Pro úspěšné vyhledávání stačí vyplnit pole s datem, všechna další pole jsou nepovinná.

List of changes			
Change document object	<input type="text"/>	to	<input type="text"/>
Object value	<input type="text"/>	to	<input type="text"/>
Document number	<input type="text"/>	to	<input type="text"/>
User	<input type="text"/>	to	<input type="text"/>
Date	<input checked="" type="checkbox"/>	to	<input type="text"/>
Time	00:00:00	to	00:00:00
Transaction	<input type="text"/>	to	<input type="text"/>

Obrázek 17: První hodnota pro zadání kritérií

Do řádku vyznačeným červeným obdelníkem na obr. 17 lze zadat hodnotu, která bude vyhledávána ve sloupci tabulky CDHDR objectclass.

List of changes			
Change document object	<input type="text"/>	to	<input type="text"/>
Object value	<input type="text"/>	to	<input type="text"/>
Document number	<input type="text"/>	to	<input type="text"/>
User	<input type="text"/>	to	<input type="text"/>
Date	<input checked="" type="checkbox"/>	to	<input type="text"/>
Time	00:00:00	to	00:00:00
Transaction	<input type="text"/>	to	<input type="text"/>

Obrázek 18: Druhá hodnota pro zadání kritérií

Do řádku vyznačeným červeným obdelníkem na obr. 18 lze zadat hodnotu, která bude vyhledávána ve sloupci tabulky CDHDR objectid.

Do řádku vyznačeným červeným obdelníkem na obr. 19 lze zadat hodnotu, která bude vyhledávána ve sloupci tabulky CDHDR changenr.

List of changes			
Change document object	<input type="text"/>	to	<input type="text"/>
Object value	<input type="text"/>	to	<input type="text"/>
Document number	<input type="text"/>	to	<input type="text"/>
User	<input type="text"/>	to	<input type="text"/>
Date	<input checked="" type="checkbox"/>	to	<input type="text"/>
Time	<input type="text" value="00:00:00"/>	to	<input type="text" value="00:00:00"/>
Transaction	<input type="text"/>	to	<input type="text"/>

Obrázek 19: Třetí hodnota pro zadání kritérií

List of changes			
Change document object	<input type="text"/>	to	<input type="text"/>
Object value	<input type="text"/>	to	<input type="text"/>
Document number	<input type="text"/>	to	<input type="text"/>
User	<input type="text"/>	to	<input type="text"/>
Date	<input checked="" type="checkbox"/>	to	<input type="text"/>
Time	<input type="text" value="00:00:00"/>	to	<input type="text" value="00:00:00"/>
Transaction	<input type="text"/>	to	<input type="text"/>

Obrázek 20: Čtvrtá hodnota pro zadání kritérií

Do řádku vyznačeným červeným obdelníkem na obr. 20 lze zadat hodnotu, která bude vyhledávána ve sloupci tabulky CDHDR username. Do tohoto pole se zadává uživatelské jméno uživatele, který změnu provedl.

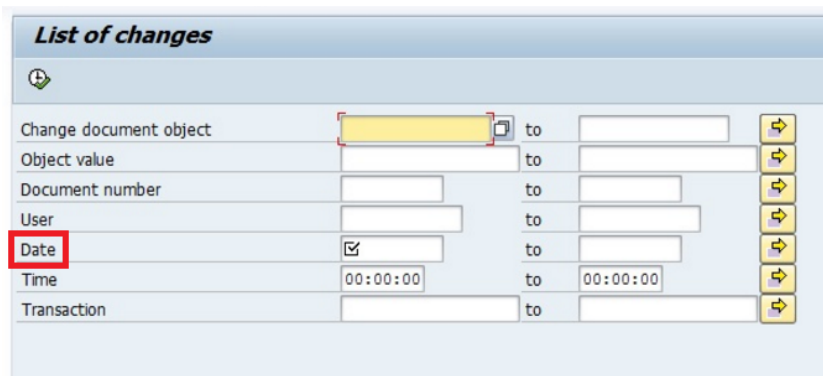
Do řádku vyznačeným červeným obdelníkem na obr. 21 lze zadat hodnotu, která bude vyhledávána ve sloupci tabulky CDHDR tdate. Toto pole je jedinná povinná hodnota, kterou je nutné zadat. Po kliknutí do pole se zobrazí kalendář, kde kliknutím vyberete datum.

Do řádku vyznačeným červeným obdelníkem 22 lze zadat hodnotu, která bude vyhledávána ve sloupci tabulky CDHDR ttime. Zadáním této hodnoty určíte čas, během kterého měla být změna provedena. Zadání hodnoty má smysl pouze v intervalové podobě. Po kliknutí do pole se zobrazí počítadlo hodin, kde kliknutím vyberete čas.

Do řádku vyznačeným červeným obdelníkem 23 lze zadat hodnotu, která bude vyhledávána ve sloupci tabulky CDHDR tcode. Tohle pole se zadává v případě, kdy je hledána změna provedená jistou transakcí.

Stisknutím červeně vyznačeného tlačítka 24 se spustí vyhledávání.

Po provedení vyhledání se zobrazí výsledná tabulka, pod kterou bude uveden počet nalezených změn a všichni uživatelé, kteří změny prováděli s rozpočítanými změnami na



List of changes

Change document object to

Object value to

Document number to

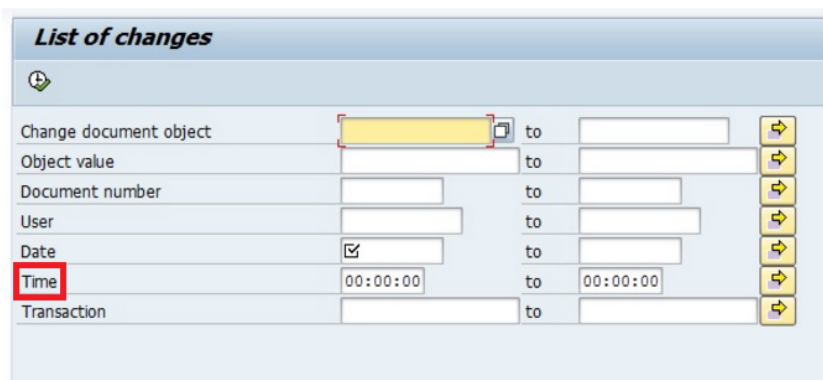
User to

Date to

Time to

Transaction to

Obrázek 21: Pátá hodnota pro zadání kritérií



List of changes

Change document object to

Object value to

Document number to

User to

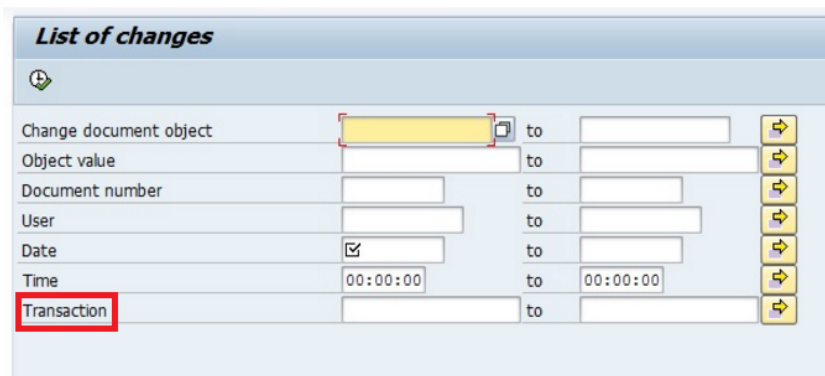
Date to

Time to

Transaction to

Obrázek 22: Šestá hodnota pro zadání kritérií

každého uživatele.

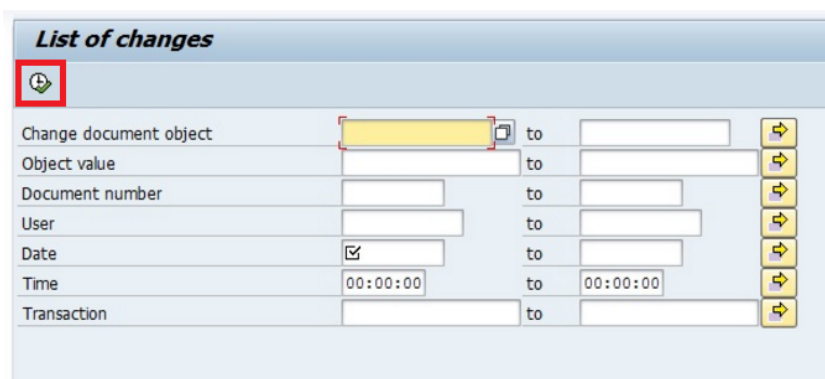


List of changes

⏴

Change document object	<input type="text"/>	to	<input type="text"/>	↕
Object value	<input type="text"/>	to	<input type="text"/>	↕
Document number	<input type="text"/>	to	<input type="text"/>	↕
User	<input type="text"/>	to	<input type="text"/>	↕
Date	<input checked="" type="checkbox"/>	to	<input type="text"/>	↕
Time	00:00:00	to	00:00:00	↕
Transaction	<input type="text"/>	to	<input type="text"/>	↕

Obrázek 23: Sedmá hodnota pro zadání kritérií



List of changes

⏴

Change document object	<input type="text"/>	to	<input type="text"/>	↕
Object value	<input type="text"/>	to	<input type="text"/>	↕
Document number	<input type="text"/>	to	<input type="text"/>	↕
User	<input type="text"/>	to	<input type="text"/>	↕
Date	<input checked="" type="checkbox"/>	to	<input type="text"/>	↕
Time	00:00:00	to	00:00:00	↕
Transaction	<input type="text"/>	to	<input type="text"/>	↕

Obrázek 24: Tlačítko ze spuštění vyhledávání