

MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY



Analýza systému
pro obchodování na internetu

BAKALÁRSKA PRÁCA

Brno, 2005

Tomáš Ludík



Masarykova universita v Brně
Fakulta informatiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Datum: 20.4.2005

Student: Tomáš Ludík

Vedoucí: RNDr. Jaroslav Ráček, Ph.D.

Katedra: Katedra programových systémů a komunikací

Studijní obor: Informatika

Specializace: Databáze

Garant spec.: prof. Ing. Pavel Zezula, CSc.

Název: Analýza systému pro obchodování na internetu

Zadání:

Seznámit se s problematikou strukturované analýzy informačních systémů. Zaměřit se zejména na metodiky SASS (DeMarco) a YMSA (Yourdon). V prostředí systému CASE Studio 2 provést analýzu informačního systému internetového obchodu oběma metodikami. Na základě získaných výsledků a praktických zkušeností vzájemně porovnat SASS a YMSA. Dále posoudit vhodnost použití nástroje CASE Studio 2 pro obě metodiky.

Základní literatura:

- Kung, Solvberg: Information Systeme Engineering
- Král: Informační systémy
- Richta, Sochor: Softwarové inženýrství I.
- Dokumentace CASE Studio 2

Požadovaná forma realizace:

- Tištěná verze práce
- Elektronická verze práce na CD
- Modely analyzovaného systému (dle SASS, YMSA)

student, datum

vedoucí BP

vedoucí katedry

garant specializace

Prehlásenie

Prehlasujem, že túto prácu som vypracoval samostatne s použitím uvedenej literatúry.

V Brne 20. apríla 2005

Tomáš Ludík

Pod'akovanie

Ďakujem vedúcemu mojej bakalárskej práce RNDr. Jaroslavu Ráčkovi, Ph.D. za vedenie a ústretový prístup, ktorý mi v priebehu písania tohto projektu poskytoval.

Ďalej by som chcel poďakovať mojim rodičom a priateľom, ktorý ma neustále podporovali a povzbudzovali, čím mi vytvorili vynikajúce podmienky nielen k písaniu tejto práce, ale aj počas celého štúdia na Masarykovej univerzite v Brne.

Zhrnutie

Práca zoznamuje čitateľa s modelmi systémov, ktoré sa využívajú pri štruktúrovanej analýze. Pomocou metód YMSA a SASS som spravil analýzu systému pre obchodovanie na internete. Následne som tieto dve metódy porovnal a popísal ich jednotlivé výhody a nevýhody. Pre tvorbu modelov som použil nástroj CASE Studio 2.

Kľúčové slová

Štruktúrovaná analýza, SASS - Structured Analysis and System Specification, YMSA - Yourdon Modern Structured Analysis, DFD - Data Flow Diagram, ERD - Entity Relationship Diagram, CASE Studio 2.

Obsah

1.	Úvod	7
2.	Analýza systému	8
3.	Nástroje analýzy	9
3.1	Účel systému	9
3.2	Diagramy dátových tokov (DFD)	9
3.3	Modely vonkajšieho chovania systému	10
3.4	ERD	10
3.5	Dátový slovník	11
4.	Štruktúrovaná analýza	13
4.1	SASS (Structured Analysis and System Specification)	14
4.2	YMSA (Yourdon Modern Structured Analysis)	15
5.	Analýza systému pre obchodovanie na internete	18
5.1	Analýza systému pomocou metódy SASS	18
5.2	Analýza systému pomocou metódy YMSA	21
6.	Vyhodnotenie	25
6.1	Porovnanie metód SASS a YMSA	25
6.2	Zhodnotenie vhodnosti nástroja CASE Studio 2 pre obidve metódy analýzy	25
7.	Záver	26
8.	Literatúra	27
9.	Internetové zdroje	28
	Príloha A: Štruktúra adresárov na CD	29

1. Úvod

V dnešnej spoločnosti je v popredí internet. Je to celosvetové spojenie počítačových sietí. Bežný užívateľ tu nachádza obrovské množstvo informácií a služieb. Jednou z týchto služieb je obchodovanie na internete. Užívateľ si zo svojho domáceho prostredia objedná ľubovoľný produkt a ten mu je doručený až domov. Pre túto skutočnosť sa obchodovanie na internete stáva populárne. Preto vznikajú požiadavky od obchodíkov na tvorbu špecifického systému. Tento systém je tvorený na zákazku a má spĺňať všetky požiadavky, ktoré obchodník pre svoj internetový obchod potrebuje. Základom úspechu je porozumenie potrebám a požiadavkám obchodníka, čo je cieľom analýzy príslušného systému.

Preto si táto práca zvolila za cieľ analyzovať systém pre obchodovanie na internete. Pri analýze sú použité najrozšírenejšie techniky štruktúrovanej analýzy. Systém bude modelovaný pomocou štruktúrovanej analýzy SASS a YMSA. Výsledkom však nemajú byť len jednotlivé analýzy, ale aj ich porovnanie, zistenie výhod a nevýhod jednotlivých postupov a následne určiť vhodnejšiu techniku pre štruktúrovanú analýzu.

Pri oboch analýzach bude použitý softwarový produkt Case Studio 2 od spoločnosti Charonware. Tento produkt bol zvolený práve preto, lebo je vhodný pre tvorbu diagramov štruktúrovanej analýzy SASS a YMSA. Ďalej sa jedná o jediný produkt tohto charakteru, ktorý je tvorený českou spoločnosťou a je lokalizovaný do českého i anglického jazyka. Ďalej je tento produkt aj viac cenovo prístupnejší ako iné nástroje podobného charakteru.

2. Analýza systému

Analýza znamená štúdium problému pred tým, než podnikneme nejaké akcie smerujúce k jeho riešeniu. Za riešenie analyzovaného problému považujeme i odôvodnené negatívne stanovisko a odmietnutie realizácie nového systému. Akciu, ktorú podnikneme na základe analýzy, väčšinou vedie k implementácii nového systému. Najčastejším predmetom analýzy je existujúci systém, ktorého štruktúra a funkcie nie sú zrejmé zákazníkovi ani riešiteľovi, prípadne zákazník nevie štruktúru a chovanie systému riešiteľovi vysvetliť. Alebo neexistujúci systém, o ktorom má zákazník nepresnú predstavu a nevie požadované funkcie riešiteľovi vysvetliť.

Analytik vo veľkej väčšine prípadov realizuje tieto neznáme a málo formulované systémy. Produktom analýzy je špecifikačný dokument. Pri tvorbe špecifikačného dokumentu je analytik ovplyvnený individuálnymi predstavami užívateľov, objektívnymi požiadavkami na systém a vlastným pohľadom na uvedenú aplikačnú oblasť.

Z týchto často rozporných pohľadov na systém musí vytvoriť dokument, ktorý bude úplným podkladom pre návrh funkčného riešenia. Špecifikačný dokument má často i právnu platnosť, lebo sa stáva súčasťou obchodnej zmluvy ako doklad, ktorý presne definuje predmet a rozsah zákazky. Špecifikačný dokument stanovuje cieľ riešenia, požadovaný výsledok, podrobne dokumentovaný cieľový stav v takej podobe, aby bolo možné posúdiť, či implementácia uvedeného stavu dosiahla a ako posledné dôležité sprievodné parametre spojené s riešením a údržbou.

Špecifikačný dokument je závažným podkladom pre návrh a realizáciu systému. Výsledok riešenia je zachytený vo forme technickej a užívateľskej dokumentácie. Nový systém často nahrádza súčasný systém. Pri náhrade požadujeme, aby nový systém splnil dve základné požiadavky. Prvou požiadavkou je, aby systém porozumel všetkým relevantným vonkajším podnetom (príkazom, správam), ktoré vnímal i predchádzajúci systém. A druhou požiadavkou je, aby systém zaručil rovnaké a nové požadované funkcie a chovanie, tak že bude mať zhodnú alebo dokonalejšiu odozvu na podnety z okolia.

Návrh nového systému najviac uľahčia dva druhy poznatkov. Znalosť prostredia, v ktorom bude systém pracovať, nám pomáha odhaliť, aké vplyvy môžu pôsobiť na systém. A znalosť štruktúry systému a funkčných elementárnych častí, čo nám pomáha pri poznaní a pochopení logiky tých funkcií systému, ktoré sú "viditeľné" v okolí systému.

Z predchádzajúcich úvah vyplýva, že pre pochopenie a návrh systému je vhodné nájsť alebo vytvoriť rozklad systému, ktorý sa čo najviac priblíži prirodzenej dekompozícii reálneho systému. Pri analýze budeme vnímať systémy ako sústavy, zložené z jednoduchších systémov a prvkov, ktoré majú určitú štruktúru a chovanie. Predmetom analýzy bude nájdenie rôznych dekompozícií zložitých systémov na jednoduchšie. Vytvorená dekompozícia (sada dekompozícií) bude základom špecifikácie nového systému.

3. Nástroje analýzy

Najväčšiu časť práce systémového analytika je v tvorbe rôznych modelov systému. Modely slúžia k tomu, aby sa analytik zamerlal na dôležité rysy systému a potlačil menej významné rysy. Diskutoval zmeny a opravy v požiadavkách užívateľa pri minimálnych nákladoch a minimálnom riziku a ukázal, že správne rozumie užívateľskému prostrediu. A svoje poznanie dokumentoval takým spôsobom, že návrhári a programátori môžu z príslušných podkladov vytvoriť systém.

Užívateľ i analytik však môže systém vnímať z niekoľkých rôznych pohľadov ako je *Funkčný model systému*, ktorý chápeme predovšetkým ako množinu funkcií, ktoré transformujú dáta zadaným spôsobom. Alebo *Model vonkajšieho chovania systému*, kde systém pre vonkajšieho pozorovateľa predstavuje čiernu skrinku, ktorá na určité udalosti reaguje požadovaným spôsobom. Posledným pohľadom je *Dátový model systému*. Tu systém považujeme za prostriedok pre uloženie a opätovné získanie (transformovanej) informácie [RiSo96].

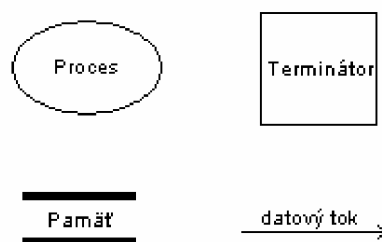
3.1 Účel systému

Východiskom každej analýzy systému je zreteľne sformulovaný *Účel systému*. Ide o krátky text, ktorý vystihuje, k akému účelu má slúžiť vytváraný systém. Pre analytika bude tento model predovšetkým zdrojom inšpirácie pre formuláciu mnohých ďalších otázok.

3.2 Diagramy dátových tokov (DFD)

Diagramy dátových tokov patria medzi najpoužívannejšie modelovacie nástroje štruktúrovanej analýzy. Pomocou nich sa modeluje funkcionality systému. DeMarco definuje DFD takto: *Diagram dátových tokov je sieťovou reprezentáciou systému. Systém môže byť automatizovaný, manuálny alebo zmiešaný. DFD znázorňuje systém pomocou jeho komponentov a určuje rozhranie medzi komponentmi* [DeM79]. DFD ukazuje transformácie dátových tokov, ktoré vstupujú do funkčných jednotiek, na výstupné dátové toky. Jednoduché DFD s vhodne pomenovanými komponentmi sú bez špeciálneho školenia zrozumiteľné i pre tých užívateľov, ktorí síce nemajú znalosti z informatiky, ale poznajú problematiku modelovaného systému.

DFD diagramy sa skladajú zo 4 komponentov: *procesy, terminátory, pamäte a dátové toky*. *Proces* predstavuje tú časť systému, ktorá transformuje dáta alebo mení nejaké vstupy a výstupy. *Terminátor* reprezentuje externú entitu, s ktorou systém komunikuje a je zdrojom alebo príjemcom všetkých informácií, ktoré do systému vstupujú alebo z neho vystupujú. *Pamäť* je pasívnym prvkom systému, ktorý slúži pre uloženie dát, za účelom neskoršieho spracovania. *Dátové toky* znázorňujú cestu, po ktorých sa informácie v systéme pohybujú. Tieto komponenty sú znázornené na obrázku 3.1 na nasledujúcej strane.



Obrázok 3.1: Komponenty DFD diagramu (notácia CASE Studio 2)

3.3 Modely vonkajšieho chovania systému

Kontextový diagram:

Pri formulácii vonkajšieho chovania systému je nutné jednoznačne vymedziť hranicu systému a entity, s ktorými systém komunikuje. Kontextový diagram je špeciálnym prípadom DFD, kde sa systém predstavuje ako jediný proces. Kontextový diagram zdôrazňuje nasledujúce rysy systému: kto so systémom komunikuje (ľudia, organizácie, iné systémy), dáta, ktoré do systému vstupujú z okolitého sveta a ktoré majú byť systémom nejakým spôsobom spracovávané (transformované) a dátové pamäte, ktoré sú zdieľané so systémom a terminátormi. Tieto pamäte buď vytvoril vonkajší svet a systém ich používa, alebo je pamäť produktom systému a používa ju vonkajší svet a hranice medzi systémom a okolitým svetom.

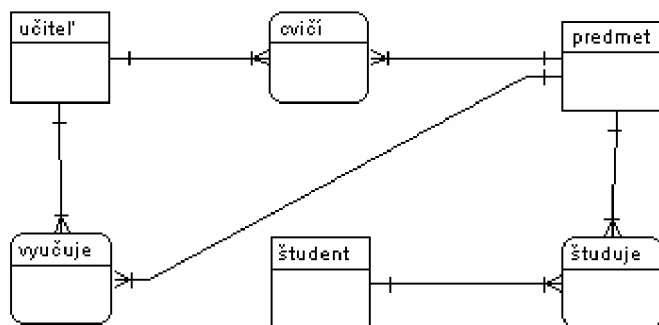
Zoznam udalostí:

Zoznam udalostí, používaný v štruktúrovanej analýze YMSA, je tvorený súpisom rôznych podnetov, ktoré pôsobia na systém z jeho okolia. Každá udalosť môže byť podrobnejšie klasifikovaná podľa toho, aký má charakter. F označuje udalosť, ktorá je spojená so vstupom dát do systému (Flow). T označuje časovú udalosť (Temporal) a C udalosť príkazového typu (Control).

Podrobnejšie rozoberieme iba *F-udalosť*, ktorá je charakterizovaná vstúpením dát do systému. Za *F-udalosť* však nepovažujeme odvodené vstupy dát. Je zrejmé, že medzi zoznamom udalostí a dátovými tokmi kontextového diagramu nie je jednoznačná korešpondencia. Dá sa však povedať, že každý vstupný dátový tok buď indikuje výskyt udalosti, alebo je vyžiadaný systémom, aby bolo možné nejakú udalosť spracovať.

3.4 ERD

Modelujú dáta a statický pohľad na systém. Preto pri tvorbe dátového modelu si najprv kladieme otázku, ktoré údaje musia byť v systéme uchované. Objekty, ktoré sú charakterizované skupinou k sebe prislúchajúcich údajov, sa nazývajú dátové entity. Pre grafické vyjadrenie dátového modelu sa najčastejšie používajú ER diagramy (entity-relationship, ERD). Znázorňujú dátové objekty - entitné množiny a ich vzájomné vzťahy. Zatiaľ čo DFD ukazuje súčasné procesy i dátové entity (vo forme pamätí), ERD sa zameriava iba na dátové entity. Príklad ERD si môžete pozrieť na obrázku 3.2 zobrazenom na nasledujúcej strane.



Obrázok 3.2: Príklad ERD (notácia CASE Studio 2)

Popis vzťahu medzi entitami:

Pri každej dvojici nájdených entít skúmame vzájomné vzťahy. Významnou charakteristikou každého vzťahu je jeho arita. Charakterizuje počet entít, ktoré vstupujú do daného vzťahu. Najčastejšie sa rozlišujú vzťahy $1:1$, $1:M$ (veľa), $N:M$. Medzi entitami môže existovať viacej vzťahov, ktoré môžu byť voliteľné alebo povinné. Voliteľné vzťahy sú také, ktoré nemusia nastať a sú typu $1:0..N$.

3.5 Dátový slovník

Všetky informácie uvedené v popise systému sú navzájom odlišné svojou identifikáciou. Pre jednoduché hľadanie je užitočný významový slovník všetkých identifikátorov a kľúčových spojení, vyskytujúcich sa v popise systému. Budeme ju tradične označovať termínom dátový slovník (data dictionary). Popisované dáta môžu byť elementárne alebo zložené. Elementárne položky predstavujú známe typy dát, je u nich nutné zadať prípustný obor hodnôt a prípadne použité jednotky. Zložené dáta sa dajú popísať odkazom na jednotlivé zložky. Zložky sa tiež môžu vyskytovať voliteľne alebo opakovane. Definícia notácie dátového slovníku nemusí byť preto zložitá. Väčšinou sa jedná o niektorú variantu regulárnej gramatiky. V [You89] je uvedená notácia (neterminály sú zapísané pomocou veľkých písmen):

Symbol	Význam	Príklad	
=	skladá sa z	$X = Y$	X sa skladá z Y
+	a	$Z = X + Y$	Z sa skladá z X a Y
()	môže chýbať	$Z = X + (Y)$	Z sa skladá z X príp. z Y
{ }	opakovanie	$Z = \{X\}$	Z sa skladá z niekoľkých X
[]	jeden z možných	$Z = [X Y]$	Z sa skladá z X alebo z Y
**	komentár	*toto je komentár*	
@<cislo>	kľúčová položka		
@	časť zloženého kľúča		
	oddeľovač variant v []		

Počet opakovaní je všeobecne ľubovoľný, vrátane žiadneho. Pokiaľ je nutné počet opakovaní limitovať, môžeme prípadnú dolnú hranicu zapísať pred zložené zátvorky a prípadnú hornú hranicu za zátvorky. Prípustné kombinácie sú:

$X = \{Y\}$	X sa skladá z ľubovoľného počtu Y, vrátane žiadneho
$X = 1 \{Y\}$	X sa skladá z najmenej jedného Y
$X = \{Y\} 1$	X sa skladá zo žiadneho alebo najviac z jedného Y
$X = 1 \{Y\} 10$	X sa skladá z jedného až desiatich Y

4. Štruktúrovaná analýza

Pomocou náhodne zvolených modelov rôzneho druhu ide vytvoriť veľmi rozsiahly a nečitateľný špecifikačný dokument. V takomto dokumente ide len ťažko nájsť chyby a preveriť úplnosť špecifikácie a návrhu.

Pre rozvoj softwarového inžinierstva zohrala významnú úlohu klasická práca Tom DeMarco „Štruktúrovaná analýza a špecifikácia systému“ [DeM78]. Autor sa v tejto práci zameriava na analýzu informačných systémov a na použitie metódy štyroch modelov s cieľom vytvoriť prehľadnú, vhodne štruktúrovanú dokumentáciu systému. Aj keď bola metóda štyroch modelov v 80. rokoch podrobená kritike a upravovaná, hlavné myšlienky, ktoré sa týkali štruktúrovaného prístupu, sa uplatňujú dodnes vo všetkých moderných analytických metódach. DeMarco nazval svoju metódu *Štruktúrovaná analýza* a jej podstatu popisujem v nasledujúcich riadkoch.

Štruktúrovaná analýza slúži k riešeniu nových cieľov. Sú to: Produkty analýzy, ktoré musia byť veľmi dobre udržiavateľné. Toto platí predovšetkým pre Cieľový dokument (definíciu riešeného systému). Veľké problémy musia byť efektívne rozdelené na menšie. Cieľový dokument vo forme románu je neúnosný. Všade, kde je to možné sa musí použiť grafické vyjadrenie (obrázkami). Je treba rozlíšiť logické a fyzické aspekty a podľa toho rozdeliť zodpovednosť medzi užívateľa a analytikov. Logický model je nutné vytvoriť preto, aby sa užívateľ mohol zoznámiť zo systémom ešte pred jeho implementáciou.

Štruktúrovaná analýza vyžaduje nástroje s nasledujúcimi charakteristikami: Nástroj musí pomôcť pri delení našich požiadaviek a musí dokumentovať toto delenie pred tým, než vytvoríme špecifikáciu (napíšeme Cieľový dokument). Musí poskytnúť prostriedky pre sledovanie rozhraní, ktoré sa objavia ako dôsledok uvedeného delenia a umožniť ich vyhodnotenie bez toho, aby bolo nutné popísať fyzickú podobu rozhraní. Nové nástroje musia umožniť popísanie logiky a zmysel jednotlivých častí systému.

Štruktúrované analytické metódy zdieľajú nasledujúce charakteristiky: rozdelenie projektu na malé dobre definované aktivity a určujú postupnosť a interakciu týchto aktivít a používajú diagramatické a ďalšie modelovacie techniky, pomocou ktorých je vytvorená presnejšia a úplnejšia (štruktúrovaná) špecifikácia, ktorej rozumejú súčasne užívatelia i návrhári systému.

Výhody a nevýhody Štruktúrovanej analýzy

Štruktúrovaná analýza formuluje všeobecne zrozumiteľné požiadavky na systém, ktoré sú pevným základom pre nasledujúci návrh a implementáciu. V praxi je veľmi ťažké zistiť, čo užívateľ od systému požaduje. Analytik nemôže položiť niekoľko málo otázok a z odpovedí hneď odvodiť súhrn požiadaviek. Štruktúrovaná analýza kladie otázky systematickým spôsobom. Pritom sa ale vyhýba technickým výrazom. Koncept systému vysvetľuje analytik budúcim užívateľom pomocou obrázkov a jednoduchých textov.

Efektívnejšie využitie skúsených a menej skúsených pracovníkov. Schopnosť pýtať sa správne otázky a sformulovať dobre projekt nemá každý z pracovníkov riešiteľského tímu. Štruktúrované metódy sa síce neobídu bez vysoko kvalifikovaných pracovníkov, umožňujú však špecializáciu odborných činností a realizáciu niektorých činností menej kvalifikovanými pracovníkmi.

Zlepšené plánovanie a riadenie. Rozdelenie projektu na jednotlivé etapy a jednotlivé kroky umožňuje dokonalejší odhad času a skoré riadiace zásahy a kontrolu výsledkov.

Zvýšenie kvality systému. Úplná štruktúrovaná špecifikácia ovplyvňuje pozitívne kvalitu systému. Budúci užívatelia systému spolupracujú tvorivým spôsobom na vývoji a overovaní kvality systému, štruktúrované metódy formulujú úlohy užívateľov a riešiteľov v jednotlivých etapách. Ešte pred vlastnou realizáciou systému je zaistená priebežná kontrola, že systém odpovedá predstavám užívateľa.

Naopak nevýhodami tohto postupu je ťažšie namodelovanie náročnejších oblastí. Z toho je zrejmé, že komunikácia medzi analytikom a užívateľom nie je vždy úplne jednoduchá a jednoznačná. Občas analytik nenavrhuje systém presne podľa užívateľových predstáv. Niekedy sa pri postupoch štruktúrovanej analýzy stáva, že výsledky analýzy nie sú úplne konzistentné. Jej ďalšou nevýhodou sú dva pohľady na systém. Ide o funkčný pohľad, ktorý je tvorený sadou DFD a pohľad dátový, pre ktorého znázornenie sa používa ERD v YMSA alebo dátový slovník v SASS. Ale aj napriek týmto nevýhodám sú postupy štruktúrovanej analýzy veľmi dobre prepracované a v praxi bez problémov použiteľné.

4.1 SASS (Structured Analysis and System Specification)

T. DeMarcovi je pripisované prvenstvo pri použití diagramov toku dát pre analýzu informačných systémov určených pre bežné obchodné a správne organizácie. Svoju metódu formuluje takto:

SASS je metóda, pri ktorej s použitím nástrojov

- Diagram toku dát
- Dátový slovník
- Štruktúrovaná angličtina
- Rozhodovacie tabuľky
- Rozhodovacie stromy

vytvoríme nový druh Cieľového dokumentu, tzv. *Štruktúrovanú špecifikáciu*.

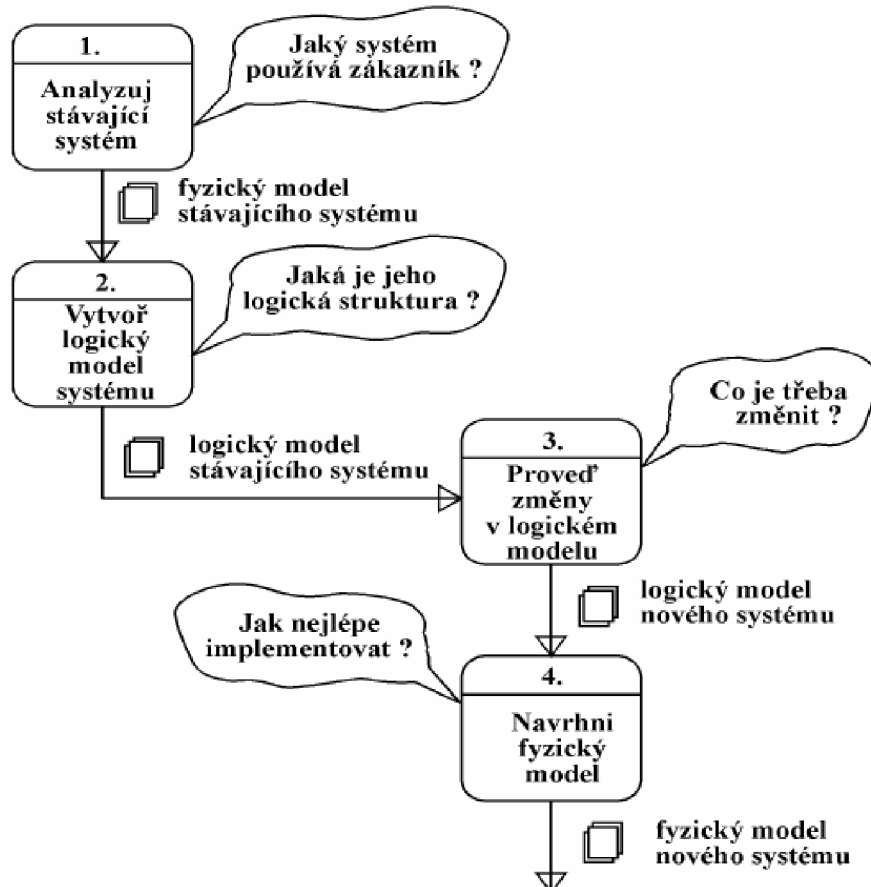
DeMarco zaradil Štruktúrovanú analýzu ako druhý proces životného cyklu systému. Vstupom pre štruktúrovanú analýzu je súhrn užívateľských požiadaviek, ktoré boli podrobené analýze prevedenia. Štruktúrovaná špecifikácia vytvorená DeMarcovou metódou je zložená zo série hierarchicky usporiadaných dokumentov. Systém je špecifikovaný pomocou niekoľkých diagramov dátových tokov, na ktorých sú uvedené podstatné procesy, pamäte a údaje, ktoré do procesu vstupujú a sú nimi transformované. Ak sú procesy v diagrame uvedené príliš komplikovane, je prevedená ich dekompozícia na skupinu jednoduchších procesov a pamätí a táto dekompozícia je zakreslená v diagrame nižšej úrovne. Logika (algoritmus) dostatočne jednoduchých procesov sa popisuje pomocou štruktúrovanej angličtiny, rozhodovacou tabuľkou alebo rozhodovacím stromom. Dátová časť systému je popísaná v dátovom slovníku, ktorý umožňuje popísať štruktúrované (zložené) údaje jednoduchým spôsobom.

DeMarco delí metódu na 4 etapy:

1. Štúdie súčasného fyzického prostredia. Výsledkom je dokumentácia súčasného stavu tvorená fyzickými DFD súčasného systému.

2. Odvodenie logického ekvivalentu fyzického systému. Výsledkom sú logické DFD súčasného systému.
3. Odvodenie nového logického systému. Výsledok dokumentuje sada logických DFD, popis transformácií (mini špecifikácie, rozhodovacie tabuľky alebo rozhodovacie stromy) a dátový slovník.
4. Tvorba a začlenenie nových fyzických DFD do štruktúrovanej špecifikácie spoločne s dátovým slovníkom a popisy procesov.

Etapy sú znázornené na obrázku 4.1:



Obrázok 4.1: Štyri etapy v analýze DeMarco [W1]

Súčasťou DeMarcovej metódy sú odporúčania týkajúce sa voľby mien procesov, tokov, pamätí, pravidla pre zostavenie a dekompozíciu DFD, pravidla pre zostavenie dátového slovníku a prechod od špecifikácie k implementácii.

4.2 YMSA (Yourdon Modern Structured Analysis)

V roku 1989 publikuje E. Yourdon knihu Moderná štruktúrovaná analýza [You89]. V knihe uvádza vlastné poznatky získané v priebehu dvoch desaťročí svojej práce na vývoji veľkých systémov a učenia metód štruktúrovanej analýzy. Metodológia publikovaná v tejto knihe sa sústreďuje na nájdenie *esenciálneho modelu systému*. Jedná sa o model, ktorý vyjadruje podstatu systému a nezávisí na technických a ďalších imple-

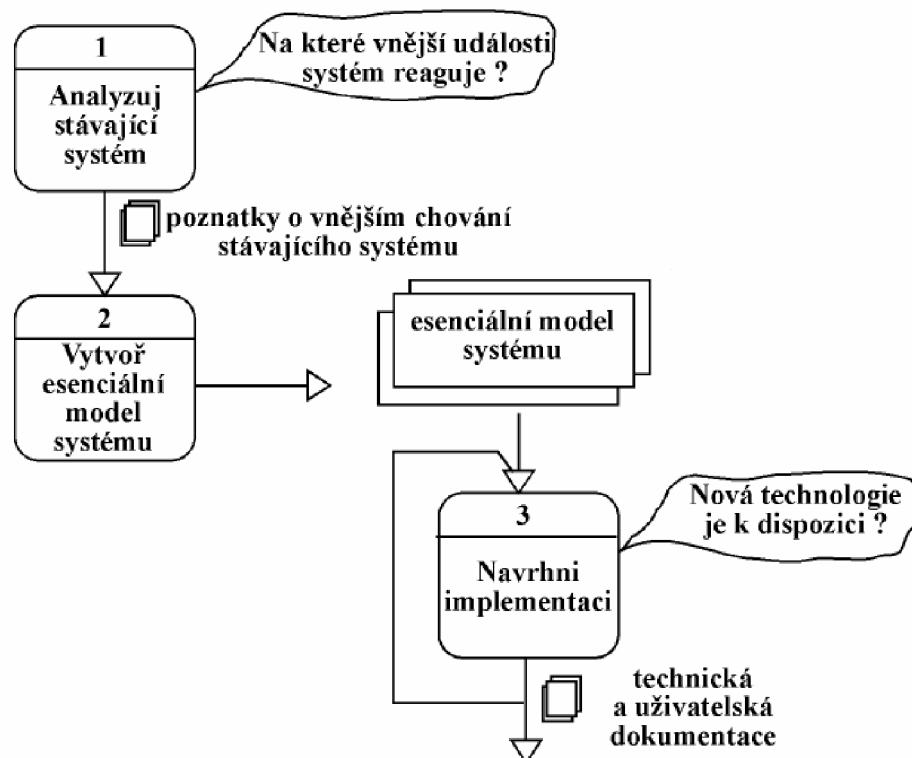
mentačných obmedzeniach. Esenciálny model má dve časti, *model okolia systému* a *model chovania systému*.

Pre oba modely sa používa nástroj DFD - model okolia systému známy pod názvom kontextový diagram. Model chovania systému obsahuje sadu hierarchicky usporiadaných DFD s procesmi, tokmi a pamäťami, ktoré zaisťujú podstatné činnosti systému. Yourdon konštatuje, že klasický dekompozičný postup metódou zvrchu-dole od kontextového diagramu k sade DFD často v praxi zlyháva. Vytvorené systémy sú nevhodne štruktúrované a ťažko udržiavateľné. Yourdonov prístup používa pre vytvorenie štruktúrovanej špecifikácie dekompozície na základe udalostí, ktorá vychádza z dopredu pripraveného zoznamu udalostí. Cieľom je vytvoriť DFD, ktorý je modelom chovania systému.

Metodológia YMSA:

1. Tvorba modelu okolia pomocou analýzy súčasného systému, ktorý sa skladá z
 - kontextového diagramu a
 - zoznamu udalostí.
2. Prvotný model chovania systému (esenciálny model systému).
3. Vyvažovanie modelu chovania a návrh implementácie systému.

Postup je znázornený na obrázku 4.2.



Obrázok 4.2: Postup v analýze YMSA [W1]

Identifikácia udalostí

Pre vytvorenie zoznamu udalostí navrhuje Yourdon použitie ďalších modelov ako je kontextový diagram a dátový model ERD. Uvádza dve možnosti:

Identifikácia udalostí pomocou entít:

Analýzu systému začneme vytvorením prvotného dátového modelu, v ktorom sú uvedené nájdené dátové entity. Všetky vzťahy nemusia byť ešte celkom zrejmé, do ERD zakreslíme zistené (odhadnuté) vzťahy. Pre každú entitu hľadáme vonkajšie udalosti, ktoré vedú k ich použitiu a zmene. Zo zistených poznatkov zostavíme predbežný zoznam udalostí. Na jeho základe vytvoríme kontextový diagram - model okolia systému. Trojicu modelov vzájomne preveríme a upravíme.

Identifikácia udalostí pomocou kontextového diagramu:

V tomto prípade zostavíme kontextový diagram, na ktorom vyznačíme terminátory a príslušné toky spájajúce systém s okolím. Skúmaním terminátorov a tokov odhadneme udalosti. Pre ich overenie opäť použijeme ďalšie modely (procesné i dátové).

Tvorba DFD

Pri tvorbe DFD sa postupuje tak, že sa vytvorí zoznam udalostí a pre každú udalosť zo zoznamu sa vytvorí jeden proces na DFD. Procesy sa pomenujú podľa odozvy, ktorá je požadovaná pre príslušnú udalosť. Ďalej je treba zakresliť potrebné vstupné toky, požadované výstupné toky a pamäte údajov, ktoré sú potrebné pre odozvu. Výsledný prvotný DFD sa porovná s kontextovým diagramom a so zoznamom udalostí a overí sa úplnosť a bezspornosť prvotného DFD.

Vyvažovanie DFD

Naznačený postup tvorby DFD je priamočiary a jednoduchý. Prvotný DFD je však neprehľadný, lebo obsahuje toľko procesov, koľko je vonkajších udalostí. V ďalších krokoch prebieha *vyvažovanie DFD na rôznych úrovniach*.

Vyvažovanie prebieha vo dvoch smeroch. Príbuzné procesy s (takmer) rovnakými vstupnými a výstupnými dátovými tokmi nahradíme v nadradenom DFD jedným procesom - DFD sa zjednoduší. Ak procesy pristupujú k pamäti, ktorú ostatné procesy nepoužívajú, tak zakryjeme na vyššej úrovni i túto pamäť. Tento postup sa nazýva *ukrývanie informácie*.

Niekedy je naopak nutné vyvážiť DFD smerom dole. Proces uvedený na danej úrovni DFD je príliš zložitý (napr. v porovnaní s ostatnými procesmi) a je vhodné spraviť dekompozíciu podľa formálnej prevedenej dekompozície, pri ktorej pre každý dátový tok pripravíme jeden proces na nižšej úrovni.

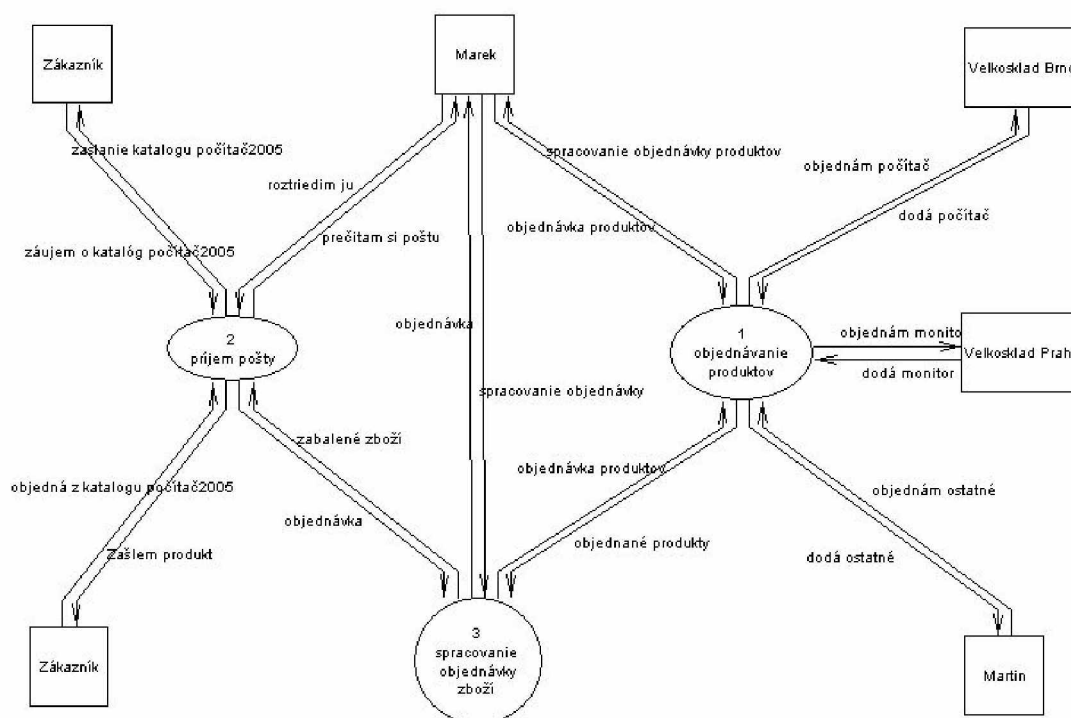
5. Analýza systému pre obchodovanie na internete

V tejto kapitole analyzujem systém pre obchodovanie na internete. Ide o jednoduchý internetový obchod, ktorý nahrádza klasické obchodovanie. Na tejto analýze ukážem postupy SASS a YMSA, ktoré sú popísané v nasledujúcich dvoch kapitolách.

5.1 Analýza systému pomocou metódy SASS

Tento spôsob analýzy systému je jeden z najstarších. Postupuje sa podľa SASS vid' kapitola 4.1. Analýza postupuje od fyzického modelu súčasného systému cez logický model. Tento model je vyvažovaný v smere z hora nadol. Ako posledná fáza tohto postupu je tvorba fyzického modelu nového systému, ale to už nespadá do tejto bakalárskej práce. A preto aj v našej analýze systému pre obchodovanie na internete začneme najskôr tvorbou fyzického modelu súčasného systému. Je znázornený na obrázku 5.1.

Fyzický model súčasného systému



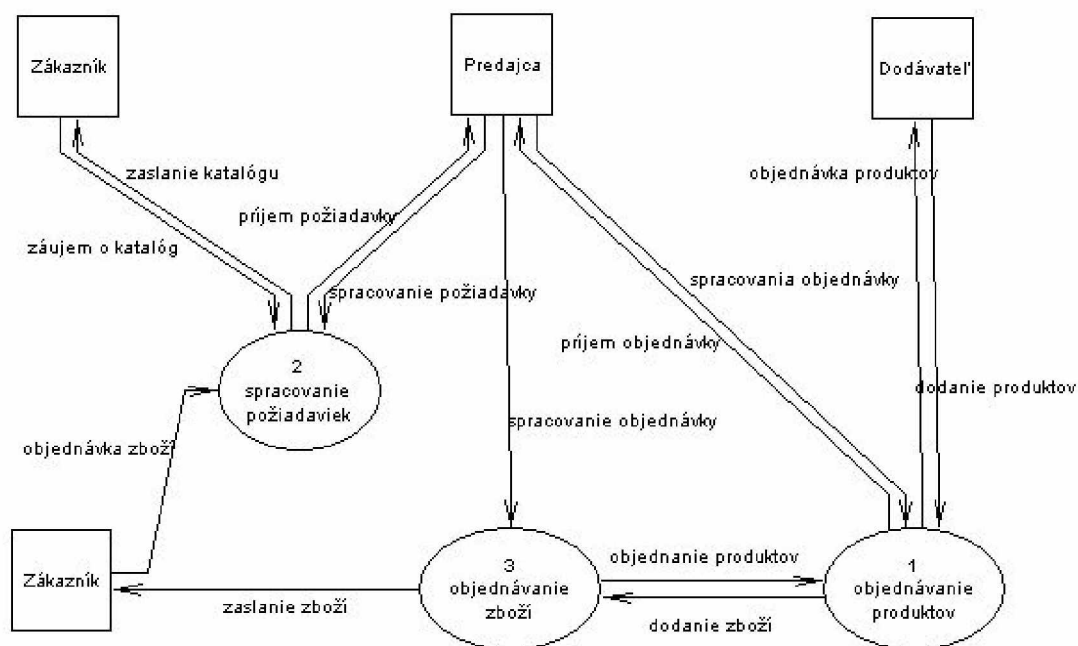
Obrázok 5.1: Fyzický model súčasného systému

Ďalším krokom analýzy SASS je vytvorenie logického modelu súčasného systému. Jeho tvorba spočíva v pretvorení už vytvoreného fyzického modelu súčasného systému. Je znázornený na obrázku 5.2.

Týmto máme za sebou fázu príprav na tvorbu nového systému a začíname tvoriť logický model súčasného systému. Ten sa skladá z dvoch častí. Prvou z nich je vytvorenie dátového slovníku. Ten slúži pre lepšiu analýzu nami tvoreného systému. Popisuje všetky identifikátory a kľúčové spojenia vyskytujúce sa v analyzovanom systéme. Pre jeho rozsah ho v tejto práci nebudem spomínať celý, ale pre ukážku uvediem dve

entity. Ich charakterizácia je uvedená pod obrázkom 5.2. Druhou časťou je vytvoriť sadu DFD diagramov, ktoré budú popisovať celý logický model budúceho systému. Pri tvorení týchto modelov postupujeme pri analýze smerom zhora dole. Na obrázku 5.3 máme prvú úroveň týchto diagramov. Jednotlivé procesy postupne dekomponujeme až pomocou tohto postupu máme kompletný návrh budúceho systému pre obchodovanie na internete. Ďalšia dekompozícia DFD už v tejto práci nebude znázornená, ale môžeme ju nájsť na priloženom CD.

Logický model súčasného systému



Obrázok 5.2: Logický model súčasného systému

Dátový slovník

- **pre entitu Dodávateľ:**

Dodávateľ = id_dodávateľ + meno + priezvisko + adresa + telefónne číslo + (email)

id_dodávateľ = {[0-9]}

meno = {[A-Z|a-z]}

priezvisko = {[A-Z|a-z]}

adresa = ulica + číslo domu + mesto + PSČ

ulica = {[A-Z|a-z]}

číslo domu = {[0-9]}

mesto = {[A-Z|a-z]}

PSČ = {[0-9]}

telefónne číslo = {[0-9]}

email = {[A-Z|a-z|0-9|@|.]}

- **pre entitu Zboží:**

Zboží = id_zboží + názov + počet kusov na sklade + predajná cena

id_zboží = {[0-9]}

názov = {[A-Z|a-z|0-9]}

5.2 Analýza systému pomocou metódy YMSA

Základom tejto analýzy je nájdenie Esenciálneho modelu systému. Tento model sa skladá z modelu okolia a modelu systému, ako som už spomínal v kapitole 4.2. Preto aj pri našej analýze systému pre obchodovanie na internete začneme hľadaním modelu okolia. Model okolia sa skladá z troch častí. Prvou časťou je účel systému, kde slovné popíšeme funkcie systému. V našom prípade bude vyzeráť takto:

Účel systému

Systém sa snaží napodobniť klasický obchod v prostredí internetu, pričom využíva všetky jeho výhody. Zboží je vystavované v elektronickej podobe, v ktorej prebieha aj celý spôsob predaja a nákupu. Tento informačný systém značne zjednoduší nákup. Zákazník si v pohodlí domova vyberie v internetovom obchode zboží, objedná si ho a na dobierku ho dostane až domov. Priebežne môže kontrolovať aj stav svojich objednávok. Predajcovi sa zjednoduší kompletná správa jeho obchodu. Objednáva zásoby od dodávateľa, ktorého si sám zaregistruje, alebo ktorý sa v jeho internetovom obchode zaregistruje sám. Vybavuje objednávky od zákazníkov a zároveň ma presné informácie o zásobách na jeho sklade. Každý deň sa vyhodnocujú presné informácie o zboží, ktoré sa za určitý deň predalo. Na základe tohto systému sa aj produkty dodávateľov stávajú prístupnejšie.

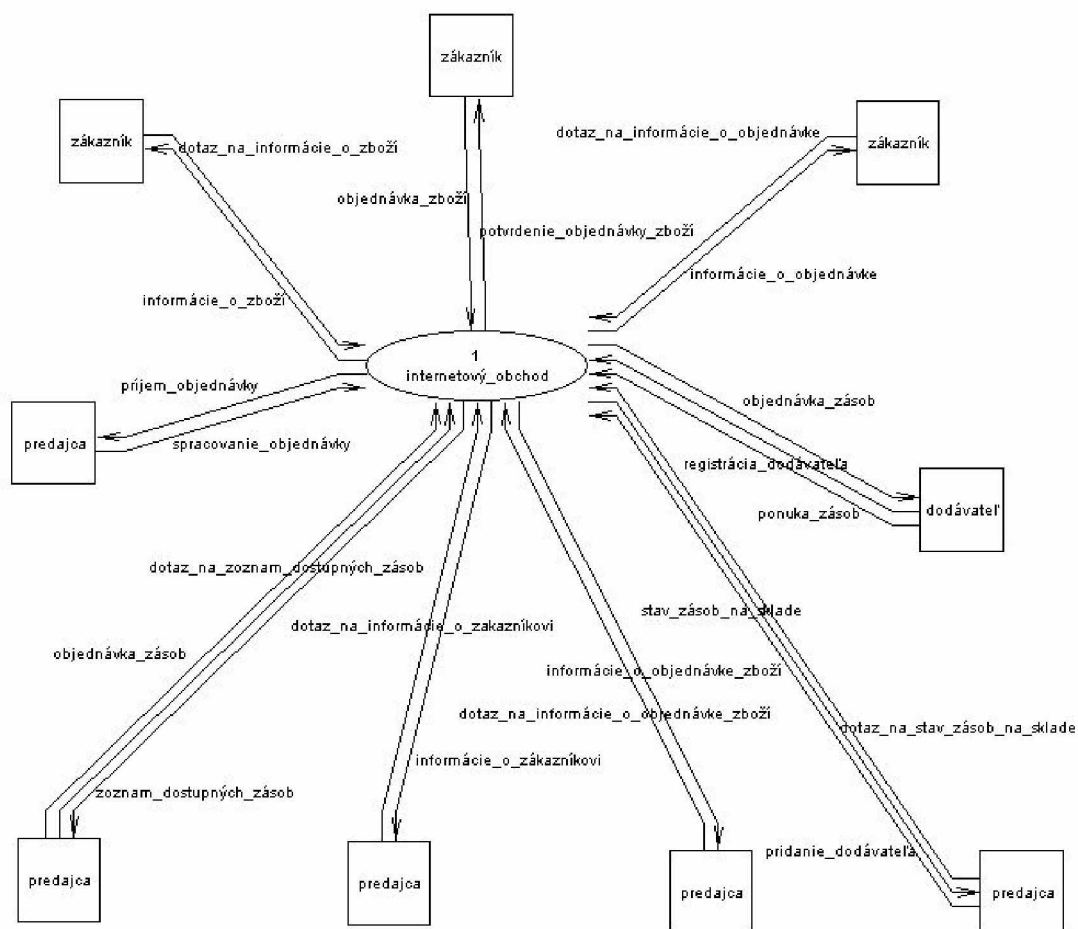
Druhou najdôležitejšou časťou je tvorba zoznamu udalostí. Tu analytik presne špecifikuje udalosti, na ktoré bude vytvorený systém reagovať. Udalosti sa delia podľa užívateľov, ktorí budú systém používať. V našom systéme to sú: Zákazník, Predajca a Dodávateľ a celý zoznam udalostí vyzerá takto:

Zoznam udalostí

- Zákazník
 1. Zákazník sa pýta na informácie o zboží (F)
 2. Zákazník objednáva zboží (F)
 3. Zákazník sa pýta na informácie o objednávke zboží (F)
- Predajca
 4. Predajca spracováva objednávky (F)
 5. Predajca objednáva zásoby (F)
 6. Predajca pridáva dodávateľa (F)
 7. dotaz na stav zásob na sklade (C)
 8. dotaz na informácie o zákazníkovi (C)
 9. dotaz na informácie o objednávke zboží (C)
 10. dotaz na zoznam dostupných zásob (C)
 11. denný prehľad predaja (T)
- Dodávateľ
 12. Dodávateľ dodáva zboží (F)
 13. Ponuka zásob (F)
 14. Registrácia dodávateľa (F)

Tretou časťou je Kontextový diagram. Ide o DFD diagram, ktorý obsahuje jeden proces predstavujúci celý budúci systém. Sú v ňom znázornené požiadavky užívateľov na systém a následne odpovede systému na jednotlivé požiadavky.

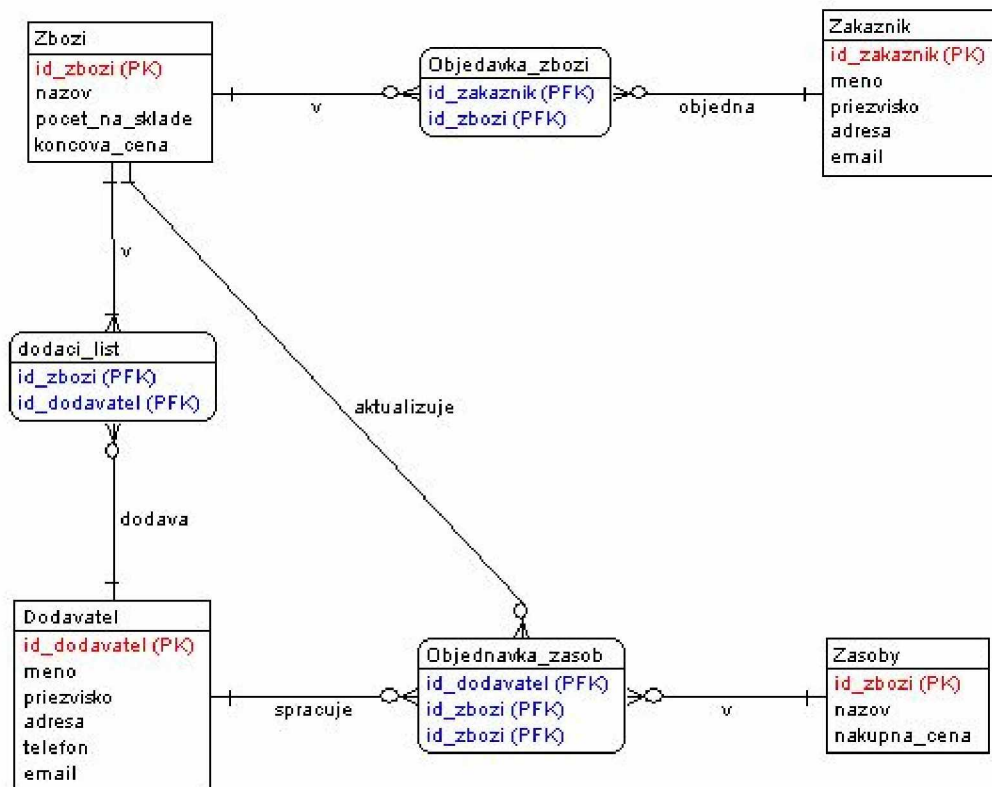
Kontextový diagram



Obrázok 5.4: Kontextový diagram

Týmto sme vytvorili kompletný model okolia systému a môžeme sa pustiť do tvorby modelu chovania nášho systému. Model chovania systému pri analýze YMSA sa tvorí tak, že prvotné DFD vychádza kompletne zo zoznamu udalostí. Pre každú udalosť sa vytvorí jeden proces, ktorý pomenujeme podľa očakávanej odozvy na túto udalosť. Pre priblíženie je prvotné DFD nášho systému znázornené na obrázku 5.5. Ako je na prvý pohľad vidieť, toto DFD je do veľkej miery neprehľadné, a preto sa model snažíme vyvažovať smerom hore a podobné procesy zlučujeme. Toto zlučovanie zvyčajne prebieha vo viacerých úrovniach. Pomocou tohto postupu sa však môže stať, že vzniknú nové procesy, ktoré sú neprehľadné a je treba ich ďalej dekomponovať. Týmto vznikne čistá funkčná dekompozícia. Pre dôkladnejšie pochopenie tohto vyvažovania, sú na priloženom CD znázornené všetky úrovne, ktoré vznikli pri zlučovaní jednotlivých procesom, ako aj ich neskoršie prípadné dekomponovanie.

ERD



Obrázok 5.6: ERD - diagram

6. Vyhodnotenie

6.1 Porovnanie metód SASS a YMSA

Pôvodná DeMarcova metodológia kládla veľký dôraz na modelovanie pomocou DFD. Štruktúrovaná špecifikácia zložená z hierarchie DFD modelu je vytvorená na základe rozkladu systému zhora dolu - od kontextového diagramu k DFD na najnižších úrovniach. Primitívne procesy nájdené pri rozklade, sú popísané minišpecifikáciami. Dátová časť systému je popísaná dátovým slovníkom. Rozhranie medzi jednotlivými procesmi tvoria dátové toky. Aj keď DeMarco navrhol metódu pre normalizáciu existujúcich dátových súborov na množinu dátových pamätí bez redundantných údajov, v praxi pri použití jeho metódy boli pamäte zložené z náhodne zoskupených položiek.

Avšak toto v analýze SASS nebol jediný problém. Už pri tvorbe fyzického dátového modelu mala väčšina užívateľov pocit, že analytik analyzovanému systému nerozumie a učí sa mu porozumieť za peniaze zákazníka. Ďalšou nevýhodou je, že užívateľ nechce spolupracovať pri tvorbe logického modelu. Užívatelia nerozumeli, ako môže analytik vytvoriť logický model, keď fyzickému ani trochu nerozumel a užívatelia majú pocit, že analytická fáza je obdobím nič nerobenia. Skutočná práca začne, až keď sa systém začne programovať. Veľa užívateľov spochybňuje tiež tvorbu systému, ktorý bude aj tak postupom času úplne zahodený a nahradený novým systémom.

Ďalšou zásadnou nevýhodou analýzy SASS je, že už vychádza z nejakého vytvoreného systému. Na prvý pohľad sa to nezdá byť žiadny problém, ale s tvorbou úplne nového systému analýza SASS vôbec nepočíta, a preto sa tieto systémy tiež ťažko analyzujú.

Problémy s definíciami dátových pamätí ďalej viedli k tomu, že pôvodné techniky DFD boli doplnené o dátové modelovanie založené na ER diagramoch a relačných dátových modeloch. ERD alebo obdobný dátový model je použitý pre identifikáciu entít (objektu), o ktorých sa uchováva informácia. ERD slúži ako základ pre definíciu pamäte na DFD modeli. Postup, ktorý využíva kombináciu DFD a ERD.

Yourdon použil kombináciu vyššie uvedených metód pre metodológiu *Moderná štruktúrovaná analýza*. V nej používa nástroje DFD a ERD pre analýzu, ktorá vedie k vytvoreniu *esenciálneho modelu systému*, a štruktúrogram (SC - structure chart) pre vytvorenie funkčného modelu zloženého z funkčných jednotiek. Pre analýzu systému pracujúcich v reálnom čase navrhuje použitie DFD rozšírených o riadiace procesy. Pre špecifikáciu procesu môžu byť použité klasické nástroje - minišpecifikácia, vývojový diagram. Táto analýza sa poučila z nedostatkov analýzy SASS a odstránila ich.

6.2 Zhodnotenie vhodnosti nástroja CASE Studio 2 pre obidve metódy analýzy

Nástroj Case Studio 2 bol zvolený pomerne vhodne. Pomocou neho sa dá ľahko vytvoriť DFD a ERD pre obidve metódy a poskytuje aj veľké množstvo podporných nástrojov. Nie je síce až tak užívateľsky príjemný ako niektoré ďalšie programy podobného charakteru, ale je prístupný za pomerne nízku cenu. Jeho asi najväčšou nevýhodou je, že neposkytuje vhodné nástroje pre analýzu zdola hore, ktorá je základom pre metódu YMSA.

7. Záver

Analýza je jedna z najdôležitejších častí tvorby každého systému. Dnes už existuje množstvo ďalších moderných techník ako systém analyzovať. Do popredia sa dostávajú hlavne objektovo orientované prístupy. Som si vedomý, že tieto postupy štruktúrovanú analýzu systému nakoniec vytlačia, avšak v blízkej budúcnosti bude mať táto analýza ešte veľký význam.

Hlavne metóda YMSA bude mať v budúcnosti svoje miesto pri analýze menších systémov. Je to jedna z najrozšírenejších a najlepších metód štruktúrovanej analýzy. Sú v nej zhrnuté všetky výhody štruktúrovanej analýzy a jej nedostatky sú minimálne. Preto si myslím, že zaoberať sa touto analýzou je výhodné aj v budúcnosti.

8. Literatura

- [DeM79] DeMarco T.: Structured Analysis and System Specification. Prentice Hall PTR, Facsimile edition, 1979.
- [Drb03] Drbal P.: Základy softwarového inženýrství. VŠE, Praha, 2003.
- [Fro00] Frolík V.: CASE Studio 2 – user's manual. Charonware, Ostrava, 2004.
- [Krá98] Král J.: Informační systémy. Science, Veletiny, 1998.
- [KrDe91] Král J., Demner J.: Softwarové inženýrství. Academia, Praha, 1991.
- [RiSo96] Richta K., Sochor J.: Softwarové inženýrství I. Vydavatelství ČVUT, Praha, 1996.
- [ŠeMiČe01] Šešera L., Mičovský A., Červeň J.: Datové modelování v příkladech. Grada, Praha, 2001.
- [SøKu93] Sølvsberg A., Kung D.C.: Information system engineering. Springer-Verlag, Berlin, 1993.
- [You89] Yourdon E.: Modern structured analysis. Prentice-Hall, Englewood Cliffs, 1989.

9. Internetové zdroje

- [W1] Sochor J., Ráček J.: Výukové materiály k předmětu PB007 Analýza a návrh systémů, Fakulta informatiky Masarykovy univerzity, 2004, Brno,
<http://www.fi.muni.cz/~racek/vyuka/ananas/>
- [W2] Ráček J.: Výukové materiály k předmětu PV165 Procesní řízení, Fakulta informatiky Masarykovy univerzity, 2004, Brno,
<http://www.fi.muni.cz/~racek/vyuka/procriz/>
- [W3] Adrian Kieß: Tom DeMarco Strukturierte Analyse und System Spezifikation, 2004,
<http://ebus.informatik.uni-leipzig.de/www/media/lehre/seminar-pioniere04/kiess-folien-demarco.pdf>
- [W4] Webové stránky CASE Studio 2,
<http://www.casestudio.com>
- [W5] Sochor J.: Výukové materiály k předmětu PA103 Objektové metody návrhu informačních systémů, Fakulta informatiky Masarykovy univerzity, 2004, Brno,
<http://www.fi.muni.cz/~sochor/PA103/Slajdy/>

Uvedené adresy boli platné k dátumu 20.4.2005.

Príloha A:

Štruktúra adresárov na CD:

/Bp – adresár obsahujúci text bakalárskej práce

/SASS – štruktúrovaná analýza pomocou metódy DeMarco

/YMSA – štruktúrovaná analýza pomocou metódy Yordon