

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

FRANCI DRNOVŠEK
RAZVOJ INFORMACIJSKEGA SISTEMA V NOVI LJUBLJANSKI BANKI Z OPISOM
PRIMERA NA ZBIRNEM CENTRU

VISOKOŠOLSKI STROKOVNI PROGRAM

Ljubljana, 2008

Mentor:
Doc. dr. Marko Bajec

Zahvala

V prvi vrsti bi se želel zahvaliti svojemu oddelčnemu vodji Davidu Podlipniku. Nadalje bi se rad zahvalil vsem sodelavcem, ki so mi svetovali pri izdelavi naloge, še posebej sodelavcu Burgar Tomažu. Zahvala gre tudi mojemu mentorju doc. dr. Marku Bajc-u za svetovanje.

Kazalo

1. POVZETEK	1
2. UVOD	3
3. Kratka predstavitev banke	5
3.1 UCIT.....	5
3.1.1 Oddelek za razvoj podpore plačilnim sistemom.....	7
4. Zbirni center NLB	8
4.1. Standardi in postopki izmenjave podatkov.....	9
4.1.1. Sestava paketov.....	9
4.1.2. Informacija o uspešnem prevzemu podatkov.....	10
4.1.3. Označevanje osnovnih zapisov.....	10
4.1.4. Označevanje paketov.....	10
4.1.5. Format polj.....	11
4.1.6. Struktura zapisov.....	11
4.2. Vodilni zapis.....	11
4.3. Zbirni zapis.....	12
4.4. Direktne bremenitve - DB.....	13
4.4.1. Postopek izmenjave podatkov.....	14
4.5. Primer pravilno izdelane datoteke DB.....	17
4.6. Vključevanje poslovnega partnerja v Zbirni center NLB.....	17
5. Spletna stran	19
5.1. Oblikovanje spletnih strani.....	19
5.1.1. Porazdelitev ekranskega prostora.....	19
5.1.2. Oblikovanje za različne tipe brskalnikov.....	20
5.1.3. Ločevanje pomena od prikaza.....	20
5.1.4. Odzivni čas.....	21
5.1.5. Povezave.....	22
5.1.6. Zbirka slogov.....	22
5.1.7. Okvirji.....	23
5.2. Vsebina spletnih strani.....	23
5.2.1. Vsebina navigacije.....	24
5.2.2. Čitljivost.....	24
5.2.3. Spletna dokumentacija.....	24
5.3. Oblika lokacije.....	25
5.3.1. Navigacija.....	25
6. CICS	26
6.1. Kaj je CICS.....	26
6.1.1. Potreba po interaktivnem dostopu.....	26
6.2. CICS transakcijski vmesnik.....	27
7. MQ	31
7.1. Kaj je MQ in kaj so njegove najpomembnejše lastnosti?.....	31
7.1.1. Zakaj v banki potreba po MQ Series?.....	32
7.1.2. MQ vloge za delo v banki.....	32
7.1.3. MQ na Z/OS.....	32
7.1.4. MQI (Message Queuing Interface).....	33
7.2. PREDSTAVITEV MQ Series programov.....	34
7.2.1. MQ (batch) programi za prenose podatkov.....	34
7.2.2. Procedura za brisanje z zagonom JOB-a JMQEMPTD.....	35
7.2.3. JOB JMQEMPTD.....	35

7.3. MQ (online) programi za transakcije.....	36
7.3.1. Kratek opis poteka proženja akcij in opis poti podatkov.....	36
8. DB2	38
8.1. Varnost v DB2 bazi	39
8.2. DCLGEN	40
9. COBOL DB2	43
9.1. Copybook	44
10. JCL (Job Control Language)	46
11. SKLEP.....	48
12. SEZNAM SLIK	49
13. SEZNAM TABEL	49
14. LITERATURA IN VIRI.....	50
15. Izjava o avtorstvu.....	51

Uporabljene kratice in simboli

ZC – Zbirni Center
NLB – Nova Ljubljanska Banka
MQ – Message Queue
CICS – Customer Information Control System
HTML - HyperText Markup Language
UCIT – Upravljalni Center za Informacijsko Tehnologijo
SUPB – Sistem za Upravljanje s Podatkovnimi Bazami
OE – Oddelčna Enota
OR – Osebni Račun
GK – Glavna Knjiga
SKP – Sektor Kadrovskega Področja
SEPA – Single Europe Payment Area
ADONIS – Aplikacija Direktnih Obremenitev Nacionalna In Sepa shema
RAD - Rational Application Developer
NPI – Novi Plačilni Instrumenti
DO – Direktna Odobritev – zapis 03
DB – Direktna Brementev – zapis 04
PR – Preverjanje Računa – zapis 80
TN – Trajni Nalog - zapis 05
PP – Posebna Položnica - zapis 01
PN – Posebna Nakaznica - zapis 02
DZZ – Delni Zbirni apis
MEDUZA - spletna aplikacija Bankarta, ki udeležencem ZC omogoča izmenjavo podatkov
DB2 – DataBase
IBM - International Business Machines
SQL - Structured Query Language
SAA - zbir dogovorov in protokolov, ki zagotavljajo konsistentno okolje za razvoj aplikacij,
neodvisno od vrste računalnika, na katerem se bo aplikacija izvajala
ASP – Active Server Pages
JCL – Job Control Language

1. POVZETEK

V začetnem poglavju sem na kratko predstavil podjetje in oddelek v katerem delujem, v vseh naslednjih poglavjih pa sem opisal aplikacijo Zbirni center Nove ljubljanske banke (v nadaljevanju ZC NLB) in njene glavne prvine.

ZC NLB je računalniško podprta aplikacija, preko katere se izvaja zbiranje, sortiranje, kontroliranje in posredovanje podatkov prevzetih od poslovnih partnerjev in bank.

Predstavil sem Zbirni center Bankart, s katerim si ZC NLB izmenjuje podatke, ki so v točno določeni strukturi. Tej strukturi se po vstopu v aplikacijo ZC NLB dodajo dodatni atributi, s katerimi transakciji zagotovimo sledljivost, jo zapisujemo v tabele ter beležimo njeno stanje, v katerem se trenutno nahaja in v katerega mora preiti. Podatke ZC NLB prevzema iz več vhodov. Eden izmed teh vhodov je tudi spletni vmesnik, zato sem predstavil glavne lastnosti in nevarnosti na katere moramo biti pazljivi pri izdelavi.

Ker je večina programov napisana v programskem jeziku COBOL, ti pa dostopajo do baze DB2, sem opisal programski jezik in predstavil DB2. Za izvajanje cobolskih programov je potrebno poznati tudi skriptni jezik JCL, ki sem ga opisal v naslednjem poglavju.

V nadaljevanju sem opisal MQ, s katerim poteka asinhrona ali sinhrona komunikacija med različnimi sistemi znotraj aplikacije.

S spletnim delom aplikacije pa je močno povezan večnamenski podatkovni komunikacijski sistem CICS. Lahko si ga predstavljamo kot posebni operacijski podsistem ali okolje za izvrševanje online aplikacijskih programov kot tudi vmesnik za dostop do datotek in baz podatkov.

Ključne besede

aplikacija

Zbirni center

komunikacija

ABSTRACT

In the first chapter the company and the department I work in are briefly presented. In the following chapters the ZC NLB application and its main components are described.

ZC NLB (the NLB bank assembly center) is a computerised application through which data is sorted, collected, controlled and forwarded. This data is taken from business partners and other banks.

I introduced the ZC Bankart which the ZC NLB exchange its data with. The structure of the data is precisely fixed. When data comes in ZC NLB the structure is expanded with some new attributes. With these attributes we can insure transaction traceability. Transaction is written in the database where we can monitor its status, the state the transaction is in and the one it must *transfer* into next. The ZC NLB receives its data out of a number of inputs. A web interface represents one of these inputs, which is why I presented the main properties and risks to be aware of in the creation of the web interface.

As most programmes are written in the COBOL computer language, and they access the DB2 database, the programme language and the DB2 are also presented. To execute the Cobol programmes one needs to be familiar with the JCL, which is also illustrated in the chapter following.

In the continuation, the MQ is depicted. This is a way of asynchronous and synchronised communication among different systems within an application.

The multipurpose data communication system CICS is tightly connected to the web part of the application. It can function as a special operating subsystem, as environment for the executing of the online application programme, or as an interface for file and database access.

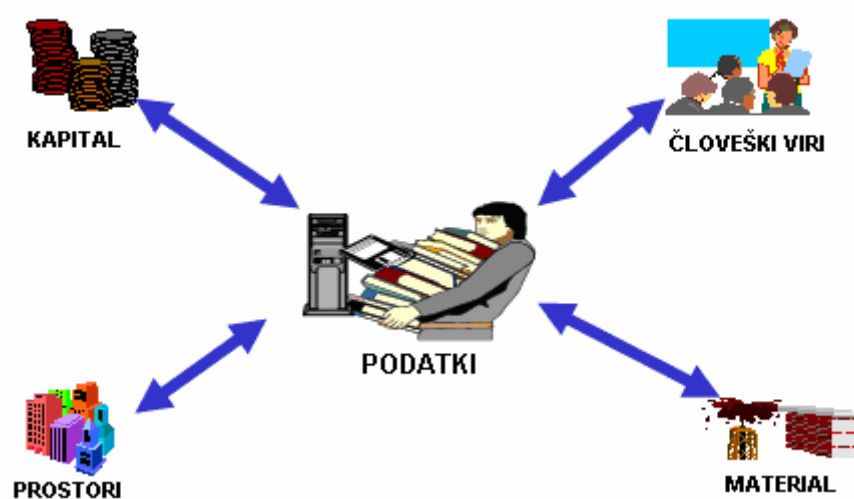
Key words

application
Assembly center
communication

2. UVOD

V diplomski nalogi bom s predstavitvijo praktičnih primerov poskušal opisati delovanje in komunikacijo znotraj ZC NLB. Na kratko bom opisal uporabljena orodja, podatkovno bazo, programske in skriptne jezike ter navedel infrastrukturo, na kateri sem pomagal razvijati programsko opremo in na kateri se izvajajo programi.

V sodobnih podjetjih je pomembno racionalno upravljanje z vsemi viri z namenom, da podjetje dosega čim večji profit ob čim manjših stroških. Poleg osnovnih tradicionalnih virov (kot na primer človeški viri, kapital, prostori, material) je pomembno, da se v podjetju zavedajo pomena podatkov kot enega najpomembnejših in osrednjih virov v podjetju (Slika 1).



Slika 1. Podatki kot osrednji vir v podjetju

Kot najbolj primerno orodje za upravljanje s podatki je računalniško podprto upravljanje s podatki s pomočjo sistemov za upravljanje z bazami podatkov.

Tako kot v večini večjih podjetij je tudi v NLB potrebno upravljati z velikimi količinami podatkov. V ta namen se uporablja računalniška podpora, tako imenovane baze podatkov. Preko njih poteka obdelava, shranjevanje ter uporaba nepreglednih množic podatkov.

NLB je finančna organizacija, ki za svoje komitente (fizične in pravne osebe) opravlja različne finančne storitve. To so poleg aktivnih in pasivnih poslov tudi nevtralni posli (posredovanje plačil v državi in s tujino). To pomeni, da je banka ponudnik storitev svojim komitentom. Svoje storitve zaračunava v obliki obresti, provizij in stroškov.

Vedno hitrejši tempo razvoja na področju najrazličnejše informacijske tehnologije narekuje vedno nove smernice za povezovanje informacijskih okolij, ki pa so po večini dokaj različna. Elektronsko poslovanje je ne samo poslovna priložnost, ampak tudi nuja za podjetja, ki želijo avtomatizirati svoje poslovanje in zmanjšati število napak, poleg tega pa še znižati stroške poslovanja, olajšati delo. Vsi se zavedamo, da je danes informacijska tehnologija nekaj

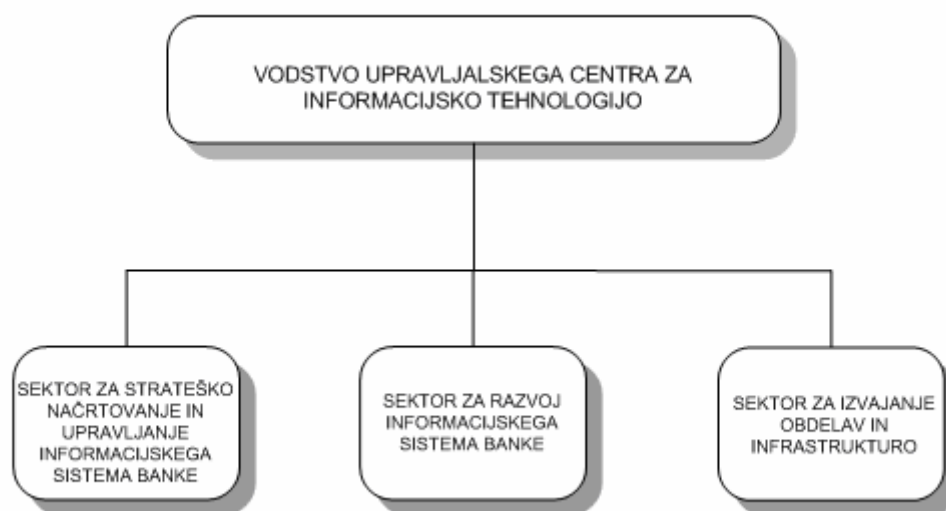
obveznega. V današnjem poslovnem svetu je pravočasen dostop do potrebne informacije ključen dejavnik uspeha. Osebni računalniki, preko katerih se medmrežje največ uporablja, zaradi svoje nemobilnosti ne zadovoljujejo več potreb sodobnega človeka, saj mora biti informacija dostopna kjerkoli in kadarkoli. Različna informacijska okolja pa predstavljajo več ovir za enostaven pretok podatkov, zato se srečujemo z različnimi vmesniki za povezovanje okolij, vmesniki za konverzije znakov, programskimi vmesniki in različnimi pripomočki za povezovanje informacijskih okolij različnih podjetij, ne samo med podjetji, temveč tudi za povezovanje znotraj podjetij. Kako torej povezati podjetje oz. več podjetij z informacijskimi sistemi oz. aplikacijami, da bomo zagotovili nemoten tok prenosa podatkov oz. nemoteno delovanje aplikacij kljub temu, da se srečujemo z različnimi okolji? Eden izmed odgovorov na omenjen problem prenosa podatkov med različnimi okolji leži v IBM-ovem produktu MQ Series.

3. Kratka predstavitev banke

NLB je delniška družba, katere delni lastnik je z 35,41 % Republika Slovenija. Odločitev o ustanovitvi današnje NLB d.d. je Državni zbor sprejel julija 1994 z dopolnili k Ustavnemu zakonu za izvedbo temeljne listine o samostojnosti in neodvisnosti Republike Slovenije, ki je bila sprejeta ob razglasitvi samostojne države Slovenije. NLB d. d. je pričela poslovati 28. julija 1994. Prevzela je premoženje, poslovanje in večji del obveznosti Ljubljanske banke d.d. (Letno poročilo 2007).

3.1 UCIT

UCIT oblikuje in nadzira izvajanje strategije in politike razvoja informacijskega sistema banke in Skupine NLB. Sodeluje pri oblikovanju in nadziranju izvajanja nabavne politike za področje informacijske tehnologije in skrbi za izpolnitev lastnih razvojnih in letnih poslovnih načrtov (Slika 2).



Slika 2. Struktura upravljalnega centra za informacijsko tehnologijo

Naloge UCIT-a so:

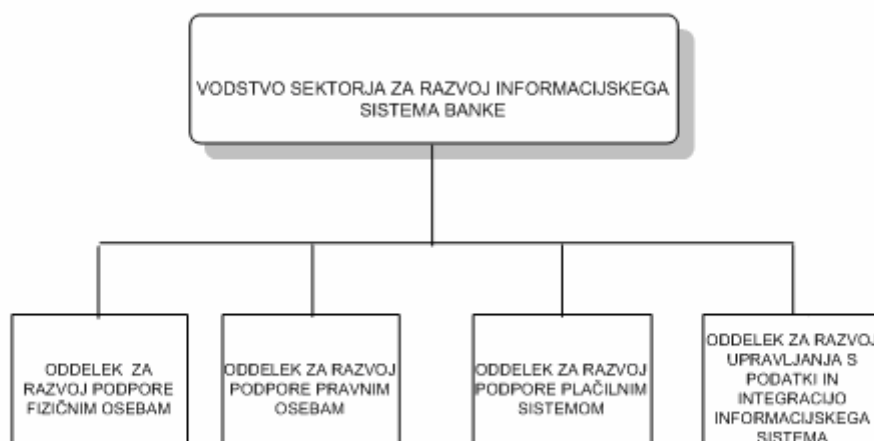
- priprava strategije in politike razvoja informacijskega sistema banke in Skupine NLB,
- načrtovanje optimalnega razvoja tehnološke podpore in uvajanje novih tehnologij,
- načrtovanje tehnične in tehnološko programske arhitekture in nadzor nad izvedbo,
- spremljanje in ukrepanje pri izvajanju strategije in politike razvoja informacijskega sistema banke,
- opredelitev postopkov neprekinjenega delovanja in obnovitve sistema v primeru motenj v delovanju ali nesreči,
- upravljanje s kadri in znanji delavcev UCIT-a,
- izvajanje nalog in aktivnosti gospodarjenja s poslovnim prostorom in tehnološko opremo v uporabi vodstva centra,

- izvajanje nalog tajniško-administrativne podpore zastopanju, vodenju in logistiki vodstva centra.

Center se deli na tri sektorje:

- sektor za strateško načrtovanje in upravljanje informacijskega sistema banke,
- sektor za izvajanje obdelav in infrastrukturo,
- sektor za razvoj informacijskega sistema banke.

Delam v sektorju za razvoj informacijskega sistema banke v oddelku za razvoj podpore plačilnim sistemom, zato ga bom na kratko predstavil (Slika 3).



Slika 3. Struktura sektorja za razvoj informacijskega sistema banke

V sektorju se skrbi za načrtovanje in izvedbo prenove tehnologije dela poslovanja banke. Nadzira se izvajanje načrtovanih razvojnih nalog in letnega poslovnega načrta, projektira in izvaja razvoj programskih rešitev računalniške podpore poslovnih funkcij banke.

Naloge sektorja so:

- vodenje ključnih projektov/programov z elementi informacijske tehnologije,
- sodelovanje pri oblikovanju in izvajanju strategije in politike razvoja informacijskega sistema banke in Skupine NLB,
- načrtovanje in obvladovanje arhitekture tehnološko-programskih rešitev banke in Skupine NLB,
- oblikovanje, uvajanje in nadzor sistema skrbništva nad informacijskim sistemom banke,
- izvajanje nalog in aktivnosti gospodarjenja s poslovnim prostorom in tehnološko opremo v uporabi sektorja,
- izvajanje nalog organizacijsko-administrativne podpore zastopanju, vodenju in logistiki sektorja.

3.1.1 Oddelek za razvoj podpore plačilnim sistemom

Naloge:

- razvijanje načrtovanih rešitev za informacijske sisteme poročanja in upravljanja,
- priprava specifikacij sprememb programske opreme,
- vzdrževanje in uvajanje sprememb rešitev za informacijske sisteme poročanja in upravljanja,
- priprava in skrbništvo nad tehnološkimi in tehničnimi navodili,
- vodenje projektov z elementi informacijske tehnologije,
- sodelovanje pri načrtovanju in uvajanju novih tehnoloških rešitev za poslovne procese banke,
- načrtovanje, sodelovanje in koordinacija aktivnosti pri izvajanju razvojnih nalog in sodelovanje z uporabniki,
- upravljanje z razvojnimi in testnimi okolji,
- izvajanje skrbništva nad programsko opremo za informacijske sisteme poročanja in upravljanja,
- sodelovanje pri načrtovanju projektnih rešitev banke,
- skrbništvo podatkovnega modela,
- oblikovanje in upravljanje sprememb podatkovnih modelov banke.

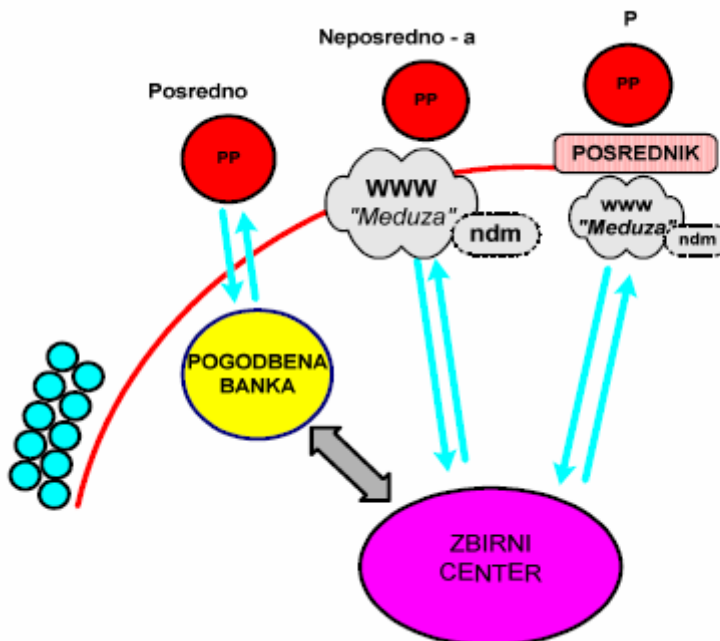
4. Zbirni center NLB

Zbirni center NLB je aplikativna podpora preko katere se izvaja zbiranje, sortiranje, kontroliranje in posredovanje podatkov v obdelavo, prevzetih od poslovnih partnerjev. Namenjen je za prevzem podatkov od poslovnih partnerjev, ki so vključeni v ZC Bankart in poslujejo preko pogodbene banke ter izključno internim obdelavam za komitente, ki niso vključeni v ZC Bankart. Deluje v okviru bank: NLB, Gorenjska banka d.d., Banka Celje d.d.

Register Zbirnega centra NLB je del aplikativne podpore ZC NLB, v katerega vključujemo poslovne partnerje, ki izmenjujejo podatke za direktne odobritve v skladu s standardi NPI. Poslovne partnerje, ki izmenjujejo podatke za direktne odobritve v skladu z standardi NPI in niso vključeni v ZC Bankart, vključujemo INTERNO v Register ZC NLB. Poslovne partnerje, ki izmenjujejo podatke za direktne odobritve v skladu z standardi NPI in se vključujejo v ZC Bankart, ne glede na izmenjavo podatkov (kot je razvidno iz Slike 4) direktno z ZC Bankart ali preko pogodbene banke, vključujemo EKSTERNO v Register Zbirnega centra NLB.

Zbirni center je računalniški servis, ki v elektronski obliki sprejema, sortira in posreduje podatke/informacije udeležencem sistema in hkrati vodi ustrezno statistično spremljavo poslovanja.

V okviru družbe Bankart d.o.o. je bil ustanovljen Zbirni center Bankart z namenom poenotenja in racionalizacije postopkov pri poslovanju z NPI med udeleženci plačilnega sistema [1].



Slika 4: Načini izmenjave podatkov

4.1. Standardi in postopki izmenjave podatkov

V sistemu ZC nastopa več različnih struktur zapisov:

- vodilni zapis (enoten za vse vrste storitev),
- zbirni zapis (enoten za vse vrste storitev),
- kontrolni zapis (enoten za vse vrste storitev),
- informativni zapis (INFO datoteka - vključitev novih poslovnih partnerjev v ZC),
- osnovni zapis za PP in PN,
- delni zbirni zapis za PP in PN,
- osnovni zapis za DO,
- delni zbirni zapis za DO,
- zapis za preverjanje računov fizičnih oseb pri bankah (PR),
- osnovni zapis za DB ,
- delni zbirni zapis za DB ,
- osnovni zapis za TN ,
- delni zbirni zapis za TN.

4.1.1. Sestava paketov

Izmenjava podatkov v sistemu ZC poteka v paketih – paketna izmenjava podatkov. Paket sme vsebovati le osnovne zapise za eno storitev! Obvezna sestava vsakega vhodnega paketa v ZC je **vodilni zapis**, eden ali več osnovnih zapisov (za katerikoli inštrument NPI; PP, PN, DO, DB, PR, TN) ter zbirni zapis.

Primer vhodnega paketa za storitev direktne odobritve - DO (vrsta zapisa „03“):

90.....vodilni zapis
 03.....osnovni zapis
 03.....osnovni zapis
 03.....osnovni zapis
 99.....zbirni zapis

Pri izhodnih paketih iz ZC se v obdelavah ZC obveznim zapisom iz vhodnega paketa dodajo še delni zbirni zapisi (DZZ), ki jih odvisno od posamezne storitve ustvari Zbirni center pri obdelavi prvega oz. drugega koraka (podrobnosti opisane v poglavju Delni zbirni zapisi).

Primer izhodnega paketa DO iz ZC z dodanim DZZ:

90.....vodilni zapis
 03.....osnovni zapis
 03.....osnovni zapis
 03.....osnovni zapis
 93.....delni zbirni zapis
 99.....zbirni zapis

4.1.2. Informacija o uspešnem prevzemu podatkov

Udeleženci v sistemu ZC dobijo informacijo o uspešnem prenosu podatkov preko kontrolnih zapisov. Zbirni center za vsak prejeti paket pošiljatelju posreduje kontrolni zapis (KZ) po isti komunikacijski poti kot je prejel podatke. Izdelava kontrolnih zapisov ni odvisna od urnika obdelav in se izdelava takoj po prevzemu vhodnih podatkov (cca. v roku 30 min., odvisno od izbranega načina izmenjave podatkov). Oznake „vrsta zapisa“ določajo pravilnost posredovanih paketov in so naslednje:

00.....kontrolni zapis	– paket je pravilen
11.....kontrolni zapis	- paket je napačen

4.1.3. Označevanje osnovnih zapisov

Vsak osnovni zapis dobi ob nastanku enolično določeno vrednost v sistemu - id zapisa, ki jo ustvari iniciator v dolžini 24 mest. Vrednost id zapisa enolično določa zapis in ostane nespremenjena v vseh korakih obdelave od iniciatorja do končnega uporabnika.

4.1.4. Označevanje paketov

Paketi vsebujejo enolično oznako številka paketa, ki jo ustvari iniciator podatkov v dolžini 18 mest in v sistemu ne sme biti ponovljiva. Številka paketa se lahko spremeni v obdelavah pri pogodbeni banki, ko le ta "prepakira" podatke za svojega poslovnega partnerja, ter v vseh obdelavah ZC. ZC pri ustvarjanju številke paketa vedno uporabi (na prvih osmih mestih) davčno številko končnega uporabnika podatkov.

Banka, ki hkrati nastopa v vlogi pogodbene in matične banke iz DŠ, ki se nahaja v številki paketa, ugotovi ali so podatki namenjeni banki za obdelavo (npr. izvedbo DB na računu komitenta) ali poslovnemu partnerju, ki podatke izmenjuje preko pogodbene banke.

Primer:

- če je v številki paketa davčna številka banke, potem gre za paket, ki je namenjen banki in ga mora obdelati,
- če je v številki paketa davčna številka poslovnega partnerja, potem gre za paket, ki vsebuje že obdelane podatke in jih mora banka le posredovati končnemu prejemniku.

Tudi ko komitenti ZC izmenjujejo podatke preko izbranega posrednika (pogodbene banke ali drugega posrednika, npr.: računovodskega servisa,...), ZC pripravi pakete za posamezne komitente (DŠ komitenta ZC) in jih posreduje posredniku.

4.1.5. Format polj

Polja v zapisih so lahko C – tekstovnega ali N – numeričnega tipa za katere veljajo naslednje značilnosti:

- C = tekstovni (poravnava levo, če ni podatka, se polje zapolni s presledki),
- N = numerični (polje mora biti zapolnjeno z vodilnimi ničlami oz. ničlami v celoti, če ni podatka).

4.1.6. Struktura zapisov

Pri vseh storitvah ZC so enotne naslednje strukture zapisov:

- vodilni zapis,
- zbirni zapis,
- kontrolni zapis,
- delni zbirni zapis,
- informativni zapis.

4.2. Vodilni zapis

Vsak paket obvezno začneja **vodilni zapis**, katerega vsebina in dolžina je enotna za vse vrste storitev. Struktura vodilnega zapisa je prikazana v Tabeli 1.

polje	Tip	zac	dolzina	obveznos t	opis polja - vsebina	nabor vrednosti - format	spremembe
vrsta zapisa	C	1	2	DA	oznaka vodilnega zapisa	“90”	NE
številka paketa	N	3	18	DA	številka paketa	99999999LLMMDDS SPP	NE
število transakcij	N	21	6	NE	uporabi se konstanta	“000000”	NE
oznaka knjiženja	N	27	1	DA	uporabi se konstanta	„0”	NE
datum	N	28	8	DA	datum kreiranja vodilnega zapisa	LLLLMMDD	DA
ura	N	36	6	DA	ura kreiranja vodilnega zapisa	HHNNSS	DA
prazno	C	42	152	NE			

Tabela 1: Struktura vodilnega zapisa

Primer:

Podjetje LORA d.o.o.; davčna št.: 47851716; dan in ura kreiranja vodilnega zapisa:
05.11.2002, 11:30:26; storitev: DO, 1. paket

90478517160211050301000000020021105113026.....

4.3. Zbirni zapis

Vsak paket obvezno zaključuje **zbirni zapis**, ki omogoča nadzor nad celovitostjo posredovanih podatkov. Vsebina in dolžina tega zapisa je enotna za vse vrste storitev. Struktura zbirnega zapisa je prikazana v Tabeli 2.

Polje	tip	Zac	dolzina	obveznost	opis polja - vsebina	nabor vrednosti – format	spremembe
vrsta zapisa	C	1	2	DA	oznaka zbirnega zapisa	“99”	NE
številka paketa	N	3	18	DA	številka paketa	99999999LLMMDDS SPP	NE
število transakcij	N	21	6	DA	št. transakcij znotraj enega paketa (brez delnih zbirnih zapisov!)		DA
oznaka knjiženja	N	27	1	DA	uporabi se konstanta	„0”	NE
datum	N	28	8	DA	datum kreiranja zbirnega zapisa	LLLLMMDD	DA
znesek	N	36	15	DA	skupni znesek vseh transakcij znotraj enega paketa, ne glede na oznako knjiženja (brez delnih zbirnih zapisov)		DA
oznaka valute	N	51	3	DA	uporabi se konstanta	„705“	NE
prazno	C	54	140	NE			

Tabela 2: Struktura zbirnega zapisa

Primer:

Podjetje LORA d.o.o.; davčna št.: 47851716; dan in ura kreiranja zbirnega zapisa: 05.11.2002; storitev: DO, 1. paket; št. transakcij: 13; znesek vseh transakcij:12345678,00

99478517160211050301000013020021105000000012345678705.....

4.4. Direktne bremenitve - DB

Struktura osnovnega zapisa za DB je prikazana v Tabeli 3.

polje	tip	zac	dolž	obveznost	opis polja - vsebina	nabor vrednosti - format	spremembe
vrsta zapisa	C	1	2	DA	oznaka za DB	“04”	NE
id zapisa	N	3	24	DA	oznaka osnovnega zapisa	99999999LLM MDDSSPPZZZ ZZZ	NE
oznaka knjiženja	N	27	1	DA	uporabi se konstanta	„0“	DA
datum	N	28	8	DA	datum valute oz. datum otvoritve ali ukinitve DB, odvisno od posamezne situacije	LLLLMMDD	DA
znesek	N	36	15	DA	znesek transakcije		NE
oznaka valute	N	51	3	DA	uporabi se konstanta	„705“	NE
račun komitenta ZC	C	54	18	DA	račun prejemnika sredstev		NE
referenčna številka	C	72	20	DA	Podatek je namenjen upniku		NE
namen	C	92	35	NE	Poljubno besedilo za na izpisek		NE
šifra nakazila	C	127	3	NE	uporabi se vrednost iz Tabele S11 - Šifrant namenov, polje sif_nakazila; (Opomba: šifro 000 se uporablja le v primeru , ko je polje „namen“ izpolnjeno)		NE
poravnalni račun	C	130	15	DA	TRR račun dolžnika oz. TRR poravnalne banke, pri kateri ima dolžnik odprt račun (če dolžnik še nima TRR računa)		DA
enota	N	145	3	NE	enota banke dolžnika (izpolni se samo v primeru, ko dolžnik še nima TRR računa, če ga ima so tu “000”)		NE
vrsta posla	N	148	2	DA	- v primeru, ko dolžnik nima TRR računa se uporabijo vrednosti iz Tabela S3 – Šifrant vrste posla, polje vrsta_posla, - v primeru, ko ima dolžnik TRR račun se uporabi konstanta	“11”, “13”, “15”, “21”, “25” “00”	NE

polje	tip	zac	dolž	obveznost	opis polja - vsebina	nabor vrednosti - format	spremembe
partija	N	150	10	NE	*razlaga pod tabelo		NE
šifra prejemka / izdatka	C	160	5	NE	Prazno		NE
vrsta informacije	N	165	2	DA	uporabi se vrednost iz Tabele S4 - Šifrant vrst informacij, polje vrsta_inf	„01“, „02“, „03“, „04“	NE
status	N	167	2	DA	uporabi se vrednost iz Tabele S5 - Šifrant statusov, polje status	„01“, „02“, „03“, „04“, „12“, „13“, „14“, „16“, „21“, „22“, „30“, „31“, „32“, „41“, „52“, „53“	DA
šifra konta	N	169	3	NE	banka lahko določi šifro konta in na podlagi le-te usmeri zapis v knjiženje		DA
frekvenca	N	172	2	DA	uporabi se vrednost iz Tabele S7 – Šifrant frekvenc, polje frek		NE
zap. št. DB upnika	N	174	5	DA	uporabi se vnaprej določena vrednost iz baze DB, polje DB_ZapStUpnika		NE
zap. št. DB dolžnika	N	179	10	DA	podatek, ki ga vpiše in vodi banka, ki izvede otvoritev		NE
operater/ blagajnik	C	189	5	NE			NE

Tabela 3: Struktura osnovnega zapisa za DB

* V polje “partija” se v primeru, ko je v polju “poravnalni račun” TRR račun fizične osebe, uporabi konstanta “0000000000”, v ostalih primerih se v polje polni partija TRR oziroma partija dokumenta posamezne vrste posla (npr. pri nakazilu kredita – partija kredita,...).

4.4.1. Postopek izmenjave podatkov

Iniciator (upnik) formira paket s podatki o DB. Paket na dogovorjen način posreduje v ZC. ZC prejete podatke DB različnih iniciatorjev združi ter formira nove pakete za posamezne matične banke pri katerih se vodijo računi dolžnikov in se izvaja obdelava posameznih transakcij. Paket za matično banko vsebuje vodilni, zbirni in poljubno število osnovnih zapisov DB.

Primer vhodnega paketa v ZC s podatki DB:

90.....	vodilni zapis
04.....	osnovni zapis za DB
04.....	osnovni zapis za DB
04.....	osnovni zapis za DB
99.....	zbirni zapis

Primer izhodnega paketa iz ZC s podatki DB za matično banko (v ZC so prispeli podatki DB od treh iniciatorjev, dolžniki sredstev pa imajo račune v tej matični banki) :

90.....	vodilni zapis
04.....	osnovni zapis za DB od posl. part. A
04.....	osnovni zapis za DB od posl. part. A
04.....	osnovni zapis za DB od posl. part. B
04.....	osnovni zapis za DB od posl. part. B
04.....	osnovni zapis za DB od posl. part. C
04.....	osnovni zapis za DB od posl. part. C
99.....	zbirni zapis

Matična banka obdela podatke na datum valute bremenitve računa dolžnika in kreira nove zapise s povratnimi informacijami za upnika o uspešno in neuspešno izvedenih transakcijah. ZC prejete podatke ponovno združi ter formira nove pakete DB za posamezne iniciatorje. Paket za posameznega iniciatorja vsebuje vodilni, zbirni, določeno število osnovnih zapisov s povratnimi informacijami DB in delni zbirni zapis **za uspešno izvedene transakcije DB**.

Primer izhodnega paketa iz ZC s podatki DB za iniciatorja podatkov – poslovnega partnerja (v ZC so prispeli obdelani podatki DB od treh matičnih bank) :

90.....	vodilni zapis
04.....	osnovni zapis za DB iz banke A
04.....	osnovni zapis za DB iz banke A
94.....	delni zbirni zapis za DB
04.....	osnovni zapis za DB iz banke B
04.....	osnovni zapis za DB iz banke B
94.....	delni zbirni zapis za DB
04.....	osnovni zapis za DB iz banke C
04.....	osnovni zapis za DB iz banke C
94.....	delni zbirni zapis za DB
99.....	zbirni zapis

Urnik izmenjave podatkov DB:

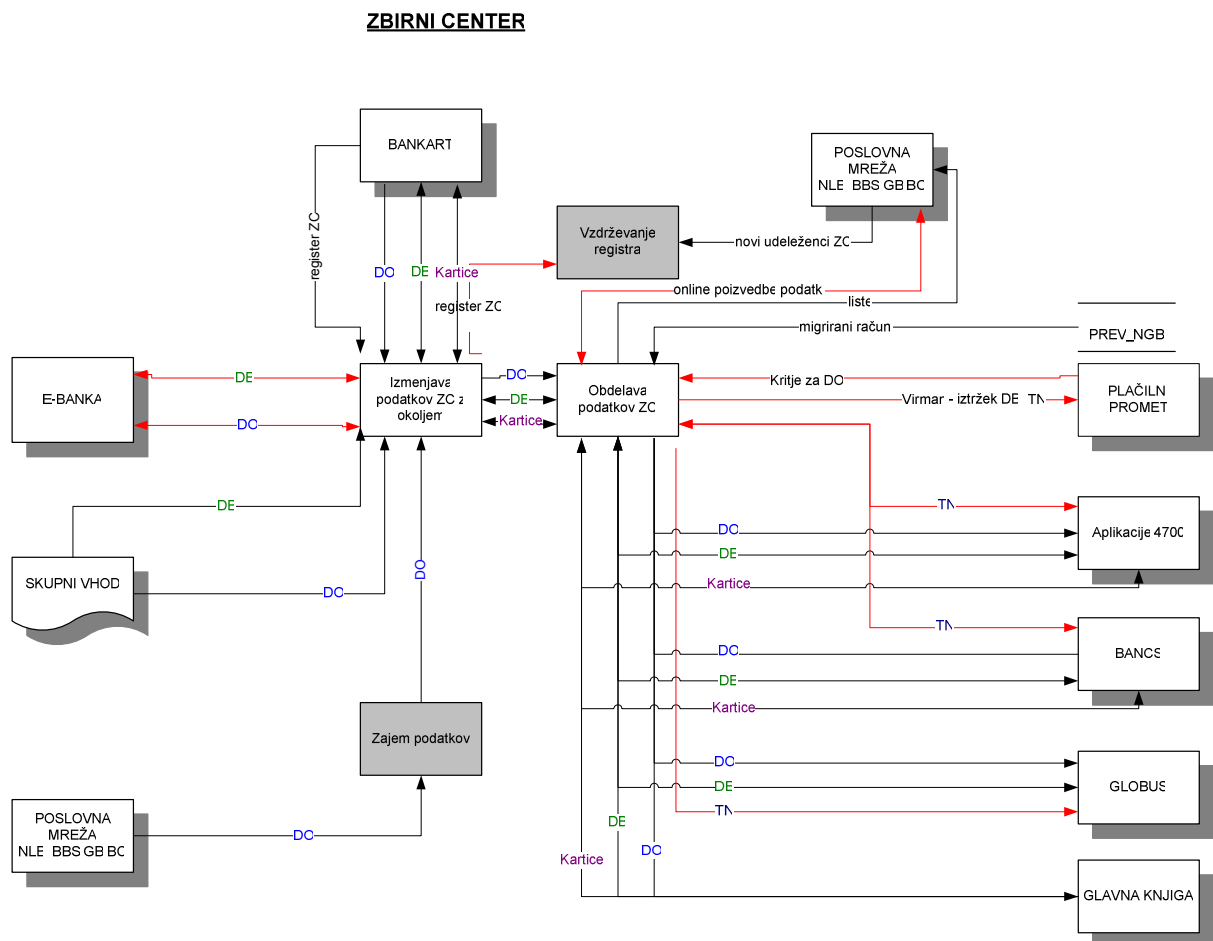
- podatki za DB morajo biti Zbirnemu centru posredovani najkasneje do 24. ure dva delovna dneva pred datumom bremenitve računa dolžnika. Če poslovni partner izmenjuje podatke preko posrednika, mora poskrbeti za pravočasnost posredovanih podatkov in upoštevati čas obdelav pri posredniku,
- ZC vse prejete podatke različnih iniciatorjev združi ter formira nove pakete za posamezne matične banke najkasneje do 06. ure zjutraj en delovni dan pred datumom bremenitve računa dolžnika,

- matična banka obdela podatke na datum bremenitve računa dolžnika oziroma prvi naslednji delovni dan, kreirane povratne informacije pošlje v ZC,
- ZC jih v prvi obdelavi DB (po urniku obdelav) obdela in jih posreduje upniku oz. pogodbeni banki, v kolikor le-ta nastopa kot posrednik pri izmenjavi podatkov,
- posamezna matična banka bo upniku nakazala sredstva v skupnem znesku uspešno realiziranih DB v dobro računa upnika s sklicem odobritve.

Pri DB se pojavljajo različne kombinacije polj, vrsta informacije in status. Iz kombinacij je razvidno v kateri smeri potekajo podatki (iniciator preko ZC matični banki ali obratno).

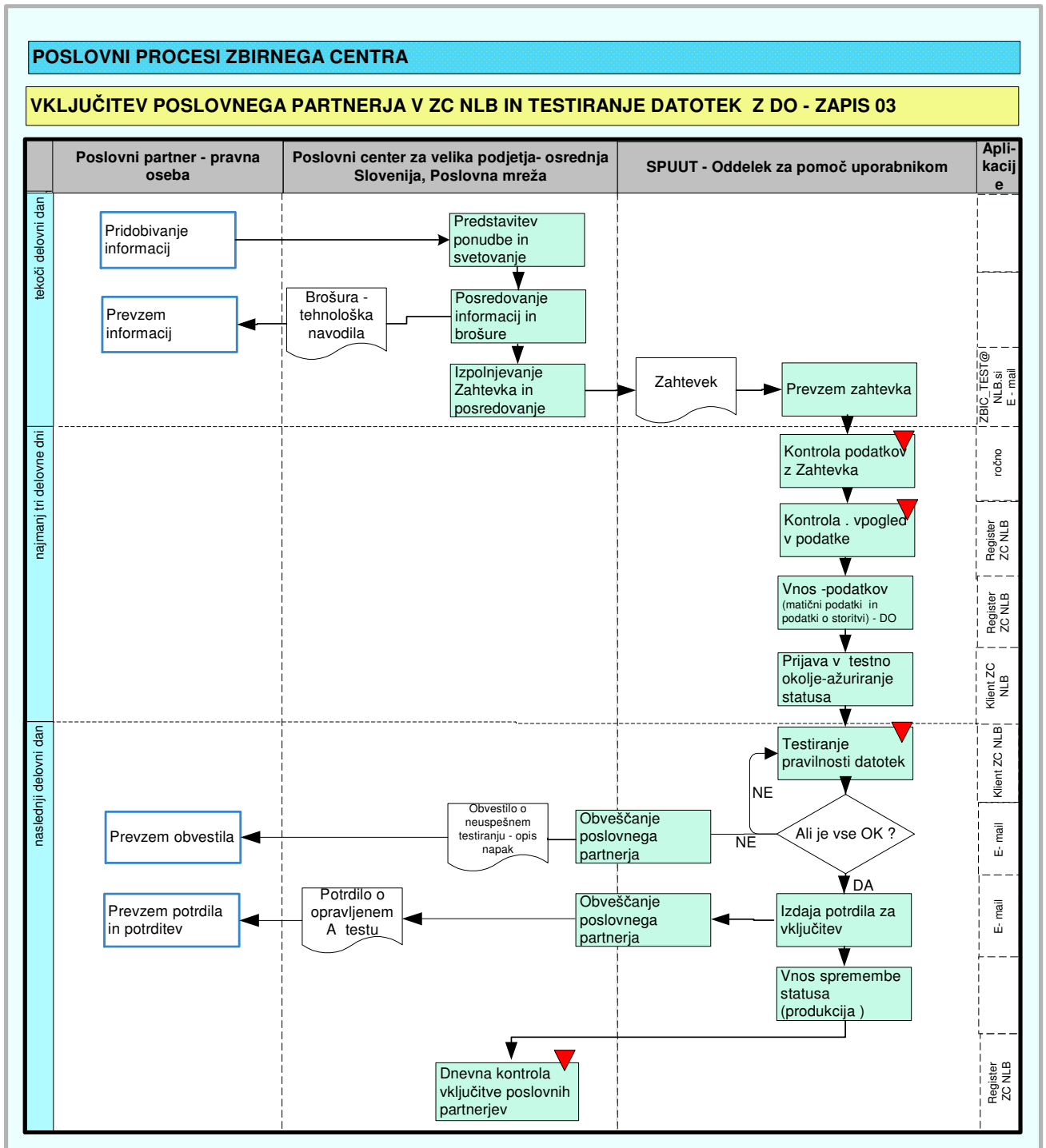
V prvem koraku se transakcija prvič pojavi v sistemu ZC, ko poslovni partner preko ZC pošlje na matično banko prve informacije o DB oz. ko matična banka posreduje preko ZC poslovnemu partnerju zapis za otvoritev, ukinitvev ali spremembo frekvence pri transakcijah DB.

V drugem koraku transakcija že obstaja v sistemu, pri tem gre za povratno informacijo o uspešno oz. neuspešno izvedenih transakcijah DB kot so ugovori, ukinitve ali spremembe pri transakcijah DB.



Slika 5 : Prikaz glavnih gradnikov v ZC NLB

vklučitev poslovnega partnerja v ZC NLB. Po uspešno zaključenem testiranju pripravljenih datotek na prenosnem mediju, vključitvi v produkcijsko okolje, lahko poslovni partner prične posredovati podatke za plače in ostale prejeme (Slika 7).



Slika 7: Prikaz poslovnega procesa

5. Spletna stran

5.1. Oblikovanje spletnih strani

Cilj oblikovanja spletne strani naj bo enostavnost. Uporabniki so redko na izbrani spletni lokaciji samo zato, da bi občudovali njeno obliko in grafični dizajn; namesto tega jih zanima vsebina. Važno je zagotoviti, da je oblika spletnih strani funkcionalna za veliko število platform in da lahko k njej pristopijo tudi uporabniki s starejšo tehnologijo. Potrebno je preveriti, kako strani izgledajo na malih ekranih kot tudi, če je odzivni čas branja strani primeren za tiste, ki dostopajo do medmrežja s povezavami nižjih hitrosti. Vsi ti elementi so predstavljeni v obliki priporočil za snovanje dostopnih spletnih strani.

```
<HTML>
<HEAD>
<TITLE>Tu je naslov spletne strani</TITLE>
</HEAD>
<FRAMESET COLS="100", FRAMEBORDER="no">
<FRAME SRC="links.htm" NAME=navi>
<FRAME SRC="tag.htm" NAME=main>
</FRAMESET>
<NOFRAMES>
Oprostite, brskalnik ne more prikazati okvirov.
</NOFRAMES>
</HTML>
<STYLE TYPE="TEXT/CSS">
<!--
body { background: yellow; margin-left; 20pt }
p { font-family: helvetica, arial; font-size; 12pt; color:
blue }
/-->
</STYLE>
```

5.1.1. Porazdelitev ekranskega prostora

Za uporabnika je najbolj pomembna vsebinska plat spletne strani, toda na spletu lahko najdemo veliko primerov slabe porazdelitve prostora in postavitve komponent.

Nekatere strani porabijo večino prostora za navigacijo. Z vidika spletne uporabnosti bi bilo najbolje, da je spletna stran brez reklamnih sporočil; če pa že moramo objaviti oglase, je najbolje, da so obvezni elementi strani ali del navigacije.

Uporabnik lahko dojame grupiranje informacij tudi s pravilno porazdelitvijo praznega prostora, torej je priporočljivo načrtovati strani z ne preveč informacijami. Za razmejitev med vsebinskimi segmenti je zato mnogokrat bolje uporabiti prazen prostor kot debele črte, saj s tem dosežemo, da uporabnik hitreje prebere vsebino.

Osnovni princip oblikovanja uporabniškega vmesnika je pregled vseh segmentov

oblikovanja. Če uporabniški vmesnik dobro deluje brez nekaterih elementov, jih lahko brez škode odstranimo s spletne strani.

5.1.2. Oblikovanje za različne tipe brskalnikov

Za razliko od običajnih grafičnih uporabniških vmesnikov, kjer programer pozna sistem za katerega gradi vmesnik, instalirane pisave, tipično velikost uporabljenih ekranov, vrstni red prikazovanja posameznih menijev, so na medmrežju uporabniki tisti, ki kontrolirajo navigacijo med spletnimi stranmi. Načelno lahko oblikovalci prisilijo uporabnika, da se giblje po točno določenem vrstnem redu strani, toda take lokacije so odbijajoče in nudijo majhno stopnjo svobode. Bolje je z oblikovanjem omogočiti svobodno gibanje po straneh, na vsako stran pa postaviti logo (povezava na začetno stran), s čimer se omogoči smiselno povezano navigacijo za uporabnike, ki so prišli na neko notranjo stran.

Glede na to, da ne moremo vnaprej vedeti kakšno velikost ekrana bodo uporabljali uporabniki, mora biti dizajn prilagojen vsem ločljivostim, povedano drugače; potrebno je graditi strani, ki so neodvisne od ločljivosti ekranov, na katerem so prikazane. Glavni princip oblikovanja je, da nikoli ne smemo vnaprej določiti velikosti tabel, okvirjev ali ostalih elementov. Namesto tega je potrebno podajati vrednosti v odstotkih razpoložljivega prostora. Potrebno je še omeniti, da uporabniki uporabljajo različne velikosti pisav. Zato je potrebno zagotoviti, da oblika zglada dobro na velikostih, ki so večje ali manjše od tistih, ki jih ponavadi uporabljamo. Različne velikosti pisav se uporabljajo zaradi različnih potreb uporabnikov ali morda zato, ker so na malih ekranih velikih ločljivosti črke enostavno premajhne, da bi bile čitljive.

Pri oblikovanju grafičnih elementov je prav tako treba paziti na različne uporabljene ločljivosti. Posebej je potrebno paziti na ikone, ki morajo biti dobro vidne tudi na večji ločljivosti. Čim večja je ločljivost, tem manjši je grafični element, tekst, ki je vgrajen v slike pa mora biti relativno velik, da bi bil čitljiv.

Spletne strani, ki niso neodvisne od ločljivosti, se dostikrat ne tiskajo kot je treba. Razlog je v tem, da tiskalnik ni tako širok kot velikost monitorja. Zato je najbolje, da se za tiskanje pripravijo ločene verzije dokumentov za tiskanje daljših strani, čeprav uporabniki pogosto tiskajo direktno na tiskalnik. Nestandardne uporabe HTML-ja se je potrebno izogibati.

5.1.3. Ločevanje pomena od prikaza

Oblika strani, ki je odvisna od prikaza, je uspešna samo v primeru, če lahko predvidimo strojno in programsko opremo in parametre na strani uporabnika. Težko je pričakovati, da brskalnik omogoča optimalen uporabniški vmesnik za tako različne pogoje uporabe – od malih ekranov, mobilnih telefonov do virtualnih časopisov na ravnih ekranih.

Nove naprave imajo grafične sposobnosti, ki se močno razlikujejo od tradicionalnih zaslonov in ne morejo dobro prikazati spletnih strani, ki so oblikovane za prikaz na standardnem zaslonu. Uporaba semantičnega kodiranja omogoča napravi, da optimizira prikaz skladno s svojo zmožnostjo.

Razlog, da se vračamo pomenskemu načinu kodiranja, je vse večja raznolikost programja, ki se uporablja za pristop do spletnih informacij. Če primerjamo izgled iste strani na različnih čitalcih (ali celo istih brskalnikih različne verzije ali na različnih platformah), postane jasno, da se rezultirajoči prikazi zelo razlikujejo. Namesto vgrajevanja specifikacij prikaza v vsebino, je boljša rešitev ločevanje vsebine in oblike.

5.1.4. Odzivni čas

Najvažnejši kriterij pri oblikovanju spletnih strani je odzivni čas. Raziskave na velikem številu sistemov so pokazale, da je potreben odzivni čas za občutek svobodnega sprehajanja skozi informacijski prostor manjši od 1 sekunde pri prehodu iz ene na drugo stran.

Največji odzivni čas ne sme presegati 10 sekund, to je meja do katere zdrži pozornost uporabnika. Jasno je, da mora biti odzivni čas čim krajši, toda treba je paziti, da brskalnik ne odreagira prehitro (na primer pri pomikanju vrstic v dokumentu, kjer ga uporabnik ne more pravočasno zaustaviti, da bi prebral zelen stavek). Poleg hitrosti, je za odzivni čas pomembna tudi enakost. Na žalost se odzivni čas na spletu zelo pogosto menja, tudi neodvisno od same strani, zaradi obremenjenosti strežnika ali omrežja, kar je glavni razlog za slabo voljo uporabnikov.

Zadovoljstvo uporabnikov je odvisno tako od njihovega pričakovanja, kot tudi od učinka. Če se ista operacija včasih izvede hitro, včasih pa počasi, uporabniki ne vedo, kaj naj pričakujejo in zato ne morejo prilagoditi svojega obnašanja, da bi optimizirali uporabo sistema. Če predpostavljajo, da se bo nekaj zgodilo hitro, bodo razočarani, če bo odzivni čas dolg. Ta pojav je eden od razlogov, da je potrebno odstopanje odzivnega časa zmanjšati na minimum. Uporabnikom je potrebno za prenos velikih dokumentov na lokalnih računalnikih pomagati v tej meri, da lahko predvidijo odzivni čas na način, da se zraven povezave označi še dolžino datoteke. Nepisano pravilo je, da je za datoteke, katerih prenos traja več kot 10s v pogojih povprečne propustnosti omrežja, potrebno označiti še velikost.

Boljša spletna uporabnost je posledica stabilizacije odzivnega časa. Na odzivni čas spletne strani vpliva najslabši člen v verigi med strežnikom in brskalnikom, ki se sestoji iz:

- . izhodne moči strežnika,
- . fizične povezave strežnika z medmrežjem,
- . obremenjenost omrežja,
- . kapaciteta povezave uporabnika do medmrežja,
- . hitrost brskalnika in računalnika uporabnika.

Za zmanjšanje velikosti spletne strani, mora biti grafike čim manj, večpredstavnostne efekte je potrebno uporabiti samo v primeru, če s tem dosežemo, da uporabnik bolje razume informacijo. Torej, če odstranimo grafiko, povečamo promet. To je relativno enostavno pravilo.

Na drugi strani pa omejena uporaba grafike pomeni, da bodo strani nezanimive. Mnogo se da storiti z uporabo različnih barv ali kreativno uporabo različnih pisav. Oblika se lahko popravi tudi z uporabo definicij stilov. Če pa že moramo uporabljati grafike, poskušajmo iste čim večkrat izkoristiti. Na spletni strani se lahko ista slika uporablja za več stvari, kot so gumbi, ikone in puščice. Dostikrat lahko isto sliko postavljamo na več strani, s tem da mora imeti slika na vseh straneh isti pomen.

Če upoštevamo osnovna dejstva, lahko pri oblikovanju spletnih strani zaključimo samo eno. Hitrost nalaganja strani je zelo pomemben faktor; v bistvu mora biti hitrost osnovni kriterij.

5.1.5. Povezave

Povezave so najvažnejši del pri oblikovanju spletnih strani. Povezujejo strani in omogočajo uporabnikom, da obiskujejo nova mesta na spletu. Poznamo tri glavne oblike povezav:

- *Strukturne navigacijske povezave*; odredajo strukturo prostora informacije in omogočajo uporabnikom, da preidejo v njegove ostale dele. Tipični primeri so gumbi, ki vodijo na začetno stran in povezave na strani, ki so podrejene tekoči.
- *Asociativne povezave znotraj vsebine strani*; To so ponavadi podčrtane besede ali slike, ki kažejo na stran z detajlnimi informacijami o izvornem tekstu.
- *Lista dodatnih referenc*; Namen teh povezav je, da pomagajo uporabnikom pri iskanju iskane informacije, če se ta ne nahaja na tekoči lokaciji. Dober seznam dodatnih povezav je lahko v veliko pomoč uporabnikom.

Povezave v hipertekstu so vsebovane v tekstu, na katerega uporabnik pritiska z miško. Tekst spletnih povezav ne sme biti predolg, ker uporabniki pogosto preletijo strani samo z namenom, da bi poiskali pripadajoče povezave. Povezave imajo podobno vlogo kot podnaslovi v tiskanem časopisu. Če se za njih uporablja preveč besed, uporabnik ne more naenkrat dojeti njihovega pomena. Samo pojmi, ki so najpomembnejši z vidika prenosa informacij, so lahko spletne povezave.

Najstarejše pravilo spletnega oblikovanja je, da se izogibamo spletnih povezav oblike »pritisni tukaj«. Čeprav naj spletna povezava ne bi bila daljša od 2 do 4 besed, je priporočljivo dodati še tekst, ki ne spada v povezavo, ampak jo bolj natančno opisuje. Priporočila za informacije, ki so lahko vključene v podnapise povezav:

- ime lokacije, na katero povezava kaže (če se razlikuje od tekoče),
- ime podlokacije (če gre povezava na drugi del iste lokacije),
- opis vrste informacij, ki se nahaja na ciljni strani, kot tudi njihov odnos proti tekstu povezave in vsebini tekoče strani,
- opozorila o mogočih problemih na drugi povezavi (npr. obvezna registracija uporabnika).

Podnapisi povezav morajo biti krajši od 80 znakov; brez velike potrebe ni potrebno uporabljati več kot 60 znakov.

Splet je tako počasen, da se od uporabnika ne more pričakovati, da bo obiskal vsako povezavo samo zato, da bi ugotovil kaj je na drugi strani. Na srečo novejši brskalniki omogočajo prikaz kratke vsebine povezave, ki se prikaže še preden uporabnik to povezavo dejansko izbere.

5.1.6. Zbirka slogov

Zbirka slogov (angl. *cascade style sheet* – CSS) predstavlja največje upanje za doseganje ideala razdvajanja postavitve in formata spletne strani od njene vsebine. Glede na to, da je lahko spletna vsebina prikazana na velikem številu različnih naprav, mora označevalni jezik

definirati samo vsebino, ostalo pa prepusti zbirki slogov ali osebnim nastavitvam uporabnika. Z zbirko slogov definiramo lokacijo, format in druge lastnosti elementov na spletni strani. Priporoča se, da se za celotno spletišče uporablja samo eno zbirko slogov. Poznamo dva načina za implementacijo zbirke slogov:

- vgrajena definicija kot del strani,
- . povezana definicija, ki se hrani v posebni datoteki.

Pri tem mora vsaka spletna stran vsebovati povezavo, ki kaže na to datoteko. Še nekaj priporočil za uporabo zbirke slogov:

- v zbirko slogov je poleg nekaj osnovnih definicij pisav priporočljivo vključiti še velik spisak alternativnih,
- . velikost pisave se podaja glede na osnovno vrednost, ki jo je postavil uporabnik.

Vzrokov zakaj imajo ljudje radi tako velike kot male pisave, je kar nekaj. Med njimi so bodisi različne velikosti in ločljivosti zaslonov za prikaz informacij bodisi različna kvaliteta vida uporabnikov in dejstvo ali stran prikazujejo drugim ali sebi. Zelo je neprijetno, če obiščemo spletno stran, na kateri je tekst tako majhen, da se ne more brati, ali še slabše, da se nič ne zgodi, če pritisnemo »povečaj tekst«, ker je velikost pisave definirana kot absolutna vrednost [2].

5.1.7. Okvirji

Osnovno priporočilo je: »Okvirji? Ne hvala.«.

Splet je v osnovi koncipiran na način, da vsaka stran predstavlja nedeljivo enoto informacije, pojem strani pa prehaja v vse aspekte spleta. Enostavnost prvotnega spleta je v veliki meri prinesla lahkotnost uporabe in hitro popularnost.

Okvirji razbijajo enotni model in uvajajo nov način pogleda na predstavljene podatke. Z okvirji je pogled uporabnika nad podatki določen z nizom navigacijskih akcij namesto z eno. Navigacija z okvirji zato ni dobra, ker se enota navigacije razlikuje od enote pogleda. Če uporabniki v brskalniku označijo neko lokacijo, ta namreč ne vsebuje stanja okvirjev. Ko je kasneje ta spletna stran ponovno izbrana, obstaja velika verjetnost, da bo njen videz drugačen, kot smo pričakovali. Dodatne težave, ki so možne, se pojavijo pri tiskanju, saj ima večina tiskalnikov probleme pri tiskanju strani, kjer so prisotni okvirji. HTML 4.0 je uvedel nov tip okvirjev (angl. *inline frame*). Kar se tiče uporabnika, je z uporabo teh okvirjev ohranjen osnovni model strani; zanj je nevidno, da se del strani polni z vsebino iz zasebne datoteke, neodvisno od glavne vsebine. Taki okvirji so odlična rešitev za vnos navigacijskih gumbov, ker se lahko prepreči premikanje njihove vsebine, berejo pa se samo enkrat.

5.2. Vsebina spletnih strani

Poleg informacij, ki jih je potrebno na ustrezen način posredovati uporabniku, moramo zagotoviti še nekaj dodatnih pogojev, ki so zelo pomembni z vidika dostopnosti. Značilnost spletnih strani, ki imajo visoko stopnjo uporabnosti, je tudi ustrezno urejena spletna dokumentacija.

5.2.1. Vsebina navigacije

Za strukturne navigacijske povezave, ki se nahajajo na spletnih straneh, velja pravilo, da uporabniki ponavadi izbirajo samo tiste opcije, ki so vidne na ekranu. Torej se zelo priporoča, da se vse razpoložljive opcije vidijo naenkrat. Uporabnik se v nasprotnem primeru ne more takoj odločiti, katera bo njegova naslednja akcija. Povečanje obremenitve spomina uporabnika je vedno slabo z vidika uporabnosti in povečuje možnost napake.

5.2.2. Čitljivost

Obstaja nekaj osnovnih pravil, ki jih je potrebno upoštevati, da zagotovimo čitljivost spletne strani:

- barva ozadja in barva teksta morata biti v velikem kontrastu. Čitljivost drastično pade, če je tekst svetlejši od čiste črne barve, še posebej, če je ozadje temnejše od bele,
- ozadje mora biti enobarvno ali s prisotnostjo blage teksture. Slike v ozadju motijo oko pri prepoznavanju linij črk in oblike besed,
- velikost črk mora biti zadostno velika, da tekst lahko berejo tudi ljudje, ki nimajo popolnega vida,
- tekst mora biti nepomičen. Pomikanje, trepetanje ali povečava teksta znatno otežuje branje.

5.2.3. Spletna dokumentacija

Uporabniški vmesnik je potrebno oblikovati tako, da je lahek za učenje. S tem dosežemo, da uporabniki posegajo po dokumentaciji samo v redkih primerih.

Uporabniki medmrežja pa so v večini primerov tako nestrpni, da ne bodo hoteli brati prav nobene dokumentacije. Nekaj več časa bodo pripravljene žrtvovati za učenje pri uporabi interaktivnih spletnih strani, ki jih stalno uporabljajo, saj bodo lahko s tem določene akcije izvedli veliko bolj učinkovito. Zato je potrebno omogočiti priročnike in spletno podporo tudi za zahtevne interakcije.

Osnovna pravila za pisanje spletne dokumentacije:

- ker se uporabniki ponavadi obračajo k dokumentaciji samo takrat, ko se soočijo s problemom, je osnovno vodilo, da se dokumentacija enostavno preiskuje,
- vsebovati mora veliko primerov. Uporabniki veliko lažje spremljajo primere in jih tako prilagajajo svojim problemom, kot da si s pomočjo splošne obrazložitve sami izmišljujejo, kaj je potrebno storiti,
- navodila morajo biti orientirana k specifičnim nalogam in morajo opisovati, kako se določeni problemi rešujejo po posameznih korakih. Ni potrebno izgubljati preveč besed za uvodne informacije, ker jih uporabniki verjetno ne bodo prebrali,
- vsak koncept ali težko razumljive besede je potrebno povezati s pojmovnikom,
- kot vedno na medmrežju - bodimo kratki.

Obstajajo določeni kriteriji, ki ločijo slabo oblikovano navigacijo od tiste, ki omogoča lagoden neomejen dostop do posameznih virov.

5.3. Oblika lokacije

Splet je navigacijski sistem. Osnovna uporabniška interakcija je pritisk na spletno povezavo, s katero se uporabnik pomika skozi ogromni informacijski prostor.

5.3.1. Navigacija

Navigacijski vmesnik mora pomagati uporabniku, da odgovori na tri osnovna vprašanja v zvezi z navigacijo:

1. Kje se nahajam?
2. Kje sem bil?
3. Kam lahko grem?

Kje se nahajam?

Uporabniki nimajo možnosti dojeti strukturo lokacije, če ne vedo kje so. Trenutno pozicijo uporabnika je potrebno prikazati na dveh različnih nivojih:

- v odnosu na splet kot celoto,
- v odnosu na strukturo lokacije.

Navigacijsko pravilo št. 1 je, da se na vsaki strani nahaja logotip (ali nek drug identifikator lokacije). Logotip mora imeti stalno mesto in mora vsebovati tekstovno spletno povezavo na začetno stran. Relativni položaj v odnosu na strukturo lokacije se običajno prikazuje tako, da se poda slika lokacije na kateri se poudari del v katerem se nahaja tekoča stran. Zelo je pomembno, da ima vsaka stran jasen glavni naslov, ki vsebuje ime ali kratko opisuje vsebino.

Kje sem bil?

Zaradi omejitev standardnih medmrežnih tehnologij, ki ne morejo spremljati stanja, je na to vprašanje težko odgovoriti. Informacija je namreč brez uporabe piškotkov in ostalih meril za spremljanje uporabnika nedostopna. Na srečo si lahko uporabnik pomaga s tipko »Back«, ki ga vrne na predhodno stran, obstaja pa tudi spisek že obiskanih strani, kjer so že obiskane povezave, prikazane v drugi barvi. Priporoča se, da se standardna barva povezav ne menja, ker bodo uporabniki samo tako dojeli njihov pomen.

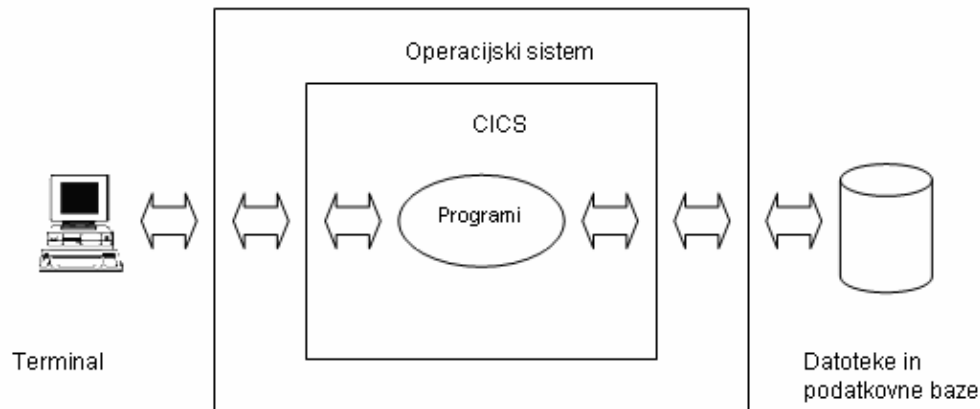
Kam lahko grem?

Odgovor na to vprašanje dajo vidne opcije za navigacijo in ostale povezave na strani. Poleg tega gre lahko uporabnik ob predpostavki, da je osvojil osnovno poznavanje strukture lokacije, tudi na mesta, ki so na spletni strani skrita v nadaljevanju dokumenta in niso takoj vidna.

6. CICS

6.1. Kaj je CICS

CICS je večnamenski podatkovni komunikacijski sistem, ki lahko podpira mrežo več terminalov. Lahko si ga predstavljamo bodisi kot posebni operacijski podsistem ali okolje za izvrševanje online aplikacijskih programov bodisi kot tudi vmesnik za dostop do datotek in baz podatkov. Kot je razvidno s Slike 8, je CICS vmesnik med OS in terminalom [3].



Slika 8: Osnovni prikaz CICS-a

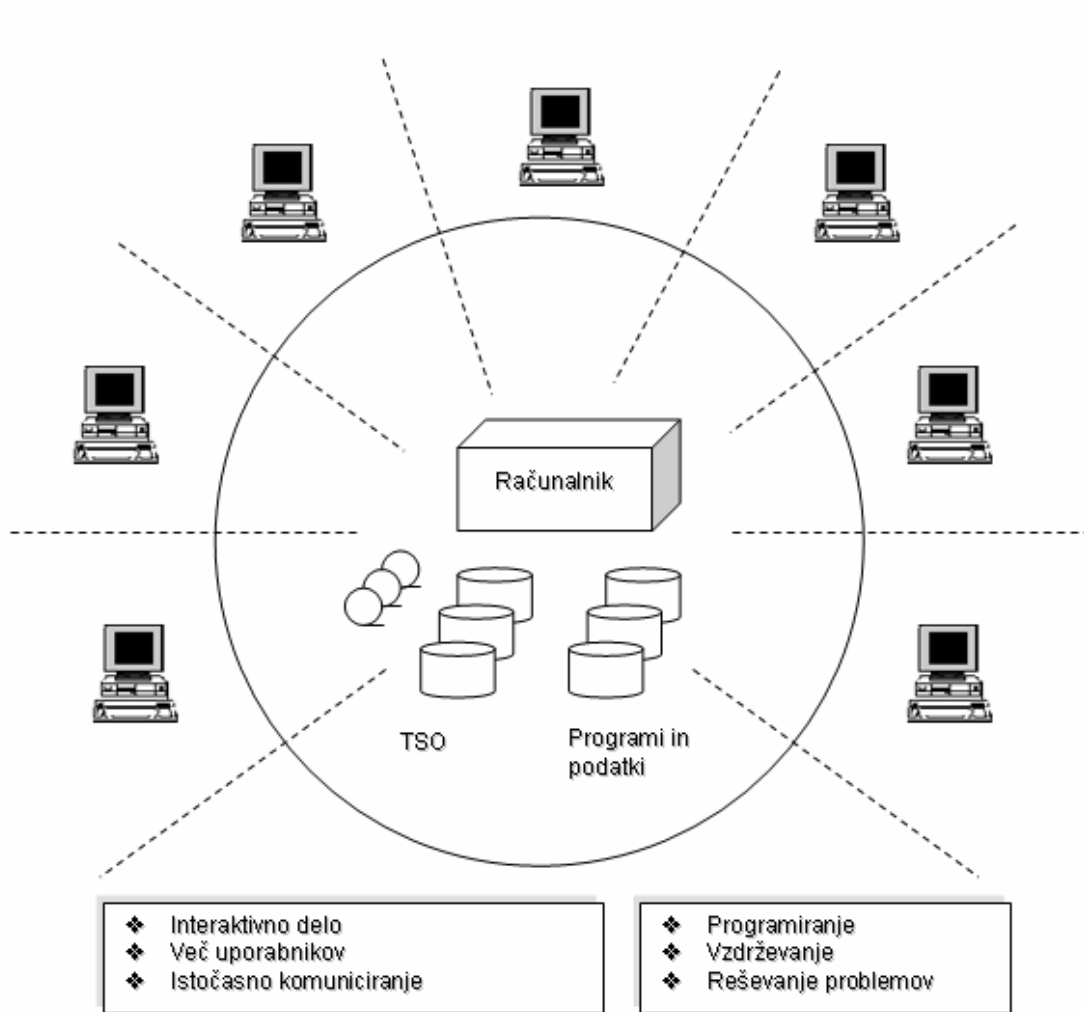
6.1.1. Potreba po interaktivnem dostopu

Z vidika operaterja morata biti obdelava podatkov in odziv na terminalu čim hitrejša, da bi lahko naredil čim več dela. Zaradi relativne počasnosti operaterja glede na sistem, je nesprejemljiva programska obdelava ene same naloge v nekem trenutku. Program naj ne odloča, kdaj se bodo podatki vnašali, ampak naj podatki določajo kdaj jih lahko program obdeluje.

Končni online uporabniki, ki so povezani v mreži, lahko zahtevajo različna opravila na različnih datotekah. Postopki, ki jih zahtevajo, so ponavadi kratki. Različni uporabniki si med seboj delijo iste datoteke ali programe, izmenjave pa so sestavljene iz več zahtev in odgovorov.

CICS zagotavlja:

- funkcije, ki jih zahteva program za komunikacijo s terminali in podsistemi,
- kontrolo nad hkrati delujočimi programi,
- dostop do datotek in podatkovnih baz,
- možnost komunikacije med različnimi podatkovnimi bazami in CICS-i.



Slika 9: CICS – več uporabnikov

Transakcija je postopek, ki ga sproži posamezna zahteva in jo izda operater. V večini primerov je sestavljena iz enega ali več aplikacijskih programov, ki so medsebojno povezani in na koncu izpolnijo podano zahtevo.

Opravilo (angl. *task*) je posamezna izvršitev iste transakcije. CICS upravlja z več opravili istočasno, vendar v danem trenutku izvaja samo enega. TCP (Task Control Program) je glavni upravljalni modul, ki prevzema kontrolo neposredno od operacijskega sistema in odloča kateremu opravilu bo predana kontrola (odvisno od prioritete) in komu se vrne kontrola po zaključku opravila ali njegovega čakanja na konec [3].

6.2. CICS transakcijski vmesnik

CICS transakcijski vmesnik (angl. *CICS Transaction Gateway*) je vodilni povezovalc na trgu, ki nam omogoča visoko učinkovitost, varnost, stabilnost in integrirano metodo e-poslovnega dostopa do CICS-a. CICS uporabnikom omogoča razširiti vrednost njihove investicije v klasični CICS aplikaciji ter ponavadi ne zahteva nobenih sprememb do obstoječih CICS aplikacij. Preprosto ga je namestiti in ima prilagodljive nastavitvene možnosti, katere zahtevajo manjše spremembe na CICS-u. Nudi nam širok spekter mrežnih rešitev in omogoča delo z javanskimi oziroma ne-javanskimi programskimi vmesniki .

CICS transakcijski vmesnik je skupek odjemalca in strežnika, ki nam omogoča oddaljeno odjemavanje programov, ki kličejo podporo v CICS-u. Odjemalci programov so lahko cobolski ali drugi programski vmesniki, kot so Java,C, C++,ali COM [4].

```

SET PROG(P86*)
STATUS: RESULTS - OVERTYPE TO MODIFY
  Prog(P86ADONI) Leng(0000019952) Cob Pro Ena Pri Ced
    Res(000) Use(0000000168) Any Uex Ful Qua Cic Len
  Prog(P86ERROR) Leng(0000017464) Cob Pro Ena Pri Ced
    Res(000) Use(0000000059) Any Uex Ful Qua Cic Len
  Prog(P86FKATA) Leng(0000022040) Cob Pro Ena Pri Ced
    Res(000) Use(0000000536) Any Uex Ful Qua Cic Len
  Prog(P86FNBSF) Leng(0000015536) Cob Pro Ena Pri Ced
    Res(000) Use(0000000006) Any Uex Ful Qua Cic Len
  Prog(P86FPKDB) Leng(0000018624) Cob Pro Ena Pri Ced
    Res(000) Use(0000000002) Any Uex Ful Qua Cic Len
  Prog(P86FPKON) Leng(0000017360) Cob Pro Ena Pri Ced
    Res(000) Use(0000000002) Any Uex Ful Qua Cic Len
  Prog(P86FPPDB) Leng(0000026688) Cob Pro Ena Pri Ced
    Res(000) Use(0000000053) Any Uex Ful Qua Cic Len
  Prog(P86FPPRO) Leng(0000016936) Cob Pro Ena Pri Ced
    Res(000) Use(0000000002) Any Uex Ful Qua Cic Len
+ Prog(P86FPRSN) Leng(0000033768) Cob Pro Ena Pri Ced
    Res(000) Use(0000000004) Any Uex Ful Qua Cic Len

                                SYSID=86A1 APPLID=CR86A01
RESPONSE: NORMAL                                TIME: 12.25.19 DATE: 12.02.08
PF 1 HELP          3 END          5 VAR          7 SBH 8 SFH 9 MSG 10 SB 11 SF

```

Slika 10 : Prikazuje trenutno stanje CICS programov

Predajanje parametrov (podatkov) med programi

Primer kode programa, ki kliče:

IDENTIFICATION DIVISION.
PROGRAM-ID. 'SAM-0'.

WORKING-STORAGE SECTION.
01 TRANS-INFO.
02 FLD1 PIC X(4).

PROCEDURE DIVISION.
MOVE 'INFO' TO FLD1.
EXEC CICS LINK PROGRAM ('SAM-0') COMMAREA(TRANS-INFO)
LENGTH(4) END-EXEC.

Primer kode klicanega programa:

IDENTIFICATION DIVISION.

PROGRAM-ID. 'SAM-1'.

LINKAGE SECTION.

01 DFHCOMMAREA.

02 FLD2 PIC X(4).

PROCEDURE DIVISION.

IF EIBCALEN > ZERO THEN

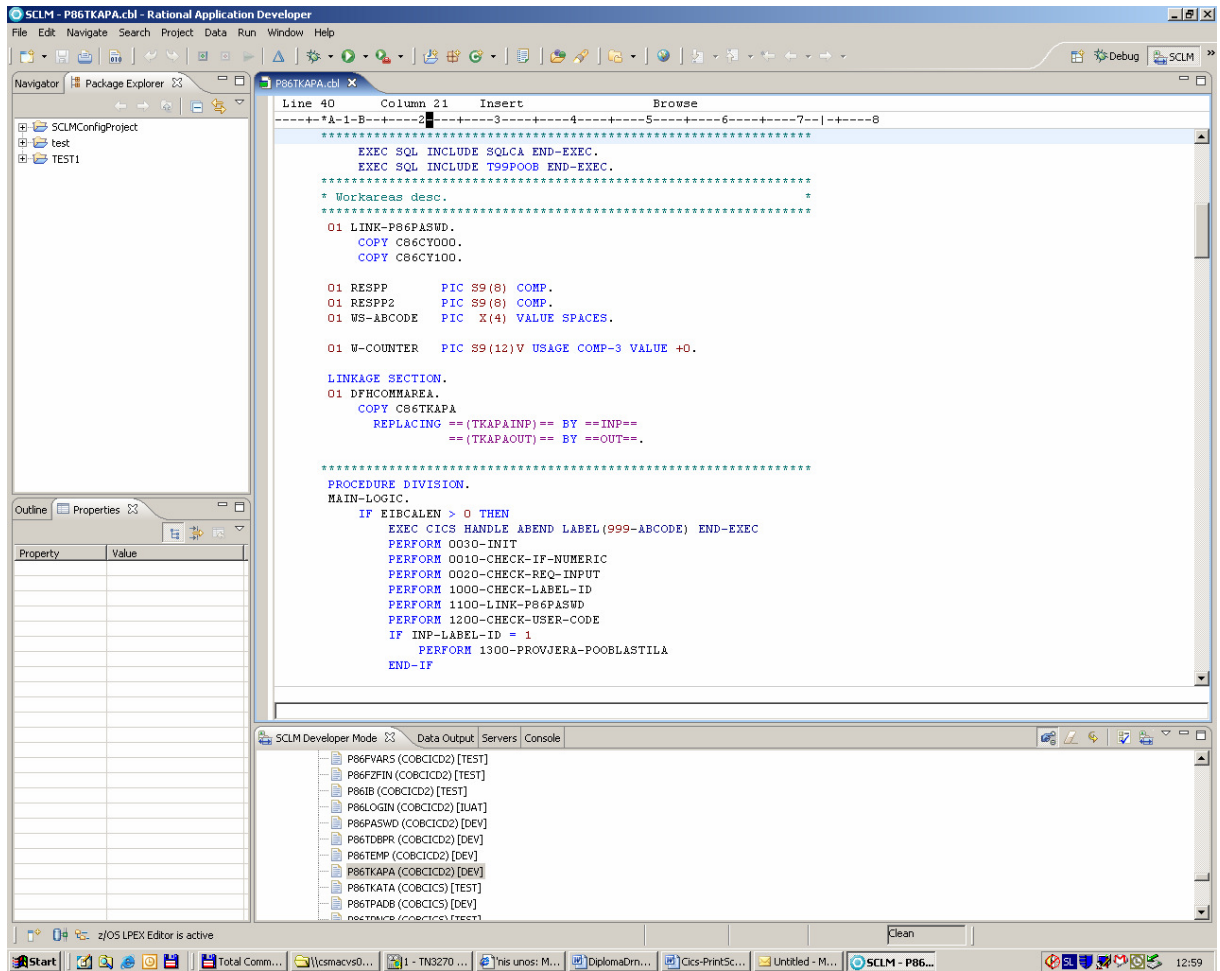
IF FLD = 'INFO' THEN

Pogosto prenašamo parametre med programi pri uporabi Program Control ukazov LINK in RETURN. V programu, ki kliče, je potrebno kodirati komunikacijsko področje (angl. *COMMAREA*), ki v bistvu predstavlja drugo ime za področje podatkov. V klicanem programu so podatki sprejeti kot parametri. Zaradi tega mora tudi v tem programu obstajati definicija področja podatkov, da bi bil mogoč pristop do njih. Nahaja se na prvem nivoju LINKAGE sekcije z rezerviranim imenom DFHCOMMAREA. Dolžina komunikacijskega področja programa, ki kliče, je shranjena v spremenljivki EIBCALEN (Exec Interface Blok).

Primer ukaza RETURN:

```
EXEC CICS RETURN [TRANSID (ime) [COMMAREA (ime)
LENGTH (xxxx)]] END-EXEC.
```

Ukaz RETURN preda kontrolo programu na višjem nivoju hierarhije ali CICS-u, opcija TRANSID pa pove, katero transakcijo je potrebno zagnati po sprejemu vhoda s terminala vezanega na ta opravilo.



Slika11: Prikazuje CISICS DB2 program v RAD-u

7. MQ

Kratek opis ključnih pojmov za nadaljnjo razlago MQ Series:

Z/OS: je IBM-ov produkt 64 bitnega operacijskega sistema za »mainframe« računalnike in nova inačica OS/390.

DB2: je sistem za upravljanje baz podatkov. Baza podatkov je urejena zbirka medsebojno povezanih podatkov, ki je shranjena na računalniškem nosilcu podatkov. Poznamo hierarhične, mrežne, relacijske, objektne in večdimenzionalne podatkovne modele. Za delo z bazami uporabljamo SQL jezik.

MVS (Multiple Virtual Storage): večopravilni OS, ki je sposoben hkrati vzdrževati več naslovnih prostorov z navideznimi naslovi. Zagotavlja razpoložljivost virov (resursov) in omogoča organiziranje podatkov na sekvenčni način ali z direktnim dostopom. V osnovi je MVS prilagojen predvsem »batch« obdelavi, omogoča pa tudi online delo preko podsistema za neposredno delo (transakcijski monitor). Eden takih je MVS-ov podsistem CICS.

Transakcija: je postopek, ki sproži posamezno zahtevo in jo izda operater. V večini primerov je sestavljena iz enega ali več aplikacijskih programov, ki so medsebojno povezani, in na koncu izpolnijo podano zahtevo.

7.1. Kaj je MQ in kaj so njegove najpomembnejše lastnosti?

MQ Series je torej IBM-ov produkt za prenašanje podatkov med različnimi platformami, pri katerem je potrebno izpostaviti nekaj njegovih najpomembnejših lastnosti po katerih se razlikuje od podobnih produktov oz. združuje njihove lastnosti.

MQ se uporablja za pošiljanje sporočil. Deluje podobno kot FTP (vendar ima več dodanih vrednosti). MQ ni medij za shranjevanje podatkov (Slika 12).

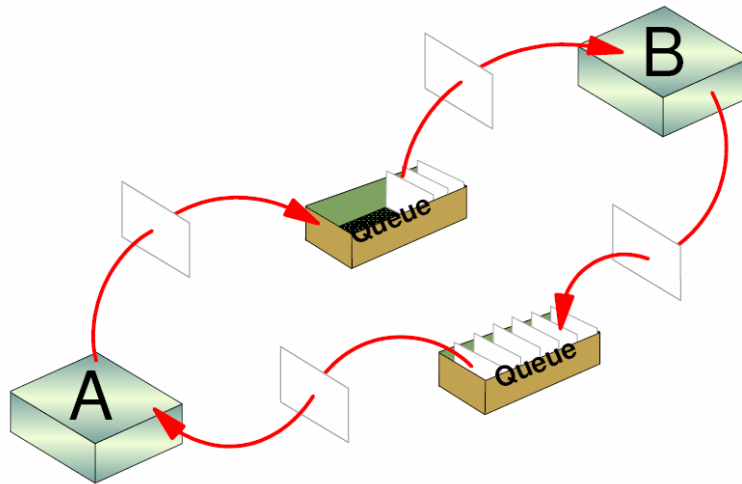
Asinhrona povezava (časovna neodvisnost - ni potrebe, da je ciljni sistem ob prenosu dosegljiv).

Potrdi (angl. *Commit*) in rezveljavitev (angl. *RollBack*)- ko DB2 program zaključi delo oz. v primeru zahteve ali napake izvede rezveljavitev, kar pomeni povrnitev virov v začetno stanje.

Proženje (angl. *Triggering*) - ko določeno število sporočil prispe v Q se lahko sproži obdelava ali proces, definira ga MQ Administrator.

Sporočila se ne izgubljajo ali podvajajo. Podprt praktično za vse platforme, tako omogoča večjo povezljivost med različnimi okolji in podatkovnimi bazami.

Persistenca – beleženje sporočil (restavriranje v primeru prekinitve MQSeries – ne pa obdelave ali programa) [5].



Slika12: Prikazuje delovanje MQ -ja

7.1.1. Zakaj v banki potreba po MQ Series?

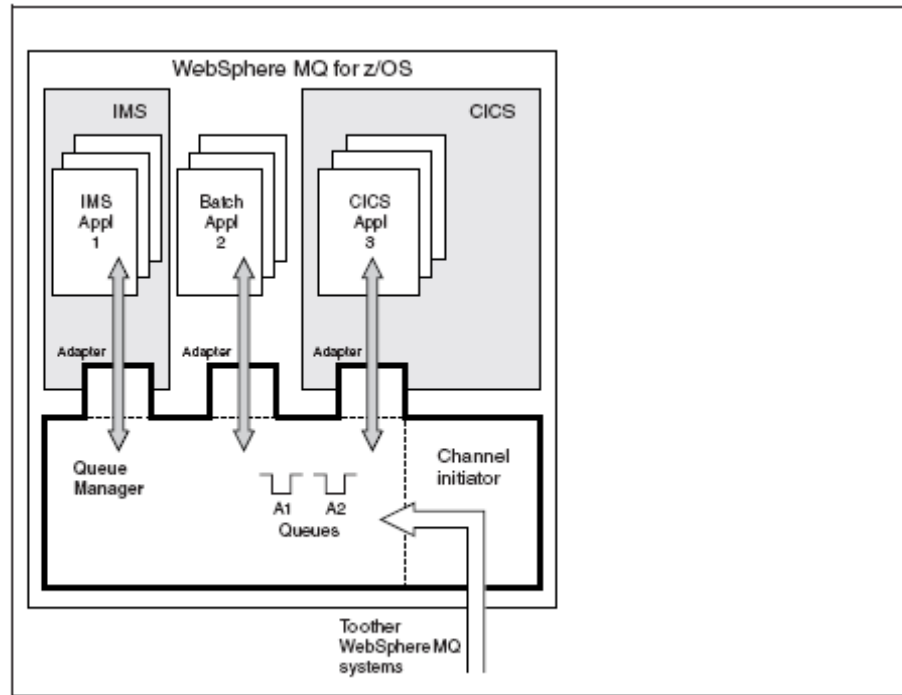
FTP pokriva 95% vseh prenosov datotek; je dober in izpopolnjen, vendar se v bančne namene z vidika sigurnosti, varnosti (sistem pooblastil) in nadzora uporablja MQ Series. MQ omogoča tudi arhiviranje; v 7 letih sta bila le dva vpogleda v arhiv z namenom dokazati, da je bilo neko sporočilo res poslano.

7.1.2. MQ vloge za delo v banki

MQ Administratorji namestijo MQ aplikacijo, postavijo Q-managerje in kreirajo objekte. Skrbijo za učinkovito delovanje in na splošno za razne probleme na teh področjih. Razvojniki razvijejo ali uporabijo programe za komunikacijo z MQ. Za varnost skrbi TOP Secret kot del Host okolja.

7.1.3. MQ na Z/OS

Na Z/OS lahko pišemo aplikacije, ki uporabljajo MQ z CICS-om ali na IMS-u. Lahko preko MQ-ja pošljamo sporočila med »batch«, CICS in IMS aplikacijami. Prav tako lahko pošljamo sporočila na druga okolja npr. Windows. Na Z/OS lahko sporočila, ki so npr. del datoteke, na koncu uspešno izvedene operacije, potrdimo (Commit) ali pa jih v primeru napake povrnemo v začetno stanje (Roll-back). Obstaja več omejitev katere moramo poznati, npr.: CICS okolje ima lahko povezavo samo z enim Q-managerjem hkrati, na »batch« in IMS okolju pa je lahko povezava na več Q-managerjev hkrati, omejitve so tudi v velikosti sporočil, katere prenašamo in se z verzijami spreminjajo (izboljšujejo). Na Z/OS obstajajo tudi knjižnice z definicijskimi podatki, ki nam olajšajo delo pri klicanju konstant, obstajajo pa tudi sistemski programčki, ki nam olajšajo delo pri aplikacijah.



Slika 13: MQ na z/OS

7.1.4. MQI (Message Queuing Interface)

MQI sestoji iz klicev in struktur, preko katerih programi pošiljajo in prejemajo sporočila iz Q-ja. Poznamo več klicev in sicer: MQCONN, MQCONNX, MQOPEN, MQPUT, MQPUT1, MQGET, MQCLOSE, MQDISC, MQINQ, MQSET, MQBEGIN, MQCMIT, MQBACK. V večini uporabljamo samo določene klice, programska logika večine MQ batch programov je:

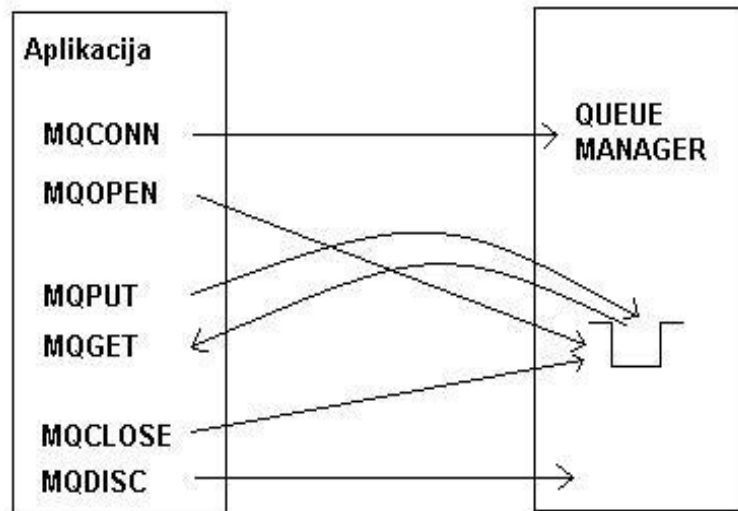
Najprej se je potrebno povezati s Q-managerjem – MQCONN (Batch aplikacije), nakar dobimo odgovor v MQ »Reason« in »Completion« kodi ali je povezava uspela ali ne in dobimo vrnjen podatek o »conn-handle«, katerega potrebujemo za nadaljnje delo.

Če je uspela, nadaljujemo s povezavo na MQ objekt torej Q – MQOPEN (obstaja več drugih objektov kot so npr. kanali, listenerji, različne vrste q-jev.). Q-managerju posredujemo določene parametre za povezavo in sicer ime Q-ja kot »object descriptor«, »open options«, »conn-handle«, vrnjen pa dobimo odgovor v kodah enako kot pri MQCONN klicu.

Če nam Q uspe odpreti, nam ponovno vrne podatke, kot že omenjena MQ koda o uspešnosti povezave oz. o nastali napaki in »conn-handle«. Ostale podatke o našem sporočilu pa bomo posredovali z naslednjim klicem in sicer MQPUT (MQGET) in sicer z opcijama »open-options« in »message-descriptors«. S temi dvema opcijama opišemo lastnosti našega sporočila in način dostave, trajanja, pomembnosti. Nakar določimo vsebino sporočila in ga s tem klicem pošljemo na Q. Nepravilne nastavitve nam lahko povzročajo probleme in povzročajo napake, katere razberemo iz MQ »Reason-Code«.

Ko prenesemo sporočila, pa zapremo objekt, torej Q – MQCLOSE in ponovno dobimo vrnjene MQ kode, kot pri vseh klicih. Se odjavimo od Q-managerja- MQDISC (Slika 14).

Programska logika MQ CICS programov se malce razlikuje; ni potrebno izvesti klicev MQCONN in MQDISC, kajti določen CICS strežnik je ponavadi že povezan s Q-managerjem. Prav tako za Commit in Roll-back skrbi CICS aplikacija.



Slika 14: MQ klici

7.2. PREDSTAVITEV MQ Series programov

7.2.1. MQ (batch) programi za prenose podatkov

P99QPUT in P99QGET – za prenos sekvenčnih datotek (št. vrstic je št. Q sporočil) – prenos v tekstovnem načinu

P99QPUT1 in P99QGET1 – za prenos datotek (na začetku glava 1k) → sistem prenosa datotek po principu FTP - prenos v tekstovnem načinu

P99QGET2 – za prenos stisnjenih datotek (ZIP, TERSE - prenos v binarnem formatu)
 P99QGETG- za pobiranje sporočil iz Q-ja določene grupe oz. pošiljatelja (z »group-id« ločimo sporočila različnih pošiljateljev na enem Q-ju).

Procedura za brisanje z zagonom JOB-a JMQEMPTD - za čiščenje vseh Q-jev določenega Q managerja (določimo lahko naslednje parametre: Q-managerja, enega ali več Q-jev, čas v dnevih za brisanje sporočil, ki so starejša od nastavljenih dni). Ta program je moje izdelave in je sestavljen iz štirih programov.

7.2.2. Procedura za brisanje z zagonom JOB-a JMQEMPTD

Procedura je sestavljena iz več programov, ki v zankah ali po zaključku enega kličejo drugega. Programi so: P99BQZ, P99QINPD, P99QDEL.. Za zagon programa uporabimo JOB JMQEMPTD. Poleg tega program uporablja knjižnice C99INP, C99OUT in že prej omenjene knjižnice za MQ, ki so: CMQDV, CMQODV, CMQGMV, CMQV. Prav tako uporabi modula (programa) S99INP in S99OUT.

Procedura najprej kliče program, ki v datoteko izpiše vse Q-je določenega Q-managerja (katerega skupaj z ostalimi parametri določimo v JOB-u), vendar poleg tega izpiše še kar nekaj sistemskih podatkov, katerih ne potrebujemo. Program P99BQZ izloči vse nepotrebne podatke in zapiše v datoteko samo imena Q-jev. Program P99QINPD iz datoteke prebere ime Q-ja in pregleda vsebino celotnega Q-ja, nakar pa glede na v JOB-u nastavljeno število dni, pobriše vsa sporočila, ki so starejša, nato program P99QINPD v zanki kliče program P99QDEL in mu v LINKAGE sekciji preda določene parametre. Ko program P99QDEL opravi delo v enem Q-ju, se v zanki vrne nazaj na program P99QINPD, ki prebere ime novega Q-ja in ponovi postopek, toliko časa dokler ne pregleda vseh Q-jev in njihove vsebine. Ob koncu izvajanja programa se vse sekvenčne datoteke, ki so se ustvarile, pobrišejo, na konzolo pa se nam izpiše »display«, ki nam javi koliko sporočil je bilo pobrisanih na posameznem Q-ju.

7.2.3. JOB JMQEMPTD

```
//APLDRFQD JOB ,MSGLEVEL=(1,1),MSGCLASS=X,NOTIFY=&SYSUID
//*****
/* PROGRAM ZA BRISANJE SPOROČIL IZ Q-JEV,
/* KI SO STAREJSA OD NASTAVLJENIH DNI.
//*****
//JOB LIB DD DSN=APLDRF.USER.LOAD,DISP=SHR
// DD DSN=R99.USER.LOAD,DISP=SHR
// DD DSN=MQM.SCSQANLE,DISP=SHR
// DD DSN=MQM.SCSQAUTH,DISP=SHR
// DD DSN=CEE.SCEERUN,DISP=SHR
//*****
/* PRVI KORAK IZPIŠE VSE Q-JE, TER DRUGE LASTNOSTI
//*****
/* -----
/* 1.POD PARM NASTAVITE ŽELJENI Q-MANAGER
/* 2.ZA QLOCAL VPIŠETE IME Q-JA ALI 'R*'ZA VSE Q-JE
/* -----
//STEP1 EXEC PGM=CSQUTIL,PARM='CRQ1'
//OUTPUT1 DD DSN=&&QLOCALD1,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(5,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(CMDINP) MAKEDEF(OUTPUT1)
/*
//CMDINP DD *
DISPLAY QLOCAL(R18_HM_NBO) USAGE
/*
//*****
/* DRUGI KORAK POPRAVI ZAPIS NA ZAPIS SAMO Q-JEV
```



```

//*****
//STEP2 EXEC PGM=P99BQZ
//PGMINP1 DD DSN=&&QLOCALD1,
// DISP=(OLD,DELETE)
//PGMOUT1 DD DSN=&&QLOCALD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(5,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)
//SYSABOUT DD SYSOUT=*
//SYSDBOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//*****
/* TRETJI KORAK BERE Q-JE V ZANKI IN
/* KLIČE PROGRAM ZA BRISANJE
/* SPOROČIL NA (VSEH) TEH Q-JIH
//*****
/* -----
/* 3.POD PARM NASTAVITE STEVILO DNI,
/* SPOROCILA, KI SO STAREJSA SE BRISEJO.
/* PARM MORA BITI NUMERICEN PODATEK DOLZINE 3!
/*-----
//STEP3 EXEC PGM=P99QINPD,PARM='001'
//PGMINP1 DD DSN=&&QLOCALD,
// DISP=(OLD,DELETE)
//SYSABOUT DD SYSOUT=*
//SYSDBOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
/*

```

7.3. MQ (online) programi za transakcije

P21HIBIS – program za poizvedbo o stanju na računih,

P21HKIVI – program za transakcije (dvig, polog, storno).

Programa samega ne bom prikazal, zato pa bom na kratko opisal sam potek proženja akcij za lažje razumevanje.

7.3.1. Kratak opis poteka proženja akcij in opis poti podatkov

Oba omenjena programa sta naložena na CICS-ovem sistemu (podsistem na Z/OS), prvi je direktno na CR52Q01, za drugega pa je narejena oddaljena definicija na istem sistemu, vendar se nahaja na CR98D01. Program spada v grupo programov in je klican, če programu T21TRAN podamo parameter kontrola = 2, to pomeni banko Celje.

Praviloma program zažene uporabnik na terminalu, bančnem okencu ali bankomatu.

Uporabnika je bilo za potrebe programiranja in preizkušanja delovanja programa potrebno simulirati. To je narejeno s pomočjo ASP programa, ki preko ECI link klicev iz spletne strani kliče CICS program in mu posreduje podatke katere program zahteva, nakar naredi poizvedbo (poizvedba opisana posebej) in nam vrne podatke (podatki med programi se prenašajo v DFHCOMMAREA), ki jih vrne ASP programu v Windows okolju (potrebno je izvajati tudi konverzijo znakov zaradi različnih okolij Host-Windows). ASP program je dejansko spletna stran z Visual Basic programsko kodo v ozadju in vnosi podatke, katere simulirani

uporabnik vnese. ASP programe pa »ženemo« preko IIS (Internet Information Service) v Windows okolju.

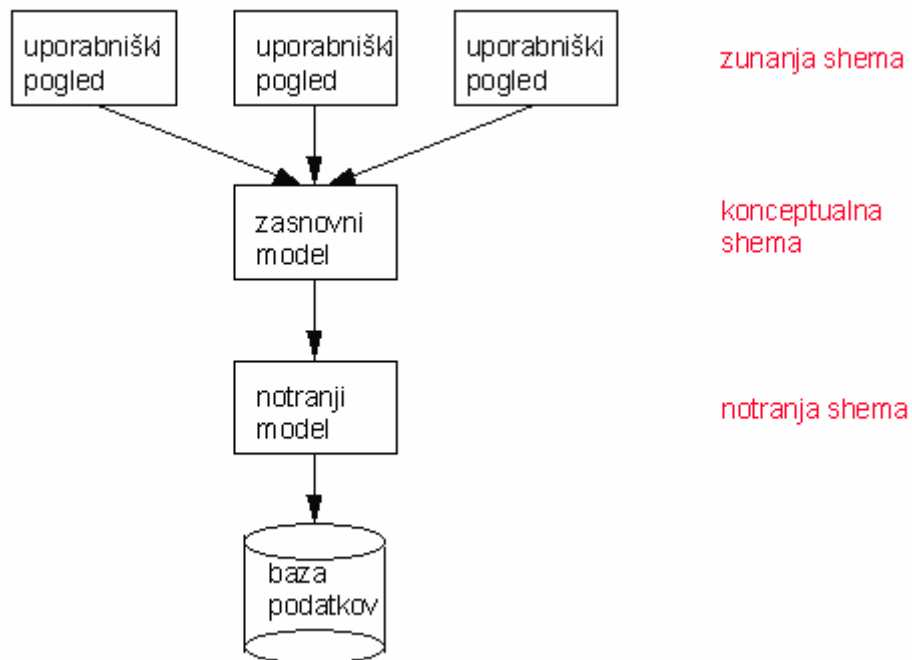
Poizvedbo katero naredimo preko ASP spletne strani in programa na CICS-ovem podsistemu pa proži akcija sporočila poslanega v oddaljen Q (je definicija Q-ja na nekem drugem Q-managerju, ponavadi na nekem drugem okolju, podjetju), ki ima (lahko) nastavljen sprožilec na vsako poslano sporočilo (sprožilce lahko nastavimo tudi drugače). Ko je sporočilo torej poslano v Celjski Q (za nas oddaljeni Q), se na celjski strani sproži proces, ki prevzame sporočilo in ga obdela, ter nam pošlje nazaj obdelane podatke in na naši spletni strani preko ECI link-a od CICS programa na ASP program dobimo rezultat na našo poizvedbo.

8. DB2

Baza podatkov je urejena zbirka medsebojno povezanih podatkov, ki je shranjena na računalniškem nosilcu podatkov. Poznamo hierarhične, mrežne, relacijske, objektne in večdimenzionalne podatkovne modele.

Baza podatkov IBM DB2 je relacijska baza podatkov, za katero je značilna tri nivojska zgradba (Slika 15):

- zunanja shema,
- konceptualna shema,
- notranja ali fizična shema.



Slika 15: Tri nivojska zgradba baze podatkov

Tabela je določena z imenom in obsega poljubno število stolpcev ter poljubno število vrstic. Stolpci so ravno tako določljivi le z imenom. V okviru vsakega tipa podatkov obstaja tudi nedefinirana oziroma ničelna vrednost. Več logično povezanih tabel zberemo v bazo podatkov. Sistem za upravljanje podatkovnih baz lahko upravlja z več tovrstnimi bazami.

Pogoj, da se ustvari baza podatkov, je vsaj ena tabela, kateri moramo fizično določiti količino prostora na disku (table space). Vsak table space pa ima tudi svoje unikatno ime, prav tako kot tudi vsaka tabela. Poleg imena se pripne še uporabniško ime, ki služi za identifikacijo tabele (kdo je tabelo kreiral).

Prostor, ki predstavlja vrstico, se imenuje zapis, prostor, ki predstavlja stolpec, pa se imenuje polje. Za večino nalog ni potrebno vedeti kako zapis izgleda. Vsi podatki v danem stolpcu morajo biti istega tipa.

Tabela lahko vsebuje primarni ključ (angl. *primary key*). Ključ je stolpec ali niz stolpcev, katerih vrednosti unikatno identificirajo vsako vrstico. Stolpci drugih tabel so lahko tuji ključi (angl. *foreign keys*), katerih vrednosti morajo biti enake vrednostim primarnega ključa prve tabele. DB2 lahko avtomatsko uveljavi integriteto odnosov med tujim in primarnim ključem z varovanjem proti vnosom, posodabljanjem ali brisanjem, ki kršijo to integriteto.

V NLB se na osrednjem IBM računalniku uporablja en sistem za upravljanje podatkovnih baz in sicer DB2. Le-ta temelji na razvoju formalne teorije relacijskega podatkovnega modela.

V DB2 se tako za upravljanje kot za opis podatkov in nadzor SUPB uporablja en sam jezik - SQL (angl. *Structured Query Language*). Ta jezik je namenjen dostopu do podatkov v vseh SAA okoljih. SQL je že postal standard pri jezikih za opis in ravnanje s podatki, ki ga v različnih oblikah uporablja večina SUPB [6].

8.1. Varnost v DB2 bazi

Vprašanje varnosti baze je rešeno na več načinov. V DB2 ima vsak uporabnik v Sistemu za upravljanje podatkovnih baz več identifikacijskih oznak:

- primarna identifikacijska oznaka (angl. *primary ID*) služi predvsem za dostop do SUPB samega. Na podlagi te oznake so uporabniki razdeljeni v več kategorij,
- dodatne identifikacijske oznake (angl. *secondary ID*) pa služijo predvsem za dodeljevanje dodatnih pravic uporabe sistema,
- obstaja še posebna identifikacijska oznaka (SQL – ID) za uporabo jezika SQL.

Ob implementaciji SUPB-a ima dostop do vseh objektov sistema le varnostni vzdrževalec podatkovnih baz (angl. *security administrator*), ki z ukazi dodeljuje in odvzema pravico do kreiranja in uporabe celotne baze podatkov, posamezne tabele, pogleda, aplikacije, sistemskega kataloga in tudi do posameznih ukazov.

Standardni jezik za delo z relacijskimi bazami podatkov je SQL.

Z njim lahko:

- definiramo in spreminjamo strukturo tabel,
- brišemo tabele,
- vnašamo podatke,
- brišemo vrstice tabel,
- spreminjamo podatke,
- poizvedujemo.

8.2. DCLGEN

Dclgen je tabelna shema, ki jo uporabljajo cobolski programi, ki delujejo z podatkovno bazo. Se uporablja kadar programiramo s kurzorjem . Je nekakšna vmesna vez med DB2-jem in COBOL-om. V Dclgen-u je najprej naveden opis polj v DB2 strukturi, kasneje pa v enakem vrstnem redu v cobolski strukturi. Dclgen lahko vključimo v naš program, kar nam skrajša čas, ki bi ga potrebovali za ročni vnos.

```

*****
* DCLGEN TABLE (DBU01.T52ZC_ZAPIS) *
* LIBRARY (DB2.SRCLIB.DSNT (T52ZCZAP) ) *
* LANGUAGE (COBOL) *
* STRUCTURE (T52ZCZAP) *
* APOST *
* DECSDELIM (NO) *
* INDVAR (YES) *
* ... IS THE DCLGEN COMMAND THAT MADE THE FOLLOWING STATEMENTS *
*****
EXEC SQL DECLARE T52ZC_ZAPIS TABLE
( DAVCNA_ST INTEGER NOT NULL,
LETO SMALLINT NOT NULL,
MESEC SMALLINT NOT NULL,
DAN SMALLINT NOT NULL,
ZAPOREDNA_STEVIKA SMALLINT NOT NULL,
ZAPIS INTEGER NOT NULL,
DAT_VALUTACIJE DATE NOT NULL,
ZNESEK DECIMAL(15, 2) NOT NULL,
FIN_INST_RK SMALLINT NOT NULL,
ORG_ENOTA_RK SMALLINT NOT NULL,
LASTNA_ST_RK INTEGER NOT NULL,
KST_RK SMALLINT NOT NULL,
REFERENCNA_ST CHAR(20),
NAMEN CHAR(35),
FIN_INST_PR SMALLINT NOT NULL,
ORG_ENOTA_PR SMALLINT NOT NULL,
LASTNA_ST_PR INTEGER NOT NULL,
KST_PR SMALLINT NOT NULL,
ORG_ENOTA SMALLINT,
LASTNA_ST DECIMAL(10, 0),
KONTO SMALLINT,
FREKVENCA INTEGER,
ST_DB_DOLZNIKA DECIMAL(10, 0),
ST_DB_UPNIKA INTEGER,
OPERATER_BLAGAJNIK CHAR(5),
ID_PAKETA_BANCS INTEGER NOT NULL,
APLIKACIJA_OUT INTEGER,
STAT_KONTROLE INTEGER NOT NULL,
TS_OBDELAVE TIMESTAMP NOT NULL,
STAT_INFORMACIJE SMALLINT NOT NULL,
TS_INFORMACIJE TIMESTAMP NOT NULL,
ID_PAKETA_POSLAN INTEGER NOT NULL,
SIFRA_VAL INTEGER NOT NULL,
SIFRA_INF INTEGER NOT NULL,
VRS_POSLA SMALLINT,
NAPAKA CHAR(1) NOT NULL,
SIFRA_NAK CHAR(3),
SIFRA_STAT INTEGER NOT NULL,
SIFRA_NAM CHAR(5),
FIN_INST SMALLINT NOT NULL,

```

```

PAKET                                INTEGER NOT NULL,
ST_VPL_DNEVNIKA                       INTEGER,
VRS_ZAPISA                             SMALLINT NOT NULL,
SIFRA_KNJ                              INTEGER NOT NULL,
VRSTA_PROCESA                          INTEGER NOT NULL,
TS_KNJIZENJA                           TIMESTAMP NOT NULL,
FIN_INST_CILJ                           SMALLINT,
ORG_ENOTA_CILJ                           SMALLINT,
KODA_TRANSAKCIJE                       CHAR(8),
TS_ZAPISA                               TIMESTAMP,
ORG_ENOTA_INI                           SMALLINT,
UPORABNIK                              CHAR(8),
TS_SPREMEMBE                            TIMESTAMP,
PAKET_VIR                              CHAR(8),
TS_VIR                                  TIMESTAMP,
REFERENCNA_ST_2                         CHAR(20),
VRS_POSLA_ORIG                          SMALLINT,
VRS_PR                                  INTEGER,
ST_DB_DOLZNIKA_CIL                     DECIMAL(10, 0) NOT NULL,
FORMAT_NALOGA                           SMALLINT NOT NULL,
VRSTA_NALOGA                            SMALLINT NOT NULL,
FIN_INST_PR_ORIG                        SMALLINT NOT NULL,
ORG_ENOTA_PR_ORIG                       SMALLINT NOT NULL,
LASTNA_ST_PR_ORIG                       INTEGER NOT NULL,
KST_PR_ORIG                             SMALLINT NOT NULL
) END-EXEC.
*****
* COBOL DECLARATION FOR TABLE DBU01.T52ZC_ZAPIS *
*****
01  T52ZCZAP.
    10  DAVCNA-ST                PIC S9(9) USAGE COMP.
    10  LETO                     PIC S9(4) USAGE COMP.
    10  MESEC                    PIC S9(4) USAGE COMP.
    10  DAN                      PIC S9(4) USAGE COMP.
    10  ZAPOREDNA-STEVIKA       PIC S9(4) USAGE COMP.
    10  ZAPIS                    PIC S9(9) USAGE COMP.
    10  DAT-VALUTACIJE         PIC X(10).
    10  ZNESEK                  PIC S9(13)V9(2) USAGE COMP-3.
    10  FIN-INST-RK             PIC S9(4) USAGE COMP.
    10  ORG-ENOTA-RK           PIC S9(4) USAGE COMP.
    10  LASTNA-ST-RK          PIC S9(9) USAGE COMP.
    10  KST-RK                 PIC S9(4) USAGE COMP.
    10  REFERENCNA-ST         PIC X(20).
    10  NAMEN                  PIC X(35).
    10  FIN-INST-PR           PIC S9(4) USAGE COMP.
    10  ORG-ENOTA-PR         PIC S9(4) USAGE COMP.
    10  LASTNA-ST-PR        PIC S9(9) USAGE COMP.
    10  KST-PR                PIC S9(4) USAGE COMP.
    10  ORG-ENOTA            PIC S9(4) USAGE COMP.
    10  LASTNA-ST           PIC S9(10)V USAGE COMP-3.
    10  KONTO                 PIC S9(4) USAGE COMP.
    10  FREKVENCA            PIC S9(9) USAGE COMP.
    10  ST-DB-DOLZNIKA       PIC S9(10)V USAGE COMP-3.
    10  ST-DB-UPNIKA         PIC S9(9) USAGE COMP.
    10  OPERATER-BLAGAJNIK   PIC X(5).
    10  ID-PAKETA-BANCS      PIC S9(9) USAGE COMP.
    10  APLIKACIJA-OUT      PIC S9(9) USAGE COMP.
    10  STAT-KONTROLE       PIC S9(9) USAGE COMP.
    10  TS-OBDELAVE         PIC X(26).
    10  STAT-INFORMACIJE    PIC S9(4) USAGE COMP.
    10  TS-INFORMACIJE     PIC X(26).

```

```

10 ID-PAKETA-POSLAN      PIC S9(9) USAGE COMP.
10 SIFRA-VAL             PIC S9(9) USAGE COMP.
10 SIFRA-INF             PIC S9(9) USAGE COMP.
10 VRS-POSLA            PIC S9(4) USAGE COMP.
10 NAPAKA                PIC X(1) .
10 SIFRA-NAK            PIC X(3) .
10 SIFRA-STAT           PIC S9(9) USAGE COMP.
10 SIFRA-NAM            PIC X(5) .
10 FIN-INST             PIC S9(4) USAGE COMP.
10 PAKET                 PIC S9(9) USAGE COMP.
10 ST-VPL-DNEVNIKA      PIC S9(9) USAGE COMP.
10 VRS-ZAPISA           PIC S9(4) USAGE COMP.
10 SIFRA-KNJ            PIC S9(9) USAGE COMP.
10 VRSTA-PROCESA        PIC S9(9) USAGE COMP.
10 TS-KNJIZENJA         PIC X(26) .
10 FIN-INST-CILJ        PIC S9(4) USAGE COMP.
10 ORG-ENOTA-CILJ       PIC S9(4) USAGE COMP.
10 KODA-TRANSAKCIJE     PIC X(8) .
10 TS-ZAPISA            PIC X(26) .
10 ORG-ENOTA-INI        PIC S9(4) USAGE COMP.
10 UPORABNIK            PIC X(8) .
10 TS-SPREMEMBE         PIC X(26) .
10 PAKET-VIR            PIC X(8) .
10 TS-VIR                PIC X(26) .
10 REFERENCNA-ST-2      PIC X(20) .
10 VRS-POSLA-ORIG       PIC S9(4) USAGE COMP.
10 VRS-PR                PIC S9(9) USAGE COMP.
10 ST-DB-DOLZNIKA-CIL  PIC S9(10)V USAGE COMP-3.
10 FORMAT-NALOGA        PIC S9(4) USAGE COMP.
10 VRSTA-NALOGA         PIC S9(4) USAGE COMP.
10 FIN-INST-PR-ORIG     PIC S9(4) USAGE COMP.
10 ORG-ENOTA-PR-ORIG    PIC S9(4) USAGE COMP.
10 LASTNA-ST-PR-ORIG    PIC S9(9) USAGE COMP.
10 KST-PR-ORIG          PIC S9(4) USAGE COMP.
*****
* INDICATOR VARIABLE STRUCTURE *
*****
01 IT52ZC-ZAPIS.
   10 INDSTRUC           PIC S9(4) USAGE COMP OCCURS 65 TIMES.
*****
* THE NUMBER OF COLUMNS DESCRIBED BY THIS DECLARATION IS 65 *
*****

```

9. COBOL DB2

COBOL, akronim za Common Business-Oriented Language. Ta jezik se uporablja v poslovne namene. COBOL, napisan z uporabo angleških stavkov, je namenjen reševanju problemov, ki ne zahtevajo mnogo matematičnih preračunavanj. Prvič je bil predstavljen leta 1959. COBOL pripada jezikom tretje generacije.

COBOL program v osnovi sestavljajo štirje deli:

IDENTIFICATION DIVISION

ENVIRONMENT DIVISION

DATA DIVISION

PROCEDURE DIVISION.

IDENTIFICATION DIVISION

»Identification division« se uporablja za poimenovanje našega programa. Če želimo lahko napišemo tudi ime avtorja programa, čas nastanka programa in še nekatere druge lastnosti našega programa. S poimenovanjem našega programa, omogočimo drugim programom, da kličejo naš program. Ime se pojavlja v naslovu vsake strani, razen prve strani, pri prevajanju programa.

IDENTIFICATION DIVISION.

Program-ID. P86B0001.

Author. F.Drnovsek.

Installation. NLB.

Date-Written. 12/09/2008.

Date-Compiled. 02/10/2008.

ENVIRONMENT DIVISION

»Environment division« se uporablja za opisovanje z vidika programa, ki temelji na računalniškem okolju (vhodni ter izhodni tok programa).

DATA DIVISION

»Data division« se uporablja za določitev podatkov uporabljenih v vhodno-izhodnih operacijah, za določitev podatkov, ki so statično dodeljeni in obstoje za čas izvajanja programa (WORKING-STORAGE SECTION), za določitev lokalno dodeljenih podatkov (LOCAL-STORAGE SECTION) ter za opis podatkov iz drugega programa (LINKAGE SECTION).

PROCEDURE DIVISION

»Procedure division« se uporablja za pisanje izvedljivih programskih stavkov, da obdelajo podatke, ki smo jih določili v ostalih oddelkih (divisions). Ta oddelek vsebuje celotno logiko našega programa [7].

Kurzor (angl. *CURSOR*)

Pri uporabi kurzorja moramo biti zelo previdni, saj moramo poleg pravil za pisanje kurzorja upoštevati tudi hitrost izvedbe le tega. Pri manjši količini transakcij se hitrost izvedbe bistveno ne pozna, ko pa imamo nekaj sto tisoč transakcij, pa je dobro napisan kurzor odločilnega pomena, saj nam kar za nekajkrat lahko prihrani čas izvajanja programa.

```
*****
* SQL CURSOR                                     *
*****
*      KURZOR BAN$NI PRENOSI                     *
*****

EXEC SQL DECLARE OBDELA CURSOR WITH HOLD FOR
SELECT FIN_INST_CILJ
FROM   T52ZC_ZAPIS
WHERE  VRS_ZAPISA = 4 AND
       SIFRA_INF = 1 AND
       DAT_VALUTACIJE = '28.10.2008' AND
       APLIKACIJA_OUT BETWEEN 1 AND 6 AND
       FIN_INST_RK = 4 AND
       ORG_ENOTA_RK = 515 AND
       LASTNA_ST_RK = 4980 AND
       KST_RK = 21
FOR UPDATE
END-EXEC.
```

9.1. Copybook

Copybook je del programske kode, ki je shranjen izven programa in se kopira v program oziroma v več različnih programov (ali na različna mesta v enem programu). Najpogosteje se uporablja za definiranje fizičnega razporejanja podatkov (lahko kot vhodna ali izhodna datoteka) [8].

Glavni razlogi za uporabo copybook-ov so:

- zagotovi, da vsakdo uporablja enako različico razporejanja podatkov,
- omogoča lažje navzkrižno sklicevanje, kjer so sestavni deli uporabljeni v sistemu,
- omogoča lažjo spremembo programov (popravi se samo en copybook),
- proumerju skrajša čas, saj mu ni potrebno te strukture ugnezditi v program.

Copybook vključimo v naš program z ukazom COPY in imenom našega copybook-a.

Primer copybook-a za DB:

```
*****
* C86NPI04 - COPYBOOK                             *
* =====*
* DATE | VER. | COMMENT                             *
* -----+-----+-----*
* 080514 | 1.0 | INITIAL VERSION                     *
* Vsebina: Zapis 04 - direktna obremenitev         *
*****
```

13 (DB)-C86NPI04.
 15 (DB)-VZ PIC XX.
 88 (DB)-DIREKTNA-BREMENITEV VALUE '04'.
 15 (DB)-ID-ZAPISA.
 17 (DB)-DAV-ST-INIC PIC 9(8).
 17 (DB)-DATUM PIC 9(6).
 17 (DB)-SS PIC 99.
 88 (DB)-SS-OK VALUE 04.
 17 (DB)-ZAP-ST-PAKET PIC 99.
 17 (DB)-ZAP-ST-ZAPIS PIC 9(6).
 15 (DB)-OZN-KNJ PIC 9.
 88 (DB)-KNJIZENJE VALUE 0.
 15 (DB)-DATUM-VALUTE PIC 9(8).
 15 (DB)-DATUM-VALUTE-R REDEFINES (DB)-DATUM-VALUTE.
 17 (DB)-DAT-VAL-LLLL PIC 9(4).
 17 (DB)-DAT-VAL-MM PIC 99.
 17 (DB)-DAT-VAL-DD PIC 99.
 15 (DB)-ZNESEK PIC 9(13)V99.
 15 (DB)-VALUTA PIC 999.
 15 (DB)-RACUN-DOBRO.
 17 (DB)-RACUN-DOBRO-TRR.
 19 (DB)-RAC-DO-BANKA PIC 99.
 19 (DB)-RAC-DO-PE PIC 999.
 19 (DB)-RAC-DO-PART PIC 9(8).
 19 (DB)-RAC-DO-KST PIC 99.
 17 (DB)-FILLER PIC XXX.
 15 (DB)-REFER PIC X(20).
 15 (DB)-NAMEN PIC X(35).
 15 (DB)-SIF-NAMEN PIC 999.
 15 (DB)-RACUN-BREME-TRR.
 17 (DB)-RAC-BR-TRR-BANKA PIC 99.
 17 (DB)-RAC-BR-TRR-PE PIC 999.
 17 (DB)-RAC-BR-TRR-PART PIC 9(8).
 17 (DB)-RAC-BR-TRR-KST PIC 99.
 15 (DB)-RACUN-BR.
 17 (DB)-RAC-BR-PE PIC 999.
 17 (DB)-RAC-BR-APL PIC 99.
 17 (DB)-RAC-BR-PART PIC 9(8).
 17 (DB)-RAC-BR-KST PIC 99.
 15 (DB)-SIF-PREJ-IZD PIC X(5).
 15 (DB)-VRSTA-INF PIC 99.
 15 (DB)-STATUS PIC 99.
 15 (DB)-SIFRA-KONTA PIC 999.
 15 (DB)-FREKVENCA PIC 99.
 15 (DB)-ZAP-ST-DB-UPNIKA PIC 9(5).
 15 (DB)-ZAP-ST-DB-DOLZNIKA PIC 9(10).
 15 (DB)-OP-ID PIC X(5).

* LENGTH = 193 BYTES *

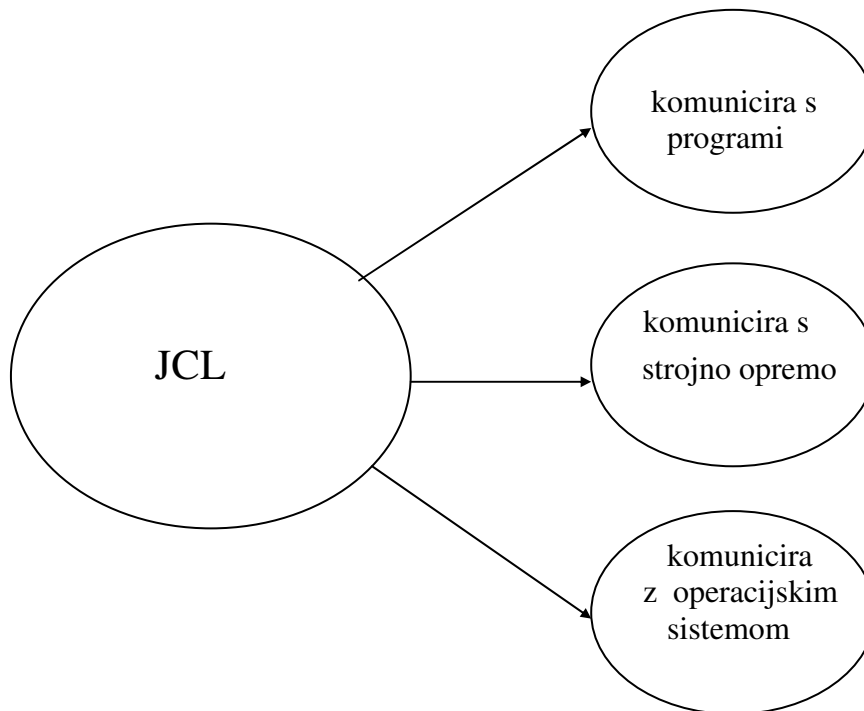
10. JCL (Job Control Language)

Razvil ga je IBM leta 1965. JCL se ne uporablja za pisanje programov. Je skriptni jezik, ki se uporablja na IBM-ovih osrednjih operacijskih sistemih. Pouči sistem kako izvajati »batch« ukaze ali zagnati podsisteme. Sam »Job« je sestavljen iz enega ali več korakov, vsak ta korak pa poganja določen program. V vsakem koraku podamo ime programa, kateremu določimo vhod in izhod programa, dolžino datotek in še mnogo drugih parametrov.

Primer JOB-a:

```
//J57PDHH1 JOB ,MSGLEVEL=(1,1),MSGCLASS=M,REGION=33M,USER=SSCHD
//JOBLIB DD DSN=PRD1.LINKLIB,DISP=SHR
/*-----
/* LO~EVANJE PODATKOV IZ SKUPNEGA VHODA
/*-----
//JOBIME EXEC PGM=P90KOM,
/*===== *
// PARM='*** ZBIRNI CENTER - SKUPNI VHOD ***'
/*===== *
/*-----
/*ZC001BSV - LO~EVANJE SKUPNEGA VHODA NA NOV, STAR FORMAT TER OSTALO
/*-----
//ZC001BSV EXEC PGM=P57BERSV
//PGMINP1 DD DSN=P93.SK.VHOD.K.DISK(+0),DISP=OLD
// DD DSN=P93.SK.VHOD.NDM(+0),DISP=OLD
//PGMOUT1 DD DSN=P57.SK.SKVHOD.LOCEN.NOV,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(140,5),RLSE),
// DCB=(RECFM=FB,LRECL=255)
//PGMOUT2 DD DUMMY
//PGMOUT3 DD DSN=P57.SK.SKVHOD.LOCEN.OSTALO,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(200,10),RLSE),
// DCB=(RECFM=FB,LRECL=255)
//SYSOUT DD SYSOUT=*
/*-----
/* VARNOSTNA KOPIJA
/*-----
//ZCCOPY1 EXEC PGM=P99COPY
//PGMINP1 DD DSN=P57.SK.SKVHOD.LOCEN.NOV,DISP=OLD
//PGMOUT1 DD DSN=P57.SK.SKVHOD.LOCEN.NOV.KOPIJA(+1),
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(100,30),RLSE),
// DCB=(RECFM=FB,LRECL=255,BLKSIZE=27795)
//SYSOUT DD SYSOUT=*
/*-----
/*ZCEOF01 - EOF -BRISANJE VHODNIH FILE-OV NLB
/*-----
//ZCEOF01 EXEC PGM=P99EOF
//PGMOUT1 DD DSN=P93.SK.VHOD.K.DISK(+1),
// DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(1,1),RLSE),
// DCB=(SYS1.MODEL,RECFM=FB,BLKSIZE=27795,LRECL=255)
//SYSOUT DD SYSOUT=*
/*-----
/*ZCEOF02 - EOF -BRISANJE VHODNIH FILE-OV NLB
/*-----
//ZCEOF02 EXEC PGM=P99EOF
```

```
//PGMOUT1 DD DSN=P93.SK.VHOD.NDM(+1),  
//  DISP=(NEW,CATLG,DELETE),  
//  SPACE=(TRK,(1,1),RLSE),  
//  DCB=(SYS1.MODEL,RECFM=FB,BLKSIZE=27795,LRECL=255)  
//SYSOUT DD SYSOUT=*  
/*-----  
/*- ----->KONEC OBDELAVE<----- -*  
/*-----  
//JOBKONEC EXEC PGM=P90KONEC,RD=R
```



Slika 18: Glavne lastnosti JCL -a

11. SKLEP

Sodobno spreminjajoče poslovno okolje od bank zahteva prožnost v poslovanju in zagotavljanje čim boljših storitev. Le tako lahko banka posluje uspešno in dosega zastavljene cilje.

Na prvi pogled bi človek rekel, da temu ni tako, saj vemo, da je večina programov v aplikaciji ZC NLB napisana v programskem jeziku COBOL, ki pa je bil predstavljen že daljnega leta 1959 in pripada 3. generaciji jezikov. A vendar je COBOL jezik, ki se še vedno razvija in je zelo uporaben, posebno z uporabo DB2-ja, kar se izkaže pri obdelavah masovnih količin podatkov. Za hiter odzivni čas obdelav v ZC NLB, pa potrebno zagotoviti kakovostno infrastrukturo.

Z vstopom v Evropsko unijo in prevzemom nove valute, je k nam prišla tudi SEPA struktura, ki bo postopoma nadomestila NPI strukturo. SEPA je oznaka enotnega območja plačil v evrih, s katerim se bo poenostavilo poslovanje tako podjetij kot fizičnih oseb. SEPA struktura je zapisana v XML obliki. S prihodom modernejših jezikov, je tudi NLB primorana k uvedbi modernejših orodji. Z orodjem RAD, smo se temu že zelo približali, saj lahko s tem orodjem razvijamo aplikacije v različnih programskih jezikih kot so COBOL, Java, XML, C, C++. Razvijamo zapletene spletne aplikacije s pomočjo WAS-a (Web Service), delamo poizvedbe in popravke v bazi.

Trenutno se del aplikacije ZC NLB nadgrajuje z aplikacijo ADONIS, ki bo odpravila del napak, ki so bile opažene pri ZC NLB, predvsem pa bo nova aplikacija bolj prijazna do uporabnika in učinkovitejša. Za to gredo velike zasluge novim programskim orodjem, boljši infrastrukturi ter zgodovina, od katere smo se naučili nekaj napak, ki jih ne bomo ponovili.

12. SEZNAM SLIK

Slika 1. Podatki kot osrednji vir v podjetju	3
Slika 2. Struktura upravljalnega centra za informacijsko tehnologijo.....	5
Slika 3. Struktura sektorja za razvoj informacijskega sistema banke	6
Slika 4: Načini izmenjave podatkov	8
Slika 5 : Prikaz glavnih gradnikov v ZC NLB.....	16
Slika 6: Primer pravilno izdelane datoteke DB	17
Slika 7: Prikaz poslovnega procesa.....	18
Slika 8: Osnovni prikaz CICS-a.....	26
Slika 9: CICS – več uporabnikov	27
Slika 10 : Prikazuje trenutno stanje CICS programov.....	28
Slika11: Prikazuje CISICS DB2 program v RAD-u	30
Slika12: Prikazuje delovanje MQ -ja	32
Slika 13: MQ na z/OS	33
Slika 14: MQ klici.....	34
Slika 15: Tri nivojska zgradba baze podatkov	38
Slika 18: Glavne lastnosti JCL -a.....	47

13. SEZNAM TABEL

Tabela 1: Struktura vodilnega zapisa	11
Tabela 2: Struktura zbirnega zapisa	12
Tabela 3: Struktura osnovnega zapisa za DB	14

14. LITERATURA IN VIRI

- [1] Bankart, *Navodila in standardi za izmenjavo podatkov prek Zbirnega centra*, Dostopno na: http://www.bankart.si/si/ponudba/zbirni_center
- [2] Chisholm W.: *CSS Techniques for Web Content Accesibility Guidelines 1.0*. <http://www.w3.org/TR/WCAG10-CSS-TECHS/>, 20.10.2002b.
- [3] IBM – *CICS Application Programming Guide 3*. Zvezek: 1. Izdaja, April 2007, str. 3 – 37.
- [4] N. Williams, *Connecting WebSphere To CICS With CICS Transaction Gateway*, Montpellier, Francija, september 2002, str. 4 – 9.
- [5] IBM, *A Technical Introduction To WebSphere MQ (Course Code MQ01)*, 1998, 2005, str. 8 – 30.
- [6] IBM, *DB2 For z/OS And WebSphere: The Perfect Couple*, januar 2005, pogl. 1.4.
- [7] IBM, *WebSphere Studio COBOL for Windows*, marc 2004, str. 5 -11.
- [8] (2008) Wikipedia.org, *Copybook*, dostopno na: <http://en.wikipedia.org/wiki/Copybook>

15. Izjava o avtorstvu

Izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja doc. dr. Marka Bajca. Za izkazano pomoč sodelavcev sem se zahvalil uvodoma.