

PROGRAMOVÁNÍ SYSTÉMŮ DOTAZ - ODPOVĚĎ

1. Úvod

Rozvoj technického a programového zabezpečení počítačů postavil "člověka - uživatele výpočetní techniky" do průsečíku nových problémů. Možnost přímé spolupráce s počítačem /terminály/, dálkové zpracování dat naplňují sen o práci v reálném čase. Přitom se setkáváme s různými režimy zpracování úloh, u kterých je základním kritériem úroveň komunikace člověk-počítač.

Z tohoto funkčního hlediska považují za nejobecnější rovinu interaktivní systémy, kde jde o velmi široké rozmezí vzájemného ovlivňování na úrovni člověk-počítač /dispečerské systémy, dálkový vstup úloh, terminálové zpracování s různými časovými nároky, navrhování obvodů, dialogové systémy aj./.

Dialogové systémy jsou determinovány formalizovaným stykem člověka s počítačem a to na úrovni komunikačních prostředků, které jsou okamžitě transponovatelné do strojového jazyka /konverzační programování, výpočty na minipočítačích, práce v režimu dotaz-odpověď aj./.

Práce v režimu dotaz - odpověď představuje speciální kategorii, ve které operátor zpracovává data podle připravených programů. Dotazy i odpovědi jsou formalizovány předem s ohledem na okamžité zpracování informací vesměs pro potřeby vnějšího reálného procesu. Proto je zde daleko užší vazba mezi projektováním a programováním dialogu.

2. Dialog člověk - počítač

Navzdory propracovaným metodikám projektování velkých systémů, systémů dálkového zpracování dat a jiným, vychází ze čtyř základních etap, jejichž charakteristiky musí být determinovány na začátku projektu :

- /1/ vymezení účelu dialogu,
- /2/ kriteria doby odezvy,
- /3/ návrh struktury dialogu,
- /4/ programování dialogu.

S těmito etapami se prolínají čtyři funkce všech systémů pracujících v režimu dotaz - odpověď :

- /1/ vytvoření datové základny,
- /2/ dotaz na datovou základnu,
- /3/ aktualizace záznamu,
- /4/ aktualizace datové základny.

Klasik dialogových systémů James Martin uvádí v monografii

/1/ celkem 12 různých typů dialogu :

1. Konverzační programování - použití programovacích jazyků.
2. Dialog pomocí přirozeného jazyka.
3. Použití podmnožiny přirozeného jazyka.
4. Použití mnemonických výrazů.
5. Použití speciálních operátorů analogických s operátory programovými.
6. Dialog vyvolaný počítačem.
7. Vyplňování generovaných formulářů.
8. Hybridní dialog.
9. Odpovědi pomocí speciálních tlačítek.
10. Dialog neškoleného operátora.
11. Odpovědi z předem připravené množiny odpovědí.
12. Dialog pomocí obrezů.

Najdříve se budeme zabývat extrémny.

Ad 1/ Konverzační programování tvoří velkou samostatnou oblast s vlastními pravidly a požadavkem jisté matematické erudice a schopnosti abstrahovat. Již v úvodu jsem vymežil sféru tohoto příspěvku /dialogy pracující s daty/, do které konverzační programování nespadá.

Ad 12/ Zaměřujeme se pouze na textové dialogy jako na nejrozšířenější prezentaci dotazů a odpovědí v ASŘ.

U ostatních typů dialogů se domnívám, že J.Martin směřuje několik hledisek, které je nezbytné - právě v projektech režimů dotaz-odpověď - rozlišovat. Jsou to :

- /a/ použitý jazyk pro dialog operátor - počítač,
- /b/ řídicí logický směr dialogu /dialog vedený operátorem nebo počítačem/,
- /c/ úroveň vyškolení operátora.

Podle těchto hledisek navrhuje projektant dotazy a odpovědi a vybírá adekvátní programovací nástroje.

- Ad a/ Škála "jazyků" :
- programovací jazyk,
 - přirozený jazyk,
 - podskupina přirozeného jazyka,
 - mnemonické výrazy,
 - speciální klíče na terminálech.

Ad b/ Dialog vedený operátorem :

- aktivní spolupráce /aktualizace záznamu nebo datové základny/,
- dotazy operátora.

Dialog vedený počítačem :

- výběr variant /"menu", jídelníček/; operátor se nemusí učit mnemoniku, jen vybírá větvení postupu a doplňuje data,
- vyplňování formulářů : obdoba předcházejícího případu, ale formulář je pevně stanoven a proto stačí do počítače vnášet pouze vstupní data,
- odpovědi pomocí speciálních kláves, které korespondují s funkcemi, o kterých operátor rozhodl, že se má realizovat /případ dispečerského pultu nebo odpovědi neškoleného operátora/.

Ad c/ Úroveň vyškolení operátora :

- operátor-dispečer : je plně obeznámen s řídicím systémem reálného procesu, ve kterém je terminálová síť nasazena. Je vyškolen v obsluze terminálu, ovládá komunikační prostředky /jazyk,

memoniku/ a je samostatným aktivním činitelem v procesu řízení. Svými znalostmi a zkušenostmi završuje automatizovaný systém řízení a zajišťuje jeho operativnost,

- vyškolený operátor : ovládá komunikační prostředky i techniku interakce pomocí terminálu, ale operuje s daty, které převzal /nebo dodává/ ostatním lidem spojeným s reálným procesem,

- . občasný operátor,
- . stálý operátor,

- neškolený operátor : má bezprostřední styk s reálným procesem, ale informace předává v jednoduché formě nebo používá jako mezičlánek vyškoleného operátora /"trojstranný dialog"/.

Je zřejmé, že v této stupnici existují další mezivrstvy /kritériem zde může být schopnost použití více jazyků, individuální rychlost odpovědi, stupeň řízení atp./.

3. Návrh formátu a textů na terminálu

Vlastní komunikace člověka a počítačem se realizuje prostřednictvím terminálu. V režimech dotaz-odpověď nejčastěji používáme displeje s klávesnicí nebo dálnopisné terminály. Důležitou etapou projektu /bezprostředně před programováním/ je navrhování formátu a textů zpráv pro optimální interakci člověk - terminál. U každého kritéria budeme rozlišovat mezi možnostmi pro dálnopis a pro displej.

/a/ Volba jazyka

Nejčastěji se používá podmožina přirozeného jazyka nebo mnemonické kódy. Mnemonika je vhodná u jednocelových systémů, kde s terminálem přijdou do styku vesměs operátoři seznámení s reálným procesem. Mnemonika je také vhodnější pro vstupní informace : zkracuje dobu vstupu informace, pro počítač je dlouhá textová informace redundantní, rychlý zápis vstupní informace zvyšuje jistotu a sebedůvěru operátora. Důležitou rolí zde hraje také proces kódování v ušlání operátora.

Pro výstup jsou vhodnější delší textové řetězce. Dekódování v mysli průměrného operátora trvá déle než kódování, delší text nepůsobí depresivně na rozdíl od neznámé nebo málo používané zkratky z vymezené podмноžiny mnemonik. Navíc se v delším čase výstupu překryje doba pro zpracování a zpětný přenos.

/b/ Kapacita informace dostupné na terminálu je u displeje omezená informačním rozsahem obrazovky, u dálnopisu prakticky nekonečná, ale pro interakci v reálném čase omezená několika málo řádky tisku.

/c/ S tím souvisí označení zpracovávaného místa a indikace zda terminál čeká na údaje operátora nebo na příjem výsledků.

Označení místa : u dálnopisu se obvykle vytiskne ? a systém čeká na zprávu operátora, momentální místo zpracování je určeno polohou tiskového koše, u displeje se používá zvláštní značka /běžec, kurzor/, podtržení místa, blikající nebo zvýrazněný poslední znak, resp. spolupráce světelného pera.

/d/ Označení informací vysílaných počítačem a informací vkládaných operátorem je důležité při delších sekvencích dialogu nebo při jeho rekonstrukci.

U dálnopisu se používá dvoubarevná tiskařská páska.

U displeje se využívá ustálených míst pro rozvržení legendy a vstupního textu na obrazovce. Vychází se z běžné praxe formulářů, kde vlevo máme "rubriky" a vpravo vyplňujeme text. Zde je nebezpečí při nenasplnění nebo překročení kapacity, která v programu je vymezena a která nemusí být známa operátorovi.

Některé projekty řeší tuto otázku méně vhodně vydáváním manuálů a tabulek. Navíc se objevují v jedné zprávě texty s fakultativní shora omezenou délkou - např. adresa, a současně řetězce s implicitní délkou, která pro smysluplnost musí být vyčerpána - například PSČ.

Vhodnější je ponechat shora omezenou délku položek /volná místa doplní systém/ a při nedodržení pravidel okamžitě

hlásit operátorovi chybu, bez které dialog nemůže pokračovat.

/e/ Rozvedením předcházejícího bodu je oddělení dynamických a statických informací na terminálu.

Statické informace /přehledové, doplňkové, společné, statistické - např. datum běžného dne, průběžný stav ve skladu, počet operací z daného terminálu atp./ se na obrazovku umísťují zpravidla do pravého horního rohu.

Dynamické informace /vstupní i výstupní/ se řadí z levého horního rohu a /pokud to formáty dovolují/ postupují jako "pyramida" k základně obrazovky. Odpovídá to i odvíjení dialogu od obecných ke speciálním informacím. Dalším důvodem pro obsazení dolní strany obrazovky dynamickými informacemi je korespondence s funkčními klíči přímo pod obrazovkou a tak nenásilně přechází dialog na větvení.

U dálnopisného terminálu je rozdělení obtížnější a statické informace se objevují pouze v záhlaví.

/f/ Identifikace operátora je nezbytná v terminálových sítích s přístupem různých operátorů /peněžnictví, rezervace a jiné/. V praxi se kód operátora objevuje a ověřuje pouze na začátku a na konci dialogu a pod tímto kódem se provádějí všechny následné operace z daného terminálu.

/g/ Signalizace chyb je důležitá pro rychlý průběh dialogu. Některé chyby jsou indikovány systémem a signalizovány pomocí výrazných indikátorů na panelech terminálů.

Chyby zjištěné v průběhu dialogu je nejvhodnější signalizovat jednoznačným textem ve spodní části obrazovky.

Z uvedeného rozboru je zřejmé, že návrh informací pro zpracování v režimu dotaz - odpověď je velmi úzce spjat s programováním. Zejména když si uvědomíme, že do hry vstupuje různorodý lidský faktor.

4. Příklad programování dotazů a odpovědí v COBOLU

Ukázka programování vybrané části dialogu je rozpracováním projektu rezervace místenek pro železnice. Rozebíraný příklad je součástí operace "Vytvoření jízdního řádu" /viz "Vytvo-

hlásit operátorovi chybu, bez které dialog nemůže pokračovat.

/e/ Rozvedením předcházejícího bodu je oddělení dynamických a statických informací na terminálu.

Statické informace /přehledové, doplňkové, společné, statistické - např. datum běžného dne, průběžný stav ve skladu, počet operací z daného terminálu atp./ se na obrazovku umísťují zpravidla do pravého horního rohu.

Dynamické informace /vstupní i výstupní/ se řadí z levého horního rohu a /pokud to formáty dovolují/ postupují jako "pyramída" k základně obrazovky. Odpovídá to i odvíjení dialogu od obecných ke speciálním informacím. Dalším důvodem pro obsazení dolní strany obrazovky dynamickými informacemi je korespondence s funkčními klíči přímo pod obrazovkou a tak nenásilně přechází dialog na větvení.

U dálnopisného terminálu je rozdělení obtížnější a statické informace se objevují pouze v záhlaví.

/f/ Identifikace operátora je nezbytná v terminálových sítích s přístupem různých operátorů /peněžnictví, rezervace a jiné/. V praxi se kód operátora objevuje a ověřuje pouze na začátku a na konci dialogu a pod tímto kódem se provedějí všechny následné operace z daného terminálu.

/g/ Signalizace chyb je důležitá pro rychlý průběh dialogu. Některé chyby jsou indikovány systémem a signalizovány pomocí výrazných indikátorů na panelech terminálů.

Chyby zjištěné v průběhu dialogu je nejvhodnější signalizovat jednoznačným textem ve spodní části obrazovky.

Z uvedeného rozboru je zřejmé, že návrh informací pro zpracování v režimu dotaz - odpověď je velmi úzce spjat s programováním. Zejména když si uvědomíme, že do hry vstupuje různorodý lidský faktor.

4. Příklad programování dotazů a odpovědí v COSOLU

Ukázka programování vybrané části dialogu je rozpracováním projektu rezervace místenek pro železnice. Rozebíraný příklad je součástí operace "Vytvoření jízdního řádu" /viz "Vytvo-

fení datové základny" v kap. 2./ . Programový modul M018 zajišťuje opětne vyvolání vstupních dat při generování věty kmenového souboru /JÍZDNÍ ŘÁD/ pro kontrolu případné chyby vstupních dat. /obr. 1./

Program byl napsán v COBOLU pro počítač IRIS-80 s operačním systémem SIRIS na úrovni OS/MPT a s monitorem dálkového zpracování dat STRATEGE./2/ Protože se nám jedná především o způsob programování a nikoliv manuální návod, ponechávám původní text.

COBOL ve verzi pro SIRIS obsahuje několik modifikací :

- a/ Vyhrazení WORKING-STORAGE SECTION místní zóně a COMMON-STORAGE SECTION výměnné zóně.
- b/ Vyloučení použití FILE SECTION a INPUT-OUTPUT SECTION.
- c/ Speciální makroinstrukce pro dialogový režim, které se vyvolávají příkazem ENTER /viz dále/.
- d/ Deklarování aktivních míst na displeji pomocí množiny speciálních znaků a hodnot souřadnic.

Na obr. 2. je návrh mřížky s textem i místy vyhrazenými pro příjem dat.

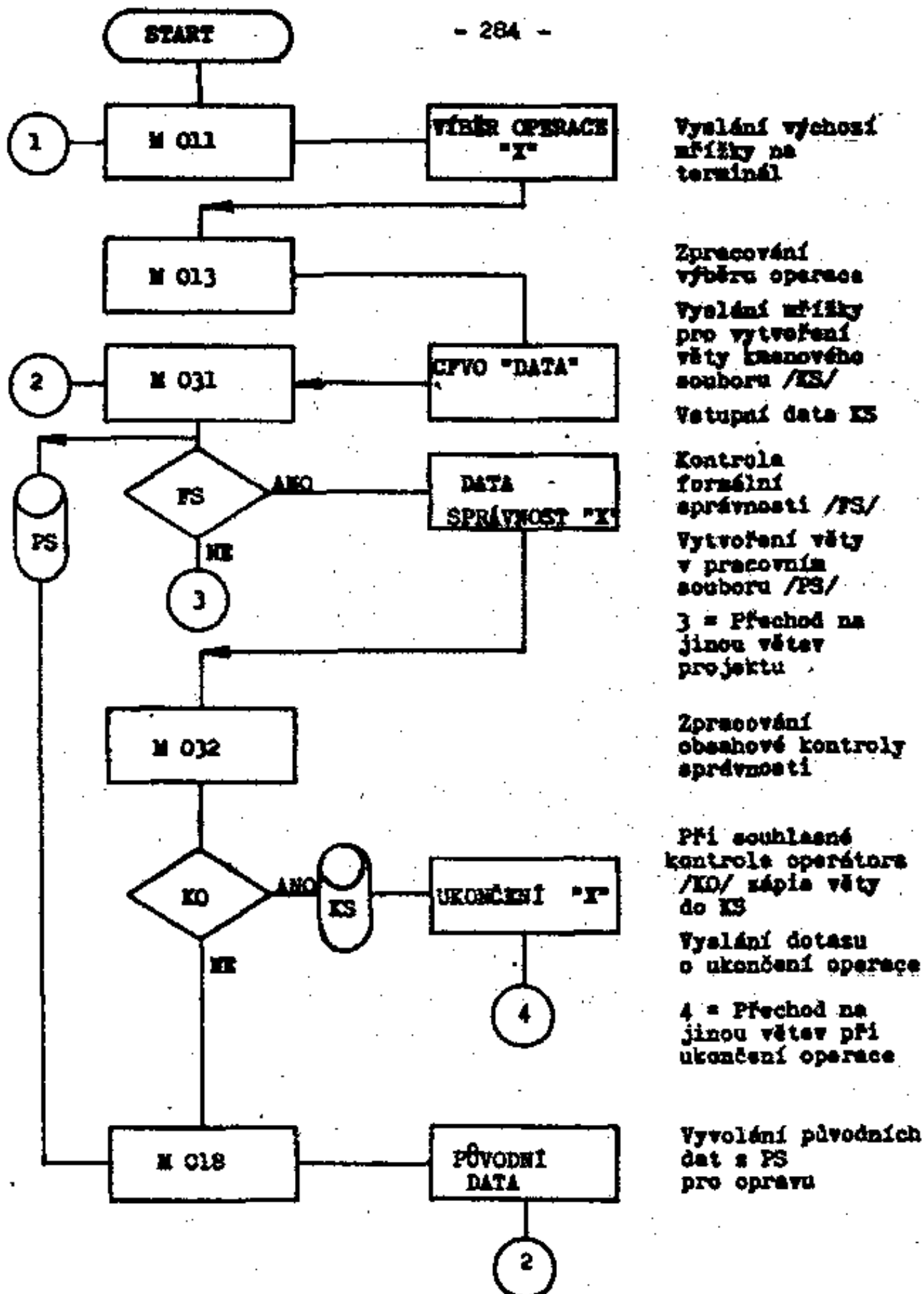
Na obr. 3. je program modulu označeného jako M018.

Vstupem programu je pracovní soubor. Funkce modulu M018 spočívá v přenesení mřížky s daty pro vytvoření nového záznamu na displej po neúspěšném pokusu o vytvoření věty v kmenovém souboru / v případě, že nenastal návrat na úplný začátek dialogu/. Výstupem je vyslání mřížky s původními daty pro opravu operátorem.

Rozmístění textu na mřížce obrazovky /viz obr. 2./ odpovídá naprogramování mřížky /GRILLE/ ve WORKING-STORAGE SECTION na řádcích 2010 až 4030.

Pro to, abychom nemuseli vypisovat hluchá místa na displeji do DATA DIVISION, používá tato verze COBOLU mj. dva servisní znaky :

- X - odpovídá ESCAPE a předchází před všemi funkčními kódy,
- K - odpovídá DEVICE CONTROL 2 a používá se pro adresování souřadnic x a y pracovního místa na displeji.



Obr. 1. Vývojový diagram části dialogu

DATA	Program: _____
	Rechercher: _____
CFVO	<p>1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0</p> <p>CREATION D UN VOYAGE</p> <p>NUMERO DE DEPART : <</p> <p>JOUR DU DEPART : <</p> <p>HEURE DU DEPART : < HEURE D ARRIVEE : ></p> <p>GARE DE DEPART : < EVENTUELLEMENT CLE D ACCES : ></p> <p>GARE D ARRIVEE : < EVENTUELLEMENT CLE D ACCES : ></p> <p>PLACES COTE FENETRE : > PLACES COTE COULOIR : <</p>

Obr. 2. Návrh mřížky pro displej

```

001010 IDENTIFICATION DIVISION.
001020 PROGRAM-ID. AQ18.
001030 AUTHOR. TOMAN.
001040* MODULE DE RETOUR D ECRAN EN CAS DE CREATION INVALIDE
001041* TROISIEME PHASE DE LA TRANSACTION C F V O
001042* ENTREE : LE FICHIER TRAVAIL ( LA GRILLE RENSEIGNEE )
001043* EFREY : SI VALIDATION NON 'X' SUR L ECRAN MODULE M 018
001044* RENVOIE LA GRILLE CREATION D UN VOYAGE RENSEIGNEE
001045* SORTIE : EMISSION DE LA GRILLE CREATION D UN VOYAGE
001046* FICHIERS : EN LECTURE FICHIER TRAVAIL
001060 ENVIRONNMENT DIVISION.
001070 CONFIGURATION SECTION.
001080 SOURCE-COMPUTER, IRIS-80.
001090 OBJECT-COMPUTER, IRIS-80.
001100 DATA DIVISION.
001130 WORKING-STORAGE SECTION.
001150 77 MODUL COMP VALUE 18.
001160 77 LGM COMP VALUE 340.
001170 77 RTFM PIC X(80).
001180 77 MOINSUN COMP VALUE -1.
002000* 01 GRILLE CREATION D UN VOYAGE
002010 01 GRILLE
002020 02 FILLER PIC X(6) VALUE 'XNXXXQ'.
002030 02 FILLER PIC XXXX VALUE 'CFVO'.
002040 02 FILLER PIC XX VALUE 'XN'.
002050 02 FILLER PIC XXX VALUE 'K3'.
002060 02 FILLER PIC X(20) VALUE
002070 'CREATION D UN VOYAGE'.
002080 02 FILLER PIC XXX VALUE 'K3'.
002090 02 FILLER PIC X(18) VALUE 'NUMERO DE DEPART 1'.
002100 02 FILLER PIC XX VALUE 'X0'.
002110 02 NO-DEPTX PIC X(5).
002120 02 FILLER PIC XX VALUE 'XN'.
002130 02 FILLER PIC XXX VALUE 'K3'.
002140 02 FILLER PIC X(16) VALUE 'JOUR DU DEPART 1'.
002150 02 FILLER PIC XX VALUE 'X0'.
002160 02 JU-DEPTX PIC X(6).
002170 02 FILLER PIC XX VALUE 'XN'.
002180 02 FILLER PIC XXX VALUE 'K3'.
002190 02 FILLER PIC X(18) VALUE 'HEURE DU DEPART 1'.
002200 02 FILLER PIC XX VALUE 'X0'.
002210 02 HE-DEPTX PIC XXXX.
002220 02 FILLER PIC XX VALUE 'XN'.
002230 02 FILLER PIC X(17) VALUE 'HEURE D ARRIVEE 1'.
002240 02 FILLER PIC XX VALUE 'X0'.
002250 02 HE-ARRX PIC XXXX.
002260 02 FILLER PIC XX VALUE 'XN'.
003010 02 FILLER PIC XXX VALUE 'K3'.
003020 02 FILLER PIC X(16) VALUE 'SARE DE DEPART 1'.
003030 02 FILLER PIC XX VALUE 'X0'.
003040 02 COD-POSTX PIC X(5).
003050 02 FILLER PIC XX VALUE 'XN'.
003060 02 FILLER PIC X(28) VALUE

```

```

003070 *EVENUELLEMENT CLE D ACCES 1 VALUE 'XO'.
003080 FILLER PIC XX VALUE 'XO'.
003090 CLE-DEPX PIC XXXX. VALUE 'K18'.
003100 FILLER PIC XXX VALUE 'GARE'.
003110 FILLER PIC X(16) VALUE 'D ARRIVEE :'.
003120 FILLER PIC XX VALUE 'XO'.
003130 COD-POSTAX PIC X(5). VALUE 'XO'.
003140 FILLER PIC XX VALUE 'XN'.
003150 FILLER PIC X(28) VALUE 'XN'.
003160 *EVENUELLEMENT CLE D ACCES 1 VALUE 'XO'.
003170 FILLER PIC XX VALUE 'XO'.
003180 CLE-ARRX PIC XXXX. VALUE 'XN'.
003190 FILLER PIC XX VALUE 'XN'.
003200 FILLER PIC XX VALUE 'K3'.
003210 FILLER PIC X(21) VALUE 'QUOTE'.
003220 FILLER PIC XX VALUE 'PLACES COTE FENETRE 1'.
003230 NO-FFNX PIC XX VALUE 'XO'.
003240 FILLER PIC XX VALUE 'XN'.
003250 FILLER PIC X(21) VALUE 'PLACES COTE COULOIR 1'.
003260 FILLER PIC XX VALUE 'XO'.
003270 NO-COULX PIC XX. VALUE 'XN'.
004010 FILLER PIC XXX VALUE 'K'.
004020 FILLER PIC XX VALUE 'XO'.
004030 FILLER PIC XXXX VALUE 'XIXX'.
004150 COMMON-STORAGE SECTION.
004160 MESSAGE EN ENTREE COMP.
004170 77 POSTE COMP.
004180 77 LONG COMP.
004200 04 HEURE.
004210 FILLER PIC XXX.
004220 MIN PIC 99.
004230 FILLER PIC XXX.
005010 01 TR.
005030 COD-TRANS PIC XXXX.
005040 NO-DEPT PIC 9(5).
005050 JU-DEPT.
005060 JJ PIC 99.
005070 MM PIC 99.
005080 AA PIC 99.
005100 02 HEURE-DEPT.
005110 NO DEPT. PIC 99.
005120 NO DEPT. PIC 99.
005140 02 HEURE-ARR.
005150 NA PIC 99.
005160 NA PIC 99.
005180 02 COD-POSTO PIC 9(5).
005190 CLE-DEPX PIC XXXX.
005210 COD-POSTA PIC 9(5).
005220 CLE-ARR PIC XXXX.
005240 NO-FFN PIC 99.
005250 NO-COUL PIC 99.
006010 01 MESS.
006020 02 LMESS COMP.
006030 02 MESSAGE PIC X(400).
006070 PROCEDURE DIVISION.
006071 DEFRT.
006080 ENTER COBOL M018.
006090 M018.
006091 LECTURE DE FICHER TRAVAIL ( LA GRILLE RENSEIGNEE)
006100 ENTER GETX, BYFN, TITRES, POSTE,
006110 LA PREPARATION DE GRILLE RENSEIGNEE
006120 MOVE LGH TO LMESS.
006140 MOVE NO-DEPT TO NO-DEPTX.
006150 MOVE JU-DEPT TO JU-DEPTX.
006160 MOVE HE-DEPT TO HE-DEPTX.
006170 MOVE HE-ARR TO HE-ARRX.
006180 MOVE COD-POSTO TO COD-POSTOX.
006190 MOVE CLE-DEPX TO CLE-DEPTX.
006200 MOVE COD-POSTA TO COD-POSTAX.
006210 MOVE CLE-ARR TO CLE-ARRX.
006220 MOVE NO-FFN TO NO-FFNX.
006230 MOVE NO-COUL TO NO-COULX.
006240 MOVE GRILLE TO MESS.
006250 EMISSION DE LA GRILLE (RENSEIGNEE) CREATION D UN VOYAGE
006260 ENTER DISPLAY, MESS, MOINSUN,
006260 ENTER TERM.

```

Ve výměnné zóně /COMMON-STORAGE SECTION od řádku 4150/ je uvedena struktura vyrovnávací paměti záznamu /TITRES - řádek 5010/.

V PROCEDURE DIVISION /od řádku 6070/ se nejdříve čte záznam pracovního souboru /vyvolání makroinstrukce ENTER GETW - řádek 6100/, dále se naplní mřížka konkrétními údaji pracovního souboru /řádky 6120 až 6240/ a zobrazí se na displeji /vyvolání makroinstrukce ENTER DISPLAI - řádek 6250/.

Programový modul je zakončen ENTER TERM.

Výklad použitých makroinstrukcí :

ENTER GETW, BTF, BUFA, ID

Čtení jedné věty pracovního souboru /tato věta byla před tím zapsána do pracovního souboru makroinstrukcí PUTW/.

BTF - název pole, které je rezervováno pro uživatele v místní nebo výměnné zóně,

BUFA - název pole vyrovnávací paměti věty,

ID - identifikátor označující organizaci souboru.

ENTER DISPLAI, MESS, NMOD.

Výstup zprávy /MESS/ na obrazovku displeje.

MESS - označení pole ve výměnné zóně; obsahuje zprávu: délku /LDM - binárně/ a text / v EBCDIC/,

NMOD - modus ukončení : - 1 konec, nastavení začátku
0 zpracování dalších zpráv
n číslo následujícího modulu.

ENTER TERM.

Konec procesu, který byl vyvolán makroinstrukcí ACTIV.

/ V našem příkladu je ACTIV v předcházejícím modulu./

Z příkladu je zřejmé, že vlastní programování mřížky bude /i na jiných počítačích/ rutinní záležitostí. Při použití COBOLU je výhodou možnost týmové spolupráce. Práci rozdělíme na jednotlivé větve hlavního vývojového diagramu dialogu tak, aby jeden pracovník mohl pracovat samostatně na ucelené části /např.: vytvoření výchozí mřížky, identifikace a vyhodnocení neoprávněných vstupních dat a návraz na výchozí mřížku ap./.

5. Příklad programování dotazů a odpovědí v assembleru

Výhoda assembleru pro programování dialogu spočívá v jednoduché možnosti generovat uživatelské makroinstrukce pro přenos a zpracování zpráv. Vymezením dílčích operací a vytvořením makroinstrukcí je pro ně se paměť zbytečně nezahlcuje málo používanými podprogramy a zkracuje se doba zpracování a doba odezvy.

Příkladem programování dotazů a odpovědí je návrh systému zpracování a zobrazení informací z datového souboru na terminálu /SZIT/. Jedná se o dílčí funkci aktualizace záznamu, která byla navržena a odzkoušena na počítači UNIVAC 90/30. /4/

Programové vybavení počítače UNIVAC 90/30 úrovně DDI /Direct Data Interface = přímý přenos dat/pro tvorbu uživatelských programů předpokládá použití assembleru. Analýza řešené úlohy /přenos mezi terminálem a souborem/ ukazuje, že při aktualizaci záznamu je třeba provádět množství operací společných různým souborům. V SZIT jsou proto navrženy moduly pro tyto společné funkce a odpovídající uživatelské makroinstrukce.

Makroinstrukce FOBR /Funkce obrazovky/

Tato makroinstrukce vytváří mřížku pro zobrazení položek věty na displeji a provádí práce s vyplněním věty a mřížky konkrétními hodnotami položek. Dále FOBR deklaruje tabulku informací o používaném souboru a text nadpisu na displej. Má poziční parametry, z nichž každý popisuje jednu položku zpracovávané věty.

Je specifikován název položky, délka položky, případně název uživatelské makroinstrukce, kterou se bude položka kontrolovat. Z těchto informací vytváří makroinstrukce FOBR mřížku, ve které každá položka zaujímá jednu řádku na displeji. Na každé řádce je název položky, pole pro hodnotu položky a pole pro chybové hlášení při chybném vyplnění položky. Programátor je tedy oprostěn od nutnosti pracného formování mřížky kombinováním čtyřbytových DICE /souřadnic kurzoru/, textu a speciálních znaků.

Příklad úseku programu pro práci s terminálem na obr.5 je podle návrhu obrazovky /obr. 6./.

1.	TZPAR	POBR	36,(CU,4,CISLO UCTU :),	X
2.			(NZF,25,NAZEV FIRMY :),	X
3.			(PSC,5,PSC :),	X
4.			(MIS,17,MISTO :),	X
5.			(CBU,12,CISLO BANKOV. UCTU :),	X
6.			(ZNP,1,ZNAK PRO PRIKAZ :,RZNP),	X
7.			RBC=64,BLK=485,KLOC=0,	X
8.			KLEN=4,NADP=TOUZEMSKI PARTNER	
9.	CU	FCNT	ZL=P,ZN=N	
10.	NZF	FCNT		
11.	PSC	FCNT	ZL=P,ZN=N	
12.	MIS	FCNT		
13.	CBU	FCNT	ZN=N,ZP=A,ZD=A,ZV=N	
14.	ZNP	FCNT		
15.	RZNP	BCU	*	
16.		CLI	IZNP,C'I'	
17.		BE	RZND	
18.		CLI	IZNP,C'P'	
19.		BE	RZND	
20.		CLI	IZNP,C'V'	
21.		BE	RZND	
22.		MVC	EZNP,=C'NEPRIPUSTNY ZNAK'	
23.	RZND	BCU	*	
24.		BR	14	
25.		END		

Obr. 5. Úsek programu pro práci s terminálem

První poziční parametr v řádku 1 určuje vzdálenost začátku hodnot všech položek od levého okraje displeje. Každý poziční parametr v řádcích 1 - 6 má tyto subparametry : interní název položky v rámci fáze, délka položky, název položky. Poziční parametr v řádku 6 má navíc jméno rutiny RZNP, která bude provedena při zpracování položky ZNP. V řádku 7 jsou klíčové parametry charakteristik indexsekvenčního souboru : délka věty, délka bloku, poloha klíče ve větě. V řádku 8 jsou klíčové parametry pro délku klíče a nadpis na displeji, který se objeví po vyvolání této fáze.

V řádku 9 je makroinstrukce FCNT pro zpracování položky CU. ZL=P znamená povinnost dodržet délku 4. ZN=N znamená, že pole je číselné. Ostatní charakteristiky této položky se dosadí automaticky. Obdobně i u řádku 11. Parametr ZP=A ve 13. řádku určuje umístění čísla účtu zleva v poli pro položku a ZD=A určuje doplnění mezerami u číselné položky. ZV=N povoluje tuto mezeru nevyplňovat. U abecedněčíslených položek v řádcích 10, 12 a 14 jsou hodnoty všech parametrů dosazeny automaticky.

V řádcích 15 až 24 je uživatelská makroinstrukce RZNP, která kontroluje výskyt předem určených znaků I, P, V. Tato rutina přebírá řízení až po řídicím modulu SZIT vyvolaném makroinstrukcí FCNT /řádek 14/.

Takto zapsaný zdrojový modul se zpracovává standardním postupem /spojovacím programem aj./, připojí se cílový modul a vzniklá fáze se uloží do knihovny.

6. Závěr

Programování dialogů v režimech dotaz-odpověď je komplexnější prací než tomu bylo dosud u samostatných programů pro agendy. Novinkou je bezprostřední práce operátora s informacemi. Kromě časových nároků vyžadují projekty velmi často poskytnutí komplexní informace na terminálu. Operátor potom pracuje se třemi druhy informací : z reálného procesu, z terminálu, vlastní zkušeností.

Pro optimalizaci programování můžeme použít buď dodaného programového vybavení nebo uživatelských makroinstrukcí "na míru".

TUZEMSKY PARTNER

CISLO UCTU :	5553
NAZEV FIRMY :	RADIANA
PSC :	75601
MÍSTO :	HOVĚZI
CISLO BANKOV. UCTU :	123456789
ZNAK PRO PŘIKAZ :	P

Obr. 6. Rozmístění informací na displeji

Literatura :

- /1/ Martin, J.T.: Systems Analysis for Data Transmission. Englewood Cliffs, Prentice-Hall 1972.
- /2/ STRATEGE. Manuel d'utilisation et d'opérations. Paris, CII 1974.
- /3/ Botek, Z., Štětina, I., Toman, P.: Interaktivní systémy. In : SOPSEM'79. Zborník referátov, Bratislava, VVS 1979.
- /4/ Janata, J.: Systém zpracování a zobrazení informací z datového souboru na terminálu. Praha, VŠE 1979.