

Ing. Stanislav TOMKO
INORGA, pobočka Košice

Úvod

Návrh a tvorba údajovej základne jednoduchých, účelovo orientovaných informačných systémov patrí v súčasnosti k činnostiam často sa opakujúcim. Kvalita projekčných a realizačných prác má priamy dopad na funkčné vlastnosti riešeného systému. Obmedzené finančné prostriedky používateľov a dostupné počítačové zdroje v mnohých prípadoch neumožňujú aplikáciu komerčného databázového systému. Pravdou je tiež, že väčšina aplikácií vystačí s jednoduchým systémom ovládania súborov, a nie komplikovaným systémom riadenia bázy dát.

Príspevok si kladie za cieľ zoznámenie s technológiou tvorby AIS a používaným programovým vybavením, tak ako sa používa pri riešení niektorých úloh v INORGE, pobočke Košice. Hlavný dôraz je venovaný tvorbe údajovej základne na základe modulovej koncepcie v operačnom prostredí počítačov radu SMDP.

Charakteristiky prístupu

Pri súčasnom tempe nasadzovania malej výpočtovej techniky do praxe, používateľské organizácie narážajú na problémy dostupnosti vhodného aplikačného programového vybavenia, resp. zápasia s nedostatkom kvalifikovaných programátorských kapacít. Nasadenie dostupných databázových systémov naráža na naše špecifické problémy, ako sú: minimálna konfigurácia výpočtového systému, nedostatok projekčných organizácií, neúmerne dlhá doba riešenia a pod. Používatelia sú nútení riešiť úlohy vlastnými, aj keď skromnými prostriedkami, pričom výsledky nie vždy splnia očakávanie.

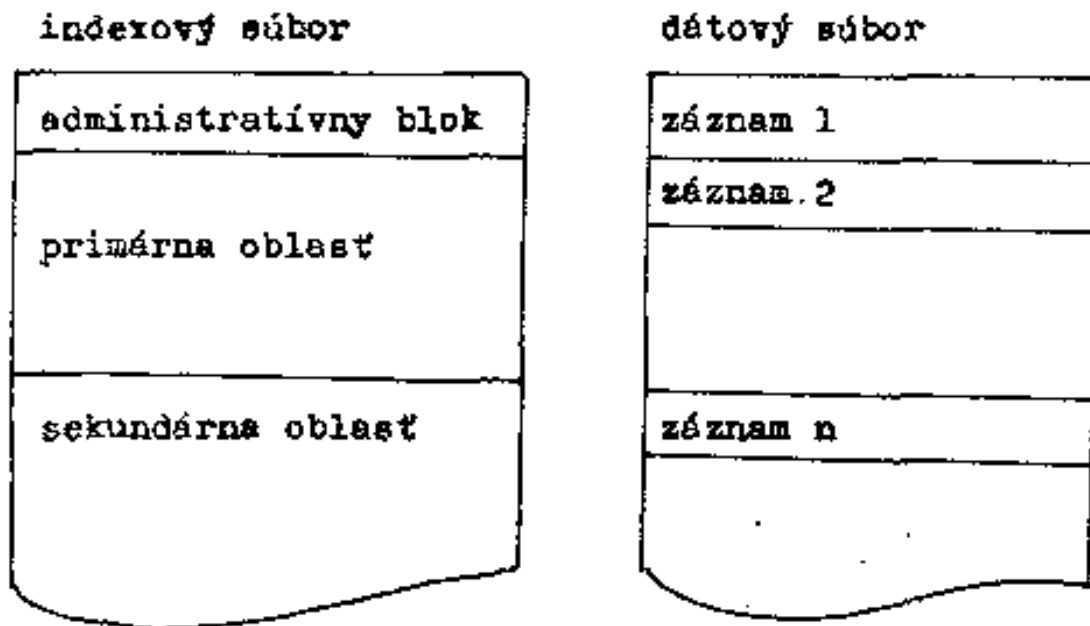
Naším cieľom, na základe získaných skúseností, bolo vyvinúť také nástroje, ktoré by pri minimálnych nárokoch na zdroje boli schopné vytvoriť podmienky pre efektívny a korektný vývoj aplikačného programového vybavenia, a to ako vo fáze návrhu, tak aj pri realizácii. Pretože vývoj nástrojov automatizovaného projektovania IS je v našich podmienkach prakticky ukončený, sústredili sme sa na vývoj prostriedkov pre racionalizáciu tvorby údajovej základne. Výsledkom nášho snaženia je otvorený súbor modulov na úrovni riadenia súborov a záznamov, ktoré sú využívané na základe konkrétnych potrieb tvorených programov. Je síce pravdou, že komerčný systém umožní v určitých prípadoch rýchlejšiu realizáciu, je však pre používateľa uzatvorený, obmedzuje čiastočne jeho kreativitu. Analógiu je možné pozorovať v stavebníctve, ktoré vo veľkom merítku využíva prefabrikáciu, uzavreté konštrukčné systémy. Výsledkom je nie príliš vábny vzhľad našich nových sídlisk. Avšak už malé prispôbenie niektorých komponentov, alebo aplikácia netradičného prvku spôsobí oživenie vzhľadu, zmenu kvality.

Moduly, ktoré v súčasnosti využívame predstavujú úroveň práce so súborami a záznamami. K tomu nás viedli tieto dôvody :

- názory na dátové modely sa rôznia, avšak dobrý systém pre prácu so súborami umožňuje aplikáciu každého modelu,
- väčšina aplikčných programov má podobné požiadavky na údajovú základňu, ktoré môžeme zhrnúť do týchto základných operácií : zápis, rušenie záznamu, modifikácia celého obsahu, alebo časti záznamu, vyhľadanie záznamu na základe určitej podmienky. Na úrovni súborov je to : vytvorenie, inicializácia, otvorenie, prípadne zrušenie. Častou požiadavkou je súčasný prístup viacerých úloh k súborom.
- z praxe vieme, že väčšina súčasných aplikácií požaduje jednoduchý systém pre prácu so súborami namiesto veľkých, komplexných a komplikovaných systémov riadenia bázy dát.

Práca so súbormi

Našou snahou bolo využiť iba služby FCS, vzhľadom na zvýšené pamäťové nároky RMS, pri zaistení indexsekvenčného prístupu. Súbor v našom ponímaní reprezentuje dvojica fyzických súborov : indexový a dátový súbor. Ich štruktúra je zrejmá z obr. 1.



Obr. 1

Súbor tohto typu poskytuje prostredníctvom dostupných modulov pomerne bohaté možnosti pre aplikáciu rôznych prístupových metód. Ako príklad uvedieme, že záznam v dátovom súbore je prístupný prostredníctvom :

- relatívnej adresy v dátovom súbore,
- hodnoty kľúča, dĺžka ktorého je maximálne 16 znakov,
- hodnoty časti kľúča,
- hodnot sekundárnych kľúčov, ktorých počet závisí na úvahe riešiteľa, prakticky každá položka/atribút môže byť uvažovaná ako sekundárny kľúč.

To, či primárny kľúč bude unikátnym v rámci súboru zostáva tak-

tiež závislé na riešení daného problému. Na rovnakom type súboru je postavený mechanizmus tvorby a obsluhy invertovaných súborov. Invertovaný súbor uchováva aktuálne tabuľky smerníkov pre rýchly výber záznamov na základe hodnôt jednotlivých položiek záznamu. Má to veľký význam pri tvorbe dotazov na údajovú základňu z hľadiska časových nárokov. Počet invertovaných súborov a stupeň inverzie je vecou návrhu, pretože jedná sa o určitý stupeň redundancie uloženia dát a zvýšenie réžie pri prevádzkovaní.

Aplikačný program komunikuje so súborami prostredníctvom volaní podprogramov, k čomu je ešte potrebné deklarovat' v programe dátové oblasti, ktorých štandardné popisy sú k dispozícii v zdrojovom tvare. Súčasne je možné pracovať so štvoricou súborov.

Zdieľanie súborov

Súčasný prístup viacerých úloh k určitej množine súborov je taktiež riešený bez použitia služieb pre riadenie záznamov. Použitá je zásada, že k súborom má prístup iba jedna úloha, tzv. dátový manažér, ktoré pokrýva základné funkcie požadované pre navrhovaný súbor aplikačných programov. Manažér na základe konfiguračnej tabuľky komunikuje s jednotlivými úlohami, ktoré požadujú akciu na niektorom zo súborov. Týmto mechanizmom je pomerne jednoduché zaistiť konzistentnosť súborov a ochranu dát. Komunikácia je možná nielen v rámci jedného počítača, ale po overení sa uvažuje o práci v počítačovej sieti typu hviezda. V súčasnosti je k dispozícii obecný typ manažéra, ktorý môže spolupracovať súčasne až so 16 úlohami. V prípade špeciálnych požiadaviek je potrebné regenerovať novú konfiguračnú tabuľku s väzbou na nové požiadavky. Manažér komunikuje s úlohami prostredníctvom "SEND/RECEIVE" mechanizmu cez komunikačný blok premenlivej dĺžky, závislej od veľkosti záznamu obsluhovaného súboru. Generovanie konfiguračnej tabuľky je jednoduché,

tvorí sa interakčne pomocou štandardného povelového súboru. Tabuľka musí odrážať konfiguráciu realizovaného systému, t.j. počítače, úlohy a funkcie jednotlivých úloh. Definuje sa pomocou troch príkazov : "COMP", "TASK" a "FUN", kde je potrebné zadať číslo počítača, mená úloh a kódy funkcií. Pre ilustráciu uvedieme hierarchicky zápis tabuľky, ktorá popisuje programový systém realizovaný na jednom počítači. Je zložený z manažéra súborov a 10 dialógových úloh :

COMP	Ø	hostiteľský počítač Ø
TASK	HMAN	úloha HMAN - manažér
FUN	FUNØ	funkcia Ø - čítanie záznamu
FUN	FUN1	funkcia 1 - zápis záznamu
FUN	FUN2	funkcia 2 - rušenie záznamu
:		
FUN	FUN9	funkcia 9 - ukončenie úlohy
TASK	TSH11	úloha TSH11
FUN	FSH11	funkcia FSH11 úlohy FSH11
:		
TASK	TSH11Ø	úloha TSH11Ø
FUN	FSH11Ø	funkcia FSH11Ø úlohy TSH11Ø

Štruktúra tohto systému je v zjednodušenej forme zobrazená na obr.2. Úloha HMAN je aktívovaná ako prvá, jediná pracuje so súborami. Prijaté požiadavky, ktoré vlastne predetavujú elementárne V/V operácie vybavuje sekvenčne.

Papís základných modulov

Pre lepšiu orientáciu v našom prístupe uvedieme stručný popis základných modulov. Na úrovni súborov sa využívajú tieto moduly :

CREATE - Vytvorenie a inicializácia súboru

Modul vytvára dvojicu súborov t.j. indexový a dátový súbor. Na základe vstupných parametrov alokuje diskový priestor a vypočíta parametre indexového súboru, veľkosti jednotlivých oblastí.

CONNECT - Pripojenie súboru

Zabezpečuje otvorenie žiadaného súboru. Prítom je načítaný administratívny blok do príslušného bafra aplikačného programu.

DELETE - Zrušenie súboru

Používa sa v prípade potreby zrušenia súborov počas behu aplikačného programu.

Pre prácu so záznamami je určených viac modulov, tak aby pokryli čo najširšie požiadavky realizátorov. Okrem bežných modulov READO, WRITEO a DELETEO, ktoré slúžia pre čítanie, zápis a rušenie záznamov prostredníctvom kľúčov, alebo adres majú najširšie použitie :

FINDO - Vyhľadanie záznamu

Modul je určený pre vyhľadanie a načítanie záznamu na základe časti hodnoty primárneho kľúča.

NEXTO - Čítanie ďalšieho záznamu

Číta sa nasledujúci záznam s rovnakým kľúčom ako pri predchádzajúcom volaní, alebo nasledujúci záznam v zmysle relatívnej adresy v dátovom súbore.

READR - Čítanie záznamu na základe hodnoty atribútu

Prostredníctvom invertovaného súboru je vyhľadaná a načítaná veta s hľadanou hodnotou príslušného atribútu-položky. Taktiež je dispozíciou hodnota o počte záznamov, vyhovujúcich podmienke. Jeden invertovaný súbor sa môže vzťahovať k viacerým dátovým súborom, podmienkou je unikátne meno položky.

WRITER - Zápis do invertovaného súboru

Na základe hodnoty položky a jej názvu, modul zapíše do príslušnej tabuľky v invertovanom súbore index /ukazovateľ na záznam/. V prípade, že tabuľka nie je ešte vytvorená, automaticky ju vytvorí.

DELETR - Rušenie indexu

Zruší sa žiadaný index z príslušnej tabuľky v invertovanom súbore.

Predchádzajúce tri moduly pre prácu s invertovanými súborami sú určené pre priebežnú aktualizáciu indexov. Služobným programom HRF je možné vytvoriť, resp. aktualizovať jednoduchým spôsobom aj dosť často.

Komunikáciu medzi jednotlivými úlohami v prostredí operačného systému DOS RV zabezpečujú dva moduly pre distribúciu správ :

LSFM - Odoslanie správy

Dátový bafr je odoslaný konkrétnej úlohe na základe kódu funkcie v zmysle použitej konfiguračnej tabuľky. Veľkosť preneseného bafra je maximálne 512 slov.

RCVM - Príjem správy

V prípade, že úloha má vo fronte správu, modul zabezpečí prenos bafra do dátovej oblasti úlohy. Ináč je úloha v stave čakania, aktivuje sa až v prípade, že niektorá iná úloha jej vyšle správu.

Ďalšie použiteľné programové prostriedky je možné rozdeliť do týchto skupín :

- štandardné popisy dátových oblastí a komunikačných blokov,
- moduly pre prácu s kľúčmi v indexových súboroch
- pomocné moduly pre tvorbu dialógu, hash algoritmov a pod.
- obslužné úlohy informačného charakteru o stave súborov, úlohy umožňujúce tvorbu dotazov, renováciu, kopírovanie, spájanie súborov a pod.

Projekčná fáza využíva projekčné postupy deklarované technológiou projektovania SDT [1]. Použitie tejto technológie sa opiera o programový nástroj pre automatizované projektovanie SAD/SADEX, ktorý obsahuje celý rad projekčných štandardov pre všetky etapy projektovania, resp. umožňuje používateľovi vytvoriť vlastný, špecifický jazyk pre popis projektovaného systému, optimálny z hľadiska všetkých účastníkov tvorčieho procesu. Tento aparát dovoľuje realizovať dátový model riešeného systému, ktorý obsahuje funkčnú, dátovú a procesnú štruktúru, pričom poskytuje možnosť tvorby technologickej dokumentácie s dobrou vypovedacou schopnosťou. Analytické možnosti systému umožňujú iteratívne sa približovať k modelu, ktorý sa najviac približuje predstave projektanta. Hĺbka popisu systému závisí od použitého jazyka pre popis, resp. od úrovne naviazujúcich programových prostriedkov ako sú transformačné programy, predkompilátory a pod. V súčasnosti sa využíva automatické generovanie popisu dát v zdrojovom tvare pre kompilátor jazyka fortran. Týmto je zabezpečený jednotný dátový popis pre každý aplikačný program. Okrem tohto popisu programov až do úrovne algoritmu resp. volaní doporučených štandardných modulov nám zároveň slúži ako zadanie pre aplikačného programátora.

Nasledujúci príklad ilustruje prezentovaný prístup na realizáciu programového vybavenia pre riadenie skladového hospodárstva na báze počítačov SMEP. V projekčnej fáze bol použitý systém SADEX. Funkčná a dátová štruktúra sú, hoci len útržkovite podané formou technologickej dokumentácie, v tvare v akom s ňou pracuje projektant. Použitý jazyk pre popis systému bol zvolený aj vzhľadom na použitý programovací jazyk, FORTRAN 77.

Príklad :

SSH1 : SEKCIA "Funkčná štruktúra subsystému SH"
SH1 : SUBSYST "Riadenie sklad.hospodárstva" TYP MIX
SH11 : SU "Sledovanie pohybu na skladoch" TYP INTERACT
SH111 : UL "Evidencia skladu nakup.mater." TYP INTERACT
SHF11 : FUN "Prehľad stavu nakup. mater." TYP INTERACT
REALIZUJE TASK TSHF11
VSTUP "Parametre prehľadu" / TERMINAL
IDM : -hľadaného materiálu-
VYSTUP "Údaje o stave materiálu" / TERMINAL
STAVN : -aktuálny stav- TVAR 9999
SKUPR : -rezervácie na zakázky- POCET 20
IDZ : -identifikátor zakázky- TVAR X/6/
:
SHF12 : FUN "Príjem nakup.materiálu" TYP INTERACT
REALIZUJE TASK TSHF11
VSTUP "Prijatý materiál" / TERMINAL
IDM : -kód prijatého materiálu-
PMN : -prijaté množstvo-
:
SHF13 : FUN "Výdaj nakupovaného materiálu" TYP INTERACT
REALIZUJE TASK TSHF13
:
SSH111 : xx Spracovateľská štruktúra SH111 xx
MENU1 : MENU "Voľba funkcie" TYP INTERACT
VYKONAJ SHF11, SKOK-NA MENU1
VYKONAJ SHF12, SKOK-NA MENU1
VYKONAJ SHF13, SKOK-NA MENU1
:

```

SHLATA : SEKCIA "Dátové štruktúry subsystému SH" TYP D-STRUKT
BD-ON  : SEKCIA "Štruktúra on-line BD SH" TYP D-STRUKT
FMAT   : SET "Údaje o materiáloch" PRIMAR FILE MAT
IDM    : SKUPINA "Prevádzkový kód materiálu" TVAR X/6/
SKUPM  : TEXT "Skupina materiálu" TVAR XXX
PCM    : CKOD "Poradové číslo mat-skup" TVAR 999
NAZM   : TEXT "Názov materiálu" TVAR X/24/
MCENA  : KCS "Jednotková cena materiálu" TVAR 99999.99
JKPOVM : CKOD "Plné číslo materiálu" TVAR X/17/
MJM    : TEXT "Merná jednotka" TVAR XX
        ...ATR... PRIMAR == IDM

SHLU   : SEKCIA "Katalóg používateľských typov dát" TYP D-STR
LATUM  : SKUPINA "Podnikový dátum"
ROK    : CKOD "Rok" TVAR 9
MES    : CKOD "Mesiac" TVAR 99
        :

SHTASKS : SEKCIA "Katalóg úloh subsystému SH" TYP KATALOG
TSH11  : TASK "Prehľad stavu nakup. materiálu"
        JAZYK F77 VERZIA 1 AUTOR Hlava-Jan
        SEKCIA "Návrh algoritmu programu"
        INCLUDE HCOM1, FMAT
        CALL CONNECT FILE MAT AREA 1
        CALL CONNECT FILE ZAK AREA 2
        CALL CONNECT FILE CIS AREA 3
        CALL CONNECT FILE IMAT AREA 4
        CALL SHM1 "obrazovka 1"
        DO WHILE "koniec vstupu"
        ACCEPT IDM
        GET RECORD VIA IDM AREA 1
        IF "veta sa nenašla" THEN
        TYPE "materiál nie je na sklade"
        ELSE
        TYPE NAZM,MCENA,STAVM
        END-IF
        END-WHILE
        :

```

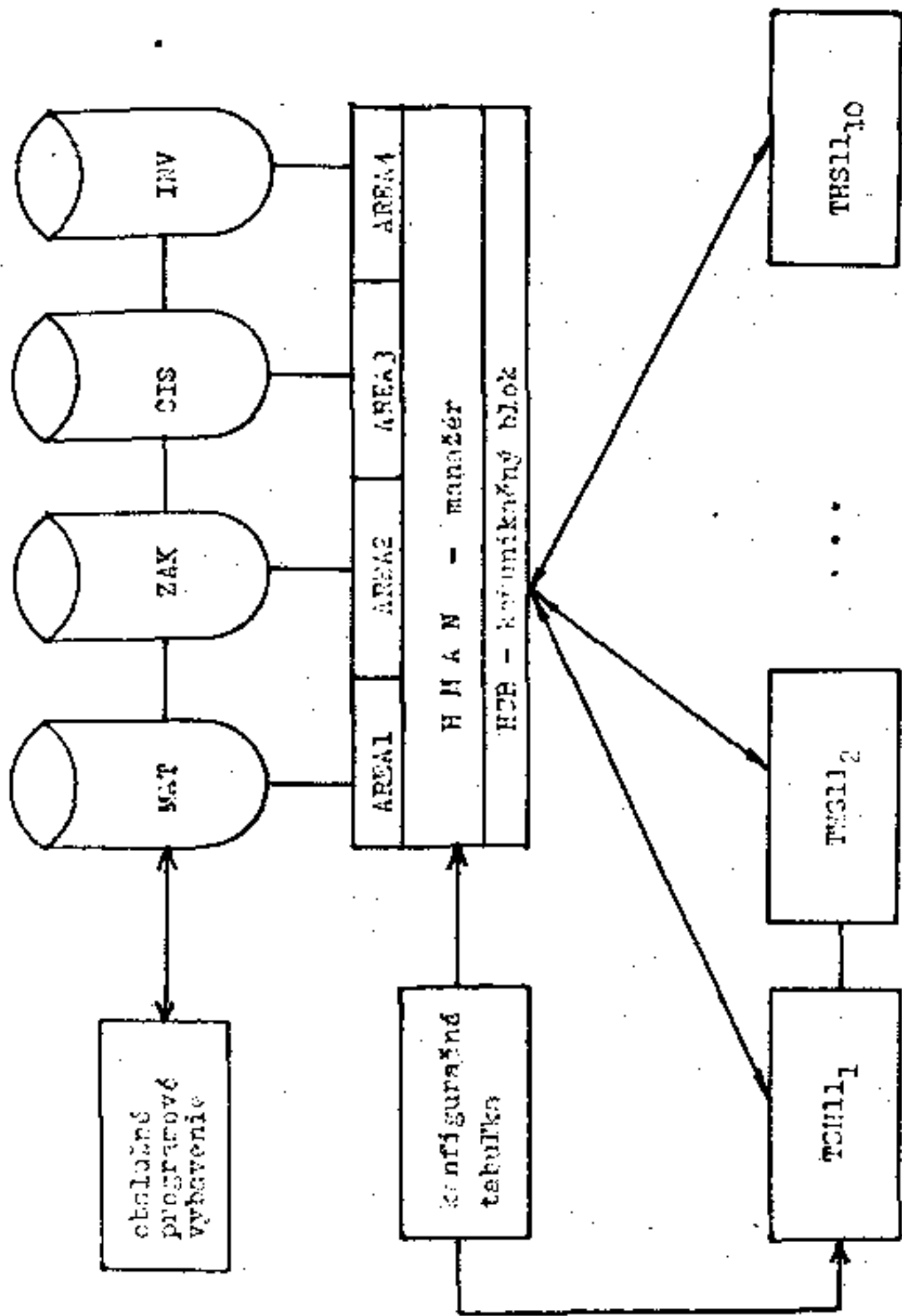


Schéma riešeného systému z príkladu /jednej zo skupín úloh/ je na obrázku 2. Súborný súbor je obsluhovaný manažérom HMAN, ktorý má každý zo štvorice súborov MAT /materiály/, ZAK /zá- kazky/, CIS /číselníky/ a INV /invertovaný súbor/ fixne pri- pojený k segmentom AREA1 až AREA4. Úlohy TSH11 - TSH110 pred- stavujú používateľsky orientované dialógy pre prácu s údajo- vou základňou.

Záver

Prezentovaná technológia tvorby jednoduchých, účelovo orien- tovaných AIS bola overená na viacerých úlohách s dobrými výsled- kami a tvorí východziu základňu pre ďalší rozvoj nástrojov auto- matizovaného projektovania a realizácie AIS. Použitý modulárny prístup je zvlášť vhodný vzhľadom na malé nároky na zdroje a otvorenosť k ďalšiemu rozvoju.

Literatúra

- /1/ Tóth A., Bial Š., Jakabčín M., Kuruc V. : Automatizované projektovanie ASR. Sešity INORGA, č.86, Praha, 1982
- /2/ Tóth A. : A Software Computer-aided System Description. System Description Methodologies. D.Teichrow and G.David /Editors/, Elsevier Science Publishers, IFIP, 1985
- /3/ Tomko S. : Základné informácie o programovom systéme SALEX - SMEP. Dom techniky ČSVTS Ústí nad Labem, 1985