

ŠTANDARDIZÁCIA PROGRAMOV PRE AKTUALIZÁCIU BÁZY DÁT

Databázový systém IDMS (Integrated Database Management System) používaný vo výpočtovom stredisku Ekonomického výskumného ústavu chemického priemyslu tvorí jeden zo základných nástrojov budovaného Automatizovaného systému riadenia pre generálne riaditeľstvo VHM Slovchemie. Vyžiadal si vytvorenie programového prostriedku pre aktualizáciu báz dát, ktoré reprezentujú hlavnú údajovú základňu.

Z množiny činností potrebných pre narábanie s bazou dát pokrýva Aktualizačný systém (neskôr AS), vytvorený na našom ústave, množinu užívateľských programov pre aktualizáciu údajov v báze dát. Znamená to, že zabezpečujú zapisovanie a údržbu všetkých údajov sústredených v báze dát. AS tvorí samostatnú časť databázového systému. Realizuje aktualizáciu údajovej základne primárneho charakteru. Pracuje v dávkovom režime spracovania údajov, ktoré vstupujú v prevažnej miere vo forme diernych štítkov. Pri pomerne nízkom počte programov zabezpečuje systém aktualizáciu všetkých údajov v báze dát dostatočne efektívnym spôsobom. Spôsob a forma požiadaviek na aktualizáciu zo strany užívateľa predstavuje vyplnenie príslušného typu formulára. Tento je navrhnutý tak, aby bol pre užívateľa po vecnej stránke zrozumiteľný a zároveň aby bolo možné podľa neho nedierovať štítky.

Súbor vstupných diernych štítkov je v jednom behu spracovaný Aktualizačným systémom, pričom riadenie aktualizácie zabezpečuje spracovanie jednotlivých typov dát zodpovedajúcim programom. Architektúra a vnútorná stavba AS v sebe zohľadňuje zároveň aj myšlienku zefektívnenia tvorby programov systému. Programy dostali jednotnú štruktúru, ich jednotlivé časti sú už predprogramované. Zvyšok, určený na kódovanie programátorom, predstavuje podľa okolností 15 - 50 % riadkov výsledného programu. Táto metodika tvorby aktualizáčnych programov používajú všetky čiastkové úlohy riešené na našom ústave, stala sa v našich podmienkach záväznou.

Pre prácu s bázou dát treba v podstate zabezpečiť tri základné, navzájom sa líšiace činnosti. Ide o

- a) - aktualizáciu údajov t.j. ukladanie, zmena a rušenie existujúcich údajov
- b) - výstup informácií a operácie s nimi t.j. rôzne výstupné zostavy, klasické - pomocné súbory a pod.,
- c) - zmena údajov v báze dát na základe výpočtov.

AS rieši v plnom rozsahu problematiku prvého bodu. Môže byť tiež použitý na nevyhnutné úvodné napĺňanie bázy dát, ktoré môže predstavovať zápis značného množstva dát. Potom ho možno bezo zmeny využívať na samotné priebežné aktualizovanie údajov podľa požiadaviek vznikajúcich v priebehu rutínnej prevádzky bázy dát. Pre obe spôsoby použitia vykazujú programy dostatočnú efektívnosť.

Ciele

Pri návrhu AS bolo cieľom vytvoriť nástroj, ktorý by spĺňal viacero stanovených požiadaviek. Predovšetkým to bola požiadavka, aby začévanie aktualizáčnych dát nevyžadovalo od užívateľa špeciálne znalosti napr. o tom v akom poradí musia byť vstupné údaje, v akej forme sú reprezentované v báze dát a pod. Ďalej bolo nutné, aby bol systém použiteľný pre všetky naše rozpracované úlohy. Vzhľadom na dávkový spôsob spracovania žiadalo sa sústreďiť popis činnosti do jednej procedúry v jazyku riadenia prác (JCL). To by zaručovalo dostatočnú prehľadnosť pri prevádzke databázového systému. Cieľom bolo tiež vytvoriť optimálny vzťah medzi typom vety v báze dát a množinu programov, ktoré budú aktualizovať daný typ vety. Pretože vybudovanie AS pre všetky naše rozpracované úlohy predstavovalo vytvorenie značného počtu programov, cieľom bolo tiež štandardizovať tvorbu týchto programov, zefektívniť proces ich vytvárania. Nasledujúci popis aktualizáčného systému odráža spôsob, ktorým sme sa snažili dosiahnuť naznačené ciele.

Popis aktualizáčnej procedúry

Prehľadnosť práce AS zabezpečuje to, že jeho činnosť vskutku

určuje jedna procedúra na úrovni jazyka riadenia prác. Táto procedúra, znázornená schématicky na obr. č. 1, sa skladá z piatich krokov, pričom dva predstavujú presunutie programových modulov z pásek do knižnice na disku. Táto činnosť vykonávajú programy LLS - systému (Load Librarian System - zlepšovací návrh Ing. Karla Vrbenského), ale z nášho hľadiska sú zaujímavé hlavne zostávajúce kroky. V prvom z nich sa aktivuje program, vykonávajúci predspracovanie aktualizáčnych dát (na obrázku označený menom WKBTD). Predspracovanie spočíva vo vylúčení dát s chybným transakčným kódom (pozri ďalej), v rozšírení a doplnení transakčných štítkov na transakčné vety, ktoré sú už schopné triedenia, a vo vytvorení riadiacich viet transakčnej dávky, čo sú tzv. úvodné a koncové vety pre každý typ transakcie (pojmy sú ozrejmene v časti Stanovenia základných pojmov). Pri predspracovaní sa tiež vytvorí informácie o programoch potrebných k samotnej aktualizácii a zapíšu sa do pomocného súboru (FLLS). Navyše program stanoví, či ide o aktualizáciu rutínnej, tzv. ostrej bázy dát alebo testovacej bázy dát.

Ďalším krokom procedúry sa triedia dáta tak, aby bola aktualizácia vykonávaná vždy v dohodnutom poradí. Užívateľ teda môže zoradiť dáta do ľubovoľnej postupnosti na vstupe, triedením podľa presne stanovených sortovacích kritérií sa vždy dosiahne správne poradie. AS je teda zabezpečený voči nepozornosti užívateľa, i voči nehode akou je rozsypanie balíka diernych štítkov.

Vlastnú aktualizáciu, resp. jej riadenie zabezpečuje riadiaci modul (WKBRM) aktivovaný v poslednom kroku procedúry. Program v závislosti od vstupných dát volá príslušné aktualizáčne programové moduly. Množina modulov potrebných na spracovanie daných výstupných dát bola už určená pri predspracovaní a je k dispozícii v knižnici systému LLS na disku.

Kompilátor pre programovací jazyk PL/I optimalizujúca verzia umožňuje dynamické volania, zavádzanie a uvoľňovanie programových modulov, ktoré boli samostatne skompilované ako externé

procedúry. Pre kompilátor jazyka PL/1 verzia F bola vytvorená verzia AS, v ktorej sa tento riadiaci program vytvára pospájaním všetkých aktualizáčnych modulov do prekrývanej štruktúry. Znamená to určitú nevýhodu, pretože po každej úprave niektorého z modulov je nutné opätovné pospájanie, avšak metodika riešenia má naďalej pôvodný význam. S týmto problémom sa však potýkame prevažne vo fáze logického ladenia modulov systému, vo fáze rutínnej prevádzky už prakticky odpadá. Táto verzia bola vytvorená z toho dôvodu, že bolo treba zaviesť AS na podnikové výpočtové strediská, ktoré používajú kompilátory PL/1-F.

Pozn.: Uvedená verzia v sebe odráža tiež rozdiely v programovani jazykov oboch verzii. Niektoré typy príkazov ako je SELECT, ktoré sú použiteľné iba v optimalizujúcej verzii, boli prepísané vo verzii F do príkazov, ktoré nahrádzajú ich funkciu.

Stanovenie základných pojmov

Štruktúra riadiaceho programu odráža spôsob práce AS. Pre ujasnenie treba uviesť niektoré základné pojmy. V popise bázy dát sú vyznačené a popísané tzv. typy viet. Zapísaním údajov do bázy dát vznikne výskyt vety, ktorý reprezentuje jeden z typov vety. Táto dvojica pojmov v analógii s reálnym svetom zodpovedá dvojici: typ objektu - výskyt objektu (reálny objekt). Výskyt vety bol zvolený za jednotkové množstvo práce AS. Používateľ si teda môže byť vedomý existencie typov viet a výskytov viet, ale zaobíde sa aj bez týchto vedomostí. Zadanie aktualizácie pre neho predstavuje vyplnenie užívateľsky zrozumiteľného formulára.

Nad jednotlivým výskytom vety možno vykonať v zásade štyri rôzne operácie:

1. pridanie nového výskytu vety
2. zmena údajových položiek výskytu vety
3. vymazanie výskytu vety
4. zmena kľúčových položiek výskytu vety.

Od toho sú odvodené názvy operácií, hovoríme potom o pridaní, zmene, výmaze a kľúčovej zmene. Každú takúto operáciu nad výskytom vety voláme transakcia.

Analogicky potom operácia pre daný typ vety sa nazýva typ transakcie.

Tým sme vytvorili množinu typov transakcií, ktoré umožňujú užívateľovi aktualizovať bázu dát v celom rozsahu. Každý transakčný štítok obsahuje na dohodnutej pozícii tzv. transakčný kód. Každému typu transakcie zodpovedá jeden transakčný kód. Štítok obsahuje číslo štítku transakcie, nakoľko jedna transakcia sa môže skladať podľa potreby z viacerých štítkov (maximálne 9), aby bol poskytnutý dostatočný priestor pre zápis dátových položiek. Triediacimi hľadiskami pri usporadúvaní transakčných dát sú potom: transakčný kód (resp. poradové číslo, ktoré danému typu transakcie prislúcha v postupnosti spracovania), kľúčová časť dátových položiek (ich dĺžka je pre daný typ transakcie presne stanovená) a konečné poradové číslo štítku (pri triedení už transakčná veta). Týmto triedením sa dosiahne správne poradie dát na vstupe a zgrupovanie typov transakcií.

Je zrejmé, že počet programov takéhoto systému môže byť až štyrikrát väčší ako počet typov viet. Niektoré typy viet však tvoria v báze dát iba logické spojenie medzi vetami iných typov a netreba pre ne písať zvláštne programy. Operácie s nimi sú vedľajším účinkom iných aktualizáčnych programov. Navyše pre niektoré typy viet netreba mať k dispozícii všetky typy operácií, takže počet programov býva v skutočnosti nižší.

Pozn.: AS obsahuje navyše štandardne ďalšie dva programy. Je to inicializačný modul (na obr. č. 1 WKBINT) a terminačný modul (WKBTRM). Obidva moduly zisťujú resp. overujú dodatočné informácie. Napr. či je správne číslo dávky v úvodnom štítku dávky, čítajú tzv. vrcholové vety z danej bázy dát, zapisujú do vrcholovej vety údaje metodicko-organizačného charakteru a pod.

Zhrnutie vyššie uvedených myšlienok možno formulovať takto:

- a) Operácia - pridanie, zmena, výmaz alebo zmena kľúčových položiek
- b) Transakcia - operácia nad konkrétnym výskytom vety

- c) Typ transakcie - všeobecný pojem zavedený pre AS označujúci operáciu nad typom vety, ktorej vykonanie možno žiadať od AS.
- d) Množina typov transakcií - celá škála aktualizáčnych aktivít, ktoré zabezpečujú AS.

Ak sa vrátíme k riadiacemu programu pre aktualizáciu, je zrejmé, že transakčná dávka, t.j. postupnosť transakcií na vstupe AS, je spracovaná postupne tak, že typ transakcie je vykonaný jedným zodpovedajúcim programovým modulom. Tento je aktivovaný pri rozpoznaní príslušného kódu transakcie a spracuje všetky dáta s týmto kódom. Priebežne vytvára protokol o aktualizácii, v ktorom uvádza orientačné údaje (meno organizácie - z bázy dát, číslo strany protokolu, stručný popis transakcie, identifikáciu modulu, čas a dátum vytvorenia modulu a aktualizácie a iné údaje) a opis vstupných dát.

Všetky programy pre aktualizáciu pracujú pod Centrálnou verziou IDMS, takže daný krok aktualizáčnej procedúry v jazyku riadenia prác neobsahuje žiadne DD-štítky, okrem štítkov popisujúcich pomocné súbory. (Pozri obr. č. 1). Riadiaci modul v súčasnosti obsluhuje podsystémy zodpovedajúce piatim nezávislým aplikáciám. Obsahuje doväďna 288 typov transakcií. Veľkosť modulu po preklade je 92kB.

Transakčný modul - typový program

Z popisu riadiaceho modulu vyplýva, že jednotlivé aktivity pri aktualizácii budú vykonávané príslušnými programami, ktorých počet sa zvyšuje s narastajúcim počtom typov viet. Bolo teda na mieste vytvoriť metodický prostriedok pre efektívne programovanie týchto činností. Vytvorenie takejto metodiky bolo ovplyvnené návrhom samotného AS a naopak. Z toho dôvodu bolo nutné popísať jeho výstavbu.

Bola teda zostavená kostra aktualizáčného programu pre náš AS, v ďalšom transakčný modul, ktorá zahrňuje určité štandardne sa vyskytujúce objekty a činnosti (deklarácie, kontrola dát, algoritmus čítania a spracovania dát a pod.). Samotný transakčný modul, ako prvok systému, ale aj svojou vnútornou stavbou bol

zostavený na základe modulárnosti. Jeho jednotlivé vnútorné časti sú členené do samostatných celkov. Grafické znázornenie skladby transakčného modulu je na obr. č. 2. Modulárnosť umožňuje jednoduchú orientáciu, prehľadné dopĺňanie kódu zdrojového textu, jednoduchú modifikáciu prípadne ďalšie rozšírenie.

Transakčný modul bol zároveň vytváraný s ohľadom na štruktúrované programovanie. Pri vytvorení typového modulu a pri návrhu jeho stavby sme sa snažili sledovať prístup zhora na dol.

Postupné zjemňovanie a rozčleňovanie štruktúry vykonávaných činností možno sledovať už pri sledovaní zdrojového textu riadiaceho modulu resp. transakčného modulu. Tento prístup je podporovaný tiež využívaním programových prostriedkov jazyka PL/1 určených na tento účel - štruktúrovaných príkazov.

Konečný produkt týchto snáh je typový program. V praxi sa používa zdrojový text tzv. vzorového modulu, ktorý zodpovedá navrhutej štruktúre transakčného-typového modulu. Časť zdrojového textu v ňom zostáva bezo zmien (napr. deklarácie niektorých objektov, algoritmus čítania vstupných dát a pod.), časť vyžaduje úpravu textu programátorom v závislosti od navrhovanej činnosti (napr. meno modulu, deklarácie IDMS, generovanie modulu chybových oznamov preprocesorom, určenie dĺžok niektorých štandardných premenných a iné) a časť navrhuje programátor celú, (formálna kontrola údajov, logická kontrola, samotné spracovanie). Posledne uvedená časť je vlastne predmetom zadania pre vytvorenie programového modulu. Práca pri vytváraní zdrojového textu je uľahčená aj tým, že programátor priamo tvorí program dopisovaním a úpravou textu cez obrazovkový terminál podľa vopred pripravenej predlohy, zvyčajne modifikovaného listingu vzorového modulu.

Transakčný modul potom vykoná tieto činnosti :

- a) Číta transakčné údaje
- b) Vykonáva formálnu kontrolu údajov
- c) Transformuje ich do formy, v ktorej sa nachádzajú uložené v báze dát

- d) Kontroluje logické vzťahy medzi údajmi na vstupe a údajmi existujúcimi v báze dát
- e) Vyhľadáva v báze dát tie výskyty viet, pre ktoré sa má transakcia vykonať
- f) Vykonáva vlastnú aktualizáciu bázy dát t.j. pridávanie výskytov viet, zmenu položiek vo výskytoch viet, vymazávanie výskytov viet, resp. zmenu logických vzťahov medzi zadanými výskytmi viet
- g) Priebežne vyhotovuje protokol o vykonanej aktualizácii resp. chybové a informačné oznamy o mimoriadnych situáciách pri spracovaní transakčných údajov vo forme prístupnej užívateľovi. Zároveň je vyhodnotený stupeň závažnosti situácie a podniknuté opatrenia. Buď sa vydá len informačný oznam, alebo sa prejde na spracovanie ďalšej transakcie bez ukončenia práve spracovávanej transakcie, alebo je ukončené celé spracovanie okamžite.
- h) Odkladá informácie potrebné v prípade potreby reštartu AS.

Pre typový-transakčný modul bol tiež navrhnutý jednotný spôsob komunikácie modulu s okolím. Komunikácia je zabezpečená:

- a) cez parametre externej procedúry tvoriacej samotný transakčný modul
- b) cez bázu dát
- c) cez pomocný stavový súbor AS.

Parametre obsahujú informačnú štruktúru a stavovú vetu. Informačná štruktúra pokrýva objekty potrebné pre prácu transakčného modulu, ktoré mu odovzdáva riadiaci modul. Stavová veta obsahuje odkladné informácie o priebehu doterajšieho spracovania. Treba zdôrazniť, že každý transakčný modul predstavuje samostatnú aktualizáciu bázy dát t.j. obsahuje aj príkaz otvorenia bázy dát, príkaz READY resp. príkaz ukončenia práce s ňou, príkaz FINISH.

Na tomto mieste sa žiada poznamenať, že pre zefektívnenie práce systému obsahuje typový - vzorový modul príkaz vytvorenia kontrolného bodu, príkaz COMMIT. Vykonanie príkazu je užitočné

hlavne v prípade pádu aktualizácie s väčším množstvom transakčných dát. Príkaz sa štandardne vydáva po 100 vykonaných transakciách a zabráni tomu, aby sa muselo spracovanie vykonať znovu od začiatku, nakoľko sa zároveň odloží informácia o tom, kde aktualizácia skončila. Upravená procedúra AS pre reštart pokračuje v spracovaní od ďalšej nasledujúcej transakcie za posledne úspešne vykonanou transakciou, za ktorou bol vydaný príkaz COMMIT. Štandardne tiež na záver svojej činnosti transakčný modul poskytuje na výstupe štatistické informácie o aktivitách modulu v báze dát. Tieto údaje priebežne vytvára a sprostredkuje Centrálna verzia IDMS a sú programovo dostupné. Dávajú užitočný obraz o tom ako program realizuje to, čo sme od neho požadovali, t.j. koľko prístupov na disk požadoval, koľko viet čítal v danom behu a pod.

Podrobnejší popis funkčných častí transakčného modulu neuvádzame. Schématicky sú naznačené na obr. č. 2 a na záver uvedieme len niektoré podstatné skutočnosti.

Všetky transakčné moduly budú mať predpísanú zhodnú štruktúru. Táto určuje:

- miesta deklarácie objektov
- tok riadenia v module
- rozdelenie transakčného modulu do funkčne ucelených procedúr
- spôsob komunikácie týchto procedúr
- úseky, v ktorých sa líšia jednotlivé moduly určené k individuálnemu programovaniu.

Celkove obsahuje každý modul 8 hlavných častí - segmentov (podľa terminológie štruktúrovaného programovania). Pri použití tejto metodiky sa samozrejme vyskytli programy s dosť rozdielnym stupňom zložitosti a rozsiahlosti. Možno však stanoviť, že veľkosť takto budovaných programov sa pohybuje spravidla v rozsahu od 30 do 65 kB.

Rozsah použitej metodiky

Naznačená metodika tvorby aktualizáčnych programov vytvorená a aplikovaná na našom ústave sa používa ako záväzná pri

návrhu aktualizácie bázy dát vo všetkých aplikáciách. AS tvorí v rámci riešenia všetkých čiastkových úloh budovaného ASR pre VZJ Slovchémia záväzný a určujúci materiál, či už je to problematika na úrovni GR alebo na podnikovej úrovni. Štyri projekty riešiace čiastkové úlohy už boli odovzdané do rutínnej prevádzky vo výpočtovom stredisku pri ŠVÚCHP. Pre ilustráciu uvádzame ich názvy s príslušným počtom transakčných modulov patriacich danej aplikácii.

Sledovanie obchodných činností - vývoz	-30	transakčných programov
Priebežné sledovanie hlavnej výroby	-76	"-
Starostlivosť o základné prostriedky - úroveň GR	-36	"-
Starostlivosť o základné prostriedky - podniková úroveň	-56	"-

Zároveň sa riešia ďalšie dve čiastkové úlohy pomocou tejto metodiky mimo ústav a to na podnikoch. Dva projekty sú v štádiu Vykonávacieho projektu:

Sledovanie investičnej výstavby - Duslo Šaľa	-21	transakčných programov
Sledovanie mater. - techn. zabezpečenia - Chemlon Humenné	-53	"-

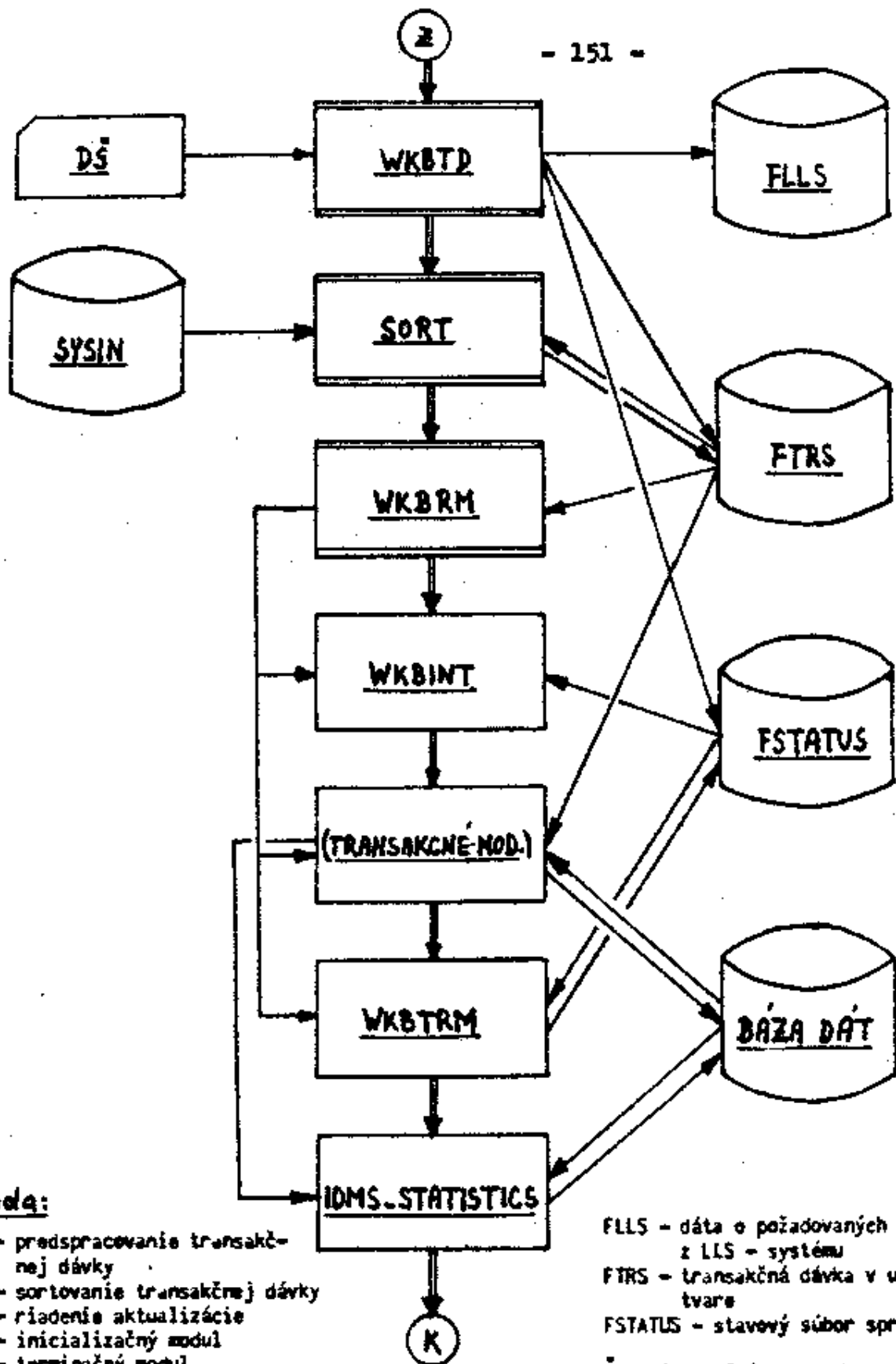
Pre ilustráciu je tiež uvedená v prílohe tabuľka č. 1, v ktorej sú uvedené náhodne vybrané údaje o "rýchlosti" spracovania pomocou AS.

Zhrnutie

Na základe viacročných skúseností získaných pri navrhovaní, realizácii a prevádzkovaní viacerých projektov využívajúcich pre aktualizáciu príslušných báz dát služby resp. metódu AS možno usúdiť na viacero kladných prínosov.

Značne sa znížila pracnosť písania programov, zjednodušil ich návrh a dokumentácia. Do značnej miery sa obmedzil vznik

chýb zhodných častiach programov (vďaka predprogramovanej pred-
iohe). Uľahčila sa ich čitateľnosť, zvýšila prehľadnosť, zjednodu-
šila modifikovanosť i v prípade, že programátor nie je autorom.
S tým úzko súvisí údržba programov už odovzdaných pre rutinné
spracovanie. V prípade, že sa vyskytne pri používaní programu
chyba, prevádzkový programátor má uľahčenú úlohu pri jej loka-
lizácii a oprave a to už z vyššie uvedených dôvodov (čitateľ-
nosť, prehľadnosť, jednotná stavba). V konečnom dôsledku možno
AS ako taký modifikovať pre použitie v ďalších aplikáciách a to
jednoduchými úpravami v niektorých častiach jeho riadiacích
programov. Ešte jednoduchšie je to pri rozširovaní existujú-
cich aplikácií (napr. pridávanie nových transakčných modulov),
kedy ide len o úpravu internej tabuľky, resp. rozšírenie ria-
diacej štruktúry SELECT o ďalšiu klauzulu a pod. Z tohoto hľa-
diska je AS ešte schopný určitého vývoja v prípade ďalších po-
žiadaviek.



Legenda:

- WKBD - predspracovanie transakčnej dávky
- SORT - sortovanie transakčnej dávky
- WKBRM - riadenie aktualizácie
- WKBINT - inicializačný modul
- WKBTM - terminačný modul
- IDMS-STATISTICS - výpis štatistiky

- FLLS - dáta o požadovaných moduloch z LLS - systému
- FTRS - transakčná dávka v upravenom tvare
- FSTATUS - stavový súbor spracovania
- DŠ - dierne štítky
- SYSIN - riadiace príkazy pre program SORT

Obr. č. 1

MOUL: PROC (INFO);

DEKLARÁCIA IDMS

- štruktúry viet IDMS
- inicializačné príkazy IDMS
- stavové vety STATUS0, STATUS1

DEKLARÁCIE GLOBALNÝCH PREMENNÝCH

- štruktúra parametrov INFO
- transakčné dáta TDATA
- pomocné premenné

EXEKUTÍVNA ČASŤ HLAVNEJ PROCEDÚRY

- ON podmienky
- volá čítanie transakčných dát RDFK
- volá logickú kontrolu
- volá spracovanie
- odkladá reštartovacie údaje

RDFK: PROC;

- čítanie výskytu transakčných dát, konverzia dát

FKCT: - celková kontrola transakčných dát

FKDS: PROC;

- formálna kontrola dátových položiek jednotlivých štítkov

FKKL: - formálna kontrola kľúčovej časti dát

LOGK: PROC;

- logická kontrola transakčných dát
- vyhľadanie potrebných výskytov viet v báze dát

SPTR: PROC;

- vykonanie aktualizácie v báze dát

CYB: PROC;

CN1: ENTRY /číslo správy,

CN2: ENTRY /číslo správy, text správy/;

- výstup správ do protokolu
- riešenie chybového stavu

<u>Spracovanie</u> Názov: (JOB): Aplikácia:	<u>Hrubý čas</u> Začiatok: Ukončenie:	<u>Čistý čas</u> (CPU) min. sek.	Počet typov transakcií v spracovaní	Celkový počet zadaných transakcií	Celkový počet transakčných štitkov
XDBQAV Priebežné sledovanie hlavnej výroby	13 : 46 : 19 14 : 00 : 38	28,91 s	3	86	516
XDBQAV Priebežné sledovanie hlavnej výroby	11 : 24 : 53 11 : 56 : 13	30,64 s	1	160	160
ZP01QGA Sterostlivosť o zák- ladné prostriedky GR	9 : 22 : 08 9 : 41 : 13	1 m 14,61 s	6	191	261
IV01QIA Sledovanie investičnej výstavby	17 : 16 : 41 17 : 22 : 33	30,68 s	11	493	724

Tabuľka č. 1.: Niektoré údaje o vybraných aktualizáciách