

Miroslav Vykydal, ZVT ČSAD BRNO

První aplikace IDMS v extrémních podmínkách, organizace práce realizačního týmu, návrh logické sítě a jeho modifikace, vznik navigačního algoritmu, vznik zpracovatelských algoritmů, kompilace a ladění programů.

1. Úvod

Abychom ušetřili čas těch čtenářů, kteří pracují v jiných podmínkách, popíšeme stručně vstupní situaci. Je rozhodnuto přejít na databázový systém IDMS. Jako první krok je podán návrh na zpracování netriviálního příkladu nasazení IDMS s cílem nejen zvládnout prostředky IDMS, ale produkovat údaje vhodné pro uživatele. Vzhledem k tomu, že je nutno zajišťovat běžné úkoly v analýze a programování /nové projekty, změny ve stávajícím zpracování/, je možné vyčlenit zpočátku extrémně malý realizační tým. Skládá se z databázového administrátora, problémového analytika a programátora. Databázový administrátor ovládá detailně možnosti IDMS, avšak na počátku nemá samozřejmě žádné vlastní praktické zkušenosti. Navíc nemá téměř žádné odborné znalosti z problemové oblasti, která byla pro nasazení IDMS zvolena a ovládá pouze základy jazyka Cobol. Problémový analytik je rámcově seznámen s prostředky, které pro organizaci a přístup k datům nabízí IDMS. Díky analytickým a programátorským pracím, kterých se zúčastnil v rámci automatizace zpracování údajů ze zvolené problemové oblasti, je s touto oblastí podrobně seznámen, zná i vhodné konzultanty z řad uživatele. Programátor ovládá programovací jazyk Cobol, o systému IDMS ví pouze z doslechu, nelze jej z časových důvodů zatěžovat školením nebo studiem manuálů. Stručně shrnuto - databázový administrátor dokáže technicky realizovat navrženou síť, ale pouze po logickém návrhu od problémového analytika. Ten dokáže formulovat požadavky na výstupní údaje, ale přístupový algoritmus umí vypracovat pouze databázový administrátor. Pomocné výpočty a realizaci tiskových sestav zvládne pouze programátor, neboť databázový administrátor to neumí a problémový analytik nemá čas.

Z hlediska časových kapacit si situace vynutila, aby databázový administrátor se věnoval nasazení IDMS celou svou časovou ka-

pacitou, problémový analytik jen polovinou /zpočátku/ až třetí-
nou /později/ své pracovní doby a programátor v omezeném inter-
valu 2-3 týdny celou svou kapacitou. Zadání pro programátora ne-
smělo obsahovat žádné mimořádnosti, zvláště ne nic z oblasti
IDMS, neboť jak již bylo uvedeno, byl programátor databankový
analfabet.

Extrémní situace byla v dostupné konfiguraci počítače - EC
1030, operační paměť 256 KB, 12 diskových jednotek 7,25 MB s
operačním systémem OS MFT. Bylo jasné, že při provozování data-
banky nebude možné zpracovávat žádnou provozní rutinní úlohu.
Práci s IDMS bude možné provádět pouze v čase určeném pro ladě-
ní programů a v době minimálního provozního zatížení /mimo zá-
věrkové a termínované práce/.

Každý uvažující člověk si po dočtení textu až do tohoto mis-
ta musí zákonitě položit otázku, zda je vůbec reálná aplikace
IDMS v tak extrémních podmínkách a co může přinést.

2. Důvody

Aplikace se zdála reálná, ačkoliv pro důkaz tohoto tvrzení
byl prakticky jediný argument - termín předání výsledků uživateli
nebyl pevně stanoven, práce započala pod heslem - až to bude, tak
se pochlubíme. Od realizace se však očekávaly především dva efek-
ty. Samozřejmě jako u každé novinky šlo o potření názorů typu "to
nikdy v našich podmínkách nepůjde" praktickým důkazem "že to jde
a docela dobře". Hlavní efekt se však očekával v oblasti změny
myšlení při návrzích přístupových /navigačních/ algoritmů. Vyšli
jeme z předpokladu, že i po prostudování všech příruček, manuálů,
zahraníční literatury a vyslechnutí mnoha referátů, úvah i nepod-
ložených sdělení nám pořád chybí to hlavní - praktické zkušenosti.
Vždyť kuchařem se člověk obvykle nestává jen četbou receptů, ale
zpravidla jejich realizací, byť zpočátku třeba drasticky neúspěš-
nou.

3. Práce realizačního týmu

Nejprve bylo nutné provést návrh logické sítě. Problémový
analytik znovu prověřil datovou základnu problémové oblasti. Byl
proveden odhad maximálních četností jednotlivých údajů a maximál-
ních rozsahů kombinací identifikačních údajů /např. průměrně pro

jednoho přepravce bude maximálně přepravováno deset různých druhů zboží/. Zkoumala se činnost uživatele, kdy a s jakými údaji pracuje, které údaje používá nejčastěji, jaké výkazy a přehledy vyhotovuje, co musí znát pro určité rozhodnutí. Vycházelo se z dřívějších zkušeností a poznatků, neboť oblast byla již částečně zasažena sekvenčním zpracováním, ale byly vzaty v úvahu i požadavky uživatele, které byly dříve odmítány jako obtížně řešitelné. Svoje poznatky sděloval problémový analytik databázovému administrátorovi ve formě výroků /např. závod přepravuje jeden měsíc pro přepravce zboží, sleduje podíl přepravy pro přepravce dle závodů nebo zboží, srovnáváme přepravu určitého zboží s celkovou přepravou za určité časové období/. Po návrhu logické sítě následovalo její prověřování. Požadavky na výběr údajů jsou realizovány symbolickou navigací po síti, požadavky jsou rozděleny na standardní /přesně definované/ a nestandardní /obecně definované/. Např. standardně budou vyžadovány po uplynutí měsíce pro každý závod statistické výsledky v členění dle přepravců a zboží. Náhodně bude vyžadováno porovnání výsledků dosažených pro určité přepravce u zvoleného závodu s celkovými výsledky celého podniku.

Postupně byl návrh sítě upřesňován tak, aby byly efektivně plněny požadavky na výběr údajů. Problémový analytik zformuloval základní zadání standardních programů a parametrických programů /předprogramové dotazy/.

4. Vznik algoritmu navigace a zpracování

Na základě zadání problémového analytika zpracoval databázový administrátor základní algoritmus navigace po databázi. Při generování databáze bylo v názvech rekordů a jejich položek používáno systematicky číslování. Např.

RECORD NAME ZBOZI.

RECORD ID 103.

LOCATION MODE DIRECT.

WITHIN STRUKTURY-AREA.

05 AB3-CISLO-ZBOZI PIC XX.

05 AB3-JMENO-ZBOZI PIC X/15/.

Několik vyjimek, které se zdály nevýznamné, způsobily později při komunikaci mezi členy týmu řadu nedorozumění. V navigačních algoritmech byla část zpracování údajů naznačena pouze symbolicky pomocí příkazu PERFORM. Např. PERFORM ZPRACOVANI-VYKONU THRU EX12

nebo PERFORM ZPRAVA-O-NEEXISTUJICIM-IZOZI THRU EX20. Vždy byla uváděna v poznámce čísla rekordů, které je možno použít /jsou v daném místě CURRENCY/. Problémový analytik pak zpracoval zadání pro programátora.

Každý příkaz PERFORM představoval jedno zadání. Byly specifikovány požadavky na operace s údaji, např. sečti do součtu za závod, vypočti plnění plánu dle vzorce skutečnost/plán * 100. Odvolávky na konkrétní proměnné byly řešeny přes popis v RECORD DESCRIPTION, který obdržel jako základní dokumentaci i programátor. Např. zadání na výpočet plnění plánu pak mělo tvar:

Výpočet plnění plánu:

```
/a/ IF A09-VYKON-SKUT ≠ 0 AND A09-VYKON-PLAN≠0
    THEN PLNENI-XXXX =  $\frac{XXXX-SKUT}{XXXX-PLAN} * 100$ , XXXX=/TUNY,TKM,TRIBA/,
        PLNENI tiskni ve tvaru 999.99.
/b/ IF A09-VYKON-SKUT ≠ 0 AND A09-VYKON-PLAN≠0
    THEN PLNENI se nepočítá a netiskne /neplánovaný výkon/
/c/ IF A09-VYKON-SKUT = 0 AND A09-VYKON-PLAN≠0
    THEN PLNENI se nepočítá a netiskne /nulové plnění/
/d/ IF A09-VYKON-SKUT = 0 AND A09-VYKON-PLAN=0
    Nemůže nastat!
```

Protože bylo nutné používat 6 diskových jednotek /vzhledem k předpokládanému rozsahu dat/, byla jednou z klíčových otázek otázka spotřeby strojového času. Navigační algoritmus prověřil po stránce formální databázový administrátor ručně. Zpracovatelský algoritmus reprezentovaný samostatnými moduly byl doplněn o formální náležitosti /úvod programu v Cobolu/ a o deklaraci použitých proměnných z RECORD DESCRIPTION. Takto vzniklý program /bez navigačního algoritmu/ byl formálně zkompilován a odstraněny syntaktické chyby v běžném režimu ladění programů. Teprve potom spojil problémový analytik navigační a zpracovatelský algoritmus v konečnou verzi programu /pomocné deklarace proměnných z RECORD DESCRIPTION byly vyloučeny/, která byla bez ztrát zkompilována.

Při ověřování správnosti jednodušších programů /aktualizace, tisk standardních sestav/ bylo využíváno pomocných tisků, které podávaly zprávy o průběhu běhu programu. Tyto tisky byly z konečných verzí programu odstraněny. Program realizující parametrický dotaz na databázi byl odlaďován postupně. Celý program se skládá ze tří částí - zpracování parametrů, výběr údajů, zpracování a tisk výsledků. Hlavní - navigační algoritmus byl odlaďován tak,

že zpracování parametrů bylo situováno přímo naplněním proměnných pro výběr a místo tisku výsledné sestavy byly opisovány vybrané údaje z databáze. Obdobně byly samozřejmě /navíc v normálním režimu/ laděny zbylé dvě části programu.

4. Závěr

Ačkoliv podmínky nebyly optimální a pro rutinní práci chybí např. terminál, byly získány základní praktické zkušenosti. Samozřejmě, že např. práce u terminálové sítě bude zcela odlišná, ale to pro nás zůstává stále hudbou budoucnosti. Byl odzkoušen proces tvorby a ověřování správnosti programového vybavení v systému IDMS, v praxi byla ověřena metoda týmové práce a základní možnosti systému IDMS.