

Ing. M. Balíková, RNDr. V. Karásek, Ing. V. Podlana
Výpočetní centrum ČKD Praha

1. Ú v o d

Stále širší využívání počítačů v oblasti řízení staví do popředí požadavek na vytvoření vhodných nástrojů pro racionalizaci procesu automatizace. Rozvoj automatizovaných systémů řízení je obecně charakterizován zvyšujícími se nároky na množství, přesnost, spolehlivost a pohotovost získání informací.

I když v současném stadiu uplatnění automatizace se stále důrazněji požaduje, aby automatizace nebyla pouze technicky vyspělou formou zpracování dat, ale kvalitativní změnou v účinnosti řízení, přesto je třeba vidět, že právě data jsou rozhodujícími zdrojem pro získání informací pro řízení a že tedy racionální práce s daty je jedním ze základních předpokladů pro koncipování forem ASŘ.

Způsob a rozsah zpracování dat a nárůst jejich objemu vyvolává i růst vazeb uvnitř datových souborů, mezi soubory, mezi programy, které data zpracovávají, a tím i mezi jednotlivými subsystémy. Tato situace nutně vede k tomu, že je nezbytné zajímat se o vlastnosti dat, o jejich organizaci, možnosti a četnost přístupu k nim tak, aby nebyly narušeny vazby na různých úrovních. Zajištění jedinečnosti popisů dat při malém objemu lze manuálně a udržení celkového přehledu vyžaduje pouze soustavný a zodpovědný přístup pracovníků jednotlivých systémů.

Pro velké systémy je však nutno řešit evidenci a formulaci dat centrálně. To vedlo k myšlence vytvoření samostatného automatizovaného systému pro údržbu informací o tvaru dat, o jejich uspořádání v subsystémech a pro evidenci vazeb mezi položkami, soubory, programy a subsystémy.

Jedním z problémů, který se u velkých systémech objevuje, je redundantnost informací o datech. Datové položky, jejichž jména

jsou různé a jsou použita na více místech v subsystému, popisují jednu informaci a přitom jsou definovány různým způsobem, ať již co do rozsahu, formátu nebo úrovně v souboru.

Neexistuje centrální přehled definic dat a datových struktur v souborech, a tím ani údaje o celkovém rozsahu datové základny pro centrální řízení.

Dalším nedostatkem je nekonzistentnost definic datových položek, kdy pod stejným názvem položky jsou definovány dva nebo více údajů s různým významem.

Nejednotnost a nejednoznačnost definic datových položek vede k jejich různému označování a chápání ve výstupních sestavách a v dokumentaci s nutným dopadem na nedorozumění mezi analytiky, programátory a uživateli. Toto se projevuje prakticky ve všech stádiích projekční činnosti při řešení automatizovaných systémů řízení i při údržbě programů.

Všechny tyto problémy vyplývají z toho, že rostoucí objem dat, rozsah a četnost jejich zpracování se v určitém momentu stanou neúnosnými pro dosavadní vžitý systém manuální evidence obsahu datové základny a vazeb a vznikne potřeba sledovat informace o datech a vazbách samostatným automatizovaným systémem.

Informace uložené v tomto systému je možno pak vhodnými softwarovými prostředky kontrolovat, třídít a udržovat.

Integrované systémy informací o používaných a zpracovávaných datech, které centrálně obhospodařují definice dat a umožňují standardizaci dokumentace o datech, byly již ve světě vyvinuty a jsou úspěšně využívány. Jsou nazývány Slovníky dat (Data Dictionary System - DDS) nebo systémové adresáře dat (SAD).

2. Základní vlastnosti systémového adresáře dat

Systémové adresáře dat nesou nejen informace o souborech, větách a položkách, ale i o vazbách mezi nimi. Nejznámějšími systémy jsou dnes např. systémy slovníků a adresářů dat firmy IBM (IBM DB/DC Dictionary), dále ICL 2900 Data Dictionary, British Rail Data Dictionary System, systém Lexicon a další.

Stejnou funkci plní i soustava katalogů VVS Bratislava. Rovněž firma Cullinane - tvůrce IDMS - vyvinula vlastní Data Dictionary, který je možno používat jak v rámci IDMS, tak i samostatně.

Vzhledem k tomu, že žádný z těchto softwarových prostředků nebyl a není pro VVČ ČKD Praha dostupný, byl v r.1979 na základě skutečné potřeby, studia a zkušeností pracovníků VVC vypracován návrh vlastního adresáře dat s cílem realizovat systém, který bude v co největší možné míře přenositelný na libovolný jiný počítač.

Systémový adresářem dat (SAD) budeme rozumět jakýkoliv ucelený systém, který uchovává informace o datových položkách, větech souboru a souborech, které organizace používá a zpracovává. Další funkcí SADu je aktualizace a tisk uchovávaných informací a generování části Data Division programů v jazyce COBOL.

Základní funkce SADu zabezpečuje jazyk SADu - Data Dictionary Language.

Některé systémy umožňují uchovávat i informace o programech, které data zpracovávají, o periferních jednotkách, o uživatelích apod. Tyto druhotné funkce nebyly zatím do základní koncepce SADu ve VVC ČKD Praha zahrnuty.

3. Koncepce řešení SADu ve VVC ČKD Praha

Při návrhu SADu vycházeli autoři ze skutečnosti, že systém bude realizován ve výpočetním středisku, které je již v provozu dlouhou dobu a kde se již běžně zpracovává velký objem rutinních úloh. Jejich datová základna existuje, je využívána, ale není dosud souhrnně evidována. Systém kromě již výše zmíněné přenositelnosti na jiný počítač, musí mít tedy schopnost pojmout informace o položkách a souborech stávajících tak, jak jsou v hotových programech používány a náležitě přijímat i další informace o datech úloh nových.

Prvním krokem bylo navržení vlastních systémových souborů SADu, a to jejich obsahu, formátu, verzí a přístupu k nim.

SAD má základní informace uloženy ve třech diskových souborech, které lze chápat jako tři úrovně přístupu k datům. Soubory byly nazvány : POLOZKA, SCHEMA a SUBSCHEMA. Toto uspořádání informací SADu vychází z toho, že každý soubor dat, se kterým pracujeme lze popsat dvěma způsoby, a to logicky a fyzicky.

Fyzický popis (soubor SCHEMA) zahrnuje informace o vlastnostech souboru samotného (medium, organizace souboru, formát vět atd.) a dále popis vět souboru na úrovni nedělitelných elementárních datových položek. Tento fyzický popis je pro každý soubor, resp. každý typ věty souboru jediný, plně charakterizuje jeho obsah, ale^{ne} postihuje žádné sdružování položek do vyšších logických celků (skupinové položky apod.).

Logický popisem (soubor SUBSCHEMA) rozumíme popis vět souboru včetně členění na skupinové položky podle specifických požadavků konkrétních programů, které se souborem, resp. s větou pracují. Jeden soubor může mít více logických popisů a představují různý pohled na soubor/větu. Cílem zavedení pojmu "logický popis" do SADu je dát programátorovi možnost popsat větu pouze těmi položkami, s nimiž v určitém programu bude pracovat a v takových úrovních, jaké jsou pro řešení daného algoritmu účelné. V každém případě je nutné pro definici SUBSCHEMA důsledně vycházet ze zadání fyzického popisu, tj. ze SCHEMA.

Tyto dva základní druhy popisů souboru, resp. jeho vět jsou plně využívány v analytické a programátorské práci a jejich aktuálnost a jednoznačnost vzájemné vazby a formátu je předpokladem pro úspěšný postup prací na úlohách. Fyzické popisy vytváření dokumentací souborů pro analytickou práci, zatímco logické popisy odpovídají požadavkům na úrovni programů a slouží pro automatické vytvoření sekce souboru (FILE SECTION) vkopírováním do Cobolekých programů (viz dále).

Jednoznačnost a vzájemnou vazbu fyzických a logických popisů zajišťuje třetí systémový soubor SADu - soubor POLOZKA. Je v něm uložen popis všech položek fyzického popisu souboru a tím i těch položek logického popisu, které je třeba pro dané SUBSCHEMA deklarovat.

V základní koncepci řešení bylo dále nutno rozhodnout o prostředku pro dorozumění uživatele se systémem. Vzhledem k orientaci na programovací jazyk COBOL a na záměr postupného přechodu k hanco dat, bylo rozhodnuto respektovat normu CODASYL, která specifikuje jazyk pro práci s bankov dat a vyhovuje potřebám VVO.

Programové řešení SADA je provedeno v jazyce PASCAL a využívá příkazů ICL pro přímý přístup do systémových diskových souborů. Překladač jazyka PASCAL firmy ICL v sobě totiž zahrnuje příkazy pro přímý přístup k datům v diskových souborech.

4. Popis struktury a obsahu souborů SADA

Jak vyplývá z předchozí části, pracuje současná verze SADA se třemi základními druhy informací a souborů. Jsou to :

fyzické popisy - soubor SCHEMA

logické popisy - soubor SUBSCHEMA

popis elementárních položek - soubor POLOZKA.

Až posud jsme se zabývali pouze uspořádáním souborů SADA z hlediska postavení dat ve zpracování. Byl to analytický pohled na soubor daný plným fyzickým popisem souboru a jeho vět, logický popis souboru a vět důležitý na úrovni technického řešení práce s daty, tj. konkrétní aplikace v programu a nakonec (ne ovšem svým významem) soubor elementárních položek, z nichž vychází fyzický popis zcela a logický popis částečně.

Kromě tohoto pohledu na soubory respektuje SAD ještě hledisko tématické. Je to příslušnost dat k jednotlivým projektům. Každý z výše zmíněných tří souborů je členěn dle projektů. Projekt se tak stává prvním stupněm v hierarchii uvnitř souborů.

V souboru POLOZKA existuje navíc možnost definovat položky společné pro více projektů. Můžeme je nazvat např. "obecně platná". Protože i tyto položky musí mít tématickou příslušnost, byl v souboru POLOZKA kromě skutečných projektů definován "fiktivní projekt", nazvaný COMMON. Patří sem položky, které by měly mít jednotný popis v rámci celé organizace, jako je např. běžné datum, číslo závodu, střediska, číslo zaměstnance, zakázky, základního prostředku, materiálů atd.

Takové položky je zbytečné deklarovat pro každý projekt samostatně a hlídat, zda mají ve všech zadáních stejný popis. Měly by být uloženy v souboru POLOZKA jen jednou, a to právě v části COMMON.

V těch organizacích, kde SAD zavádíme pro stávající projekty bude výběr těchto obecně platných položek problematický. Při přípravě dat pro SAD totiž zjistíme, že uložení např. běžného data se nejen mezi projekty, ale někdy i mezi soubory jednoho projektu liší. Samo zavedení SADu nemůže ovšem tuto nejednotnost vyřešit ani není reálné provádět změny ve všech zdrojových programech, které s příslušnou rozpornou položkou, např. datem, pracují. V prvním kroku je třeba respektovat obsah a rozsah takových položek, definovat je případ od případu do souboru POLOZKA a provádět sjednocení postupně; nejprve u všech nových souborů nahrávaných do souborů SCHEMA a SUBSCHEMA a dále v rámci úprav existujících programů (inovace, údržba).

Pro soubor fyzického a logického popisu byla použita stromová struktura (viz obr.1) s úrovněmi projekt-soubor-věta-položka. Na úrovni projekt a soubor je kromě vlastního popisu projektu a souboru uložen i seznam a adresy prvků úrovně o jeden stupeň nižší. Tedy na úrovni projekt je seznam souborů projektu s adresy jejich uložení a na úrovni soubor seznam vět souboru a adresy uložení. Na úrovni věta je uložen pouze seznam položek věty (bez adresy uložení položky). Pro fyzický popis - SCHEMA se všechny položky vyhledávají v souboru POLOZKA dle svého názvu. Pro logický popis - SUBSCHEMA se položky buď rovněž vyhledávají v souboru POLOZKA nebo je přímo zadán jejich popis a rozsah. Skutečný obraz každé věty, tj. určení adres položek ve větě, se tvoří při každém dotazu na příslušnou větu, ať se jedná o výpis informací o souboru, aktualizaci souboru nebo včlenění popisu do COBOLského programu. Tento způsob opakovaného výpočtu adres vět byl zvolen proto, aby aktualizace souborů POLOZKA mohla probíhat nezávisle na souborech fyzického a logického popisu souboru, resp. vět.

Umístění (místo pro uložení informace) jednotlivých položek do souboru POLOZKA je určeno na základě algoritmu, který vychází z kódu projektu, do něhož položka patří a zejména položky.

Dosavadní zkušenosti ukázaly, že algoritmus pro uložení plně vyhovuje a rozmístění položek je rovnoměrné. Stejný algoritmus platí i pro vyhledávání položek v souboru při získávání informací pro adresaci ve větách.

5. Zkušenosti se zaváděním současné verze SADu ve VVC ČKD Praha

V současné době je zahrnuto v systému SAD cca 70% informací o datové základně z pěti nosných projektů. Vstupní data pro SAD byla připravována prakticky již v průběhu celého období programování SADu, jakmile byl formulován jazyk pro SAD. Naplňování jednotlivých systémových souborů SADu mohlo být zahájeno ihned po odladění jednotlivých programů pro pořízení a aktualizaci souborů.

Ukázalo se, že získání informací o již existujících datech v projektech je velmi pracné. Znamená to důsledné procházení popisů souborů v programech a v dokumentaci a utřídění informací pro vytypování položek společných pro více souborů projektu, uspořádání logických popisů vět souborů v programech a určení vazeb na základní fyzický popis souboru.

Jen díky zahájení přípravy vstupů v dostatečném předstihu a díky aktivnímu přístupu pracovníků k pořízení dodatečné evidence existujících dat, se podařilo systémový adresář naplnit.

Popisy dat všech nových projektových řešení se do SADu začleňují předem a logický popis souborů a vět je již součástí specifikace programu.

Přehledy získávané ze SADu jsou zdrojem přesných a užitečných informací o již existujících datech a vazbách mezi nimi a jsou důležité zejména pro údržbu programů. Výpisy fyzických popisů souborů slouží při analýze funkcí jednotlivých programů a stanovení datových struktur. V oblasti programovacích prací je výhodou již výše zmíněná jednoznačnost specifikace logického souboru. Programátor může logický popis uložený v souboru SUBSCHEMA včlenit automaticky do FILE SECTION zdrojového programu příkazem

$\left\{ \begin{array}{l} \text{FD} \\ \text{OL} \end{array} \right\}$ COPY jméno $\left\{ \begin{array}{l} \text{souboru ze SUBSCHEMA} \\ \text{věty ze SUBSCHEMA} \end{array} \right\}$

Příkaz COPY způsobí, že se před překladem včlení do programu popis souboru a zadaných vět upravených do standardního formátu pro FILE SECTION! Takto doplněný zdrojový program se potom překládá. V příkazu COPY je možno zadat parametr PREFIX resp. SUFFIX, který přidá ke všem jménům dat, přebíraným ze SADu předponu resp. příponu oddělenou od původního jména pomlčkou.

I když většina informací byla do SADu začleněna dodatečně a šlo tedy o pouhou evidenci existujících dat, je získání přehledu velkým přínosem pro odstranění zjištěných nesjednotností v popisech dat. Přehled má význam nejen pro současnou práci analytiků a programátorů, ale je i předpokladem a logickým krokem pro připravovaný přechod k bance dat, a to i v případě, že by aplikace byla realizována na jiném počítači a používala jiného jazyka pro popis dat.

6. Záměr na rozšíření funkcí SADu

Realizovaná verze SADu řeší pouze základní funkce adresáře dat. Systém má však předpoklady pro další rozšíření funkcí a aplikace: Jsou to např.:

- pořízení souboru PROGRAMY včetně vyřešení vazeb na soubor SUBSCHEMA, tj. na logické soubory, s nimiž programy pracují;
- kontrola dodržení vazeb mezi soubory při aktualizaci. Např. při zrušení položky ze souboru POLOZKA dostane uživatel zprávu, ve kterých fyzických a logických popisech se položka vyskytovala. Při změně popisu položky bude zajištěna vazba na soubor PROGRAMY. Uživatel dostane zprávu, které programy je nutno znovu přeložit, protože došlo ke změně v adresaci vět;
- generování parametrů pro třídění a výběry. Parametry budou generovány na základě zadání jména logického souboru a jména klíčových položek v pořadí priority klíčů.

Technické řešení SADu bylo voleno tak, aby přechod na jiný počítačový systém byl organizačně i programově nenáročný. Volba programovacího jazyka PASCAL, který vyžaduje přísnou strukturalizaci

programů a přímý přístup do všech souborů SADu, dávají reálné předpoklady k transformaci systému na jiný počítač a operační systém.

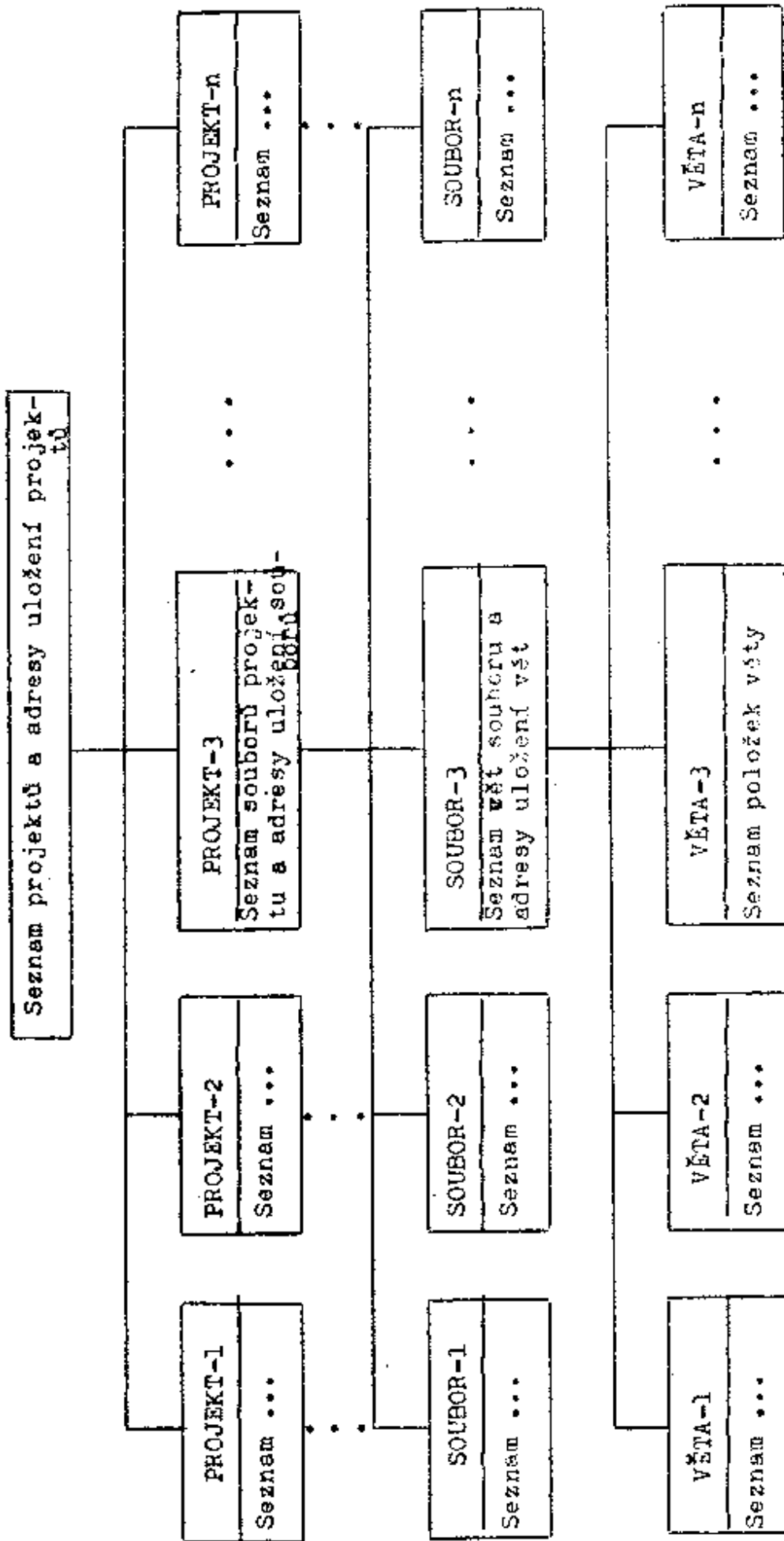
Realizace adresáře dat na počítači ICL 1900/ODRA 1300 je přípravou pro převod programů v jazyce COBOL 1900 na verzi COBOL ICL 2900. Pro převod programů budou pro všechny používané soubory vytvořeny ekvivalentní popisy datových struktur, které budou plně respektovat nové hardwarové a softwarové podmínky. Pomocí těchto ekvivalentních popisů bude provedena transformace FILE SECTION programů pro COBOL ICL 2900.

Článek svým rozsahem nemůže postihnout detailně technické řešení SADu ani všechny možnosti, které jazyk SADu uživateli dává. Má pouze informovat o důvodech vytvoření a aplikace ve VVC ČKD Praha, jeho přínosu a výhledu na rozšíření funkcí.

Literatura

- /1/ Lomax, J.D.: Data Dictionary Systems, NCC Publication 1977
- /2/ Martin, G.N.: Data Dictionary Directory System. Journal of Systems Management, 1973, č.12, str. 12-19
- /3/ Níchthurger, E. - Cibulka, I. - Karásek, V.: Centralizované popisy souborů. Přednáška na semináři ODRA Klubu 1980
- /4/ Interní materiály VVC ČKD Praha - zprávy technického rozvoje, technický popis a technické řešení, r.1980

SCHEMA/SUBSCHEMA



Obr. 1