

# PROJEKČNÁ DATABÁZA A JEJ VYUŽITIE V OBLASTI PROGRAMOVANIA

Ing. Pavol H R I C

INORGA Praha, pobočka K o š i c e

Príspevok sa zaoberá spôsobom tvorby projekčnej databázy pomocou metajazyka. Ukazuje postup tvorby metajazyka pre interaktívny režim práce a použitie metajazyka pri projektovaní. Na jednoduchom príklade ukazuje aplikáciu pre dokumentovanie programových systémov.

## 1. ÚVOD

Rozvoj počítačovej techniky priniesol stále sa zväčšujúcu zložitosť programových systémov. Narastajúce úlohy vedú k tomu, že je nutné racionalizovať tvorbu programových systémov začínajúc projektovaním a končiac vlastnou programovou prácou. Pre oblasť projektovania v súčasnosti existuje pomerne málo vhodných prostriedkov, ktoré by uľahčili formuláciu cieľov a ich uskutočňovanie. Väčšinou ide o také spôsoby, ktoré nie sú vhodné pre počítačové spracovanie.

Pri počítačovom spracovaní je nutné vychádzať z toho, aby formulácia bola jednoznačná-zrozumiteľná - to znamená, že musí ísť o taký prostriedok, ktorý by bol ľahko pochopiteľný pre užívateľov a zároveň by mohol byť spracovaný počítačom.

Základným prostriedkom formulácie a komunikácie je prirodzený jazyk so širokým spektrom vyjadrovacích možností so schopnosťou vyjadriť nejakú skutočnosť rôznymi spôsobmi. Prirodzený jazyk je však často nepresný, alebo viacznačný a preto nie je vhodný pre počítačové spracovanie, ak by sme chceli použiť za základ komunikácie prirodzený jazyk, museli by sme presne špecifikovať tie objekty "o ktorých sa bude hovoriť" a tiež určiť "ivar viet", ktoré by počítač "vedel" prečítať, alebo zapísať.

Z prirodzeného jazyka prostredníctvom deklarácie conceptov je možné utvoriť ucelený systém pojmov problémovo orientovaných, ktoré je možno použiť pre prezentáciu aj komunikáciu znalostí, ktoré sú potrebné k projektovaniu a ktoré zároveň vie spracovať aj počítač.

System pojmov vytvára kompaktný prostriedok pre výstavbu systému reprezentácie znalostí o projekte. Takýto systém pojmov problémovo orientovaný tvorí metajazyk.

Metajazyk realizuje popis konceptuálnej schémy. Táto základná filozofia je v príspevku stručne uvedená pomocou jednoduchého príkladu pre dokumentáciu programov a je programovo podporená systémom SAD, ktorý vznikol v Ústave pre automatizáciu riadenia v priemysle na košickej pobočke INORGY Praha. Systém SAD pracuje v interaktívnom a batchovom režime na počítači EC 1011.

## 2. STRUČNÁ DEFINÍCIA SYSTÉMU:

Pojem systém je východiskovým pre ďalší výklad a preto pristúpime najprv k jeho definícii.

Systém je množina objektov, ktoré sú v určitom vzájomnom vzťahu.

Definícia systému ako množiny objektov, ktoré sú vo vzájomných vzťahoch je na prvý pohľad krátka a všeobecná. V literatúre zvyčajne nachádzame špecificky vymedzenejšie a preto aj dlhšie definície, avšak táto všeobecnejšia definícia je pre ďalší výklad užitočná a postačujúca.

Dôvodom pre túto veľmi stručnú definíciu je konštatovanie, že väčšina čo platí o jednom systéme platí aj o ľubovoľne inom systéme. Skutočnosť, že sa systémy skladajú z častí, ktoré sú vo vzájomnom vzťahu je určujúcim dôvodom mnohých systémových vlastností. Táto krátka definícia kombinuje výhodu stručnosti a jasnosti s výhodou použitia pre mnohé systémy najrozličnejšieho druhu.

Táto definícia systému poukazuje na závislosti správania sa systému od určitých vlastností jeho častí. Ak nejaký objekt nazveme systémom a ten sa skladá z množiny častí s atribútmi, ktorého časti sú vo vzájomných vzťahoch, potom vieme definovať aj "vypočítat" množinu atribútov systému ako celku.

### 3. NIEKTORE ZÁKLADNÉ KRITÉRIA PRI PROJEKTOVANÍ SYSTÉMU

Pre systém je typické, že býva zložitý. Treba vedieť, ako definovať vlastnosti systému ako celku, aby tento slúžil svojim užívateľom. Pri navrhovaní systému je nevyhnutná kooperácia rozličných skupín odborníkov, najmä užívateľov, systémových analytikov, riadiacich pracovníkov a programátorov.

Dokumentácia systému má podávať súčasný stav systému a má spĺňať užívateľské nároky.

Pre návrh zložitých systémov je nevyhnutné počítačové spracovanie návrhu a dokumentácie.

Pri navrhovaní veľkých systémov sa často treba vyrovnávať s problémami, ktoré sú na prvý pohľad jednoduché až triviálne. Napríklad často treba identifikovať množstvo objektov, ktoré patria do nejakej množiny. Na základe týchto požiadaviek vzniká potreba hľadania vhodných pracovných nástrojov pre systémovú prácu. Ďalej sa v stručnosti pokúsime podať pracovný nástroj, ktorý rešpektuje naznačené požiadavky.

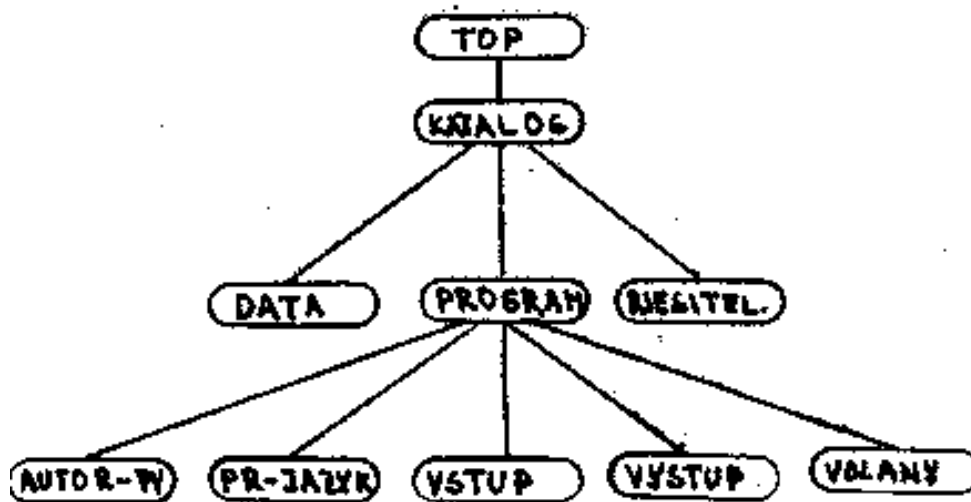
### 4. FORMÁLNY POPIS SYSTÉMU:

Definícia formálneho objektu /CONCEPT-U/.

Formálny objekt zastupuje množinu reálnych objektov rovnakého typu. Formálne skúmanie určitého systému sa nezaobera popisom reálnych objektov systému, ale určitými formálnymi objektami, ktoré zastupujú reálne objekty systému.

Popis štruktúry systému sa uskutoční pomocou formálnych objektov. Tým sa podstatne zjednoduší kontrola konzistentnosti navrhovaného systému. Systém formálnych objektov nemusíme skúmať z hľadiska množstva objektov, ale len z hľadiska štruktúry formálnych objektov.

Obr.č.1 /Štruktúra z príkladu - pozri nižšie/



### Atribúty formálneho objektu.

Formálny objekt môže byť definovaný pomocou iných, už definovaných formálnych objektov. Hovoríme, že formálny objekt má identifikačné /odvodené/ atribúty, alebo formálny objekt môže mať špecifické atribúty, hovoríme, že formálny objekt má atomické atribúty.

PROGRAM CONCEPT /NÁZOV:TEXT, TYPP:PROGTYP/;

## 5. INTERAKTÍVNA TVORBA METAJAZYKA

### 5.1 Tvorba konceptov a ich atribútov

Pre tvorbu konceptov platia vzťahy:

MENO-1	CONCEPT	'NÁZOV CONCEPTU.....';	/1/
MENOATR	ATOM	ATRIBÚT;	/2/

MENO-1 je identifikátor vytvoreného konceptu, predstavuje množinu určitého typu.

NÁZOV CONCEPTU- je úplne vyjadrenie jeho názvu.

ATRIBÚT CONCEPTU MENO-1./Môže byť identifikačný, atomický/

MENOATR je priradenie mena atribútu, ktorý je rovnakého typu ako je atribút.

Ak je atribút CONCEPTU neodvodený, ale textový /specifický/ stanovuje jeho dĺžku pomocou vzťahu /3/.

TXT20            ATOMTYPE            '20;            /3/

Kde TXT20 je identifikátor atomického typu, ktorý môže byť používaný pre definovanie textových atribútov ktoréhokolvek conceptu.

5.2. Zaradenie CONCEPTU do štruktúry pomocou formátu, podľa nasledujúcich vzťahov:

MENO-1	CONCEPT	'Názov CONCEPTU...'	/1/
FORM-1	FORMAT	POD MENO-I ;	/4/
	KEYW	TEXT	/5/
	ITEM	MENOATR	/6/

---

MENO-1	Meno CONCEPTU, ktorý chceme pomocou formátu zaradiť do štruktúry
FORM1	Identifikátor formátu
MENO-I	Meno ľubovoľného conceptu pod ktorý chceme zaradiť definovaný concept. V prípade sieťovej štruktúry namiesto conceptu tu môže byť meno skupiny conceptov.
TEXT	Je ľubovoľný text, ktorý robí formát zrozumiteľný pre užívateľa
MENOATR	Meno atributu vytvorené pre daný concept podľa /2/.

5.3. Skupina CONCEPTOV.

Skupinu CONCEPTOV môžeme vytvoriť nasledovne:

SET1	SET	Definícia SETU ;	/7/
MEMB1	INCLUDE	MENO-2;	/8/

Vzťahy /1/ až /8/ môžeme realizovať v interaktívnom režime pomocou SĀDU pre ľubovoľný meta-jazyk.

Takýmto spôsobom bol vytvorený meta-jazyk pre dokumentáciu programov. Schéma Je zobrazená na obr.č.1

## 6. Príklad použitia .

Pre lepšiu ilustráciu tvorby metajazyka, jeho použitie pre interaktívne projektovanie a niektorých dotazov na projekčnú databázu sme zvolili príklad pre dokumentáciu programov, na ktorom si v krátkosti ukážeme jednotlivé postupy, ktoré sú doplnené aj v prílohách.

### A. Definícia atomických typov atribútov.

TYXT10	ATOMTYPE	'10';
TYXT20	ATOMTYPE	'20';
TYXT30	ATOMTYPE	'30';
TYXT40	ATOMTYPE	'40';
TYXT60	ATOMTYPE	'60';

### B. Definícia CONCEPTOV a ich ATRIBÚTOV.

KATALÓG	CONCEPT	'Dokumentačný katalóg' /Názov -K:TYXT40/;
DATA	CONCEPT	Popis dat /popis-D:TYXT50, typ:DATATYP/;
RIEŠITEĽ	CONCEPT	'Riešiteľ' /Meno:TYXT20/;
PROGRAM	CONCEPT	Program /Názov-T:TYXT40, typ:PROGTYP/;
AUTOR-PV	CONCEPT	'Autor programovej verzie' /Autor:Riešiteľ,Verzia/;
PR-JAZYK	CONCEPT	'Program napísaný v jazyku...' /Jazyk:P-Jazyk,Program/;
VSTUP	CONCEPT	'vstup do programu' / ID-VSTUP: Data, Medium: TYXT10
VÝSTUP	CONCEPT	VÝSTUP 7 Programu / ID-vypt: Data, Medium V: TYXT10
VOLANÝ	CONCEPT	'Volanie programu' /M-PROG: PROGRAM/;
DATATYP	CONCEPT	'Datový typ' /Typ-D:TYXT40/;
PROGTYP	CONCEPT	'Programový typ' /Typ-P:TYXT40/;
P-JAZYKY	CONCEPT	'Programovací jazyk' /Druh:TYXT20/;
POZNÁMKA	CONCEPT	Poznámka /Riadok: TYXT60/

### C. Vytvorenie štruktúry conceptov pomocou forzatov.

Zápis formátov pre definovanie concepty bol utvorený podľa vzťahov /4/ a /6/.

Uvádza sa tvary týchto formátov v prílohe č.1.

Odlepený metajazyk slúži pre popis konkrétnych objektov programovej dokumentácie SAD-u.

V našom prípade sme popis dosiahli automatizovane zo zdrojových tvarových programov. Syntaktickou analýzou boli naplnené formáty, ktoré sú vstupom do projekčnej databázy.

V prílohe 2,3 uvádzame jednoduché dotazy na projekčnú databázu.

Dotazy na obsah katalogov.

KATALF - KATALOG PROGRAMOV

KATALR - KATALOG RIEŠITEĽOV

KATALD - KATALOG DAT

Dotaz na popis programu dialog v katalógu programov.

Dotaz na programy, ktoré sú volané z programu dialog. Tieto dotazy sú ďalej spracovávané do grafických výstupov, ktoré v tomto rozsahu nie je možné uviesť.

### 7. Záver

Jednoduchou ukážkou sme chceli predviesť spôsob použitia metajazyka pre tvorbu projekčnej databázy pre dokumentáciu už existujúcich prog. Prezentovaný prístup poskytuje široké možnosti aplikácie nie len v OB automatizovanej tvorby dokumentácie, ale predovšetkým v oblasti projektovania ASR. System SAD je súčasťou projekčnej technológie SDT-1, ktorá má všetky etapy návrhu a tvorby ASR. Doterajšie výsledky poukazujú na použitie v oblasti návrhu algoritmov programov, logickej schémy <sup>dat</sup>, sledovanie terminovaných činností a všade tam, kde sa vyžaduje interaktívne automatizované spracovanie znalostí.

### 8. Literatúra

- /1/ ISDOS: Technical memorandum 208 Overview of PSL/PSA, Version A5.1, University of Michigan, 1980
- /2/ SDLA - System Descriptor and Logical Analyzer, General Description, MTA SZTAKI, Budapest 1980
- /3/ SDT-1: System Design Technology, Výsk.zpráva INORGA, 1982
- /4/ SAD: INORGA, Interná dokumentácia, 1983

DISPLAY FORMATS AVAILABLE IN DESIGN PHASE

SAD/DOCU01  
META-DB: SADDK

UNOER	CONCEPT	FORM-ID	SEQ.NO	FORMATS:	10	20	30	40	50	60	70
I TOP	I KATALOG	I KATG	I 1	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I	I	I	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I KATALOG	I PROGRAM	I PRG	I 1	I PROGRAM	I JAZYK	I .....	I .....	I .....	I .....	I .....	I TYP
I	I	I	I	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I TYP
I	I DATA	I DAT	I 1	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I	I	I	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I RIESITEL	I RITS	I 1	I RIESITEL	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I	I	I	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I PROGRAM	I AUTOR	I AUTOP	I 1	I AUTOR	I VERZIA	I .....	I .....	I .....	I .....	I .....	I .....
I	I	I	I	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I P-JAZYK	I PRJAZ	I 1	I PISANY V JAZYKU	I JAZYK	I .....	I .....	I .....	I .....	I .....	I POCT RIADKOV
I	I	I	I	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I POCT
I	I VSTUP	I VST	I 1	I VSTUP	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I	I	I	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I VYSTUP	I VYST	I 1	I VYSTUP	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I	I	I	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I VOLANY	I VOLA	I 1	I VOLANY Z	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I	I	I	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I ALL	I DATATYP	I TYPD	I 1	I TYP-DAT	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I	I	I	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I JAZYKY	I JAZ	I 1	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I	I	I	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I POZVANKA	I **	I 1	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I	I	I	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I POCET	I TYP	I 1	I TYP-PROGRAMU	I .....	I .....	I .....	I .....	I .....	I .....	I .....
I	I	I	I	I .....	I .....	I .....	I .....	I .....	I .....	I .....	I .....



```
*****
SAD/SHOW : KATALP      DEPTH=1      PROJECT : SADOXP
*****
KATALP   : == '===== KATALOG PROGRAMOV ====='
DIALOG   : . PROGRAM 'DIALOG' TYP PROG
AOECLA   : . PROGRAM 'VYMAZANIE RIADKU POOLA IDENTIFIKATORA' TYP P-PROG
BAT       : . PROGRAM 'DAVKOVY VSTUP DO DB' TYP PROG
WRITE    : . PROGRAM 'ZAPIS VETY DO DATABAZY' TYP P-PROG
READ     : . PROGRAM 'CITANIE VETY Z DATABAZY' TYP P-PROG
ZEI:DB   : . PROGRAM 'INICIALIZACIA DATABAZY PROJEKTU' TYP PROG
MAINTM   : . PROGRAM 'UDRZBA DATABAZY' TYP PROG
SHOW     : . PROGRAM 'SAMOSTATNA VERZIA DOTAZU SHOW' TYP P-PROG
TRACE    : . PROGRAM 'SAMOSTATNA VERZIA DOTAZU TRACE' TYP P-PROG
FASTC    : . PROGRAM 'PRESUN CONCEPTOV DO DB PROJ' TYP PROG
*****
SAD/SHOW : KATALP      END          PROJECT : SADOXP
*****
```

```
*****
SAD/SHOW : KATALR      DEPTH=30     PROJECT : SADOXP
*****
KATALR   : == '===== RIESITELIA ====='
AZOR     : . RIESITEL 'ING.AZOR SERGEJ'
RODON    : . RIESITEL 'ING.RODON GEJZA'
BIAL     : . RIESITEL 'ING.BIAL STEPAN'
HANAK    : . RIESITEL 'ING.HANAK LUBOR'
HRIC     : . RIESITEL 'ING.HRIC PAVOL'
JAROS    : . RIESITEL 'ING.JAROS PETER'
TOMKO    : . RIESITEL 'ING.TOMKO STANISLAV'
ZEMIAN   : . RIESITEL 'ZEMIANSKY JAK'
SUCHKO   : . RIESITEL 'ING.SUCHKOULSKA TATIANA'
KURINEC  : . RIESITEL 'KURINEC STEFAN, PROM.MAT.'
*****
SAD/SHOW : KATALR      END          PROJECT : SADOXP
*****
```

```
*****
SAD/SHOW : KATALD      DEPTH=30     PROJECT : SADOXP
*****
KATALD   : == '===== KATALOG DAT ====='
RECP     : . DATA 'ROLE VETY' TYP COMMON
IPAR     : . DATA 'ROLE PARAMETROV' TYP COMMON
EDIT     : . DATA 'CITATELNY OBSAH MASTER INDEXU' TYP DISPLAY
INDEX    : . DATA 'INICIALIZOVANY MASTER A INDEX FILE' TYP SUBOR
EDI      : . DATA 'VYSTUPNY TEXTOVY SUBOR' TYP SUBOR
DIAL     : . DATA 'DIALOG S DISPLAYOM' TYP DISPLAY
ENRO     : . DATA 'SUBOR OZNAMOV O CHYSACH' TYP SUBOR
INDX     : . DATA 'INDEX FILE' TYP SUBOR
INDY     : . DATA 'INDEX FILE' TYP SUBOR
MASTIN   : . DATA 'MASTER INDEX' TYP COMMON
TEXT     : . DATA 'VYSTUPNY TEXTOVY SUBOR' TYP SUBOR
DBFI     : . DATA 'SUBOR DATABAZY' TYP SUBOR
*****
SAD/SHOW : KATALD      END          PROJECT : SADOXP
*****
```

PRÍLOHA C.3

\* DATE:18/ 3/83 \* TIME:14: 41 3 \* - 185 -

```
*****
SAD/SHOW : DIALOG      DEPTH=30      PROJECT :
*****
DIALOG : PROGRAM 'DIALOG' TYP PROG
: . AUTOR AZOR VERZIA '2'
: . PISANY V JAZYKU FORT POCET RIADKOV '1000'
: . VSTUP ERRO MEDIUM 'DISK'
: . VSTUP DBFS MEDIUM 'DISK'
: . VSTUP DBFS MEDIUM 'DISK'
: . VSTUP INDY MEDIUM 'DISK'
: . VSTUP INDY MEDIUM 'DISK'
: . VSTUP MASTIN MEDIUM 'COMMON'
: . VSTUP MASTIO MEDIUM 'COMMON'
: . VSTUP RECD MEDIUM 'COMMON'
: . VSTUP INTFOR MEDIUM 'COMMON'
: . VSTUP IPAR MEDIUM 'COMMON'
: . VYSTUP DBFS MEDIUM 'DISK'
: . VYSTUP INOX MEDIUM 'DISK'
: . VYSTUP END MEDIUM 'DISPLAY'
*****
```

SAD/SHOW : DIALOG END PROJECT : SADDKP

```
*****
SAD/TRACE : DIALOG      PROJECT :
*****
```

```
DIALOG : PROGRAM 'DIALOG' TYP PROG
----- D E C L A R E D ----- 2 -----
> VOLANY Z DIALOG
--> 3 --> ADELA
> VOLANY Z DIALOG
--> 3 --> ERMES
> VOLANY Z DIALOG
--> 3 --> PAX
> VOLANY Z DIALOG
--> 3 --> HEAUT
> VOLANY Z DIALOG
--> 3 --> DECODE
> VOLANY Z DIALOG
--> 3 --> ENCODE
> VOLANY Z DIALOG
--> 3 --> DAJFOR
> VOLANY Z DIALOG
--> 3 --> ACHECK
> VOLANY Z DIALOG
--> 3 --> WRITER
> VOLANY Z DIALOG
--> 3 --> WRITE
> VOLANY Z DIALOG
--> 3 --> READ
> VOLANY Z DIALOG
--> 3 --> READR
> VOLANY Z DIALOG
--> 3 --> TRABRA
--> 3 --> AISHF
*****
```

SAD/TRACE : DIALOG END PROJECT : SADDKP

EGF