

Ing. Erhard Tschernoster.

## Úvod

Problém který musí řešit všichni tvůrci aplikačního software je návrh a implementace uživatelského rozhraní. Za uživatelské rozhraní budeme v našem případě považovat výrazové prostředky, které umožní uživateli popsat úlohu, kterou chce provádět a data, která chce zpracovávat. V současné době jsou nejrozšířenější dva typy uživatelského rozhraní:

- a/ menu
- b/ příkazový jazyk.

Každý z uvedených typů rozhraní má své přednosti i nedostatky. Každý z uvedených typů rozhraní má své přívržence i odpůrce. Není účelem tohoto článku porovnávat oba přístupy, zaměříme se pouze na příkazové jazyky programových systémů pro zpracování data v oblasti matematické statistiky.

• Typický program počítající úlohu v oboru matematické statistiky pracuje následujícím způsobem:

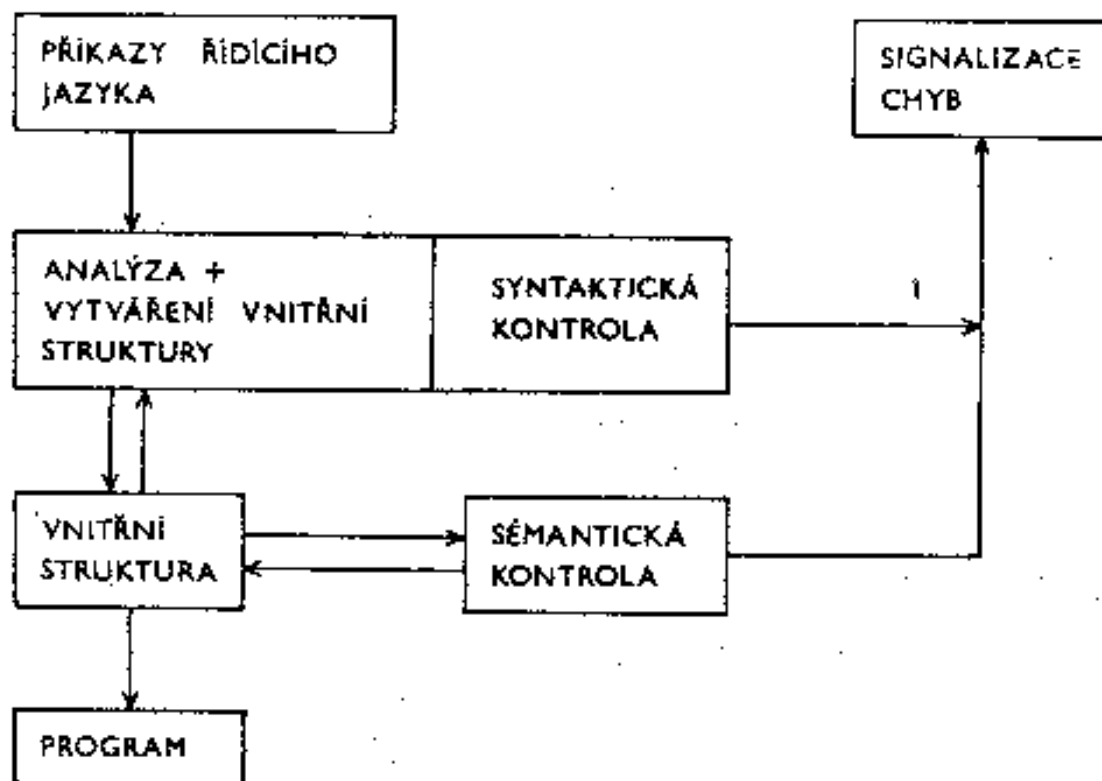
- a/ přečte datovou matici, která bývá obvykle organizována tak, že jednotlivé sloupce odpovídají proměnným, řádky pak analyzovaným objektům,
- b/ vybere z datové matice požadované proměnné a v případě potřeby je příslušným způsobem transformuje,
- c/ provede požadovaný výpočet parametrizovaný uživatelem,
- d/ vytiskne výsledky ve tvaru, který požaduje uživatel.

Úkolem řídicího jazyka je dodat spouštěnému programu všechny

informace, které potřebuje k práci podle uvedeného schématu.

Z toho, co již bylo řečeno vyplývá, že každý systém programů popisovaného typu musí obsahovat mechanismus čtení a analýzy řídicích informací.

Obecné schéma funkce mechanismu čtení řídicího jazyka je uvedeno na obr. 1.



obr. 1.

Uživatel zformuluje své požadavky na systém ve formě posloupnosti příkazů řídicího jazyka. Tato posloupnost je postupně zpracovávána syntaktickým analyzátozem a současně další komponenty systému vytvářejí vnitřní datovou strukturu, do které se ukládají veškeré informace obsažené v řídicím jazyce. V případě, že syntaktický analyzátor objeví závažné chyby, vydá chybové hlášení a činnost celého systému je po přečtení všech příkazů automaticky ukončena. Další části systému, které na čtení a analýzu řídicího jazyka navazují, používají ke své

práci pouze vnitřní datovou strukturu. Před spuštěním vlastního výpočtu je obvykle prováděna schématická kontrola informací obsažených ve vytvořené vnitřní struktuře. Sémantická kontrola je silně závislá na daném programu a v rámci obecného mechanismu zpracování řídicího jazyka je možno realizovat pouze její nepatrnou část.

V současné době existuje na trhu řada systémů pro statistickou analýzu dat, které mají uživatelské rozhraní realizované pomocí řídicího jazyka. Mezi nejvýznamnější patří systémy BMDP, SPSS a SAS, které jsou běžně dostupné i u nás. Všechny uvedené systémy mají společné to, že byly původně implementovány na velkých střediskových počítačích, i když dnes již existují jejich verze na osobní počítače, jako např. STATGRAPHICS, mají uživatelské rozhraní realizované většinou ve formě menu.

#### Řídicí jazyk systému GUHA

V našem institutu jsme ve spolupráci s MSBÚ - ČSAV před časem řešili problém vytvoření obecných prostředků pro práci s řídicím jazykem, které by bylo možno použít při implementaci nové verze programů systému GUHA. Navržený řídicí jazyk jsme nazvali GCL (GUHA Control Language) a jeho struktura je silně inspirována strukturou řídicího jazyka systému BMDP.

Zápis v jazyce GCL se skládá z paragrafů, které jsou dále děleny na položky. Každý paragraf začíná jednoznačně přiřazeným jménem, které je ukončeno znakem ":". Informace potřebné pro jedno spuštění požadovaného programu tvoří t.zv. zadání. Zadání je posloupnost paragrafů začínající paragrafem PROBLEM a končící paragrafem END.

Příklad zadání je uveden na obr. 2.

```

PROBLEM: TITLE='GURA SAMPLE';
INPUT: SOURCE=CARDS; NVAR=4; NCASE=100;
      FORMAT='(F2.0,3F3.0)';
VARIABLES: NAMES=SEX,VEK,VAHA,VYSKA;
      MISSING=4*-1; MIN=(2)15,35,155;
      MAX=(2)99,125,210;
      CODE=(1)<1,2>;
      CUTPOINT=(2)<25,45,65>,<70,80,90>;
      . . . . .
END:

```

obr. 2.

Členění informací do paragrafů je zcela libovolné a jak vyplývá z dalšího, je závislé pouze na autorovi programu, který služeb GČL používá. Vzhledem k všeobecně uznávanému požadavku strukturovanosti je však vhodné, aby informace popsané jedním paragrafem byly spojeny logickou vazbou.

Položky ze kterých se jednotlivé paragrafy skládají mají tento obecný tvar:

<klíč. slovo> = <hodnota> ;

Klíčové slovo označující položku je identifikátor, jednoznačný v rámci jednoho paragrafu, jehož maximální délka může být 8 znaků. Hodnotou položky může být:

- jednoduchá proměnná, t.j. číslo typu integer nebo real, index (kladné celé číslo) nebo řetězec,

- jméno, t.j. slovo z předem definovaného seznamu přiřazeného danému klíčovému slovu,
- seznam jednoduchých proměnných nebo jmen,
- seznam seznamů.

Položka se může v rámci paragrafu vyskytnout pouze jednou, na pořadí položek v paragrafu nezáleží. Pro každou položku je možno předem definovat implicitní hodnotu.(default).

Implementace mechanismu zajišťujícího čtení řídicího jazyka a vytvoření vnitřní struktury je znázorněno na obr. 3. Autor programu, který chce GCL použít, musí nejdříve přesně definovat strukturu kterou chce použít, tj. určit počet paragrafů, jejich jména, počet položek v jednotlivých paragrafech, typy hodnot položek, implicitní hodnoty, atd. Tato definice je zpracována programem GCLCMP, který na základě vstupující definice vygeneruje proceduru v jazyce FORTRAN. Vygenerovaná procedura, jejíž jméno je volitelné, slouží jako procedura inicializující proces čtení GCL. Programátor aktivuje celý proces zavoláním automaticky vytvořené procedury, která spustí obecný mechanismus čtení GCL a předá mu veškeré informace o konkrétním tvaru požadované struktury. Jednotlivé procedury čtecího mechanismu provádějí během čtení syntaktickou analýzu vstupujícího jazyka a vytvářejí vnitřní strukturu GCL. V případě, že během čtení byly objeveny závažné chyby, proces čtení svoji práci automaticky ukončí. V opačném případě systém předá volajícímu programu jako výsledek své práce odkaz na vnitřní strukturu, kterou vytvořil.

Vnitřní struktura GCL je stromová struktura zobrazena na obr. 4. Kořenem stromu je hlavní ukazatel, který odkazuje na seznam paragrafů, který obsahuje tolik elementů, kolik je v zadání definováno paragrafů. Obsahem každého elementu seznamu paragrafů je další ukazatel, který odkazuje na pole jednotlivých paragrafů. Pole paragrafu obsahuje tolik elementů, kolik položek je v rámci paragrafu definováno. Jednotlivé položky pole paragrafu obsahují hodnoty příslušných položek

v případě, že hodnota je zobrazitelná na čtyřech bytech, nebo odkaz na další seznamovou strukturu. V uvedené struktuře je možno zobrazit libovolné informace, které je možno pomocí GCL popsat.

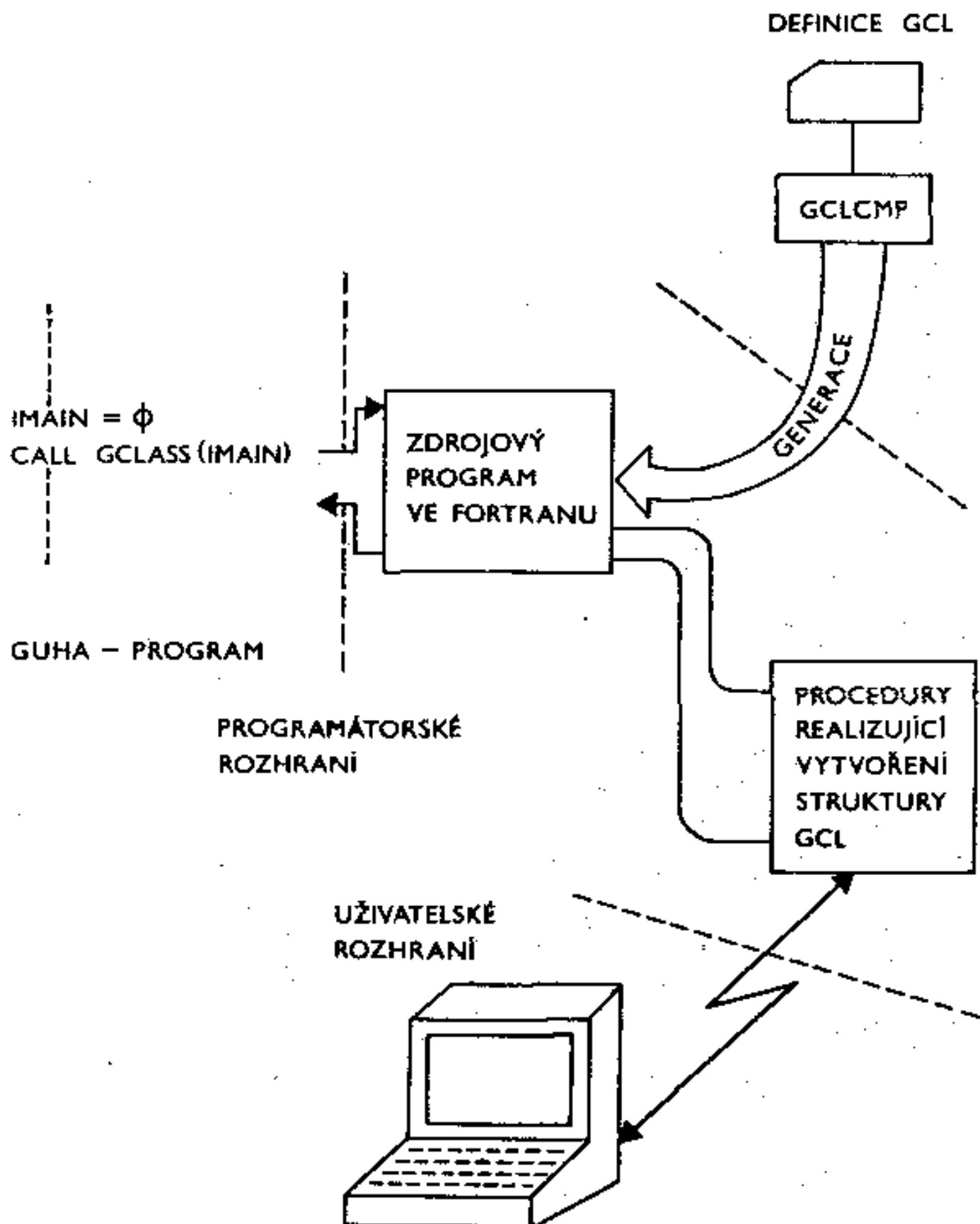
Součástí softwarového vybavení GCL jsou pochopitelně i procedury, které programátorovi umožňují přístup do vnitřní datové struktury na její libovolnou úroveň. Přístupové procedury předávají buď hodnotu položky paragrafu, případně ukazatel, který může sloužit k přímému přístupu k požadovaným informacím v případě seznamových struktur.

## Závěr

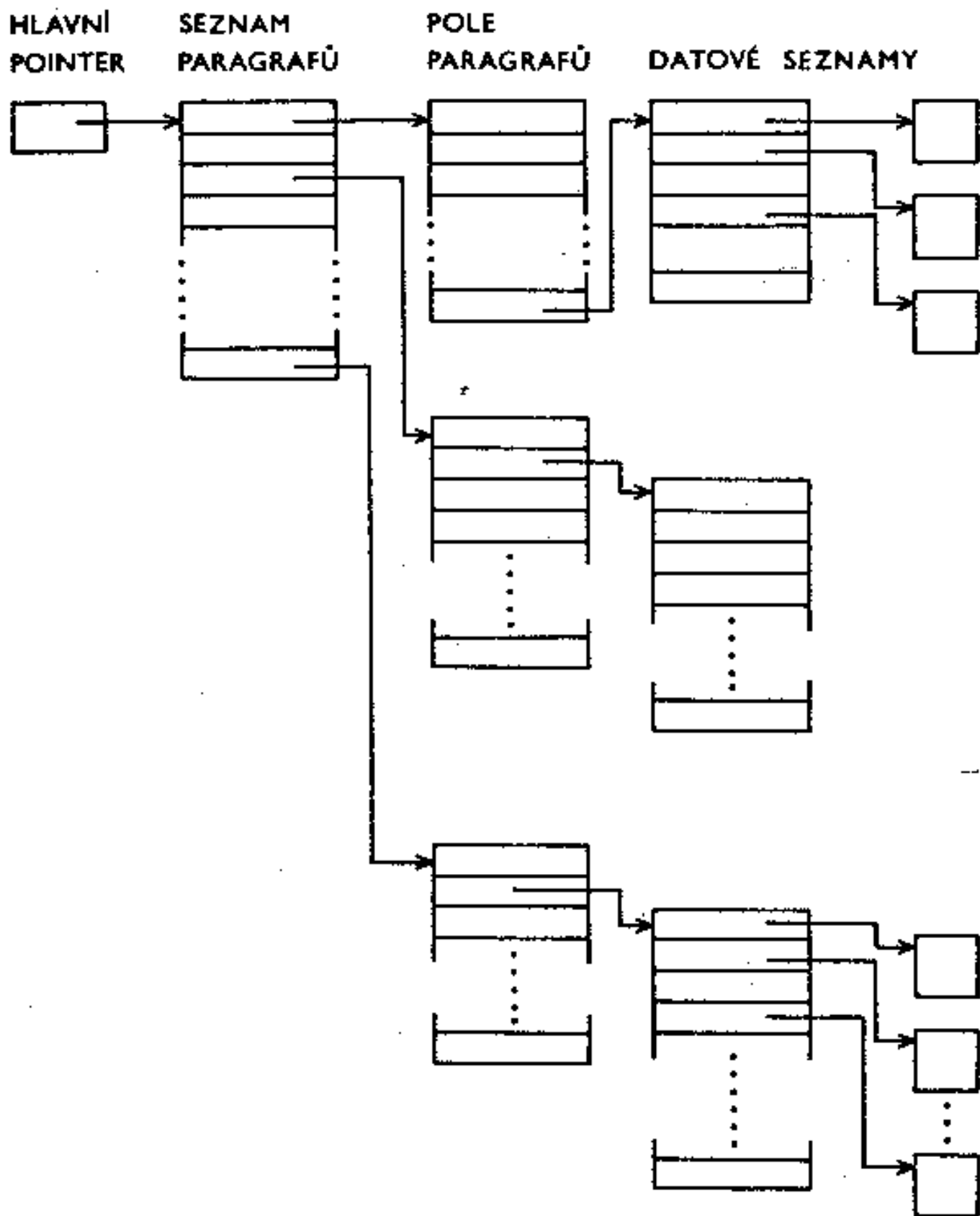
Realizací GCL se nám podařilo vyřešit problém čtení řídicích informací pro aplikační programy na poměrně obecné úrovni. Uvedené prostředky, které jsou realizované pro použití v prostředí programovacího jazyka FORTRAN, jsme použili při implementaci programů systému GUHA. Tím se nám podařilo vytvořit jednatelná rozhraní jak z pohledu uživatele, tak i z pohledu tvůrců systému.

## Literatura

- Dixon W.J. et al.: BMDP Statistical Software 1981, Berkeley: University of California Press.
- Gruenberg U.: Einheitliche Syntax von Kommandosprachen und Probleme der Realisierung in Programmsystemen, Statistical Software Newsletter B5, 1983.
- Havránek T., Tachernoster E.: GUHA - its interpretation and computer realization, (v tisku).
- Nie N.H.: SPSS, 2nd edition. McGraw-Hill, 1975.
- Payer M.: Syntaxgesteuerte Interpretation von Sprachen, Statistical Software Newsletter B5, 1983.



obr. 3.



obr. 4.