

NÁVRH STRUKTURY KONVERZAČNÍHO PROGRAMOVÉHO SYSTÉMU PRO VTV

RNDr. Petr Hanzálek, RNDr. Jiří Hřebíček, CSc.

Ústav fyzikální metalurgie ČSAV Brno

Abstrakt: V příspěvku bude popsána struktura konverzačního programového systému pro VTV, implementovatelného jak na velkém počítači, tak i na minipočítači s omezenou kapacitou programově adresovatelné paměti a dále nezbytně nutné obslužné programové prostředky a datové soubory potřebné pro zabezpečení práce programového systému.

1. Úvod

Na pracovištích ČSAV se v současné době používají dva programové soubory pro řešení optimalizačních úloh. Jedná se o programové soubory OPTIPACK [3] a SPONA [4].

Oba soubory řeší optimalizační úlohy různých typů, přitom jejich použitelnost se částečně překrývá. Jsou koncipovány pro práci na středním až velkém počítači v režimu dávkového zpracování, v OPTIPACKu je v omezené míře umožněna interaktivní práce a to při zadávání vstupních dat. Podprogramy obou souborů jsou napsány v programovacím jazyku FORTRAN IV nebo IBM FORTRAN a implementovány na počítači ICL 2950/10 pod operačním systémem VME 2900 a na počítačích řady JSEP pod operačními systémy DOS a OS.

Programový soubor OPTIPACK tvoří cca 280 optimalizačních a servisních podprogramů s celkovým počtem zhruba 30 000 FORTRANských příkazů. Uživatel musí při řešení optimalizační úlohy naprogramovat uživatelské podprogramy předepsaného tvaru, popisující matematický model řešené úlohy a napsat řídicí program, v němž stanoví strategii výpočtu, tzn. pořadí volání knihovnických podprogramů realizujících optimalizační algoritmy. Soubor OPTIPACK pracuje důsledně s dynamickými mezemi polí, takže rozsah řešené úlohy není shora omezen - je ohraničen pou-

ze dostupnou velikostí operační paměti.

Programový soubor SPONA tvoří cca 120 optimalizačních podprogramů s celkovým počtem zhruba 45 000 FORTRANských příkazů. Uživatel musí opět specifikovat podprogramy pro popis modelu úlohy. Strategie výpočtu se zadává pomocí jednoduchého řídicího jazyka. Uživatel nemusí znát konkrétní formu knihovnic optimalizačních podprogramů, protože řídicí program je generován překladačem řídicího jazyka. Nevýhodou SPONY je pevná délka pracovních polí uložených v COMMON blocích a odtud vyplývající omezení na maximální velikost řešených úloh.

2. Programový systém UFO

Na základě analýzy programových souborů OPTIPACK a SPONA dospěli jejich autoři k názoru, že je nutno vytvořit nový programový soubor pro řešení optimalizačních úloh odpovídající současnému stavu rozvoje výpočetní techniky. Tento soubor bude zahrnovat progresivní prvky obou výše zmíněných souborů, tzn. bude obsahovat ověřené efektivní algoritmy pro řešení většiny optimalizačních úloh, nebude mít omezení na velikost úloh, bude snadno modifikovatelný pro případ řešení nestandardních typů úloh. Bude vybaven řídicím jazykem pro popis strategie výpočtu a kompilátorem řídicího jazyka. Dále bude mít dostatečný uživatelský komfort a bude snadno přenositelný na jiné počítače vybavené kompilátorem jazyka "FORTRAN".

Tento nový navrhovaný soubor optimalizačních programů dostal pracovní název UFO (Univerzální Funkcionální Optimalizace) [5].

Soubor UFO nebude vytvářen jako problémově orientovaná knihovna optimalizačních podprogramů, ale jako uživatelsky orientovaný konverzační programový systém. Tento systém by měl uživateli v co největší míře usnadňovat práci při řešení optimalizačních úloh a napomáhat mu při:

- volbě strategie řešení optimalizační úlohy
- vytváření uživatelských podprogramů popisujících matematický model fyzikální, či technické úlohy

- volitelném přerušování zpracování optimalizační úlohy s pozdějším pokračováním
- provedení vytvořeného optimalizačního programu buď v dávkovém režimu, nebo v režimu dovolujícím interaktivní zásah do optimalizačního procesu
- poskytování informací o implementovaných algoritmech a doporučených metodách řešení úloh.

Z předpokládaného rozvoje výpočetní techniky vyplynul další důležitý požadavek na programový systém UFO. Systém musí být provozuschopný nejen na velkých počítačích (ICL, SIEMENS, JSEP), ale i na minipočítačích řady SMP (SM, ADT). Odtud vyplývají další požadavky na formální strukturu knihovnic podprogramů a na tvar výsledného optimalizačního programu.

Programový systém UFO má tři části:

- a) Programové jádro UFO, které je tvořeno knihovnou obsahující všechny optimalizační, vstupní, výstupní a pomocné podprogramy, pomocí nichž uživatel ve spolupráci se systémem vytváří konverzačním způsobem optimalizační program.
- b) Servisní vrstva UFO, v rámci které je organizována prakticky veškerá komunikace uživatele se systémem UFO. Součástí servisní vrstvy jsou programové prostředky pro vytváření souborů vstupních dat, pro vytváření uživatelských podprogramů, pro tvorbu výsledných optimalizačních programů. Servisní vrstva rovněž vytváří, používá, případně ruší komunikační datové soubory.
- c) Datové soubory, což jsou jednak knihovní soubory obsahující zdrojové texty i relokatibilní tvar podprogramů systému, dále různé informační soubory, soubor chybových hlášení, soubor se vzory tisků a dočasné soubory sloužící ke komunikaci mezi jednotlivými částmi systému.

3. Programové jádro systému UFO

Systém UFO je problémově orientovaný aplikační programový systém, určený pro řešení určité třídy optimalizačních úloh.

Z formálního hlediska je charakterizován tím, že uživatel musí sám při vytváření optimalizačního programu vytvořit uživatelské podprogramy charakterizující typ řešené úlohy z následujícího menu:

- úloha nelineárního matematického programování s konečným i nekonečným počtem omezujících podmínek,
- diskrétní a spojitá aproximace,
- diskrétní a spojitá optimální řízení,
- celočíselné, racionální a geometrické programování,
- vícekritériální optimalizace,
- globální optimalizace,
- řešení soustav nelineárních funkcionálních rovnic,
- řešení soustav nelineárních diferenciálních rovnic.

Algoritmy pro řešení specifikovaných úloh jsou součástí knihovny podprogramů IPO. Podprogramy knihovny se z funkčního hlediska dělí do následujících skupin:

- optimalizační podprogramy realizující optimalizační algoritmy
- vstupní podprogramy
- výstupní podprogramy
- podprogramy pro konverzaci s uživatelem
- podprogramy pro různé servisní činnosti
- testovací podprogramy pro ověřování algoritmů, uživatelských podprogramů apod.
- podprogramy pro realizaci programových prostředků servisní vrstvy, např. generátoru programů, generátoru vstupních dat apod.

Všechny podprogramy jádra budou používat normovaného jazyka "FORTRAN", přičemž se nesmí používat rozšíření jazyka přípustného v konkrétních verzích kompilátoru. Dále všechny vstupní a výstupní operace budou soustředěny do vstupních a výstupních podprogramů a případné (výjimečné) použití strojově závislých činností se realizuje pomocí podprogramů ve strojově závislém jazyce, které budou pro různé implementace různé.

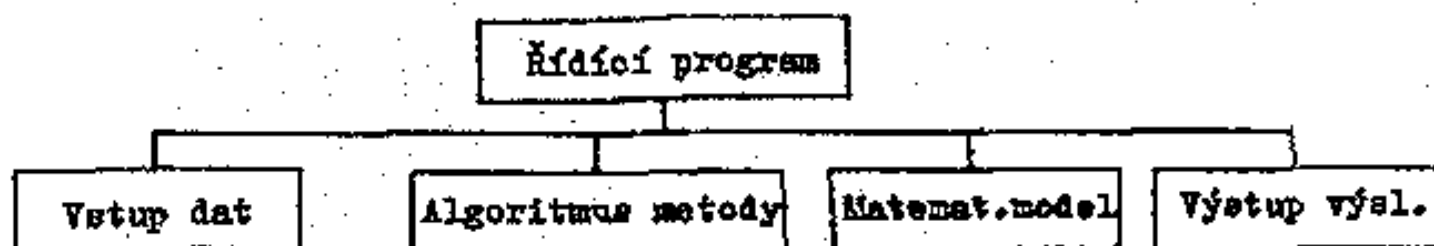
Modifikovatelnost systému bude umožněna následujícími konvencemi:

- komunikace mezi programovými jednotkami bude prováděna především prostřednictvím parametrů podprogramů. Pouze některé řídicí veličiny (jednoduché proměnné) budou ukládány do COMMON bloků.
- přenos řízení mezi metodou a modelem, tj. mezi systémovými podprogramy pro optimalizační algoritmus a uživatelskými podprogramy pro matematický model, se provede pomocí stavových proměnných. Podprogram pro metodu nebude vyvolávat přímo uživatelský model, ale přenos řízení proběhne následovně:
 1. Jakmile podprogram pro metodu zjistí požadavek na provedení uživatelského modelu (např. výpočet funkční hodnoty) nastaví stavovou proměnnou specifikující druh požadované informace a stavovou proměnnou pro návrat zpět a předá řízení do hlavního programu.
 2. Hlavní program vyvolá uživatelský model.
 3. Podprogram pro uživatelský model vypočítá požadovanou informaci a předá řízení zpět do hlavního programu.
 4. Hlavní program vyvolá znovu podprogram pro metodu.
 5. Podprogram pro metodu pokračuje ve výpočtu v závislosti na hodnotě návratové proměnné.

Takto navržený přenos řízení mezi metodou a modelem umožňuje poměrně snadno modifikovat knihovní podprogramy při řešení netypické optimalizační úlohy. Dalším přínosem této značné nezávislosti metody a modelu je skutečnost, že tyto dva úseky programu mohou být organizovány jako překrývající se segmenty, což je důležité při implementaci na minipočítačích s poměrně malou programově adresovatelnou pamětí.

Obdobným způsobem bude organizován i vstup dat do optimalizačního programu a výstup výsledků.

Optimalizační program lze znázornit následujícím schématem:



Obrázek 1: Struktura optimalizačního programu.

Pomocí výše uvedeného způsobu lze v řídicím programu jednoduše organizovat alternativní volání několika metod nebo požadovat provedení různých druhů výstupu a/nebo dodatečného zpracování, aniž se tím skomplikuje struktura řídicího programu. To je důležité pro konstrukci překladače vstupního řídicího jazyka a rovněž i pro vytváření popisu segmentační struktury optimalizačního programu.

4. Servisní vrstva systému UFO

Servisní vrstvou rozumíme souhrn všech prostředků sloužících k zabezpečení činnosti programového jádra systému. Struktura servisní vrstvy je znázorněna na obr. 2.

Prostřednictvím servisní vrstvy se specifikují jednotlivé typy činnosti a to správa systému, aktualizace systému, uživatelský servis, vytváření optimalizačních programů, vytváření souborů vstupních dat a provádění optimalizačních programů.

Zpřístupnit všechny výše uvedené typy činností každému uživateli systému není vhodné. Proto jsou rozlišeny tři režimy práce systému a to:

- režim správce systému
- režim programátora systému
- režim uživatele systému

Správce systému má přístup ke všem komponentám, ke všem datovým a informačním souborům. Má právo vytvářet a rušit datové i knihovní soubory, může provádět globální aktualizaci systému. Pouze do režimu správce systému náleží:

- generování systému, tj. oživení systému UFO z distribučního média
- globální aktualizace systému, tj. provedení rozsáhlých aktualizací směn programového jádra, servisní vrstvy a datových souborů
- výpis neveřejných informací, tj. výpis obsahu souborů, které uchovávají informace o využívání systému, o jednotlivých uživateliích a různé statistické informace o četnosti využívání a časových nárocích jednotlivých komponent systému a jednotlivých optimalizačních metod.

Režim správce systému pod sebou zahrnuje i režimy programátora a uživatele systému.

Programátor systému má přístup ke všem komponentám s výjimkou těch, jejichž použití je výsadou správce systému. Programátor může měnit obsah programového jádra, obsah systémových, datových a informačních souborů. Nemůže rušit systémové, knihovny, datové a informační soubory. V režimu programátora jsou prováděny činnosti potřebné pro vývoj a úpravy systému:

- opravy a vytváření podprogramů jádra
- opravy a doplňování systémových souborů, tj. souborů obsahujících informace o systému a algoritmech, souborů s chybovými hlášeními, souborů se vzory výstupních textů, souborů se vzory uživatelských podprogramů, atd.

Režim programátora systému zahrnuje i režim uživatele systému v jehož rámci jsou zpřístupněny tyto činnosti:

- uživatelský servis, tj. výpis informačních souborů, výpis veřejných systémových souborů (např. informace o délce trvání a struktuře řešené úlohy).
- vytváření optimalizačních programů, tj. jednak vytváření podprogramů popisujících matematický model, jednak vytváření procedury řídicího jazyka pro generování hlavního řídicího programu a jednak (pro některé implementace) sestavení segmentovaného či nesegmentovaného optimalizačního programu.
- vytváření souborů vstupních dat, kde uživatel má možnost vytvářet a provádět kontrolu vstupních údajů pro vytvořený optimalizační program.
- provedení optimalizačního programu, kde uživatel může spustit optimalizační program buď v dávkovém nebo v interaktivním režimu a určit, co se má provést s výstupními soubory (zda vytisknout, zrušit, ponechat).

Pro zabezpečení činností požadovaných servisní vrstvou systému se použijí následující programové servisní prostředky:

- program pro práci se soubory systému UFO, který provádí aktualizaci záznamů v systémových souborech a jejich výpis.
- generátor systémových podprogramů, pomocí kterého se vytváří a opravují podprogramy systému včetně jejich kompilace a katalogizace do knihovnických souborů

- generátor matematických modelů, který v závislosti na typu a rozsahu řešené úlohy poskytne uživateli vzor uživatelského podprogramu, do něhož uživatel doplní konkrétní matematické či fyzikální závislosti.
- generátor řídicího programu, který v závislosti na typu, velikosti řešené úlohy a požadované strategii výpočtu vytvoří procedura ve vstupním řídicím jazyku a přeloží ji do tvaru řídicí programové jednotky (hlavního programu, podprogramu), přičemž vytvoří soubor s popisem segmentace optimalizačního programu.
- generátor vstupních dat, který v závislosti na typu a velikosti řešené úlohy a na tvaru vytvářeného řídicího programu umožní uživateli vytvořit a skontrolovat soubor vstupních dat.
- prováděcí program, který provede sestavení programu a spustí výpočet buď v dávkovém nebo v interaktivním režimu a po skončení výpočtu ošetří výstupní soubory.

Dále se předpokládá, že se budou používat některé standardní funkce a prostředky hostitelského počítače a jeho operačního systému, jako textový editor, program pro práci s informační datovou základnou (HELP-program), kompilátor jazyka "FORTRAN", případně kompilátor strojově orientovaného jazyka (ASSEMBLER), systémový sestavující program, knihovní program, program pro práci se systémem souborů a systémový prostředek pro spuštění proveditelného uživatelského, či systémového programu.

Celá servisní vrstva systému UFO bude umožňovat konverzační způsob práce formou tzv. nabídkového dialogu. S ohledem na rozsah příspěvku neuvádíme konkrétní příklady této konverzace. Základní principy této problematiky již byly na semináři PROGRAMOVÁNÍ několikrát diskutovány (viz např. [6]) a lze je rovněž nalézt v [2].

5. Datové soubory v systému UFO

Vzhledem k rozsahu příspěvku uvedeme jen stručný přehled souborů systému UFO.

Na konkrétní implementaci jsou závislé následující dva soubory:

- soubor pro generování systému, který obsahuje posloupnost příkazů jazyka pro řízení úloh zabezpečující instalaci systému UFO

na konkrétním počítači.

- soubor pro aktualizaci systému, který obsahuje posloupnost příkazů jazyka pro řízení úloh zabezpečující aktualizaci systému UFO na konkrétním počítači

Další soubory mají trvalý charakter a nejsou závislé na konkrétní implementaci:

- informační soubor, který obsahuje informace pro používání systému, popisy optimalizačních algoritmů, informace o doporučených optimalizačních strategiích, atd.
- soubor chybových a varovných hlášení
- soubor vzorů výstupních textů, který obsahuje masky pro vytváření částí výstupních tisků.
- soubor vzorů matematických modelů, v němž jsou uloženy kostry jednotlivých typů uživatelských podprogramů.
- soubor s popisem segmentace, který obsahuje úplný popis struktury překrývání podprogramů programového jádra systému UFO.
- soubor statistických informací, který umožní sběr informací o jednotlivých uživateli, evidenci četnosti vyvolání jednotlivých sledovaných komponent a podprogramů systému.

V rámci každé řešené optimalizační úlohy jsou jednotlivým uživatelům vytvořeny dočasné komunikační soubory:

- základní komunikační soubor, který obsahuje informace o uživateli a řešené úloze, uchovává historii prováděných prací a stav řešení úlohy. Slouží rovněž jako zápisník názvů uživateli vytvořených objektů a je využíván servisní vrstvou systému UFO.
- soubor s popisem segmentace optimalizačního programu, který slouží jako informační soubor pro systémový sestavující program. Na velkých počítačích se nebude vytvářet.

Dalším typem souborů jsou uživatelem vytvářené soubory vstupních dat a soubory s výsledky optimalizačních výpočtů. Jejich ošetření je ponecháno uživateli.

Posledním typem souborů systému UFO jsou knihovní soubory:

- systémová knihovna zdrojových textů
- systémová knihovna relokatabilních modulů
- uživatelská knihovna zdrojových textů
- uživatelská knihovna relokatabilních modulů

Literatura

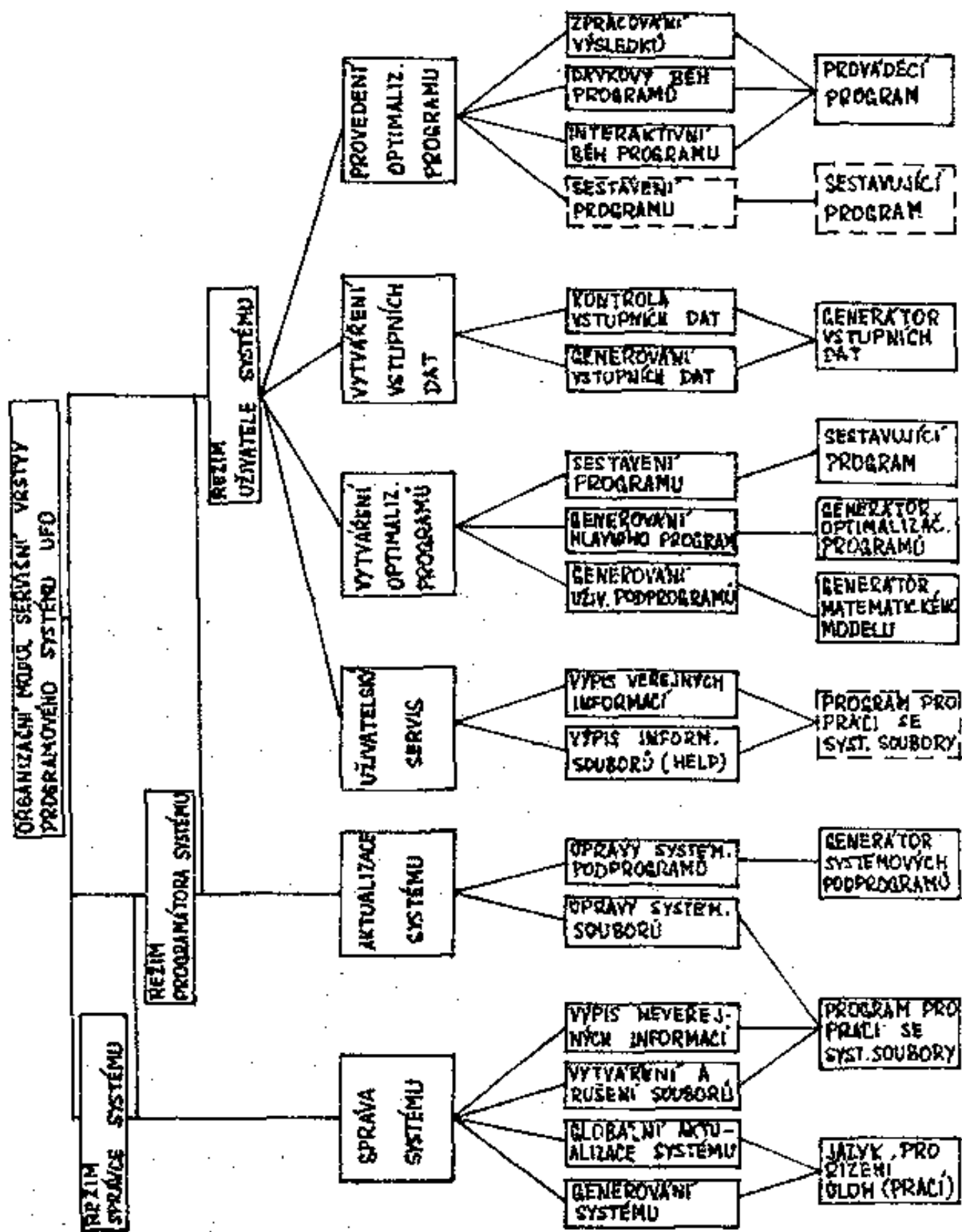
- [1] Madnick S.E., Donovan J.J.: Operační systémy. SNTL Praha 1981. (Překlad z angličtiny).
- [2] Martin J.: Design of Man-Computer Dialogues. Prentice-Hall, Inc., Englewood Cliffs, NY 1973.
- [3] Hřebíček J., Kopeček I., Kučera J., Lukšan L.: OPTIPACK, Uživatelský popis modifikace 2.1. ÚFM ČSAV Brno, 1983.
- [4] Lukšan L., Zamazal M., Kočková S.: SPONA-82, Uživatelský popis souboru programů pro optimalizaci a nelineární aproximaci /verze 1982/. SVT ČSAV Praha, 1983.
- [5] Lukšan L., Kučera J., Hřebíček J.: Koncepce systému UFO (Univerzální Funkcionální Optimalizace). SVT ČSAV Praha, ÚFM ČSAV Brno, 1983.
- [6] Bébr R.: Recept na jídelníček, aneb metoda dialogu formou "Menu". Sborník semináře Programování '84, str. 143-154.

Režim práce
systému UFO

Typ činnosti

Prováděná činnost

Programové
prostředky



Obrázek 2: Struktura servisní vrstvy systému UFO