

Dr. Jaroslav Široký
TOS Kufin

HIERARCHICKÉ SYSTÉMY

Systemy s hierarchickou výstavbou se vyskytují v mnoha oblastech lidské činnosti, jako jsou např.: organizace státu, armáda, průmyslový podnik, ASŘ. Všem těmto systémům je společné to, že se rozpadají na řadu subsystémů, a nichž každý je jedinému vyššímu celku podřízen a jeden nebo více nižších celků ovládá. Řízení systému je uskutečněno pomocí různých ovládacích a kontrolních prvků, figurují zde zpětné vazby apod. V terminologii ASŘ se rozpadá systém na subsystémy, nižší prvky jsou pak skupiny úloh, úlohy, funkce, chody, programy, podprogramy, příkazy.

Pro tvorbu jedné kategorie hierarchických systémů - programů - byla vypracována pozoruhodná metodika - strukturované programování. Pokusíme se ukázat, jak lze uplatnit metody osvědčené při tvorbě programů v širší oblasti - při tvorbě ASŘ, což je v podstatě rozšíření ideí strukturovaného programování na práci analytickou. Je ovšem pochopitelné, že některé myšlenky se nedají aplikovat vůbec, jiné lze převzít v pozměněné formě.

Takto nově aplikované ideje je možno rozčlenit do sedmi skupin, které zde uvedeme jako zásady.

1. Řeš problém od střechy

Najdříve je třeba vymezit systém jako celek, pak teprve postupně rozpracujeme jeho části: subsystémy, skupiny úloh, úlohy, funkce, chody, programy, podprogramy. Přístup "shora dolů" je zde přísnější, avšak opakovaný přístup nelze zcela ignorovat. Při návrhu systému musíme dbát na to, aby se daly při stavbě celku použít ty stavební kameny (prvky stojící v hierarchii na nižších úrovních), jejichž použití je vhodné nebo dokonce nutné. Mohou to např. být: různá zařízení pro vstup a výstup informací, technické možnosti počítače, programovací jazyk apod.

2. Divide et impera

Rozsáhlé úkoly, jakým je např. budování AŠP, rozdělíme na podúkoly ve dvou směrech. Pokud se týče podrobnosti zpracování, vytváříme několik etap, počínaje návrhem a konče prováděcím projektem. Každý další stupeň projektu představuje zpřesnění, je tedy rozsáhlejší, než stupeň předchozí a dělíme jej proto zpravidla na několik paralelně zpracovávaných odvětví, z nichž každou má na starosti jiný pracovník.

Každá část úkolu samostatně zpracovávaná musí mít definovanou oblast působení a návaznosti na jiné části. Při zpracování úkolu musíme zvládnout, zda každá činnost systému je zachycena právě v jednom podúkolu. Zvláštní pozornost musíme věnovat datové struktuře, která musí umožňovat přenos údajů mezi podúkoly.

Je vhodné omezit i velikost dokumentů, jimiž systém a jeho části popisujeme. Grafické formy, jako jsou např. vývojové diagramy, se stávají těžko přehlednými, pokud jsou rozměřeny na formátu větším než A3. U rozsáhlých schémat si pomůžeme tak, že jedním diagramem zachytíme hrubou strukturu a v dalších diagramech pak zobrazíme detaily.

3. Raději organizuji, než pracuji

Zkušenosti ukazují, že maximální péče se musí věnovat při navrhování systému jeho celkové struktury, vzájemným vztahům systému a okolím, i vztahům uvnitř systému. Musíme mít neustále na paměti účel, jemuž má systém sloužit. Proto především vymezíme požadky ze strany uživatele a odezvy na ně. To se týká nejen systému jako celku, nýbrž každé jeho samostatně vypracovávané části.

4. Dodržuji pravidla slušebního postupu

Je nanejvýše žádoucí dodržovat v systému pravidla o podřízenosti a nadřízenosti, jako je tomu např. v armádě. Každý prvek systému musí mít přesně definovaný okruh informací, které může zpracovávat. Komunikace o jiných prvech musí procházet hraničními prvky jednotlivých hierarchických stupňů směrem nahoru i dolů. Je zřejmé, že je žádoucí redukovat počet hraničních prvků na minimum.

5. Co nemusíš udělat dnes, odlož na zítra

V každé etapě řešení problému řešíme jen ty problémy, které v dané etapě musíme řešit. Např. při návrhu informačního systému potřebujeme v první fázi vědět jen to, že budeme pracovat se souborem zaměstnanců. Při dalším zpracovávání systému doplníme seznam údajů, při psaní programů musíme o každém údaji vědět, jakého je typu, z kolika znaků se může skládat apod. Výhoda takového postupného zpřesňování je také v tom, že při eventuelních změnách, které se vyskytnou především v počátečních fázích projektování, neplytváme příliš lidskou prací při případných změnách.

Postupné zpracovávání by mělo postupovat v ideálním případě rovnoměrně ve všech fázích systému. To ale není vždy možné z různých důvodů. U rozsáhlejších systémů odložíme někdy

s kapacitních důvodů zpracování celých částí. Jindy opět urychlíme rozhodnutí u některých dílčích problémech, je-li např. realizace vázána na koupi speciálního zařízení s dlouhou dodací lhůtou.

6. O práci co nejvíce mluve

Každý systém musí být popsán jasně, přehledně, výrazně, aby i nezavěšený pochopil ze studia dokumentace, jak systém funguje. Podle typu systému a určení dokumentace je třeba volit různé prostředky:

- slovní popis
- grafy, např. vývojové diagramy, Hipo diagramy
- tabulky, např. rozhodovací tabulky, matice
- formuláře, např. pro popis souboru, popis údaje, text programu
- grafické úprava textu především u programových textů děrovaných do černé pásky
- vysvětlivky především u programových textů

Mnohohlavnost však není vždy dobrá. Jestliže je o nějakém prvku zmínka v dokumentaci na zbytečně mnoha místech, pak nastávají problémy při eventuelních změnách, které mají za následek opravy na mnoha místech.

7. Kontroluj práci kolektivu

Rozsáhlý úkol nemůže řešit jedinec, musí zde fungovat tým, často i více týmů. Dobrou koordinaci mezi členy týmu nebo mezi týmy zajišťujeme zvlášť vyčleněnými pracovníky. Pečlivě musíme kontrolovat styčná místa těch částí systému, která jsou zpracovávána různými lidmi, zda stále umožňují přenos patřičných informací.

Je dobré mít přehled o tom, ve kterých funkcích se soubory systému (subsystému) uplatňují. Tento vztah lze zachytit zvláště dobře formou matice, kde v řádcích jsou soubory a

ve sloupcích funkce. Prvky matice pak mohou nabývat hodnot:

- I - funkce spíná soubor
- O - funkce zapisuje soubor
- A - funkce aktualizuje soubor
- - funkce se souborem npracuje

Jako příklad uvedeme matici zobrazující vztah mezi soubory S1, S2, S3, funkcemi F1, F2.

	F1	F2
S1	I	O
S2	A	-
S3	-	I

Jednou z výhod maticového zápisu je to, že při eventuálním rozšiřování systému o další funkce opravujeme jedinou tabulku.

Platné služby zde prokáže i počítač. Abychom mohli automatizovaného zpracování využít, zapíšeme formou seznamu vztah mezi prvky a jeho komponentami na nejbližší nižším stupni hierarchie (vyjma nejnižšího stupně). To učiníme pro tolik hierarchických stupňů, kolik jich chceme sledovat. Seznamy uložíme jako data určená pro zpracování kusovníkovým procesorem. Jeden z možných příkladů členění systému uvedeme v následující tabulce.

vyšší prvek	nižší prvek
system	subsystem
subsystem	skupina úloh
skupina úloh	úloha
úloha	funkce
funkce	soubor
soubor	data

Pak můžete automatizovaně získat např. seznam všech údajů buď v celém subsystému, nebo v některé jeho části. Některé kusovníkové procesory dovedou řešit i úlohu opakovanou: např. které úlohy, skupiny úloh a subsystémy začínají změna vyvolávané korektí některého souboru.

Literatura

- (1) Král J. : Strukturované i nestrukturované programování v podmínkách parkinsonovského podání, Informační systémy č.1/1975
- (2) Metody tvorby informačního systému, výzkumné výpočtové středisko Bratislava, 1976
- (3) Polach O. : Dokumentační prostředek HIPO, Metody programování počítačů III.generace, Havířov 1976
- (4) Lochman B. a kol : Aplikace banky dat, socialistická akademie ČSR, Praha 1975.