

Programové řešení typových projektů pro různorodé uživatele

Úvod:

S příchodem III. generace vyrojila se řada programovacích metod a metodik, z nichž většina má být ideálními řešeními všech programátorských problémů.

Tento příspěvek nechce v žádném případě rozmožnit tuto řadu o nový, dosud nevidaný a jedině správný systém. Úkolem následujících řádků je shrnutí několika postřehů a poznatků z jedné zajímavé oblasti projektování a programování.

Laskavý čtenář nechť chápe celý text jako hezlovitý sáznem různých nápadů; ucelená práce by spotřebovala značné množství papíru, což nebylo autorovým úmyslem.

1. Základní pojmy:

1.1 Projektové a programové zabezpečení:

Jak známo, panuje v oblasti terminologie výpočetní techniky jistá anarchie (i když v poslední době nastává obrát k lepším). Pojmy "software", "projekt", "programové zabezpečení" a pod. jsou různě vykládány a interpretovány.

Pro účely této práce budeme pod pojmem "projekt" rozumět kompletní návrh automatizovaného zpracování infor-

mací včetně programového zabezpečení.

Projekt řeší nějaký systém, členěný na subsystémy a prvky propojené interakcemi ve smyslu teorie systémů.

Nebudeme se zabývat formální stránkou projektu a jeho členěním do etap a fází. Předmětem této práce je obsahová náplň projektu.

1.2 Specifika III. generace.

Existuje dostatek literatury, která se zabývá odlišnostmi projektů pro III. generaci počítačů proti "klasickému" projektování II. generace. Z hlediska našich úvah je významné především:

- rozsáhlé užívání vyšších programovacích jazyků
- orientování projektů zpracování hromadných dat na řešení souborů jako klíčových bodů projektu (lit./1/)
- modulární řešení projektů
- využívání možností a vlastností operačních systémů.

1.3 Typové projekty.

Ekonomická hlediska vedou k tomu, aby jednou již zpracovaný projekt (většinou velmi nákladný) byl co nejvíce využíván a to nejen opakováním zpracování v čase, ale i použitím pro více uživatelů. Toho lze dosáhnout vypracování typových projektů. Potíž je v tom, že představy o obsahu a vlastnostech typového projektu se dosti různí.

Základní požadavek bývá formulován v tom smyslu, aby podle jediného projektu bylo možno řešenou úlohu zpracovávat pro nejrůznější uživatele v rozličných podmínkách. Tento požadavek je maximalistický a nelze mu beze zbytku vyhovět. Doplníme tedy formulaci tak, že od typového projektu se žádá nejvyšší možná přispůsobivost; tato vlastnost nesmí ovšem narušit základní kvality typového projektu.

Přizpůsobení konkrétnímu uživateli

- musí být snadné a levné
- nemůže být podmíněno zásahem do základních struktur projektu
- musí mít zajištění proti narušení systému.

Poslední podmínka je velmi důležitá a rozumíme jí takto: jestliže upravujeme část systému pro uživatelské potřeby a dopustíme se přitom chyb, nesmí se tyto chyby přenést do jiných částí systému.

1.4 Zájmy uživatelů.

Položme si nyní otázku, jakým zájmům a potřebám uživatelů musí se typový projekt přizpůsobovat. Konkrétní uživatelé se různí především

- ve vnitřní struktuře organizace
- v sortimentu používaných dokladů
- ve způsobu dodávání dat ke zpracování (formulář, děrná páska, přenos dat,)
- v kvalitativních i kvantitativních požadavcích na výstupy
- v rozsahu funkcí, které od systému požadují.

Teoreticky by bylo možné direktivně sjednotit všechna hlediska u všech uživatelů; v řadě prací se dokazuje, že to je jediný možný způsob efektivního zavádění automatizace. V praxi však je výhodnější přizpůsobit se podmínkám uživatele a nekomplikovat mu přechod na nový systém, který je sám o sobě vysoce náročnou operací.

1.5 Krize systémového přístupu.

Myšlenky, vyslovené v předcházejícím odstavci budou napadány jako ukázka nesystémového přístupu. Ve své podstatě však naplňují základní hlediska systémového inženýrství.

Setkáváme se tu s jevem, který nelze nazvat jinak než krize systémového inženýrství a který právě v oblasti typových projektů vede k hrubým omylům.

Hlavní ideou systémového přístupu je řešení nějakého úkolu v plné komplexnosti s respektováním všech hledisek (tedy i lidských vztahů, reálných možností a pod.). V podstatě jde o nalezení optimálního kompromisu. V poslední době se však bobulel stal "systémový přístup" módním pojmem; je používán všude, v možných i nemožných souvislostech, většinou bez základních znalostí příslušné teorie. Při takovémto masovém používání bylo nutno celé systémové inženýrství redukovat na několik snadno zapamatovatelných pouček, které se pak bez dalších úvah aplikují na cokoliv. Krize spočívá v tom, že právě tento styl práce se snažilo systémové inženýrství vymýtit.

V oblasti typových projektů se popisná "metoda" projevuje tak, že je kádán projekt univerzální, neměnicí se, zpracovaný hned na začátku pro celý (sebevícе rozsáhlý) systém; dále je tvrdě požadováno, aby se uživatel zcela přizpůsobil systému. Tyto požadavky jsou ve své nezákladnější podstatě správné, avšak jejich aplikace je nesystémová. V praxi musíme konstatovat, že

- univerzální řešení, pokrývající celý systém je ve většině případů nerealizovatelná v úcné době; řešitelské kapacity bývají omezeny a tak se projektování prodlouží na řadu let; v době realizace je systém sice krásný, avšak zastaralý;
- projekt má pro uživatele význam, jestliže mu pomůže při řešení problému, a to včas; leckdy bývá vysoce efektivní realizovat v poměrně krátké době jen jistou část systému, které ihned funguje; rozšiřování si vyžádá sice případně vícenásledy, ty však převyšuje užitek, získaný fungující částí;
- pro uživatele, který má s automatizovanými systémy malé zkušenosti znamená komplexní nasazení rozsáhlého systému takřka pohromu, ze které se dlouho vzpamatovává; nasazování po částech umožňuje adaptaci uživatele, jak personální, tak i organizační.

Ve výčtu hledisek bychom mohli dále pokračovat, avšak většině praktických jsou tyto poznatky známy. Pro metodiku typových projektů mají naznačené myšlenky velký význam; pokud jim typový projekt vyhoví, je z praktického hlediska vysoce užitečný, i když může být napaden pro "nesystémový přístup". Ve skutečnosti je takový projekt právě ukázkou systémového myšlení.

2. Vymezení problému:

2.1 Výchozí předpoklady.

Dosud jsme hledali požadavky, kterým by měl typový projekt vyhovět. Pokusme se nyní najít způsob, jakým tyto požadavky realizovat.

Tento způsob by neměl narušit žádnou známou metodiku projektování a neměl by být v rozporu s užívanými nebo zaváděnými programovacími technikami.

2.2 Užívané metodiky.

V poslední době se věnuje velká pozornost efektivním metodám tvorby programů (viz na př. /2/). Rozpracováno je normované programování, do praxe proniká strukturované programování (viz též /3/). Většina programátorů III. generace si osvojila modulární řešení programů a programových osků.

Všechny uvedené metody i řada dalších jsou velmi perspektivní právě pro aplikace v typových projektech. Položme si tedy jako podmínku, že návrhy k řešení typových projektů budou s použitím zmíněných metod počítat.

2.3 Princip tvorby zón.

Při studiu problematiky typových projektů se ukázalo, že dobře vyhovíme výše formulovaným požadavkům, rozdělíme-li systém na tři základní zóny:

- vstupní zóna (jejímž úkolem je převést jakékoliv vstupy na jednotnou základnu)
 - zóna jádra (která provádí vlastní transformaci údajů podle algoritmi úlohy)
 - výstupní zóna (převede jednotnou výstupní základnu na konkrétní požadované výstupy).
- Uvedené zóny komunikují výhradně pomocí souborů, které tvoří hraniční základny mezi zónami.

3. Popis zónového způsobu:

3.1 Vymezení zón:

Obtížnou úlohou je vymezení rozsahu zón. Postupujeme od zóny jádra, kterou stanovíme co nejrozsáhlejší. Požadujeme však, aby jádro nebylo závislé na vstupech a výstupech co do fyzické podoby (media) i co do logické struktury. Jádro bude tvořit dlouhodobě stálou složku systému a bude základem "typovosti" projektu.

3.2 Hranice zón:

Je-li vymezena rozsah zón, stanovíme hraniční základny (soubory) na straně vstupu i výstupu. To je nejdůležitější úkol celé analýzy a na jeho úspěšném zvládnutí závisí úspěch projektu. K řešení se vyžaduje výborná znalost práce s datovými soubory a zkušenost při tvorbě vět a jejich sledů.

Nejprve stanovíme logickou strukturu hraničního souboru; navrhneme věty a jejich sledy, definujeme položky a formáty. Dále je nutno určit fyzickou podobu hraničního souboru: médium, kód, konvence, návěští a pod. Soubor přesně popíšeme i z hlediska hardware (u mag. pásky počet stop, hustota zápisu atd.).

Ukazuje se, že výhodným médiem hraničních souborů je magnetická páska.

3.3 Zóna jádra.

Jádro musí respektovat všechny podmínky převodu vstupních informací na výstupní, dané vyhláškami, směrnici, technologickými předpisy, základními algoritmy atd., platnými pro danou úlohu.

Jádro přebírá se vstupního hraničního souboru "vyčištěné" údaje. Do výstupního hraničního souboru odevzdá jádro veškeré informace, které se mohou v různých výstupech objevit.

Jádro řešíme modulárně a snažíme se o stavebnicovost. Používáme vyšších programovacích jazyků, u nichž odpadnou některé potíže (místo se vstupy a výstupy pracujeme s definovanými soubory). Výhodný je COBOL, též vzhledem k přehlednosti a čitelnosti zápisu algoritmů a popisu souborů.

3.4 Zóna vstupu.

Ve vstupní zóně se provádí na příklad

- převody medií
- převody kódů
- převody konvencí
- kontrolní a opravné řízení vstupujících dat.

Předpokládá se využívání elementárních strojových funkcí a tudíž aplikace assembleru. S výhodou pracujeme s firemním software (konverzní programy).

Ve vstupní zóně se setkáme se dvěma skupinami problémy, které je nutno řešit a vzájemně sladit:

- technické (překlady, převody, rychlost počítače)
- systémové (čištění dat, opravná řízení).

3.5 Zóna výstupu.

Ve výstupní zóně se provádí na příklad

- celková redakce výstupů
- výběr a řazení výstupních informací

- převod kódů při výstupu do medií
 - úprava do pořadované grafické podoby u sestav.
- Výběr programovacích prostředků je pestrý - od assembleru pro speciální funkce až po RPG pro tvorbu sestav. Řešíme opět dva okruhy problémů:
- technické (převody, využití periférií, efektivní tisk)
 - systémové (grafika tiskového obrazu, jazyková úroveň zkratk a záhlaví, přehlednost, využití symboliky, účelnost).

V této souvislosti je možno upozornit na dnešní zcela nedostatečnou grafickou a jazykovou úroveň výstupních sestav. Oddělené řešení zóny výstupů dává u typových projektů možnost využít ke zlepšení úrovně příslušné odborníky.

3.6 Přizpůsobení uživateli.

Adaptace typového projektu pro specifické potřeby konkrétního uživatele se provádí změnou nebo výměnou modulů v zóně vstupů a výstupů. Je účelné vytvářet si knihovny nejužívanějších modulů pro zóny vstupu a výstupu.

Uživateli lze na příklad umožnit v první etapě práci s formuláři (vstupní zóna řeší děrování štítků a jejich přepis do hraničního souboru), v další etapě pak přechod na pořizování děrné pásky přímo u uživatele (konverzní modul štítků se nahradí modulem převodu děrné pásky) atd. V zónách vstupů a výstupů můžeme se poměrně jednoduchými úpravami přizpůsobit i organizačním zvláštnostem uživatelů.

Komplikovanější situace nastává u systémů, kde uživatel může požadovat různý rozsah funkce jádra. Zde se vyplatí stavebnicové řešení jádra. Hraniční soubory musí být ovšem definovány pro plnou skladbu stavebnice.

4. Organizace prací na projektech:

4.1 Vztah analytik - programátor.

S příchodem III. generace vzniká nový pohled na práci analytika a programátora. Problém je výborně popsán v lit. /4/. V podstatě jde o zanikání rozdílu mezi uvedenými profesemi a jakýsi obsahový posun pojmů.

Klasické členění na oddělení analytická a oddělení programátorská (s komplikovanou komunikací) se ukazuje jako nevýhodné.

4.2 Vztah vedoucí týmu - kolektiv.

Mnoho pozornosti se věnuje výběru vedoucích pracovních týmů. Hledá se kompromis mezi krajnostmi

- špičkový odborník (s velkou autoritou a schopností odborně vést a řídit řešení)
- organizátor (schopný zvládnout rozsáhlou administrativu velkých projektů).

Praxe se přiklání k výběru špičkových odborníků, kteří pak zápolí s problémy administrativy, což brání plnému využití jejich odborné kvalifikace.

4.3 Vztahy mezi týmy.

Je známo, že pracovní tým podává nejvyšší výkon při určitém počtu pracovníků. Snížení i zvýšení počtu znamená vždy snížení výkonu. Na větších projektech - zvláště pak typových - se musí tedy podílet několik pracovních týmů. To klade vysoké nároky na organizaci prací, komunikaci mezi týmy a navíc přináší problém odpovědnosti za projekt jako celek.

4.4 Organizace projekčního útvaru.

Zkušenosti světových i tuzemských pracovišť ukazují, že je účelné organizovat větší projekční útvar jinak, než

bylo zvykem u II. generace. Rozdělení na týmy (pracovní skupiny) zůstává, mění se však náplň práce a pracovní vztahy. Skupiny se rozpadají na dvě hlavní třídy:

- technologické pracovní skupiny
- servisní pracovní skupiny.

Technologické skupiny se věnují vlastnímu řešení projektů a servisní skupiny jim k této práci vytvářejí podmínky.

4.5 Práce technologických skupin.

Technologické skupiny řeší zónu jádra a zóny vstupů a výstupů.

Pro zóny vstupů a výstupů je výhodné určit některou skupinu (nebo skupiny) natrvalo. Skupina vstupů a výstupů se pak specializuje, nabývá velikých zkušeností a zřizuje si knihovny modulů pro vstup a výstup. Ve skupině jsou programátoři znalí assembleru; většinu prací tvoří rutinní psaní překladových tabulek a převodních programů. Pro výstupní programy je vhodná znalost specializovaných jazyků, většinu práce tvoří počítání pozic na výstupních sestavách. Skupina musí mít však i špičkové odborníky pro řešení koncepce vstupních a výstupních zón a pro systémové problémy. Vyplatí se i specialisté na grafický obraz sestav a na jazykové otázky.

Ostatní technologické skupiny řeší jádra různých úloh. Každá skupina provádí analýzu i programování, což je při použití vyšších programovacích jazyků možné a účelné.

4.6 Práce servisních skupin.

Servisní skupiny mají za úkol zajišťovat některé práce pro usnadnění činnosti technologických skupin. Základní servisní skupiny jsou tyto:

- organizace: zajišťuje veškeré administrativní a pomocné práce (vedení evidencí, psaní zpráv, reprografie, personální záležitosti, výkazy práce,

plánování, fakturace a pod.) tím se podstatně odlehčí vedoucím technologických skupin a je možno do čela skupin postavit odborníky bez jejich přetěžování administrativou;

- koordinace: plní dva hlavní úkoly:
 - zajišťování "odbytu" projekčního útvaru v plné komplexnosti
 - koordinaci prací technologických skupin včetně plánování práce (jsou i názory, že by tato skupina měla přebírat odpovědnost za každý projekt jako celek);
- operační systémy: v této skupině jsou systémoví programátoři se všemi svými funkcemi a povinnostmi.

V některých případech bude patrně vhodné, aby skupina vstupů a výstupů byla vedena jako skupina servisu.

4.7 Využití kooperace.

Zónový způsob s hlavní myšlenkou přesné definice hraničních souborů umožňuje kooperaci projekčních útvarů. Je možno využít i subdodávek od jiných podniků. Komunikace a formulace požadavků je zjednodušena. Formální záležitosti řeší koordináční skupina.

Ve větších podnicích na zpracování dat je možno provádět úpravy vstupních a výstupních souborů přímo na závodě. S výhodou zde využijeme i práce provozních programátorů, kteří mohou vstupy a výstupy přizpůsobit potřebám a možnostem vlastního závodu a jeho uživatelů.

4.8 Dokumentace.

Pro typové projekty se soubory dokumentujeme odděleně jádro a soubory vstupů a výstupů. Zatímco jádro je popsáno detailně, pro soubory vstupů a výstupů můžeme vypracovat

obecný popis, přehled použitelných knihovních modulů a návod k modulární výstavbě zón.

Při aplikaci pro konkrétního uživatele dokumentujeme zónu vstupů a výstupů podrobně.

5. Provozní hlediska:

5.1 Vytížení počítačů, dislokace prací.

Každý projekt - a zvláště typový - má dávat provozu výpočetního střediska možnost co nejvýhodnějšího provozního režimu. Pro splnění tohoto požadavku musí platit:

- projekt nesvazuje provozu ruce
- provoz je dobře organizován a řízen kvalifikovanými pracovníky.

Projekty členěné do zón dávají provozu dostatek volnosti. Princip hraničního souboru umožňuje časové i prostorové členění prací. Lze na příklad zpracovávat vstupy a výstupy lokálně (blíže k uživateli - což je důležité zvláště pro opravná řízení), jádro centrálně. Přenos hraničních souborů se může dít fyzickou cestou nebo i pomocí přenosu dat.

Vhodně zvolený provozní režim umožňuje lepší využití dostupné techniky. Jádro zpracováváme na větších počítačích, vhodných pro manipulace se soubory a řešení složitých algoritmů (u takových počítačů bývají i efektivní kompilátory vyšších jazyků, ve kterých je jádro programováno); zpracování je rychlé, ve hře nejsou snímače a tiskárny. Vstupy a výstupy lze přidělit menším, pomalejším a jednodušším počítačům. Práce je zde limitována rychlostmi periférií.

5.2 Spolupráce II. a III. generace.

Většina středisek přechází na III. generaci v době, kdy počítače II. generace ještě slouží. Je možné přejít na III. generaci především s jádrem (využije se rychlost,

paměť, práce se soubory atd.), kdežto vstupy a výstupy zpracováváme nadále na II. generaci. Je-li zajištěna kompatibilita strojového parku na úrovni hraničních souborů (na příklad magnetické pásky), mohou být stávající počítače II. generace ještě dlouho využívány. Přejechod na novou techniku je postupný, usnadní se i řešení takových problémů, jako je převod střediska z 90-sloupcové na 80-sloupcovou soustavu.

5.3 Problém knihovny.

Typové projekty pro různorodé uživatele vždy komplikují uživatelské knihovny počítače. Ani sčnový způsob nevede k významnému zjednodušení této problematiky.

V podstatě se rozhodujeme mezi dvěma možnostmi:

- a) typový projekt je uložen v knihovně relocatable modulů a linkuje se těsně před výkonem nebo krátce před ním
- b) typový projekt je linkován pro každého uživatele a uložen v knihovně absolutních programů.

Způsob "a" klade nároky na organizaci provozu, kvalifikaci provozu a perfektní dokumentaci. Způsob "b" je sice pohodlný, má však zvláštní nároky na evidenci a na místo na discích. Projeví se to zvláště v operačních systémech typu DOS, kdy provoz většinou požaduje linkování pro všechny partition a máme tedy každý projekt linkován pro každého uživatele několikrát.

Osobně nám za to, že práce se příkloní ke způsobu "a".

5.4 Zvláštní nároky.

Při použití typových projektů pro různorodé uživatele musí se provoz střediska vypořádat s některými zvláštními problémy.

Především musí být v naprostém pořádku veškerá dokumentace a evidence. Datové soubory stejného typu a názvu

se opakují pro každou zakázku. Ve většině případů pak přibývají ještě generace těchto souborů s nutností archivace. Pracuje se podle těchto programů, které se v detailech liší pro různé uživatele a je třeba se vyvarovat záměny. Pokud jsou k dispozici absolutní programy, musí být odlišeny podle uživatelů. Linkujeme-li před výkonem, musíme pracovat podle určitého plánu, aby výsledkem byla skutečně žádaná uživatelská verze.

Tyto nároky se soustředí na osobě, která je někde nazývána "systémový organizátor", jinde jinak a jinde vůbec chybí. Její hlavní úkolem je plánovat a rozvrhovat práci, připravit job-streamy a řídit přísuny dat. Kvalifikace a schopnosti "systémového organizátora" jsou limitujícími prvky provozních úspěchů. Toto tvrzení platí pro III. generaci obecně, v oblasti typových projektů pro různorodé uživatele má klíčový význam.

6. Závěr:

Základní myšlenky tohoto příspěvku nejsou možná pro mnohé pracovníky ničím novým. Snahou autora bylo ukázat, že i jednoduchý nápad vyžaduje ke své realizaci vytvoření komplexních podmínek, od teoretické základny přes organizaci až po provozní zajištění. Jestliže to někomu připomíná systémový přístup, nemá autor proti takové úvaze námitek.

Literatura:

/1/ Judd, D.R.: Use of files, Macdonald, London & American Elsevier Inc., New York, 1973.

slovensky: Použitie súborov, ALFA 1975.

/2/ Vlček, J.: Vývoj programovacích prostředků. Aktuality výpočetní techniky, VÚMS 1974, č. 1.

/3/ Hořejš, J.: Principy strukturovaného programování. Informačné systémy, ALFA 1975, č. 2 a 3.

/4/ Grégrová, M.: Generační přelom ve výpočetní technice a v automatizaci řízení. Výběr informací, ISn.p. 1975, č. 5.