

... METASYSTÉMOVÉ JAZYKY
A POČÍTAČOVÁ PODPORA
ANALYTICKO-PROGRAMÁTORSKÉ PRÁCE

Ing. Vlastimil Čevela, VS Ingstav n.p. Brno

Cílem předkládaného příspěvku je pokus o celkový pohled na analyticko-programátorskou práci se záměrem rozebrat některé možnosti využívání počítače ke zvýšení její kvality a produktivity.

1. Základní pojmy

Jedním z vážných problémů při zavádění automatizace do budování ASŘ je terminologie. Potíž vzniká v tom, že s pomocí pojmů, které běžně používáme v rámci procesu vývoje programového vybavení, máme nyní popisovat tento proces jako takový. Zkušenosti však ukazují, že takto podávaný výklad se velice rychle stává nejednoznačným a nesrozumitelným. Proto nejprve některé užitečné definice:

Automatizovaný systém řízení dle /1/ je systém řízení, ve kterém jsou vybrané činnosti a procesy automatizovány s využitím prostředků výpočetní a jiné techniky...; vytváří se pro jednotlivé stupně řízení a jednotlivé účelové činnosti a procesy - např. pro výrobní, technologické, obslužné, skladovací procesy, projektové činnosti apod.

Pro souhrn prostředků k tvorbě, řízení a údržbě ASŘ si zavedeme pojem metasytém, který je jedním z jeho zabezpečovacích subystémů. Metasytémový jazyk pak budeme definovat jako prostředek pro popis systému nebo jeho částí, jinými slovy tedy jako jazyk pro tvorbu projektové dokumentace.

Obsahem projektové dokumentace dle /1/ je především:

- popis struktury, funkcí a vuseb /vstupů - výstupů/ systému, podsystémů, jejich ucelených částí a úloh
- popis struktury datové základny, specifikace datových prvků a číselníků
- programová dokumentace, operátorská a uživatelská příručka

Kadřně tedy do ní patří nejen zdrojové texty programů a návody k jejich použití, ale i další dokumentační prvky, tvořené v průběhu analytické práce.

2. Metodika řešení systému

V návaznosti na /2/ můžeme za princip systémové práce považovat následující činnosti:

- A - definice celku /systému/ jako množiny částí
seznam všech částí, ze kterých je celek utvořen, logických částí by přitom nemělo být více než 10 a méně než 4
- B - definice struktury celku
stanovení vlastností částí provedením činností A, B, D na dalším stupni podrobnosti
- D - určení vlastností celku
syntéza výsledků z činností A, B, C

Použití uvedeného principu je základním předpokladem jak pro úspěšné vyžití dělby práce, tak pro řešení i velice složitých úkolů. Jde totiž o jádro přístupu, kterým se lze v každé fázi práce oprostit od právě nepodstatných detailů, učinit si pořádek v množství nejrozsáhlejších aspektů a najít zvládnutelně jednoduchou podobu problému.

Jestliže výsledkem analyticko-programátorské práce je pro-

jektová dokumentace, pak k zajištění počítačové podpory této činnosti je třeba:

- rozdělit prvky projektové dokumentace na prvotní, které musí vytvořit a do počítače uložit člověk a na odvozené, vytvářené automatizovaně
- zjednodušit a sjednotit formu jednotlivých dokumentačních prvků do několika základních kategorií a každé z nich přiřadit příslušně formalizovaný metajazyk
- mít k dispozici takové programové vybavení pro zpracování metasytémových dat /počítačové dokumentace/, které umožní splnění všech potřebných funkcí, ale nebude příliš náročné na lidskou práci, ani na zdroje počítače.

V další kapitole jsou charakterizovány metasytémové jazyky a forma prvotní dokumentace, které byly dle víceletých zkušeností na počítačích Tesla 200 a EC 1026 ve VS Ingstav Brno předloženy v návrhu /8/ pro resort stavebnictví ČSR.

3. Metasytémové jazyky

3.1 Funkční popisy

Funkční popisy jsou prvotními dokumentačními prvky pro všechny automatizované procesy, tj. systém jako celek, subsystémy /= agendy/, skupina úloh /= řešitelské části/, úlohy /= výpočtové chody/, samostatně využívané procedury příkazových štítků JCL, programy, podprogramy, terminálová či jiná makra a příp. další /podrobněji v /7//.

Cílem při sestavování funkčních popisů na každé úrovni dekompozice /U/ je definovat celek jako množinu dílčích částí, specifikovat vnější vazby tohoto celku a stanovit jeho základní vlastnosti.

Navrhovaný metasytémový jazyk funkčních popisů představuje členění do dále specifikovaných popisných bloků P1 až P5, s předepsaným řazením jednotlivých údajů ve formalizovaných řádcích

bloků P1 až P3 a využívání dekompozičních kódů I, O, W v bloku P2. Tato forma pak umožňuje automatizovanou tvorbu odvozeného přehledu prvků a vazeb.

P1 Funkční popis

'textový název popisovaného prvku' jm-prvku U 'řešitel'

P2 Vstup/výstupní charakteristika

I 'textový název datového vstupu' jm-prvku

I

O 'textový název datového výstupu' jm-prvku

O

W 'text. název aktualizovaných dat' jm-prvku

W

P3 Dílejší procesy a číselníky

'textový název procesu' jm-prvku

... ..

'textový název číselníku' jm-prvku

... ..

P4 Cíle a hlavní funkce

volným textem psaná charakteristika funkcí popisovaného řešení, které nevyplývají z výše uvedených bloků

P5 Podmínky

volným textem psaná specifikace vnějších podmínek, které nevyplývají z výše uvedených bloků

3.2 Datové popisy

Datové popisy jsou prvotními dokumentačními prvky pro popis veškerých datových struktur, využívaných v rámci automatizovaného zpracování dat.

Účelem datových popisů je stanovit existenci a vzájemné vztahy vět, řádků, setů a dalších dílčích datových struktur v rámci vyšších celků, jako jsou datové soubory, prvotní doklady, výstupní sestavy, databankové schémata apod., a pro každou z těchto dílčích struktur pak definovat výskyt a řazení jednotlivých datových údajů.

Metasystémovým jazykem pro datové popisy je ÚROVŇOVÝ POPIS dle norem vyšších jazyků jako je Cobol, DDL-IDMS atd, opatřený kromě symbolických identifikátorů též srozumitelným textovým popisem každého údaje a sestavený tak, aby byl zpracovatelný příslušným překladačem.

Datové popisy jsou definovány v maximální podrobnosti, tj. jako skladba jednotlivých elementárních údajů, zpravidla pouze na jedné úrovni pro celou datovou strukturu.

Datové popisy jsou zásadně ukládány odděleně od programů ve zdrojovém tvaru, aby mohly být vícenásobně využívány např. pomocí příkazu Copy v Cobolu či podobně a sloužit pro automatizovaně odvozený adresář výskytu datových údajů.

Podmínkou pro automatizovanou tvorbu adresáře dat je, aby identifikátory každého údaje byly v rámci jeho výskytů v různých datových postupech systému nebo jeho části shodné a jednoznačné. Metodika definování datových údajů včetně identifikace pomocí kombinovaných, v podstatě pořadově přidělovaných písmeno-číselnicových 1+3 místných kódů je vyčerpávajícím způsobem popsána v pravidlech pro tvorbu jednotné údajové základny organizací dle /10/. Pro definování číselníků pak je možno výhodně využít jazyk číselníkůvých modulů, popisovaný v další kapitole.

S využitím slovníku dat lze tvorbu datových popisů částečně automatizovat dosazováním definic a komentářů na základě uvedení příslušných identifikátorů. Přitom je však třeba zajistit, aby bylo možno zobrazení jednotlivých údajů v různých datových popisech výjimečně dle potřeby též individualizovat.

3.3 Číselníky

Práce s informacemi číselníkového charakteru tvoří jednu z nejdůležitějších funkcí programového vybavení. Číselníky tedy mají svým způsobem jeden z největších podílů na algoritmicizaci řešených problémů a přitom podléhají nejčastěji změnám. Racionální přístup k analyticko-programátorské práci proto vyžaduje, aby číselníky byly odděleny od vlastních programů.

Číselníky představují prvotní dokumentační prvky, které definují vztahy číselníkového charakteru a tím vlastně řeší velkou část algoritmicizace při zpracování hromadných dat.

Vytvoření originálního metasystémového jazyka a překladače číselníkových modulů dle /3/ umožnilo samostatné navrhování číselníků a jejich dodatečné /i dynamicky, tj. kdykoliv/ přápojení programům.

Jednou z významných vlastností jazyka číselníkových modulů je, že jej lze používat prakticky ve zcela shodné formě jak pro ruční, tak pro počítačový záznam příslušných informací. Zápis číselníků tímto jazykem, uložený ve zdrojové knihovně tak slouží jako víceúčelové informační rozhraní mezi analytikem, programátorem, počítačem a uživatelem.

Podstatou sestavení číselníku v jazyku číselníkových modulů je záznam kombinace vstupních a výstupních údajů, který je možno přitom napsat maximálně úsporně a tím i velice srozumitelně. Opakované údaje lze uvést na počátku odstavce pouze jedenkrát, číselné údaje je možno vymezovat intervalem od-do a současně i řádem /zleva i zprava/, z jednoho zdrojového textu lze pomocí parametrů vytvořit i více variant číselníku apod.

Číselník svým způsobem reprezentuje určitou datovou strukturu, takže je třeba v jeho záhlaví zaznamenat též jistou formu datového popisu pro převzetí těchto informací do adresáře dat.

Metasystémový jazyk číselníkových modulů umožňuje sestavo-

vat číselníky pro všechny běžné funkce, tj. dogazovací /texty, jmenovky/, kontrolní /přípustnost údajů a jejich kombinací, souvztažnosti/ i řídící /výběr algoritmů, větvení programu/ a tím výrazně přispívá k automatizaci analyticko-programátorské práce.

3.4 Realizační popisy

Realizační popisy jsou prvotními dokumentačními prvky, které doplňují funkční popisy příslušného automatizovaného procesu o definici konkrétního realizačního prostředí. Jejich cílem je vyčerpávajícím způsobem charakterizovat typy zařízení a konfiguraci výpočetní techniky, kvantifikovat a rozvrhnout do času zpracovávané objemy dat i postup realizace po jednotlivých technologických krocích a tak dokumentovat praktickou provozovatelnost navrhovaného řešení.

Realizační popisy jsou základním podkladem, který obsahuje všechny informace, potřebné pro přípravu, provedení a kontrolu určitého úseku zpracování v podmínkách konkrétního počítačového, provozního i uživatelského prostředí určitého výpočetního střediska včetně pořizování dat a expedice sestav.

Realizační popisy slouží jako hlavní prvotní doklad, předávaný z řešitelského pracoviště útvaru analýzy a programování do provozu VS jako dokument o předání určitého úseku zpracování do rutinní realizace.

Realizační popisy jsou zpracovávány pro úroveň systému, příp. subsystému, kde definují řazení chodů, ale především pro úlohu, tj. výpočtový chod, který je základní jednotkou souvislého počítačového zpracování.

Navrhovaným metasystémovým jazykem realizačních popisů je jazyk příkazových štítků pro řízení výpočtů/JCL/ pod operačním systémem příslušného počítače, doplněný analogicky s funkčními popisy o bloky P1 a P2 s důsledně formalizovanými dokumentač-

ními řádky. Obsah jednotlivých typů těchto řádků je navržen tak, aby z nich bylo možno automatizovaně odvodit souhrnné podklady pro realizaci. V bloku P3 realizačních popisů výpočtových chodů jsou pak běžné příkazy pro řazení jednotlivých programových kroků, včetně řešení případných incidentů. Detailní příkazy v krocích /tvorbu možno dle /6/ automatizovat/ jsou však již součástí dokumentace programových prvků - viz dále.

Pro decentralizovaná výpočetní nebo terminálová pracoviště, obeluhovaná přímo uživateli zpracovávaných informací, mohou být výše popisované informace zařazené přímo do uživatelských příruček.

3.5 Uživatelské příručky

Uživatelské příručky jsou prvotními dokumentačními prvky, které shrnují průřezovým způsobem informace o určitém úseku počítačového zpracování a poskytují je uživateli tohoto zpracování.

Hlavní podmínkou jejich praktické použitelnosti tedy je zachování srozumitelnosti ve vztahu k uživateli, tj. především k odborným útvarům. To znamená, že v nich sice může být použita odborná terminologie z řešené problémové oblasti, ale speciální výrazy z oblasti automatizace by se v nich neměly vyskytovat vůbec.

Cílem uživatelských příruček je poskytnout návod ke spolupráci s automatizovanou oblastí zpracování dat a s provozem výpočetního střediska. Mohou však být vytvářeny i pro využívání decentralizovaných výpočetních nebo terminálových pracovišť, případně pro ovládání standardních programových prostředků ve VS.

Významnou funkcí uživatelských příruček je, že vlastně potvrzují pravidla příslušného počítačového zpracování, a tím se stávají oficiálními dokumenty informačního rozhraní mezi zada-

vatelem /uživatelé/ a řešitelem /provozovatelem/ příslušné části ASŘ.

Metasystémovým jazykem uživatelských příruček je především přirozený hovorový jazyk. V návaznosti na členění funkčních popisů jsou kromě identifikace v bloku P1 formalizovaně navrhovány pouze řádky seznamů vstupních dokladů, výstupních sestav a hlavních počítačových kartoték v bloku I/O charakteristiky P2 a přehledy hlavních výpočtových chodů, číselníků, příp. dalších souvisejících prvků v bloku P3.

Jádrem uživatelských příruček je popis zásad počítačového zpracování v bloku P4. Obsahuje popis průběhu zpracování, pokyny ke kompletaci dokladů, informaci o prováděných kontrolách včetně přehledu významných chyb, které ovlivňují další zpracování, stručné popisy všech vstupních dokladů a výstupních sestav, číselníků a všech ostatních potřebných algoritmů, které by se mohly stát předmětem nedorozumnění při metodické prověře správnosti a úplnosti počítačového zpracování.

Přílohou uživatelských příruček jsou vzory a datové popisy dokladů, počítačem tištěné vzory sestav a především číselníky.

3.6 Programové prvky

Za programové prvky lze považovat všechny součásti projektové dokumentace, psané v některém z běžně používaných programovacích jazyků, včetně jazyka příkazových štítků pro řízení výpočtů. Jedná se o programy, podprogramy, makra, procedury atd.

Metasystémovými jazyky programových prvků jsou tedy programovací jazyky, JCL a manipulační jazyky databankových systémů, ale i pravidla pro ovládání parametrických programů, formule zadání pro generátory programů, anebo komunikační prostředky k práci s terminály.

Metodika vlastní tvorby programových prvků a používání

uvedených metasystémových jazyků je popisována např. v /4/, /5/, /6/ a systematicky je jí věnována pozornost ve sbornících /9/. Nejstručnější shrnutí zásad moderního programování na bázi systémového přístupu je asi toto:

Programové prvky je třeba konstruovat tak, aby byly čitelné a srozumitelné. Potom je možno provádět kontrolu jejich logiky, uplatnit dělbu práce a výrazně snížit celkové náklady na vývoj a údržbu. Metodou je přístup k řešení po různých úrovních podrobnosti, bloková struktura a využívání pouze logických schémat postupného, alternativního a opakovaného výskytu.

Při volbě metasystémového jazyka k definování programových prvků je nutno rozlišovat účel, ke kterému má dokumentace sloužit:

- pro prvotní vývoj a údržbu programového vybavení po dobu jeho životnosti jsou nezbytné zdrojové texty
- k informaci o funkcích, které programový prvek zajišťuje i o způsobu jeho spuštění, včetně zadávání příp. parametrů zcela vyhovuje funkční popis, který by tedy měl být postačujícím dokumentem i pro převzetí prvku do jiné organizace.

Dokumentace jednotlivých programových prvků formou funkčních popisů vytváří jeden ze základních předpokladů pro modulární skladbu programového vybavení. Tento přístup má značný význam nejen pro specifikaci funkcí a tvorbu příslušného modulu, ale při dodržení příslušných zásad modularity umožňuje též jeho opakované samostatné využívání.

3.7 Charakteristika podmínek řešení

Pracoviště, která se na různých organizačních úrovních zabývají vývojem projektů ASŘ pracují s různou technikou, mají zpravidla své vlastní dokumentační zvyklosti a nacházejí se na různém vývojovém stupni zavádění počítačového zpracování.

V úvodu projektové dokumentace je proto nutno zařadit souhrn hlavních podmínek, pro které je řešena. Tato charakteristika je zaměřena na popis způsobu identifikace dokumentačních a programových prvků, formu uložení dokumentace, používané programovací a vůbec metasystémové jazyky, úroveň interní normalizace, stupeň využívání automatizace v analyticko-programátorské práci a při řízení provozu i výpočetního střediska jako celku, přehled vybraných konvencí atd.

Cílem při zpracování charakteristiky podmínek řešení je poskytnout ve stručnosti veškeré informace, které jsou pro příslušné řešitelské pracoviště individuální, a proto nutné k orientaci v dalších kapitolách projektové dokumentace. Metasystémovým jazykem charakteristiky podmínek řešení je přirozený hovorový jazyk, avšak může být doplněna přílohami např. v podobě číselníků a pod.

Pokud jde o vybavení výpočetní technikou, měla by v této kapitole být pouze ta část informací, které charakterizují práci ve výpočetním středisku z hlediska vývoje a provozu, jako je operační systém, způsob práce s knihovnamí, interaktivní lašící prostředky a pod. Systematický přehled o typech používané výpočetní techniky, konfiguraci a dalších podmínkách k rutinní realizaci zpracování dat pro konkrétního uživatele souvisí především s kvantitativními ukazateli, a proto je obsažen v realizačním popisu systému, příp. jeho částí.

Výhledovým cílem by mělo být uplatnění alespoň částečné normalizace do projektování a realizace ASŘ, tak jak je tomu v jiných oblastech inženýrských činností, a k tomu snažení směřuje vlastně i celý tento příspěvek. Avšak vzhledem k mimořádně

rychlému rozvoji oboru a zcela roztráštěné technice toto není zatím reálné. Proto vhodně sestavená charakteristika podmínek řešení, s příp. detailními popisy metodik v příloze, je nejen významným faktorem interní racionalizace práce, ale především ovlivňuje opakovanou využitelnost projektů v dalších organizacích.

4. Počítačová dokumentace

Již samotné uložení projektové dokumentace na počítačová média přináší řadu výhod, známých ze zpracování dat. Jednou pořízené informace je možno zpravidla již s pomocí standardních editačních prostředků aktualizovat a pružně v libovolných výběrech poskytovat i širšímu okruhu zájemců.

Při využití výše poskytovaných metasystémových jazyků pro záznam prvotních dokumentačních prvků pak lze i poměrně jednoduchým programovým vybavením získat odvozenou dokumentaci řádově vyšší vyhovující schopnosti a kvality:

- nejjednodušší formou automatizace adresáře dat je výběr jednotlivých definičních a komentářových řádků z datových popisů a jejich seřídění dle identifikátorů údajů a jmen dokumentačních prvků výskytu
- přehled funkcí a vazeb a tím strukturu systému nebo jeho části včetně datových toků po souborech získáme výběrem řádků z formalizovaných bloků P1, P2, P3 ve funkčních a realizačních popisech a jejich seřídění dle jmen prvků, dekompozičních kódů I, O, W a výskytu
- úpravou formalizovaných řádků z bloků P1, P2, P3 realizačních a funkčních popisů, jejich rozmnožením a seříděním vzniknou libovolně podrobné a z různých hledisek seřazené souhrnné podklady pro realizaci, poskytující komplexní informace pro řízení, provádění a kontrolu pro-

vozní realizace systému nebo jeho části, včetně automatizovaně vytvářených průvodek pro zadávání a kontrolu výpočtových chodů.

Sestavování textových částí počítačové dokumentace má však též některé zvláštnosti: Na stránku tištěnou počítačem je totiž možno umístit až 80 řádků po 80 znacích, zatímco obdobná stránka z psacího stroje obsahuje 30 řádků po 60 znacích. Z tohoto zhuštění informací, chybějících háčeků a čárek a jednotvárnosti písmen vyplývají značně vyšší nároky na čtení souvislých textů.

Proto je potřebné volit kratší a jednodušší věty, spojované do odstavců menšího rozsahu, mezi kterými pak jsou výraznější mezery.

5. Organizace a dělba práce

Zavedení formalizovaných metasystémových jazyků do všech podstatných částí projektové dokumentace a její komplexní uložení do počítače představuje významný krok k jednoznačné průmyslové technologii analyticko-programátorských prací i k posouvání možností přebírání programových produktů. Současně se však jedná o průlom do vžitých tradic a s tím souvisí i řada problémů.

Prvním obtížným momentem je vůbec získání vývojových pracovníků VS pro roli uživatele ASŘ, tj. pro respektování pravidel počítačového zpracování. Prakticky všechny jejich výsledky se totiž stávají hromadně zpracovávanými daty, se všemi důsledky na termíny a kvalitu vstupů, znalost souvislostí počítačového zpracování těchto dat i na kvalifikované využívání výstupů.

Forma dokumentace funkčních a realizačních popisů přitom nedovoluje jiný, než značně zhuštěný a zcela jednoznačný záznam definic, celikého množství informací a jejich vztahů. Rovněž definování výslovných a datových popisů mimo vlastní program

vyžaduje důsledné vyjasnění všech detailů, týkajících se jednotlivých údajů. Tvorba uživatelských příruček pak znamená nejen utříděnou rekapitulaci celého zpracování, ale i nutnost vystihnout přesně a srozumitelně volným textem všechna důležitá fakta.

Výše uvedené výsledky je přitom třeba zpracovat písemně, resp. uložit v počítači nebo předložit ke kontrole či oponentuře ještě před zahájením vlastního programování. Tato pravidla sice obecně platila již dříve, ale nyní na dodržování jednotlivých kroků této pomoci metasystémových jazyků podstatně přesněji definované technologie vyvíjí silný tlak přímo počítač.

V této souvislosti se významně princíp aktivních dokumentačních prvků, tj. takových, které přímo podmiňují možnost vykonání dalších činností. Např. bez datových popisů archivovaných souborů a číselníků nelze ladit program, bez realizačních popisů nelze vytvořit zadávací průvodku ani pro ověřovací práce, chybějící funkční popis signalizuje neoprávněné čerpání strojového času apod.

Podklady a literatura

- /1/ Vyhláška č. 57 Státní komise pro vědeckotechnický a investiční rozvoj ze dne 18. srpna 1986 o budování automatizovaných systémů řízení, Sb. zákonů č. 19/1986
- /2/ Langefors B., Teoretická analýza informačních systémů, Alfa Bratislava, 1981
- /3/ Jiříček P. a kolektiv, Uživatelské příručky k překladači číselníkových modulů, CKR Ostrava k.ú.o. ASŘ 1976-1983, varianta pro DOS4 - VS Ingstav n.p. Brno 1984
- /4/ Čevela V. a kolektiv, Základy technologie programování v jazyku Cobol, DT ČSVTS Pardubice 1981
- /5/ Čevela V., Systémový přístup a logická stavba programu, Sborník Programování 1976, DT ČSVTS Ostrava
- /6/ -, Generátor programů Cobol a procedur DOS4, Sborník Programování 1985, DT ČSVTS Ostrava
- /7/ -, Funkční a realizační popisy v dokumentaci programového vybavení, Sborník Programování 1986, DT ČSVTS Ostrava
- /8/ -, Návrh jednotné automatizovaně vedené projektové dokumentace ASŘ, VS Ingstav Brno 1986 pro URS Praha a MSV ČSR
- /9/ Sborníky ze semináře Programování DT ČSVTS Ostrava
- /10/ Vyhláška č. 173/1980 Sb. o jednotné údajové základně organizací a související výnosy PSÚ, Příloha Hospodářských novin č. 24/1983 - Racionalizace informační soustavy organizací