

Ing. Jiří TURJANICA

PVT Praha - Vývojové středisko Brno

## METODIKA PROGRAMOVÁNÍ A DOKUMENTACE PROVÁDĚCÍHO PROJEKTU PŘI BUDOVÁNÍ AIS NV

Automatizovaný informační systém národních výborů, AIS NV, je projekt, na jehož tvorbě se zúčastňují projektanti a programátoři prakticky všech výpočetních středisek Podniku výpočetní techniky. Programy jednotlivých podsystémů jsou rutinně zpracovávány v celé síti výpočetních středisek PVT. Vzhledem k těmto skutečnostem je nutno AIS NV projekčně i programově řešit kvalitativně jinak než projekty, které vznikly pouze v rámci jednoho nebo omezeného počtu výpočetních středisek a které byly pouze v těchto střediscích zpracovávány, to znamená v dosahu autorského týmu. Tento kvalitativní rozdíl je velmi podstatný a dotýká se všech stadií tvorby projektu. Ve svém příspěvku se vzhledem k jeho možnému rozsahu zaměřím pouze na oblast tvorby a dokumentace prováděcího projektu. Právě v této oblasti se velmi výrazně projevují některé specifické rysy tvorby celopodnikových projektů. Je to především:

1. Na tvorbě programů jednotlivých podsystémů se v řadě případů podílí programátoři více výpočetních středisek. Vývoj podsystému není ukončen předáním do rutinního zpracování, ale pokračuje dále a další rozvoj podsystému sice zajišťuje též autorský tým, ovšem jen vyjimečně ve stejném a trvalém složení.
2. Rutinní zpracování jednotlivých podsystémů AIS NV probíhá v celé síti výpočetních středisek PVT, tedy na území celé ČSR /v současné době se projekt začíná rozšiřovat i na SRN/. Provozy jednotlivých výpočetních středisek nemají možnost přímé komunikace s autorskými týmy rutinně zpracovávaných podsystémů. Většina prací je přísně termínovaná a v řadě případů na sebe navazují zpracování různých úrovní /okres, kraj, republika/.

Z těchto skutečností vyplývají tyto požadavky na tvorbu programů a dokumentaci prováděcího projektu:

ad 1/ Hlavním kritériem na program není optimálnost algoritmu z hlediska rychlosti výpočtu, požadavku na paměť a podobně, ale přehlednost a snadná pochopitelnost zdrojového textu.

ad 2/ Maximální provozní spolehlivost a jednoduchost rutinního zpracování.

Účinnými prostředky, které umožňují tyto požadavky splnit, je metodika programování a metodika zpracování dokumentace prováděcího projektu platná pro všechny pracovníky, kteří se podílejí na tvorbě systému. Naše zkušenost prokázala, že naprosto nepostačuje metodiky napsat a vyhlásit jejich platnost příkazem ředitele. Proto, aby metodika byla akceptována jednotlivými pracovníky a uplatněna v praxi je především nutno:

- vydat metodiku jako ucelený text řešící problém komplexně v podstatné části jeho rozsahu. Velmi neúčinné je postupné vydávání jednotlivých metodických pokynů řešících dílčí problémy
- pracovníci musí mít metodiku k dispozici v potřebném počtu výtisků
- nejlépe všechny nebo alespoň podstatnou část pracovníků je nutno s metodikou seznámit formou školení. Školení by měl vést autor nebo jeho spolupracovníci. Přednášky mají být zaměřeny na vysvětlení a zdůvodnění, proč byla metodika řešena právě tímto způsobem. Cílem školení je, aby posluchači pochopili a akceptovali principy metodikou předpisované
- při vytváření prvních projektů a jejich dokumentací dle metodiky je nutný trvalý kontakt autora metodiky s řešitelskými týmy, aby se v praxi upřesnil způsob aplikace metodiky
- důsledná kontrola v průběhu vytváření programů a dokumentace je podstatně účinnější než spor při oponentním řízení
- text metodiky lze dle potřeby a nových požadavků průběžně rozšiřovat, nelze ho však podstatně měnit. K podstatné změně textu lze přistoupit až po důsledném ověření stávající

metodiky prací. Metodika programování AIS NV je v platnosti od června 1977 a v současné době, t.j. po dvou letech, začínáme získávat první praktické zkušenosti z rutinního provozu dle metodiky vytvořených

- v případě, že pro určitý podsystém je některá zásada stanovená v metodice prokazatelně vhodná, je účelnější pro tento případ volit vyjimku, než ukvapeně měnit metodiku.

Metodiku programování i metodiku dokumentace prováděcího projektu je nutno vytvářet ve vzájemném souladu. Oba materiály na sebe musí těsně navazovat. Jen tak mohou být účelné a plnit svoji funkci. Pro metodiku dokumentace prováděcího projektu je nutné, aby prováděcí projekt byl programován dle jednoznačné metodiky programování. Jedině tak může být metodika dokumentace konkrétní a tím i napsaná dokumentace funkční. Pokud metodika dokumentace nemůže vycházet z jednoznačného a konkrétního způsobu pracovního zabezpečení, řízení zpracování a jednotného programovacího stylu, musí být příliš obecná. Důsledkem toho jsou potom dokumentace které obsahují vždy něco jiného dle toho, jak si autor tyto obecně formulované požadavky vysvětlil. Vždy je to však dokumentace neúměrně rozsáhlá, která se velmi těžko vytváří, a kterou většinou nikdo nečte.

Metodika programování je velmi úzce vázaná na operační systém počítače, na kterém se budou projekty zpracovávat. V případě AIS NV je to MOS, operační systém počítačů EC 1021. Ignorovat vlastnosti operačního systému nelze. Naopak specifických vlastností operačního systému je třeba výhodně využít. Pokus o napsání metodiky programování, která by platila pro programy psané a provozované v různých operačních systémech, vede k tomu, že metodika nemůže být dostatečně konkrétní, autor má příliš velkou možnost volby a kvalita celkového řešení je velmi závislá na jeho zkušenosti. A to je v příkrém rozporu s posláním metodiky, která by měla zabezpečit efektivní a kvalitní řešení i v případě málo zkušených autorů.

Metodika programování AIS NV řeší dva základní okruhy problémů:

- vytváření programů
- provozní zabezpečení rutinního zpracování projektu.

V prvním řešeném tématickém okruhu metodika povinně předepisuje používání:

- programovacího jazyka COBOL
- metodiky normovaného programování
- zdrojové knihovny pro archivaci programů a jednotnou deklaraci.

Všechny tyto zásady dohromady zabezpečují přehlednost a čitelnost zdrojového textu a jsou dobrým předpokladem pro jednotnou a účelnou programovou dokumentaci.

Programovací jazyk COBOL byl zvolen především proto, že je nejrozšířenějším jazykem v PVT. Praxe prokazuje, že toto kritérium je nejdůležitější. Zároveň však splňuje i další věcná kritéria, jako je vhodnost pro zpracování hromadných dat, dobrá čitelnost a přehlednost zdrojového textu, možnost využívání zdrojové knihovny apod.

Metoda normovaného programování zabezpečuje, že i programy úplných začátečníků budou mít dobrou úroveň. Praxe prokazuje, že dobrý programátor po několika letech práce většinou vymyslí jakousi svoji metodu, která je vždy velmi blízká metodě normovaného programování. Těchto několik let hledání a omylů chceme metodika ušetřit a dosáhnout plného výkonu nezkoušených programátorů podstatně dříve.

Neméně významná je metoda normovaného programování pro přehlednost zdrojového textu. Zpracováním jednotné aplikace této metody při programování AIS NV jsme dosáhli toho, že zdrojové texty jsou dobře čitelné a pochopitelné nejen pro autora.

Používání zdrojové knihovny pro archivaci programů je běžné. Metodika však předepisuje využívání zdrojové knihovny i pro jednotné deklarace, především pro jednotnou deklaraci logických vět souborů. Tím, že každá věta je deklarována pouze jednou, jsou vyloučeny chyby při ladění, je zabezpečeno jednotné promítnutí případných změn do všech programů a

je velmi usnadněno dokumentování dat v programové dokumentaci.

Provozní spolehlivost přímo určuje úspěšnost celého projektu. Provozní spolehlivost je však málo závislá na provozních pracovnících. Ti projekt přebírají již hotový. Úroveň provozní spolehlivosti určuje řešitelský tým prováděcího projektu. To jsou většinou lidé, kteří mají relativně málo provozních zkušeností. V řadě případů jejich snaha nevede k cíli a vytvořené projekty se jen velmi obtížně rutinně zpracovávají. Z těchto důvodů musí metodika programování jednoznačně předpisovat, jak mají být zabezpečena rutinní zpracování projektů. Cílem je, aby se všechny podsystémy AIS NV zpracovávaly obdobným způsobem, a aby byla vždy zabezpečena určitá standardní úroveň kontrol a spolehlivost. Důsledkem tohoto jednotného řešení aplikovaného ve všech podsystémech AIS NV je rychlé a snadné přejímání nových podsystémů do rutinního zpracování, protože pro provozní pracovníky neznamenaají studium nových ale aplikací již známých a prověřených pracovních postupů.

Metodika programování rozčleňuje zpracování na etapy, etapy na chody a chody na kroky. Tyto části zpracování jsou definovány takto:

- a/ Etapa zpracování je část zpracování, která se provádí s určitou periodicitou, např. denní zpracování, měsíční zpracování apod., nebo jednorázově či neperiodicky s povně stanoveným technologickým režimem, např. oprava číselníku, inventurní zpracování apod.
- b/ chod je část etapy zpracování, která může proběhnout nepřetržitě na určité konfiguraci počítače bez nutnosti její změny nebo rozšíření a bez nutnosti lidského rozhodnutí o dalším postupu. Přenos dat mezi chody se děje výhradně prostřednictvím archivních paměťových médií.

c) Krok je část chodu, za jehož začátek je možné se vrátit při chybě výpočetního systému nebo obsluhy počítače.

Na základě tohoto členění zpracování je vybudována metodika řízení zpracování provozní dokumentace, využívání médií apod.

Zpracování chodu je řízeno výhradně pomocí procedur. Pro vytváření procedur platí tyto hlavní zásady:

- každému kroku úlohy odpovídá minimálně jedna procedura. Pokud krok obsahuje program, který vytváří body opakování, je restart tohoto programu vytvářen procedurou pro restart. Pro samostatné odstartování kroku /přerušeni zpracování chodu z provozních důvodů nebo spuštění kroku po havárii v job-stepu, který neobsahuje checkpoint/ se start provádí procedurou pro samostatný start kroku. Pro vytváření jednotlivých druhů procedur platí:
- v procedurách je povoleno modifikovat prázdný příkaz NOP na UPSI pro nastavení UPSI-byte
- operátor dává příkaz k vyvolání procedury pro zpracování prvního kroku chodu. Vyvolání procedury pro zpracování dalšího kroku je vždy zajištěno automaticky procedurou pro zpracování předchozího kroku
- modifikaci provádí operátor vždy pouze jednou a to při vyvolání první procedury bez ohledu na to, zda jsou modifikované příkazy použity již v tomto nebo až v dalších krocích chodu. Modifikace se do dalších automaticky volených procedur přenáší pomocí tzv. přenášečí procedury.

Velmi důležité je zabezpečit přiřazení symbolického zařízení /SYSnnn/ správnému fyzickému zařízení, t.j. zařízení, v němž je založeno médium s požadovaným souborem. Toto přiřazení nesmí být závislé na operátorovi a v případě založení špatného média do mechaniky nesmí zpracování pokračovat. Jedině tak lze vyhnout velmi závažné a zároveň velmi časté chyby obsluhy počítače, zpracování se starou verzí souboru, přepsání souboru. Metodika programování tento problém řeší tak, že předpisuje pro každé médium unikátní jméno cívky,

svazku /VOL1-návěští/. Přiřazování se povinně děje pro magnetické disky monitorským příkazem //ASSGN SYSnnn,VOL=jméno a pro magnetické pásky programem ASVOL, rovněž dle jména ve VOL1-návěští. Jméno VOL1-návěští je tvořeno tak, že první tři znaky určují zakázku v rámci PVT, čtvrtý znak použití média v rámci zakázky /např. páska pro uchovávání změn/. Pokud se na stejné využívané médium nahrávají soubory, u nichž musíme rozlišovat generaci, používají se fyzicky různá média, která se liší v pátém znaku VOL1-návěští. Pro rozlišení uživatele slouží šestý znak /případ, že jedno středisko zabezpečuje zpracování např. pro několik okresů-uživatelů/. Obdobně jsou vytvářena jména chodů, kroků a jména procedur. Jestliže operátor zadá příkaz pro zpracování chodu pro určitého uživatele a období /např. sudý měsíc - pro zpracování, které rozlišuje dvě generace souborů/, přiřadí příslušná procedura správná média se starým i novým stavem naprosto nezávisle na operátorovi.

Protože operační systém MOS nezná standardní uživatelské návěští /UHL, UTL/, jsou metodikou předepsány jako povinná součást souborů dat. Pro každou větu jsou povinně definovány první tři byty. První byte má význam typu věty a obsahuje:

- 0 - věta má význam počátečního standardního uživatelského návěští
- 5 - věta má význam věty souboru dat
- 9 - věta má význam koncového standardního uživatelského návěští

Další dva byty mají význam druhu věty a programátor je může využít k rozlišení vět v rámci souboru. Vyhrazeny jsou pouze tyto druhy vět:

- 00 - prázdná věta
- 01-08 - věty počátečního standardního uživatelského návěští

Počáteční návěští nemá pevně definovaný obsah, metodika doporučuje jeho využívání především dvěma způsoby:

- pro uchování informace zadané při zahájení zpracování po celé zpracování, neboť takto uchovaná informace prochází jako součást souboru dat celým zpracováním /např. období zpracování, název uživatele apod./

- pro uložení upřesňujících informací o souboru /např. jméno programu, datum a hodina vytvoření souboru/. Tyto informace mohou být velmi cenné při řešení havarijního stavu.

Vhodným využitím počátečního návěští lze ušetřit celou řadu operátorských zásahů a průběh zpracování automatizovat.

Koncové návěští umožňuje zabezpečit důslednou kontrolu úplnosti souborů. Praxe prokázala, že kontrola zajištěná operačním systémem není postačující, a proto metodika předpisuje při zápisu souboru vytvořit minimálně dvě kontrolní čísla; jedno jako počet datových vět souboru, druhé jako suma dat souboru /např. suma přes položku, binární součet binárních součtů všech vět souboru apod./. Tato kontrolní čísla jsou součástí koncového návěští. Pro usnadnění programování těchto uživatelských návěští jsou k dispozici moduly, které zabezpečují jejich vytváření a zápis, vytváření kontrolních čísel a kontrola úplnosti u vstupních souborů.

Pro usnadnění činnosti operátora byly definovány čtyři kategorie zpráv a byl sjednán formát zpráv. Druh zprávy tak do značné míry určuje reakci operátora na zprávu. Tím se především zrychluje obsluha programu operátorem.

Toto jsou pouze stručně uvedené některé hlavní oblasti, kterými se metodika zabývá. Rozsah řešené problematiky je podstatně větší.

Na základě metodiky programování byla zpracována metodika dokumentace prováděcího projektu. Tuto dokumentaci tvoří:

- provozní dokumentace
- programová dokumentace
- uživatelská příručka

Všechny tyto dokumentace jsou řešeny jako společná dokumentace prováděcího projektu. To se projevilo především takto:

- bylo vyloučeno duplicitní dokumentování stejných informací v různých dokumentacích
- v maximálně možné míře byla sjednocena formální úprava
- jsou využívány stejné formuláře.



Na provozní dokumentaci jsou kladeny dva požadavky. Jednak musí být velmi podrobná, aby umožnila zpracování i bez účasti autorského týmu, jednak musí být stručná a úplná, aby ji bylo možno přímo využívat jako pracovní pomůcku. Tento rozpor jsme řešili tím, že jsme dokumentaci rozdělili do pěti částí:

- organizační část
- pokyny pro pořizení vstupních dat
- pokyny pro vstupní a výstupní kontrolu
- pokyny pro operátorskou obsluhu počítače
- provozní programátorská část

Druhá, třetí a čtvrtá část provozní dokumentace jsou velmi stručné a přehledné. Jsou využívány přímo provozními pracovníky při rutinním zpracování projektu. Organizační část je určena vedoucím pracovníkům provozu a obsahuje celkové provozní informace o projektu, popisuje způsob zavedení rutinního zpracování ve středisku a dává podklady pro plánování a řízení rutinního zpracování ve výpočetním středisku jako celku, a to nejen z hlediska návaznosti činnosti jednotlivých oddělení provozu, ale i včetně potřebných časových kalkulací. Provozně programátorská část obsahuje všechny údaje, které jsou zapotřebí k řešení havarijních stavů a k úplnému pochopení režimu zpracování projektu. Provozní dokumentaci obdrží každé výpočetní středisko ve dvou vyhotoveních. Jedno má k dispozici provozní programátor, druhé se rozdělí jednotlivým oddělením provozu.

Vzhledem k tomu, že programy projektu udržuje výhradně pouze autorský tým, vytváří a udržuje se programová dokumentace pouze v jednom vyhotovení pro jeho potřebu. Provozní dokumentace je tvořena těmito částmi:

- obecná část
- popis programových prvků
- popis dat

Vzhledem k tomu, že metodika programování předpisuje takový způsob tvorby programu, že zdrojový text je velmi přehledný a snadno pochopitelný, je převážná část programové dokumentace tvořena zdrojovým textem s výjimkou obecné části, kterou je nutno napsat. Pro ostatní části platí, že rozsah jednotlivých kapitol a odstavců je závislý na složitosti programu a na podrobnosti komentářů uvedených ve zdrojovém programu.

Data jsou v programové dokumentaci dokumentována tak, že v rámci popisu programu je uvedeno jméno DTF /externí jméno souboru/ a jméno souboru dat /jméno v HDAL, ve VTOC/. V části popis dat je v kapitole popis souborů dat pro každé jméno souboru dat uvedeno třídění souboru a soupis všech druhů vět v souboru. Každá věta je potom popsána v kapitole popis vět. Protože všechny věty jsou deklarovány jako knihy ve zdrojové knihovně, je tato kapitola tvořena výpisy těchto knih. Tímto způsobem dokumentace dat jsou vyloučeny duplicity a v případě postupného vytváření jednotlivých skupin úloh lze vytvořit dokumentaci dat celého pod systému postupným rozšiřováním dokumentace dat první skupiny úloh, neboť řada vět a souborů je společná celému pod systému.

Problematika dokumentace prováděcího projektu je příliš rozsáhlá, než aby ji bylo možno v tomto omezeném příspěvku podrobně probrat. Doufám však, že i z uvedených faktů vyplývá výhodnost řešení problematiky prováděcího projektu jako celku.

Problematikou metodiky programování a metodiky projektové dokumentace se zabývají pracovníci řídicí skupiny hlavního projektanta AIS NV. Tato skupina, která je součástí vývojového střediska PVT v Brně, koordinuje a řídí projektování a programové práce na AIS NV. Vzhledem k tomu, že svoji činnost v oblasti metodiky zdaleka nepovažujeme za skončenou, uvítáme spolupráci všech pracovišť, které se touto problematikou rovněž zabývají.