Ing. Jaroslav Horyna

ÚVT TESLA

METODIKA TVORBY TYPOVÝCH PROGRAMOVÝCH PRVKŮ PRO ASŘP

Obsah:

1. Úvod.

Metodice projektování ASŘP se věnuje v poslední době zvýšená pozornost. Byly zpracovány celostátně platné metodické pokyny pro vytváření ASŘP a rovněž na mezinárodní úrovni v radě specialistů Al byly vypracovány návrhy na metodiku řešení typových prvků ASŘP (konkretně na bázi OS/JSEP). Cílem těchto materiálů je jednak formulovat požadavky na výsledné řešení ASŘP, jednak sjednotit pracovní postupy při projektování, určit etapy projektové přípravy, stanovit jednotné předpisy pro projektovou dokumentaci a obecně racionalizovat projektové práce. Projevuje se rovněž snaha vypracovat obecnou terminologii pro ASŘP, která by umožnila srozumitelnou a jednoznačnou komunikaci

jak mezi tvůrci tak i mezi uživateli vyprojektovaných systémů.

Zkušenost ukazuje, že téměř neexistuje případ, aby projekt ASŘP zpracovaný pro jeden podnik byl beze změny uplatněn i v jiném podniku třeba stejného typu. V tomto smyslu je každý ASŘ unikátní. Přesto ovšem mohou být i v těchto individualizovaných ASŘ uplatňovány standardní adaptabilní prvky, jež řeší typické, v jednotlivých aplikacích se opakující úlohy. Využití takových prvků v ASŘ výrazně snižuje náklady na projekt i zavedení ASŘ, šetří kapacity projektantů a zkracuje čas na projekt i jeho zavedení. Míra využití takových typových prvků by měla sloužit k hodnocení projektantů a řídících pracovníků zodpovědných za přípravu a zavedení ASŘ.

2. Definice typového prvku.

V metodických pokynech FMTIR se hovoří o t.zv. typových projektových řešeních. Metodické pokyny považují typová projektová řešení za jeden z racionalizačních prvků procesu vytváření ASŘP. Za typové projektové řešení se považuje modulární prvek ASŘP (jeho rozlišovací úroveň se v metodických pokynech nedefinuje), jehož funkce nezávisí na konkretní struktuře dat. To znamená, že typové projektové řešení představuje fungující modifikovatelný prvek automatizovaného systému řízení podniku, který je možné přizpůsobit (na základě modulární stavby tohoto prvku) konkretním podmínkám ASŘ různých podniků. Přitom se nevylučuje možnost existence určitých omezení použitelnosti typového projektového řešení.

V resortních prováděcích pokynech pro budování automatizovaných systémů řízení podniků resortu FMVS se za typové řešení považuje část systému určená k opakovanému použití. Mohou jím být:

- programové vybavení dodávané výrobcem počítače,
- systémy a programy zařazené do knihovny KOTO,

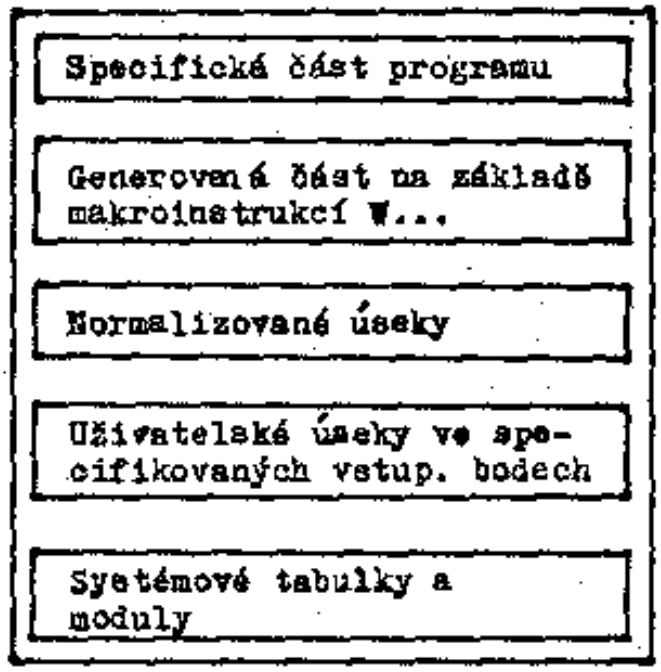
- aplikační programové vybavení vyvinuté pro využití v rámci resortu FMVS nebo VHJ.

Současně se uvádí, že typové řešení zvyšuje hospodárnost projektových prací.

Obecně lze za typový prvek považovat logicky a funkčně uzavřenou část systému zpracování ASŘ se zadanou oblastí využití, která obsahuje varianty alternativních řešení a je předurčena k mnohonásobnému využití.

Pro účely metodiky tvorby typových programových prvků ASŘ je vhodné tuto definici dále zúžit a upřesnit. V projektu ISŘT, ve kterém byla metodika typových prvků široce uplatněna, nazýváme typový prvek programového řešení normalizovaným úsekem a rozumíme pod tímto pojmem sekvenci instrukcí, podprogram, makrodefinici, modul nebo program, které jsou vytvořeny podle interních pracovních norem a systémových konvencí, s definovanou funkcí a s opakovatelným využitím.

Na základě takto definovaných a konstruovaných typových prvků programového řešení (normalizovaných úseků) lze vytvářet programové celky, jejichž struktura je znázorněna na obr. 1.



Obr. 1 - Struktura programů s využitím normalizovaných úseků.

Uvedená definice typových prvků umožňuje vytvářet typové prvky různých úrovní. Typové projektové řešení budované na konceptu typových prvků se pak skládá z hierarchie typových prvků. Struktura typových prvků typového projektového řešení je znázorněna na obr. 2.

Struktura typových prvků	
Technický projekt	Prováděcí projekt
Systém Subsystém Skupina úloh Úloha Výpočtový postup	Programový produkt Chod programů Program Úsek programu - normalizovaný - generovaný - specifický - uživatelský - systémový

Obr. 2 - Hierarchie typových prvků v ISŘT.

Jak mezi programy tak i mezi úseky se často realizuje ještě další hierarchický vztah, a to mezi programy resp. úseky řídícími a výkonnými. Soubor všech programů prováděcího projektu ISŘT tvoří programový produkt.

3. Vlastnosti typových prvků.

Typové prvky podle uvedené definice, ale i podle specifické programátorské definice musí mít určité vlastnosti, aby mohly být mezi typové prvky vůbec zařazeny.

Podle našeho názoru to jsou:

1) modularita
2) variabilita
3) adaptabilita
4) transparentnost
5) jednotná metodika tvorby
6) spolehlivost
7) efektivnost.

Modularita typových prvků je vlastnost, která umožňuje jejich uspořádání v různých kombinacích pro řešení různých úloh. Předpokládá takovou konstrukci typových prvků, která umožňuje jejich bezkonfliktní řazení do vyšších celků. Umožňuje budovat stavebnicové systémy a měnit je podle potřeby výměnou jejich prvků a vytvářet tak mnohotvárné varianty řešení.

Variabilita typových prvků je dána jejich funkční pružností. Jednotlivé typové prvky mohou plnit různé funkce podle svého začlenění do vyšších celků popř. podle své vnitřní struktury. Na př. úseky programu mohou být parametricky přizpůsobeny pro aplikace v ročním, operativním i denním plánování výroby nebo různou kombinací programů v programovém chodu lze dosáhnout variantních výstupů.

Adaptabilita umožňuje přizpůsobení typových prvků požadavkům a potřebám uživatele. Přizpůsobení může být jak funkční (změna algoritmů, postupů řešení) tak i obsahové (týká se především souborů dat, t.j. délky údajů, struktury vět, měřítek aritmetických hodnot, alokace souborů na různých mediích popř. jejich organizace). Přizpůsobení lze provádět na všech úrovních projektového řešení, na př. volbou a uspořádáním nižších typových prvků do vyšších celků, uvnitř jednotlivých úrovní v různých časových okamžicích implementace (u programů před kompilací, při překladu nebo

spojování a při provádění programu).

Transparentnost znamená přehlednost řešení a jeho srozumitelnost pro uživatele ale i tvůrce typového prvku. Transparentní řešení umožňuje snadnou údržbu a promítání požadovaných změn. Transparentní řešení je značně nezávislé na svém tvůrci, protože má být srozumitelné pro každého, kdo s ním pracuje. Je umožněno vhodnou technologií programování (na př. strukturované programování) a řadou pravidel, která upravují formální stránku zápisu programů.

Jednotná metodika tvorby je základním rysem celého řešení a je dána soustavou pravidel, interních norem a normalizovaných postupů, které určují celou filosofii řešení, jeho cíle a prostředky k jejich dosažení, pracovní terminologii a pracovní postupy. Sjednocují pracovní kolektiv v tvůrčí tým a umožňují jeho cílevědomou a efektivní činnost. Jsou významným činitelem v racionalizaci týmové práce.

Spolehlivost programu je třeba posuzovat ze dvou aspektů:

1) bezchybovost programu
2) funkční spolehlivost.

Bezchybovost programu je dána tím, že program plně odpovídá specifikaci a při jeho provádění nenastanou nedefinované stavy. Docílit této kvality u složitých programových systémů v průběhu ladění programu je někdy velmi obtížné a některé chyby se mohou objevit i při ověřování popř. rutinním provádění programu. Součástí řešení musí být tedy i metodika údržby programového produktu předaného uživatelům.

Funkční spolehlivost programu vyplývá do značné míry už z analýzy problému a je dána tím, že program vykazuje inteligentní, smysluplné chování i v mimořádných situacích při provádění (na př. při výskytu chyb v datech, nekonzistenci

souborů a pod.).

Efektivnost typových prvků má rovněž dvě stránky:

1) efektivnost tvorby, t.j. minimalizace pracnosti při vytváření prvku,

2) efektivnost při využívání, t.j. minimalizace strojního času.

Obě stránky působí do jisté míry protichůdně (programování v Assembleru je pracné, může však vést ke zkrácení prováděcích časů)

Uvedené vlastnosti typových prvků působí společně a většinou se i navzájem ovlivňují.

4. Metodické zásady a prostředky pro tvorbu typových prvků ISŘT.

V rámci projektu ISŘT řešeného jako státní úkol P04-124-034 byla vypracována pro dosažení požadovaných vlastností typových prvků metodika ve formě interních pracovních norem (IPN) a prostředky ve formě normalizovaných úseků (NÚ). Interní pracovní normy mají obecnou platnost a lze je v přiměřené míře aplikovat na jakékoliv programové řešení. Normalizované úseky jsou programovány v Assembleru a operačním systému DOS/EC 1.3.počítačů JSEP. Budou pochopitelně přizpůsobovány i pro vyšší verze operačního systému.

V souhrnu lze metodické zásady a prostředky pro tvorbu typových prvků shrnout do těchto bodů:

1) Pro programové řešení je využíván operační systém DOS/EC verze 1 modifikace 3. Funkce operačního systému nesmějí být měněny. Mohou být doplňovány vlastními programy ISŘT.

2) Programy jsou zpracovávány v jazyce Assembler, který poskytuje ve zvoleném řešení nejvíce prostředků pro realizaci

požadovaných vlastností výsledného programového produktu a všech jeho komponent.

3) V rozsáhlé míře je využíván makrojazyk Assembleru. Je nejdůležitějším nástrojem při generování programových úseků a prostředkem umožňujícím rozsáhlé přizpůsobení.

4) Struktura programů je modulární. Je využívána metoda strukturovaného programování přizpůsobená pro Assembler a doplněná řadou předpisů pro formální úpravu programů.

5) Postupně je vytvářena množina normalizovaných úseků. Její využívání přispívá ke sjednocování formální úpravy a struktury programů, ke sjednocování algoritmů, ke sjednocování komunikačních vstupů a výstupů. Práce programátorů se tím racionalizuje, snižují se kapacitní nároky na programování a náklady na ladění.

6) Jsou vypracovávány normalizované postupy řešení některých programových problémů, které formou předpisu anebo příkladu umožňují jednotné řešení (na př. redukce programových funkcí podle definice výstupního souboru, příklady využití makroinstrukcí, ostatních normalizovaných úseků a pod.).

7) Jsou vydávány, udržovány a doplňovány interní pracovní normy, závazné pro všechny programátory, které určují formální náležitosti programů, způsob evidence programů, způsoby tvorby návěští a názvů polí, normalizují názvy některých polí, obsahují zásady pro konstrukci programů, jež vylučují možné kolize při řazení úseků do programových celků, stanoví náležitosti a vzory programové dokumentace, používané formuláře, postupy ladění a oprav programů, sjednocují pracovní terminologii a pod.

5. Typové prvky jako komponenty projektu ISŘT.

Zásady, metodika a prostředky popsané v předchozích kapitolách byly uplatněny v řešení typových prvků v projektu ISŘT.

Systém.

Popis systému ISŘT je obsažen v technickém projektu systému. Systém ISŘT je koncipován pro podnik se středně až velkosériovou strojírenskou výrobou. Typovost řešení je dána obecností použitých algoritmů. Základní algoritmy jsou vybudovány na sekvenčním zpracování souborů a jejich vzájemném přiřazování. Tím se ISŘT liší od metod využívajících zřetězení souborů, především maticních (typ master - informace o součástech a pracovištích) a strukturních (typ subordinate - kusovníky, normy výkonové). Sekvenční algoritmy vynikají jednoduchostí a rychlostí, celkové doby výpočtů však do jisté míry závisejí na systémovém programu SORT, který je v DOS 1.3 vysoce efektivní.

Uživatel dostává se systémem seznamy maximálních variet symbolů, s kterými programy pracují. Má možnost vybrat podmnožinu maximální variety a tím přizpůsobit systém svým existujícím souborům dat, popř. svým požadavkům. Je pochopitelné, že požadované výstupy musí být kryty potřebnými vstupy. Uživatel má však možnost nadefinovat i soubory s vlastními uživatelskými údaji, jejichž zpracování pak musí zajistit vlastními programovými úseky. Pouhé přesuny mezi soubory jsou i pro uživatelské údaje většinou zajišťovány specifickým normalizovaným úsekem ÁØ7A.

Považujeme za pozitivní rys systému, že uživatel musí k jeho účinnému využívání přispět aktivně tím, že provede důkladnou analýzu svých informačních potřeb. Výsledkem této analýzy jsou uživatelské definice souborů systému ISŘT. Tyto definice se provádějí v tabulkách popisu souboru (TPS) v podstatě neprogramátorskými prostředky. Symbolické názvy údajů v ISŘT jsou pro uživatele závazné a v rámci projektu celého systému jednotné.

Celý systém má atributy typových prvků popsané v kapitole o vlastnostech typových prvků a je adaptabilní pro aplikace v různých podnicích popř. i v různých oborech.

Subsystém.

Volba subsystému v ISŘT je motivována pragmaticky z hlediska co nejrychlejšího využívání základních souborů dat (báze normativních dat), jejichž příprava a vytvoření má nutně prioritu před výpočetními chody.

Báze dat tvoří základní subsystém, do něhož jsou začleněny i projekty bází dat bezprostředně rozvíjející popř. ji rozšiřující (na př. kalkulace). Samotná báze dat se skládá z problémově orientovaných skupin souborů a může být vytvářena postupně podle zaváděných subsystémů. V současné verzi jde v podstatě o nezávislé soubory obhospodařované jednotnou skupinou programů pro tvorbu, údržbu a výpisy souborů dat.

V inovačních záměrech se předpokládá integrace souborů podle databankového konceptu.

Uživatel může vybrat pro vlastní využívání subsystémy podle svých potřeb a požadavků. Jednotlivé subsystémy jsou projektovány tak, že mohou být zaváděny postupně a po částech. Subsystém báze dat má ovšem vždy prioritu.

Skupina úloh - chod programů.

Skupina úloh řeší funkčně a logicky související činnosti (na př. aktualizace souborů dat, roční plán výroby - materiálu - nákladů a pod.). Skupina úloh je proměnný prvek subsystému. Vzniká kombinací jednotlivých úloh (programů) v podstatě podle uživatelovy volby. Technický projekt ISŘT předkládá jen příklad vzorového řešení, které je pro uživatele vodítkem.

Skupině úloh odpovídají v prováděcím projektu chody programů. Struktura chodu může být jakákoliv smysluplná kombinace programů (výpočetních, pomocných, univerzálních, systémových). Tak na př. zařazením třídícího programu, doplňovacího programu (doplňuje soubor až ze dvou dalších souborů), slučovacího

programu (načítá 1 až 252 hodnot podle 1 až 9 kriterií) a univerzálního tiskového programu lze produkovat variantní výstupy.

Adaptabilita skupiny úloh spočívá tedy především na volbě vyhovující struktury, t.j. na smysluplné kombinaci jednotlivých úloh (programů).

Úloha - program.

Úloha řeší zpravidla určitou činnost. V provádějícím projektu odpovídá úloha programu (programovému celku podle terminologie IPN). Adaptabilita úloh je tedy založena na adaptabilitě programů, jež je popsána v popisu metodiky programového řešení v IPN.

Programy jsou přizpůsobitelné zásadně ve čtyřech úrovních:

1) Před kompilací:
   Uživatel provádí dovolený výběr programových úseků, které zařazuje do kompilace. Připravuje vlastní programové úseky a struktury (tabulky popisu souborů a jiné), které do programu vkládá. Může rovněž eventuelně měnit sekvence instrukcí v programu.

2) Při kompilaci:
   Na základě vložených struktur a parametrů (podle popisu příkazových štítků pro generování programu) se generuje při kompilaci uživatelská verze programu z obecné verze uložené v privátní zdrojové knihovně. Volí se především struktury vět a délky údajů, bloků, měřítka aritmetických hodnot, kontroly vstupního souboru, alokace a dovolená organizace souboru.

3) Před výpočtem:
   Uživatel volí t.zv. dynamické parametry programu, kterými se vybírá varianta výpočtu, vkládají různé konstanty, určuje termín platnosti vět v souborech a pod. Parametry se zadávají parametrickými štítky zpravidla podle klíčových slov (VV=, DATUM=, PHI= atd), popř. řídícím štítkem

// UPSI. Výjimečně se připravují štítky hodnot (na př. seznam sledů s hodnotou posunu pro každý sled pro plánování výroby).

4) Při výpočtu:
Na základě hodnot ve vstupních nebo výstupních souborech se provádí automaticky aktivace výpočetních algoritmů, které jsou potřebné z hlediska požadovaných výstupů (výběr funkcí programu). V některých případech dochází k dynamické rekonfiguraci programu na základě vstupních parametrů nebo vstupních/výstupních souborů (na př. v přiřazovacím programu).

Programové celky lze rozdělit do těchto kategorií:

1) Výpočetní obecné programy: řeší konkretní úlohy z technického projektu, na př. výpočet operativního plánu výroby. Tyto programy se generují z obecné verze do uživatelské verze podle struktur uživatele.

2) Pomocné obecné programy: řeší obecné úlohy, generují se z obecné do uživatelské verze podle struktur uživatele. Příklad: slučovací program.

3) Univerzální programy: řeší obecné úlohy. Negenerují se, pracují podle vstupních parametrů. Příklad: doplňovací program, univerzální tiskový program.

4) Systémové programy: jsou součástí operačního systému. Příklad: SORT, MERGE, Utility a pod. Přizpůsobují se vesměs parametricky.

Pro uživatele je programový celek základním stavebním prvkem. Po přizpůsobení těchto prvků podle svých potřeb a požadavků může kombinovat výpočetní, pomocné a systémové programy do programových chodů k řešení různých skupin úloh a tak postupně budovat uživatelské subsystémy pro automatizované zpracování informací v automatizovaných systémech řízení.

Výpočetní postupy - programové úseky.

Volba výpočetních postupů realizovaných programovými úseky se dá provádět buď dynamicky v programu, do něhož jsou předem známé postupy zahrnuty, na základě parametrů nebo před kompilací volbou příslušných úseků. Výhoda prvního způsobu spočívá v tom, že změnou parametrů lze měnit i výpočetní postup bez nové kompilace. V současné verzi ISŘT tento způsob převládá.

Uživatel dostává se systémem jak obecné a univerzální programy tak i normalizované úseky ve formě zdrojových knih a relativních modulů. Ve zdrojové knihovně je řada obecně využitelných makrodefinic. Součástí dokumentace je i podrobný programátorský popis všech úseků. Pro programy v Assembleru může uživatel všech těchto programových prostředků a celou metodiku a výhodou využít, což podstatně sefektivní programování.

6. Postup při zavádění systému ISŘT.

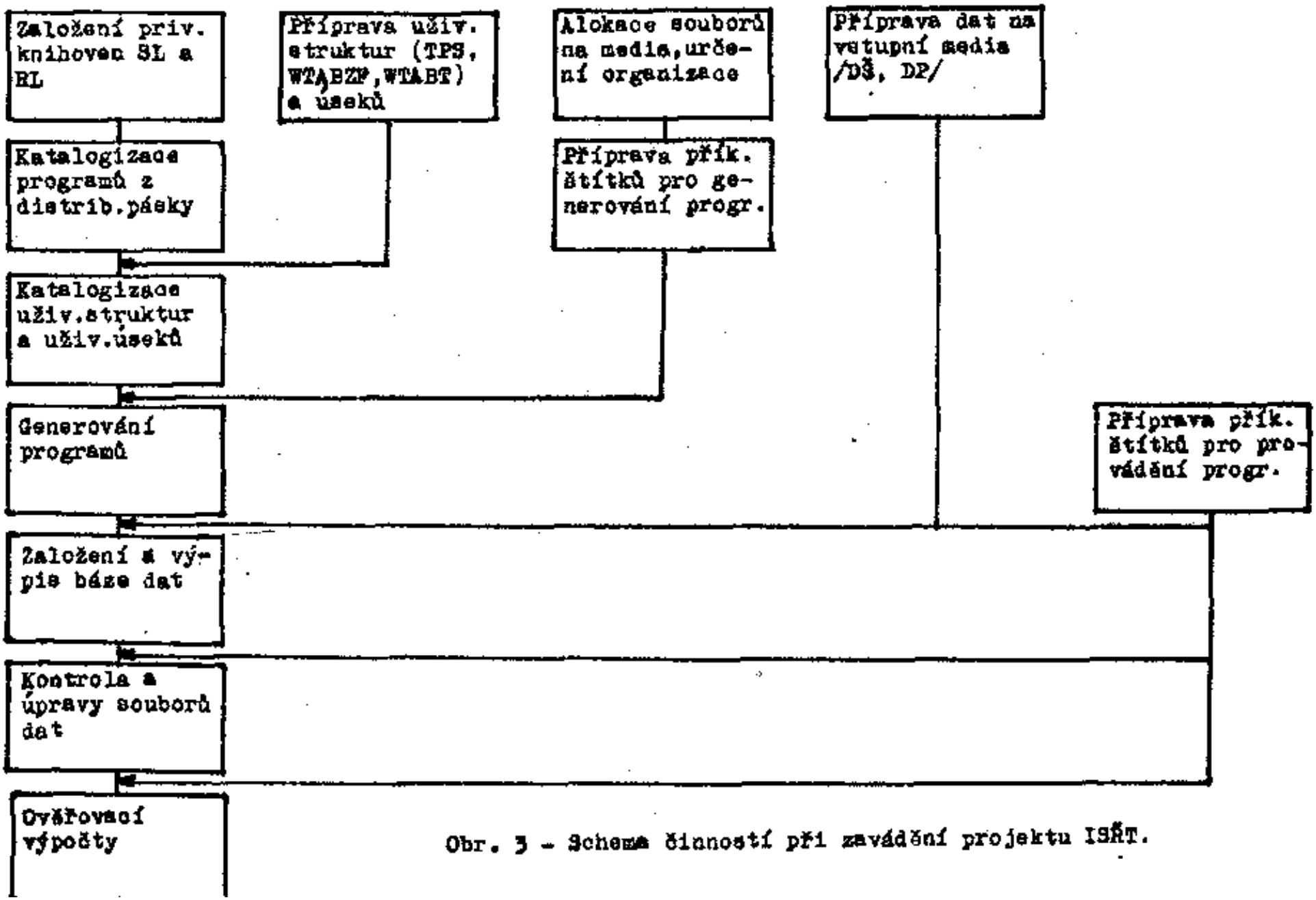
Metodice zavádění systému ISŘT jsou věnovány kapitoly v technickém projektu systému i v prováděcích projektech. Prováděcí projekty obsahují i vzory příkazových štítků pro založení systému (od inicializace diskového svazku přes vytvoření privátních knihoven až ke katalogizaci programů z distribuční magnetické pásky a uživatelských struktur) a generování jednotlivých programů. V podrobnostech odkazujeme na literaturu prováděcích projektů. V této kapitole zůstaneme na úrovni popisu a schematického znázornění. Studium projektu ISŘT si vyžádá značný čas, protože jde o rozsáhlý soubor dokumentů. Tak jako je možné postupné zavádění projektu, lze s určitým rizikem připustit i postupné seznamování s projektem. Filosofii řešení je však třeba pochopit již v samém začátku. Možnost přizpůsobování novým a postupným generováním systému čelí riziku nesystémového dílčího přístupu, při kterém mohou vznikat ne-

dostatky v definicích souborů, které by se mohly stát brzdou při pozdějších aplikacích dalších částí systému. Z toho též plyne, že definice souborů jsou nejdůležitější přípravnou činností uživatele při zavádění systému ISŘT/JSEP.

Seznámení s projektem usnadňují i kursy ISŘT, které jsou pravidelně organizovány a autorská pomoc tvůrců projektu.

Jednotlivé kroky při zavádění projektu po prostudování jeho obsahu a výběru subsystémů resp. skupin úloh jsou schematicky znázorněny na obr. 3.

Inovační záměry směřují k parametricky řízeným programům pro tvorbu a údržbu báze dat, k databankovému konceptu a k uplatnění operačního systému OS/EC v inovovaném řešení. Takto koncipovaný projekt by mohl být schopen vyplnit mezeru v aplikačním programovém vybavení na vyšší typy počítačů řady JSEP.



Obr. 3 - Schema činností při zavádění projektu ISŘT.