

# VOĽBA PROGRAMOVEJ KONCEPCIE PRI TVORBE ASRTP MONTÁŽNYCH PREVÁDZOK

Ing. Vladimír Borský  
VUMA Nové Mesto nad Váhom

## 1. Úvod

Cieľom príspevku je popísať koncepciu programového systému pre riadenie technologických procesov montážnych prevádzok, predovšetkým niektoré metodické postupy podporujúce jeho typovosť.

Uvedený programový systém je jedným z výstupov štátnej úlohy F15-119-408. Pôvodné zadanie predpokladalo jeho zavedenie v dvoch montážnych prevádzkach - montáž ističov v SEZ Krompáchy a montáž dosiek plošných spojov v Tesle Vráble. Počas projekčných prác bola predložená požiadavka na aplikáciu u ďalších používateľov. Z toho dôvodu stanovili riešitelia za jednu zo základných vlastností systému jeho typovosť. Pojem "typovosť" je potrebné chápať vo vzťahu k riešenej úlohe. To znamená, že snaha o typovosť je identická so snahou o minimalizáciu množstva práce potrebnej na aplikáciu programového systému u ďalších používateľov.

## 2. Pružnosť programového systému

Požiadavka typovosti programového systému priniesla dva hlavné problémy:

- a/ zabezpečenie diferenciácie funkcií riadiaceho systému u jednotlivých používateľov,
- b/ umožnenie práce s rozdielnymi dátovými základňami.

Riešenie prvého problému spočíva vo využití modularity pri tvorbe riadiaceho systému. Dekompozíciou požadovaných funkcií systému v etape analýzy a projektovania vznikli požiadavky na čiastkové funkcie, ktoré sa už ďalej nedelia. Každú takto definovanú čiastkovú funkciu zabezpečuje práve jeden aplikačný modul. Tento nie je, čo sa týka rozsahu, modulom v zmysle modulárneho

programovania. Dolná hranica jeho rozsahu je limitovaná požiadavkou nedeliteľnosti, čo v praxi znamená, že obsahuje ucelenú množinu operácií nad dátovým súborom. Pri znalosti potrieb riadenia montážnej prevádzky možno uvedené čiastkové funkcie, reprezentované aplikačnými modulmi, syntetizovať do požadovanej funkcie aplikačného programu, čím sa zabezpečí pružnosť programového systému vzhľadom na funkciu.

Uvedeným postupom však nemožno docieľiť pružnosť vzhľadom na dáta. Štruktúra dát na vstupe a výstupe aplikačného modulu a rozsah jednotlivých údajov v dátových záznamoch sú v rôznych aplikáciách riadiaceho systému odlišné. Vhodnou metódou pre zabezpečenie pružnosti aplikačného modulu vzhľadom na dáta je vloženie programovej nadstavby medzi aplikačný modul a systém pre prácu so súbormi. V skutočnosti ide o definovanie konkrétnych a typových štruktúr dát. Konkrétne štruktúry dát sú obrazom reálnej dátovej základne v konkrétnej riadenej prevádzke. Typové štruktúry dát sú fiktívne, pretože nezobrazujú realitu a predstavujú nadmnožinu reálnych štruktúr dát rovnakého dátového súboru v rôznych prevádzkach.

V praxi to vyzerá tak, že aplikačný modul nevykonáva operácie nad súborom iba prostredníctvom systému pre prácu so súbormi, ale má k dispozícii pre každý súbor jeden obslužný modul pre prácu so súborom. Takýto obslužný modul má tieto dve základné funkcie:

- sprostredkovať styk so súborom,
- transformovať záznam typovej štruktúry na záznam konkrétnej štruktúry pri operácii typu zápis a vykonať opačnú transformáciu pri operácii typu čítanie.

Obslužné moduly pre prácu so súbormi je výhodné programovať tak, aby spĺňali podmienky reentrantnosti. Potom ich možno fyzicky oddeliť od aplikačných programov, ktoré ich vyvolávajú a možno ich sústrediť do spoločnej oblasti vnútornej pamäti, prístupnej všetkým programom. Týmto postupom sa docieľa, že pre každý súbor dátovej základne je vo vnútornej pamäti v určitom čase prítomný skutočne iba jeden obslužný modul.

### 3. Stavba programového systému

Z hľadiska vzťahu medzi časom prijatia informácií riadiaceho systému a časom ich spracovania sa programový systém skladá z podsystemu riadiaceho v reálnom čase a z podsystemu pracujúceho dávkovým spôsobom. Z hľadiska cieľa tohto príspevku je zaujímavý podsystem riadiaci v reálnom čase. Má tieto funkcie:

- komunikovať s okolím /prijímať informácie z riadeného objektu, vydávať príkazy riadenému objektu, viesť dialóg s dispečerom riadiaceho systému/,
- vyhodnocovať hlásené udalosti a plánovať ďalšie činnosti,
- vytvárať pre nasledujúce spracovanie súbor dát o hlásených udalostiach a o reakcii riadiaceho systému na ne.

Podsystem má hierarchickú štruktúru a je znázornený na obrázku 1.

Nadradené postavenie medzi programami má **h l a v n ý p r o g r a m**. Má tieto funkcie:

- prijať hlásenie z riadeného objektu,
- spracovať hlásenie spustenia príslušného aplikačného programu a odovzdať mu vstupné informácie,
- prijať správu o skončení spracovania hlásenia od aplikačného programu,
- v tzv. žurnálovom súbore evidovať informácie o významných udalostiach, ktoré nastali v riadenom objekte.

Vhodné je rozdeliť hlavný program na dve časti. Prvá - procedurálna časť programu - môže spĺňať podmienky nasadenia riadiaceho systému v rôznych aplikáciách. Druhá - dátová časť - opisuje konfiguráciu vlastného počítačového systému a riadený objekt z hľadiska potrieb riadiaceho programu. Vhodným programovacím jazykom pre hlavný program je strojovo orientovaný jazyk. Dôvodom na jeho použitie je potreba efektívne využiť služby, ktoré operačný systém poskytuje.

Nižšie hierarchickú úroveň tvoria **a p l i k a č n é p r o g r a m y**. Každý z nich je určený na spracovanie práve jedného druhu hlásenia z riadeného objektu, pričom pre každý druh hlásenia je k dispozícii aspoň jeden aplikačný program. Vlastná štruktúra aplikačných programov je hierarchická a modulová /obr.2/.

Nadradeným je r i a d i a c i m o d u l aplikačného programu. Má tieto funkcie:

- schopnosť začať vykonávať činnosť na základe impulzu vyslaného z hlavného programu,
- poznať radenie aplikačných modulov v aplikačnom programe a vyvolávať ich vo vyžadovanej postupnosti,
- odovzdať vyvolávaným aplikačným modulom ich vstupné informácie a prebrať od nich po skončení ich činnosti výstupné informácie,
- odoslať hlavnému programu informácie o skončení spracovania hlásenia.

Riadiaci modul aplikačného programu je vhodné rozdeliť na tieto dve časti:

- a/ procedurálnu a dátovú časť pre komunikáciu a hlavným programom /na programovanie je vhodný strojovo orientovaný jazyk/,
- b/ dátovú časť pre opis štruktúry aplikačného programu a procedurálnu časť pre vyvolávanie aplikačných modulov a pre distribúciu ich vstupných a výstupných informácií /na programovanie je výhodné použiť vyšší programovací jazyk/.

Ďalšiu hierarchickú úroveň tvoria a p l i k a č n é m o d u l y . Aplikačný modul predstavuje základný stavebnicový prvok programového systému. Jeho významným znakom je typovosť. Na programovanie aplikačného modulu je vhodný vyšší programovací jazyk.

Samostatnou skupinou aplikačných modulov sú moduly, ktoré zabezpečujú výstup na papierové médiá /tabelačný papier, dierna páska/. Vzhľadom na to, že výstup na papierové médiá je relatívne pomalý, pričom hrozí aj porucha na výstupnom zariadení, a tým aj potreba opakovať celý výstup alebo jeho časť, treba oddeliť výstup na papierové médium od spracovania hlásenia, ktoré ho vyvolalo. Vhodnou technikou je "spooling" výstupného súboru, t.j. zápis na magnetické médium s nasledujúcou konverziou na papierové médium, uskutočňovanou paralelne s ďalším spracovaním hlásenia. Umožňujú to také programy v riadiacom systéme, ktoré čakážu vykonať jednocúčelovú konverziu súboru na papierové médium a možno ich vyvolať aplikačnými modulmi.

#### 4. Záver

Typovosť programového vybavenia riadiaceho systému prevádzok podporuje jestvovanie súboru parametrov - zdroja informácií - ovplyvňujúcich funkcie aplikačných modulov. Ďalším priaznivým činiteľom je oddelenie procedurálnej časti od ľahko modifikovateľnej opisnej časti v hlavnom programe a v riadiacom module aplikačného programu. Úpravou opisnej časti má používateľ možnosť meniť jestvujúce funkcie modulov, reagovať na zmenenú konfiguráciu riadeného objektu alebo vlastného počítačového systému i jednoducho vkladať do programového systému nové moduly alebo programy. Najpodstatnejším prvkom podporujúcim typovosť riešenia je úplné abstrahovanie od reálnych štruktúr dátových súborov použitím obslužných modulov pre prácu so súbormi, ktoré umožňuje vykonávať operácie nad fiktívnymi typovými štruktúrami dát.

Úsilie o dosiahnutie typovosti programového riešenia vyvoláva potreba aplikovať riadiaci systém v odlišných prevádzkových podmienkach. Treba však zdôrazniť, že typovosť programového vybavenia možno dosiahnuť iba vtedy, ak bola cieľom vo všetkých predchádzajúcich fázach vzniku riadiaceho systému, a to už od jeho koncepčného návrhu.

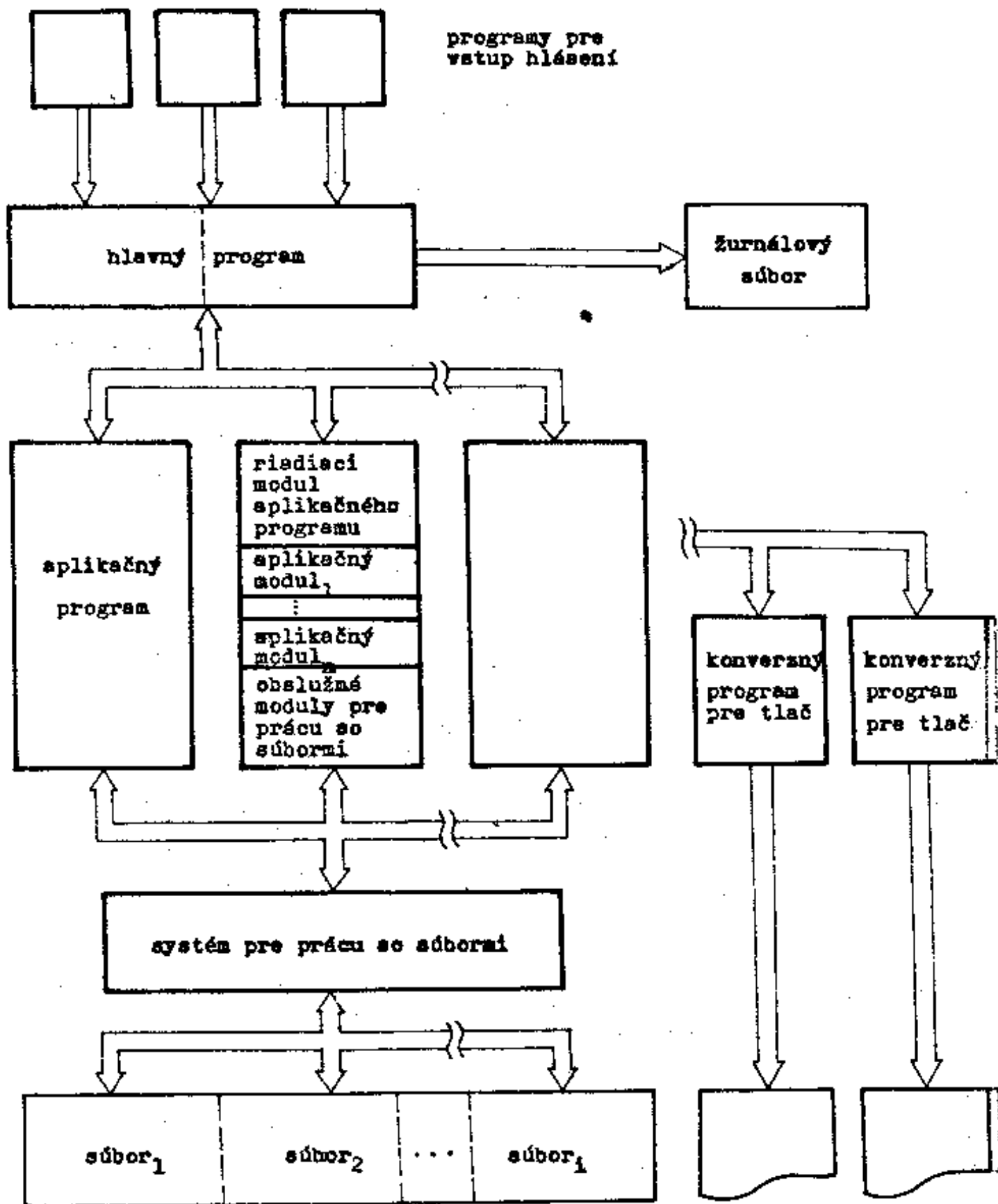
Skúsenosti z použitia rôznych typových programových produktov ukazujú, že užitočné typové programy sú iba tie, ktoré sa koncipujú ako otvorené, z ktorých používateľ vyberá iba potrebné funkcie, a ktoré neskôr - na základe nadobudnutých skúseností - môže sám dopĺňať a zlepšovať.

V príspevku popísaný programový systém je vytvorený na počítači SM 4/20 pod operačným systémom DOS RV. Jadro systému /aplikačné moduly/ a popisné časti riadiacich modulov aplikačných programov sú napísané v jazyku COBOL, všetky ostatné časti /zabezpečenie komunikácie medzi úlohami navzájom, medzi úlohami a dátovou základňou, medzi riadeným objektom a riadiacim systémom/ sú naprogramované v jazyku Assembler.

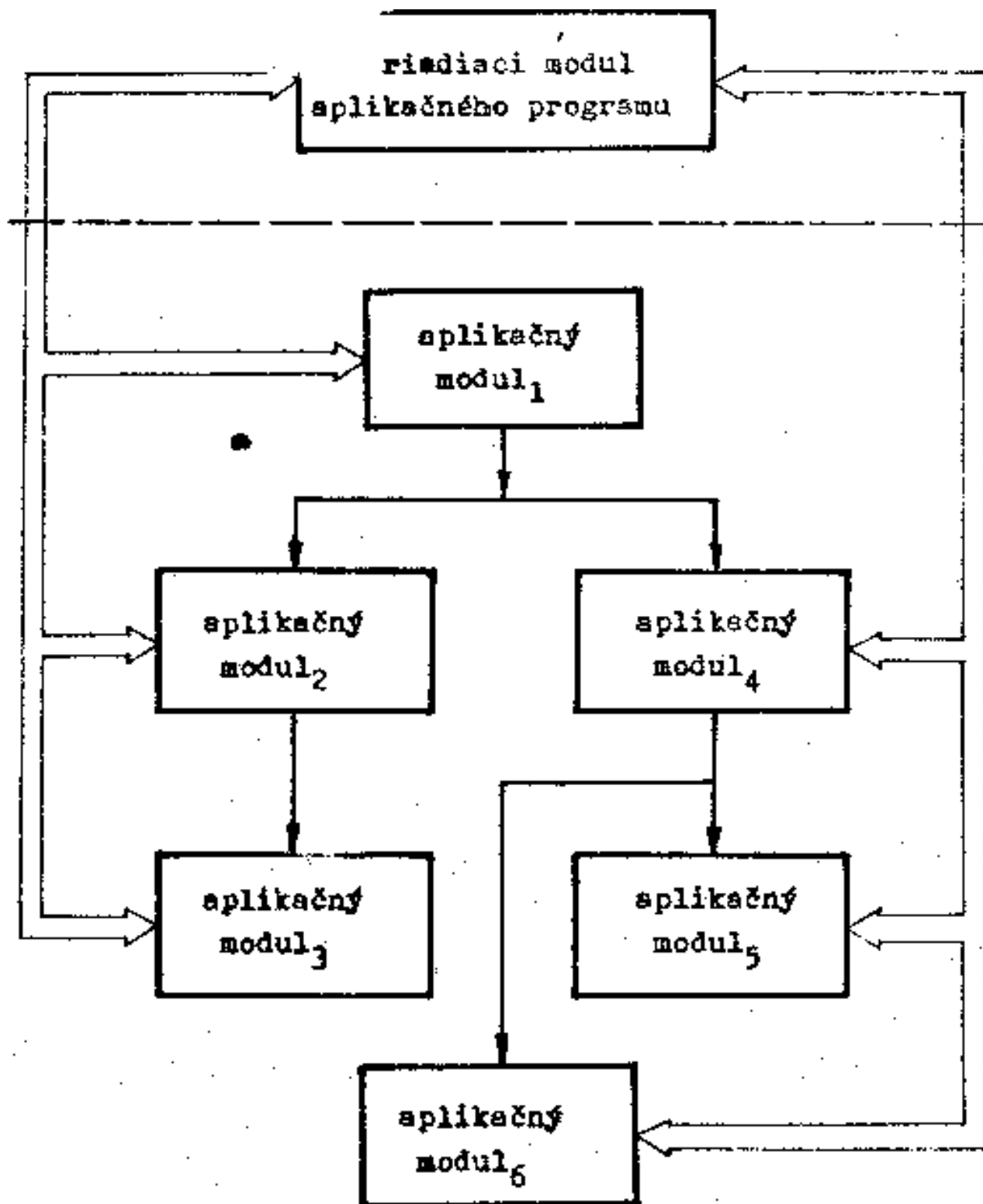
Riadiaci systém prešiel skúškami v laboratórnych podmienkach a v súčasnosti je pripravený na skúšobnú prevádzku u prvého používateľa.


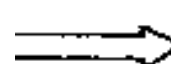
## Literatúra

- /1/ JANIGA J. a kol.: Algoritmizácia riadenia montážneho procesu.  
Výskumná správa, VUMÁ, Nové Mesto nad Váhom  
1983
- /2/ BÉBR R. : Programové řešení typových projektů pro  
různorodé uživatele.  
Zborník zo seminára Metody programování  
počítačů 3.generace, Havířov 1976



Obr.1. Schéma podsystemu riadiaceho v reálnom čase



 väzba, vyjadrujúca postupnosť v raďení modulov  
 informačný tok

Obr.2. Príklad usporiadania aplikačných modulov v aplikačnom programe