

Ing. Richard Bébr

Příspěvek se zabývá současným stavem a perspektivami projektování ASŘ. Cílem příspěvku je soustředění některých nápadů a námětů, které autorovi připadají zajímavé a využitelné v praxi.

== "Tady se buduje ASŘ, cizínče.

Tady si užijete srandý, až se budete třást děsem!"

(dle Jiránka).

1. Úvod

Nad kypícím programátorským životem zůstává jako neměnné memento kategorický imperativ: vše, co činiš, čin v rámci ASŘ, aby s úspěchem vyplňovány býti mohly pestré a rozsáhlé formuláře a odesílána hlášení o plnění plánů ASŘP, ASŘSč, ASŘO, ASŘTP a dalších různých ASŘxx. Agendy buďtež řešeny tak jako dříve, avšak navíc k vypracovaným programům jest určití, do které úlohy a skupiny úloh a do kterého subsystému agenda patří.

Fokusme se bez emocí prostudovat skutečný smysl a význam ASŘ a hledejme takové zásady, pravidla, náměty a rady, které mohou být užitečné jak ASŘ samotnému, tak i programátorům.

Autorovo snažení budiž v každé kapitole podpořeno obecně platnými pravidly (převzatými z různých světově proslulých literárních děl), která jako motto navodí obsah více či méně vědeckých úvah. Výskyt těchto citátů označíme symbolem "==".

2. Pojem ASŘ

2.1 Sémantika pojmu:

Jak známo "ASŘ" značí "Automatizovaný systém řízení".

Z gramatického hlediska je pozoruhodné použití rodu postatného jména: místo správného "ten ASŘ" používá se obecně "to ASŘ" (verzi "ta ASŘ" autor ještě neslyšel, byla by nicméně zajímavá). V praxi to znamená, že jako pojem je chápána sama zkratka, nikoliv její obsah. Jak ukážeme dále,

je tento fakt spíše užitečný.

Vůdčím pojmu ASŘ nutno hledat v překladu ruského АСУ (Автоматическая система управления). Potíž je ovšem v tom, že slovo "upravlenija" má v ruštině širší význam než české "řízení". Podle slovníku

upravlenija = řízení, ovládání, vedení, spravování (správa)

a je zajímavé, že i v angličtině

management = řízení, obhospodařování, správa

(a překládá se také jen jako "řízení"). Pojem "správa" se v češtině vytratil a tudíž z ASŘ sémanticky zmizely i správní (administrativní) agendy.

Doslovné chápání překladu vedlo k roztomilým zmatkům. Učení teoretici vedli spory na př. zda agenda soustředěného inkasa plateb obyvatelstva (všichni ji znáte) má být chápána jako ASŘ, když se v ní nic neřídí. Jestliže však to není ASŘ - co to tedy je? (Díky známému absolutizování módních pojmů bylo všechno, co není ASŘ prohlášeno za méněcenné a podezřelé; výpočetní technika musí být využívána pro ASŘ a na nějaké pokoutní agendy je jí škoda). Jak už to bývá, teoretici vedli spory a praxe běžela po svém: o každé agendě lze dokonce prohlásit, že poskytuje přinejmenším podklady pro řízení; každý šikovný programátor se naučil, jak vhodným popisem zařadit jakýkoliv program do nějakého toho ASŘ.

Nicméně mnohé organizace doplatily na doslovný překlad, mají různé manažerské systémy a databanky pro řízení, ale účtuje se shánějí ručně, neboť obyčejné a běžné agendy nechodí.

Většinou však dnes máme všude "to ASŘ", kde zkratka je přímo názvem, který obsáhne cokoliv (viz výše). Lze tudíž i v rámci ASŘ racionalizovat administrativu.

2.2 Práce na ASŘ:

=> Nemusíš být cvok, abys zde mohl pracovat, ale pomáhá to.

Na počátku ASŘ nebylo nic, jen několik zmatených a protichůdných výkladů tohoto nového pojmu. Brzy však začaly vznikat předpisy, směrnice a metodiky, které se podle principů zhubného bujení rozrůstaly do celých knihoven více či méně užitečných svazků. Konečně pak bylo dosaženo rozkošného stavu, kdy kdokoliv mohl kohokoliv za cokoliv pochválit nebo potrestat a vždy pro to našel v metodice ASŘ podklad.

=> Je účelné, aby v každé skupině zaměstnanců byl aspoň jeden pracovník.

V nedávné době se zrodila éra rozumnosti (vrátím se k ní v kapitole 3), která dosáhla velkého úspěchu vydáním vyhlášky 57/86 Sb. Je zajímavé, že vyhláška byla příznivě přijata odborně zdatnými profesionály, kdežto úředníkům, snobům a šéfům typu "00" (viz lit./4/) není po chuti. Už teď se projevují snahy "rozpracovat" vyhlášku pro podniky, organizace nebo i resorty. Vyhláška preferuje jednoduchost a odborný přístup a skrývá tudíž nebezpečí silného omezení administrativy a planého úřačování kolem ASŘ; správným uplatněním vyhlášky by zřejmě někteří lidé přišli o slušně placené zaměstnání.

2.3 Novinky v ASŘ:

=> Program bývá ceněn podle hmotnosti výstupních sestav.

V dřívějších dobách bylo ASŘ jednoznačně dávkovou záležitostí: vstupy na formulářích, výstupy na tiskových sestavách. Dnes se nám systémy hemží terminály, přenosy dat, dialogy a interakcemi; růžky vystrkuje grafika s barva.

Autor s dojetím vzpomíná na jeden z nepříliš dávných seminářů Programování ??, kdy z podla zazněla zmínka o terminálovém dialogu a publikum pobouřeně hučelo: "Dejte nám pokoj s terminály, my děláme pořádné ASŘ!"

Technika se však řítí dopředu a dnes je žádoucí spáchat v každém ASŘ alespoň nějaký ten interaktivní dialog (i když často provedením připomíná klasický systém "pár karet tam - balík papíru ven").

Na skutečně nejnovější novinky se podíváme v kapitole 5.

3. Přístupy a metody

3.1 Postupy tradiční a netradiční:

= Nepoužívejte počítač na úkoly, které lze efektivněji zpracovat ručně. (R.Hill, Hewlett-Packard).

Na semináři Programování '86 vzbudil zaslouženou pozornost příspěvek RNDr. Jiříčka (lit. /2/), který exaktně vyjádřil to, co mnozí mlhavě tušili (a báli se to být i jen formulovat, natož nahlas říci): bývá výhodné nenavrhnout rozsáhlé systémy jako celek, ale budovat je postupně po částech (někdy i docela malých, vždy však ihned chodivých). Citovaný příspěvek názorně ukazuje výhody tohoto postupu. Chtěl bych dodat, že právě moderní technika a především moderní programové prostředky nám umožňují efektivní postupné vytváření systému. Potíž snad je jediné v tom, jak tuto metodu nazvat, aby to lahodilo oku (i uchu) schvalovacích orgánů. Obrátíme-li se ke klasikům, zjistíme s údivem, že již sám "otec ASŘ" akademik Gluškov metodu postupného budování předvídal a nazval ji "metodou syntetickou" (lit./1/), kdežto navrhování systému jako rozsáhlého celku pojmenoval "metodou analytickou". Z novější doby doporučuji prostudovat lit./5/, kde se pro pružný postupný proces razí pojem "prototypový přístup".

K navrhování a budování ASŘ lze tedy použít metody syntetické nebo analytické. Považují však za nutné upozornit na dvojici přístupů, jejichž obsah se s výše uvedenými metodami nekryje. Je to přístup úřednický a přístup systémový. V praxi je nutno oba tyto přístupy bedlivě rozlišovat.

Úřednický přístup spočívá ve snaze, aby se nějaká agenda nebo skupina agend bez většího dumání vyprojektovala, uvedla do provozu a vykázala jako splněný úkol. Zásadně se nebere

ohled na takové podružnosti, jako je užitečnost agendy, spokojenost uživatele a podobně. Praktické nebezpečí tkví v tom, že úřednický přístup rád sáhne po "syntetické metodě", neboť se tím umožní dřívější realizace a tedy i dřívější výplata odměn zasloužilým šéfům.

Praxe ukázala, že právě syntetické postupné budování vyžaduje mimořádně důkladné systémové myšlení: čím lépe vystihneme v každém okamžiku všechny vazby a souvislosti (včetně vazeb na budoucnost) a v řešení je zachytíme (třeba tím, že necháváme účelová "fotra", ovšem důmyslně navržená), tím lépe se nám později pracuje na dalším rozšiřování a vylepšování.

Syntetická metoda předpokládá, že se budeme k dokončeným částem vracet, zlepšovat je a doplňovat; to však úřednický přístup nepřipouští. Syntetická metoda klade velký důraz na nenásilné získání uživatele a jeho zapojení do procesu návrhu i realizace; úřednický přístup vychází z daných pravidel typu "uživatel je povinen ...".

Praxe dále potvrzuje, že řešitel (dnes "projektant") nesmí podléhat emocím a módním vlnám. Největším neštěstím pro syntetickou metodu by bylo její nekritické zbožňování (a třeba povinné používání ve všech případech).

Jestliže použijeme systémového přístupu, pak již v první fázi rozumně zvolíme některou metodu a pak tuto metodu rozumně aplikujeme. Ač to zní podivně, není vyloučena ani kombinace obou metod. Za všech okolností se však vyvarujeme úřednického přístupu!

3.2 Ochrana dat:

== Potřebuješ-li skutečný zmatek, vezmi na pomoc počítač.

V dávkovém režimu to s ochranou nebylo tak horké. Stačila nějaká ta archívace a za to ostatní byl odpovědný provoz nebo uživatel. Interaktivní zpracování se v oblasti ochrany potkává s velkými problémy. Ve světovém tisku se dočítáme

o počítačových podvodech a s naivitou řidiče, který po namrzlé vozovce jede svou stodesátkou si říkáme: nám se přece nemůže nic stát! A přijde velké podivení, až bude jednou odsouzeno pár programátorů pro zanedbání ochrany důvěrných údajů.

Rozeberme tedy alespoň trochu tento problém. V programech musíme zajistit ochranu

- proti zničení
- proti zneužití
- proti zlomyslnému narušení.

Ochranu proti zničení v dávkovém zpracování umíme; jsou to archivní kopie. V interaktivním provozu už nestačí známé "tak to sjedeme ještě jednou". Ochrana proti zničení musí být vypracována tak, jak ji realizují lepší databankové systémy: záznamem všech transakcí (s časovými a lokalizačními údaji). Systémy typu "děd-otec-matka-sluzka a já" pro interaktivní provoz už nevyhovují!

Ochranou proti zneužití rozumíme takové zajištění údajů, které umožní pasivní přístup (čtení) jen oprávněným uživatelům. Ochrana proti zlomyslnému narušení se týká aktivního přístupu (zápisů, změn, výmazů). Ochrany v interaktivním režimu vyžadují komplexní řešení - programové, organizační i technické. K podrobnostem se vrátím snad až za rok, dnes jen něco málo o programátorské ochraně. Nejúčinnější zůstává ochrana heslem, vyžaduje však důkladné propracování; zamýšlíme se nad každým detailem.

Na příklad - má se heslo objevit na obrazovce, když je uživatel vkládá? Mnohé systémy zobrazení hesla potlačují ("aby to někdo neviděl"), což však asi není správné. Uživatel může vždy zajistit, aby nepovolaný neviděl heslo při vkládání (zvláště když pomůžeme výmazem hesla ihned po "RETURN"). Jestliže ale psané heslo uživatel vidí, můžeme žádat, aby vložil heslo správně hned napoprvé! Systémy s neviditelným heslem musí dát uživateli možnost několika pokusů, systém se zjevným heslem může vždy po vložení chybného hesla

spustit poplach ("poplach" nemusí znamenat konkrétní událost, uzavření hermetických vrat; stačí upozornit operátora a říkat nepřát: kdy který terminál co udělal).

Další (nemátkou vybrané) náměty:

Heslo má být na mediu uloženo kódovaně. Nejlepší je kód jednosměrný, při kterém ani autor kódovacího algoritmu nedokáže kódované heslo dešifrovat. Většinou stačí použít nějakého běžného šifrování, kde šifrovacím klíčem je samo šifrované slovo.

K heslu má být ještě přidáno doplňující jištění. Žádáme na příklad vložení rodného čísla uživatele; v souboru hesel pak máme u každého hesla seznam rodných čísel uživatelů, kteří s heslem mohou pracovat. Uživatel nejprve vloží své rodné číslo. Šikovnější program si ihned ověří, zda uživatel nemá dnes náhodou narozeniny a v kladném případě zdvořile gratuluje; je-li v ASŘ evidence docházky, zkontrolujeme, zda uživatel je vůbec v práci a podobně. Program potom obdrží heslo a zjistí, zda takové heslo existuje a zda je dotyčný uživatel oprávněn s ním pracovat.

Ošetříme i případy, kdy uživatel chce změnit heslo a náhodou vloží takovou kombinaci písmen, která už v seznamu hesel existuje (jako hesel se většinou používá neslušných slov se zdvojenými písmeny - na př. LL na konci - nebo se záměnou V za W - uprostřed - a podobně). Pokus o vložení existujícího hesla odmítneme, protokolujeme a protože odmítnutí může být pro zlého uživatele důležitou indikací, vyzveme nic netušícího uživatele "napadeného" hesla, aby své heslo rychle změnil.

K heslu a jemu příslušným rodným číslům přidáme seznam souborů (nebo třeba subsetů), s kterými smí toto heslo pracovat (včetně údaje o aktivním a pasivním přístupu).

Jednou z dobrých ochran je metoda, kdy ani programátoři neznají jména souborů ASŘ. Aplikační program zajistí pouze volání hesla a odněkud obdrží pro toto heslo tabulku "číslo - jméno" souboru. Aplikační programátor ví jen to,

že bude pracovat se souborem určitého čísla. Konkrétní jméno tohoto souboru se dozví až v okamžiku výkonu programu (jméno souboru musí být v programu jako proměnná). Není problémem zajistit, aby se čas od času v celém ASŘ změnila jména souborů (zvláštním programem, který nová jména generuje a nabrazuje jak v systému, tak i ve všech tabulkách a seznamech). Tím snadno donutíme programátory, aby ve svých programech museli hesla používat (a aby si nepsali nějaké účelové rutinky pro výpis nebo žertovnou změnu důležitých dat).

=> Kdykoliv se zdá, že se věci začínají dařit, bylo něco přehlédnuto. (Chilsholm).

Vynyslíte-li jaký systém ochrany, užasnete, kolik programátorů vám bude (i prakticky) dokazovat, že váš systém je průstřelný. Takovéto dokazování je velice veselé a recesní, nemuselo by však být takto chápáno na příklad pracovníky prokuratury.

4. Typová řešení

4.1 Prvky ASŘ:

=> Jestliže se program osvědčí, je třeba jej změnit.
Jestliže se program neosvědčí, je třeba o něm pořídit důkladnou dokumentaci.

Protože náplň ASŘ je všude zhruba tatáž, programují všude všichni všechno. Málokterý čtenář těchto řádků nikdy v životě neprogramoval základní prostředky, DKP, něco z personálistiky nebo jiný prvek ASŘ. Strašlivé multiplicitě prací by měly odpomoci typové projekty. Leč kupodivu se tak neděje.

Chceme-li pro nějakou agendu pořídit (zakoupit) typový projekt, zvedne se bouře odporu: máme specifické podmínky! V skrytu duše všichni víme, že třeba u základních prostředků nebo u DKP jsou "specifické podmínky" ptákovina (cizím slovem

"ptičestvo"). Podstata je, že nechceme typový projekt.
Proč?

4.2 Typové projekty:

Potíž spočívá v tom, že typové projekty mají řadu vlastností, které podporují jejich neoblíbenost. Uveďme jen některé:

- a) Typový projekt bývá zastaralý - nikoliv algoritmicky, ale programátorsky. Typové projekty potvrzují téze RNDr. Jiříčka (viz 3.1 a lit./2/), neboť jsou řešeny jako ohromný systém naráz v celé úplnosti; protože nikdy není dostatek lidí, řeší se rozsáhlý systém tak dlouho, že v době dokončení je už archaismem. Na tento problém poukázal i autor tohoto příspěvku v Havířově r. 1976 (!) - viz lit./3/.
- b) Z "a" plyne jistý konzervatismus, v typových projektech zakotvený. Nejsou typové projekty pro interaktivní práci, nevyužívá se databank, grafika je neznámá atd. Klasický je v tomto směru výrok raději nejmenovaného odborného pracovníka - "na co potřebujete databanku, když máte MARS ?!".
- c) S "b" také souvisí nevlídnost typových projektů k uživateli. Vychází se většinou ze strohých a ponurých předpisů a především ze zcela falešného předpokladu: "Uživatel si strašně přeje mít ASŘ; pro splnění tohoto přání je ochoten nejen cokoliv udělat, ale i trpět !". (Nevím, kde se tato téze vzala; ještě jsem neviděl uživatele, který by si zničeho nic začal vroucně přát ASŘ). Uživateli se tedy uloží splnění řady náročných požadavků a nenabídne se mu nic k usnadnění jeho nezáviděníhodné role. Uživatel je nucen pořizovat krkolomné vstupny, vyplňovat spousty nelogických formulářů, závažně měnit organizaci práce a luštit výstupní sestavy ničemu známému nepodobné; po zavedení typového projektu zhusta vidíme mnohé uživatele, jak na kalkulačkách dopočítávají potřebné údaje a vepisují je do výstupních sestav.
- d) Z "c" pak vyplývá známá slabina typových projektů - číselníky; uživatel je obvykle nucen přejít na zcela jiné, než dosud používal. Avšak číselníky typových projektů

bývají řešeny zastarale, podle zásad platných patrně ještě pro II. generaci a tudíž pro moderního uživatele zcela nevhodně. Takoveto číselníky si vyžadují celé soustavy kontrolních chodů a optických kontrol; v interaktivním provozu jsou stěží použitelné.

e) Nejzajímavější vlastností typových projektů je fakt, že jejich zavedením na jedné straně přibude řada problémů (viz "c" a "d"), na druhé straně se však nic nezíská! Zavede-li typový projekt automatizace mezd, přidám účtařkám spoustu práce, prodloužím termíny od uzávěrky k výplatě, vnesu do agendy plno ohyb (které se obtížně opravují) a celé je to dražší než ruční zpracování.

f) Vlastnost "e" je asi dána vlastností "c". Agendy jsou řešeny z hlediska počítače, nikoliv z hlediska uživatele. Ve mzdové agendě se schválně soustřeďují zvlášť dovolené, zvlášť nemoci, zvlášť to a ono. Mzdářky jsou však zvyklé zpracovávat každého pracovníka jako celek! Mzdářka si pamatuje "normální" pracovníky, které vybaví velmi rychle (na způsob "default" hodnot) a ví přesně, u kterého pracovníka ohlídat vyjimečné údaje. Tento efektivní způsob práce typový projekt promptně zlikviduje.

g) Typové projekty nejsou řešitelsky jednotně řízeny a nelze je spojovat. Nemůžeme propojit na př. typový projekt MZ s typovým projektem řízení výzkumu (aby bylo možno sledovat materiálové náklady na výzkumné úkoly); v typové výdejce není počítáno s číslem výzkumného úkolu a podobně.

4.3 Jak dál?

Bylo by asi účelné poučit se u "výrobců" univerzálního programového vybavení pro osobní počítače (zvládnutí jednotných datových struktur, možnost přizpůsobení prvku potřebám uživatele = "šití na míru", uživatelská přítulnost atd.).

Podle mého (a nejen mého) názoru:

- nevyrábějme "typové projekty", ale "univerzální programové vybavení", ze kterého si pár šikovných aplikačních programátorů vždy "ušije" své prvky ASŘ s minimálními náklady i pracností
- "univerzální programové vybavení" nechť mají na starosti specializované programovací podniky, závislé na prodejnosti svých produktů.

Současný stav, kdy typový projekt vznikne z účelového pouze administrativním úkonem je neudržitelný.

5. Nová pojetí ASŘ

== Optimista se domnívá, že svět, ve kterém žijeme, je nejlepší ze všech možných. Pesimista se obává, že je to skutečně tak.

5.1 Nové prostředky:

Mějme za to, že terminálový interaktivní způsob práce je už běžnou součástí ASŘ; umíme pracovat s databankami, nedělá nám potíže uživatelský dialog.

Budeme se tedy trochu zabírat nejnovějším problémem - osobními počítači a jejich zapojením do ASŘ. Za osobní počítač považujeme cokoliv, co má aspoň 64 KB, dvojici floppy a tiskárnu (samozřejmě i obrazovku a klávesnici).

5.2 ASŘ z osobních počítačů:

Slyšel jsem názor, že celé ASŘ lze vybudovat pomocí množiny vhodně zvolených osobních počítačů, navzájem propojených; většího počítače netřeba.

== Pesimista je dobře informovaný optimista.

Osobně si myslím, že ASŘ výhradně na osobních počítačích má (možná) perspektivu jen ve velmi malých organizacích nebo při velkém počtu malých detašovaných pracovišť. Pro

malých a kompaktnější podniky toto řešení nepřipadá v úvahu.
Důvody:

a) Pořádné ASŘ má (s ohledem na vyloučení duplicit) vždy několik "společných" souborů, ze kterých čerpá řada agend a úloh a jejichž údaje se využívají na různých pracovištích (na př. organizační schéma podniku, seznam pracovníků, šiferník školů a poč.). Tyto soubory musí být v každém okamžiku kterémkoliv programu k dispozici a to nelze osobními počítači efektivně zajistit.

b) V interaktivní síti osobních počítačů se velmi obtížně zabezpečuje dobrá ochrana dat (viz 3.2).

c) Vždy budou existovat agendy s prostředím tiskových výstupů (opět mě napadá evidence základních prostředků a I&F).

Pořízení na př. měsíčního výpisu ZP a I&F za podnik bude na osobním počítači trvat dlouho a pravděpodobně nebude dokončeno, neboť se malá běžná tiskárnička rozloží na prvotní.

Řadu běžných agend pro větší podniky si na osobních počítačích nelze vůbec představit.

5.3 Osobní počítače v ASŘ:

Chytrá aplikace osobních počítačů (jako inteligentních terminálů, nepojených na ústřední velký počítač) má však značné výhody. Přenesením jisté dávky inteligence k uživateli

- odlehčíme centrální počítač
- snížíme zatížení spojovacího vedení
- umožníme pořizování řady dokumentů a výstupů přímo na uživatelském pracovišti (schopnost "hard-copy" a běžných terminálů je zavrženíhodná a v dobrém ASŘ nepoužitelná)
- zvýšíme uživatelský komfort.

Snadno nalezneme řadu úloh, o kterých nemusí centrální počítač ani vědět; u jiných stačí jen předat souhrnné výsledky lokálního zpracování. Chytrý osobní počítač v účtárně, v pokladně, u plánovače a na podobných místech je k nezaplacení.

U vedoucích pracovníků instalujeme osobní počítač s barevnou grafikou. Data dodává centrální počítač ve standardní formě a vedoucí pracovník obdrží na obrazovce tabulku (lze i vytisknout), kterou vlastní programové vybavení osobního počítače může na požádání převést do přehledné grafické podoby.

Postarejme se vždy o to, aby programy pro chytré terminály byly dělány jednotně (viz "typové programy") a neplýtvale se prací. Dobře řízené nasazení osobních počítačů v ASŘ podporuje "syntetickou metodu", popsanou v 3.1.

6. Závěr

== Jsou věci, o nichž nelze nic zjistit;
nelze ani zjistit, které to jsou. (Jaffe).

Po mnoha letech projektování a budování ASŘ zjišťujeme, že celá problematika je vlastně pole neorané (jsem ochoten diskutovat i o tom, že jde o pole zorané). Rychlost vývoje technického a programového vybavení se často dostává do ostrého konfliktu s retardačními vlivy konzervativního úřednického myšlení.

Tam, kde ponechali volné ruce profesionálním odborníkům, mají dnes užitečné ASŘ; tam, kde dovolili řídit byrokratickým amatérům, mají dnes pozoruhodná monstra.

Protože na několika stránkách není možno celou problematiku ASŘ vyčerpat, uvedl jsem jen namátkou vybrané problémy, které by měly sloužit jako stimul diskuse a vlastních úvah čtenářů. Jestliže bude příspěvek takto pochopen, pak splnil svůj účel.

Literatura:

- /1/ Gluškov, V.M.: Vveděnije v ASU. Technika Kijev, 1974.
-česky: Automatizované systémy řízení. Institut řízení, Praha, 1976.
- /2/ Jiříček, P.: Racionální projektování složitých systémů.
Sborník Programování 86, DTČSVTS Ostrava, 1986.
- /3/ Bébr, R.: Programové řešení typových projektů pro různorodé uživatele. Sborník Metody programování..., DTČSVTS Ostrava, 1976.
- /4/ Bébr, R.: Programátor a jeho svět.
Sborník Programování 86, DTČSVTS Ostrava, 1986.
- /5/ Havlík, Zralý: Prototypový přístup k projektování informačního systému pro řízení. Sešity Inorge č.115, Praha, 1985.