

Ing. Vlastimil Čevela
Výpočetní středisko Ingstav Brno

PROBLEMATIKA KOMUNIKACE MEZI ÚČASTNÍKY ZPRACOVÁNÍ HROMADNÝCH DAT

1. Úvod

Všichni, kdo se zabýváme hromadnými daty, jsme více či méně odborníky v oblasti zpracování informací. Dokážeme navrhnout rozsáhlé automatizované systémy, snažíme se vést uživatele k moderním systémovým přístupům, určujeme přesná pravidla pro tvorbu prvotních dokladů a využívání počítačových výsledků - zkrátka snažíme se všude o maximální efektivnost.

Podívejme se však na informační přenosy v přípravě a realizaci automatizovaných systémů řízení ze širšího hlediska. Zamysleme se nad tím, že nejen uživatelé a počítač, ale i my programátoři jsme vlastně účastníky procesu zpracování informací a posuzujeme své konání i z tohoto pohledu.

Především zde tedy máme uživatelské prvotní data, která mu počítačem zpracováváme, a v některé vyšší formě výstupů vracíme. Kromě toho však existují další veliká množství informací, které si musí uživatel, programátor a počítač vyměnit. Nejprve v souvislosti s tím, aby ty všechny různorodé představy mohly být dovedeny do realizace a potom proto, aby to celé mohlo průběžně užitečně fungovat.

K tomu ještě přistupuje nutnost koordinace práce mezi uživateli navzájem, vazby jednotlivých etap vlastní technické realizace a v neposlední řadě vzájemná komunikace mezi pro-

gramátory, a to jak při postupu od prvních analytických úvah k odladenému programovému vybavení, tak i při studiu a zvyšování kvalifikace, nebo vzájemném předávání znalostí.

Předkládaný příspěvek je proto motivován snahou upozornit na důležitost všech těchto - vlastně vedlejších - informačních přenosů mezi uživateli, programátory a počítačem, které však mohou velice významně ovlivňovat efektivnost dosahovaných výsledků.

Tématika je dále zúžena na dávkové zpracování hromadných dat, což odpovídá současnému stavu v našich podmínkách zatím nejčastějšího využívání počítačů.

2. Komunikace při tvorbě a realizaci informačního systému

V rámci každého informačního systému, ať již ve stádiu jeho tvorby, nebo při pravidelné realizaci zpracování dat, je možno z pohledu předávání informací vyčlenit 3 hlavní činitele /prvky/.

Nejprve je to množina uživatelů, t.j. pracovníků odborných útvarů podniku nebo organizace, kteří se tím či oním způsobem podílejí na jeho přípravě, vlastní rutinní funkci anebo další inovaci.

Dále jde o množinu programátorů, chápanou zde v širším slova smyslu od systémových analytiků přes vývojové programátory až k případným provozním programátorům, zajišťujícím drobnější údržbu programového vybavení.

Množina počítače pak představuje nejen technické vybavení pro pořizování dat a vlastní počítač včetně zařízení pro úpravu a předávání výsledků, ale též všechny pracovníky, kteří se na přípravném nebo rutinním provozu ve výpočetním centru podílejí.

Prvek uživatelé tedy

- podává informace o současném stavu

- formuluje požadavky na vývoj systému a jeho další inovaci
- nejprve spolupracuje při určování pravidel pro tvorbu prvotních dokladů a organizaci práce v příslušné automatizované oblasti a potom potřebné činnosti pravidelně realizuje
- dokáže správným způsobem reagovat na chyby a poruchy systému
- efektivně využívá údaje z počítačových výstupů k plnění podnikových úkolů

Prvek programátor počtem

- provádí analýzu současného stavu a s použitím znalostí o počítači, odborných metod a potřebné formalizace, navrhuje nové automatizované řešení a jeho další úpravy
- převádí myšlenkový, více či méně formalizovaný algoritmus zpracování do programovacího jazyka počítače nebo jiných zúčastněných technických zařízení
- spolupracuje při formulaci pravidel pro činnost uživatele v příslušné automatizované oblasti
- určuje informační návaznosti a parametry jednotlivých pracovních postupů v provozu výpočetního centra

Prvek počítač pak

- poskytuje informace o technickém a programovém vybavení výpočetního centra a o zásadách jeho provozu
- dle podkladů programátora realizuje výběr existujícího a tvorbu nového programového vybavení

- pravidelně realizuje jednotlivé pracovní postupy dle informačních návazností a parametrů, předepsaných programátorem
- dokáže správným způsobem reagovat na chyby a poruchy systému.

Pro informační vazby výše popsaných prvků si zavedme pojem komunikace, jako označení pro přenos informací /sdělení/ mezi dvěma prvky.

Ze stručného přehledu činností uživatele, programátora a počítače nám vyplývají některé důležité poznatky.

- Komunikace probíhá všemi směry, t.j. výměna informací je vždy oboustranná a mezi všemi prvky vzájemná.
- Informace, předávané mezi jednotlivými prvky tvoří vlastně celková pravidla hry, takže na kvalitě vzájemné komunikace /resp. na komunikaci v nejslabším článku tohoto řetězu, který kromě toho musí být úplný/ záleží kvalita výsledků vlastního technologického procesu zpracování hromadných dat.
- Vzájemné informační vazby existují trvale a to nejen mezi uživatelem a počítačem, ale i ve vztahu k programátorovi. Praktický život totiž nelze charakterizovat jako stav věcí neměnných, ale jako neustálý sled průběžně probíhajících menších či větších změn, a ta objektivně existující skutečnost se musí nutně promítat i do našich býti sebelépe postavených pravidel hry. Programátor tedy musí soustavně zajišťovat realizaci potřebných úprav, resp. další vývoj, a občas řešit i neočekávané mimořádné situace.

Je vhodné se sice zaměřit i o nutnosti spolehlivé a přesné komunikace mezi uživateli - uživateli při částečné práci v rámci organizace počítače, počítač - počítač na př. při předávání nezávislých nebo jednotlivých strojů, ale také mezi lidmi třeba při používání sítě a s velice důležitým vztahem programátor - programátor, ať již v průběhu procesu od analýzy k programu, nebo při vzájemné výměně znalostí.

3. Řídicí pracovník jako uživatel

Od toho momentu, kdy počítač pro podnik něco zpracovává, stal se fakticky jeho zaměstnancem. Z toho ale dále vyplývá, že ať se to komu líbí nebo nelíbí, je v určité oblasti zpracování informací podnik na počítači, resp. na výsledcích jeho práce zcela závislý.

Pro každého řídicího pracovníka by z toho mělo plynout především jednoznačné vědomí, že počítač v žádném případě nebude stát místo, ale že se určitě do podnikových činností zapadá. Na příslušném vedoucím pak bude záležet pouze, zda to bude pod jeho vedením a plnou kontrolou jednotlivých kroků, nebo živelně.

Je pravda, že dnes je situace snadnější než před několika lety, poněvadž existují příslušné vyhlášky a prováděcí předpisy, které upravují postup při budování ASŘP. Ovšem rámcová vyhláška je jedna věc, a realizace jejích ustanovení v praxi je jiná a spravedlivě často obtížnější záležitost. A tedy už opět hraje podstatnou roli komunikace, v tomto případě mezi programátorem /počítačem/ a uživatelem.

Na okraji věcných pracovníků je třeba uvést, že to s námi vůbec nejsou snadné. Spolu s nashromážděním počítače jsou jim předkládány problémy, se kterými se ve své třeba i dlouhodobě musí ještě vyrovnávat, a v řadě případů se jim souvislosti zcela vymykají a jejich dosavadních zkušeností.

Vesmíru si třeba následující příklad. Zkušenému programátorovi je bez jakýchkoliv komentářů úplně sřejmá, že pokud

v oblasti hromadných dat nedokáže důsledně realizovat taková pravidla zpracování, která by zajistila přesnou a úplnou dávku potřebných informací včas a na místě, kde mají být, seobelepší vzorec na výpočet na základě těchto údajů je k ničemu.

Je mu jasné, že spravidla existuje řetěz navazujících lidských a strojových činností, často se značně složitými věcnými i organizačními vztahy. A teď se vlijme do situace, že toto máas dokázat krátce a srozumitelně vysvětlit tak, aby v posluchači vzniklo nevratné přesvědčení, že organizační opatření je určitě naléhavější než vzorec /už proto, že ten potom dokáže zcela nezávisle realizovat jeden programátor sám, zatímco nyní je třeba zapojit několik desítek lidí/. Přitom se jedná o technicky na slovo vzatého odborníka, který z toho titulu perfektně ovládá různé optimalizační metody své oblasti, ale poprvé v životě slyší něco o chybách v děrování, spožděných vlakových spojích, nedoplněných přepočítacích koeficientech a podobně.

Při různých příležitostech se hovoří o nezájmu vedoucích pracovníků o problémy ASŘ. Jistě je na tom mnoho pravdy:

- uživatelé neumí exaktně formulovat svoje představy a cíle
- nemají chuť se zabývat řešením odborných problémů, které jsou jim cizí
- nechtějí rozhodovat v situacích, kdy jim nejsou zcela jasné všechny souvislosti atd.

Zásadní změna tohoto stavu je asi možná jen během dalšího vývoje ale přece jen by se neměly skládat ruce v klín. Jedná se totiž především o otázku informovanosti, a tu je někdy možno alespoň částečně řešit vlastními silami. Tím je míněno seřazení tematiky ASŘ do cyklické přípravy vedoucích pracovníků na všech úrovních podniku, důkladná uživatelská příprava každé konkrétní akce, všeobecná osvětové články v závodních časopisech, přednášky ČVTS, dálkové kurzy a pod.

Rovněž podklady, které jsou z oblasti ASŘ předkládány vedoucím pracovníkům, by se alespoň pro začátek měly snažit o menší podíl efektní odborné terminologie a více se opírat o konkrétní, normálnímu člověku běžně srozumitelné fakty a jasné souvislosti.

4. Předávání podkladů mezi programátorem a uživatelem

Měli bychom si uvědomit rozdílné přístupy, které dosud většinou existují. Zatímco pro lidi od počítačů jsou obvyklé více či méně formalizované informace - ať již ve formulářích pro zobrazení informací v datových souborech, různé tabulkové popisy systému atd., v uživatelské oblasti, t.j. pracovníkům odborných útvarů ještě velice často není takový způsob vyjadřování běžný.

Nebo i logické vztahy. Když napíšu, že "vybírám účet 101", je mi jasné, že ostatní účty jsou ignorovány. Časem se však nestačím divit, kolik lidí "myslelo", že třeba účet 103 bude taky "nějak zpracován". A tak příště poučen, už raději připišu, že "ostatní účty do sestavy nebudou zahrnuty".

Chceme-li se dostat k praktickým výsledkům, musíme v našich pokynech a jejich výkladech občas slevit a sestoupit s olympu systémů a subaystémů, zobrazení informací či vývojových diagramů toků dat, na úroveň běžně srozumitelné řeči šoféra, stavbyvedoucího anebo starší paní z domácnosti, která momentálně dělá skladovou evidenci třeba v dílně. Bez dohodné přesné spolupráce při vystavování dokladů na všech těchto pracovištích se totiž veškeré naše snažení mine cíle.

Pro komunikaci "programátor - uživatel" /dle dřívějšího vymezení těchto prvků, t.j. i systémový analytik = programátor/ by tedy bylo možno formulovat tyto základní požadavky:

- záznamy o jednáních s odbornými útvary, podklady k oponenturám z jejich strany a vůbec celou dokumentaci pro uživatele formulovat stylem, který odpovídá současné vzájemné komunikační úrovni

- uživatelská dokumentace musí být co nejtručnější a přehledná, ale přitom naprosto srozumitelná i pro nejmenší kvalifikované zainteresované pracovníky
- je nutné účastnit se tvorby návrhů a pokusů, které na základě dohod s programátorem zpracovává uživatel, nebo alespoň důsledně vyhodnotit jejich opozitaci před vydáním; často totiž dochází k nepřesnostem nebo k různým výkladům
- je potřeba kontrolovat, zda uživatelské návrhy či pokusy jsou upraveny na všechny pracovníky, kterých se týkají; pracovníci obou úrovní často totiž často nevidí k vedlejším stolu, či do sousední kanceláře, kde se nacházejí se shodně výchovně podklady, třeba proto, že se jedná o úroveň jiného náčestka
- podle možnosti se pokusit sestavit modelové příklady nebo testy, na jejichž základě by bylo možno ověřit, zda pracovníci uživatele správně pochopili naše představy o složitější vyplňování dokladů, opravě chyb a podobně

5. Chyby při zpracování informací

Všechné komunikace mezi uživateli, počítačem i programátorem, včetně vlastní realizace zpracování dat, nejsou vlastně nic jiného než přenosy informací. Při přenosu jakéhokoliv signálu a tedy i informace může nastat určitá jistá úroveň rušení nebo šumu, která překrývá informaci různě intenzivně skreňuje. Běžná praxe nás též učí, že nikdy není bezchybný, a každý člověk a dokonce i každý stroj se přesto může chybně. Totiž je ovšem v tom, že již při vlastním technologickém zpracování informací a přenosu informací z oblasti zpracování dat se jedná o složitější procesy informace. Je přitom ještě složitější s lidmi a různými společnostmi, od kterých

či méně kvalifikovaného člověka až po nejmodernější výrobky průmyslu elektroniky, a průběhem jí dealova miliony informačních znaků. K tomu přidejme ještě možná nejasnosti, nepřesnosti a nedorozumění při zpracování projektu a programu, tvorbě uživatelských prováděcích pokynů a realizačních směrnic pro počítač a vše alespoň přibližně vynesenu šifri problému.

Z předchozích řádků by tedy mělo být zřejmé, že chyby především v datech se prostě nelze vyhnout, takže by z toho mělo zcela jednoznačně vyplynout, že s nimi vždy /podtrženo!/ musíme počítat.

Při původních úvahách o těchto odstavcích vznikla představa, že by snad bylo možno vytvořit nějakou klasifikaci chyb dle jednotlivých komunikačních vztahů. V průběhu dalšího rozpracování se však ukázalo, že věci skutečně ani zde nelze oddělovat, a snad by zachován celkový systémový pohled ve všech souvislostech.

Vezměme si pro příklad nějakou uživatelskou chybu na prvotním dokladu, třeba nesprávně uvedené číslo nákladového objektu na účtářce materiálu. Ani sdaleka se vždy nedá jednoznačně říct, že je to pouhý omyl nebo nedbalost skladníka či evidanky. Je nepochybně docela dobře možné, že nesprávnost vznikla proto, že nebyl k dispozici jednoznačný a srozumitelný pokyn pro vyplnění, nebo že došlo ke změně číslování, která se na toto pracoviště zatím vůbec nedostala, anebo dokonce ten, kdo vydal pokyn k vystavování dokladu, ani svoje správné číslo pro sledování nákladů nezadal.

Uvažuje-li případ nejasného pokynu, je to chyba komunikace mezi programátorem a uživatelem, to ostatní je nedostatečná komunikace a koordinace mezi útvary uživatele. K tomu pak přistupují další možná zkrácení, nebo i stráty informace v průběhu jejich zpracování. Od započatého dokladu v zásuvce, přes překlepy či neaktívaný děrovač, až po chybu programu nebo vyjádření i počítače, a nakonec nesprávně odebrané sestavy. Samozřejmě, že po celé této cestě, která je fixována průvodkami,

programy či operogramy, tedy lidskými výtvoři, podle kterých se řídí jiní lidé a stroje, mohou vzniknout další nepřesnosti a nedorozumění vlivem nedokonalé komunikace.

6. Chybivost a kvalita komunikace

Je velice obtížné vysleďovat nějaké obecně platné, stoprocentně účinné zásady, které by pomohly proti chybám:

- Budeme-li vyžadovat příliš podrobné návody, aby řešily vše, budou tak složité, že je nikdo nebude číst, a když ano, tak je stejně nepochopí /nehlédě na přesnost jejich tvorby a náročnost údržby/.
- Použijeme-li příliš rozsáhlé počítačové kontroly, vznikne nám /kromě nároků na přesnost a spolehlivost programů/ problém sekundárních chyb při pořizování a příp. údržbě zajišťovacích duplicitních informací /kontrolních součtů, průvozek, povolených souvztahností atd./, poněvadž vše je relativní, a chyba se vyskytne, spravidla když to nejmeně čekáme, nejen v kontrolovaném údaje, ale i v kontrolním.
- Také je nutné si uvědomit, že odhalení chyby na př. v datech problém nekončí, ale naopak začíná, poněvadž je nutné řešit, co má být provedeno dále. Jsme totiž okamžitě postaveni před rozhodování, zda chybná položka bude pouze signalizována a ponechána, či vyřazena /problém hlášení úplnosti/, nebo jen eliminována její negativní působení - pokud taková možnost existuje. To však není ještě vše, poněvadž musíme dále programově i organizačně dořešit, kde, kdy a jak provede případnou opravu, a samozřejmě ho to spolehlivě naučit.

- Z předchozích řádků vyplývá jednoznačné poučení pro stavbu vstupních modulů subsystémů s více návaznostmi na několik odborných útvarů. Pokud totiž není v pořádku některý údaj s dalšími vazbami, na př. číslo materiálu není v ceníku, a proto nemá skupinu sledovanosti pro statistiku, je nutné zcela nekompromisní blokování, i kdyby to třeba mělo za následek zkrácení stavu zásob. Jestliže totiž takový případ do zpracování pustíme, současně nám vyvstane úkol, zajistit rozpuštění stále narůstající hodnoty "ostatní" do jednotlivých řádků příslušného statistického výkazu.

Citovaný příklad je přitom ten nejjednodušší, protože vazby bývají většinou složitější.

- Nejen že musíme vědět, ale musíme naučit i ostatní, že není chyba jako chyba. Děrovačka si musí být vědoma toho, že chyba v návěští černé pásky či v záhlaví dokladu nadělá podstatně víc problémů, než překlep v jednom řádku v kolonce množství, která je hlídána kontrolním součtem. Dále s tím souvisí ekonomie práce při optické kontrole vstupních dat/

- Značný rozdíl je též mezi chybou programu, k její nápravě u všech sto tisíc položek základní kartotéky stavu zásob stačí opravit pár instrukcí a třeba i hodinové přetřídění a konverze datového souboru, a chybou v metodickém pokynu, který způsobí, že 300 skladníků vystaví nesprávně celou několikatisícovou dávku dokladů, a pak je musí všechny opravovat a dát znovu děrovat /příp. zajistit i příslušná storna/

I z tohoto krátkého výřtu je zřejmé, že stupeň chybovosti a poruch v procesu vlastního zpracování dat ukazuje nejen úroveň organizačního a programového řešení, ale v ne-

poslední řadě je také výrazným měřítkem kvality komunikace uvnitř i vně jednotlivých prvků celého informačního systému.

7. Identifikace a komunikace ve VS

Odborníci, patřící dle našich souvislostí do skupiny "programátor" by podle všech pravidel měli mít mezi sebou a samozřejmě také směrem na "počítač" realizován ten nejkvalitnější systém komunikace, poněvadž jejich specializací je právě zpracování informací.

Presentovaný příspěvek je zaměřen především do této oblasti, takže si může každý čtenář sám zodpovědět následujících několik otázek:

- Dodržujete ve Vaší organizaci pro veškerou analytickou dokumentaci /t.j. pro tu, která musí být zpracována před tím, než se začne programovat/ jednotný obsah, t.j. které kapitoly a v jakém pořadí musí vždy obsahovat?
- Máte uzákoněnou jednotnou formu sadění programování jednotlivého programu?
- Platí pro všechny subsystemy v rámci současné etapy budování Vašeho AIS nějaký souhrnný pokyn či interní normou stanovené jednotné systematické číslování či identifikace označení všech základních prvků, jako na př. dílčí projekty, programové chody, programy, soubory dat, sestavy atd.?
- Máte ve Vašem výpočetním středisku pro určitý typ výpočetního systému zavedenou pro ladění, ověřování i vlastní výpočty nekompromisně jednotnou formu zápisu na oblouku /formulář, vjevojový diagram a pod/ ?

Těchto pár bodů si samozřejmě neklade žádné nároky maserickými systematické hodnocení, ale pokud nebylo možno

ani na jednom z předložených dotazů odpovědět ano, tak by měla nastoupit větší úroveň než stavu komunikace na příslušném pracovišti.

Zavedení jednotné identifikace /4/ a jednotné formy pro dokumentaci /5/ by totiž skutečně mělo být tím nejskladnějším směrem k racionalizaci analytické a programátorské práce. Pokud tyto podmínky nejsou splněny, těžko je možno uvažovat o nějaké větší účinné práci, a samozřejmě vznikají problémy i při rozpracování analýzy do programů a tvorbě podkladů pro realizaci - zejména o starostech s pokračující údržbou.

8. Vzájemná znalosti mezi programátory

V oblasti vzájemné výšiny zkušeností mezi programátory a výchovy nových pracovníků začínají v poslední době hrát velice pozitivní roli nové metody programování, založené na systematickém přístupu. Poměrně jedním z jejich nejdůležitějších rysů je zdůraznění srozumitelnosti a přehlednosti /aby byla možná kontrola logiky programu na papíře/, stávají se tyto tendence skutečně důležitou základem pro zlepšení celkového stavu komunikace.

Systemový přístup, t.j. rozdělení problematiky řešené nejprve projektem a pak programem, postupně na různé podrobné realizovací úrovně a v jejich rámci na jednotlivé nezávislé prvky s přesnou definicí vstupu, výstupu a vzájemných vztahů, jak dovedení zkušenosti ukazují skutečně umožňuje, aby celá logika zpracování byla dokonale a beze zbytku zvládnuta.

Ač již tyto techniky byly nejvíce strukturované, systematické či například, nebo modularní programování, systém postupného zjednodušování či přístup shora dolů anebo HIPO diagramy, strukturované návrhování a podobně, v zásadě se pořád jedná o různé výše uvedené systematické principy. I když jsou používány různé názvy, tyto metody si navzájem nekolidují, ani se nepatří k nějaké přísné klasifikaci. Pouze je tím nevhodnější název vlastně nebo určitý pohled na tutéž věc, o které je pak možno samostatně hovořit.

Jako příklad úspěšné komunikace mezi různými výpočetními středisky je možno uvést algoritmy základních úloh zpracování hromadných dat, popsané ve VS Ingstav Brno metodou uspořádaného programování. Jedná se o jedno z možných řešení základní logiky nejběžnějších úloh z oblasti hromadných dat, t.j. prosté konverze, mapování a stupňové konverze /sopčtování na různých úrovních/ v jazyce COBOL TESLA 200.

Algoritmy byly jednak publikovány v literatuře /1/ a /3/ a dále též vydány ve formě stručného 16 stránkového sborníku /5/, původně sice pro vlastní potřebu, ale posléze na základě zájmu rozšířeny do řady organizací^{*} někde s příslušným výkladem, jindy i jen jako písemný materiál. Konkrétní ohlasy o jejich využitelnosti jsou jednoznačně pozitivní.

Na př. ve VS ZVE generální ředitelství Brno byla potvrzena zkušenost, že jejich aplikací se skutečně snížil počet kopií a u jednotlivých úloh na minimum.

V ÚVT Ostrava byly prakticky převzaty jako reprezentativní struktury do základní učebnice k programování /6/ a tím, že takto získaná přenositelnost programů je využívána jako výchovný moment. Pokud totiž autor dodrží daná pravidla, "za odměnu" není nadále obtěžován požadavky na údržbu programu.

Ve všech případech se potvrzuje celkem logický závěr, že noví pracovníci přijímají tyto metody zcela automaticky /poněvadž vlastně nic jiného neznají/, zatímco u zkušenějších někdy bývají problémy. Těžko se jim na příklad vysvětluje, že úspora milisekundy při provádění instrukce nebo několika minut při psaní do formaláře a děrování štítků zdrojového

* / Sborník "Algoritmy ÚP" je možno zdarma obdržet na základě písemné objednávky na adrese VS Ingstav Brno, odd. automatizace obch. úseku, 656 29 Brno, Koněvova 55.

textu, může mít také negativní vliv. třeba po roce, až bude muset do programu zasahovat někdo jiný a strátí několik hodin času anebo i kompilaci navíc, poněvadž taky nemusí dosahovat autorových kombináčních schopností. Pokroč je však omezující pracovník potom postaven před úkol současně zajistit údržbu dvou cizích programů, z nichž jeden je uspořádaný a druhý ne, žádná další argumentace již spravedliva není nutná.

9. Závěr

Cílem tohoto příspěvku nebylo uvádět něco zásadně nového, ale pokus o připomenutí některých celkem všeobecně známých věcí a jejich uvedení do vzájemných souvislostí. Z těch pak vznikají další pohledy, které mohou vést až k novým přístupům při řešení celé řady problémů.

Na závěr by snad bylo možno předložit následující tři myšlenky, které do jisté míry syntetizují předchozí úvahy o identifikaci, systémovém přístupu a komunikaci vůbec:

- Když si spočítáme, kolik času jsme v různých dokumentacích věnovali vyřizování individuálních identifikačních a klasifikačních soustav, měli bychom dojít k závěru, že nejjednodušší systém je pořadové číslování. Každému kdykoliv vzniklému prvku je možno tímto způsobem ihned přidělit jediný identifikační údaj, zatímco klasifikací nejen může být libovolné množství, ale jejich kritéria se mohou objevit teprve po čase, v dalším stádiu vývoje systému.
- Nejdůležitější zásadou při požadavku na spolupráci by mělo být neurčovat "jak dělat věci", ale důsledně jenom "co má být vytvořeno", přesně definovat vstupní omezení /pravidla hry/ a výchozí údaje a správně formulovat požadavky na výstup /výsledky/. Jenom tak je možno skutečně úspěšně delegovat pracovní prácnost vlastní realizace daného úkolu.

- Jakákoliv ústní přednáška ústní nebo písemná informace by neměla mít za cíl nějaký výjimečný efekt, či ohrožení nad analýzou obsahu, ale její užitečná hodnota vzniká teprve tím, že je správně pochopená a osvojuje se střevy posluchače nebo čtenáře, kterému je určena. Proto by první a nejdůležitějším kritériem pro hodnocení všech předkládaných materiálů měla být užitečnost a zpracovatelnost.

10. Literatura

- /1/ Sborník přednášek ze semináře DT ČVTS Ostrava "Metody programování počítačů III. generace", Havířov 1976
- /2/ Sborník přednášek ze semináře "Racionalizace analyticko-programátorské práce a provozu VS", Odd. skupina vjz. techniky ZP ČVTS Ingstev Brno 1975
- /3/ V. Čevala, Algoritmy rekursivních děloh hromadného zpracování dat metodou uspořádaného programování, Kalkula algoritmov III. díl s 3. synopsis SVTS Bratislava - Algoritmy ve výpočtové technice 1975, 41/400-41/412
- /4/ V. Čevala, Identifikace jednotlivých prvků v systému práce VS, vyjde v průběhu roku 1978 v MAA
- /5/ V. Čevala, Algoritmy DP, interní materiál E00002 VS Ingstev Brno, 1975
- /6/ V. Škarba, Strukturované programování - metodika, ÚVT Ostrava, 1977
- /7/ A. Suchotin, Věda a informace, NF Praha, 1973