

APROS - POUŽITIE PRE PROJEKTOVANIE A PROGRAMOVANIE

Balla Vladimír, ÚVVT, Východoslovenské telesiarne, n.p., Košice

1. Úvod

APROS /Automatizované Projektovanie systémov/ vznikol vo VŠE na prelome rokov 1981/82. V tom čase riešil náš útvar podstatnú časť projektu, nadaného ČSSR organizáciou UEDP UNIDO pri OSN "Použitie moderného systému riadenia údržby v hutníckom podniku". Keďže sa jednalo o projekt veľmi rozsiahly a obsahovo nerovnorodý, ktorý predpokladal pri svojom vývoji pružnú riadiacu a dokumentačnú prácu, rozhodli sme sa aspon časť týchto činností zveriť počítaču. Kanaštatne, v tom čase nebol v ČSSR používaný riadený systém, ktorý by napokojoval naše požiadavky v uvedenej sfére. Inorga začala jednanie s maďarskými partnermi o prenosení ich produktov, avšak čas implementácie bol pre nás prídlhý a ani forma a obsah vstupov a výstupov nezodpovedali našim potrebám.

Z týchto dôvodov sme sa rozhodli vybudovať nadstavbu k aktivite 8 projektu, ktorá má okrem iného za úlohu vyriešiť katalóg dát systému. Táto nadstavba predstavuje rozšírenie katalógu dát o iné, netradičné prvky /napr. moduly, identifikátory, transakcie/ a súbory, a tiež vývoj software, ktorý zabezpečí využívanie rozšíreného katalógu v oblastiach:

- projektovania
- programovania
- riadenia prác na projekte
- dokumentovania
- prevádzkovania

Pre krátkosť času nebolo možné do dnešného dňa doviest APROS do finálneho stavu. Napriek tomu nás aj čiastočne výsledky opravňujú k zverejneniu tohoto systému s dôrazom na skutočnosť, že jeho vývoj bude ďalej pokračovať a predpokladá-

me, že APROS bude vhodný pre ďalšie použitie.

V úvode je ešte potrebné zdôrazniť základnú črtu systému. Cieľom APROS-u nie je zbaviť človeka zodpovednosti za projekčné, programovacie a dokumentačné práce, iba tieto činnosti uľahčiť a v čo najväčšej miere umožniť znovupoužívanie raz použitých faktov tým, že ich udržiava vo svojom katalógu a na požiadanie sprístupní vo vhodnom tvare.

2. Centrálny katalóg dát

Základom APROS-u je jeho katalóg dát, ktorý obsahuje informácie o všetkých entitách projektu a ich vzájomných vzťahoch a väzbách. Pri návrhu APROS-u sme vychádzali z projektu ako hierarchickej štruktúry, kde každá entita je podriadená istému vyššiemu prvku, a preto sme zvolili programovú realizáciu na báze MRP databankovej podpory, ktorá je schopná zobrazit takéto hierarchické vzťahy pomerne jednoduchým spôsobom.

Pri prvom pohľade na problematiku takéhoto katalógu je jasné, že ju musíme rozdeliť na dve časti. Prvá časť, logická, sa musí zaoberať dátami katalógu po stránke ich zatriedenia a vlastností, druhá, fyzická, musí riešiť implementáciu katalógu na počítači. Začnime teda logickými problémami:

Všetky dáta v katalógu možno roztriediť do niekoľkých skupín. Medzi takéto typické skupiny možno zaradiť napríklad identifikátory a kódy, parametre, vzťahy, súborov a podobne. Tieto skupiny - triedy dát - sú veľmi dôležitou zložkou návrhu katalógu, nakoľko na ich základe je možné riešiť hierarchické štruktúry /napr. JOB - STEP - PROGRAM - MODUL/ a každý reálny záznam je vlastne len jedným z výskytov danej triedy. Tieto fakty sa odzrkadľujú aj vo zvolenom systéme identifikácie, ktorý predpisuje pre každú entitu jednotné štrukturovaný osemznakový identifikátor, pričom prvý znak určuje riešený /sub/ systém, druhý prislusnosť entity k niektorému z typov. Identifikátor teda nemá vplyv na konkrétne zaradenie danej entity na niektorú špecializovanú pozíciu v systéme, resp. katalógu. Určuje však typ entity a teda aj atribúty, ktoré táto entita má mať uvedené. Pri tak rôznorodej náplni, akú katalóg APROS-u v našej implementácii má, je toto jedná zo základných vlastností

identifikátora. Zvolený identifikačný systém nám dovoľuje okrem hierarchických prístupov tiež prístupy horizontálne, tzn. získavanie informácií o výskytoch entít ľubovoľnej triedy.

Hierarchické väzby bolo možné riešiť dvoma spôsobmi. Prvá možnosť bola formou odkazov, kedy každá entita je doprevádzaná informáciou o prvkoch, ktoré sú jej podriadené, prípadne o entitách, ktoré sama rozvíja a bližšie špecifikuje. Úloha spracovateľského programu by spočívala v rozšifrovaní daných informácií a získaní údajov o potrebných entitách v katalógu. Nevýhoda takéhoto prístupu spočíva v tom, že každá zmena vyvoláva nutnosť manuálne prevádzaných zmien v iných entitách. Druhá možnosť bola prenechať starostlivosť o vyjadrenie a udržiavanie vzťahov entít na podporný systém, ktorý musí zabezpečiť jednoznačnú a správnu interpretáciu týchto vzťahov a navyše ich perfektnú údržbu a aktualizáciu. Riešenie, zvolené vo VSŽ, predstavuje kompromis medzi týmito dvoma možnosťami a je popísané v nasledujúcich častiach príspevku.

Pri popisovaní implementácie katalógu dát na počítači je potrebné v prvom rade spomenúť, že sme pracovali s počítačom IBM 370/145 a mali k dispozícii na dvoch pracoviskách /analytickom a programátorskom/ jeho obrazovkové terminály, preto bol prístup k dátam katalógu pomerne rýchly a jednoduchý. Z analýzy a návrhu katalógu bolo jasné, že budeme musieť ustúpiť od predpokladanej koncepcie katalógu APROS-u vo forme jedného súboru a jeho definitívny tvar je takýto:

Základné informácie o entitách a ich väzbách sú uložené v j a d r e k a t a l ó g u, ktoré je tvorené dvojicou matričného a štruktúrneho súboru podľa princípov MRP databanky. V matričnom súbore /s pevnou dĺžkou vety/ sú uchované tie informácie o entitách, ktoré umožňujú ich jednoznačnú identifikáciu /identifikátor/ alebo atribúty, ktoré musí mať uvedené každá entita systému /datum vzniku záznamu, autor/. Prístup k záznamom v jadre je možný priamo použitím identifikátora entity ako kľúča. Použitie štruktúrneho súboru umožňuje zobrazit' v jadre katalógu hierarchické vzťahy medzi entitami bez zbytočných duplicit a tieto vzťahy prehľadávať zhora nadol /explózia/ alebo zdola na-

hor /izplozia/. Štruktúrny súbor navyše zaisťuje jednoduchšiu aktualizáciu zaznamenaných väzieb.

Okrem povinných atribútov /s ktorých sa ukázal ako veľmi dôležitý kód autora/ má každá trieda entít svoju špecifickú množinu atribútov, ktoré musia byť v katalógu schránené a udržiavané /pre súbory dat dlhá zámena, blokovanie, pre transakcia, frekvencia spracovania atď/. Pretože každá entita ich musí mať jednoznačne priradené, zavedli sme pre ne súbor d o p l n k o - v ý c h i n f o r m á c i í s prístupom priamym na základe identifikátora entity. Väzba jadra a tohoto súboru je teda akousi obmenou získavania väzieb pomocou odkazov. Pretože štruktúra viet doplnkových informácií je neunifikovaná, tento súbor je organizovaný ako súbor s premenlivou dĺžkou vety.

Pre potreby dokumentovania je nutné uchovávať v katalógu tiež texty, ktoré podrobnejšie a ľuďom zrozumiteľnejšie určujú príslušnú entitu. Tieto texty je možné na základe ich obsahu presne špecifikovať a zatriediť /napr. v technickom projekte musí byť uvedený pre každú funkciu popis riešenia/ a teda priradiť im identifikačný kód. Sú podstatne dlhšie než významy doplnkových informácií a majú väčšiu variabilitu dĺžok, na druhej strane ich vnútorná štruktúra je veľmi jednoduchá. V AFROS-e sú uchované v súbore s p r i e v o d n ý c h t e x t o v s priamym prístupom podľa kľúča, ktorý vznikne zretazením identifikátora entity s identifikačným kódom textu.

Poslednú zložku katalógu tvorí súbor v o I n ý c h t e x t o v , majúci za úlohu uchovávať také texty, ktoré nie sú tesne zviazané s konkrétnou entitou. Ako príklad je možné uviesť formát obrasovky, prvé strany technických projektov, adresu nášho podniku, text tohoto článku a pod. Prepojenie na ostatné zložky katalógu v podstate neexistuje, príslušné texty musia byť explicitne špecifikované použitím špeciálneho identifikátora, ktorý pochopiteľne nemá žiadnu spojitosť s identifikátormi v ostatných zložkách katalógu.

3. Vstupy do AFROS-u

Pre zadávanie jednotlivých entít a ich atribútov sme zvolili postup, ktorý vyvoláva veľa diskúzií pri prvom zoznamení sa s AFROS-om. Mali sme niekoľko možností a po zvážení ich výhod a nevýhod sme dospeli k nasledujúcemu rozhodnutiu:

Vstup je možný v BATCH i ON-LINE spracovaní a nie je to priamo vstup do vlastného katalógu dát. V oboch prípadoch je potrebné vyplniť príslušný formulár /pri ON-LINE spracovaní ho predstavuje formát obrazovky, ktorý je pre daný typ entity špeciálny. Pri spracovaní tohoto formulára sa postupuje tak, že sa kontrolujú len formálne chyby v rámci neho, nekontroluje sa návaznosť zavádzaných dát na existujúce záznamy katalógu. Takýto prístup je /vzhľadom na to, že väčšina vstupov sa realizovala cez terminály/ pomerne rýchly a hlavne nenáročný na výpočtový systém, ktorý môže pracovať na prioritných úlohách; o to viac nárokov kladie na autora informácie. Presúva totiž naňho väčšiu zodpovednosť za logickú i obsahovú správnosť dát.

Výhodou takéhoto formulárového vstupu je skutočnosť, že doplnenie nových typov entít je pomerne jednoduchým procesom a spočíva v podstate iba v definovaní nového formulára spolu s kritériami na správnosť jeho vyplnenia bez ohľadu na už existujúce a zavedené typy. Jedným z najdôležitejších vstupov je formulár /obrazovka/, ktorý zabezpečuje vytváranie nových väzieb medzi existujúcimi entitami, takže

- bez potrebného zavádzania entity je možné vyjadriť jej nové väzby,
- väčšinu väzieb je možné zrušiť i vymazaním jediného väzbového záznamu.

Formálne správne záznamy sa zhromažďujú v zbernom súbore, ktorý býva najčastejším terčom kritiky. Vlastná aktualizácia katalógu je totiž realizovaná ako jeho znovunaplnenie a prevádzkuje sa v BATCH spracovaní, teda s istým časovým oneskorením. Identifikácia vety v zbernom súbore /presnejšie povedané dvojici súborov, ktorá obsahovo zodpovedá jadrú katalógu a doplnkovým informáciám/ je možná len na základe kľúča, ktorý jej záznamu

pridelí automaticky systém v čase jeho zaradenia do súboru. Nevýhoda takéhoto prístupu je jasná - výsledok aktualizácie sa prejaví až po určitej dobe.

Na druhej strane má použitie zberného súboru ako zdroja dát pre aktualizáciu katalógu tú výhodu, že bez zozloženia vlastných vstupných transakcií je možné definovať automatické generovanie niektorých dát /väsiab pre katalóg. Napríklad, v našej implementácii platí, že každý dátum je zobrazený šestmiestnym číslom v zložení RRMDD. Pri zavádzaní nového elementu, ktorý je dátumom, do katalógu uvádza autor iba jeho identifikátor a názov, povinne začínajúci reťazcom "dátum". Záznam sa potom uloží do zberného súboru, avšak pri naplňaní katalógu spôsobí vygenerovanie štruktúry /použijeme cobolovský zápis a mnemotechnické identifikátory/:

```
X   DATUM
X+1 ROK      PICTURE 99.
X+1 MESIAC  PICTURE 99.
X+1 DEN     PICTURE 99.
```

4. Výstupy z AFROS-u

Doteraz /možno neprávom/ najviac oceňovanou časťou AFROS-u sú jeho výstupy, pretože tvoria hmatateľný výsledok celého procesu.

Pri začiatku prác na projekte systému riadenia údržby nám bolo jasné, že ak ich máme úspešne zvládnuť a odovzdať, bude nutné vypracovať enormné množstvo dokumentácie.

Vytipovali sme asi 5 skupín dokumentov. V prvom rade sa jedná o jedinú publikáciu, zvanú technický projekt systému. Keďže ten musel byť ukončený ešte pred návrhom AFROS-u, nie je jeho spracovanie týmto našim produktom riadené. Projekt vo VSŽ predstavuje 8 aktivít /približne na úrovni subsystémov/, teda ďalšiu skupinu v dokumentácii tvorí 8 technických projektov aktivít. Podľa doteraz platných metodických pokynov budeme musieť vypracovať 8 vykonávacích projektov aktivít, z ktorých každý predstavuje tri dokumenty: používateľskú príručku, operátorskú príručku a programovú dokumentáciu. Okrem týchto základných dokumentač-

ných výstupov budeme musieť v rámci záväzkov, ktoré pre nás z riešenia vyplývajú, zaistiť navyše dokumentáciu typu referát, príspevkov na semináre, vyňatkov z projektov a podobne. Poslednú skupinu výstupov budú tvoriť také, ktoré nie sme schopní vopred špecifikovať.

Po takejto analýze výstupov sme dospeli k názoru, že bude veľmi účelné využiť informácie z katalógu aj na vytváranie vlastnej dokumentácie priamo, bez prepisovania na písacom stroji alebo na podobnom zariadení. Pochopiteľne, ohýbalo nás k tomu potrebné programové zabezpečenie.

Po niekoľkomesačných skúsenostiach s AFROS-om môže dnes výstupy rozdeliť podľa iného kritéria na výstupy úzko špecializované a na výstupy s širším použitím. Podľa tohoto rozdelenia máme aj dva druhy podpory.

Špecializované výstupy predstavujú rôzne prehľady, zoznamy a prehľadavacie transakcie, ktoré umožňujú vyberať z katalógu požadované údaje /u transakcií s relatívne malým objemom/ a odovzdať ich používateľovi vo vopred definovanom konštantnom tvare. Takýto výstup predstavuje napríklad popis štruktúry vety a atribútov jej položiek, ktorý vytlačí špecializovaný program po obdržaní identifikátora vety. Výstup použije analytik pri kontrole svojho návrhu a môže ho použiť aj programátor. Podobný výstup však programátor potrebuje aj v inom tvare, v programovacom jazyku pre záznam štruktúry a položiek vety do programu. Pri potenciálnom definovaní popisu vety viacerými programátormi hrozí, že prípadná zmena vety sa neodrkadlí vo všetkých jej popisoch. Existuje všeobecne známy postup, ktorý takejto situácii zabráni. V COBOL-e ho predstavuje zápis štruktúry vety do knižnice a jeho použitie COPY príkazom. Na založenia a zmenu tejto štruktúry sú však potrebné dáta, ktoré sú uložené v AFROS-e. Preto jednou zo zložiek programového vybavenia AFROS-u je aj program, ktorý konvertuje zápisy štruktúry vety do cobolovského tvaru.

Vypracovanie programov pre úzko špecializované výstupy je relatívne jednoduchou vecou, pretože ich vstupy a výstupy je možné špecifikovať a ostávajú nemenné prakticky po celú dobu životnosti systému. Omnoho väčším problémom je však tá časť softwa-

re, ktorá má zabezpečiť výstupy, ktorých forma a obsah sa nedajú s predstihom určiť, prípadne sa budú postupne meniť. Je tiež veľmi neekonomické písať osm programov pre tlač technických projektov aktivít, ktoré sú si síce podobné, ale nedajú sa dosť dobre zhrnúť do jedného programu.

Na riešenie takýchto výstupov sme vyvinuli špeciálnu podporu, ktorá nám dovolí vytlačiť špecializovanú dokumentáciu použitím jednoduchého jazyka. Táto podpora /editor/ má dve časti. Prvú, nepovinnú, tvorí špecifikácia oblasti záujmu. Pri jej vysvetlení sa musíme vrátiť ešte raz k rozlišovaniu dokumentácie, tentokrát z pohľadu štruktúry dát. Prvú skupinu tvoria tie dokumenty, ktoré štruktúrou dát v katalógu ovplyvnené nie sú. Takéto výstupy buď vôbec nepoužívajú katalóg /rôzne rozpis, správy/ alebo využívajú len jeho dátový fond bez ohľadu na štruktúru /tento príspevok/. Napríklad výpis zoznamu všetkých aktivít nevyžaduje od zostavovateľa znalosti o ich umiestnení v hierarchii, dostatočnou informáciou pre vstup do katalógu sú ich identifikátory. Ak však chceme vypísať zoznam všetkých funkcií v danej aktivite, omnoho schodnejšou bude explózia tejto aktivity na úrovni funkcií. Po rozpracovaní tejto myšlienky je zrejmé, že sa dá postup tlače dokumentov a ich štruktúra riadiť štruktúrou dát projektu, budeme však musieť špecifikovať počiatkový bod explózie a jej hĺbku. Takýto prístup má veľkú výhodu v tom, že každá zmena štruktúry dát v katalógu sa môže - bez zásahu do iných komponentov, zaisťujúcich tlač - okamžite odraziť v zmene štruktúry príslušného dokumentu.

Špecifikácia oblasti záujmu teda určuje v hierarchickej štruktúre dát katalógu tú časť, ktorá bude použitá pre riadenie štruktúry výstupu.

Druhú časť editora tvorí špecializovaný jazyk so svojím interpretérom, ktorý dovolí zapísať postup a podmienky tlače vo forme parametrov - príkazov jazyka, parametre prekontrolovať a preložiť do medzijazyka, ktorý bude neskôr interpretovaný, doplnený a po získaní dát z katalógu a ich pretransformovaní zabezpečí vytlačenie dokumentu.

Možnosti jazyka sa ukázali ako dostačujúce pre doteraz požadované výstupy. V jeho slovníku sú tieto dovolené typy príkazov

- COPY - príkaz pre doplnenie macro /analógia cobolovského COPY včítane REPLACING možnosti/
 DO - príkaz cyklu /riadený počtom alebo explóziou položky dát/
 EJECT SKIP - príkazy pre odstránkovanie /a tlačenie čísla strany a záhlavia/ alebo odriadkovanie
 END - ukončenie syntaktickej konštrukcie /cyklu/
 ENTER EXIT - začiatok a ukončenie programu
 GEN NOGEN - príkazy pre tlač vkladaného macro alebo jej potlačenie
 HEAD HEND - začiatok a ukončenie definície hlavičky
 MACRO - začiatok definície macro
 PAGE - nastavenie stránkovania
 PRINT - príkaz pre tlač. Podľa volieb umožní tlač textovej konštanty, časti dát z jadra katalógu alebo doplnkových informácií, sprievodného textu, prípadne voľných textov alebo obsahu registrov.
 SET - nastavenie registrov
 VALUE - nastavenie hodnoty parametrov v macro /možno ich meniť v COPY príkaze/.

Tento pomerne obmedzený počet príkazov je doplnený deviatimi pracovnými registrami, tabuľkami generovanými z dát katalógu a ich indexami /relatívnymi voči štartovnému bodu/ a tiež figuratívnymi konštantami dátum, mesiac, rok a čas, ako aj počítadlami strán, riadkov a doteraz vytlačených znakov riadku.

Samotná tlač dokumentu prebieha tak, že autor špecifikuje jeho formát /napr. formát technického projektu aktivity je potrebné špecifikovať iba jedenkrát/, tento sa doplní o príslušné parametre, špecifické pre výskyt daného dokumentu /napr. technický projekt databanky a rozvrhovania opráv sa líšia textami záhlaví/, určia sa príslušné rozsahy záujmu v katalógu dát a v prípade, že program špecifikujúci definitívny formát je syntakticky správny, vytlačí sa príslušný dokument, v opačnom prípade sa musia opraviť indikované chyby, program sa znovu preloží a interpretuje. Pri tlači bude program automaticky doplnený aj o ilustrácie, ktoré sú citované v sprievodných tex-

toch a uchované vo voľných textoch. Súčasne sa zabezpečí stránkovanie a riadkovanie v tvare normalizovaného výstupu A4, ako aj tlač definovaných hlavičiek.

5. Súčasný stav APROS-u, jeho nároky a nevýhody

Ako bolo na začiatku spomenuté, APROS je ešte stále v štádiu vývoja. Najväčšia časť práce na ňom bola urobená v čase, keď sa pripravovali a tlačili technické projekty aktivít, teda v čase, keď už projektovanie bežalo naplno. Pretože jeho autori mali v rámci projektu aj iné povinnosti, postup ďalšieho vývoja bol veľmi nepravidelný, v dnešnom stave je katalóg projektu /a teda APROS-u/ naplnený údajmi o

- funkčnej dekompozícii projektu
- databanke a dátovom fonde
- identifikátoroch a kódoch
- ilustráciách.

Bielym miestom je teda ešte oblasť vlastných projekčných prác a ich postupu. APROS-om boli vytlačené všetky technické projekty aktivít a tiež niekoľko príspevkov na rôzne jednanía a semináre /napr. väčšina príspevkov v /2/, materiál /3/ atď/. Autori technických projektov sa zhodli na tom, že námaha spojená s vkladaním dát do počítača bola dostatočne kompenzovaná veľkou pružnosťou prevádzaných zariadení dokumentácie s vylúčením ďalších ľudských elementov /písarka/. Ak to bude technicky možné, pokúsime sa aj tento príspevok uverejniť v ostravskom zborníku vo forme, ktorú poskytuje APROS. V najbližšom čase sa plánuje tlač vykonávacích projektov aktivít.

Keď hovoríme o nárokoch APROS-u, musíme pripomenúť, že nami prevádzkované riešenie je v značnej miere orientované terminálove, teda možnosť terminálového prístupu je vítanou, nie však vyžadovanou. MRP podpora tvorí základnú časť databanky, a u nás je doplnená o STELA INTERFACE, čo však opäť nie je nutným predpokladom zavedenia a prevádzky APROS-u. Nároky na vnútornú pamäť počítača nie sú väčšie než pri štandardnom spracovaní MRP a teda vyhovujú pre veľkú časť počítačov EC. Podmienkou je operačný systém OS alebo systém s ním kompatibilný. Pokiaľ

sa jedná o diskovú pamäť, táto je závislá na veľkosti sledovaného /sub/systému. Pri našom pomerne rozsiahlom projekte dosahujú nároky približne hodnoty 8 MB diskového priestoru.

V čom vidíme nedostatky APROS-u? APROS vznikol a vyvíjal sa živšie podľa momentálne sa objavujúcich potrieb a požiadaviek projektu, nemá teda žiadny teoretický základ. Všetky technické projekty a rezersa, robené na základe explózie aktivity, predpokladali rovnakú relatívnu úroveň, na ktorej prechádza riešenie z fázy úlohy na funkciu. Doteraz nie je možné /a pravdepodobne ani to možné nebude bez sbytočne veľkej spotreby papiera/ meniť šírku tlače, preto ilustrácie formátov obrázkov a zostáv musia byť uvádzané vo forme výrezov.

Na druhej strane však APROS ako jeden z prvých u nás známych systémov tohoto typu sa vo svojom návrhu zaoberá otázkou obsahu vstupov a výstupov a stretol sa na väčšine akcií, kde bol prezentovaný, s pochopením u poslucháčov.

Pri oponentúrach technických projektov bol veľmi často spomínaný fakt, že tlač dokumentácie s použitím počítača je veľmi nákladná záležitosť. S touto pripomienkou snád možno čiastočne súhlasiť, avšak súčasne je potrebné zobrať do úvahy ďalšie faktory. Uvediem len niektoré z nich:

Projektanti strávia veľa času zadávaním hodnôt do počítača a pri klasickom spôsobe dokumentovania ich musia opäť prepisovať ako podklady pre písarky. Tu sa stráca časť ich pracovnej kapacity. Ďalšie časové straty vzniknú pri kontrole a oprave prepísaných textov. Okrem drobných preklepov, ktoré je možno opraviť na mieste, sa neraz vyskytnú prípady vyžadujúce prepísanie celej strany alebo kapitoly. Opäť teda narastajú nároky na ľudskú prácu.

V prípade použitia strojovo vytváranej dokumentácie odpadajú medzietapy u písarek. Strojový čas na spracovanie jedného dokumentu je krátky /u technických projektov aktivít 2 - 4 minúty/ a navyše pri eventúálnej potrebe ďalších výtlačkov je možné text nahráť na magnetickú pásku a neskôr ho vytlačiť. Ďalšími nezanedbateľnými faktormi je možnosť okamžitej zmeny štruktúry dokumentácie, ktorú som už spomenul a tiež možnosť deprogramovať výber strán pre tlač. Cena dokumentácie by mohla

poklesnúť, ak by bol celý APROS prenesený na menší počítač, čo je jednou z možností jeho ďalšieho vývoja.

Literatúra

- /1/ Technický projekt aktivity OB projektu Použitie moderného systému riadenia údržby v hutníctve, VSŽ Košice, august 1981
- /2/ Použitie moderného systému riadenia údržby v hutníctve, zborník prednášok str. 170 - 178, Dom techniky ČSVTS, Košice, 1981
- /3/ APROS - jazyk pre dokumentáciu, nepublikovaná interná dokumentácia VSŽ
- /4/ Racionalizácie projektové a programové prípravy, MAA 10/81