T V O R B A R O Z S Á H L É H O P R O J E K T U -
P Ř E D P R O J E K T O V Á P Ř Í P R A V A

Ing. Petr Veselý

1. Výchozí situace

Existuje projekt, který se využívá pro zpracování téměř jednoho milionu pracovníků v oblasti "práce a mzdy". Projekt je velmi rozsáhlý. Jeho programová část obsahuje, nepočítáme-li využívaná komponenty základního programového vybavení, přibližně 200 makroinstrukcí, 650 modulů, 200 fází. Projekt je rozsáhlý i z hlediska služeb, které svým uživatelům poskytuje. Lze říci, že oblast mzdová je řešena komplexně, oblast personální a kádrová částečně, oblast sociální politiky jen okrajově. Je zajištěna vazba na okolí (říkejme jí poloautomatická) - pro statistické výkaznictví, půjčky a spoření (spořitelny), pojištění (pojišťovny), požadované podklady pro kontrolní orgány ÚSNP a ministerstvo financí, podklady pro tvorbu pokladního plánu SBČS. Výčet není úplný a nebudu v něm pokračovat - pro charakteristiku "uživatelských možností" je dostatečný. Programový aparát je realizován, z pohledu měřítek platných nyní, na minipočítači. Provozuje se většinou na počítačích čs. výroby EC 1021 pod operačním systémem MOS, v menší míře, a to pouze jeho část, na počítačích EC 1030 a EC 1033 pod operačním systémem DOS. Počítač EC 1021 má 64 kb operační paměti, maximálně lze využít 4 stojany s magnetickými disky (1 disk má kapacitu 7,25 Mb) a 4 stojany s 1/2 palcovými magnetickými páskami. Základní soubor má organizaci, která připomíná databázovou stromovou strukturu. Základní soubor obsahuje v současnosti pro největší uživatele více než 15 000 vět (pro 1 pracovníka je 1 věta). Nejdelší věty mohou být dlouhé až 3 000 byte. Značná délka vět je způsobena tím, že ty údaje, které se vyvíjejí s časem, se udržují alespoň s jednoletou historií - programy pak dokáží reagovat na vstupní údaj zasahující do minulosti tak, že automaticky provedou potřebné přepočty tak,že výsledky přepočtů do minulosti zanesou do věty základního souboru a případné přeplatky či nedoplatky zúčtují ve vyúčtování zpracovávaného měsíce. Řešení projektu je poplatné technice, pro kterou byl tvořen i poněkud "spěšnému" tempu řešení. Nicméně projekt úspěšně pracuje a dále se rozšiřuje.

Projekt se provozuje úplně ve výpočetních střediscích PVT. Protože počítače 3. generace postupně dožívají a instalují se počítače 3,5-té generace, je zapotřebí zabezpečit zpracování na těchto nových počítačích.

Pro tvorbu nové verze projektu jsou stanoveny tyto souhrnné požadavky:

a) Požadavky uživatele:

Zvýšit kvalitu služeb poskytovaných projektem, oproti současné verzi projektu. Tedy jednak odstranit nedostatky současné verze, o kterých se sice ví, ale které je velmi obtížné odstranit, vzhledem ke koncepci a vzhledem k hardwarovému a softwarovému prostředí, v nichž je současná verze implementována. Jednak realizovat v současné verzi dosud nerealizované funkce. Jednak akceptovat požadavky uživatelů na jiné formy vstupů a výstupů (požadavky plynou ze zkušeností s využíváním současné verze projektu). V neposlední řadě reagovat na měnící se situaci ve vybavenosti uživatelů prostředky pro pořizování dat, předzpracování dat, prostředky umožňující DPD resp. DZD.

b) Požadavky provozovatele:

V maximální možné míře využít nových možností hardware počítačů 3,5-té generace a operačního systému DOS-3, resp. DOS-4 s cílem co nejekonomičtěji zpracovávat a přitom připustit velmi odlišné režimy zpracování pro jednotlivé uživatele projektu. Při řešení použít všech dostupných prostředků, které usnadní snažší aktualizaci projektu oproti obtížné aktualizaci současné verze projektu (plyne ze způsobu naprogramování a z nedostatečné dokumentace). Dále zabezpečit "jinned" převod na počítači 3,5-té generace a řešit tak jejich vytížení.

2. Způsoby řešení

Požadavek provozovatele na "okamžitou" připravenost zpracování na počítačích 3,5-té generace je v kolizi s ostatními požadavky na novou verzi projektu. Tato skutečnost si vynucuje dva způsoby řešení, které budou probíhat souběžně.

První způsob spočívá v emulaci současné verze pod operačním systémem DOS-3 resp. DOS-4. Pro toto řešení existuje v PVT vytvořený program, který emuluje operační systém MOS do prostředí počítače 3,5-té generace/DOS-3 resp. DOS-4. Tento prostředek se použije a umožní současnou verzi projektu bez jakýchkoliv úprav provozovat i na nově instalovaných počítačích 3,5-té generace.

Druhý způsob spočívá v úplném zpracování takového nového projektu, který uspokojí všechny požadavky uvedené v kapitole "Výchozí situace".

V dalším textu tohoto příspěvku se budu zabývat jen tímto druhým způsobem.

3. Postupy práce

Předpisy platné v PVT Praha určují základní etapy, kterými musí tvorba projektu projít. Etapy na sebe navazují, schválení (oponentura se schvalovacím protokolem) výsledků práce jedné etapy podmiňuje zahájení prací na následující etapě. Pro každou etapu je stanovena forma a obsah dokumentu, který formálně etapu ukončuje.

Uvedu jen nejpodstatnější:

- stadium tvorby ideového návrhu (dokumentem je projektová studie),
- stadium tvorby technického projektu (dokumentem je technický projekt),
- stadium tvorby prováděcího projektu (dokumentem jsou provozní dokumentace, uživatelská příručka).

Akceptuje se přirozený průběh prací, kdy se některé práce příslušející do vyššího stadia projektových prací provádějí již v nižším stadiu tvorby projektování.

4. Stadium ideového návrhu

Projektovou studii, na kterou již byl vystaven schvalovací protokol, jsme zpracovali s cílem uvést hlavně z uživatelského hlediska podstatné záměry nové verze projektu. Přitom jsme si byli vědomi, že technické řešení některých záměrů ještě neznáme.

Již v tomto stadiu je nutné důkladně rozpracovat některé problémy. Uvedu ty nejpodstatnější:

a) Vymezení základních problémových oblastí, na které je třeba orientovat řešení projektu. V našem případě to znamenalo vymezit základní funkce projektu - tedy funkce současné verze plus nově požadované funkce. Zdrojem informací byla znalost současné verze projektu, zadání gestora úlohy, zadání budoucích uživatelů a naše jednání se skupinou vybraných uživatelů současné verze.

b) Stanovení správných forem a postupů v těch částech projektu, s nimiž přichází do styku uživatelé. Pro řešení tohoto problému jsme zvolili cestu jednání se zástupci uživatelů. Tato forma, jejíž použití je zřejmě nezbytné, se ukázala jako dosti problematická. Volili jsme tento postup:

- Problematiku, o které bylo třeba jednat, jsme rozdělili do 3 skupin (vstupy a kontroly, struktura organizace a výstupy, vazby na okolí úlohy).
- O každé skupině jsme jednali na samostatných jednáních. Přitom jednání o každé skupině proběhla na čtyřech různých místech v republice, a to s lidmi, kteří u různých uživatelů s projektem pracují, resp. budou pracovat. Pro tato jednání měli uživatelé k dispozici rela-

tivně rozsáhlý materiál, ve kterém jsme navrhovali i některé alternativy konkrétních řešení. Jednání probíhalo neformálně, vyžadovali jsme ale od účastníků zaslání jejich názorů a připomínek písemně. Po zpracování těchto materiálů jsme vyvolali páté jednání, jehož cílem bylo rozhodnout o sporných požadavcích.

Dosud proběhlo 11 jednání. Výsledky neodpovídají úsilí, které jsme této akci věnovali. Je velmi těžké přesvědčit konkrétního pracovníka, aby přesně určil a zdůvodnil své potřeby. V našem případě hrály svou nepříznivou roli navíc dvě skutečnosti. Uživatelé jsou "zatíženi" tím, že si zvykli na současnou verzi a přizpůsobili se i nesprávným postupům. A potom náš projekt se používá v celé republice a u různých uživatelů různě. Součástí podkladů pro jednání o struktuře organizace a výstupech byl náš požadavek na písemné zpracování materiálu jako incidenční tabulky, ve které (a v poznámkách k ní) bude provedeno přiřazení výstupních informací potřebných pro plnění jednotlivých funkcí organizace.

Je možno říci, že přes podrobný návod i příklad použití jen velmi málo odpovědí uživatelů alespoň částečně splnilo náš záměr.

c) Zjištění očekávaného vybavení uživatelů technickými zařízeními s perspektivou budoucích deseti let. K tomuto bodu se nám nepodařilo získat dostatečně přesné informace, i když jsme odpovědi hledali i na úrovni PŘ i u gestorů úlohy, resp. jednotlivých aplikací (ministerstvo vnitra, ministerstvo školství, KNV, ...).

Závěrem k výše uvedeným problémům je možno říci, že při budování rozsáhlého projektu, který bude sloužit jako průřezová úloha je nutné jako nejdůležitější hledisko vyzdvihnout otevřenou architekturu projektu, která umožní projekt trvale a relativně snadno upravovat.

5. Stadium tvorby technického projektu

V této kapitole se zmiňuji o těch problémech, které byly a jsou pro nás nejzávažnější. Naprostá většina z nich je nedořešená, resp. nejsme přesvědčeni o tom, že řešení, které jsme zvolili je optimální.

Technický projekt by měl být dokument, který obsahuje úplné (alespoň ve většině případů) popisy řešení všech zásadních problémů úlohy. Protože problémů k řešení je značné množství a jsou velmi různorodé, zvolili jsme jakési "mezistádium", které bylo ukončeno dokumentem "Podrobné podklady pro tvorbu technického projektu". Tento materiál, který obsahuje přibližně 250 stran textu, napsala skupina tří pracovníků s cílem popsat všechny známé problémy, které je nutno řešit, navrhnout řešení a rozpracovat případné alternativy řešení. Každému

problému je věnována jedna kapitola, v jejímž závěru jsou uvedeny nároky na pracovníky, kteří mohou problém vyřešit (jejich odbornost, časové nároky) a návrh jednotlivých kroků postupu řešení. Závěrečná kapitola práce obsahuje soupis všech kroků, které by se měly provést při řešení všech problémů a jejich časové návaznosti. Materiál je podkladem pro rozdělení práce mezi odborníky na příslušnou problematiku.

Dále se budu zabývat jednotlivými kapitolami "Podrobných podkladů".

A. Provedení datové a funkční analýzy zpracovávaných objektů a prováděných funkcí, včetně definice činností úlohy

V kapitole jsou popsány metody, jejichž použití se předpokládá a další prostředky potřebná v jednotlivých krocích řešení. V kapitole je hrubý slovní datový popis všech typů objektů, které mají být předmětem zpracování (7 typů objektů), je zdůrazněna zásada, že informace se musí udržovat na té úrovni, na které vzniká, resp. které se týká (na př. číselníky s celostátní platností musí být udržovány a do úlohy distribuovány z jednoho místa, v našem případě z PŘ PVT).

Funkční a datová analýza probíhá v těchto fázích:

a) Zobrazení základních funkcí úlohy a vazby (data) mezi nimi. Použijeme metody HIPO. Případně v prvním kroku použijeme dokumentačních prostředků "Structured Design". Zdrojem informací je projektová studie.

b) Datová analýza. Použijeme metodu HIT, nebude-li práce efektivní, metodu opustíme a použijeme jiného prostředku. Zdrojem informací je současná verze projektu, katalogy JÚZO, některé jiné projekty z oblasti práce, mezd a sociálních věcí, rozhovory se zástupci uživatelů současné verze a zástupců rezortů a institucí.

c) Funkční analýza s použitím datových schemat, vzniklých v předchozí fázi. Funkce popsané ve fázi a) se rozvedou až do úrovně, kdy už není vhodné je dále členit. Použije se HIPO.

d) Úprava datových schemat, vzniklých ve fázi b) podle poznatků fáze c). Kompletace údajů pro implementaci datové základny s použitím typů datových struktur, uvedených v kapitole A. V této fázi se rozhodne o tom, které údaje, i když je možno je odvodit z jiných údajů, se budou do datové základny ukládat.

e) Úprava funkčního řešení, které vzniklo ve fázi c) v důsledku změn datových struktur, které se provedly ve fázi d). Dále se provedou úpravy - doplnění funkčního řešení z fáze c) o technologicky nutné algoritmy.

b. Stanovení základních režimů zpracování

Úkolem je vytipovat všechny možné režimy, na základě kterých budou probíhat různé varianty zpracování. Projekt bude muset mít prostředky, které umožní deklarované režimy provozovat. Režimy jsou v přímé vazbě na stanovené funkce úlohy. Úloha je řešena jako obecná. Z obecných prvků se budou vytvářet jednotlivé "aplikační projekty". Každý "aplikační projekt" bude mít své možnosti režimů, odpovídající funkcím, které budou z obecného řešení v aplikačním řešení zahrnuty.

Kriteria pro určení režimů jsme stanovili na základě diskusí s uživateli. Stanovili jsme 9 základních kritérií s přibližně čtyřiceti možnostmi. Výběr režimů se projeví při volbě metody tvorby "aplikačních projektů" a při tvorbě prostředků pro provozování úlohy.

C. Stanovení formy vstupů a výstupů

Stanovení formy vstupů je velmi podstatné z hlediska dobrého využívání úlohy uživatelem. Cílem je dosáhnout toho, aby se zadávaly jen prvotní informace, a to v co nejpřirozenějším tvaru. V tomto směru se jevilo být vhodné provést do současné verze podstatné zásahy. Zmíním se jen o těch podstatných hlediscích, která je nutné při návrhu formy vstupů zohledňovat. Hledisko popisovaných objektů zabezpečí definovat všechny prvotní vstupní údaje, které jsou nutné pro úplný datový popis každého objektu. Hledisko řízení zpracování, které slouží k definování všech údajů, které na vstupu identifikují popisovaný objekt, a které udávají, jak naložit s dále uvedenými prvotními datovými informacemi. Poslední hledisko je hledisko média. V tomto směru projekt počítá s tím, že uživatelé budou moci předkládat úloze vstupní údaje na formulářích, a to na formulářích, které nebudou snímány opticky. Tento způsob se bude realizovat proto, že uživatelé úlohy používají formuláře pro vedení prvotních údajů a pro vstup do úlohy je přepisují do předepsané formy vstupů. Uživatelé budou moci jako vstupy předkládat ručně vypsané formuláře, které stejně vyplňují. Formulář bude mít děrovací předpis. Formulář vstoupí do úlohy buď přes obrazovku terminálu v panelovém režimu, nebo na libovolném médiu ve volném formátu. V tomto případě je vhodný prostředek pro práci se vstupy GIMV (Systém mapování logických vět, který je součástí DOS-4) a dále možnost použít "libovolného" média jako terminálového souboru s formátem free.

Stanovení formy výstupů, resp. definitivní formy není v této etapě důležité. Pro naše záměry bylo spíše nutné zjistit, zda je správný předpoklad, že musíme počítat s těmito variantami výstupů:

- stručná zpráva předaná na terminál
- tabulka souhrnných informací předaná na obrazovku terminálu, nebo vytištěná (vyděrovaná) na libovolném médiu
- tradiční rozsáhlá sestava vytištěná na papíře, nebo na magnetickém médiu, které se otiskne na tiskárnách COM či na papír na tiskárnách u uživatele.

Přitom vycházíme z toho, že máme k dispozici všechny elementární výstupní informace, ze kterých se libovolné potřebné výstupní finální informace dají poskládat.

D. Kontroly

Ze zkušenosti se současnou verzí projektu víme, že oblast kontrol je jedna z nejdůležitějších, ne-li vůbec nejdůležitější z hlediska "průchodnosti úlohy". Toto hledisko je pro uživatele snad to nejpodstatnější. Proto jsme věnovali tomuto problému značnou pozornost. Z hlediska kontrol jsme rozdělili struktury vstupních údajů na 10 různých datových struktur a nad nimi jsme definovali 12 různých druhů formálních a logických kontrol s uvedením dalších postupů při objevení se chyby. Takto byla navržena první fáze kontrol - kontroly nahrávaných vstupních údajů. Při jednáních s uživateli jsme požadovali, aby dopracovali námi navržené kontroly nepřípustnosti některých vstupních údajů pro objekty stanovených vlastností a stanovili priority, které platí při souběhu různých vstupních údajů k témuž časovému okamžiku. Kontroly vstupů jsou navržené tak, aby byly úplné. Dobrým softwarovým prostředkem, který zajistí kontrolu formátů i hodnot vstupních údajů, je SMLV v DOS-4, který můžeme zařadit mezi příkaz vstupu, resp. výstupu v programu a LIOCS.

Druhá fáze kontrol jsou kontroly vypočtených údajů, které jsou součástí výpočtových programů a předávají se uživatelům k posouzení ještě před fází tisku výstupů.

E. Stanovení pravidel a postupů pro tvorbu programů

Tato kapitola je v citovaném materiálu nejdelší. Problémy v ní naznačené považujeme za velmi důležité vzhledem k budoucí údržbě projektu. Pracovně je možno shrnout je pod název "podmínky správného návrhu programů". V PVT se dosud nekladl potřebný důraz na strukturované programování a související návody, proto jsme se nemohli opřít o žádnou existující podnikovou metodiku a museli jsme čerpat z literatury.

Problematiku jsme rozdělili do pěti částí:

a) Podmínky správného návrhu modulů. V této části jsme uvedli:
   - obecně známé zásady při postupu dekompozice programu do modulů
   - definici a vliv soudržnosti (STRENGTH) modulů, návrh realizace správné soudržnosti
   - definici a vliv vzájemnosti (COUPLING) modulů, návrh realizace správné vzájemnosti
   - další zásady platné při psaní modulů, jako jsou zásady správné velikosti, 1 vstup a 1 výstup, minimální počet skoků, samodokumentárnost, přehlednost atd.

b) Zásady platné v jednotlivých programovacích jazycích. V této části jsme uvedli jednak řešení jednotlivých řídících programových struktur (IFELSE, DOWHILE, DOUNTIL, CASE), jednak další pravidla, která se používají pro správnou volbu příkazů jazyka, resp. sledu příkazů jazyka a pro správnou formu zdrojového textu modulu. Zásady jsou uvedeny pro ty programovací jazyky, které přicházejí v úvahu pro psaní modulů projektu.

c) Volba programovacího jazyka. V PVT je v platnosti metodika, kterou se určuje jako hlavní programování jazyk Cobol. Metodika je platná již řadu let. Ledacos se od jejího vydání přihodilo. Chtěli jsme navrhnout k použití jazyk/jazyky, které budou nejlépe vyhovovat řešené problematice. Pro tyto úvahy jsme vyšli z těchto předpokladů:

- Datové sety budou nachystány tak, že požadované výstupy nebude nutné "tisknout" programy psanými na míru, ale bude je moci "snad vyrábět i uživatel" pomocí parametrických jazyků jako je CULPRIT a pro výběr informací z datových setů jazyk IQF.
- Vlastní komunikaci s datovými sety budou pro potřeby ostatních programů zajišťovat komunikační moduly, případně psané i v Assembleru, které použijí služeb SMLV.
- Pro psaní ostatních modulů (ty budou pracovat ve fázích nahrávání, kontrol a výpočtů) je nutno zvolit jazyk, který umožňuje psát dobře strukturované programy, umožňuje psát relativně malé moduly, které se budou samostatně přeskládat. Dále musí umožnit předávání datových struktur mezi několika úrovněmi modulů ve vrstvené struktuře programu.

Z výše uvedených hledisek jsme, ovšem bez hlubší znalosti jazyků, ve kterých nejsme zvyklí programovat (Pascal, Pl/1) posuzovali COBOL, PL/1, Pascal, Assembler. Původně náš návrh zněl: Pascal. Důvodem bylo, že jazyk umožňuje efektivní psaní programů a navíc, oproti jiným jazykům, automatické provádění kontrol na úrovni kompilace i

v průběhu zpracování programu. Poněkud obtížnější je přehledné předávání datových struktur. Tato část je jedna z těch částí citovaného materiálu, ke kterým je nutný odborný posudek experta.

d) Ladění a testování. Zvolení postupů, které umožní efektivně ladit a testovat programy úlohy, považujeme za velmi důležité. Zkušenosti, které máme se současnou verzí projektu nejsou nejlepší. K nedostatečné dokumentaci se řadí i fakt, že při tvorbě programů se neuvažovalo s tím, jak se bude provádět ladění a testování.

Byli jsme nuceni vycházet z toho, že budujeme projekt s otevřenou architekturou. Musíme očekávat, že programy budou trvale živé. Budou se objevovat trvale požadavky, které se promítnou jak ve změnách v již existujících modulech, tak i ve vytváření nových modulů, které bude nutno do existující struktury modulů zabudovat.

Při řešení tohoto problému jsme se obrátili na doc. J. Hořejše z UJEP Brno a jednali jsme s těmi pracovníky VÚMS Praha, kteří jsou tvůrci prostředků SNAP a SMLV.

Z prací, s nimiž jsme se seznámili, uvedu jen ty, které budeme moci dále využít.

Seznámili jsme se s metodou ADT (abstraktní typ dat), která umožňuje přesnou specifikaci vnitřní i vnější činnosti modulu, a to pro účely implementace, dokumentace i ladění a testování. Zdrojem informací byl materiál, který pro nás zpracoval doc. J. Hořejš.

Probrali jsme praktické návody, které pro určení vhodných testovacích dat, v závislosti na detailních funkcích, které je třeba testovat, uvedl Samuel T. Redwine, Jr. ve své práci An Engineering Approach to Software Test Data Design, která byla zveřejněna ve sborníku IEEE Transactions on Software Engineering, VOL. SE-9, NO.2, March 1983.

Není možno říci, že probíraný problém jsme uzavřeli. Zatím jsme dospěli k dále uvedeným dílčím závěrům.

d1) Použijeme SNAP jako dokumentační a ladící prostředek na úrovni "vrcholového řídícího" aparátu. SNAP, vyberu-li nejpodstatnější aspekt tvorby modulární struktury projektu - tedy interface mezi jednotlivými moduly, dokáže automaticky vytvořit hierarchické (funkční) schema modulu projektu. Výsledkem této činnosti je jednak podklad dokumentační, jednak vznik interního systémového prostředku, který nadále umožňuje v rámci vzniklé hierarchie s projektem, resp. jeho částmi pomocí SNAPu pracovat (ladit, testovat). SNAP provádí před vlastním překladem programových modulů řadu kontrol modulárního systému. SNAP hlídá a upozorňuje na změny, které programátor provedl ve struktuře

systému modulů, vzhledem k jejich předcházím verzím. CMAP poskytuje různorodé dokumentační služby, pomocí kterých lze například i tisknout některé části projektové dokumentace.

d2) Použijeme po určitém dopracování SMLV jako dokumentační a kontrolní (v průběhu zpracování) prostředek pro popis a práci s daty na vstupu resp. výstupu. SMLV umožňuje vedení přehledné, úplné a částečně automatizované dokumentace datových struktur, centralizaci těchto popisů a jejich snadnou distribuci do jednotlivých částí systému. SMLV zjednodušuje práci s datovými větami, ať už jsou příslušné soubory fyzicky organizovány jakkoliv, zajišťuje při přenosu dat do(z) programu rozsáhlé kontroly.

d3) Využijeme principů metody ADT pro přesnou specifikaci vnitřní činnosti modulů a podle vyspecifikovaných detailních funkcí a v nich zpracovávaných datových typů se budou určovat nezbytně nutná testovací data.

e) Správné řešení ve virtuálním prostředí. Pokusili jsme se stanovit několik pravidel, kterými se doporučují některé programátorské postupy, resp. kterými se doporučuje vyvarovat se některých programátorských postupů.

F. Logické struktury, fyzická organizace dat

Na základě datové analýzy vytvoříme "slovník dat". Každá položka slovníku bude mít dostatečné informace k tomu, aby všude tak, kde se s datovým prvkem pracuje, bylo možno použít popisu datového prvku ze slovníku dat. Neznáme přesnou představu jak automatizovaně slovník udržovat. Určité možnosti skýtá SMLV.

Logické struktury a nejvhodnější fyzická organizace dat vyplynou až z ukončené datové a funkční analýzy. Přestože ještě nejsme tak daleko, provedli jsme odhad, z jehož korekcí počítáme. Ve spolupráci s pracovníky VÚIS Praha jsme implementovali model nejkomplikovanější části popisovaného objektu úlohy (tedy objektu pracovník) pomocí DZS-25 a dále model předpokládaných nejfrekventovanějších výstupních datových struktur pomocí indexsekvenčních souborů se sekundárními klíči.

G. Stanovení správných algoritmů výpočtů

Zde jde zřejmě o specifický problém našeho projektu. Algoritmy v oblasti práce, mezd a sociálních věcí, zvláště zajišťují-li automatické přepočítání při opravách i do "dávné minulosti" jsou velmi rozsáhlé. Vytvořit je z předpisů, které jsou mnohdy pro různé rezorty

různé (dokonce i pro různé podniky v jednom rezortu) je práce časově velmi náročná. Programy současné verze nejsou dostatečně popsány. Problém stanovení správných algoritmů výpočtů nemáme uzavřen. Ať již použijeme jakékoliv zdroje, pomocníkem nám budou principy metody ADT.

H. Stanovení správného postupu provozování úlohy

Jsme ve fázi pouhé deklarace tohoto problému. Je jasné, že projekt bude muset dokázat provozovat efektivně a bezpečně racionální kombinace režimů, o nichž jsem se zmínil v odstavci B. V DOS-4 bude k dispozici prostředek "BATCH-NET", který umožní popsat projekt jako síť dávek . Jednotlivé uzly sítě se zapojují při zpracování dynamicky v závislosti na splnění určitých podmínek. O použití tohoto prostředku uvažujeme.

I. DPD a DZD, vazba na zpracování na mini resp. mikropočítačích

Tento problém jsme rovněž otevřeli s vědomím našich malých znalostí, vzhledem k jeho závažnosti. Jeho závažnost spočívá v tom, že projekt musí být schopen spolupracovat se všemi možnými typy zařízení, kterými budou uživatelé vybaveni. Definovali jsme 5 tříd zařízení. Pro každou třídu jsme stanovili činnosti, které bude na nich moci uživatel provozovat. Z toho vyplynuly požadavky na projekt, resp. na určení bodů napojení.

Problémy DPD, resp. DZD jsme rozpracovali jako podklad pro řešení těchto konkrétních úkolů:
- navrhnout postup návrhu DPD, včetně výběru technických prostředků
- navrhnout způsoby zkoušení DPD
- programově zabezpečit úlohu pro interaktivní režim zpracování, resp. pro příjem a odesílání zpráv
- navrhnout zabezpečení konzistence distribuované datové základny.

Pro pořádek uvedu jen názvy ostatních kapitol diskutovaného materiálu:

J. Stanovení způsobu řešení jednotlivých aplikačních projektů podle potřeb konkrétních uživatelů
K. Specifika hospodářských organizací
L. Návrh způsobu vyhodnocení ekonomické efektivnosti
M. Personální zabezpečení
N. Projektová dokumentace
O. Vývoj operačního systému
P. Postup aktualizace projektu při jeho rutinním využívání
R. Další postup řešení

6. Závěr

V příspěvku jsem naznačil směr, kterým jsme se ubírali při přípravě projektu, který jsem si troufl nazvat "rozsáhlý". Naší snahou bylo na samém začátku deklarovat a navrhnout řešení těch problémů, které nepromyslí-li se včas, ztíží později průběh řešení projektu. Domnívám se, že je nutná spolupráce expertů, kteří nejsou přímo na projektu zainteresováni. Zcela na závěr bych chtěl podotknout, že znění některých kapitol, resp. tvrzení v nich obsažená, mohou působit dojmem naprosté jistoty. Bohužel ve většině případů tomu tak není.