

<https://doi.org/10.3176/hum.soc.sci.1981.2.09>

EESTI NSV TEADUSTE AKADEEMIA TOIMETISED. 30. KOIDE
OHISKONNATEADUSED, 1981, NR. 2

ИЗВЕСТИЯ АКАДЕМИИ НАУК ЭСТОНСКОЙ ССР. ТОМ 30
ОБЩЕСТВЕННЫЕ НАУКИ, 1981, № 2

A. KALLISAAR

MÕNINGAID ANDMETÖÖTLUSE AUTOMATISEERIMISE METOODIKA KÜSIMUSI

Esitanud K. Habicht

A. KALLISAAR. НЕКОТОРЫЕ ВОПРОСЫ МЕТОДИКИ АВТОМАТИЗАЦИИ ОБРАБОТКИ ДАННЫХ

A. KALLISAAR. EINIGE METHODISCHE PROBLEME ZUR AUTOMATISIERUNG DER DATENVER-
ARBEITUNG

Elektronarvuti kasutamine massiliste andmetöötlusülesannete lahendamiseks juurdub üha enam meie igapäevase elu kõigil juhtimistasandil. On juba välja kujunenud põhiliselt lahendatavate ülesannete ring ning omandatud teatud kogemusi arvuti kasutamisel. See võimaldab teha mõningaid üldistusi ja anda soovitusi selle töö hõlpsamaks ja efektiivsemaks muutmiseks.

1960-ndate aastate teisel poolel, kui elektronarvuteid hakati majandusliku informatsiooni töötlemisel massiliselt kasutama, valitses andmetöötlusülesannete püstituse metoodikas kaks diametraalselt vastandlikku seisukohta: 1) globaalne ehk süsteemitehniline lahendus ja 2) süsteemi integreerimine lokaalsetest üksiklahendustest. Esimese meetodi sisuks on automatiseeritud juhtimissüsteemide (AJS) loomine ühtse tervikuna, kus kõik esinevad informatsiooni töötlemise loogilised situatsioonid lahendatakse süsteemi kui terviku vajadustest lähtudes. AJS projekteeritakse üksikutest tüüpelementidest (tüüpilistest üksiklahendustest), mida täiendatakse konkreetse majandusüksuse spetsiifiliste ülesannetega. Teise meetodi puhul alustatakse üksikülesannete lahendustest ja automatiseeritud juhtimissüsteem saadakse üksiklahenduste ühendamise teel. Seega on lõppeesmärk mõlemal juhul sama, erinev on vaid selle saavutamise viis.

Aeg on nende mooduste vahel valiku teinud. On selgunud, et AJS-i loomine ühtse ülesandena on teostamatu, sest on võimatu korraga fikseerida kõiki mõeldavaid informatsiooni töötlemise loogilisi situatsioone terve majandusüksuse piires — need ju muutuvad ja uuenevad pidevalt ega esine kõik korraga. Seetõttu tuli ka AJS-i projekteerimist ja juurutamist alustada ikkagi üksikülesannete lahendamisest. Enamiku AJS-ide loomine ongi käesoleval ajal üksikute tüüpülesannete lahenduste kasutamise staadiumis. Süsteemi kui terviku juurutamiseni pole veel jõutud, on olemas vaid AJS-i üldised kontuurid tehnilise ülesande tasemel. Süsteemi kokkumonteerimine (üksikülesannete ühendamine) aga eeldab üksikülesannete pikaajalist ekspluatatsiooni lahenduste lõplikuks stabiliseerumi-

seks. Seega on süsteemitehnilise lahenduse meetod praktikas juba muutunud süsteemi integreerimiseks lokaalsetest üksiklahendustest. Ka viimati mainitud viisil pole seni praktikas kaugemale jõutud mõningate massiliste andmetöötlaste tüüpülesannete väljatöötamisest ja kasutamisest.

Viimasel ajal on andmetöötlaste automatiseerimise meetodikas välja kujunenud kaks suunda: 1) alustatakse andmebaasi projekteerimisest, silmas pidades konkreetseid ülesandeid; need püstitatakse siis, kui on juba loodud andmepangad; 2) alustatakse ülesannetest ja johtuvalt konkreetse ülesande vajadustest projekteeritakse sellele vastavad failid. Kummalgi suunal on omad eelised ja ka puudused. Kui alustada andmebaasi kujundamisest, seostamata seda lahendatavate ülesannetega, võib küll saavutada informatsiooni parima organisatsiooni, kuid andmebaas ei tarvitse ülesannetega sobida ja andmepanga kasutamine pole kuigi tõhus. Sel juhul pikeneb tunduvalt ka lahenduste rakendamine. Kui alustada ülesannetest, võib hiljem, ülesannete seostamisel, osutada raskeks ühtsete infovoogude kujundamine. Andmebaasi kasutamine on aga sel juhul maksimaalselt efektiivne. Arvutis pole üleliigset (mittekasutatavat) informatsiooni, ka on lahenduste rakendamine tunduvalt kiirem, mistõttu suureneb saadav majanduslik efekt.

Nii on see üldjuhul. Autori arvates tuleks andmetöötlaste automatiseerimise meetodi valikul ennekõike arvestada lahendatava ülesande (või ülesannete kompleksi) tüüpi. Andmetöötlaste valdkonnad võib nende spetsiifikast lähtudes jaotada dunaamilisteks ja staatilisteks süsteemideks. Dunaamiline süsteem on näiteks majandusüksus — ettevõtte, ametkond, koondis, ministereium vms. Sellisele süsteemile on iseloomulik informatsiooni pidev muutumine ning üha uute ja erinevate informatsiooni töötlemise loogiliste situatsioonide esinemine. Staatilise süsteemi näitena võiks tuua statistilist vaatlust (rahvaloendust vms.). Staatilises süsteemis informatsioon ei muutu ning selle töötlemise loogilisi situatsioone esineb lõplik arv kordi. Automatiseerimise meetodika valikul tulekski dunaamilise süsteemi projekteerimisel alustada ülesannetest, staatilise süsteemi puhul aga andmebaasi projekteerimisest.

Andmetöötlaste ülesannete lahendamise meetodikas võib soovitada järgmist tööde järjekorda:

- 1) ülesande uurimine;
- 2) väljundinformatsiooni projekteerimine;
- 3) sisendinformatsiooni projekteerimine;
- 4) lahenduse etappide ja nende täitmise järjekorra määramine plokk-skeemi tasemel;
- 5) lahenduse detailse loogilise skeemi koostamine;
- 6) programmide ja juhendmaterjalide koostamine;
- 7) programmide silumine ja tulemuste analüüs;
- 8) lahenduse eksperimentaalne juurutamine, kusjuures paralleelselt säilib käsitsitöö;
- 9) algoritmide ja programmide täiustamine (parandamine);
- 10) töö vormistamine ja lõplik juurutamine.

Selline tööde järjekord õigustas end igati NSVL TA Majandusmatematika Keskinstituudi Eesti Filiaali andmetöötlaste automatiseerimise laboratooriumi praktikas. Ülesande uurimise käigus piiritletakse ülesande ulatus, sisu ja seosed väljapoole. Seejärel määratakse kindlaks ja kooskõlastatakse tellijaga soovitava väljundinformatsiooni maht, sisu, kuju ja sagedus. Väljundinformatsioonist lähtudes tehakse kindlaks vajalik sisendinformatsioon ja selle klassifikatsioon. Kui ka sisendinformatsioon

on kooskõlastatud, asutakse lahendust projekteerima. Iga ülesande lahendamisel eksisteerib mingi üksiktehete (tegevuste) loogiline järjekord, mis tuleb esmalt määrata ja vormistada plokk skeemi kujul. Seejärel tuleb projekteerida ülesande täpne lahendus skeem, mis sisaldab võimalikult kõik esineda võivad informatsiooni töötlemise loogilised situatsioonid ja nende lahendamise käigu. See vormistatakse matemaatiliselt formaliseeritud kujul ja ta interpreteerib arvuti töö käiku ülesande lahendamisel. Viimase alusel koostatakse arvutile tööprogrammid. Lihtsamatel juhtudel võib loogilised skeemid ka ära jätta ning koostada programmid plokk skeemi alusel.

Dünaamilistes süsteemides on andmetöötlus ülesandeid võimalik lahendada kahel printsiibil:

1) miinimumvariant — arvutis interpreteeritakse sama meetodikat, mis on kasutusel andmete käsitsi töötlemisel. Muudetakse ainult andmete töötlemise tehnikat — käsitsitöö asendatakse masinatööga. Eesmärk on vähendada töömahukust, tõsta tööviljakust ja tulemuste kvaliteeti ning nende saamise kiirust;

2) maksimumvariant — ülesande lahenduse projekteerimisel nähakse ette peale töötlemistehnika muutmise ka meetodika muutmine.

Miinimumvarianti on kergem projekteerida ja juurutada, sest meetodika on olemas. Nõrgaks küljeks on aga see, et ei kasutata ära kõiki arvuti võimalusi. Maksimumvariant kasutab küll täielikumalt arvuti võimalusi, kuid lahendust on raske, mõningal juhul isegi võimatu juurutada. Pole ju võimalik üldisest planeerimise ja arvestuse süsteemist, mis on metodoloogiliselt ühtne tervik, mõnda osa välja võtta ja lahendada täiesti erineva meetodikaga, säilitades teiste, sama ülesandega seostuvate osade olemasolev meetodika. Autori praktikas pole meetodika muutmisega ülesandeid õnnestunud juurutada. Siit tuleneb järeldus: dünaamiliste süsteemide andmetöötlus ülesandeid tuleks lahendada iteratiivselt, pidevalt täiustuva protsessina. Alustada tuleks miinimumvariantist ja töö käigus pidevalt täiustada lahendust, muutes võimaluse piires ka meetodikat. Nii stabiliseerub lõpuks lahendus, mis sisaldab mõlema printsiibi elemente. Meetodika muutmine võib kõne alla tulla alles süsteemi kui terviku lahendamisel arvutil. Seega pole õige dünaamiliste süsteemide andmetöötlus ülesannete lahenduste projekteerimisel kohe esimesel lähendusel taotleda lõplikku varianti. See peaks kujunema kasutamise kestel. Küll aga tuleb programmeerimisel valida sellised vahendid, mis võimaldavad programme suhteliselt kergesti järk-järgult täiustada.